The University of Nottingham

Department of Mechanical, Materials and Manufacturing Engineering

Analysis of Flexible Fabric Structures

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

by

Andrew James Pimm

July 2011

Abstract

This thesis is primarily aimed at carrying out analysis of Energy Bags, reinforced fabric bags used for subsea compressed air energy storage. Subsea compressed air energy storage is a completely new method of large-scale energy storage designed to be integrated with direct-compression offshore wind turbines and wave energy converters. Energy Bags are impermeable bags anchored to the seabed at significant depths (e.g. 500m) in which high pressure air, compressed by specially designed wind turbines and wave energy converters, is stored at pressures roughly equal to the hydrostatic pressure of the surrounding water. Energy Bags do not need to be particularly strong because most of the reaction to the pressure load is provided by the surrounding water, and high energy densities are available at such depths as 500m.

This thesis investigates the deformed shapes of Energy Bags and studies optimal designs. Three analysis methods are developed which vary in their complexity, ease of use, and accuracy. First, a system of coupled ordinary differential equations (ODEs) is derived which describes the deformed shape of an axisymmetric Energy Bag. This model is later used in an optimisation study to find the shapes of bag which minimise the cost of materials (reinforcement, fabric, and ballast) per unit of energy stored. Circumferential reinforcement, hanging masses from the inside of the bag (which it was hoped would lower the total cost) and fill level are all included as variables in the optimisation, and it is found that for reasonable materials costs an Energy Bag could cost less than £10,000/MWh when anchored at 500m. This compares favourably with all other methods of large-scale energy storage. However, the bags used in the optimisation study have wide bases, which will require sealing against the seabed (unless water is to be allowed into the bags). Problems are encountered when trying to use the ODE method to find the shapes of partially inflated bags, and it is generally not very easy to use.

Next, we carry out finite element analysis (FEA) of an axisymmetric Energy Bag using cable elements. This is much more user-friendly and flexible than the ODE method. Partially inflated bag shapes are found, and pressure-volume curves are presented which show the almost isobaric performance of an Energy Bag. It is found that material mass limits the extent to which the bag can be deflated before it becomes unstable. The axisymmetric FEA is used to study bags with much more realistic circumferential reinforcement than the ODE method, and we also look at bags with an unsealed base, which allow water in through the base as they deflate.

A three-dimensional FEA tool is presented which models an Energy Bag as a cable-reinforced membrane using cable and membrane elements, and special measures had to be taken to deal with wrinkling. We assume that the bag is rotationally symmetric, comprising a number of symmetric lobes. The 3D FEA is used to find the stress distribution in the membrane of the bag, however a converged solution cannot always be found. It is not certain why this is the case but it is anticipated that it is because deformed bags are not always rotationally symmetric. The 3D FEA could also be used to model other membrane structures such as balloons, parachutes, roofs and sails, as well as nets.

The standard cutting patterns for lobes in lobed balloons are analysed, and a new cutting pattern known as the Constant Tension lobe is generated. This is an extension of the Constant Radius lobe and takes into account the pressure gradient found in both air and water, minimising waste material. The Constant Tension lobe is particularly appropriate for Energy Bags because of the large pressure gradient in water. The Ultra High Performance Vessel architecture is also presented, upon which the design of the prototype Energy Bags is based. The fabric structure of an Ultra High Performance Vessel comprises only two sheets of fabric (rather than many separate lobes welded together), and tendon shortening and "bellows" serve to ensure that there is no meridional stress in the fabric.

An analytical optimisation is used to show that the zero pressure bag that minimises cost of materials per unit of energy stored has equal costs of reinforcement and membrane. The axisymmetric FEA is also used to find the optimum bag size and maximum fill level for a bag which comes down to a single point at the base (as opposed to a wide base bag).

Finally, testing of two 1.8m diameter superpressure Energy Bags has been commenced during the course of this work, and the prototypes and test rig are documented in this thesis. The prototypes were manufactured for us by Thin Red Line Aerospace Ltd., a Canadian manufacturer of deployable fabric structures for use in space. They are being cycled back-to-back in order to prove the concept, assess the performance of an Energy Bag over time, and identify any problems that need to be addressed. One of the bags had a few small leaks from the moment it was first inflated, but the other has remained airtight to date. It was found that if an Energy Bag is to be airtight, special attention must be paid to the welds at the seams and the sealing around the airline fittings.

Acknowledgements

First and foremost I would like to thank my supervisor, Professor Seamus Garvey, for guiding me through the project with such vigour and expertise, and Dr Richard Drew of E.ON New Build and Technology for providing valuable support and ensuring that the project has progressed on schedule. I would also like to thank my internal assessor, Professor Adib Becker, for giving useful feedback at my progression assessments, and Maxim De Jong of Thin Red Line Aerospace for showing such enthusiasm for the idea, manufacturing the prototypes, and guiding me to some very useful literature. Dr Cosimo Buffone and the technicians in the university laboratories provided help with setting up the testing that is very much appreciated, and my associate supervisor, Dr Mike Wood, has also been supportive.

I would also like to thank Gran for giving me so much help with maths and physics at school and for getting me interested in renewable energy, and Jenny, Mum and Dad, and all my family and friends for their love and support.

Contents

1 Introduction 1

1.1 Fossil Fuels and their Drawbacks 1
1.2 Energy Storage and ICARES 5
1.3 Fabric Structures 16
1.4 Aims and Scope 17
1.5 Layout of the Thesis 18

2 Literature Review 20

2.1 Energy Storage 20
2.1.1 Compressed Air 21
2.1.2 Pumped Hydro 30
2.1.3 Batteries 32
2.1.4 Flow Batteries 33
2.1.5 Flywheels 34
2.1.6 Superconductors (SMES) 35
2.1.7 Hydrogen 36
2.1.8 Thermal Energy Storage 37
2.2 Underwater Inflatables and Water Load Test Weights 39
2.2.1 Lifting Bags 40
2.2.2 Water Load Test Weights 46
2.2.3 Correspondence with Manufacturers 49
2.3 Ballooning 50
2.3.1 Early Days of Ballooning 50
2.3.2 Scientific Ballooning and Superpressure Balloons 52
2.3.3 Wrinkling of Membranes 61
2.3.4 The Young-Laplace Equation 63
2.4 Techniques in Nonlinear Finite Element Analysis 64
2.4.1 The Newton-Raphson Method 64
2.4.2 Load Incrementation 67
2.4.3 Displacement Control 67
2.4.4 Boundary Nonlinearity 68
2.5 Notes on the Design of Energy Bags 68

3 Modelling Axisymmetric Structures Using Coupled ODEs 71

3.1 Derivation of the Coupled Ordinary Differential Equations 71
3.2 Solving the Ordinary Differential Equations 74
3.3 Use of the Model 76

4 Finite Element Analysis of Axisymmetric Structures 81

4.1 The Cable Element 81
4.1.1 Force Vector 82
4.1.2 Stiffness Matrix 83
4.1.3 Wrinkling 85

4.2 Circumferential Reinforcement 88
4.2.1 Force Vector 88
4.2.2 Stiffness Matrix 90
4.3 Bending 91
4.3.1 Force Vector 91
4.3.2 Stiffness Matrix 93
4.4 Loading 94
4.4.1 Load Vector 94
4.4.2 Load Stiffness Matrix 96
4.5 Solving the Equilibrium Equation 97
4.5.1 Constraints Matrix 100
4.5.2 Seabed Resistance Forces 101
4.5.3 Convergence Criteria 104
4.6 Results 105
4.6.1 Natural Shape Energy Bags 105
4.6.2 Energy Bags with Circumferential Reinforcement 111
4.6.3 Wide Base Natural Shape Energy Bags 113
4.6.4 Energy Bags with an Unsealed Base 115

5 Lobed Fabric Structures 119

5.1 Constant Angle and Constant Radius Lobes 119
5.2 Ultra High Performance Vessel Architecture 121
5.3 Membrane Stresses 125
5.4 The Constant Tension Lobe 129

6 Three-Dimensional Finite Element Analysis 136

6.1 The Membrane Element 136
6.1.1 Force Vector 136
6.1.2 Stiffness Matrix 140
6.1.3 Wrinkling 141
6.2 Loading 144
6.2.1 Load Vector 144
6.2.2 Load Stiffness Matrix 147
6.3 Solving the Equilibrium Equation 148
6.4 Results 149

7 Optimisation 156

7.1 Analytical Cost Minimisation of a Natural Shape Bag 157
7.2 Optimisation of a SPA Natural Shape Bag 162
7.2.1 Meridional Reinforcement 163
7.2.2 Ballast 166
7.2.3 Surface 167
7.2.4 Summary 170
7.3 Optimisation of a Wide Base Natural Shape Bag 172
7.3.1 Description of the Method 172
7.3.2 Results 175
7.4 Specification for a 0.2GWh Energy Bag 178

8 Testing 182

8.1 Design of the 1/10th Scale Prototypes 183
8.2 Installation and Control of the 1/10th Scale Prototypes 186
8.3 Performance of the 1/10th Scale Prototypes 191
8.4 Plans for the 1/4th Scale Prototypes 191

9 Conclusions and Future Work 193

9.1 Contributions of Present Research 193
9.2 Future Work 196
9.3 Future of ICARES 198

References 199

Appendices 212

A. Matlab Code for Axisymmetric ODE Method 213
B. Matlab Code for Axisymmetric FEA 219
C. Matlab Code for 3D FEA 250
D. Matlab Code for Valve Control 288

Symbols

Roman

a,b,c	Constants; Node position in deformed bag in x,y,z directions, respectively; Vectors forming sides 12, 13, and 23 of the deformed membrane element
A	Cross-sectional area of the cable element; Swept area of the deformed cable element; Area of the deformed membrane element
C_{ba}	Cost of ballast
C_{bm}	Cost of bag materials
C_{mr}	Cost of meridional reinforcement
C_{su}	Cost of surface
d	Constant; Depth of the base of the bag
D	Elastic matrix; Diameter
D_{c}'	Modified elastic matrix (for a wrinkling element, in the principal coordinate system)
D_{max}	Maximum diameter of the bag
$D_{\mathrm{max}}{}^{*}$	Optimum diameter
E	Stored energy; Young's modulus
f	Objective function; Force
F	Element force vector; Element load vector
F_{bh}	Differential pressure force acting across the upper bulkhead
F_{buoyancy}	Buoyancy force
F_{dp}	Element differential pressure force vector
F_{sb}	Seabed resistance force
g	Standard gravity
h	Height; Height above the base of the bag
I	Moment of inertia
J_r	Jacobian of force residuals
k	Constant; Variable in equation for F_{sb}
K	Stiffness matrix
K_{e}	Elastic stiffness matrix
K_{g}	Geometric stiffness matrix
L	Meridional length; Length of element in deformed configuration
L_0	Length of element in undeformed configuration
m_{bh}	Mass of the upper bulkhead
M	Molar mass
n	Number of tendons/lobes
$\hat{n}$	Unit normal to the element
n_w	Downward force per unit of projected area onto the seabed
p	Differential pressure across the bag (= P_{int} - P_{ext})
p_0	Differential pressure at the base of the bag
$\hat{p}$	Unit parallel from node 2 to node 1
P	Pressure; Power
r	Ratio between storage pressure and atmospheric pressure; Local radius of the bag; Base radius; Distance from axis of rotation; Force residual
$\hat{r}$	Dimensionless force residual
r_1	Radius of curvature in the direction of maximum curvature
r_2	Radius of curvature in the direction of minimum curvature
R	Universal gas constant; Load stiffness matrix; Reuter's matrix
s	Distance along the meridian from the top
S	Undeformed area of the membrane element
t	Thickness of the membrane element; Metric ton

T	Total tension in all of the meridional cables; Absolute temperature; Tension in a cable element; Transformation matrix; Torque
T_1	Meridional tension per unit length in the direction of maximum principal curvature
T_2	Meridional tension per unit length in the direction of minimum principal curvature
T_c	Circumferential tension (per unit meridional length in Chapters 3, 5 and 7)
T_m	Meridional tension (per unit circumferential angle where noted)
$T_{m,lo}$	Meridional tension in the cable element below the current node
$T_{m,up}$	Meridional tension in the cable element above the current node
u,v,w	Nodal displacement in x,y,z directions, respectively
u	Vector of nodal displacements
V	Volume
w	Height of the bottom of the air pocket above the base of the bag
w_{bh}	Weight of the upper bulkhead
W	Work
W_{adiab}	Work done in an adiabatic process
W_{isoth}	Work done in an isothermal process
x,y,z	Node position in undeformed bag in Cartesian coordinates
z	Height

Greek

α	Angle between the normal to the meridian and the vertical; Penalty function
β	Angle between the base of the bag and the seabed
ε	Strain scalar/vector ($\varepsilon_x,\varepsilon_y,\gamma_{xy}$)
ε'	Rotated strain vector
ϕ	Load fraction
γ	Heat capacity ratio; Displacement factor; Structural capacity
λ	Very small multiplication factor used in cable element wrinkling; Small height above the seabed used in seabed resistance forces
μ	Penalty parameter
ν	Poisson's ratio
θ	Angle around the circumference of the bag
θ_{p}	Angle between local coordinate system and principal stress coordinate system
ρ	Density; Mass distribution
ρ_a	Density of air
ρ_w	Density of water
σ	Stress scalar/vector ($\sigma_x,\sigma_y,\tau_{xy}$)
σ'	Rotated stress vector
σ_0	Prestress scalar/vector
σ_1,σ_2	Maximum and minimum principal stresses, respectively
σ_{c}	Compressive stress
σ_{vM}	von Mises stress
ω	Angular velocity

Acronyms

A-CAES	Adiabatic Compressed Air Energy Storage
AFCRL	Air Force Cambridge Research Laboratories
ALB	Air Lift Bag
boPET	Biaxially-oriented Polyethylene Terephthalate
CA	Constant Angle (lobe)
CR	Constant Radius (lobe)
CT	Constant Tension (lobe)
CAES	Compressed Air Energy Storage
CCS	Carbon Capture and Storage
CH_4	Methane
CIA	Central Intelligence Agency
CO_2	Carbon Dioxide
DECC	Department of Energy and Climate Change
EC	European Commission
EIA	US Energy Information Administration
EU	European Union
FEA	Finite Element Analysis
FESA	Fully Enclosed Single Attachment
GHOST	Global Horizontal Sounding Technique
GPS	Global Positioning System
HSE	Health and Safety Executive
IBU	Inflatable Buoyancy Unit
ICARES	Integrated Compressed Air Renewable Energy System
ICWT	Integral Compression Wind Turbine
IMCA	International Marine Contractors Association
IoP	Institute of Physics
IPCC	Intergovernmental Panel on Climate Change
ISEP	Iowa Stored Energy Park
LLDPE	Linear Low Density Polyethylene
Matlab	A numerical computing environment and programming language
N_20	Nitrous Oxide
NASA	National Aeronautics and Space Administration
OCAES	Offshore Compressed Air Energy Storage
ODE	Ordinary Differential Equation
OPEC	Organization of the Petroleum Exporting Countries
PV	Solar photovoltaic power
PVC	Polyvinyl Chloride
RF	Radio-Frequency
RHI	Renewable Heat Incentive
ROC	Renewables Obligation Certificate
ROV	Remotely Operated Vehicle
RPI	Retail Price Index
SMES	Superconducting Magnetic Energy Storage
SPA	Single Point of Anchorage
TRL	Thin Red Line Aerospace Ltd.
UHPV	Ultra High Performance Vessel
ULDB	Ultra Long Duration Balloon
UNFCCC	United Nations Framework Convention on Climate Change
UV	Ultraviolet
VAT	Value Added Tax
ZPNS	Zero Pressure Natural Shape

Chapter 1

Introduction

1.1 Fossil Fuels and their Drawbacks

Energy demand has roughly doubled in the last 30 years, and is set to increase by a further 50% in the next 25 years [1]. This rise in demand is driven by two key factors: a growing population and greater prosperity. The International Energy Agency predicts that 70% of the increase in energy demand over the next 20 years will come from developing nations and that China, still considered a developing nation, will account for 30% of the overall increase.

Today, humans exploit many different forms of energy, however fossil fuels (coal, oil, and natural gas) dominate the mix; they account for about four-fifths of total energy use, a share that has remained largely unchanged for around a century. These fuels dominate because they are relatively cheap, abundant, and have high energy densities. The various energy sources available tend to be suited to particular purposes, for example the transport sector relies almost exclusively on oil, while heating is largely fuelled by a mixture of oil, gas, wood, and coal. For electricity generation, fossil fuels contribute about two-thirds of the energy use, with nuclear and hydro providing most of the rest, while renewable energy sources only provide around 2%.

Fossil fuels, while having many benefits, have two main downsides: they are limited in supply, and because some of the by-products of fossil fuel combustion are greenhouse gases, they are the fundamental cause of global warming and ocean acidification. Greenhouse gases, such as carbon dioxide (CO_2), methane (CH_4), and nitrous oxide (N_2O), are gases found in the atmosphere which absorb and re-emit solar energy that has reflected from the surface of the Earth in a process called the greenhouse effect. The greenhouse effect is essential for life on Earth as we know it: greenhouse gases have a large effect on the temperature of the Earth's surface, and without them the global mean surface air temperature would be around -19°C, approximately 33°C lower than the actual mean temperature of 14°C [2]. However, since the industrial revolution (1750[1]) the atmospheric concentration of

[1] The IPCC uses the year 1750 as the start of the industrial revolution.

the three major long-lived greenhouse gases has increased by an unprecedented level: CO_2 concentration has risen by about 36% (100ppm), N_2O concentration has risen by about 18%, and CH_4 concentration has risen by about 148% [3]. This is not in keeping with any previous change; in the 10,000 years before 1750, the atmospheric concentration of CO_2 had been relatively stable between 260 and 280ppm [4]. Most worryingly, the growth rate of CO_2 concentration has also increased substantially over the past 250 years: the first half of the 100ppm increase in CO_2 concentration from 1750 was reached by 1970, taking 220 years; the second half took place over the last 40 years [5]. Fossil fuel combustion (along with contributions from cement manufacture) is responsible for over 75% of the increase in atmospheric CO_2 concentration since 1750 [6], and the IPCC claim that "it is *extremely likely*[2] that humans have exerted a substantial warming influence on climate." [7]

Global mean surface temperatures rose over the period 1906–2005 by 0.74°C (0.56°C to 0.92°C) [8], and are predicted to rise by a further 1.1–6.4°C during the 21st century, depending on worldwide greenhouse gas emissions [9]. Global warming causes three major problems: sea level rise caused by the melting of the Antarctic and Greenland ice sheets, extreme weather patterns, and drought. Unfortunately it is likely that the effects will be greater in poorer areas of the world, particularly the tropics and subtropics.

Increased concentration of atmospheric CO_2 also causes ocean acidification. In the past, carbon released from weathered rocks on land has dissolved into the sea at such a slow rate that it is deposited and buried on the sea floor without a rise in ocean acidity. However, anthropogenic emissions of CO_2 dissolve into the ocean at a higher rate than the carbon can be deposited onto the seabed, increasing the hydrogen ion (H^+) concentration in the ocean and so increasing its acidity. It is estimated that H^+ concentration has risen by 29% since the start of the industrial revolution, and could rise by a further 41% on pre-industrial levels by 2050 [10]. It is anticipated that coral and other carbonate structures will be vulnerable to dissolution, and that it will become difficult for planktonic organisms, key components of the food chain, to form shells. The impacts of ocean acidification will appear first in the Southern Ocean because cool water (such as that in the Southern Ocean) absorbs more CO_2 than warm water.

The second problem with a dependence upon fossil fuels is their limited supply. The US Energy Information Administration calculated that, at current extraction rates, world recoverable coal reserves (estimated at 929 billion short tons) would last 132 years [11]. Taking an average annual growth in coal consumption of 2% into account [12], the reserves would be depleted in only 65 years. At current consumption (84.34 million bbl/day, 2008 est.), proved oil reserves (1.365 trillion bbl, 1st Jan. 2009 est.) will last 44 years and proved

[2] The IPCC uses '*extremely likely*' to represent a 95% confidence level or higher.

natural gas reserves (182.1 trillion m^3), at current consumption (3.073 trillion m^3/year), will last 59 years [13].

There are several measures that can be taken to combat global warming and these fall into the following broad categories: reducing or even stopping/reversing deforestation (rainforests are valuable carbon sinks); improving the efficiency of machinery, appliances, and buildings; implementing carbon capture and storage (CCS); and moving away from a dependence on fossil fuels by making use of renewable energy sources such as solar, wind, and hydro, and lower carbon energy sources such as nuclear power. Improving efficiency and moving away from fossil fuels are also measures that would deal with the problem of limited fossil fuel reserves.

Another issue which should not be ignored is that of energy security. The uneven distribution of fossil fuels around the world means that some countries have much more power (in both senses of the word) than others. 79.6% of world crude oil reserves are located in the 12 OPEC Member Countries, 70.1% of the OPEC reserves being in the Middle East [14], and most of the recoverable coal reserves are located in the northern hemisphere. However, renewables are well distributed around the world and so by making use of their own natural renewable resources, countries can increase their resistance to political instability, accidents, natural disasters, and terrorist attacks.

In efforts to lower our reliance upon fossil fuels, governments around the world are setting ambitious targets for generation from renewables sources. The Kyoto Protocol, a protocol of the United Nations Framework Convention on Climate Change (UNFCCC), is an international treaty committing countries to stabilise atmospheric concentrations of greenhouse gases at a level that will prevent dangerous global warming. It was adopted in December 1997 and entered into force in February 2005, and has so far been ratified by 188 countries (though the USA is notable in having not ratified the treaty). The Kyoto Protocol requires countries to reduce their greenhouse gas emissions through national measures and three market-based mechanisms that were set up (which include emissions trading), and places a heavier burden on developed countries, recognising that they are primarily responsible for the current high levels of greenhouse gases. Specifically, the Kyoto Protocol requires the EU and the ratifying Annex I countries of the UNFCCC to reduce their collective greenhouse gas emissions by 5.2% from 1990 levels by 2012. There are 40 Annex I countries including many members of the EU, Australia, Russia, Japan, and the USA.

The EC has set a target of 20% of the EU's energy to come from renewables by 2020 ("20% by 2020") [15], though the UK, lagging far behind other EU Member States, has a lower target of 15% of energy to be supplied by renewables by 2020. This will require a seven-fold increase from 2008 levels, a higher increase than any other Member State, and will require around £100 billion of new investment, creating up to half a million jobs [16].

The UK is enforcing its target through the Renewables Obligation Order 2002 and Renewables Obligation Certificates (ROCs), placing an obligation on UK-licensed electricity suppliers to source an increasing proportion of their electricity from renewables sources. Suppliers that do not meet their required obligations are liable to pay charges equivalent to their deficit (at a set "buyout price" per ROC) into a fund which is distributed on a pro-rated basis to the suppliers that have met their obligations. This is not only an effective method of ensuring that we meet the targets set for generation from renewables, but by distributing the fund to the companies that do meet their obligations, it also serves to incentivise generation from renewables above the required levels. The buyout price is set each year, and has been set at £36.99 per ROC for the 2010-2011 compliance period. It was £30 per ROC in the base year, 2002-2003, and was £37.19 per ROC in 2009-2010 [17]. The government intends that the Renewables Order will be in place until at least 31st March 2037 [18].

Originally 1 ROC was awarded per MWh generated from renewables, no matter what generation technology was used, however since April 2009 the value of the ROC has been banded depending upon the generation type. Offshore wind turbines installed between 1st April 2010 and 31st March 2014 now receive 1 ROC per 0.5MWh generated, effectively doubling the value of offshore wind energy [19]. The Department of Energy and Climate Change (DECC), who have decided that offshore wind energy will need to be used to meet a large part of our 15% target for 2020, increased the value of offshore wind in the ROC banding after commissioning a study by Ernst & Young which suggested a value of 2-2.5 ROCs per MWh [20].

Feed-in tariffs (and the Renewable Heat Incentive in the UK) are being used to incentivise renewables generation at smaller scales (all renewable electricity generating projects below 5MW are eligible for the UK's Feed-In Tariffs programme). In the UK's scheme, which only went live on 1st April 2010, a generation tariff is paid for every kWh of electricity produced from renewables; the generation tariff depends upon the technology and the system size, providing the highest payments to the most expensive technologies and the smallest systems [21]. The generation tariff received will last for the tariff lifetime (20 years for wind and hydro, 25 years for solar PV), and be adjusted annually for inflation. An export tariff is also paid for surplus energy exported to the grid; the export tariff is independent of the technology and system size [22]. The export tariff currently stands at 3p/kWh and is also linked to inflation, but generators are free to opt-out of this fixed price and try to negotiate a better deal with an electricity supplier. In the years before Smart Meters are widely used, the export level will be deemed to be 50% of the system power, but those who believe that they export significantly more than this can install export meters and be paid according to the metered export level.

1.2 Energy Storage and ICARES

Energy demand fluctuates throughout each day, week (demand is higher during the working week than at weekends), and year (the demands of building-heating rise during the winter, and in hot countries the demands of air conditioning rise during the summer). These fluctuations can all be seen in figure 1.1, which shows the energy demand in Great Britain over three weeks in January 2006 and three weeks in June 2006.

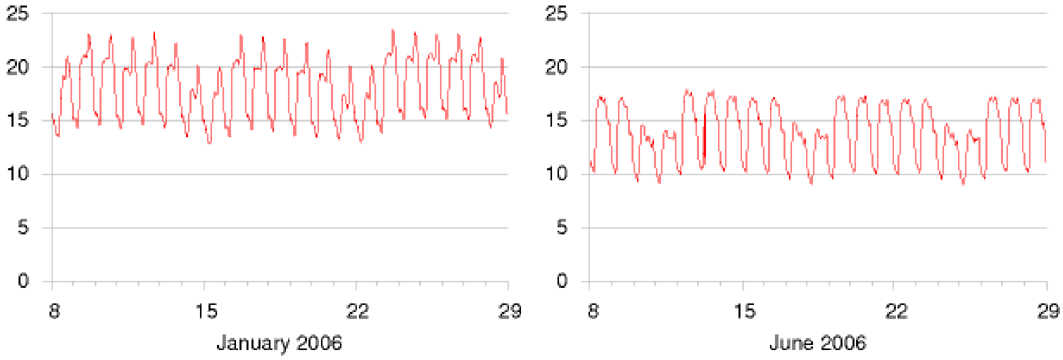


Fig. 1.1 Electricity demand in Great Britain in kWh/day per person over three weeks in January 2006 and three weeks in June 2006 [23]

Currently, power stations are split into three different classes depending on how they are used to meet daily fluctuations in demand. From most efficient through to least efficient, these are: baseload, intermediate, and peaking plants. Baseload power stations are used to meet baseload demand, the minimum power demand required by the power company's customers (typically 35-40% of the maximum load during the year). Coal-fired and nuclear power stations are used for this purpose, and are operated continuously since it is most economical to operate them at constant production levels. Intermediate load-following power stations adjust their output as electricity demand fluctuates through the day, and include gas turbine, steam turbine, and hydroelectric power stations. Peaking power stations are operated at periods of peak demand (a few hours a day at most) and include gas turbine and hydroelectric plants. Baseload power stations are generally the most efficient of the three classes because they are run for so long, and peaking power stations are the least efficient because they are only operated for short periods.

One of the good things about fossil fuels is that the power they provide is reliable – as long as the fuel is ready to be burned, useful energy is soon available. However, many sources of renewable energy (including wind, solar, and wave power) are intermittent and rarely load-following [24], and so in a future where a large amount of our energy supply comes from renewables, we will have to be able to deal with (perhaps unexpected) periods of low wind, wave, and solar power. Even if we move from fossil fuels to being fully reliant on nuclear power, there will still be problems in meeting demand fluctuations because nuclear power stations are not usually designed to be quickly turned off and on: they tend to be kept on all the time, and the power they deliver can only be ramped up or down over a period of several hours.

Essentially, wind power fluctuations can be split into two different classes: short-term slews and long-term lulls [25]. Short-term changes ("slews") happen quite regularly: in 2007, the UK Government set out its plans to expand Britain's offshore wind capacity to 33GW by 2020 [26], and by scaling up a typical slew in the Republic of Ireland in 2007, MacKay calculates that if 33GW capacity is reached (delivering 10GW on average, about a quarter of the UK's current average electricity usage), we can expect to have occasional slew rates of 3.7GW per hour. This is equivalent to 4 nuclear power stations going from no power to full power in the space of an hour. He also shows that every morning, between 6.30am and 8.30am, British electricity demand grows by about 6.5GW per hour. Long-term lulls in wind power are less frequent, but must still be dealt with. At the start of February 2007, the Republic of Ireland had a country-wide lull (total output less than 15% of total rated wind capacity) that lasted five days, and lulls of two or three days happen several times a year. It should be noted that these lulls are country-wide and not just in a particular location. MacKay calculates that if we have a wind power capacity of 33GW delivering 10GW on average, the UK would need to replace or go without 1,200GWh of energy during a five day lull.

One way of dealing with the intermittency problem brought on by renewables is to simply build more generating stations (e.g. more wind turbines) than are usually required, and if possible turn on and off the stations when necessary – wind turbines can be "turned off" by using brakes and feathering the blades. However, on a large scale, building excess capacity is very uneconomical, and there are several other methods of dealing with intermittency in a sustainable manner: increased interconnection (e.g. a European super grid), utilisation of dependable renewables (e.g. hydroelectricity, biomass, and geothermal energy), demand management (e.g. smart-charging of the batteries of electric vehicles), and energy storage, which we concentrate on here. We should also bear in mind that, in a future with high levels of wind and wave power capacity, there may be times when supply outweighs demand, particularly on windy nights, and it will be a waste to simply turn off the generators when the energy could be stored for use during periods of high demand and low supply.

We should think about how much energy to store for each MW of installed wind capacity, so that during lulls in wind we can match the year-long average power of the turbines with energy from storage. For each MW of wind capacity, assuming the turbines have a typical load factor of 30% (which is good for both onshore and offshore wind turbines in the UK), the energy required from storage over a five-day lull is

$$30\% \times 1\text{MW} \times (5 \times 24\text{h}) = 36\text{MWh}. \tag{1.1}$$

So to match the average output of a large offshore wind turbine with a capacity of 5MW with energy from storage over a five-day lull, the required storage capacity is

$$36\text{MWh/MW} \times 5\text{MW} = 180\text{MWh}. \tag{1.2}$$

Assuming that we might want to be able to store all of the energy being generated when the turbines are operating at peak power, or take energy from the store at the peak power rating of the turbines, both the input and output power capacity of the store must be equal to the total power capacity of the wind turbines attached to the store (so 5MW per large offshore wind turbine).

Even without any intermittent renewables in the system, energy storage can be used for load-levelling (as shown in figure 1.2). In a load-levelled power supply system, power stations work at a constant output (approximately the mean power demand); during periods when demand is lower than the mean, the excess generated energy is put into storage, and during periods when the demand is higher than the mean, energy is removed from storage to supplement the constant output of the power stations. With adequate storage capacities, the power system can effectively run at baseload all day and night, removing the need for costly intermediate and peaking power stations.

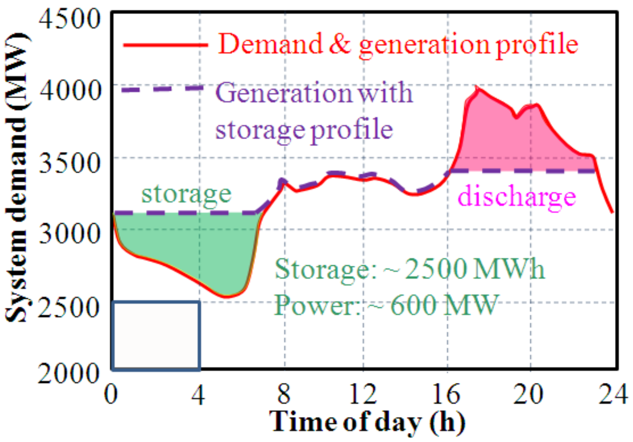


Fig. 1.2 Typical daily load profile of Scottish Power and the load-levelling abilities of energy storage [27]

There are several different forms of energy storage, and they tend to be suited to particular applications. In this work we are primarily concerned with large-scale rechargeable energy storage (also known as grid storage), of which there are currently two viable forms: pumped storage and compressed air energy storage. In a pumped storage plant, water is pumped from a low-level reservoir into a high-level reservoir during periods of low demand (typically night-time) using cheap electricity, and then during periods of peak demand the water is allowed to flow back down into the lower reservoir, driving turbines which generate electricity. Reversible pump-turbines are used, so the same machines can be used for both pumping and generating.

Pumped storage is essentially a form of hydropower; hydropower is the most established renewable energy technology, accounting for about 20% of global electricity production [28]. It has the advantages of being able to continuously respond to changes in demand, being able to start very rapidly, and being confined to mountainous areas and so mostly hidden from view. However, capital costs are high and payback periods long, as reservoirs and dams have to be built and large tunnels and caverns have to be cut through a mountain. Also, the range of locations that are suitable for pumped storage is limited and while some countries (e.g. Norway and Sweden) have many such locations, others (e.g. the Netherlands) have very few. It has been claimed that the Danes export almost all of their wind energy to neighbouring Norway and Sweden, who have hydroelectric facilities which they then turn down. The Danes later buy back electricity at a higher price during periods of low supply and high demand, which the Norwegians and Swedes can supply using hydroelectricity and pumped storage [29].

In the UK there are currently four pumped storage plants: Dinorwig and Ffestiniog in North Wales, and Cruachan and Foyers in Scotland. Details of these plants are given in table 1.1. They amount to a total storage capacity of about 27GWh and a total power capacity of about 2.8GW. Dinorwig, the largest plant in terms of both storage and power capacity, cost £0.4 billion in 1980 (about £1.3 billion in 2010 prices using the UK Retail Price Index (RPI) [30]), working out at a (2010) cost of approximately £143,000/MWh.

Station	Power (GW)	Head (m)	Volume (million m^3)	Energy stored (GWh)
Ffestiniog	0.36	320-295	1.7	1.3
Cruachan	0.40	365-334	11.3	10
Foyers	0.30	178-172	13.6	6.3
Dinorwig	1.80	542-494	6.7	9.1

Table 1.1 Pumped storage plants in the UK [31]

Clearly, the maximum power of 2.8GW and the maximum storage capacity of 30GWh are well short of the 10GW of power (on average, 33GW at peak wind) and 1,200GWh of energy storage we may need in a future in which 10GW of our electricity is generated from wind energy. There are more locations that could be used for pumped storage, but it is estimated that the UK's maximum pumped storage capacity is around 400GWh [32].

In a compressed air energy storage (CAES) plant (shown in figure 1.3), air is compressed to high pressures (e.g. 50-70 bar) during periods of low demand using a compressor powered by electricity from the grid, and stored in an underground cavern. When electricity is required from the CAES plant, the compressed air is released from the cavern and expanded to generate electricity, usually by using it in the combustion chamber of a gas turbine. Ideally, the heat generated in the compression is removed from the air before the air enters the cavern and placed into thermal storage. This heat energy can then be used later to heat the air leaving the cavern, so that the air is heated up to nearly the outlet temperature of the compressor. By storing the thermal energy generated in compression, it may not be necessary to burn gas in the expansion stage. There are currently only two large-scale CAES plants in the world (Huntorf, Germany, and McIntosh, AL, USA), and in both of these the compressed air is stored underground in excavated salt domes. Salt domes are large, naturally-formed underground domes of salt, and they are excavated by dissolving the salt in water in a process known as solution mining. Salt domes are particularly suited to CAES as salt is self-sealing under pressure [33]. Other forms of compressed air stores that have been proposed include disused mines, caves, and aquifers. It is of course possible to store compressed air in fabricated steel pressure vessels, however the cost of such a vessel makes it prohibitively expensive for large-scale CAES. All of the underground storage solutions are limited in location and (apart from aquifer storage and stores with a shuttle pond) have fixed containment volumes, meaning that the pressure of the air in the store changes quite considerably depending upon the amount of energy contained.

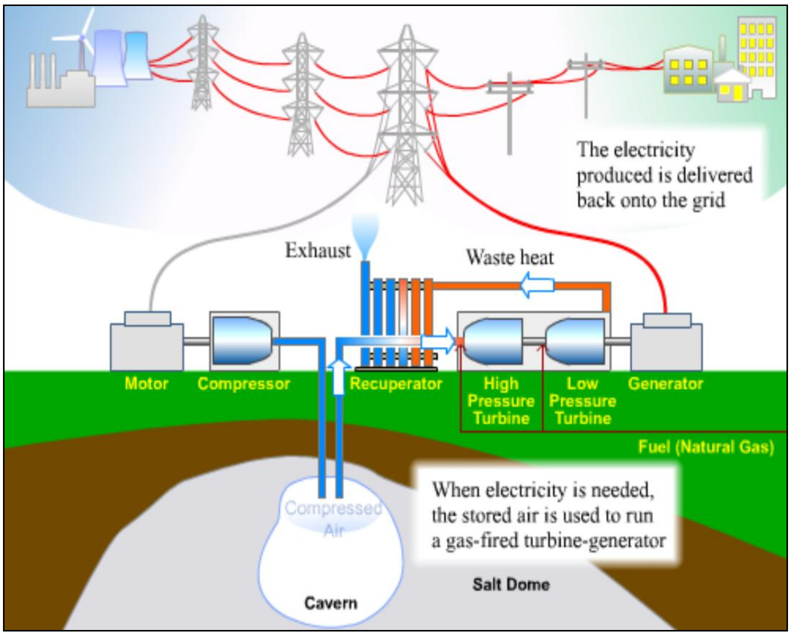

The electricity produced is delivered back onto the grid
Exhaust
Waste heat
Motor
Compressor
Recuperator
High Pressure Turbine
Low Pressure Turbine
Generator
Fuel (Natural Gas)
When electricity is needed, the stored air is used to run a gas-fired turbine-generator
Compressed Air
Cavern
Salt Dome

Fig. 1.3 Schematic of a CAES plant with two-stage expansion and a heat recuperator [34]

Professor Seamus Garvey (University of Nottingham, supervisor of this PhD) has proposed a new method of renewable energy generation and storage that makes use of proven technologies. The idea, first conceived by Garvey in 2006, is known as the Integrated Compressed Air Renewable Energy System, or ICARES. In an ICARES plant (shown in figure 1.4), renewable energy sources such as offshore wind, wave, and tidal power will be used to directly compress air to high pressures (e.g. 50-70 bar) rather than directly generate electricity, and the compressed air will either be sent immediately through an expander to generate electricity, or stored in large flexible bags (which we have termed *ENERGY BAGS*™) anchored to the seabed. If the compressed air is being placed into storage, the heat generated in compression will be extracted from it before it is pumped into the store, and stored in a thermal energy store. This stored heat can later be used to warm up the cool air entering the expander to almost the compressor outlet temperature. The thermal energy store can be topped up using solar power – it has been shown that the marginal utilisation of every MWh of solar thermal input in an ICARES can be over 60%, up to the point where the solar thermal power input is equal to about 5% of the original mechanical power input [35].

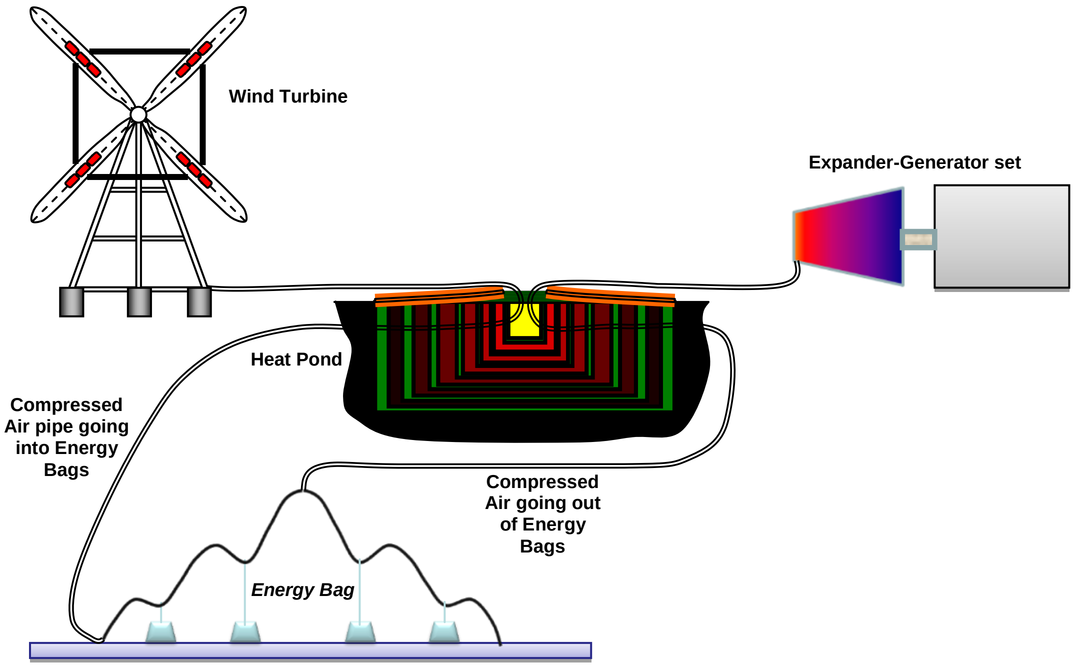

Wind Turbine
Expander-Generator set
Heat Pond
Compressed Air pipe going into Energy Bags
Compressed Air going out of Energy Bags
Energy Bag

Fig. 1.4 Schematic of an ICARES plant, combining an integral compression wind turbine with an Energy Bag (early conception with hanging masses shown), floating thermal energy storage unit, and expander-generator

The ICARES concept was envisaged following Garvey's study of the scaling laws of conventional wind turbines. Up until recently it has been the case that bigger is better for offshore wind turbines, because of the almost fixed overhead costs related to installation, planning, and maintenance, and because larger turbines access higher windspeeds. However, it is anticipated that conventional designs will soon hit a ceiling in size, and scaling them up any further will raise the cost per unit power. This is because for various parts of a wind turbine (including the generator in a direct drive machine), the cost per unit power is proportional to the blade diameter, *even though the swept area (and so power) of a wind turbine is proportional to the square of the blade diameter*. The bulk of the cost of a generator (C) is dependent upon the torque it resists (T), so if we assume that generator cost is proportional to torque, the cost per unit power (P) is given by

$$C/P = aT/P \tag{1.3}$$

where a is a constant. The torque per unit power is given by

$$T/P = 1/\omega, \tag{1.4}$$

where ω is the angular velocity of the rotor. Substituting equation (1.4) into equation (1.3),

$$C/P = a/\omega. \tag{1.5}$$

As conventional design 3-blade wind turbines, no matter what size, typically have a tip speed ratio (ratio of blade tip speed to wind speed) of about 6 to 7, the angular velocity of the blades and so generator rotor is inversely proportional to the blade diameter *D*, and is given by

$$\omega = b/D \tag{1.6}$$

where *b* is a constant. (This inverse relationship between blade diameter and angular velocity explains why a small wind turbine spins much faster than a large turbine in the same wind.) Substituting equation (1.6) into equation (1.5), the cost per unit power is obtained in terms of blade diameter:

$$C/P = (a/b)D. \tag{1.7}$$

So increasing the size of direct drive wind turbines increases the generator cost per unit power. In order to reduce the generator input torque, it is possible to design turbines with higher tip speed ratios, as offshore wind turbines do not have to meet the same noise limits as onshore turbines. However, increasing the tip speed ratio lowers the efficiency, and there is a limit to how fast a 3-blade wind turbine can be made to run because in order to raise the tip speed it is necessary to increase the slenderness of the blades, reducing their load capacity. Analysis of the cost of a gearbox is more complex but it is entirely feasible to suppose that the cost of a gearbox with a large speed-increase factor is also proportional to the input torque.

Garvey then generated a novel concept for a wind turbine that would not be subject to the same scaling laws as conventional wind turbines. An integral compression wind turbine (ICWT), two blades of which are shown in figure 1.5, would compress air directly, having tubes built into the blades with a single running mass inside each tube. As the blades turn, the weight of the running masses will cause them to slide up and down inside the tubes, compressing air. The compressed air could then be immediately expanded to generate electricity. Garvey showed that, at large scales, an ICWT could be more economical than conventional direct-generating wind turbines. It was then realised that if air is being compressed in the process of generating electricity, it would make sense to incorporate a

CAES plant to store surplus energy until required, and use the high water pressure at the bottom of the sea to provide most of the resistance, in which case a flexible bag ("Energy Bag") anchored to the seabed could be used as the containment vessel. Storing energy in this way could further lower the overall lifetime cost of the plant per unit output power. Other offshore renewable energy converters could also be attached, including direct compression wave energy converters and tidal turbines.

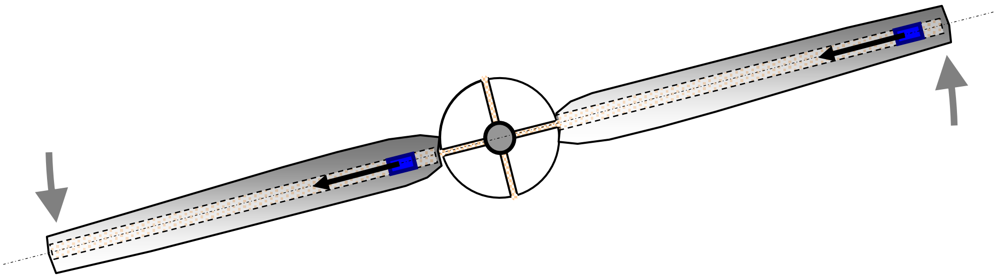


Fig. 1.5 Integral compression wind turbine concept (only 2 blades shown)

The hydrostatic pressure at a depth of 500m is approximately 51bar, and so a flexible Energy Bag anchored at such a depth could be used to contain air at pressures roughly equal to those found in a conventional CAES plant (50-70bar) without excessive stresses in the materials: being flexible, the bag's shape would depend upon the amount of air in the bag, and the surrounding water would provide most of the resistance. Also, because the shape and so volume of the bag can change, the pressure of the contained air would remain approximately constant, regardless of the amount of energy contained, allowing more efficient expansion machinery to be used than with a conventional fixed volume CAES plant. Storing air in bags anchored 500m underwater at a pressure of 51bar (equal to the hydrostatic pressure outside the base of the bag), and conservatively assuming isothermal expansion, the energy density is 5.60kWh/m^3, 3.43 times the energy density in the water in the 600m high upper reservoir at Dinorwig pumped storage plant of 1.64kWh/m^3.

As well as being used for subsea CAES, Energy Bags could also be used for natural gas storage. This would not have the same risks as those surrounding the storage of natural gas in underground caverns; a proposal by Canatxx UK to store 1.2m tonnes of natural gas in salt caverns in Preesall, Lancashire, was rejected by Lancashire County Council in January 2010 due to concerns the caverns could collapse [36]. These risks of collapse also apply to the storage of compressed air in underground caverns.

Being out at sea and underwater, Energy Bags could not be considered as being an "eyesore" as they would be hidden from view. There are many coasts around the world where depths of 500m or more are within reasonable distance of the shore, particularly in the Mediterranean. Unfortunately, the UK is surrounded by relatively shallow waters apart from off the west coast of Scotland and the north-west and south-west coasts of Ireland (see figure 1.6, red line indicates the 500m threshold). Currently shallow water wind turbines are located within a depth of 25m, and deep water wind turbines are located within a depth of 50m; these zones are shown in yellow and pink on figure 1.6. A number of companies are working on floating platforms for offshore wind turbines. Floating wind turbines can be placed further out at sea than the current generation of turbines, in stronger winds with higher power density (proportional to the cube of the wind speed). Companies working on floating wind turbines include Siemens Wind Power, who have built and installed Hywind, the world’s first large-scale floating wind turbine (2.3MW), Principle Power, with their WindFloat, and Floating Power Plant with their combined wind and wave power platform, Poseidon. The Siemens Hywind has been installed in 220m deep water, 12km off the coast of Norway, for a two year test deployment. Siemens have stated that Hywind is designed to be installed in waters between 120-700m deep [37].

As well as being located at sea, Energy Bags could also be located in deep lakes; Loch Morar, Britain’s deepest lake, is 310m deep. In such cases, they could be attached to land-based versions of the offshore ICWTs that would supply the sea-based Energy Bags, or simply be attached to compressors powered by electricity from the grid (for grid storage).

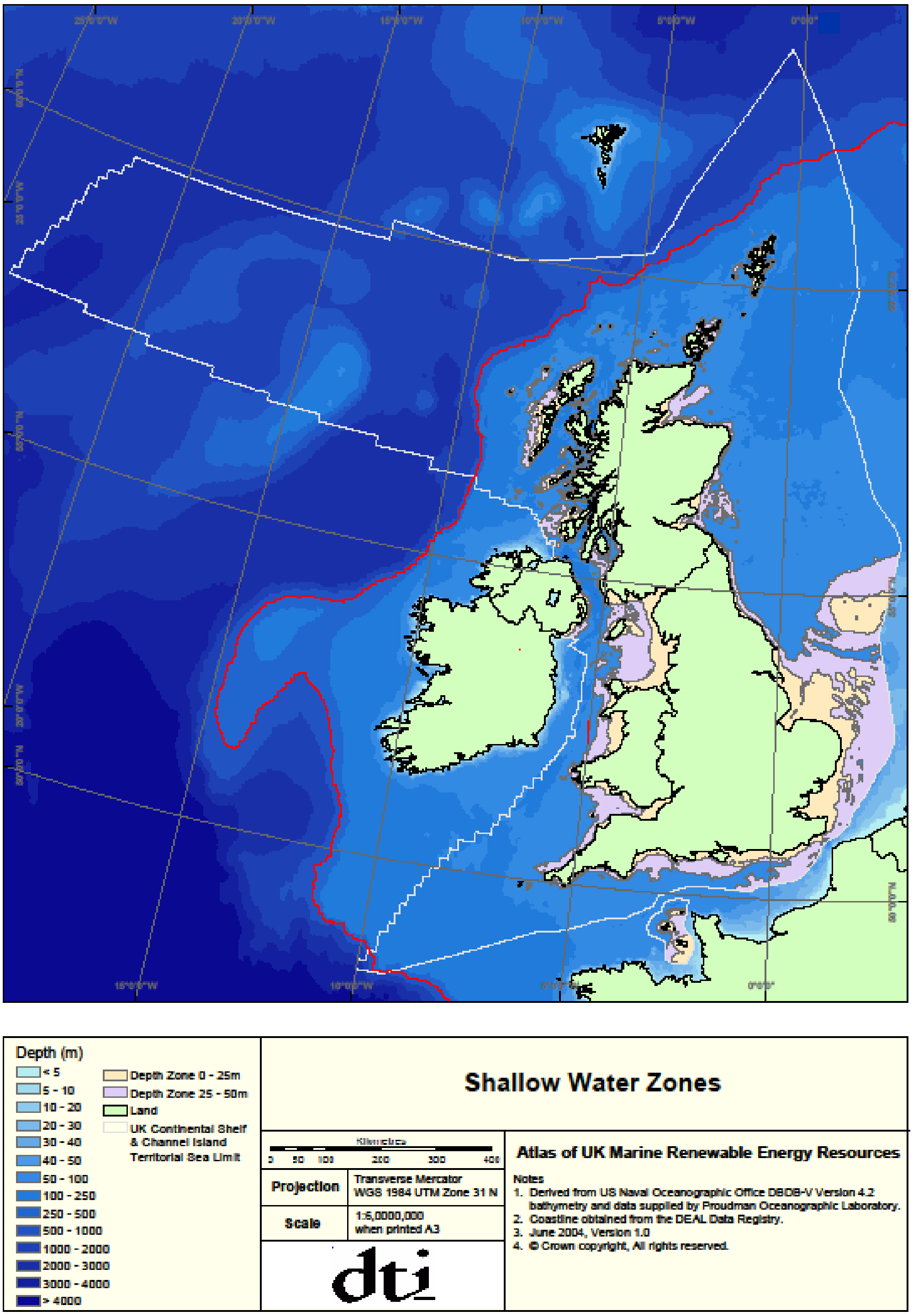

25°0'0"W
20°0'0"W
15°0'0"W
10°0'0"W
5°0'0"W
0°0'0"
Depth (m)
< 5
5 - 10
10 - 20
20 - 30
30 - 40
40 - 50
50 - 100
100 - 250
250 - 500
500 - 1000
1000 - 2000
2000 - 3000
3000 - 4000
> 4000
Depth Zone 0 - 25m
Depth Zone 25 - 50m
Land
UK Continental Shelf & Channel Island Territorial Sea Limit
Shallow Water Zones
Kilometres
0 50 100 200 300 400
Projection
Transverse Mercator WGS 1984 UTM Zone 31 N
Scale
1:6,0000,000 when printed A3
dti
Atlas of UK Marine Renewable Energy Resources
Notes
1. Derived from US Naval Oceanographic Office DBDB-V Version 4.2 bathymetry and data supplied by Proudman Oceanographic Laboratory.
2. Coastline obtained from the DEAL Data Registry.
3. June 2004, Version 1.0
4. © Crown copyright, All rights reserved.

Fig. 1.6 Plot of bathymetry and shallow water zones around the UK, red line added to indicate 500m depth threshold [38]

The market for ICARES is potentially very large. Wind energy sits at the forefront of the UK's portfolio of renewable energy and, along with wave and tidal power, could be used to sustainably meet a large fraction of our future energy needs. However in order to do so, great numbers of wind turbines must be built, and very large stores of energy will be

]

required to cost-effectively deal with short and long term changes in both supply and demand. Though ICARES is based on well-understood engineering concepts, some aspects of the plan must still be investigated and prototype ICARES plants need to be built and tested.

The work in this thesis is driven by the need to advance the Energy Bag concept. In designing an Energy Bag, it is useful to be able to carry out computational modelling of the structure in order to find its deformed shape under hydrostatic loading and to find the stresses in the materials. To the author's knowledge, there is no analysis of underwater pressurised fabric structures to be found in the literature, so we look to the fields of ballooning and tensile structures for useful insight into fabric structure analysis, since an Energy Bag can effectively be treated as a static underwater tethered balloon.

1.3 Fabric Structures

Fabric structures relevant to Energy Bags include balloons, parachutes, underwater inflatables, and tensile structures (e.g. roofs).

Balloons are particularly related to Energy Bags because they are totally flexible: it is desirable to ensure that Energy Bags are also totally flexible, so that their shape (and so volume) changes depending upon the amount of energy in the stored air and the pressure of the contained air always remains close to the hydrostatic pressure of the surrounding water. This is beneficial for two reasons: 1) the forces across the membrane will always be lower than if the structure gave any significant resistance to bending, and 2) the pressure of the air passing from the bag into the expander (when retrieving energy from storage) will be roughly constant no matter how much air is stored inside the bag, allowing more efficient turbomachinery to be used than if the storage volume was fixed.

Underwater inflatables are also of particular interest. They are primarily used for short-term lifting purposes, e.g. raising shipwrecks to the surface of the sea, and have more in common with Energy Bags than tensile structures and balloons because they are used to contain air underwater. However, no technical literature has been found on underwater inflatables. This is a marked difference from the field of ballooning, in which substantial analysis and experimental work has been carried out and documented over the years.

As the name suggests, tensile structures are structures formed from elements in tension (such as membranes, cables, and beams), though they typically require some compression elements as support. We are primarily interested in tensile membrane structures, which are most often used for roofs, e.g. the Millennium Dome/O2 Arena in London, and the Olympiastadion in Munich (shown in figure 1.7).

Fig. 1.7 The membrane roof on the Olympiastadion in Munich [39]

The analysis of flexible fabric structures requires consideration of geometric nonlinearity, because of the large deflections and rotations that a flexible structure can undergo. It is also necessary to account for membrane wrinkling and contact problems (e.g. an Energy Bag coming into contact with itself or the seabed).

1.4 Aims and Scope

The first goal of this PhD is to develop and present procedures for finding the deformed shapes of pressurised flexible fabric structures, driven by the need to be able to accurately model Energy Bags used for large-scale subsea compressed air energy storage. To this end, three modelling procedures have been developed: (a) a system of four coupled ordinary differential equations that, when solved, gives the deformed shape of an inextensible axisymmetric membrane, given in Chapter 3; (b) finite element analysis used to model an extensible axisymmetric membrane using cable elements, given in Chapter 4; and (c) finite element analysis used to model a fully three-dimensional cable-reinforced membrane, using cable and membrane elements, given in Chapter 6. These procedures may also be used to model the deformed shapes of other fabric structures, such as balloons, parachutes, roofs, airbags, sails, and solar sails, as well as cable structures such as nets and moorings.

The second goal is to present procedures for finding the optimum design of Energy Bag. Typically, we seek to minimise the cost of the bags per unit of energy stored, while meeting maximum stress constraints. Some variables that may be included in the optimisation are: size, lobe cutting pattern design, maximum differential pressure at the base (i.e. maximum fill level), and storage depth. A simple analytical optimisation is used to find the optimum-sized axisymmetric natural shape bag and it is shown that for such a bag, the cost of surface and cost of meridional reinforcement should be equal. It is found that for reasonable materials costs, bag materials can cost in the order of £1,000/MWh.

Finally, the third goal of this PhD is to commence underwater testing of prototype Energy Bags. There are several reasons for doing this: to prove that the concept of subsea compressed air energy storage is feasible and has merit, to gain an understanding of the real-world behaviour of Energy Bags and identify problems that must be dealt with in future work, and to benchmark the analysis methods. Two 1.8m diameter prototype bags based on a new modification to a well-tested scientific balloon design have been manufactured for us by a Canadian aerospace company and installed in a water tank in one of the university laboratories. Cycling of these prototypes has commenced.

1.5 Layout of the Thesis

This thesis comprises nine chapters, and the contents of the following chapters are summarised here:

Chapter 2 contains a ***Literature Review*** of recent work in the fields of energy storage, underwater inflatables, ballooning, and nonlinear finite element analysis. Details of correspondence with several manufacturers of underwater inflatables are also given.

Chapter 3 presents a method for ***Modelling Axisymmetric Structures Using Coupled Ordinary Differential Equations***. This is used to find the deformed shapes of various axisymmetric bags, including natural shape bags and bags incorporating circumferential stress and distributed hanging mass. Partially inflated bags are also modelled, though some problems are encountered when trying to model highly deflated bags.

Chapter 4 presents ***Finite Element Analysis of Axisymmetric Structures***, using extensible cable elements. The deformed shapes of various axisymmetric bags are found including bags with circumferential restraint and bags which are unsealed at the base. This method is much easier to use than the coupled ODEs method shown in Chapter 3, and can be used to model highly deflated bags.

Chapter 5 looks at ***Lobed Fabric Structures***, explaining common lobe cutting patterns that have been used in the past, the Ultra High Performance Vessel (UHPV) architecture developed by Thin Red Line Aerospace (and used in our prototype Energy Bags, manufactured by Thin Red Line), and the calculation of circumferential membrane stress. A new lobe cutting pattern is developed which minimises the surface area of a lobe while meeting a maximum stress condition. It has constant circumferential tension at all points along the lobe and so is known as a Constant Tension (CT) lobe. A procedure for finding the CT lobe which minimises this constant tension is given.

Chapter 6 presents ***Three-Dimensional Finite Element Analysis*** of cable-reinforced fabric structures, using cable and membrane elements. Special consideration is given to the wrinkling of both cable and membrane elements. It is found that sometimes the solution procedure doesn’t converge when trying to use symmetry and model only half of a single lobe in an Energy Bag with all identical lobes.

Chapter 7 contains the ***Optimisation*** that has been carried out. A simple analytical optimisation is used to show that the optimum axisymmetric bag for minimising materials cost per unit of energy stored will have cost of surface equal to cost of meridional reinforcement. The computational models are also used in optimisation routines, and a 0.2GWh bag is specified.

Chapter 8 details the ***Testing*** that is being carried out. Two 1.8m diameter prototype Energy Bags, based on superpressure balloons, have been installed in a water tank in one of the university laboratories, and will be cycled many thousands of times.

Chapter 9 contains ***Conclusions and Future Work***, reflecting on the development of the ICARES and Energy Bag concepts over the course of the PhD, and looking ahead to potential future work such as extensions of the computational models and further testing.

Chapter 2

Literature Review

In carrying out work on subsea CAES, it is important to begin with a review of other energy storage technologies, finding out their good and bad points and understanding what purposes they are particularly suited to and why. Obviously particular attention is paid to developments in conventional CAES. We then move on to consider underwater inflatables, carry out a review of the shape analysis of balloons, and look at techniques in nonlinear finite element analysis.

2.1 Energy Storage

There are many different kinds of energy storage systems, some occurring in nature and others man-made. A notable and very well-understood example of energy storage in nature is the storage of chemical energy in the fossilised remains of dead plants and animals – fossil fuels. These are excavated from the earth and burned when necessary to provide heat and to drive vehicles and electric generators. Fossil fuels have very high energy densities by both mass and volume which are yet to be matched by other energy storage methods (apart from by the energy density by mass of hydrogen) [40], however fossil fuels are non-rechargeable. We are really interested in rechargeable systems – stores of energy that can be charged using electricity.

Most energy storage systems require the useful energy to be converted from its initial form (often electrical) into another form which is more suitable for storage (such as chemical) and finally back into a useful form. In each conversion there is a loss associated with the efficiency of the conversion process, and a comparison of several energy storage methods should take the full turnaround efficiency of the storage method into account. However, to investors, cost is often more important than turnaround efficiency, particularly lifetime cost per unit of energy stored and lifetime cost per unit power. Which of these is more important depends upon the purpose of the storage scheme, and it should be noted that for some storage technologies, the power and energy ratings are coupled. Other factors of interest are the time taken to go from charging the energy store to discharging the store and vice-versa, the range of speeds with which the stored energy can be charged and discharged (power capacity), the energy density by both volume and mass, the accessibility of the storage (which can affect cost), and the rate at which energy is lost from the store to the environment. These attributes all affect how useful the storage method is for a particular application.

In the following sections we look at various energy storage methods and compare them based on the attributes described above.

2.1.1 Compressed Air

In a compressed air energy storage (CAES) plant, air is compressed to a high pressure and stored in a pressure vessel or underground cavity until the energy in the compressed air is required. The air is then released from the store, driving a turbine or series of turbines which in turn drives a generator. The heat generated in compression is removed from the compressed air before it is put into store, and this heat energy can be stored in a thermal energy store and later used to reheat the air leaving the store (or it can just be transferred to atmosphere as waste heat). In a power plant with a standard gas turbine, approximately two-thirds of the gas is used to compress the combustion air. It therefore makes sense to use off-peak electrical power to pre-compress the air, and later use the compressed air in the gas turbines when the turbines are producing electricity during peak hours. In this way, three times the power is produced for the same fuel consumption.

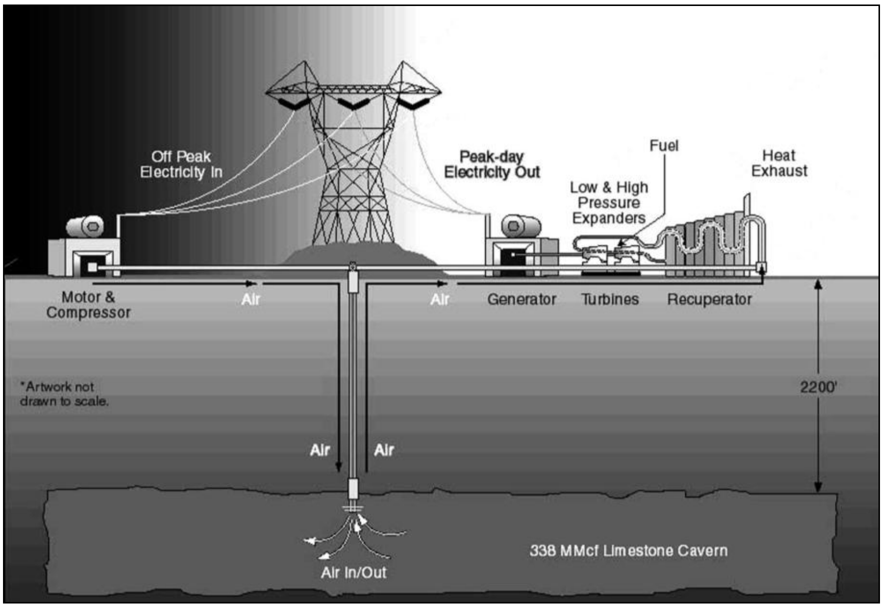


Fig. 2.1 Planned CAES plant at Norton, Ohio [41]

On a small scale, compressed gas can be stored in steel pressure vessels, but on a larger scale use of a steel pressure vessel is uneconomic and it makes much more sense to use features available in the environment. To date, three different types of underground cavities have been seriously considered: excavated salt domes, cavities in rock formations (either natural or excavated), and aquifers. Salt domes are particularly suitable because salt is self-sealing under pressure [42]. The availability of natural locations is limited and excavation can be costly, and the stability of any cavern to withstand cycling temperature and pressure must be fully tested and understood.

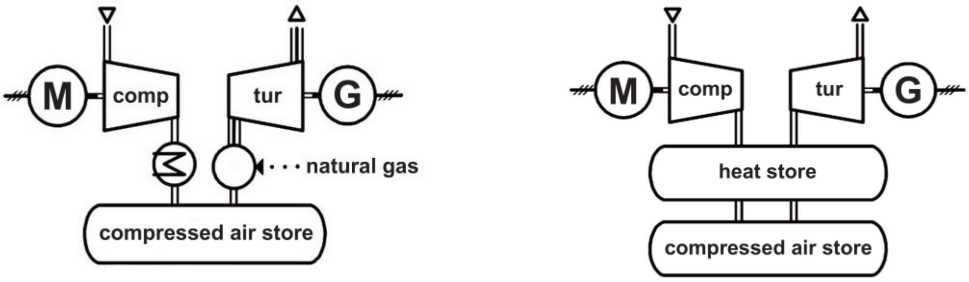
There are currently only two CAES plants in the world, one in Huntorf, Germany, and the other in McIntosh, Alabama, USA. Excavated salt domes are used as the compressed air stores in both plants. In table 2.1 it can be seen that a lower total amount of energy is required per unit output power at McIntosh, the more recently commissioned plant. This is because the McIntosh plant has a recuperator which utilises the waste heat in the turbine exhaust gases to preheat the compressed air entering the turbines. In calculating the efficiency of a plant, care should be taken over the value of input energy, particularly the value placed upon gas. The simplest method is to give gas the same value as electricity; in this case the efficiency of Huntorf (which requires 0.8kWh electricity and 1.6kWh gas for an output of 1kWh electricity) would be 1/(0.8 + 1.6) = 42%, and the efficiency of McIntosh would be 54%. However, 1kWh of gas cannot simply be converted into 1kWh of electricity – if 1kWh of gas is used in a combined cycle gas turbine with a realistic efficiency of 55%, only 0.55kWh electricity will be generated. Using this 55% efficiency, the efficiency of Huntorf becomes 1/(0.8 + 55%×1.6) = 60%, and the efficiency of McIntosh becomes 75%.

	Huntorf, Germany	McIntosh, AL, USA
Date of commission	1978	1991
Store	Two cylindrical salt caverns, each with 150,000m^3 at a depth of 600-800m (height 200m, diameter 30m)	Salt cavern, 538,000m^3 at a depth of 450-750m
Output	290MW over 2 hours	110MW over 26 hours
Energy required for 1 kWh electricity	0.8kWh electricity, 1.6kWh gas (total 2.4kWh)	0.69kWh electricity, 1.17kWh gas (total 1.86kWh)
Pressure tolerance	50-70bar	45-76bar
Remarks	World's first CAES plant	First CAES plant with recuperator

Table 2.1 CAES plants in existence today [43]

Due to its constant volume, the energy stored in an underground cavern is approximately linearly proportional to the pressure of the contained air. Therefore the turbines have to deal with a wide range of pressures and so, being optimised for certain air pressures, aren't always working at maximum efficiency. Increased efficiency is expected from Iowa Stored Energy Park (ISEP), a CAES plant currently in development, because aquifer storage will be used. The displacement of water in the aquifer results in regulation of air pressure because of the changing storage volume. This effect can also be achieved in a cavern by connecting the bottom of the cavern to a water reservoir (known as a shuttle pond) located above the cavern. Another CAES plant currently in development will be located at Norton, Ohio, USA (see figure 2.1), with an initial power capacity of 300MW and design plans to increase to 2,700MW over five years. The compressed air will be stored in a disused limestone mine at pressures between 62bar and 114bar [44].

In an adiabatic compressed air energy storage (A-CAES) plant, the heat generated in compression is stored in a thermal energy store, instead of being vented to the atmosphere as waste heat as in a conventional (diabatic) CAES plant. This stored thermal energy is later used to heat up the air passing from the store into the expander, so that no fuel needs to be burned. An A-CAES plant operating without fuel burn would have no greenhouse gas emissions and so would not incur carbon emissions costs. Improved efficiencies are possible using A-CAES, though an A-CAES plant has a higher initial cost than a conventional CAES plant because of the need for a thermal energy store and heat exchanger. As fuel and carbon emissions costs increase, A-CAES plants will become more attractive. It should be noted that the performance of a heat store improves with its size – the rate at which heat leaks from the store is proportional to its surface area, and the capacity of the store is proportional to its volume, and the ratio of heat leakage to energy capacity is proportional to the ratio of surface area to volume, which is inversely proportional to any geometric dimension (e.g. diameter).



(a) (b)

Fig. 2.2 Schematic circuit diagrams of: (a) A diabatic CAES process, and; (b) An adiabatic CAES process [45]

The energy available in a compressed air store depends upon how the compressed air is expanded. In this work we treat air as an ideal gas (the compressibility factor for air at 300°K and 60bar is 0.9901 which is very close to 1, the compressibility factor of an ideal gas [46]). We concern ourselves with two main thermodynamic processes: isothermal and adiabatic.

An isothermal process is a change of a system during which the temperature remains constant. Isothermal processes occur when the system is thermally connected to a constant-temperature external reservoir, and when the change in the system is happening so slowly that the system continually maintains the same temperature as the external reservoir through heat exchange. The ideal gas law tells us that the absolute pressure of an ideal gas is given by

$$P = \frac{nRT}{V}, \tag{2.1}$$

where V is the volume of the gas, n is the amount of substance, R is the universal gas constant, and T is the absolute temperature. In an isothermal process, T remains constant and so the numerator of equation (2.1) also remains constant. As a result, for an isothermal process,

$$PV = \text{constant.} \tag{2.2}$$

Therefore in an isothermal process, the absolute pressure of an ideal gas is inversely proportional to its volume. This relationship is known as Boyle's Law.

The work done in the isothermal expansion from stored volume V_A (with absolute pressure P_A) to volume V_B (with absolute pressure P_B) is

$$W_{A\to B} = \int_{V_A}^{V_B} PdV = \int_{V_A}^{V_B} \frac{nRT}{V} dV = nRT \int_{V_A}^{V_B} \frac{1}{V} dV = nRT \ln \frac{V_B}{V_A}. \tag{2.3}$$

As shown in equation (2.2), the product PV remains unchanged for an ideal gas undergoing an isothermal process, so $V_B / V_A = P_A / P_B$ and

$$W_{A\to B} = P_A V_A \ln \frac{P_A}{P_B}. \tag{2.4}$$

To calculate the energy available in a store of compressed air that is to be expanded isothermally, we let P_A and V_A be the pressure and volume of the air in the store and P_B be the pressure of the expanded air (atmospheric pressure if all the available energy is taken out of the compressed air by the expansion machinery).

An adiabatic process is a change of a system during which no energy enters or leaves the system through heat exchange. A purely adiabatic process can only occur if the system is thermally insulated from the surroundings. The word adiabatic comes from Greek words which literally mean impassable, in this case to heat transfer. The pressure and volume of an ideal gas undergoing a reversible adiabatic process are related by

$$PV^{\gamma} = \text{constant,} \tag{2.5}$$

where γ is the adiabatic index of the gas, given by

$$\gamma = \frac{C_P}{C_V} = \frac{\alpha + 1}{\alpha}. \tag{2.6}$$

C_P is the fluid’s specific heat capacity at constant pressure, C_V is the specific heat capacity at constant volume, and α = 5/2 for a diatomic gas and α = 3/2 for a monatomic gas. Air is essentially a diatomic gas, so we use γ = 7/5.

In calculating the net work input in the adiabatic compression of air from atmospheric pressure P_A to storage pressure P_B, we must include the work associated with moving the air from the atmosphere into the compression volume ($-P_{\text{in}}V_{\text{in}}$) and discharging the high pressure air ($P_{\text{out}}V_{\text{out}}$), causing the integration for work to become $\int VdP$, rather than $\int PdV$. The energy available in the store, if the compressed air is preheated to the temperature it has after compression and then expanded adiabatically, is equal to the net work input in the compression.

From equation (2.5),

$$PV^{\gamma} = P_A V_A{}^{\gamma} = \text{constant.} \tag{2.7}$$

We can rearrange this to obtain the fluid volume as a function of pressure,

$$V = V_A \left(\frac{P_A}{P} \right)^{\frac{1}{\gamma}}. \tag{2.8}$$

So in an adiabatic process, the net work done by the system is given by

$$W_{A\to B} = \int_{P_A}^{P_B} VdP = \int_{P_A}^{P_B} V_A \left(\frac{P_A}{P} \right)^{\frac{1}{\gamma}} dP = P_A{}^{\frac{1}{\gamma}} V_A \left(P_B{}^{\frac{\gamma-1}{\gamma}} - P_A{}^{\frac{\gamma-1}{\gamma}} \right) \left(\frac{\gamma}{\gamma-1} \right)$$

$$= V_A \left(P_A{}^{\frac{1}{\gamma}} P_B{}^{\frac{\gamma-1}{\gamma}} - P_A \right) \left(\frac{\gamma}{\gamma-1} \right) = P_A V_A \left(\left(\frac{P_B}{P_A} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) \left(\frac{\gamma}{\gamma-1} \right). \tag{2.9}$$

Therefore treating air as an ideal gas, compressing r cubic metres of air adiabatically from atmospheric pressure (101.325kPa) up to r times atmospheric pressure demands a net work input of

$$W_{adiab} = 101325r \left(r^{\frac{\gamma-1}{\gamma}} - 1 \right) \left(\frac{\gamma}{\gamma-1} \right). \tag{2.10}$$

The final absolute temperature of a gas undergoing an adiabatic process is given by

$$T_B = T_A \left(\frac{P_B}{P_A} \right)^{\frac{\gamma - 1}{\gamma}} . \tag{2.11}$$

Once the heat has been taken out of the air, the volume of the *r* cubic metres compressed from atmospheric pressure up to *r* times atmospheric pressure will become 1m^3. Therefore the work input found using equation (2.10) will be the energy density of the cool air in the store, assuming the heat energy that was taken out of the air during and after compression is all used to reheat the air before expansion.

In practice, purely isothermal or adiabatic processes cannot occur because there is no such thing as a perfect conductor or insulator, but processes can be very close to purely isothermal or adiabatic, and we use the equations describing purely isothermal and adiabatic processes as means to calculate lower and upper bounds on the amount of energy available in a store of compressed air: isothermal expansion at atmospheric temperature is the lower bound, and adiabatic expansion from the temperature of the air after adiabatic compression (so leaving the expander at atmospheric temperature and pressure) is the upper bound. How close a process is to being isothermal or adiabatic depends upon the thermal conductivity of the system boundary and the speed at which the process occurs: a very quick process, in which little heat is transferred between the system and its surroundings, may be considered adiabatic, and a very slow process, in which the system's temperature remains constant, may be considered isothermal.

Either of equations (2.4) or (2.10) can be used to gain an assessment of how much air is stored in an Energy Bag but they will provide different results. We may rely on the fact that the stored air will be cool – it will lose its heat rapidly if it is not already cool when put into storage. Thermodynamically, the optimum combination of compression and expansion processes would be isothermal compression, with all of the heat of compression being put into a thermal energy store, followed by adiabatic expansion, pre-heating the air with the stored thermal energy before expanding it adiabatically to atmospheric temperature and pressure. As mentioned above, the worst expansion process would be isothermal expansion, assuming that the high pressure air is expanded in multiple stages with inter-stage reheats where the heat in this case can simply be drawn from ambient seawater. In this case the compression process used is irrelevant because the heat taken out of the air, either during or after the compression process or through losing heat to the surrounding seawater after being put into store, is unused.

In both cases it is strictly correct to subtract the gravitational potential energy required to raise the air back to the sea surface, though this is very small in comparison with the stored energy. Table 2.2 summarises the energy densities for a number of depths, having subtracted gravitational potential. We have assumed that the air will be stored at an absolute pressure equal to the hydrostatic pressure at the given depth, given by

$$P = \rho_w gh + P_{atm}, \tag{2.12}$$

where ρ_w is the density of seawater (approximately 1,025kg/m^3), g is standard gravity, h is the depth below the surface of the sea, and P_{atm} is the atmospheric pressure (101.325kPa). The gravitational potential energy is given by

$$W = \rho_a Vgh, \tag{2.13}$$

where ρ_a is the density of the compressed air. The density of an ideal gas compressed to absolute pressure P is

$$\rho = \frac{MP}{RT}, \tag{2.14}$$

where the molar mass of air, M_{air} = 0.02897kg/mol, the universal gas constant, R = 8.314472J/(K.mol), and T is the absolute temperature of the air.

Seabed depth (m)	T_B (°C) (assuming T_A = 5°C)	Absolute compressed air pressure (bar)	Isothermal energy density (MJ/m^3)	Energy density with pre-expansion reheat (MJ/m^3)
50	190.1	6.04	1.075	1.407
100	277.6	11.07	2.633	3.797
200	389.3	21.12	6.364	10.215
300	467.2	31.18	10.569	18.135
400	528.8	41.23	15.079	27.178
500	580.4	51.29	19.813	37.135
600	625.2	61.34	24.720	47.873
700	665.0	71.40	29.767	59.297
800	701.0	81.46	34.933	71.337

Table 2.2 Energy storage densities for Energy Bags at various depths

Only one piece of work has been found on subsea CAES. Written by Richard J. Seymour, Head of the Ocean Engineering Research Group at the Scripps Institution of

Oceanography, University of California, San Diego, in 1997, Undersea Pumped Storage for Load Leveling [47] describes offshore CAES (or OCAES, as it is referred to in the paper), explaining the idea of storing compressed air in rigid tanks underwater. The tanks are unsealed at the base and so allow seawater in and out, and when air is pumped into the tanks it displaces seawater. Only one mention is made of using a flexible bladder, as a potential lining for the compressed air tank (which is shown in figure 2.3). It is noted that if no bladder is employed, air can also be removed from the pipe without creating a vacuum in the pipe (which would subject it to large crushing loads) because water will fill the pipe in place of the air. The paper discusses a 230MW OCAES system at Carlsbad, San Diego, which would have a 16km air pipe from the existing power plant on the coast at Carlsbad to 650m deep water. The air pipeline would run along a submarine canyon, providing the shortest distance to deep water and ensuring that the pipeline is kept out of "active" water.

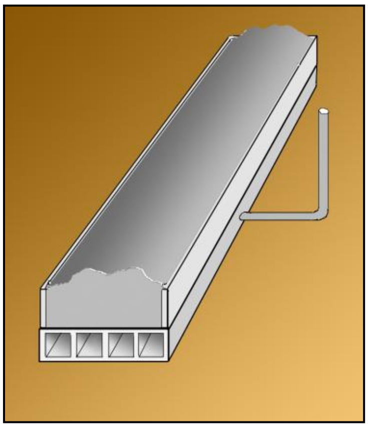


Fig. 2.3 Ballasted Compressed Air Storage Concept generated by Seymour [48]

The results of a research project by Anton Rowe (one of Seymour's undergraduate project students) are summarised in [47]: a 230MW OCAES system, generating for 10 hours a day and storing air at 60atm, would require a volume that could be contained in a pipe 3.6m in diameter and 12.8km in length. To accommodate the necessary flow rate, the attached pipeline would require a diameter of about 1.2m. It is estimated that a 230MW system operating for 10 hours per day would reduce the supply variation range in the San Diego area by about 40%. Apparently Seymour was introduced to the OCAES idea by James Wight of San Diego Gas & Electric, and Seymour is currently working with an industrial team on a preliminary study funded by the federal government.

2.1.2 Pumped Hydro

In use since 1929 and the only commercially available means of storing energy until 1970 [49], pumped storage sets a high standard for large-scale energy storage. It achieves turnaround efficiencies of around 80% [50],[51], can be called upon rapidly due to the short start-up time for hydro turbines (1/2-3 minutes from shutdown or full reversal or 10-30 seconds if kept on standby) [40] and loses little energy over time (only through evaporation). Berry estimates the cost of pumped storage at approximately £3,000/MWh [52], though costs of Dinorwig (Britain's largest pumped hydro plant, built in 1984) work out at more like £50,000/MWh.

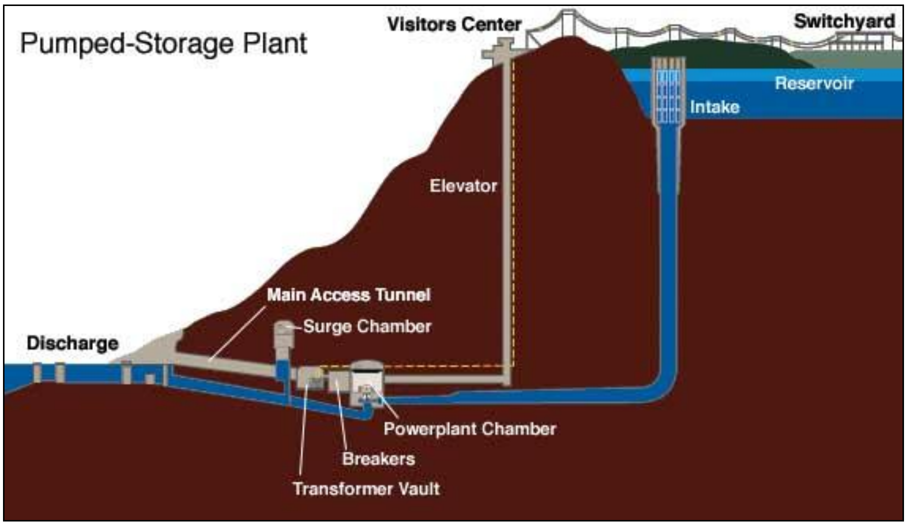


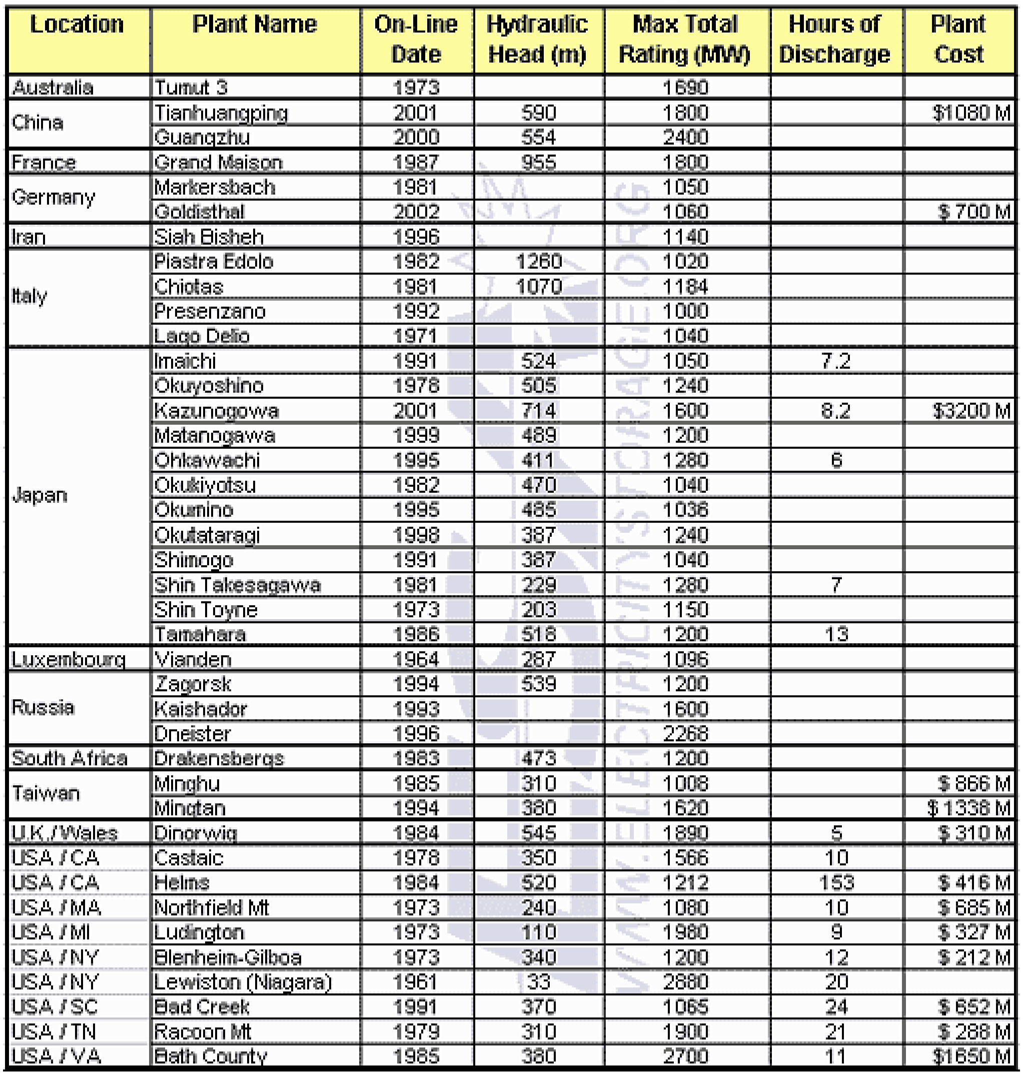
Fig. 2.4 Overview of a pumped hydro facility at Raccoon Mountain, TN, USA [53]

Energy is stored in a pumped hydro plant as gravitational potential energy; during periods of high supply and low demand, excess energy is used to pump water from a lower reservoir up into a higher reservoir, where it is stored until periods of peak demand. The water is then allowed to flow back down to the lower reservoir, driving a turbine and generator. A reversible pump-turbine is often used, so the same machine is used for both pumping and generating. The potential energy in a raised body of water of density ρ and volume V is given by

$$W = \rho V g \Delta z \tag{2.15}$$

where Δz is the difference in height between the upper and lower reservoirs. Therefore to maximise the stored energy it is necessary to maximise the volume of water that can be moved between the two reservoirs and the height of the upper reservoir above the lower reservoir.

Generally pumped storage plants have energy storage capacities of several GWh, but Helms facility at Shaver Lake, California, part of a chain of large lakes at high altitudes in the Sierra National Forest between San Francisco and Las Vegas, has a very large storage capacity of 184GWh (with a power capacity of 1,206MW) [54]. Table 2.3 lists all the pumped hydro facilities in the world with power capacities of 1,000MW or greater. Note that it is over 20 years since a large pumped hydro facility was built in the USA.



Location	Plant Name	On-Line Date	Hydraulic Head (m)	Max Total Rating (MW)	Hours of Discharge	Plant Cost
Australia	Tumut 3	1973		1690		
China	Tianhuangping	2001	590	1800		$1080 M
	Guangzhu	2000	554	2400		
France	Grand Maison	1987	955	1800		
Germany	Markersbach	1981		1050		
	Goldisthal	2002		1060		$ 700 M
Iran	Siah Bisheh	1996		1140		
Italy	Piastra Edolo	1982	1260	1020		
	Chiotas	1981	1070	1184		
	Presenzano	1992		1000		
	Lago Delio	1971		1040		
Japan	Imaichi	1991	524	1050	7.2	
	Okuyoshino	1978	505	1240		
	Kazunogowa	2001	714	1600	8.2	$3200 M
	Matanogawa	1999	489	1200		
	Ohkawachi	1995	411	1280	6	
	Okukiyotsu	1982	470	1040		
	Okumino	1995	485	1036		
	Okutataragi	1998	387	1240		
	Shimogo	1991	387	1040		
	Shin Takesagawa	1981	229	1280	7	
	Shin Toyne	1973	203	1150		
	Tamahara	1986	518	1200	13	
Luxembourg	Vianden	1964	287	1096		
Russia	Zagorsk	1994	539	1200		
	Kaishador	1993		1600		
	Dneister	1996		2268		
South Africa	Drakensbergs	1983	473	1200		
Taiwan	Minghu	1985	310	1008		$ 866 M
	Mingtan	1994	380	1620		$ 1338 M
U.K./Wales	Dinorwig	1984	545	1890	5	$ 310 M
USA / CA	Castaic	1978	350	1566	10	
USA / CA	Helms	1984	520	1212	153	$ 416 M
USA / MA	Northfield Mt	1973	240	1080	10	$ 685 M
USA / MI	Ludington	1973	110	1980	9	$ 327 M
USA / NY	Blenheim-Gilboa	1973	340	1200	12	$ 212 M
USA / NY	Lewiston (Niagara)	1961	33	2880	20	
USA / SC	Bad Creek	1991	370	1065	24	$ 652 M
USA / TN	Racoon Mt	1979	310	1900	21	$ 288 M
USA / VA	Bath County	1985	380	2700	11	$1650 M

Table 2.3 1,000 MW and larger pumped hydro installations worldwide [55]

The number of sites for pumped hydro schemes is limited by the availability of elevated natural or man-made upper reservoirs and nearby lower reservoirs into which the water can drain. The environmental impact is usually confined to mountainous regions and machinery and pipes are often located underground, so the visual impact of a pumped hydro scheme is minimal. However, with the best locations for pumped hydro schemes often being remote, grid connection and construction are not straightforward [56]. If no natural elevated reservoirs are present, underground lower reservoirs may be used below surface level upper reservoirs (such as lakes or the sea). The underground reservoirs may be natural or excavated cavities; if a cavity is excavated, the cost of the scheme scales approximately linearly with the storage capacity. Recently, seawater has been used in a pumped storage plant near Okinawa, Japan, with the upper reservoir situated on high seaside cliffs and the sea below acting as the lower reservoir (figure 2.5). Special measures were taken to prevent corrosion and the adhesion of marine organisms (e.g. barnacles) to the pumps, turbines and pipes [57].

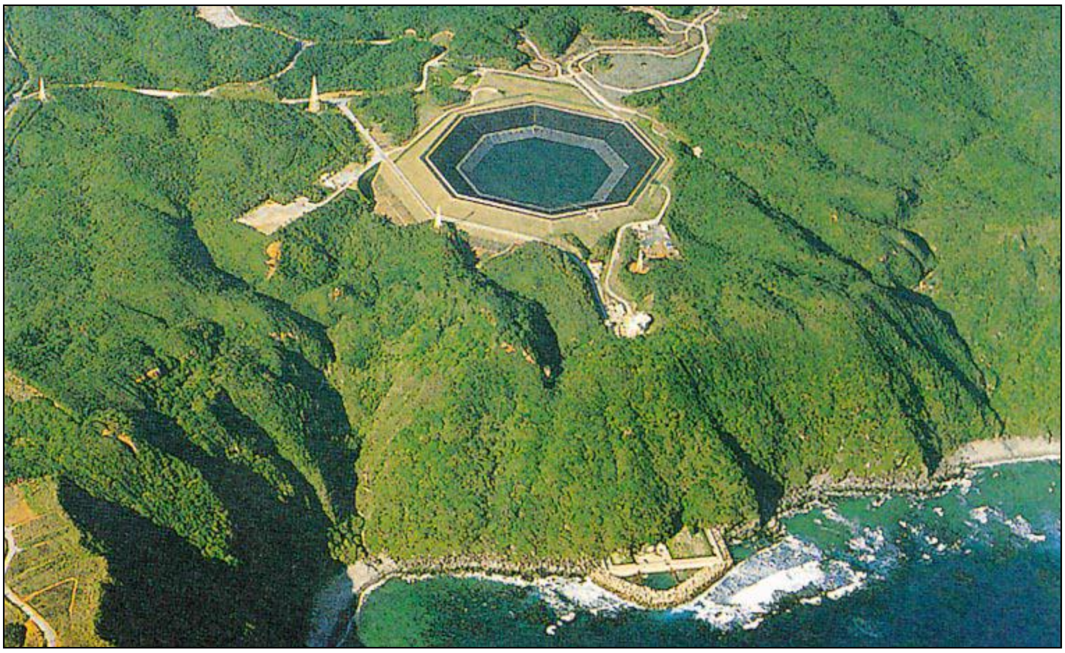


Fig. 2.5 Pumped hydroelectric energy storage plant using seawater [57]

2.1.3 Batteries

Batteries comprise of one or more electrochemical cells, each cell consisting of an electrolyte (which can be solid, liquid or paste) and two electrodes: a positive anode and a negative cathode. Electrochemical reactions at the electrodes occur during discharge, creating a flow of electrons through the external circuit. Batteries fall into two broad categories: primary batteries, which irreversibly transform chemical energy to electrical energy and so cannot be recharged (without manually replacing the components of the battery consumed by the chemical reaction); and secondary batteries, which are reversible and so can be recharged by applying an external voltage across the electrodes. Secondary batteries are of much more interest when looking at long-term storage technologies required to undergo thousands of charge/discharge cycles. A number of different materials have been used for the electrolyte and electrodes. Batteries suitable for utility scale battery storage include lead acid, nickel cadmium, sodium sulphur, sodium nickel chloride, and lithium ion [58].

2.1.4 Flow Batteries

Flow batteries store and release energy using a reversible reaction between two electrolyte solutions separated by an ion permeable membrane. The electrolytes are stored in separate tanks and pumped into a cell for energy transfer; ions are passed between the electrolytes, changing their ionic forms. During recharge this process is reversed and the ionic form of the electrolyte reverts to the charged form. Therefore the energy capacity and power capacity are essentially decoupled, giving designers the flexibility to increase either one with no effect on the other by simply increasing the volume of the solution or the size of the cell stack. With no obvious limits on the scale, flow batteries have the potential to join pumped hydro and compressed air in the large-scale energy storage market. Many different electrolyte couples have been proposed and current developments are focused on vanadium redox, sodium polysulphide/sodium bromide (Regenesys), and zinc/bromine [59].

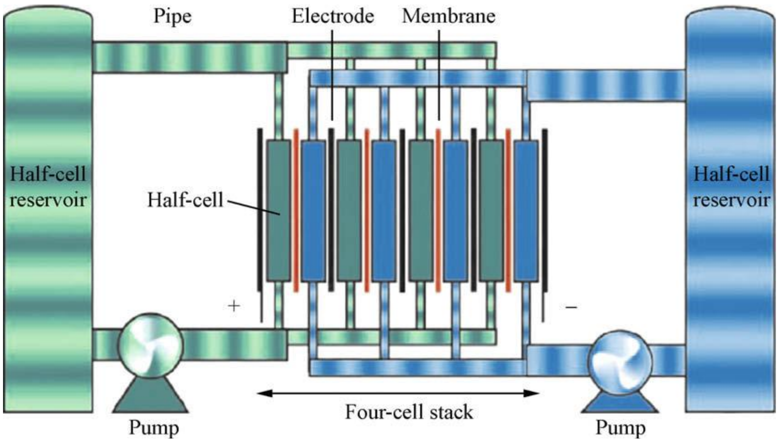


Fig. 2.6 Overview of a flow battery [60]

Flow batteries typically have efficiencies of 75-80%, a charge/discharge ratio of 1/1, and are flexible in operation, especially with respect to discharge times which can range between several minutes and many hours [56]. Vanadium redox batteries, which have the ability to go from charge to discharge in 1ms, have been integrated into some wind farms for load-levelling purposes. Cells last over 10,000 cycles, and the manufacturer Sumitomo Electric Industries recommends that the cell stack be replaced every 10 years. The electrolytes have an indefinite life [59].

Regenesys, a 120MWh sodium polysulphide/sodium bromide store with a maximum discharge rate of 15MW (so a minimum time for full charge/discharge of 8 hours), was installed at Little Barford, Cambridgeshire in the early 2000s. The cost was estimated at £1,000/kW or £125/kWh [61]. However, after severe technical difficulties the funding was withdrawn in 2003; the project was subsequently discontinued.

2.1.5 Flywheels

Energy can be stored in a flywheel in the form of rotational motion, and has been done so in many applications (such as potters' wheels) for hundreds of years. The amount of kinetic energy stored in a flywheel depends upon both the moment of inertia of the flywheel (*I*) and its angular velocity (ω):

$$W = \frac{1}{2} I\omega^2 . \tag{2.16}$$

The moment of inertia is given by

$$I = \int \rho(x) r^2 dx , \tag{2.17}$$

where $\rho(x)$ is the mass distribution and *r* is the distance from the axis of rotation. Therefore it would appear that stored energy is maximised by maximising the angular velocity and the product of mass and distance from the centreline. However, the finite strength of materials used in a flywheel limits the maximum amount of energy that can be stored. Sorensen [40] shows that a constant stress disc with exponential cross-section is a very good shape to use to maximise energy density (by mass – flat discs are more volume efficient). It is also necessary to maximise the ratio of tensile strength to material density, though the cost of doing so must be taken into account.

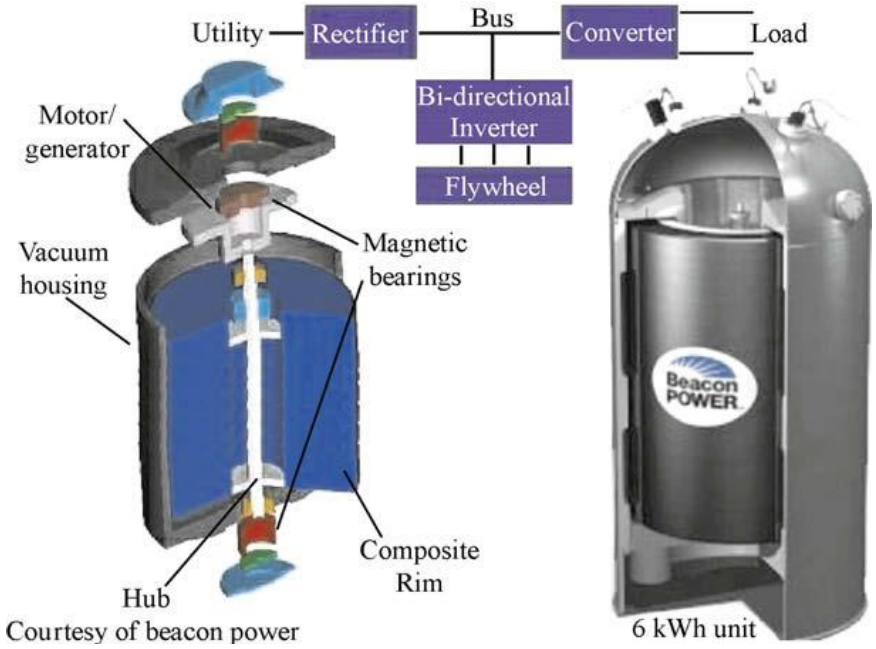


Fig. 2.7 Flywheel energy storage system [58]

Frictional losses can be reduced by operating the flywheel in near vacuum and using magnetic bearings. In principle, high strength, low density carbon-composite filaments would allow energy densities of about 3,000kJ/kg, though materials and manufacturing costs would be higher than those of conventional steel flywheels, which have energy densities of 30-120kJ/kg [40]. With no thermodynamic process involved and so no limiting Carnot efficiency, the cycle efficiency of flywheels can be as high as 95%. However, the efficiency of a 200 ton flywheel in storing energy over a full day is estimated at 45%, reducing further with time [62]. The issue of safety must be considered when rotating a very large, heavy disc at high speeds, so it is likely that flywheels will need to be housed inside special casings or underground.

Low speed flywheels cost between £150k/MWh and £200k/MWh and high speed flywheels with higher energy densities cost as much as £17m/MWh, being in the early stages of development [59].

2.1.6 Superconductors (SMES)

Initially proposed in 1969, superconducting magnetic energy storage (SMES) is the only technology known to store electrical energy directly in electric current. Normally, when current is passed through a wire, heat losses occur due to the resistance of the wire. However, a wire made from a superconducting material (such as lead, mercury, or vanadium) and kept in a superconducting state has virtually zero resistance. So by charging a magnet with a superconducting coil (e.g. a solenoid) energy is stored in the magnetic field and the heat losses in the coil are practically zero. The coil must be immersed in liquid helium contained in a vacuum-insulated cryostat in order to be kept at a temperature close to absolute zero and so in its superconducting state [56]. SMES has the advantage of high efficiency (typically >97%) and fast discharge rates (full discharge is possible within a few milliseconds if desired), but only for a short period of time. SMES also has a high cycle life. As a result, SMES is ideal for providing power quality to large industrial customers. Due to the high energy consumption of the refrigeration system, SMES is not suitable for daily cycling applications such as generation and transmission deferral [59].

The typical rating of a SMES system is 1-10MW with a storage time of seconds, but work is being carried out on larger systems with power capacities of 10-100MW and a storage time of minutes [58]. A 28MWh prototype SMES plant has been built in the USA by the Department of Defense for the purposes of rapidly releasing energy for anti-missile defense concepts [40]. SMES costs approximately £200/kW [59].

2.1.7 Hydrogen

Much research into hydrogen storage is focused on mobile applications. If hydrogen is to be used as fuel for vehicles then it is necessary to pressurise it or convert it into liquid or slush hydrogen, to increase the energy density by volume and so provide a reasonable driving range. However, the mass of the tanks required to store compressed hydrogen reduces the efficiency of the vehicle and, because the boiling point of hydrogen is 20.28K (-252.87°C), liquid hydrogen requires cryogenic storage which introduces additional costs (for both the cooling machinery and the insulation on the storage tanks) and lowers the efficiency (because energy is required to cool the hydrogen). Slush hydrogen, a mixture of solid and liquid hydrogen, is formed by further lowering the temperature of liquid hydrogen to near its melting point (14.01K), increasing the density by 15-20% relative to the density of liquid hydrogen.

Hydrogen can also be stored in a hydrogen-containing compound such as a hydride. When the hydrogen is required, the storage material is caused to decompose, yielding the hydrogen gas. However, the high temperatures and pressures that are required for hydride formation and hydrogen release present barriers to the commercial realisation of hydrides.

Hydrogen can be stored underground in caverns, excavated salt domes, disused mines, aquifers, and depleted oil and gas fields. In Teesside, ICI has stored approximately 1 million m^3 of hydrogen gas in three salt caverns at about 400m depth for a number of years. According to Taylor *et al* [63], underground storage is the least expensive method of storing large quantities of gaseous hydrogen. Indeed, underground hydrogen storage is about two orders of magnitude cheaper than tank storage when applied to volumes of several million m^3 of hydrogen. The highest capital investment is in depleted gas wells.

In depleted gas wells the gas stored in a field is divided into active working gas and cushion gas, an inactive base gas that is not recoverable. The ratio of working to cushion gas varies widely with a ratio greater than 2:1 generally being preferred. As hydrogen is a relatively expensive commodity, the cost of the cushion gas is a very significant part of the capital charges for such large storage reservoirs. However, as the cavern is repeatedly cycled, the initial cushion gas cost is amortised [64].

2.1.8 Thermal Energy Storage

Heat can be used as an effective means of storing energy, and is particularly suited to use in solar thermal power plants. Typically heat is transferred to a thermal storage medium during the day, and withdrawn for power generation at night or during overcast periods. Various thermal storage media have been used or proposed, including pressurised steam, concrete, phase change materials, and liquids such as water, oil, and molten salt.

Molten salt (60% sodium nitrate and 40% potassium nitrate) is used for heat storage at Andasol power station, Europe’s first commercial parabolic trough solar thermal power plant located in Andalusia, Spain. The salt melts at ~220°C and is kept liquid at ~290°C in an insulated “cold” storage tank [65]. When heat is to be stored the molten salt is heated using the concentrated solar power to ~570°C, and then moved into an insulated hot storage tank. When electricity is required from the store (e.g. during the evening or when the sky is overcast), the heat is used to create superheated steam for use in a conventional steam turbogenerator. The heat reservoir consists of two cylindrical molten salt storage tanks (as in figure 2.8), each one 14m high and 36m diameter, giving enough storage capacity to continue providing full power for approximately 7.5 hours. Solar Tres, another plant that is being built in Spain, will include sufficient molten salt storage to allow the plant to continue to output power throughout the night during the summer.

For plants using molten salt storage, energy storage densities of approximately 100kWh/m^3 are common, and studies have shown that a two-tank storage system could have an annual efficiency of about 99% [65]. Current research focuses on molten salts with a melting point below 100°C to avoid special treatment in order to prevent the salt from solidifying [66].

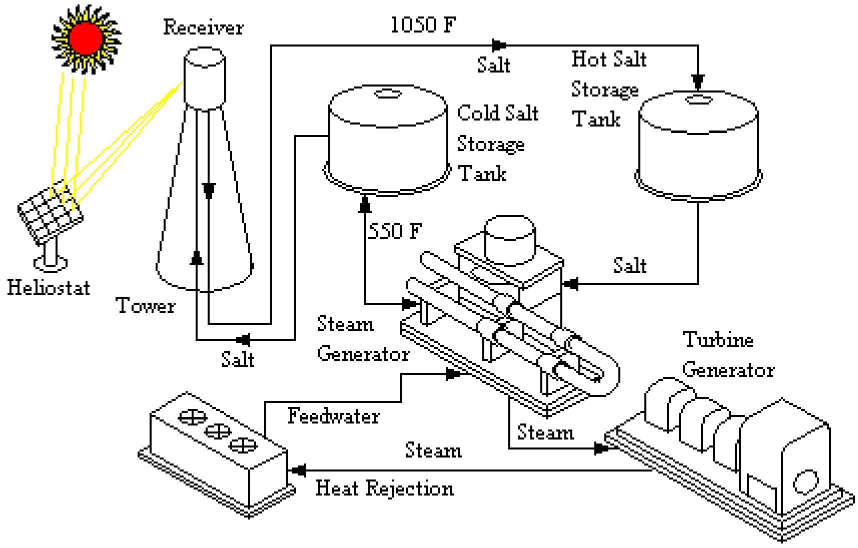


Fig. 2.8 A two-tank molten salt thermal energy storage system integrated into a solar thermal power plant [65]

Instead of using two separate tanks to store the hot and cold thermal storage medium, it is possible to use a single tank in what is known as a thermocline. Such a tank has the hot fluid at the top of the tank and the cold fluid at the bottom, with a thermocline in between. Depending on the cost of the storage fluid, thermoclines can have substantially lower costs than two-tank storage systems, but it is necessary to maintain the thermocline within the tank.

A pumped heat electricity storage system (shown in figure 2.9) is being developed by Isentropic Ltd., comprising two large heat storage containers (one hot and one cold) and a heat pump. Electrical power is used to compress/expand air in the heat pump, which is then used to raise the temperature of the gravel in one of the containers to about 500°C and lower the temperature in the other container to about -160°C. To obtain electricity from the store the cycle is reversed: air is heated in the hot container and cooled in the cold container, and the temperature difference is used to drive the heat pump in reverse as a heat engine, generating electricity. Isentropic claim that their system has a round trip efficiency of 72-80%, and because gravel is such a cheap and readily available material, the cost can be kept down to around £34/kWh (about £6/kWh at scale) [67].

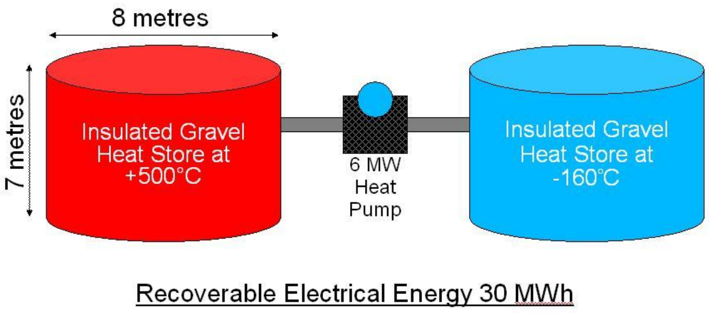

8 metres
7 metres
Insulated Gravel Heat Store at +500°C
6 MW Heat Pump
Insulated Gravel Heat Store at -160°C
Recoverable Electrical Energy 30 MWh

Fig. 2.9 Pumped heat electricity storage plant designed by Isentropic Ltd. [67]

It is also possible to store electrical energy using thermocouple batteries and Alkali Metal Thermal Electric Converters (AMTEC batteries). Thermocouple batteries are based on the Seebeck effect (upon which thermocouples are based): in a closed circuit comprising two dissimilar metals, an electric potential is created between the two junction points when one junction is heated and the other is kept cool. Thermocouple batteries are not very efficient and are only suitable for low power applications.

Thermal energy storage can also be used to reduce the electricity consumption of building heating and air conditioning systems during times of peak demand by using cheap off-peak electricity to power a hot or cold storage system.

2.2 Underwater Inflatables and Water Load Test Weights

Underwater inflatables and water load test weights are fabric structures used for the application of load: underwater inflatables are used to raise objects off the seabed, reduce the draft of ships, and support pipelines, and water load test weights are used for the proof load testing of cranes and frames. Both underwater inflatables and water load test weights share many similarities with Energy Bags, in that they are reinforced fabric structures designed to carry the same loads while separating water and air. Underwater inflatables are also designed to withstand the marine environment. No technical literature could be found on lifting bags or water load test weights, but much information could be found on company websites. We also engaged in discussions with several manufacturers of underwater inflatables and water load test weights, and these discussions are documented in section 2.2.3.

2.2.1 Lifting Bags

Inflatable lifting bags are used for the lifting of wrecks, sunken ships and submarines, concrete pipes, and other items from the seabed to the surface, or for the short term support of an underwater item. They have become popular for these uses because they are cheap, reusable, lightweight, and require relatively little storage space.

Seaflex Ltd. are an Isle of Wight based manufacturer of marine air lift bags and water load test weights with over 15 years' experience. In one of their more interesting projects, they supplied (for hire) five of their open bottom 10t Series 063 air lift bags (a 5t version is shown in figure 2.10) to Industrikonsult which were used at a depth of 293m to lift the corner of a subsea oil production template in the Vigdis oil field while seabed stabilisation operations were undertaken [68]. The five bags, providing a total of 50t of lift, were attached to a single lifting point with a complex rigging arrangement. The bags conform to IMCA D 016, an international standard for underwater air lift bags [69], and HSE regulations, and are fitted with a bottom operated dump valve. It is believed that the dump valve is located on the top of the bag (the only place where air will remain even when the bag does not contain much air), and is bottom operated so that it may be operated by a diver or ROV (remotely operated vehicle) without the risk of being hit by the stream of air bubbles that are released.

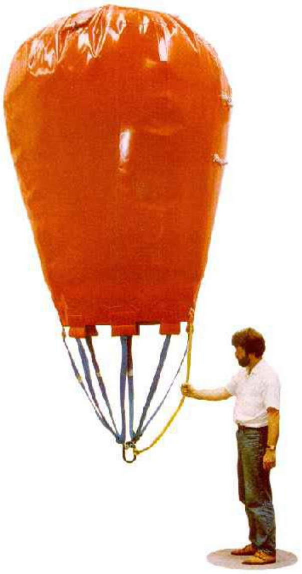


Fig. 2.10 5t air lift bag manufactured by Seaflex Ltd. [70]

The Seaflex 5t air lift bag shown in figure 2.10 appears to have 8 straps attached between the single attachment ring and the open base of the bag, 8 or 16 corresponding panels forming the sides of the bag, and a single top panel. Seaflex air lift bags have RF (radio-frequency) welded seams, a unique ‘de-tangler’ for safe zero visibility rigging, high tensile polyester 3×3 Panama basecloth with UV stabilised PVC coating on both sides (1.650kg/m^2), handling ladders, and a ¾” inflation valve. In accordance with IMCA D 016 they have a 5:1 safety factor. Figure 2.11 shows a design drawing for Seaflex’s largest air lift bag, which provides 35t of lift. Notably, the 9 webbing strops that serve as meridional reinforcement go over the top of the bag, effectively forming 18 strops and reducing the number of strop terminations by 50%. They are 75mm wide [71] and retained at the top by a stainless steel crown flange.

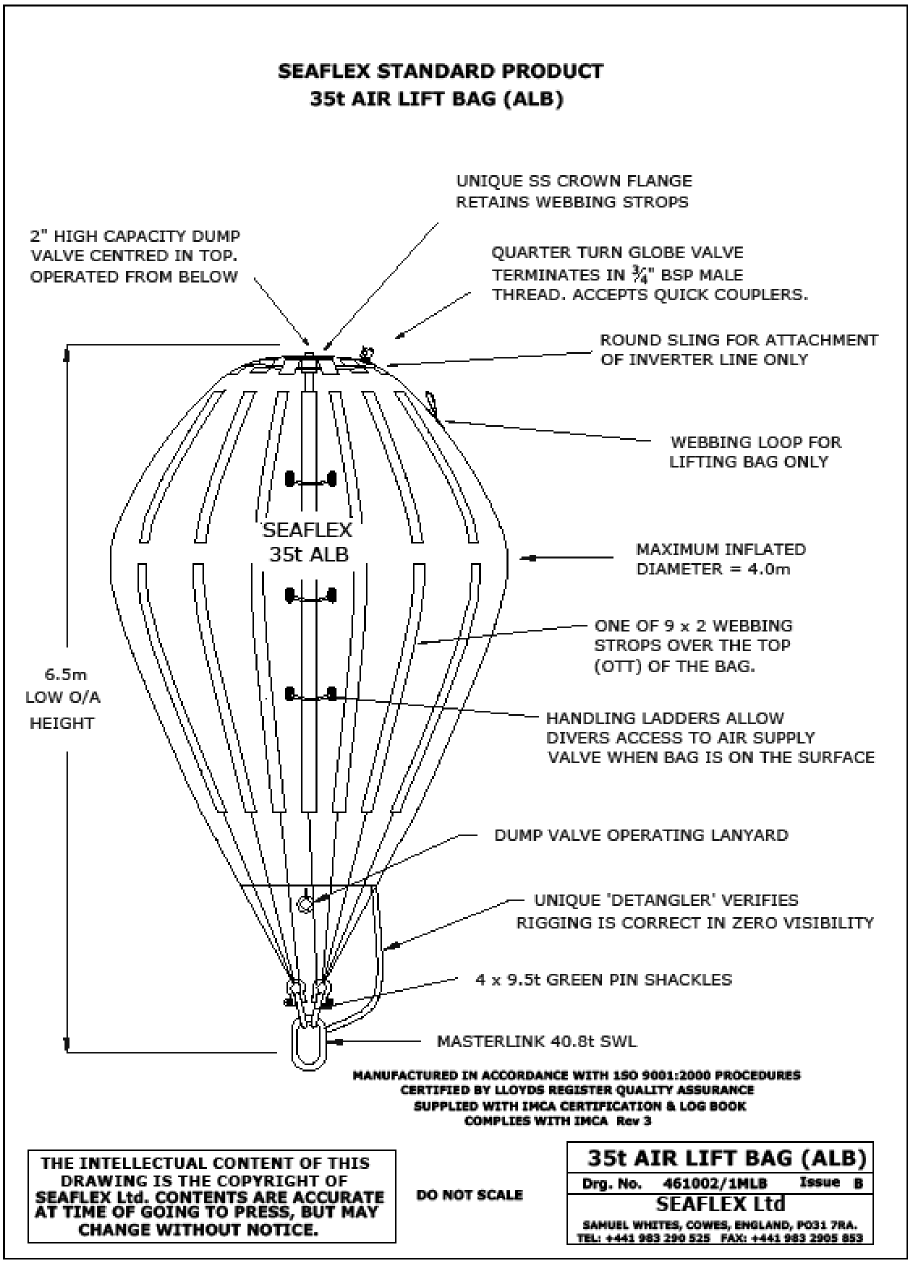

SEAFLEX STANDARD PRODUCT
35t AIR LIFT BAG (ALB)
UNIQUE SS CROWN FLANGE RETAINS WEBBING STROPS
2" HIGH CAPACITY DUMP VALVE CENTRED IN TOP. OPERATED FROM BELOW
QUARTER TURN GLOBE VALVE TERMINATES IN ¾" BSP MALE THREAD. ACCEPTS QUICK COUPLERS.
ROUND SLING FOR ATTACHMENT OF INVERTER LINE ONLY
WEBBING LOOP FOR LIFTING BAG ONLY
SEAFLEX 35t ALB
MAXIMUM INFLATED DIAMETER = 4.0m
ONE OF 9 x 2 WEBBING STROPS OVER THE TOP (OTT) OF THE BAG.
6.5m LOW O/A HEIGHT
HANDLING LADDERS ALLOW DIVERS ACCESS TO AIR SUPPLY VALVE WHEN BAG IS ON THE SURFACE
DUMP VALVE OPERATING LANYARD
UNIQUE 'DETANGLER' VERIFIES RIGGING IS CORRECT IN ZERO VISIBILITY
4 x 9.5t GREEN PIN SHACKLES
MASTERLINK 40.8t SWL
MANUFACTURED IN ACCORDANCE WITH ISO 9001:2000 PROCEDURES
CERTIFIED BY LLOYDS REGISTER QUALITY ASSURANCE
SUPPLIED WITH IMCA CERTIFICATION & LOG BOOK
COMPLIES WITH IMCA Rev 3
THE INTELLECTUAL CONTENT OF THIS DRAWING IS THE COPYRIGHT OF SEAFLEX Ltd. CONTENTS ARE ACCURATE AT TIME OF GOING TO PRESS, BUT MAY CHANGE WITHOUT NOTICE.
DO NOT SCALE
35t AIR LIFT BAG (ALB)
Drg. No. 461002/1MLB Issue B
SEAFLEX Ltd
SAMUEL WHITES, COWES, ENGLAND, PO31 7RA.
TEL: +441 983 290 525 FAX: +441 983 2905 853

Fig. 2.11 Design drawing for a Seaflex 35t air lift bag [71]

JW Automarine Ltd. of Norfolk have been designing and manufacturing inflatable lifting bags since 1972. They make parachute type lifting bags with both open and closed bottoms, and with lifting capacities of up to 50t. Their PR35 35t closed bottom parachute lifting bags were used to raise a North Korean submarine, as shown in figure 2.12. As in figure 2.12, and as advertised on Seaflex and JW Automarine's websites, closed bottom bags are used for lifting items to the surface, though IMCA D 016 clearly recommends that "only open bottom bags should be used where any form of ascent is planned or possible, such as vessel salvage or raising objects from the seabed. Fully enclosed bags should not be used for this purpose." However, integrated pressure relief valves, with adequate flow rate, provide the necessary release of air as these bags rise to the surface. Figure 2.13 shows the base of two of JW Automarine's smaller closed bottom parachute type lifting bags.

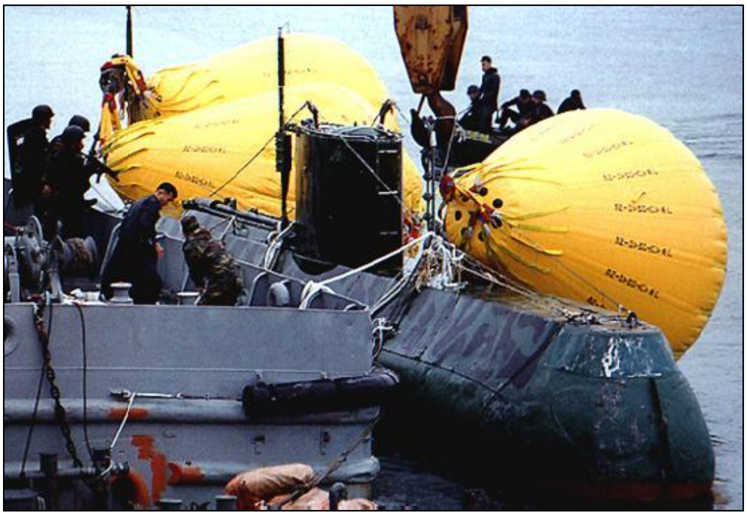


Fig. 2.12 35t closed bottom lifting bags used to raise a North Korean submarine [72]

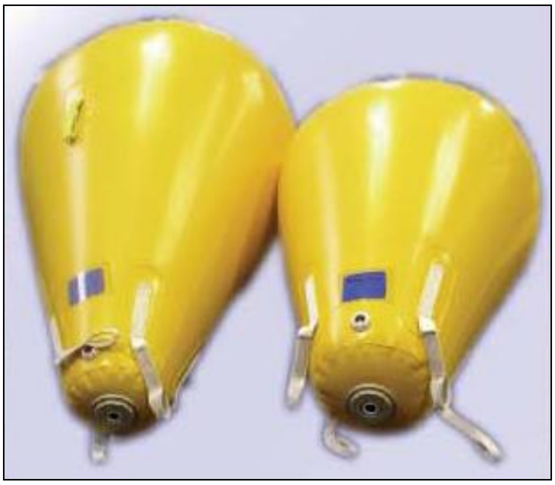


Fig. 2.13 View of the bottom of two small closed bottom lifting bags [73]

JW Automarine’s bags are manufactured from “high tenacity” Trevira™ polyester cloth with PVC coating, and have RF welded seams “for strength and integrity”. The fabric has a temperature operating range of -40°C to 700°C [74]. Their larger lifting bags (3t and upwards) feature a load restraining harness of heavy duty polyester webbing which is fabricated into the skin of the bag, and their smaller bags have load restraints of heavy duty polyester webbing which are sewn and welded to the skin of the bag. All of their lifting bags have a minimum safe working load ratio of 6:1.

The technical specifications of JW Automarine’s range of totally enclosed parachute type lifting bags are given in table 2.4, and their relative sizes are shown in figure 2.14. These bags have a lanyard-operated large dump valve, a ¾” inlet valve, a ~75mm diameter 2psi (0.138bar) overpressure relief valve at the base, and a 4” cap in the base plate that can be removed to form an open bottom bag. The flow rate from the 2psi relief valve is quoted as being in excess of 80cfm (2.265m^3/min). In marine salvage operations, totally enclosed bags are useful because they do not dump air when tilted, which would lead to a reduction in buoyancy and present a risk of dropping the item back to the seabed, so they can be towed and located in fast currents while maintaining buoyancy. The bags are also light enough to be easily transported and pack away very tightly; the largest bag in table 2.4, the 50t PR50TE, packs down to approximately 1.00×1.30m.

Model	Lift Capacity (t)	Overall Height (m)	Overall Dia. (m)	Net Weight (kg)
PR1TE	1	2.38	1.30	16.5
PR1VTE	1.5	2.86	1.30	18.5
PR2TE	2	3.02	1.40	22.0
PR3TE	3	3.95	1.50	41.5
PR4TE	4	4.00	1.75	45.0
PR5TE	5	4.25	2.20	51.0
PR10TE	10	4.85	2.76	98.5
PR30TE	20	5.60	3.67	150
PR35TE	35	6.46	4.16	250
PR50TE	50	7.50	5.00	300

Table 2.4 Technical specifications for JW Automarine’s range of totally enclosed parachute type inflatable lifting bags [75]

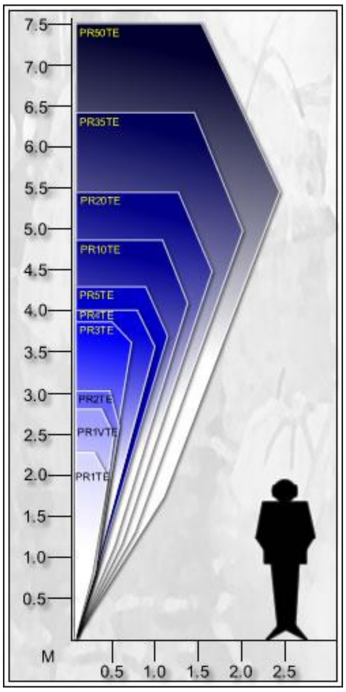


Fig. 2.14 Sizes of JW Automarine's totally enclosed parachute type lifting bags [75]

JW Automarine and Seaflex also manufacture totally enclosed cylindrical lifting bags (as shown in figure 2.15). These are more suitable for use in shallow water than parachute type bags, because of their low profile, and are appropriate for pipe and cable float out. They are also used to reduce the draft of vessels when launching or manoeuvring them in limited depth. While the parachute type bags are all designed with a single attachment point at the base, the cylindrical bags must have several rigging points (9 in the case of JW Automarine's largest – 50t – version, and 14 in the case of Seaflex's largest – 35t – version [76]). They have circumferential webbing, and it is advised that an I-beam be attached against the underside to ensure that they do not tilt towards one end during inflation. Seaflex warn that their cylindrical bags should never be inclined by more than 5° to one side, and it is believed that this is because a tilt to one side will raise the tension in the webbing strops on the upper side.

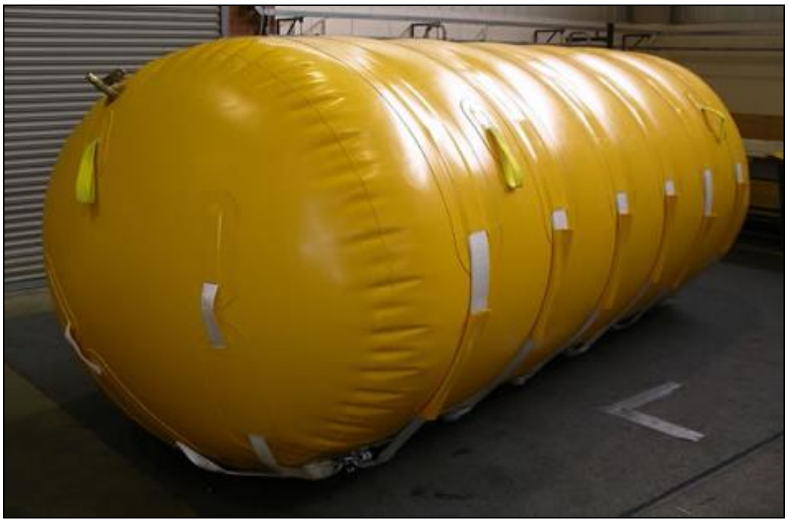


Fig. 2.15 Totally enclosed cylindrical lifting bag [77]

A US-based company called Canflex Inc. manufactures tetrahedron-shaped lifting bags (figure 2.16) in both open and closed bottom variants. However, the maximum lift capacity of their range of tetrahedron bags is only 500kg. They also sell “balloon-shape” (parachute type) and cylindrical lifting bags; their cylindrical bags incorporate two drain valves with air tight caps on the bottom, allowing the bag to be easily drained if water should find its way inside. All their bags feature a lift harness of high strength polyester webbing enclosed in pockets attached to the side of the bag. Their largest (50t) parachute type bag is made of 1.086kg/m^2 fabric coated to a total weight of 3.024kg/m^2, with tensile strength of 18,000N/5cm (meaning that a 5cm wide strip of the fabric can support a maximum tensile load of 18,000N). However, the material thickness is not given. The fabric density and tensile strength for all of the other bags they manufacture can be found at [78].

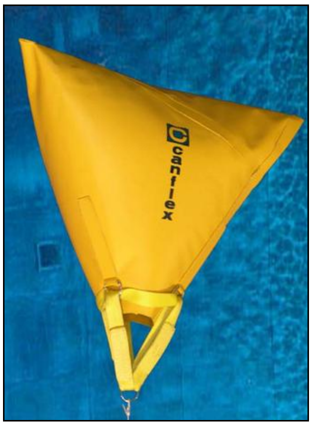


Fig. 2.16 Small tetrahedron lifting bag [78]

Companies such as JW Automarine and Seaflex also manufacture other fabric structures such as water load test weights (used for proof load testing of lifting equipment, a mandatory requirement in many countries), mine recovery bags, ocean towable bladders, land-based liquid and helium storage tanks, seamarks (such as those used to mark out boat racing courses), fenders, and drogues (to provide drag for a towed yacht to ensure that it doesn't overshoot – or crash into – the towing vessel when the towing vessel slows).

2.2.2 Water Load Test Weights

There is much similarity between a water load test weight and underwater lifting bags and open bottom Energy Bags with a similar design. The loads in the bags and the loads that the bags exert on the object they are attached to are almost identical. Ignoring material mass, the net weight per unit volume (in N/m^3) of a water load test weight is given by $(\rho_w - \rho_{atm})g$, where ρ_w is the density of the water, ρ_{atm} is the density of the surrounding air and g is standard gravity, and the net buoyancy per unit volume (also in N/m^3) of an underwater inflatable (be it lifting bag, Energy Bag, or any other device) is given by $(\rho_w - \rho_a)g$, where ρ_a is the density of the compressed air. The density of air increases with pressure because of the compressibility of air, and so the density of the compressed air in an underwater inflatable will always be greater than ρ_{atm} (because the pressure of the air in an underwater inflatable must be greater than or equal to the pressure of the water surrounding the top of the bag). Therefore the weight per unit volume of a water load test weight will be very close to (but higher than) the buoyancy per unit volume of an underwater inflatable, the difference between the two values being proportional to the pressure of the air inside the underwater inflatable (and so depth).

Also, the differential pressure across a fully filled water load test weight differs with distance from the attachment in exactly the same way that the differential pressure across a fully inflated Energy Bag differs with distance from the attachment. The differential pressure at a depth d below the water level in a water load test weight is given by $\rho_w gd$, and the differential pressure at a height h above the water level in an open bottom Energy Bag is given by $\rho_w gh$.

Therefore water load testing of the strength of open bottom Energy Bags and their seabed attachments by simply hanging the Energy Bag in open air and filling it with water would be a conservative test. The direction of the material mass relative to the applied load differs in water load test weights and underwater inflatables, but this difference serves to reduce the net load on an underwater inflatable and its seabed attachment, and increase the net load on a water load test weight and its attachment, so increasing the conservatism of a water load test of an Energy Bag. Of course there is a slight difference in the density of seawater and fresh water, but there is no reason why seawater (or man-made brine) could not be used in such a test.

Figure 2.17 shows a water load test weight manufactured by Aberdeen-based Water Weights Ltd. This bag clearly has lobing, reducing the hoop stress in the fabric (as explained in Chapter 5). The bags are manufactured from individual panels which are welded together [79]. Water Weights manufacture bags up to 35t capacity, the specified dimensions [80] being very similar to those of the 35t lifting bags manufactured by Seaflex and JW Automarine. Several water load test weights can be hung from the same rigging, as shown in figure 2.18. These seven 35t bags were used in May 2010 to provide a total of 220t load in the proof load testing of the main crane aboard a rig involved in the installation of North Sea wind farms. Note that four bags have been hung from the upper rigging point; Seaflex point out that their 35t lifting bags can be used in a three unit cluster to provide 100t buoyancy at a single point. Water Weights use calibrated flowmeters to monitor the volume (and so mass) of water that has been pumped into their bags.

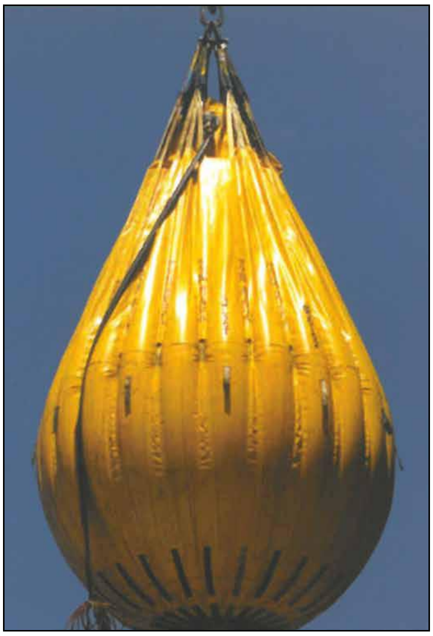


Fig. 2.17 A large water load test weight [80]

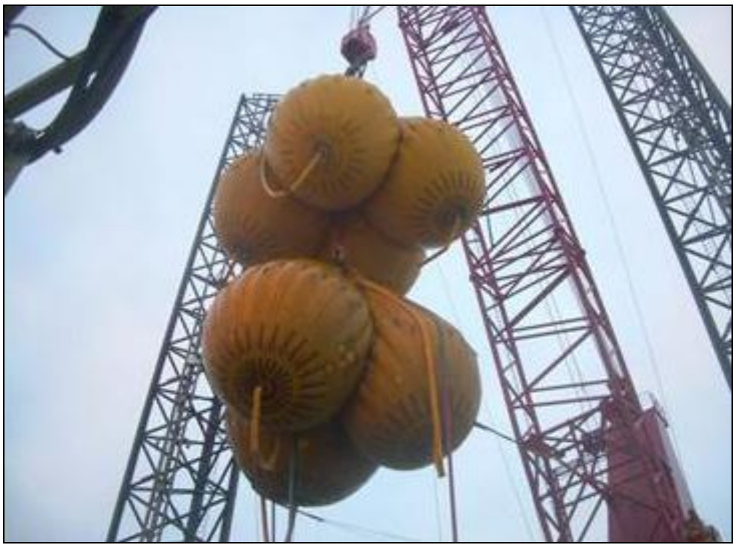


Fig. 2.18 Seven 35t water load test weights configured in one set of four and one set of three [81]

Attaching several Energy Bags to a single rigging point may potentially lower costs of installation, however it would probably not be ideal because the bags would rub against each other as they inflated and deflated, and shock loads and undesirable inflation shapes and load patterns could occur as bags are inflated in amongst other inflated bags. Water load test weights are only hung from the same rigging point to provide a larger load to a single point, and it is assumed that all the bags are filled simultaneously. Also, the filling of water load test weights is an occasional procedure that is carefully monitored by trained engineers that are able to see the bags in person and move them or stop the filling if necessary. Energy Bags will not be monitored in this way, and must be operated without direct supervision for long periods in harsh conditions. Energy Bags could be stacked on top of each other but, for a given fill level, the higher bags would have lower energy densities than the lower bags because they would be in lower pressure water, and a stack of Energy Bags would be more affected by side loading from currents than a single bag, presenting a larger total area and protruding into faster currents closer to the surface.

It should be noted that almost all of the underwater inflatables used for non-military purposes are coloured yellow or orange. This is because these colours are highly visible at sea, both at the surface and underwater. We would be inclined to use these colours for Energy Bags to ensure that they are as visible as possible, in case they need to be located without GPS (for inspection, maintenance or removal).

2.2.3 Correspondence with Manufacturers

In October 2010 I corresponded with Graham Brading at Seaflex, David Letts at Scholle Europe Ltd. (owners of JW Automarine) and Alan Milne at Water Weights, in order to find out information about the sizes of bags they manufacture, what size they think they could manufacture and what they believe limits the size (if anything), the design of the bags and what influences the design, and cost. The information is summarised in table 2.5.

	Seaflex	**JW Automarine**	**Water Weights**
Contact	Graham Brading	David Letts	Alan Milne
Largest ever bag	35t, working on 50t	50t	35t
What limits size	Hoop stress in fabric at largest diameter. Could look at higher strength fabric if necessary	Physical weight to be handled by a diver (325kg for a 50t parachute-type bag). Also manhandling their end would create a problem in packing	35t is largest that they can have proof load tested with safety factor of 6, as their test rig has a SWL of 220t
Cost to buy	£6160 for 35t open base, £6660 for 35t closed base, £5830 for 35t cylindrical bag	£8345 for 50t closed base, £6395 for 35t closed base	£13396 for 35t
Closed parachute type bags?	Are available	Are available	Could be manufactured
Relief valve size and location	2", top dead centre	75mm, not at the very base	
Inflation valve size and location	3/4", top, off-centre	Not top dead centre	
Extra comments	Valve positions are flexible. A partially inflated bag in current would probably suffer abrasion damage at the strop/canopy interface. Not carried out long term testing because their equipment is normally used for recovering shipwrecks or laying oil pipelines, so the duration outside is not usually more than a month or so		All bags have a physically proven safety factor of 6 and are subjected to a proof load test of 2:1 before entering service. Bags are made up of individual panels which are welded together

Table 2.5 Summary of the information obtained from three manufacturers of lifting bags and water load test weights

At 500m depth, and conservatively assuming isothermal expansion, a 35t closed base bag storing air at a pressure equal to 2psi above the hydrostatic pressure at the outside of the bag would store 706.85MJ, equivalent to 0.196MWh. 5.09 35t bags would be required to store 1MWh at this depth, and so the 35t Seaflex fully-enclosed air lift bag, costing £6,660, works out at £33,920/MWh (plus the cost of ballast). Of course, this price would be much improved by using adiabatic expansion, i.e. heating the air before expansion using stored heat that was taken out of the air during compression.

Graham Brading at Seaflex pointed out that the hoop stress in the fabric at the largest diameter of the bag limits the size, however upon inspection of the design drawing (figure 2.11), photographs emailed to me by Graham, and the photographs on Seaflex's website, it was clear that Seaflex's bags don't have any circumferential lobes. It is believed that these would greatly reduce the hoop stress in the fabric. Interestingly, the bags manufactured by Water Weights are lobed, and Alan Milne at Water Weights didn't mention hoop stress as a limiting factor, instead saying that their sizes are limited by the load capacity of the test rig used for the proof load testing.

Seaflex and Water Weights also hire bags by the day, and hire prices were obtained.

2.3 Ballooning

Before embarking on a short history of ballooning, it should be explained that there are essentially two different classes of balloon: the zero pressure (or zero-superpressure) balloon and the superpressure balloon. A zero pressure balloon has zero differential pressure at the base when at flight altitude, and tends to be open at the base to ensure that the differential pressure doesn't become greater than zero. A modern hot air balloon is a zero pressure balloon because it has an open base to allow the air to be heated as necessary. A superpressure balloon has greater than zero differential pressure at the base when at flight altitude, and must have a closed base otherwise lifting gas would escape until the differential pressure at the base becomes zero.

2.3.1 Early Days of Ballooning

The first untethered manned balloon flight took place in Paris on 21st November 1783, by Jean-François Pilâtre de Rozier and the Marquis François Laurent d'Arlandes. The flight lasted approximately 25 minutes and covered a distance of about 9km. The balloon, shown in figure 2.19, had been designed and built the previous year by the Montgolfier brothers and was a hot air balloon (or Montgolfière balloon as they are also known). That first balloon was not remarkably different in style or design to the hot air balloons flown today, and the concept remains exactly the same: a lighter-than-air gas (air heated by a burner in a hot air balloon) enclosed within a balloon envelope has buoyancy because the contained gas is less dense than the surrounding air (the gravity-induced pressure gradient in the atmosphere determining that the direction of the buoyancy force is upwards), and if that buoyancy is greater than the mass of the balloon materials and payload, then the balloon will rise.

Fig. 2.19 Model of the first manned balloon [82]

There are two main types of balloon: the hot air (or Montgolfière) balloon and the gas balloon. A gas balloon uses an unheated lighter-than-air gas (such as hydrogen or helium) as the lifting gas. The first manned flight with a gas balloon took place on 11th December 1783, less than a month after the first hot air balloon flight, when French physicist Jacques Charles and his associate Nicolas Robert flew a hydrogen balloon for two hours, landing 44km away from their launch pad in Paris. This balloon had a wicker gondola, stones in the gondola for ballast, an open base, and a valve at the top. The balloon envelope was surrounded by a net which was attached to the gondola, and these various aspects of the design set the standard for many subsequent generations of balloons. Hydrogen was used as the lifting gas in the early days but was later abandoned in favour of helium because of the dangers of using such a highly flammable gas.

One of the pilots of the first manned hot air balloon flight, Jean-François Pilâtre de Rozier, quickly developed a hybrid balloon, or Rozière balloon, which contains a cell for hot air and a separate cell for an unheated lighter-than-air gas. The advantage of a Rozière balloon is that the pilot has control over the buoyancy (and so altitude) without needing to carry as much fuel as in a hot air balloon. On 15th June 1785, Pilâtre de Rozier made the first flight in a Rozière balloon, attempting to cross the English Channel from France to England, but died when the balloon suddenly deflated and crashed in the Pas-de-Calais region of northern France. Today, Rozière balloons are primarily used for extremely long-duration flights, such as the first manned non-stop round-the-world flight completed by Bertrand Piccard and Brian Jones on 21st March 1999 in the Breitling Orbiter 3 (shown in figure 2.20). Their flight lasted almost 20 days, and set records for time and distance that remain unbeaten by any type of manned balloon.

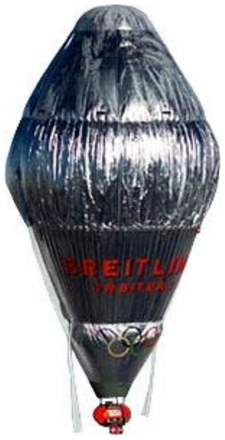


Fig. 2.20 Breitling Orbiter 3, the first balloon to be flown non-stop around the world [83]

In the years following the first balloon flights, balloons were used for military purposes, initially for observation, reconnaissance, and communication, and later on for bombing and to protect critical targets from low-level attacks (as barrage balloons). In 1852 the first engine-powered airship flight was made by Henri Giffard, covering a distance of 27km in a steam-powered balloon.

2.3.2 Scientific Ballooning and Superpressure Balloons

Balloons have been used for scientific purposes since they were first flown. Early balloonists carried thermometers and barometers with them to learn about the variation in atmospheric temperature and pressure with altitude. They also collected air samples, experienced the effects of hypoxia (oxygen deprivation) on the human body, and determined the variation in the Earth’s magnetic field. Unmanned balloon sondes were first flown in the early 1890s, carrying instruments to measure atmospheric temperature, pressure, and humidity at higher altitudes. Later these were equipped with radio transmitters (radiosonde) to send the data back to Earth, and used to measure wind speed and direction (rawinsonde).

Nowadays many radiosondes and rawinsondes are flown every day around the world to gather the data required to accurately forecast the weather using numerical simulations.

Up until the 1930s, balloons had been fully inflated on the ground. If a balloon is fully inflated on the ground, a lot of the lifting gas is ultimately wasted because it vents from the base of the balloon as the balloon rises into lower pressure air and the lifting gas expands. In 1931 and 1932, the Swiss physicist Auguste Piccard (grandfather of Bertrand Piccard and inspiration for Professor Calculus in Hergé's *The Adventures of Tintin*) performed manned flights to altitudes of 16km in a pressurised gondola and by using a balloon envelope with a fully inflated volume of five times the volume required to begin the ascent and only partially inflating the envelope on the ground (as shown in figure 2.21(b) and as in figure 2.22), he got round the problem of wasting gas. Incidentally, in the mid-1930s Piccard realised that a modification of his spherical balloon gondola could also be used for manned exploration of the deep ocean; the gondola was ultimately used on bathyscaphe FNRS-3 which descended to a depth of 4,050m in 1954.

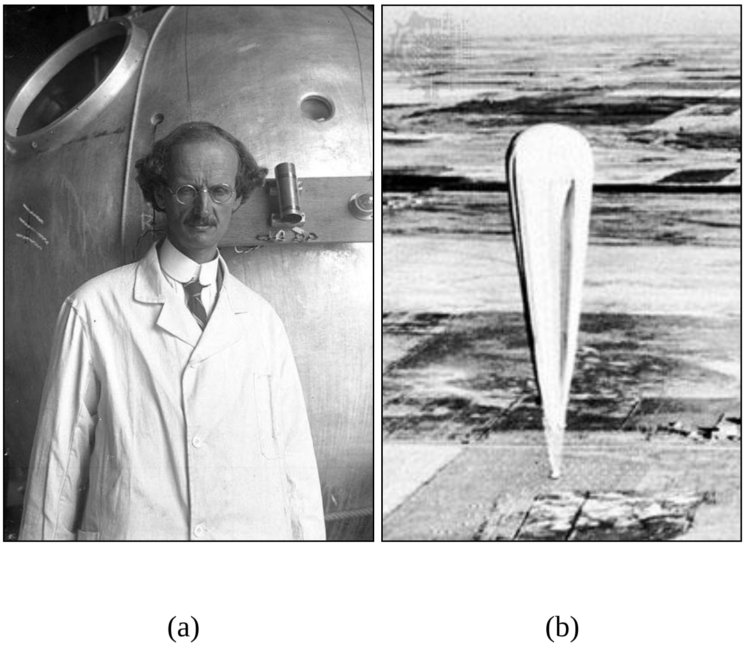


Fig. 2.21 (a) Swiss physicist Auguste Piccard [84]; (b) one of Piccard's balloons at launch [85]

In the 1950s and 1960s, Winzen Research Inc., headed by the German-American aeronautical engineer Otto Winzen, made significant contributions to the design of high-altitude balloons. In order to fly balloons at very high altitudes, it is necessary to reduce the mass of the film as much as possible; Winzen's balloons used very thin films of polyethylene resin, vastly reducing the weight in comparison with the rubber materials used previously.

The balloons were constructed from flat gores that were reinforced by load tapes (or tendons) at their boundaries. Large curvature in the membrane between tendons meant that the differential pressure was carried with low film stresses, allowing such thin materials to be used. These balloons, designed with General Mills Inc., were used in several US military (pre-NASA) projects including Helios and Skyhook (to study phenomena such as cosmic rays, plasma flow, micrometeorites, and sun spots), and Strato-Lab and Manhigh (to test equipment and life-support systems for space missions). The Skyhook balloon is shown in figure 2.22.

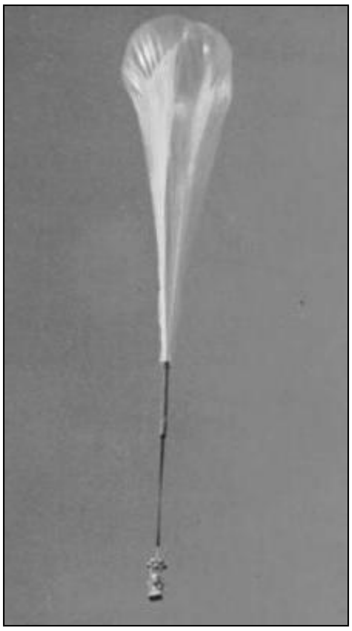


Fig. 2.22 A Skyhook balloon in ascent [86]

In the 1960s, scientists decided to study air circulation patterns over long periods using balloons at constant altitude. However, the zero pressure balloons used at the time were not capable of remaining at constant altitude for periods longer than a few days; during the diurnal (day-night) cycle, heating and cooling causes the gas inside a balloon to expand and contract. As night turns to day and solar heating causes the gas to expand, a zero pressure balloon will vent gas and remain at constant altitude. However, as day turns to night, the gas inside the balloon will contract and the balloon will reduce in volume, thus lowering the buoyancy and requiring ballast to be jettisoned. The flight duration of a zero pressure balloon is therefore limited by the amount of ballast that can be jettisoned. The more ballast carried, the longer the flight duration, but the larger the balloon must be in order to provide the necessary lift. It was found that an overpressurised balloon maintains a roughly constant volume during the diurnal cycle, drastically reducing the amount of ballast that must be carried for long duration flights. Comparative diurnal flight paths of a superpressure balloon and a zero pressure balloon with inadequate diurnal ballasting are shown in figure 2.23.

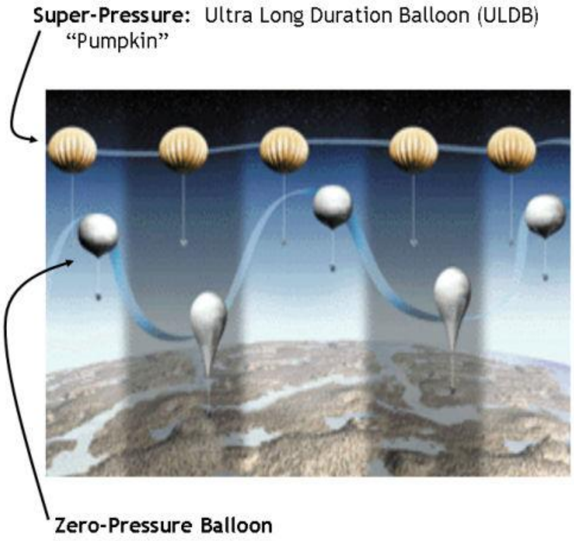


Fig. 2.23 Diurnal flight paths of super-pressure and zero-pressure balloons (with inadequate diurnal ballasting in the zero pressure balloon to maintain constant altitude) [87]

In the mid-60s, Air Force Cambridge Research Laboratories (AFCRL) designed a balloon able to remain at high constant (density) altitudes for long periods (several months or even years). The spherical Global Horizontal Sounding Technique (GHOST) balloons (shown in figure 2.24) were superpressure balloons manufactured from Mylar, a biaxially-oriented polyethylene (boPET) film with high tensile strength, and reached altitudes of up to 24km.

Fig. 2.24 A GHOST superpressure balloon, mid-1960s [88]

Since the 1960s, developments in balloon materials have allowed balloons to reach higher altitudes and carry heavier payloads. Ultra-thin Linear Low Density Polyethylene (LLDPE) film (3.4μm thick) was used on a 60,000m^3 Japanese-designed balloon that was launched to a record altitude of 53km in 2002 [89]. Though not as versatile as aeroplanes and rockets, balloons are still useful because they can remain aloft for long periods without using any energy apart from that which is required at launch. A balloon costs considerably less than a satellite, and the scientific instruments flown can be retrieved and launched again. Nowadays long duration ballooning is often carried out in Antarctica, because the constant daylight in the Antarctic summer means no day-to-night temperature fluctuations on the balloon.

The design shapes of scientific balloons are normally based on conditions at float. Since work at the University of Minnesota in the early 1950s [90], the term natural shape balloon has been used to describe axisymmetric balloon shapes with no circumferential stresses, so that all the stresses in the material are carried meridionally (also known as longitudinally). The natural shape balloon has an infinitesimal amount of excess film in the circumferential direction.

Before moving on we should introduce some notation for differential pressure across the envelope of a pressurised fabric structure. We use p to denote the differential pressure across a balloon at a given height above the base, and p_0 to denote the differential pressure at the base of the balloon. Therefore $p_0 = 0$ in a zero pressure balloon, and $p_0 > 0$ in a superpressure balloon.

The natural shape of a lobed superpressure balloon closely resembles that of a pumpkin, so superpressure natural shape balloons are also known as pumpkin balloons. Pumpkin balloons have been the subject of most of the recent developments in balloon design. Since the mid-90s, NASA has been working on its Ultra Long Duration Balloon (ULDB) system to carry payloads of several tons at constant high altitudes (above 99% of the Earth’s atmosphere) for periods of up to 100 days, as a cheap alternative to certain types of low-Earth-orbit satellites. Initially using spherical balloons similar to the GHOST balloons, NASA soon switched to pumpkin balloons (figure 2.25). Pumpkin balloons have been flown at the edge of space at altitudes of 35km on missions lasting up to 100 days.

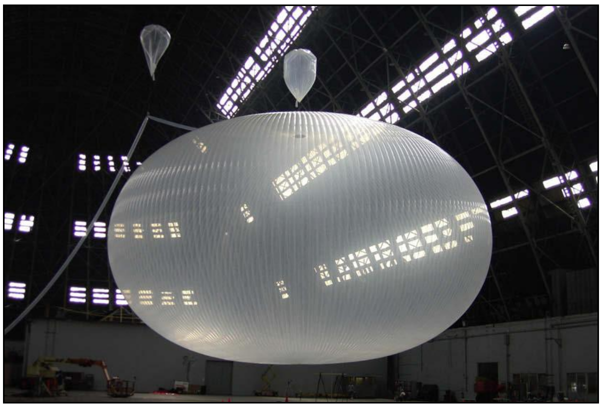


Fig. 2.25 A 27m diameter model Ultra Long Duration Balloon undergoing testing [91]

Typically, balloons are lobed so that the transverse curvature of the membrane is increased and the membrane stresses in the envelope are lower than those in a traditional, smoothly curved balloon (see the Young-Laplace equation, described later in this chapter, for the relationship between pressure, curvature, and tension). Tendons along the cusps carry most of the buoyancy load and ensure that the bag shape is stable, with the lobes forming between the tendons. The deformed shapes of balloons and stresses in the materials are generally found using numerical analysis, in the same way that tension structures are modelled [92],[93]. In finite element models of balloons, as in those of Energy Bags, it is necessary to include a load stiffness matrix that accounts for the changing pressure loads as the envelope deforms to its equilibrium shape [92].

Natural shape balloons are designed with the tendons following the natural shape, and often with the lobes subtending a constant angle or having constant radius of curvature. The "isotensoid" is the natural shape of a balloon subjected to uniform differential pressure (so with p_0 equal to p at the top of the bag), and is more relevant to balloons than Energy Bags because the differential pressure across an Energy Bag varies with height more than the differential pressure across a balloon. This is because the difference between the density of water and the density of air compressed to high pressures (e.g. 50bar) is much greater than the difference between the density of high altitude air and the density of helium, and differential pressure p at height h above the base is given by

$$p = p_0 + (\rho_{\text{ext}} - \rho_{\text{int}})gh. \tag{2.18}$$

For an Energy Bag underwater, p at 10m above the base is approximately 1bar greater than p_0, but for a helium-filled balloon at float altitude, p at 10m above the base is much closer to p_0. In space, a zero gravity environment, there is no pressure gradient and so a pressurised fabric structure in space would have uniform differential pressure across the fabric (and so zero buoyancy). Fabric structures have actually been proposed for use as habitation modules attached to the International Space Station.

Extensive work on the natural shape of both zero pressure and superpressure balloons was carried out by Smalley [94] in the 1960s, who also looked at balloon shapes with nonzero circumferential stress and ascent shapes. Smalley modelled a balloon as an axisymmetric inextensible membrane, and to get around the problem of regularly trying to solve coupled differential equations without a digital computer, presented tables of design factors. Subsequently, Smalley used a digital computer to carry out the computation. Baginski and Winker [95] give a good exposition of the use of axisymmetric inextensible balloon models to generate various balloon shapes.

The boundary conditions for a balloon are lower bulkhead radius, upper bulkhead radius, tension at the base (the vertical component of which must equal the payload), and tension at the top (the vertical component of which must equal the difference between the weight of the upper bulkhead and the differential pressure force on the bulkhead). Often it is assumed that there are no bulkheads (i.e. the radii at the top and bottom are both zero and there is no mass acting at the top of the balloon), so the top of the membrane meets the centreline of the balloon at right-angles, as in the ascent shapes shown in figure 2.26. Ascent shapes for balloons with sealed bases are found by reducing p_0. Numerical difficulties are encountered when looking at very large negative pressure differences, which Smalley [94] deals with by using a rope section for the lower part of the balloon and Baginski *et al* [96] deal with using a parallel shooting method, while using an ordinary shooting method to solve the model equations for the fully inflated balloon.

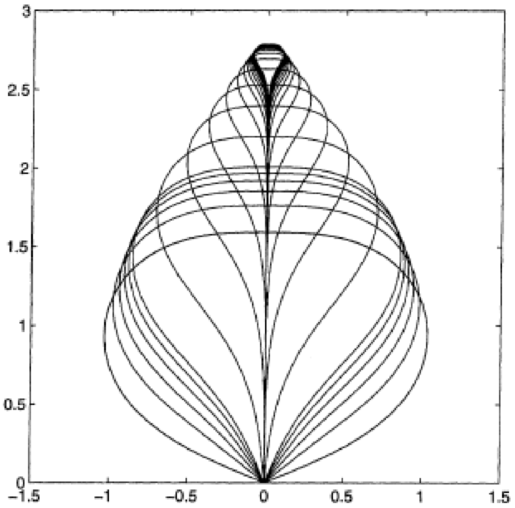


Fig. 2.26 Ascent shapes of a natural shape balloon with a sealed base, found using the parallel shooting method [96]

Superpressure balloons were first studied by Taylor in 1919 [97] in his work on the shapes of parachutes. He derived elliptic functions that describe parachute curves, and found that the curves extend to form an oval, the parachute curve being the upper part of the oval down to where the tangent to the curve has the same angle as the parachute cords. This full oval is the isotensoid shape described earlier (also known as the Taylor surface), and Taylor found that the ratio of the minimum to the maximum diameter of the isotensoid is almost exactly 0.6. Actual parachutes (figure 2.27) have a rather flatter shape to that of the isotensoid, and Taylor proposed that this is probably due to an increase in pressure difference between the inside and outside near the edge of the parachute. Taylor also showed that the isotensoid shape does not depend upon the elasticity of the membrane material by making a model of a rubber balloon with strengthening in the form of meridional threads, and then comparing its shape with that of a linen balloon with similar meridional strengthening. Notably, the linen balloon is described as consisting of two flat, circular pieces of linen joined together at their edges; the pumpkin-shaped Energy Bag prototypes manufactured for us by Thin Red Line Aerospace (detailed later) are also made of two flat circular pieces of material joined together at their edges.

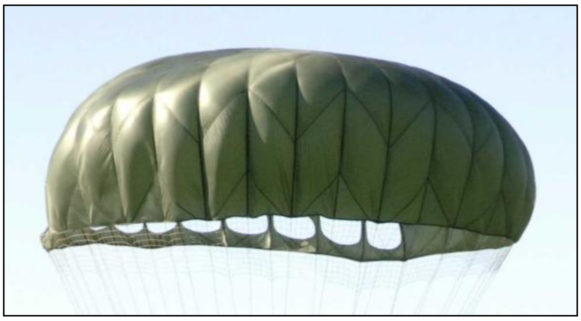
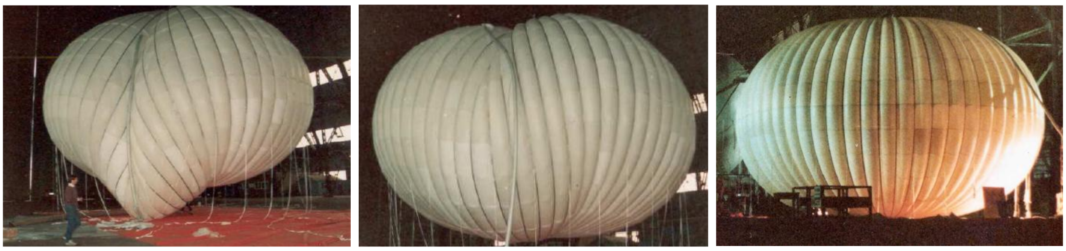


Fig. 2.27 A military parachute, with a similar shape to that of the top half of a pumpkin balloon [98]

Problems with the stability of pumpkin balloons have been encountered [92]. In 1984, Julian Nott designed the first pumpkin balloon in an attempt to become the first person to circumnavigate the Earth in a balloon. However, upon inflation it was found that the 64-lobe balloon adopted an asymmetrical shape with clefts forming in the balloon even at low pressures, and it was not until 4 lobes were removed that the balloon’s shape became symmetrical (figure 2.28).



(a) (b) (c)

Fig. 2.28 Nott’s Endeavour superpressure balloon (a) 64 lobes; (b) 62 lobes; (c) 60 lobes [99]

Calladine [100] showed that the volume enclosed by a lobed pumpkin balloon whose lobe cross-sections subtend a constant angle (so-called constant angle (CA) balloons) increases – if the number of lobes and the bulge formed by the lobes are sufficiently large – for certain small-amplitude inextensional deformations of the balloon. This lowers the potential energy of the balloon (ignoring strain energy, potential energy $E \approx -PV$) and indicates the presence of the instability encountered by Nott. Later Lennon and Pellegrino [101] showed that balloons with constant radius of curvature along the lobes (CR balloons) can also become unstable but are more stable than CA balloons, having higher buckling pressures. They found that the volume enclosed by a CR balloon decreases for small amplitude deformations but increases for sufficiently large deformations. They also looked at the dependence of the stability of a balloon on the size of the perturbation, the deformation mode, and the number of lobes in the structure.

Wakefield [102] carried out the first numerical simulations of the stability of pumpkin balloons, with Pagitz, Pellegrino, and James [92],[103] simplifying the computation of the buckling pressure by taking into account the symmetry of a pumpkin balloon and using closed form solutions for the symmetry transformation matrices. Pagitz and Pellegrino [104] optimised the cutting patterns of pumpkin balloons so that their stability is maximised for a given stress constraint, showing that maximising the buckling pressure of a balloon is equivalent to minimising the surface area of the cutting pattern, resulting in a fully-stressed design. Like Calladine, Pagitz and Pellegrino found that balloons with flatter lobes are more stable, and that the buckling pressure varies with an inverse power-law of the number of lobes [92].

2.3.3 Wrinkling of Membranes

Thin flexible membranes transmit loading via in-plane tension only. Under uniaxial compression a membrane collapses in the direction of compression, and wrinkles appear perpendicular to the direction of compression. The membrane retains its tensile load capacity in the direction perpendicular to the compression. In this work we refer to this state as uniaxial wrinkling. Under biaxial compression a membrane loses its ability to sustain tension in any direction, and we refer to a membrane in this state as undergoing biaxial wrinkling (or being slack).

The elastic matrix of a membrane element in a taut stress state is given by

$$D_t = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix} \tag{2.19}$$

where E is the Young's modulus of the material and ν is its Poisson's ratio. In a wrinkling state, this matrix should be modified to reduce the stiffness of the material in the direction of wrinkling to near zero. The simplest way of attempting to model wrinkling is to set any compressive principal stresses to zero, along with the associated stiffness matrix components. However, poor convergence has been observed when trying to use this method [105].

The first finite element solution to incorporate wrinkling was the Iterative Materials Properties (IMP) model developed by Miller *et al* [106],[107]. For a wrinkling element, Miller *et al* modelled the strain in the direction of wrinkling by using a variable effective Poisson's ratio for the element. They used the "wrinkled" elastic matrix

$$D_w = \frac{E}{4}\begin{bmatrix} 2(1+P) & 0 & Q \\ 0 & 2(1-P) & Q \\ Q & Q & 1 \end{bmatrix}, \tag{2.20}$$

where $P = (\varepsilon_x - \varepsilon_y)/(\varepsilon_1 - \varepsilon_2)$ and $Q = \gamma_{xy}/(\varepsilon_1 - \varepsilon_2)$; $\varepsilon_x, \varepsilon_y, \gamma_{xy}$ are the engineering components of plane strain; $\varepsilon_1, \varepsilon_2$ are the major and minor principal strains; and the directions 1 and 2 are parallel and perpendicular to the wrinkles, respectively. Clearly this matrix depends upon the axis system used, because of the dependence of *P* and *Q* upon ε_x,

ε_y and γ_{xy}. This is unlike the elastic matrix of a taut element (as shown in equation (2.19)), which is independent of the axis system used.

Adler [108] implemented the IMP model with a user-defined material subroutine in the ABAQUS commercial FEA software to find the deformed shapes of wrinkling balloons. However, it was found that the solution tended to diverge in the presence of many slack (biaxial wrinkling) regions. The IMP subroutine calculates the principal stresses and strains using D_t in order to check if any elements are wrinkling, and then the stresses and strains are recalculated using D_w where appropriate.

Use of the correct wrinkling criterion is critical to accurately judge whether the membrane is taut, wrinkled or slack. Several wrinkling criteria have been proposed, and are detailed here:

1) Wrinkling criterion based on principal stresses:

If $\sigma_2 > 0$, taut,
If $\sigma_2 \leq 0$ and $\sigma_1 > 0$, wrinkled,
If $\sigma_2 < 0$ and $\sigma_1 \leq 0$, slack.

This criterion misjudges the idea of “slackness” and allows for positive strain because of Poisson’s ratio effects [109],[110]. Positive strain in a slack element could be problematic in a revised materials model (to model wrinkling).

2) Wrinkling criterion based on principal strains:

If $\varepsilon_2 > 0$, taut,
If $\varepsilon_2 \leq 0$ and $\varepsilon_1 > 0$, wrinkled,
If $\varepsilon_2 < 0$ and $\varepsilon_1 \leq 0$, slack.

This criterion misjudges the idea of “tautness”; due to the effects of Poisson’s ratio it is possible to have negative minor principal strain while the minor principal stress is positive [106].

3) Wrinkling criterion based on a combination of principal stresses and principal strains:

If $\sigma_2 > 0$, taut,
If $\sigma_2 \leq 0$ and $\varepsilon_1 > 0$, wrinkled,
If $\varepsilon_1 \leq 0$, slack.

This is the most accurate criterion of the three. Though there could be times when both principal stresses are negative and the element is still not classed as slack, this would be because the major principal strain is positive.

The shape of wrinkles can be well approximated with a half sine wave in the longitudinal direction, with linearly-varying transverse wavelength in the transverse direction [111], and the bending stiffness of a membrane, although small, plays a key role in the shape and amplitude of wrinkles. A phenomenon known as mode jumping, in which the membrane suddenly changes from one wrinkled shape to another, has been observed in experiments [111] and numerical simulations [105]. It has also been found that during load cycling, corresponding forward and reverse mode jumps (e.g. a jump between 8 and 9 wrinkles and back again) occur at different displacements.

In this work we will not concern ourselves with the shape and frequency (“waviness”) of any wrinkles that may arise, and so will not include any bending stiffness.

2.3.4 The Young-Laplace Equation

The Young-Laplace equation relates the pressure difference across the interface between two static fluids to the surface tension in the interface, and takes the form

$$p = \frac{T_1}{r_1} + \frac{T_2}{r_2}, \tag{2.21}$$

where p is the differential pressure across the interface, T_1 and T_2 are the components of surface tension (per unit perpendicular length) in the directions of principal curvature, and r_1 and r_2 are the principal radii of curvature. At all points along the centreline of a symmetric lobe, the directions of principal curvature are perpendicular to the centreline and along the centreline (i.e. circumferential and meridional). Therefore equation (2.21) can be rewritten as

$$p = \frac{T_c}{r_c} + \frac{T_m}{r_m}, \tag{2.22}$$

where T_c and T_m are the circumferential and meridional tensions (per unit perpendicular length) and r_c and r_m are the circumferential and meridional radii of curvature.

The Ultra High Performance Vessel (UHPV), described in Chapter 5, is a design of lobed inflatable that has no meridional tension because of tendon shortening and the inclusion of "bellows". Since $T_m = 0$ in a UHPV, the relationship between the differential pressure, circumferential tension and circumferential radius of curvature is very simple.

2.4 Techniques in Nonlinear Finite Element Analysis

Finite element analysis (FEA) is used in two of the shape analysis tools presented in this thesis, however analysis of the deformed shape of an Energy Bag is a geometrically nonlinear problem. There are two main reasons for this nonlinearity: 1) the nodes in an FE model of a cable-reinforced fabric bag are likely to undergo large deflections because the bending stiffness in cables and membranes is negligible in comparison with the elastic modulus of the material, and 2) pressure loading requires follower loads because the force of pressure always acts normal to the surface. Here we detail the three standard techniques used in the solution of a geometrically nonlinear problem: the Newton-Raphson solution procedure, load incrementation, and displacement incrementation. We also look at ways of dealing with boundary nonlinearity, when the structure comes into contact with itself or other structures.

2.4.1 The Newton-Raphson Method

The Newton-Raphson method, also known simply as Newton's Method, is a well-known root-finding algorithm and one of the most powerful techniques available for solving nonlinear problems. It is first in the class of Householder's methods, requiring only knowledge of the value of the function and its first derivative, and is succeeded by Halley's method which also requires knowledge of the second derivative. The Newton-Raphson method is an iterative procedure which generates successively closer approximations of the root.

Given a function $f(x) = 0$ and its derivative $f'(x)$, we begin with an approximation of the root, x_n. If the function is reasonably well-behaved, a better approximation of the root is found by following the tangent to the curve at $(x_n, f(x_n))$ down to the place where the tangent line intersects the x-axis, $(x_{n+1}, 0)$. x_{n+1} is found using

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{2.23}$$

This process is repeated until convergence has occurred. Figure 2.29 shows 2 iterations of the Newton-Raphson method.

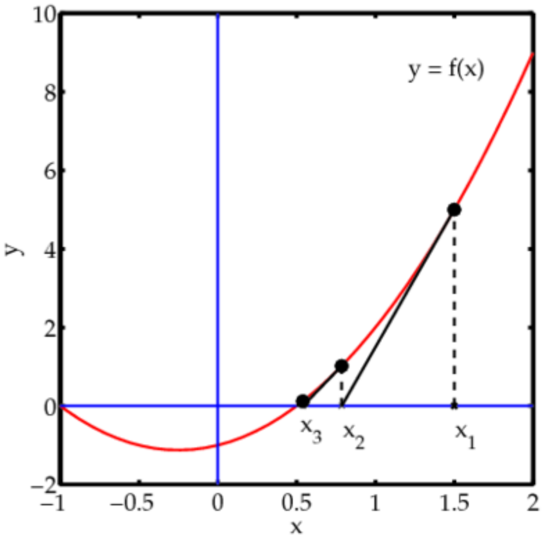


Fig. 2.29 2 iterations of the Newton-Raphson method [112]

The Newton-Raphson method can be used to solve systems of equations, and it is used in this way in finite element analysis. Given a vector (u_n) of an approximation of the solution to the set of equations $f(u) = 0$, a new vector which should be a closer approximation of the solution is given by

$$u_{n+1} = u_n - J_f(u_n)^{-1} f(u_n), \tag{2.24}$$

where $J_f(u_n)$ is the Jacobian of $f(u_n)$ – a matrix of partial derivatives of $f(u_n)$. If the structure has k nodes, each with 3 degrees of freedom, then the vector u has the form

$$u = \begin{bmatrix} u_{1x} & u_{1y} & u_{1z} & u_{2x} & \ldots & u_{kz} \end{bmatrix}^T, \tag{2.25}$$

and the Jacobian $J_f(u)$ is given by

$$J_f(u) = \begin{bmatrix} \frac{\partial f_{1x}}{\partial u_{1x}} & \frac{\partial f_{1x}}{\partial u_{1y}} & \frac{\partial f_{1x}}{\partial u_{1z}} & & \frac{\partial f_{1x}}{\partial u_{kx}} & \frac{\partial f_{1x}}{\partial u_{ky}} & \frac{\partial f_{1x}}{\partial u_{kz}} \\ \frac{\partial f_{1y}}{\partial u_{1x}} & \frac{\partial f_{1y}}{\partial u_{1y}} & \frac{\partial f_{1y}}{\partial u_{1z}} & \cdots & \frac{\partial f_{1y}}{\partial u_{kx}} & \frac{\partial f_{1y}}{\partial u_{ky}} & \frac{\partial f_{1y}}{\partial u_{kz}} \\ \frac{\partial f_{1z}}{\partial u_{1x}} & \frac{\partial f_{1z}}{\partial u_{1y}} & \frac{\partial f_{1z}}{\partial u_{1z}} & & \frac{\partial f_{1z}}{\partial u_{kx}} & \frac{\partial f_{1z}}{\partial u_{ky}} & \frac{\partial f_{1z}}{\partial u_{kz}} \\ & \vdots & & \ddots & & \vdots & \\ \frac{\partial f_{kx}}{\partial u_{1x}} & \frac{\partial f_{kx}}{\partial u_{1y}} & \frac{\partial f_{kx}}{\partial u_{1z}} & & \frac{\partial f_{kx}}{\partial u_{kx}} & \frac{\partial f_{kx}}{\partial u_{ky}} & \frac{\partial f_{kx}}{\partial u_{kz}} \\ \frac{\partial f_{ky}}{\partial u_{1x}} & \frac{\partial f_{ky}}{\partial u_{1y}} & \frac{\partial f_{ky}}{\partial u_{1z}} & \cdots & \frac{\partial f_{ky}}{\partial u_{kx}} & \frac{\partial f_{ky}}{\partial u_{ky}} & \frac{\partial f_{ky}}{\partial u_{kz}} \\ \frac{\partial f_{kz}}{\partial u_{1x}} & \frac{\partial f_{kz}}{\partial u_{1y}} & \frac{\partial f_{kz}}{\partial u_{1z}} & & \frac{\partial f_{kz}}{\partial u_{kx}} & \frac{\partial f_{kz}}{\partial u_{ky}} & \frac{\partial f_{kz}}{\partial u_{kz}} \end{bmatrix}. \tag{2.26}$$

In the FEA models presented in this thesis, the Jacobian is calculated analytically (see the derivations of the stiffness matrices and load stiffness matrices). This leads to more accurate values for the partial derivatives than if numerical differentiation of the force and load vectors were used. We seek the vector of displacements (u) at which the force residuals are zero, where the vector of force residuals (r) is the difference between the vector of applied loads (f_{app}, e.g. from differential pressure and material mass) and the vector of internal reaction forces (f_{reac}, e.g. from tension in the structure), given by

$$r = f_{reac} - f_{app}. \tag{2.27}$$

So we seek the root of the equilibrium equation, $r(u) = 0$.

There are a few difficulties with Newton-Raphson's method that must be understood. There are some circumstances in which the method may fail to converge:

- If the initial approximation of the root (x_0) is too far from the root.
- If the derivative of the function is discontinuous.
- If the derivative of the function is close to zero in any part of the function that is encountered during the iterations (so particularly near the root).

All of these issues had to be considered when developing the finite element analysis tools.

There is a slightly different version of the Newton-Raphson method known as the modified Newton-Raphson method. In the standard Newton-Raphson method, the gradient of the tangent to the function ($f'(x)$ or $J_f(x)$) is calculated for each iteration, however in the modified Newton-Raphson method the gradient is updated after several iterations. This usually requires more iterations to find the solution, but it has the benefit of being less computationally expensive per iteration (on average).

2.4.2 Load Incrementation

It is possible to improve the chance of finding a solution using the Newton-Raphson method by utilising a load incrementation procedure [113]. Instead of immediately applying the full load, a load incrementation procedure works by only applying a fraction of full load, and finding the root for this load. The load fraction (ϕ) is then increased, and the converged solution for the previous load fraction is used as the initial approximation of the root with the new load fraction. The load fraction is repeatedly increased in this way until full load is applied (which is when $\phi = 1$). Incorporating a load fraction, equation (2.27) becomes

$$r = f_{reac} - \phi f_{app}. \tag{2.28}$$

The load fraction must also be included in the Jacobian of the force residuals.

This procedure works on the idea that the initial approximation of the root will be closer to the solution for a small fraction of full load than it is to the solution for full load. A new load fraction can be chosen by assessing a prospective load fraction’s effects on displacements, rotations, or the stress-state after a small number of iterations with the new load fraction (perhaps even after just one iteration).

2.4.3 Displacement Control

In some cases the chances of finding a converged solution can be improved by only using a fraction (γ) of the node displacements calculated using the Newton-Raphson method. So equation (2.23) becomes

$$x_{n+1} = x_n - \gamma \frac{f(x_n)}{f'(x_n)}. \tag{2.29}$$

The displacement fraction used in each iteration can be calculated to limit the maximum displacement to a fraction of some geometrical dimension (e.g. the undeformed meridional length of the Energy Bag). Introducing a displacement fraction in this way slows down the rate of convergence, but in some cases it is necessary in order to find a solution.

2.4.4 Boundary Nonlinearity

Boundary nonlinearity is encountered when the structure being studied comes into contact with itself or another structure. The other structure can be modelled as an elastic body; however, care should be taken not to introduce a discontinuity into the gradient of the force-displacement curve which can cause problems with the convergence of the solution procedure. Spyrakos and Raftoyiannis [114] note that if nonlinear springs are used to model contact problems, their stiffnesses should be chosen with care in order to simulate nonlinear behaviour without causing numerical instability. They also point out that in contact problems, a refined mesh should be used in the region of contact, because high stresses and strains can develop in the area of contact and because of the need to closely monitor the size of the contact area for accurate stress analysis.

Wrinkling elements also introduce boundary nonlinearity – they have standard stress and strain characteristics when fully stressed, but they have very little stiffness in compression. In all cases of rapidly increasing or reducing the stiffness of an element to simulate contact or wrinkling, it is best if the change is implemented in such a way that the stress-strain curve and its derivative (the element stiffness) remain continuous.

2.5 Notes on the Design of Energy Bags

Unlike balloons, tensile structures, and underwater inflatables, which can all be maintained and inspected both in and out of service without too much difficulty, an Energy Bag must be designed with consideration of the substantial depth at which it will operate – divers cannot go down to such depths and ROVs must be used for installation and all other work. An Energy Bag must work as intended over the course of its life, with relatively few inspections and little to no maintenance. Consideration must be given to the effects of saltwater and sand on the life of the bag materials (which may be limited by increasing permeability).

While the lifting bag guidelines IMCA D 016 may not apply to Energy Bags (because they are not used for lifting and will be designed to remain in place), they are still very useful and shall be consulted regularly. Section 7.2 of IMCA D 016 details snatch loading that can occur in underwater lifting bags. 7.2(a) warns of snatch loads induced "when the bag is used in water depths shallow enough for wave action to cause snatching and rapid changes in the dynamic loading." Wave loading should not affect Energy Bags anchored at substantial depth, but should be considered for bags anchored at shallow depths (particularly near-shore prototypes). 7.2(c) warns of snatch loads induced "when the lift bag is incorrectly rigged", which should also be considered for Energy Bags.

Balloons are designed with material mass in mind, with the aim of maximising net buoyancy. However, material mass will not be an issue for Energy Bags because all it will do is slightly reduce the required ballast and required strength of anchorage. Long duration balloons are also designed to maintain their volume (and so buoyancy) throughout the day/night cycle, and modern scientific balloons are overpressurised at the base for this reason. Again, the day/night cycle and maintenance of buoyancy will not be an issue for Energy Bags, because the seawater temperature at depths such as 500m is not greatly affected by the day/night cycle, and because maintaining maximum buoyancy at all times is not a requirement of an Energy Bag. Because they need not be overpressurised at the base, it may be the case that Energy Bags are not affected by the stability problems that have confronted the designers of pumpkin balloons. Lifetime cost is the main focus of Energy Bag design, but cost does not seem to be such an issue for balloon designers, who have large research budgets and are driven by the need to minimise the mass of the balloon materials.

An Energy Bag must not dump air from the base if it is tilted because of currents (because dumping air reduces the amount of stored energy), therefore it makes sense to use a fully enclosed bag (even if the bag is designed with maximum p_0 less than or equal to zero) or to make sure that, at full inflation, the water level inside an open base bag is high enough to always remain inside the bag even with some tilting. If the bag is fully enclosed, adequate pressure relief valves should be incorporated into the base to ensure that the pressure of the contained air never exceeds the design value. Obviously these must be tested and set before installation.

JW Automarine point out that their lifting bags can be tested by hanging them upside down from a gantry or crane and filling them with water. We could test open bottom or zero pressure fully enclosed bags in this way. This would be a conservative test because in the test, the bag mass would act in the same direction as the loading, whereas underwater it would act in the opposite direction.

It should be noted here that a subsea CAES plant using Energy Bags should not necessarily just comprise a single bag – a number of bags could be located near to each other and attached by pipelines to a manifold, which in turn is connected by pipeline to the compression/expansion machinery. Using a larger number of smaller bags may be advantageous in case a bag were to fail (in which case only a small amount of storage capacity is lost), and the ballast for smaller bags may be easier to transport and install.

We will use static modelling of an Energy Bag, because it is anticipated that dynamic loading will be insignificant: the bags will be inflated and deflated slowly, and the currents at 500m depth are not very fast and do not rapidly change direction.

Chapter 3

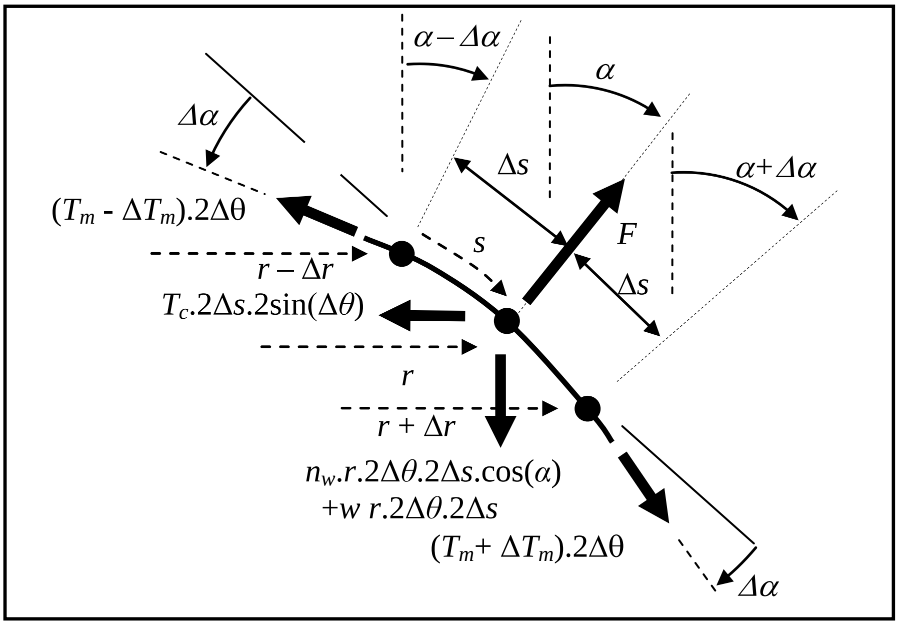
Modelling Axisymmetric Structures Using Coupled Ordinary Differential Equations

If it is assumed that an Energy Bag is axisymmetric and inextensible, it is possible to derive a system of coupled ordinary differential equations (ODEs) that, when solved, describe the deformed shape of the bag. This is presented here, including meridional, circumferential, and vertical loading. Vertical loading is necessary to model self-weight and mass hanging from the inside of the bag, which it is believed will reduce the required amount of reinforcement material and may lower the cost of the bag (per unit of energy stored). The ODEs are solved using one of the in-built functions in Matlab, ode45. Deformed balloon shapes have been found in the past using the method adopted in this chapter (see [96] and [115]) but this work represents the first time the method has been used to find the deformed shapes of Energy Bags.

3.1 Derivation of the Coupled Ordinary Differential Equations

Figure 3.1 shows an infinitesimal patch of membrane of length $2\Delta s$ resisting differential pressure, *p*, across the membrane, which gives rise to a differential pressure force, *F*, acting normal to the centre of the patch. The patch covers a circumferential angle of $2\Delta\theta$, and this curvature leads to circumferential tension in the membrane of T_c. The normal to the patch is at an angle α to the vertical, and the patch has meridional curvature such that the tangent to each end of the patch is at an angle $\Delta\alpha$ to the tangent to the centre of the patch. This curvature leads to meridional stress in the patch of T_m at the centre, $T_m - \Delta T_m$ at the upper end and $T_m + \Delta T_m$ at the lower end. The local radius of the bag is *r* at the centre of the patch, $r - \Delta r$ at the upper end and $r + \Delta r$ at the lower end. Finally, the centre of the patch is at a distance *s* along the membrane from the top of the bag.

Here, the circumferential tension T_c is in force per unit meridional length of membrane (N/m) and the hanging load n_w is in force per unit of projected area onto the seabed (N/m^2). Self-weight of the bag (w) is in force per unit of surface area (N/m^2), and the surface area of the patch is A_p. The meridional tension T_m is force per unit circumferential angle (N/rad).



$\alpha - \Delta\alpha$
α
$\Delta\alpha$
Δs
$\alpha + \Delta\alpha$
$(T_m - \Delta T_m).2\Delta\theta$
s
F
$r - \Delta r$
Δs
$T_c.2\Delta s.2\sin(\Delta\theta)$
r
$r + \Delta r$
$n_w.r.2\Delta\theta.2\Delta s.\cos(\alpha)$
$+w\ r.2\Delta\theta.2\Delta s$
$(T_m + \Delta T_m).2\Delta\theta$
$\Delta\alpha$

Fig. 3.1 Section view of a patch of membrane subjected to a differential pressure force F

The change in radius r with distance s along the membrane is easily derived.

$$\frac{dr}{ds} = \cos\alpha \tag{3.1}$$

The change in height is found in the same way.

$$\frac{dh}{ds} = -\sin\alpha \tag{3.2}$$

Subjected to a differential pressure p between inside and outside (derived later on), the applied force on the patch is

$$F = pA_p, \tag{3.3}$$

where the area of the patch is given by

$$A_p = r2\Delta\theta 2\Delta s\,. \tag{3.4}$$

Therefore the differential pressure force on the patch is

$$F = pr2\Delta\theta 2\Delta s\,. \tag{3.5}$$

For force equilibrium this must be equal to the sum of reaction forces in the opposite direction:

$$\begin{aligned}&(T_m + \Delta T_m + T_m - \Delta T_m)2\Delta\theta \sin\Delta\alpha + T_c\, 2\Delta s 2\sin\Delta\theta \sin\alpha\\ &+ n_w r2\Delta\theta 2\Delta s \cos\alpha\cos\alpha + wr2\Delta\theta 2\Delta s\cos\alpha\,.\end{aligned} \tag{3.6}$$

So the force balance in the direction of the normal, after small angle approximations, is

$$\begin{aligned}pr2\Delta\theta 2\Delta s &= 2T_m\, 2\Delta\theta\Delta\alpha + T_c\, 2\Delta s 2\Delta\theta\sin\alpha\\ &+ (n_w\cos\alpha + w)r2\Delta\theta 2\Delta s\cos\alpha\,.\end{aligned} \tag{3.7}$$

Rearranging for $\Delta\alpha$ and dividing through by $4T\Delta\theta$,

$$\Delta\alpha = \frac{pr - T_c\sin\alpha - (n_w\cos\alpha + w)r\cos\alpha}{T_m}\Delta s\,. \tag{3.8}$$

Taking the limit as Δs goes to zero gives

$$\frac{d\alpha}{ds} = \frac{pr - T_c\sin\alpha - (n_w\cos\alpha + w)r\cos\alpha}{T_m}\,. \tag{3.9}$$

Balancing forces in the direction of the tangent to the centre of the patch moving from top to bottom,

$$\begin{aligned}&(T_m + \Delta T_m)2\Delta\theta\cos\Delta\alpha + n_w r2\Delta\theta 2\Delta s\cos\alpha\sin\alpha + wr2\Delta\theta 2\Delta s\sin\alpha\\ &= (T_m - \Delta T_m)2\Delta\theta\cos\Delta\alpha + T_c\, 2\Delta s 2\sin\Delta\theta\cos\alpha\,.\end{aligned} \tag{3.10}$$

Taking small angle approximations, dividing through by $4\Delta\theta$, and rearranging for ΔT_m,

$$\Delta T_m = \Delta s\left(T_c \cos\alpha - \left(n_w \cos\alpha + w\right) r \sin\alpha\right). \tag{3.11}$$

Taking the limit as Δs goes to zero,

$$\frac{dT_m}{ds} = T_c \cos\alpha - \left(n_w \cos\alpha + w\right) r \sin\alpha \,. \tag{3.12}$$

Equations (3.1), (3.2), (3.9), and (3.12) are the coupled ODEs that are solved to find the shape of the inflated bag. This shape depends upon the initial values $r(0)$, $h(0)$, $\alpha(0)$, and $T_m(0)$, respectively the radius of the upper bulkhead, the height of the top of the bag above the seabed, the angle at which the meridian leaves the upper bulkhead, and the meridional tension at the top of the meridian (per unit circumferential angle).

3.2 Solving the Ordinary Differential Equations

The shape of the bag's cross-section is found by solving equations (3.1), (3.2), (3.9), and (3.12). This is carried out numerically, as an initial value problem or as a boundary value problem if the base radius of the bag is to be prescribed. $r(0)$, $h(0)$, $\alpha(0)$, and $T_m(0)$ must be set before using the solver. The radius at the top of the bag ($r(0)$) just depends upon the radius of the upper bulkhead. The height of the top of the bag above the seabed ($h(0)$) and the meridional tension at the top of the bag ($T_m(0)$) must be varied by the user (or the root finder used for the shooting method – described later) until the base radius/meridional length/bag volume are as desired. The angle at which the meridians leave the upper bulkhead ($\alpha(0)$) depends upon the mass and radius of the bulkhead (m_{bh} and $r(0)$ respectively). The difference between the weight of the bulkhead (w_{bh}) and the differential pressure force acting on the bulkhead (F_{bh}) is

$$w_{bh} - F_{bh} = m_{bh} g - p\pi r(0)^2 \,. \tag{3.13}$$

This must be balanced by tension in the meridians, so the meridians must leave the bulkhead at an angle

$$\alpha(0) = -\sin^{-1}\left(\frac{m_{bh}g - p\pi r(0)^2}{2\pi T_m(0)}\right). \tag{3.14}$$

The differential pressure at height *h* above the base of the bag is given by

$$p = p_0 + (\rho_w - \rho_a)gh\,, \tag{3.15}$$

where p_0 is the differential pressure at the base of the bag. The density of seawater (ρ_w) is approximately 1,025kg/m^3, and the density of compressed air at absolute pressure *P* can be calculated using the equation of state for an ideal gas,

$$\rho = \frac{MP}{RT}\,, \tag{3.16}$$

where the molar mass of air, M_{air} = 0.02897kg/mol, the universal gas constant, *R* = 8.314472J/(K.mol), and *T* is the absolute temperature of the air. In this thesis it is assumed that the air will be stored at the same temperature as the surrounding seawater (approximately 5°C), with all heat generated in the compression being taken out of the air before it enters the bag and ideally stored in thermal energy stores. The absolute pressure of the compressed air is given by

$$P = p_0 + \rho_w gd + P_{atm}\,, \tag{3.17}$$

where *d* is the depth of the base of the bag and P_{atm} is atmospheric pressure (101.325kPa). Interestingly, the net buoyancy per unit volume of contained air (given by $(\rho_w - \rho_a)g$) scales linearly with the depth at which the bag is anchored, net buoyancy *reducing* as depth increases because of the much higher compressibility of air than water.

The differential equations presented above have been derived by starting from the top of the bag, but it is also possible to start from the bottom [95],[115]. Both approaches involve some difficulties: if starting from the top, it is necessary to use a shooting method in order to set a meridional length and base radius, and if starting from the bottom, a shooting method is required in order to set a meridional length, upper bulkhead mass, and upper bulkhead radius. We chose to start from the top of the bag because there are fewer variables that need to be set using the shooting method, for which Matlab's in-built root finder was used.

3.3 Use of the Model

Figure 3.2 shows the natural shape (so purely meridional loading) that a prototype Energy Bag with meridional length L = 2.36m will take when the base is anchored in water at a depth of 2.4m and subjected to a nominal overpressure at the base of p_0 = 100kPa (1bar). Note that the effect of depth on the shape of an Energy Bag with a given p_0 is small and is only a result of changes to the density of the contained air.

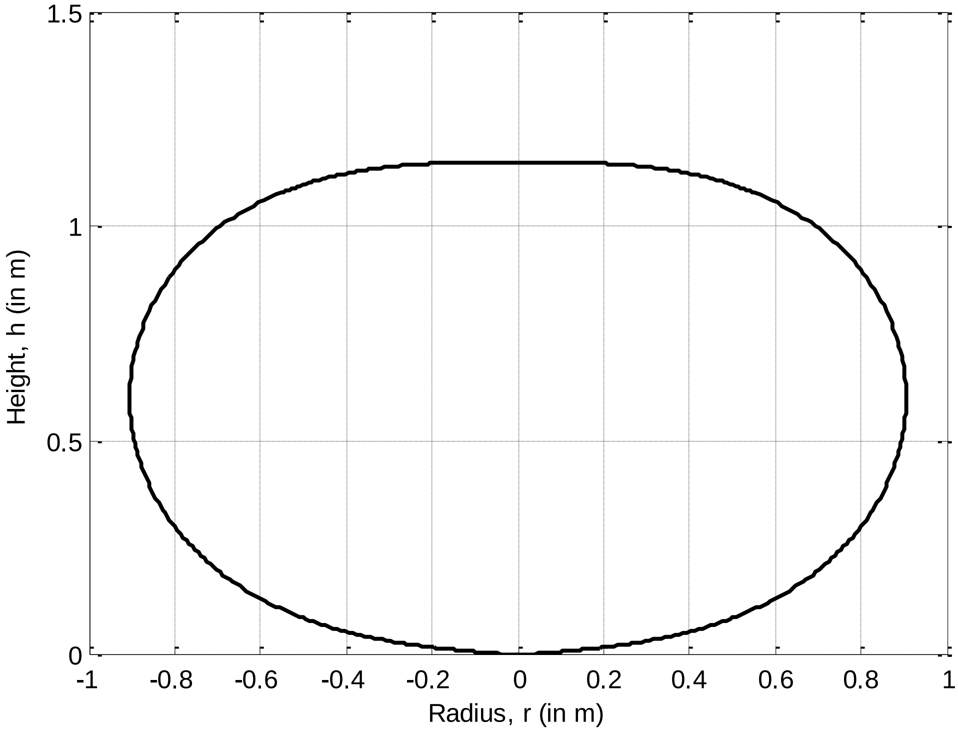


Fig. 3.2 Prototype bag with L = 2.36m and p_0 = 100kPa

Clearly a highly pressurised natural shape Energy Bag has the pumpkin shape of a superpressure balloon. The greater pressure gradient of water than atmospheric air means that the buoyancy of an Energy Bag is greater than that of a balloon with the same volume, but it has no effect on the shape because of the lack of bending stiffness in both balloons and Energy Bags.

Introducing a nominal circumferential tension of T_c = 20,000N per metre of meridian raises the centre of the bag, lowering the meridional curvature. The meridional curvature is reduced because some of the pressure load is being carried circumferentially, lowering the meridional stress. This bag is shown in figure 3.3, with the bag with T_c = 0N/m (from figure 3.2) shown with a dashed line for comparison.

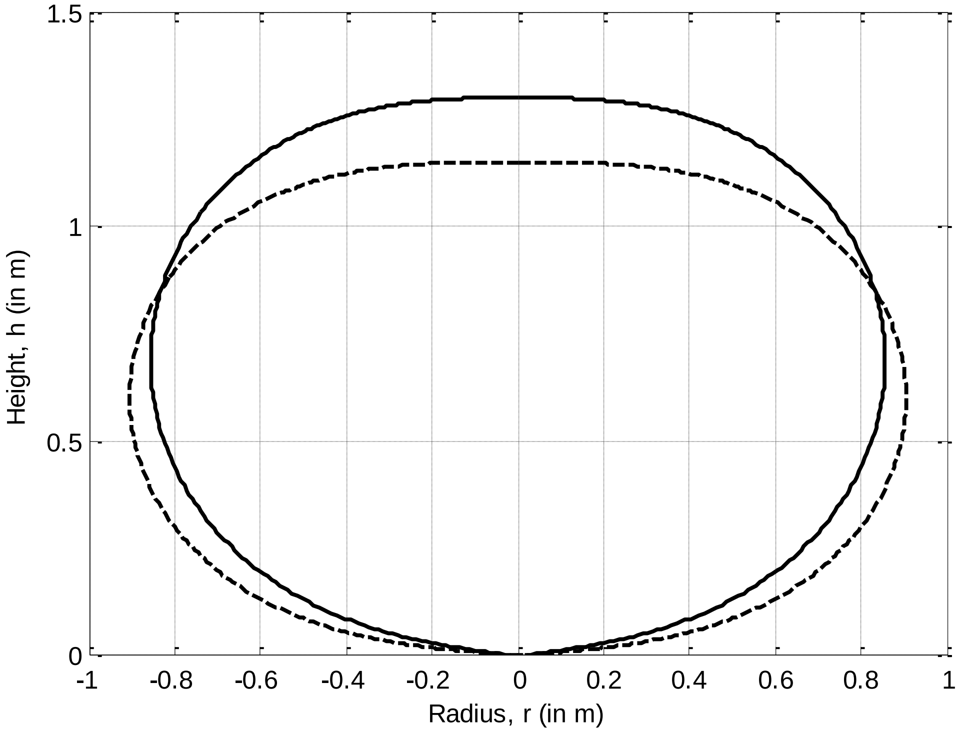


Fig. 3.3 Solid line – T_c = 20,000N/m; Dashed line – T_c = 0N/m

Bags with wide bases – $r(L) > 0$ – can also be modelled. A natural shape bag with base radius of just over 5m is shown in figure 3.4. It stores 289m^3 of air, and is anchored at a depth of 500m. The differential pressure at the base of the bag is zero, and so the absolute pressure of the contained air is 51.28bar, equating to 1.62MWh of energy (assuming isothermal expansion at 5°C). Being a natural shape bag with zero differential pressure at the base, this type of bag is known as a zero pressure natural shape (ZPNS) bag.

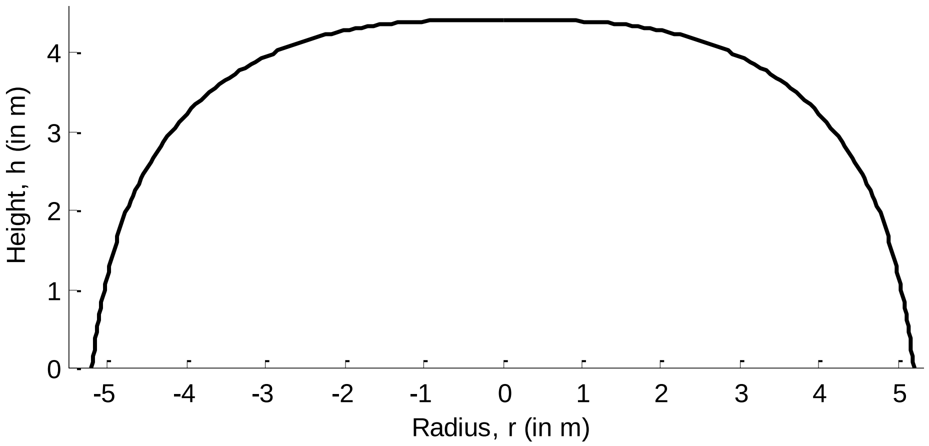
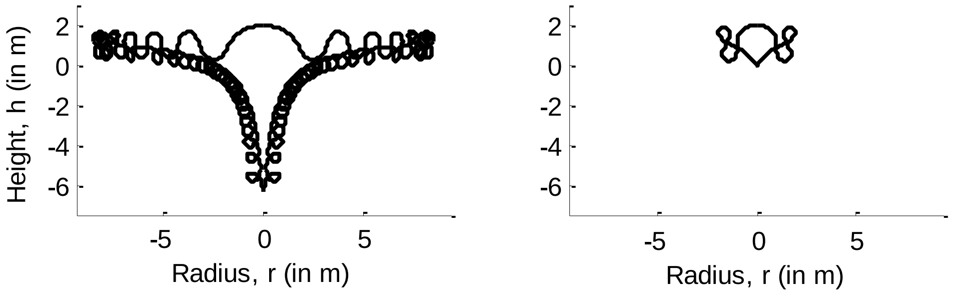


Fig. 3.4 ZPNS bag with nonzero base radius

This approach is later used to find the size of ZPNS Energy Bag which minimises the cost of materials per unit of energy stored, and to look at the effects of circumferential loading, hanging masses and nonzero differential pressure at the base, as also presented in [116]. However it is quite hard to use this approach as a root finder must be used to set L and $r(L)$, and looping can occur, especially when trying to find the shapes of highly deflated bags. By way of example, figure 3.5 shows two failed attempts to find the shape of a bag with $r(L)$ = 0m, $h(0)$ = 2m, and $p_0 = -\rho g h(0)/2$ (so the level of zero differential pressure would be halfway up the bag). In figure 3.5(a), the initial guess at $T_m(0)$ was 10,000N/rad. The resulting solution shows substantial looping, and the cross-section of the bag meets the centreline beneath the seabed. In figure 3.5(b), $T_m(0)$ was initially set to be equal to the tension in the meridian of a sphere[3] with volume of V_{sp}, given by $V_{\mathrm{sp}}b/2\pi$, where V_{sp} was chosen to be 1.6m^3 in this case and $b = g(\rho_w - \rho_a)$ (ρ_w is the density of the water and ρ_a is the density of the contained air). This makes sense because the net buoyancy force in an object with volume V is given by Vb, and it is necessary to divide this force by 2π in order to obtain the meridional tension per unit circumferential angle. The looping in 3.5(b) is much less dramatic than in 3.5(a) because the initial guess at $T_m(0)$ was more accurate. Figure 3.6 shows the realistic solution, obtained with an initial guess at $T_m(0)$ of 1,300N/rad. In this case the actual value of $T_m(0)$ is 1,358N/rad, lower than both the guesses in figure 3.5. However, using a very small guess at $T_{\mathrm{m}}(0)$ does not help; guesses at $T_{\mathrm{m}}(0)$ of 2N/rad and 1,000N/rad both give the same result as the initial guess of 2,404N/rad (as shown in figure 3.5(b)).

[3] It should be noted that an actual sphere is not possible if the anchoring is at the very bottom of the bag, because the vertical component of tension in the base of the bag would be zero, so the anchor would be providing no reaction to the buoyancy force.



(a) (b)

Fig. 3.5 Attempts to find the shape of a bag with zero base radius, 2m height, and $p_0 = -\rho g h(0)/2$, with initial guess at the meridional tension in (a) of $T_m(0)$ = 10,000N/rad, and initial guess in (b) of $T_m(0)$ = 2,404N/rad

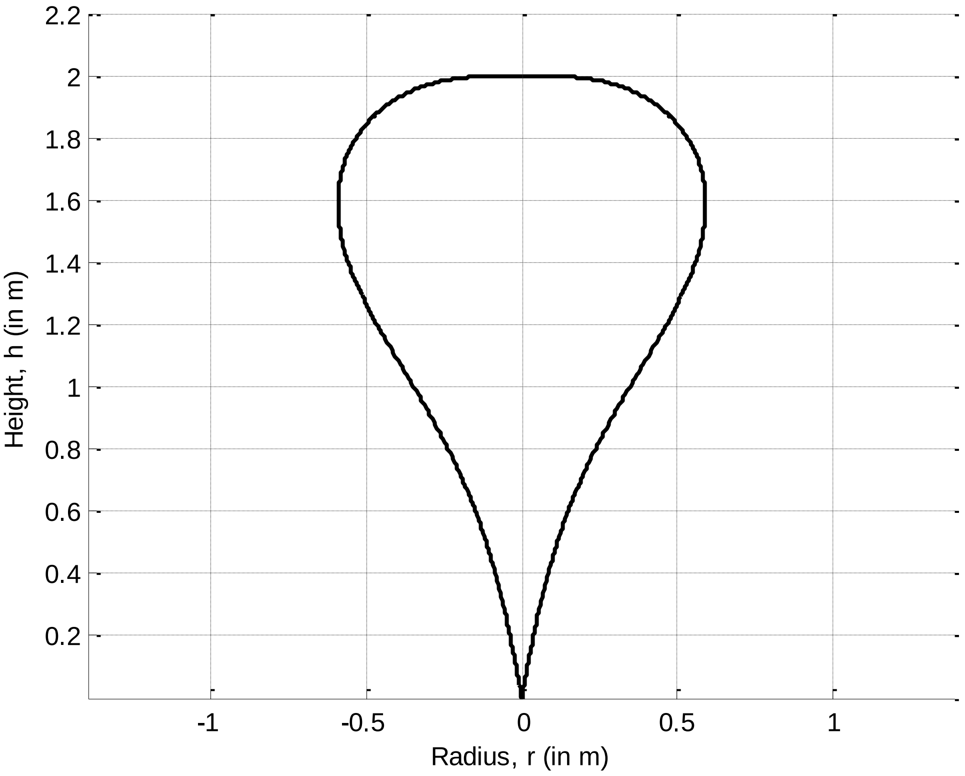


Fig. 3.6 The realistic solution is obtained if the initial guess at $T_\mathrm{m}(0)$ is 1,300N/rad. Actual value of $T_\mathrm{m}(0)$ is 1,358N/rad

In summary, the axisymmetric ODE method is not very easy to use because of the requirement to use a shooting method if we want to set one or both of the meridional length or base radius, and because of the looping problems. An accurate initial guess at the meridional tension at the top of the bag, $T_{\mathrm{m}}(0)$, is crucial to avoiding looping in the solution, and it is not clear if there is a better method of making an accurate initial guess at $T_m(0)$ than that used in figure 3.5(b) without resorting to trial and error. Penalty functions were used to keep clear of looping in an optimisation study carried out using this method, presented in Chapter 7. It was decided that a more user-friendly and robust method of analysing axisymmetric bag shapes was required, and finite element analysis was employed for this purpose.

Chapter 4

Finite Element Analysis of Axisymmetric Structures

As realistic solutions to the ordinary differential equations presented in Chapter 3 are not easy to find when trying to study the shapes of highly deflated bags (which have $p_0 < 0$), and because the method is generally hard to work with, a different method of finding the deformed shapes of Energy Bags is required. Finite element analysis (FEA) is used, and we begin with FEA of axisymmetric structures. A single tendon in a bag of *n* tendons is modelled using cable elements, one-dimensional elements with no bending stiffness between them (though bending stiffness may be introduced). Special consideration must be given to the stiffness of a cable undergoing compression, as cables have very little stiffness in compression. No axisymmetric FEA of inflatable structures (balloons, lifting bags, etc.) was found in the literature, so it is believed that the work in this chapter is original.

4.1 The Cable Element

A cable element is a simple type of finite element used to model cables that comprises of two nodes, one at each end of the element. Because cables have very little bending stiffness they can undergo large displacements, and so the force vector and stiffness matrix of a cable element are geometrically nonlinear. Bending stiffness can be introduced between cable elements if desired, and this is shown in this chapter. In the axisymmetric FEA presented here, the elements are confined to a single plane and so only have 4 degrees of freedom, but cable elements are used in the 3D FE in Chapter 6 so we just give the derivation for an element in three-dimensional space here (so with 6 degrees of freedom), and do not need to include it in Chapter 6. It is fairly straightforward to modify the force vector and stiffness matrix of a 6-dof element in order to obtain the force vector and stiffness matrix of a 4-dof element confined to a single plane.

4.1.1 Force Vector

The undeformed and deformed configurations of a cable element are shown in figure 4.1, the nodes of the element in the deformed configuration having been displaced by u_1,v_1,w_1 and u_2,v_2,w_2. As mentioned above, though in the axisymmetric model each 2-node cable element has only 4 degrees of freedom, what follows is the more general derivation of the force vector and stiffness matrix of a 2-node cable element with 6 degrees of freedom. This can be reduced for a cable element with 4 degrees of freedom by setting z_1, z_2, w_1, and w_2 to zero and removing the corresponding rows of the strain vector ε and the unit parallel vector $\hat{p}$, and the corresponding rows and columns of the stiffness matrix K.

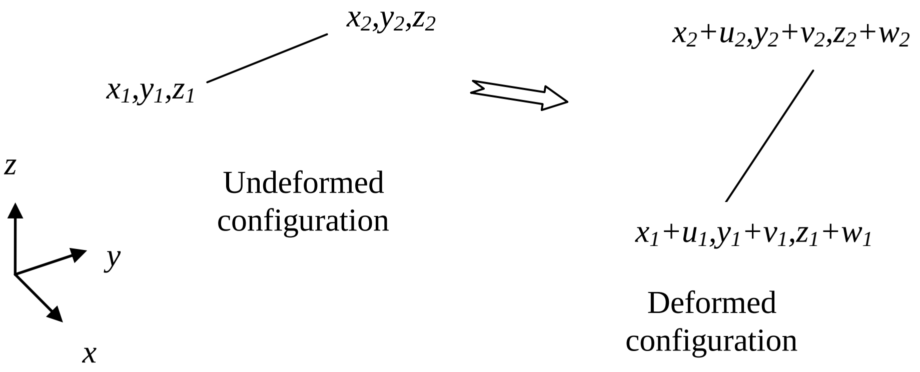


Fig. 4.1 Undeformed and deformed configurations of a cable element

The strains in a cable element are uniaxial and uniform, and the engineering strain is

$$\varepsilon = \frac{L - L_0}{L_0} \tag{4.1}$$

where L, the deformed length of the element, and L_0, the undeformed length, are given by

$$L = \sqrt{(x_{21}+u_{21})^2 + (y_{21}+v_{21})^2 + (z_{21}+w_{21})^2} \text{ and } L_0 = \sqrt{{x_{21}}^2 + {y_{21}}^2 + {z_{21}}^2}\ . \tag{4.2}$$

Note that, for example, $x_{21} = x_2 - x_1$. The stress and tension in an element are found using the Young's modulus of the cable material, E, and the cross-sectional area of the cable, A.

$$\sigma = E\varepsilon + \sigma_0 \text{ and so } T = A(E\varepsilon + \sigma_0) \tag{4.3}$$

where σ_0 is the prestress in the element.

To find the element internal force vector F it is necessary to multiply T by the unit parallel to the deformed element, which can be expressed as

$$\hat{p} = \frac{1}{L}\begin{bmatrix} -a_x & -a_y & -a_z & a_x & a_y & a_z \end{bmatrix}^T . \tag{4.4}$$

Note that a is the vector from node 1 to node 2, so $a_x = x_{21} + u_{21}$, $a_y = y_{21} + v_{21}$, and $a_z = z_{21} + w_{21}$. Hence the element internal force vector becomes

$$\begin{aligned} F &= A\sigma\hat{p} \\ &= A\left(E\varepsilon + \sigma_0\right)\hat{p} . \end{aligned} \tag{4.5}$$

4.1.2 Stiffness Matrix

The element stiffness matrix, K, is obtained by differentiating the force vector with respect to the nodal displacements, u, and can be broken down into the sum of two matrices, the elastic stiffness matrix, K_e, and the geometric stiffness matrix, K_g.

$$\begin{aligned} K &= \frac{\partial F}{\partial u} \\ &= EA\hat{p}\frac{\partial \varepsilon}{\partial u} + A\sigma\frac{\partial \hat{p}}{\partial u} \\ &= K_e + K_g \end{aligned} \tag{4.6}$$

Differentiating the strain gives

$$\frac{\partial \varepsilon}{\partial u} = \frac{1}{L_0}\frac{\partial L}{\partial u} \tag{4.7}$$

where

$$
\begin{aligned}
\frac{\partial L}{\partial u} &= \frac{1}{L}\begin{bmatrix} -a_x & -a_y & -a_z & a_x & a_y & a_z \end{bmatrix} \\
&= \hat{p}^T \,.
\end{aligned}
\tag{4.8}
$$

Differentiating the unit parallel gives

$$
\frac{\partial \hat{p}}{\partial u} = \frac{1}{L^3}\begin{bmatrix}
a_y{}^2 + a_z{}^2 & -a_x a_y & -a_x a_z & -a_y{}^2 - a_z{}^2 & a_x a_y & a_x a_z \\
-a_y a_x & a_x{}^2 + a_z{}^2 & -a_y a_z & a_y a_x & -a_x{}^2 - a_z{}^2 & a_y a_z \\
-a_z a_x & -a_z a_y & a_x{}^2 + a_y{}^2 & a_z a_x & a_z a_y & -a_x{}^2 - a_y{}^2 \\
-a_y{}^2 - a_z{}^2 & a_x a_y & a_x a_z & a_y{}^2 + a_z{}^2 & -a_x a_y & -a_x a_z \\
a_y a_x & -a_x{}^2 - a_z{}^2 & a_y a_z & -a_y a_x & a_x{}^2 + a_z{}^2 & -a_y a_z \\
a_z a_x & a_z a_y & -a_x{}^2 - a_y{}^2 & -a_z a_x & -a_z a_y & a_x{}^2 + a_y{}^2
\end{bmatrix} .
\tag{4.9}
$$

So the elastic and geometric stiffness matrices can be written as

$$
K_e = \frac{EA}{L^2 L_0}\begin{bmatrix}
a_x{}^2 & a_x a_y & a_x a_z & -a_x{}^2 & -a_x a_y & -a_x a_z \\
a_y a_x & a_y{}^2 & a_y a_z & -a_y a_x & -a_y{}^2 & -a_y a_z \\
a_z a_x & a_z a_y & a_z{}^2 & -a_z a_x & -a_z a_y & -a_z{}^2 \\
-a_x{}^2 & -a_x a_y & -a_x a_z & a_x{}^2 & a_x a_y & a_x a_z \\
-a_y a_x & -a_y{}^2 & -a_y a_z & a_y a_x & a_y{}^2 & a_y a_z \\
-a_z a_x & -a_z a_y & -a_z{}^2 & a_z a_x & a_z a_y & a_z{}^2
\end{bmatrix} ,
\tag{4.10}
$$

$$
K_g = \frac{A\sigma}{L^3}\begin{bmatrix}
a_y{}^2 + a_z{}^2 & -a_x a_y & -a_x a_z & -a_y{}^2 - a_z{}^2 & a_x a_y & a_x a_z \\
-a_y a_x & a_x{}^2 + a_z{}^2 & -a_y a_z & a_y a_x & -a_x{}^2 - a_z{}^2 & a_y a_z \\
-a_z a_x & -a_z a_y & a_x{}^2 + a_y{}^2 & a_z a_x & a_z a_y & -a_x{}^2 - a_y{}^2 \\
-a_y{}^2 - a_z{}^2 & a_x a_y & a_x a_z & a_y{}^2 + a_z{}^2 & -a_x a_y & -a_x a_z \\
a_y a_x & -a_x{}^2 - a_z{}^2 & a_y a_z & -a_y a_x & a_x{}^2 + a_z{}^2 & -a_y a_z \\
a_z a_x & a_z a_y & -a_x{}^2 - a_y{}^2 & -a_z a_x & -a_z a_y & a_x{}^2 + a_y{}^2
\end{bmatrix} .
\tag{4.11}
$$

Note that both of these matrices are symmetric and nonlinear, and so the element stiffness matrix will also be symmetric and nonlinear. In the Matlab code, the accuracy of the stiffness matrices (including circumferential stiffness and load stiffness, described later) have been checked by comparing them with the results of numerical differentiation of the vector of meridional force/circumferential force/load.

4.1.3 Wrinkling

As cables have little stiffness in compression, the stiffness of a cable element in compression ($\sigma < 0$) would ideally be set to zero, however this can be problematic because of the discontinuity introduced into the stress-strain curve. To ensure that the first derivative of the stress-strain curve is continuous at $\sigma = 0$, the arctan function is used to continuously reduce the stress in a wrinkling element towards a very small negative value. The derivation of the modified compressive stress (σ_c) in an element undergoing compression follows. It should be noted that the derived equation for modified compressive stress will only be used when $\sigma < 0$, and the normal stress-strain relationship ($\sigma = E\varepsilon + \sigma_0$) will be used in the tensile (unwrinkled) region.

We begin with the equation

$$\sigma_c = \tan^{-1}\left(b\left(\varepsilon + \frac{\sigma_0}{E}\right)\right). \tag{4.12}$$

If b is a very large number then as ε approaches -1, σ_c will approach $-\pi/2$ (the asymptote). This asymptote can be modified to a very small fraction (λ) of the Young's modulus of the cable by premultiplying by a, given by

$$a = \frac{2\lambda E}{\pi}, \tag{4.13}$$

where λ is a very small number. Therefore we now have

$$\sigma_c = a\tan^{-1}\left(b\left(\varepsilon + \frac{\sigma_0}{E}\right)\right). \tag{4.14}$$

It is good to ensure that the gradient of the stress-strain curve is continuous at all points. Therefore b needs to be chosen so that the gradient is continuous at the transition between the tensile and compressive regions (i.e. at $\sigma = 0$). Differentiating equation (4.14) with respect to strain,

$$\frac{d\sigma_c}{d\varepsilon} = \frac{ab}{1 + b^2\left(\varepsilon + \frac{\sigma_0}{E}\right)^2} . \tag{4.15}$$

$\sigma = 0$ at the transition between the tensile and compressive regions, and so $\varepsilon + \sigma_0/E = 0$ at this point. The gradient of the stress-strain curve needs to equal E at the transition, so setting equation (4.15) equal to E and letting $\varepsilon + \sigma_0/E = 0$,

$$ab = E . \tag{4.16}$$

Rearranging for b,

$$b = \frac{E}{a} . \tag{4.17}$$

At the beginning of this section it was highlighted that b should be a very large number, so we should now check that it is. Substituting equation (4.13) into (4.17) we obtain

$$b = \frac{\pi}{2\lambda} . \tag{4.18}$$

Since λ is a very small number then b will be a very large number.

To summarise, the stress in an element undergoing compression is set to

$$\sigma_c = a \tan^{-1}\left(b\left(\varepsilon + \frac{\sigma_0}{E}\right)\right) \tag{4.19}$$

where

$$a = \frac{2\lambda E}{\pi} \text{ and } b = \frac{E}{a} \tag{4.20}$$

and λ is a very small positive value (e.g. 1×10^{-6}). This means that if $\sigma_0 = 0$, $\sigma_c \approx -\lambda E$ when $\varepsilon = -1$ (-1 being the smallest value that engineering strain can take). Note that if $\sigma_0 \neq 0$, the stress-strain curve will still be continuous at $\sigma = 0$, and if $\sigma_0 > E$, the element will not go into compression even if $\varepsilon = -1$. Figure 4.2 shows the stress-strain relationship for a cable with

E = 200GPa and σ_0 = 0 at the transition between tension and compression, with $\lambda = 1\times10^{-6}$. In this case, $\sigma_c \approx -2\times10^{5}$Pa when $\varepsilon = -1$, meaning that a cable of diameter 20mm will support a maximum compressive force of 62.8N, equal to the tensile force when $\varepsilon = 0.001$.

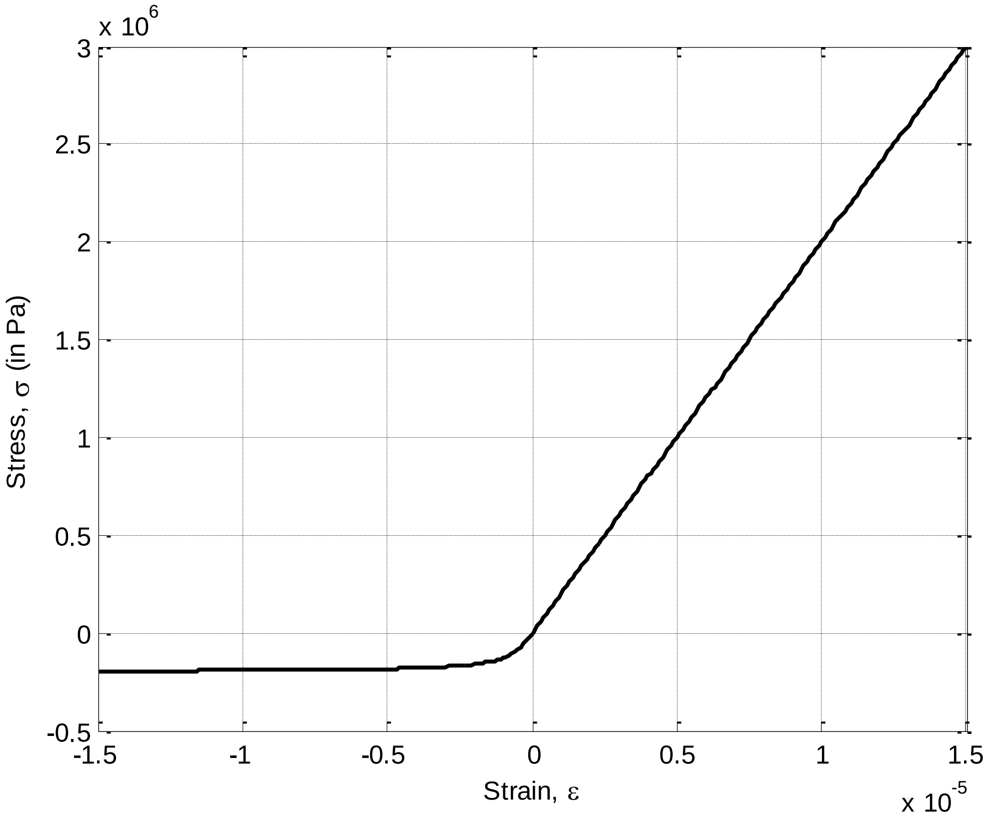


Fig. 4.2 Plot of stress against strain with E = 200GPa and σ_0 = 0

Differentiating σ_c with respect to nodal displacements and premultiplying by $A\hat{p}$ gives the elastic stiffness of an element in compression.

$$K_{e,c} = \frac{EA}{L^2 L_0\left(1+\left(\frac{\pi}{2\lambda}\left(\varepsilon+\frac{\sigma_0}{E}\right)\right)^2\right)}\begin{bmatrix} a_x{}^2 & a_x a_y & a_x a_z & -a_x{}^2 & -a_x a_y & -a_x a_z \\ a_y a_x & a_y{}^2 & a_y a_z & -a_y a_x & -a_y{}^2 & -a_y a_z \\ a_z a_x & a_z a_y & a_z{}^2 & -a_z a_x & -a_z a_y & -a_z{}^2 \\ -a_x{}^2 & -a_x a_y & -a_x a_z & a_x{}^2 & a_x a_y & a_x a_z \\ -a_y a_x & -a_y{}^2 & -a_y a_z & a_y a_x & a_y{}^2 & a_y a_z \\ -a_z a_x & -a_z a_y & -a_z{}^2 & a_z a_x & a_z a_y & a_z{}^2 \end{bmatrix}$$

(4.21)

The geometric stiffness of an element in compression need only be changed by replacing σ with σ_c, and the symmetry of the element stiffness matrix remains.

It should be noted that wrinkling elements are not generally found in commercially-available FEA packages.

4.2 Circumferential Reinforcement

It is fairly straightforward to include circumferential reinforcement at each node on the meridian, and it is necessary to do so in order to model anything other than natural shape bags. Each circumferential element is treated as a single cable that wraps round the full perimeter of the bag. The force vector and stiffness matrix for circumferential elements are derived here.

4.2.1 Force Vector

A cross-section of the bag (so looking along the axis of axisymmetry) is shown in figure 4.3.

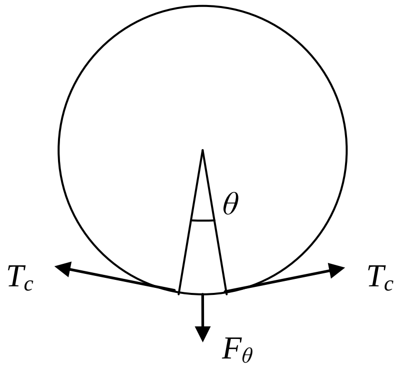


Fig. 4.3 Cross-section of the bag looking along the axis of axisymmetry

Looking at a small section of the circumference which covers an angle θ, the force required to react the circumferential tension is given by

$$F_\theta = 2T_c \sin\left(\frac{\theta}{2}\right). \tag{4.22}$$

Over the full circumference,

$$\begin{aligned} F_{2\pi} &= \frac{2\pi}{\theta} F_\theta \\ &= \frac{4\pi}{\theta} T_c \sin\left(\frac{\theta}{2}\right). \end{aligned} \tag{4.23}$$

Taking small angle approximations,

$$\begin{aligned} F_{2\pi} &= \frac{4\pi}{\theta} T_c \frac{\theta}{2} \\ &= 2\pi T_c . \end{aligned} \tag{4.24}$$

So the vector of force due to circumferential tension at the current node is

$$F = \begin{bmatrix} 2\pi T_c \\ 0 \end{bmatrix} \tag{4.25}$$

where the circumferential tension is given by

$$\begin{aligned} T_c &= A_c \sigma_c \\ &= A_c \left(E_c \varepsilon_c + \sigma_{c,0} \right). \end{aligned} \tag{4.26}$$

A_c and E_c are the cross-sectional area and Young's modulus of the circumferential cable, and σ_c and $\sigma_{c,0}$ are the stress and prestress in the cable. The circumferential strain is given by

$$\begin{aligned} \varepsilon_c &= \frac{L_c - L_{c,0}}{L_{c,0}} \\ &= \frac{2\pi(x_1 + u_1) - 2\pi x_1}{2\pi x_1} \\ &= \frac{u_1}{x_1} . \end{aligned} \tag{4.27}$$

L_c is the circumference of the deformed bag at the current node and $L_{c,0}$ is the circumference of the undeformed bag at the current node. So the force vector becomes

$$F = \begin{bmatrix} 2\pi A_c \left(E_c \dfrac{u_1}{x_1} + \sigma_{c,0} \right) \\ 0 \end{bmatrix}. \tag{4.28}$$

If $\sigma_c \le 0$, the cable must be treated as a wrinkling element. As with the meridional elements, the stress is smoothly reduced to a very small amount.

If $\sigma_c \le 0$,

$$\begin{aligned} \sigma_c &= a \tan^{-1}\left(b\left(\varepsilon_c + \frac{\sigma_{c,0}}{E_c} \right)\right) \\ &= a \tan^{-1}\left(b\left(\frac{u_1}{x_1} + \frac{\sigma_{c,0}}{E_c} \right)\right) \end{aligned} \tag{4.29}$$

where

$$a = \frac{2\lambda E_c}{\pi} \text{ and } b = \frac{E_c}{a}. \tag{4.30}$$

As before, λ is a very small value.

4.2.2 Stiffness Matrix

The stiffness matrix of a circumferential cable element is given by

$$K = \frac{\partial F}{\partial u} = 2\pi A_c \frac{E_c}{x_1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.31}$$

If $\sigma_c \le 0$,

$$K = 2\pi A_c \frac{ab}{1 + \left(b\left(\varepsilon_c + \dfrac{\sigma_{c,0}}{E_c} \right)\right)^2} \frac{1}{x_1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.32}$$

4.3 Bending

Although cables typically have very little bending stiffness and so by definition the bending stiffness in a cable element is zero, it is still useful to be able to give the cables some bending stiffness. There are two main reasons for doing this: to improve the accuracy of the model, and to be able to treat the bag as a shell so that the deformed shape will always be very close to the undeformed shape. We are really interested in the second of these two reasons, because it is necessary to ensure that an initial solution can be found, and the closer the initial solution is to the starting configuration, the higher the chances of finding the initial solution. The bending stiffness can then be gradually reduced to zero, updating the deformed shape with each step.

4.3.1 Force Vector

The bending moment *M* is proportional to the change in angle between two elements, and is given by

$$M = k(\theta - \alpha) \tag{4.33}$$

where *k* is the bending stiffness of the cable, θ is the angle between the two elements in the deformed configuration, and α is the angle between the two elements in the undeformed configuration. The bending moment is also the product of the bending force and the distance between the force and the joint,

$$M = F_b L\,. \tag{4.34}$$

We set these two equations to be equal and rearrange to obtain the bending force,

$$F_b = \frac{k(\theta - \alpha)}{L}\,. \tag{4.35}$$

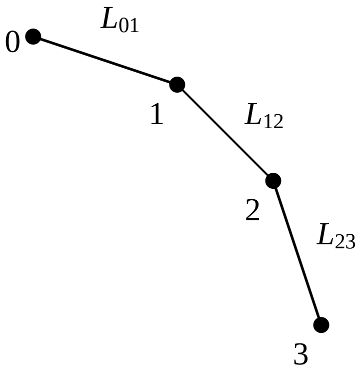


Fig. 4.4 Deformed configurations of three elements with bending stiffness between them

The element bending force vector is given by

$$F_b = \frac{1}{L}\begin{bmatrix} \hat{n}k_2(\theta_2 - \alpha_2) \\ \hat{n}k_1(\theta_1 - \alpha_1) \end{bmatrix}. \tag{4.36}$$

where

$$\theta_1 = \cos^{-1}\left(\frac{(x_1+u_1)-(x_0+u_0)}{L_{01}}\right)+\pi-\cos^{-1}\left(\frac{(x_2+u_2)-(x_1+u_1)}{L_{12}}\right), \tag{4.37}$$

$$\theta_2 = \cos^{-1}\left(\frac{(x_2+u_2)-(x_1+u_1)}{L_{12}}\right)+\pi-\cos^{-1}\left(\frac{(x_3+u_3)-(x_2+u_2)}{L_{23}}\right), \tag{4.38}$$

and

$$\alpha_1 = \cos^{-1}\left(\frac{x_1-x_0}{L_{01,0}}\right)+\pi-\cos^{-1}\left(\frac{x_2-x_1}{L_{12,0}}\right), \tag{4.39}$$

$$\alpha_2 = \cos^{-1}\left(\frac{x_2-x_1}{L_{12,0}}\right)+\pi-\cos^{-1}\left(\frac{x_3-x_2}{L_{23,0}}\right). \tag{4.40}$$

$\hat{n}$ is the unit normal to the element, given later in equation (4.45). The subscript 1 indicates that the angle is on the side of the element closest to the top of the bag (e.g. for element 12 in figure 4.4, this is the angle between 01 and 12), and the subscript 2 indicates that the angle is on the side of the element closest to the bottom of the bag (e.g. for element 12 this is the angle between 12 and 23). $L_{01,0}$ is the undeformed length of element 01, and so on. The bending force vector must be added to the force vector.

4.3.2 Stiffness Matrix

The element bending stiffness matrix must take into account the changes in bending force at each end of the element with respect to the displacements of both nodes on the element and all neighbouring nodes. In this axisymmetric model, the element bending stiffness matrix for the elements at the top and bottom of the bag has 6 columns (because these elements have only one neighbouring node, e.g. node 2 is the only neighbouring node to element 01 in figure 4.4), but has 8 columns for all the other elements in between (because internal elements have two neighbouring nodes – nodes 0 and 3 for element 12 in figure 4.4). Assuming that k_1 and k_2 are independent of θ_1 and θ_2,

$$\frac{\partial F_b}{\partial u} = \frac{1}{L}\begin{bmatrix} k_2(\theta_2-\alpha_2)\dfrac{\partial \hat{n}}{\partial u} \\ k_1(\theta_1-\alpha_1)\dfrac{\partial \hat{n}}{\partial u}\end{bmatrix} + \begin{bmatrix}\hat{n}k_2(\theta_2-\alpha_2)\\ \hat{n}k_1(\theta_1-\alpha_1)\end{bmatrix}\left(-\frac{1}{L^2}\right)\frac{\partial L}{\partial u} + \frac{1}{L}\begin{bmatrix}\hat{n}k_2\dfrac{\partial\theta_2}{\partial u}\\ \hat{n}k_1\dfrac{\partial \theta_1}{\partial u}\end{bmatrix}, \tag{4.41}$$

where, for all elements other than the top and bottom elements,

$$\begin{aligned}\frac{\partial\theta_1}{\partial u} = {} & \frac{-1}{\sqrt{1-\left(\dfrac{(x_1+u_1)-(x_0+u_0)}{L_{01}}\right)^2}}\left(((x_1+u_1)-(x_0+u_0))\left(-\frac{1}{{L_{01}}^2}\right)\frac{\partial L_{01}}{\partial u}+\frac{1}{L_{01}}\begin{bmatrix}-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\right) \\ & + \frac{1}{\sqrt{1-\left(\dfrac{(x_2+u_2)-(x_1+u_1)}{L_{12}}\right)^2}}\left(((x_2+u_2)-(x_1+u_1))\left(-\frac{1}{{L_{12}}^2}\right)\frac{\partial L_{12}}{\partial u}+\frac{1}{L_{12}}\begin{bmatrix}0 & 0 & -1 & 0 & 1 & 0 & 0 & 0\end{bmatrix}\right),\end{aligned} \tag{4.42}$$

$$\begin{aligned}\frac{\partial\theta_2}{\partial u} = {} & \frac{-1}{\sqrt{1-\left(\dfrac{(x_2+u_2)-(x_1+u_1)}{L_{12}}\right)^2}}\left(((x_2+u_2)-(x_1+u_1))\left(-\frac{1}{{L_{12}}^2}\right)\frac{\partial L_{12}}{\partial u}+\frac{1}{L_{12}}\begin{bmatrix}0 & 0 & -1 & 0 & 1 & 0 & 0 & 0\end{bmatrix}\right) \\ & + \frac{1}{\sqrt{1-\left(\dfrac{(x_3+u_3)-(x_2+u_2)}{L_{23}}\right)^2}}\left(((x_3+u_3)-(x_2+u_2))\left(-\frac{1}{{L_{23}}^2}\right)\frac{\partial L_{23}}{\partial u}+\frac{1}{L_{23}}\begin{bmatrix}0 & 0 & 0 & 0 & -1 & 0 & 1 & 0\end{bmatrix}\right).\end{aligned} \tag{4.43}$$

θ_1, α_1, and $\partial\theta_1/\partial u$ for the first element and θ_2, α_2, and $\partial\theta_2/\partial u$ for the last element must be given special consideration, as must $\partial\hat{n}/\partial u$ and $\partial L/\partial u$. The bending stiffness matrix must be added to the stiffness matrix.

4.4 Loading

We calculate the magnitude of the differential pressure load on each element in an axisymmetric bag by multiplying the differential pressure at the centre of the element by the swept area of the element, ensuring that we take the number of meridional reinforcing cables (henceforth known as tendons) *n* into account by only sweeping through 2π/*n* radians. The differential pressure load is multiplied by the unit normal to the element (in the plane defined by the element and axis of axisymmetry) to obtain the differential pressure force vector at each node on the element. The element load vector is formed from the sum of the element differential pressure force vector and the element mass vector.

4.4.1 Load Vector

Figure 4.5 shows the coordinate system used in this chapter; the axis of axisymmetry is aligned with gravity (so perpendicular to a flat seabed), the origin of the global coordinate system is located at the centre of the base of the bag, *x* and *u* are in the radial direction, and *y* and *v* are aligned with the axis of axisymmetry.

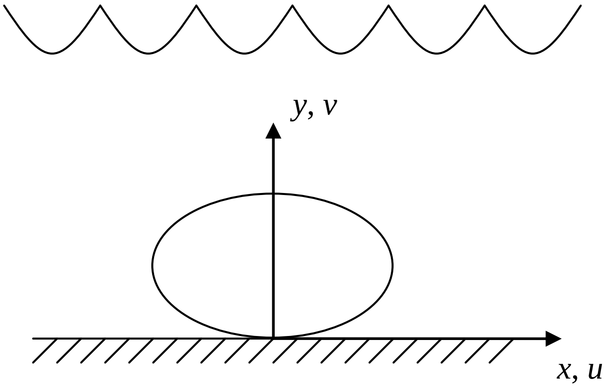


Fig. 4.5 Coordinate system for an axisymmetric bag

The vector of differential pressure force on a single element in an axisymmetric bag of *n* tendons is the product of the differential pressure at the centre of the element, *p*, the swept area of the element, *A*, and the unit normal to the element, $\hat{n}$. The element differential pressure force vector is given by

$$F_{dp} = \frac{1}{2} pA\hat{n} \tag{4.44}$$

where the unit normal is

$$\hat{n} = \frac{1}{L}\begin{bmatrix} -a_y & a_x & -a_y & a_x \end{bmatrix}^T , \tag{4.45}$$

the swept area of a single tendon in a bag of *n* tendons is

$$\begin{aligned} A &= \frac{2\pi}{n}\frac{1}{2}(x_1 + u_1 + x_2 + u_2)L \\ &= \frac{\pi L}{n}(x_1 + u_1 + x_2 + u_2), \end{aligned} \tag{4.46}$$

and the differential pressure at the centre of the element is

$$p = p_0 + (\rho_w - \rho_a)g\frac{1}{2}(y_1 + v_1 + y_2 + v_2). \tag{4.47}$$

The density of seawater is approximately 1,025kg/m^3 and the density of the contained air is calculated using the equation of state for an ideal gas (as in equation (3.16)). So the element differential pressure vector may be written as

$$F_{dp} = \frac{p\pi}{2n}(x_1 + u_1 + x_2 + u_2)\begin{bmatrix} -a_y & a_x & -a_y & a_x \end{bmatrix}^T . \tag{4.48}$$

It is also possible to include the mass of the bag materials as the mass acts vertically downwards, so in the plane of the element and axis of axisymmetry. For an element with mass *m*, the element load vector is then calculated using

$$F = F_{dp} + mg[0 \quad -0.5 \quad 0 \quad -0.5]^T . \tag{4.49}$$

m must take into account the mass of the membrane and tendons, and can be calculated using

$$\begin{aligned} m &= 2\pi 0.5(x_1 + x_2)L_0 t\rho_m + L_0 A_t \rho_t \\ &= \pi(x_1 + x_2)L_0 t\rho_m + L_0 A_t \rho_t , \end{aligned} \tag{4.50}$$

where *t* and ρ_m are the thickness and density of the membrane, and A_t and ρ_t are the cross-sectional area and density of the tendon.

It is quite straightforward to include the mass of bulkheads, so much so that it need not be described here except to say that care should be taken to ensure that the weight of the bulkhead is divided by *n*, where *n* is the number of tendons, before being added to the load vector of the node on the edge of the bulkhead.

4.4.2 Load Stiffness Matrix

The element load stiffness matrix, *R*, is found by differentiating the load vector with respect to the nodal displacements.

$$\begin{aligned} R &= \frac{\partial F}{\partial u} \\ &= \frac{1}{2}\left(\hat{n}A\frac{\partial p}{\partial u} + \hat{n}p\frac{\partial A}{\partial u} + pA\frac{\partial \hat{n}}{\partial u} \right) \end{aligned} \tag{4.51}$$

Note that the mass of materials (the membrane, tendons, and bulkheads) are unchanged under deformation, and always point vertically downwards, so have no effect on the load stiffness matrix. Differentiating the differential pressure at the centre of the element and the swept area of the element gives

$$\frac{\partial p}{\partial u} = \frac{1}{2}(\rho_w - \rho_a)g[0 \quad 1 \quad 0 \quad 1], \tag{4.52}$$

$$\frac{\partial A}{\partial u} = \frac{\pi L}{n}[1 \quad 0 \quad 1 \quad 0] + \frac{\pi}{n}(x_1 + u_1 + x_2 + u_2)\frac{1}{L}[-a_x \quad -a_y \quad a_x \quad a_y]. \tag{4.53}$$

Differentiating the unit normal gives

$$\frac{\partial \hat{n}}{\partial u} = \frac{1}{L^2}\left(L\begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} -a_y & a_x & -a_y & a_x \end{bmatrix}^T \frac{1}{L}\begin{bmatrix} -a_x & -a_y & a_x & a_y \end{bmatrix} \right)$$

$$= \frac{1}{L^3}\begin{bmatrix} -a_x a_y & {a_x}^2 + {a_z}^2 & a_x a_y & -{a_x}^2 - {a_z}^2 \\ -{a_y}^2 - {a_z}^2 & a_x a_y & {a_y}^2 + {a_z}^2 & -a_x a_y \\ -a_x a_y & {a_x}^2 + {a_z}^2 & a_x a_y & -{a_x}^2 - {a_z}^2 \\ -{a_y}^2 - {a_z}^2 & a_x a_y & {a_y}^2 + {a_z}^2 & -a_x a_y \end{bmatrix}. \tag{4.54}$$

Like the element stiffness matrix, the element load stiffness matrix is nonlinear.

4.5 Solving the Equilibrium Equation

Due to the large displacements and rotations that an Energy Bag can undergo, the force equilibrium equation is geometrically nonlinear and is solved using the Newton-Raphson method. A new set of node positions which should be closer to the solution is calculated using

$$u_{i+1} = u_i - \left(K_i - R_i\right)^{-1}\left(f_{reac,i} - f_{load,i}\right). \tag{4.55}$$

Following the convention of section 2.4.1, the difference between the force vector (f_{reac}) and load vector (f_{load}) is the vector of force residuals

$$r = f_{reac} - f_{load}, \tag{4.56}$$

and the difference between the stiffness matrix (K) and the load stiffness matrix (R) is the Jacobian of the force residuals

$$J_r = K - R. \tag{4.57}$$

To ensure that convergence occurs, it is necessary to start with an initial configuration that is not too far from a solution. Finding the deformed shape of a fully inflated bag with circumferential reinforcement usually presents no problems because the shape of the fully inflated bag is often very similar to the undeformed configuration of the bag. (Note that “fully inflated” effectively means that the bag volume cannot be further increased without significantly straining the materials. With a sealed natural shape bag, fully inflated means a bag with a high overpressure at the base, so with $p_0 >> 0$.) However, trying to find the shapes of partially inflated bags, which can be quite different to that of the undeformed bag, can be problematic if trying to start with the undeformed bag shape and go straight to the target pressure. Trying to find the deformed shapes of bags with no circumferential reinforcement (i.e. natural shape bags) can also be hard because there is no obvious starting configuration.

To overcome these problems, in all circumstances we begin by finding the solution for an overpressurised bag while using the load incrementation procedure described in section 2.4.2. The load fraction (ϕ) is updated each time by searching for a load fraction which, after one iteration of the Newton-Raphson method, gives a maximum change in angle between all adjacent elements of just less than a certain value (e.g. 3°). We also start off with some prestress in the elements (because J_r is singular without prestress) and then remove the prestress after one iteration. Interestingly, the deformed shape found with a very small fraction of full load is very close to that found with full load. This can be seen in figure 4.6, but note that going from a load fraction of $\phi = 5.2429\times10^{-5}$ straight to full load ($\phi = 1$) does not work, even though the deformed shapes at these load fractions are almost identical. It is still necessary to step through the load fractions using the maximum change in angle method described above.

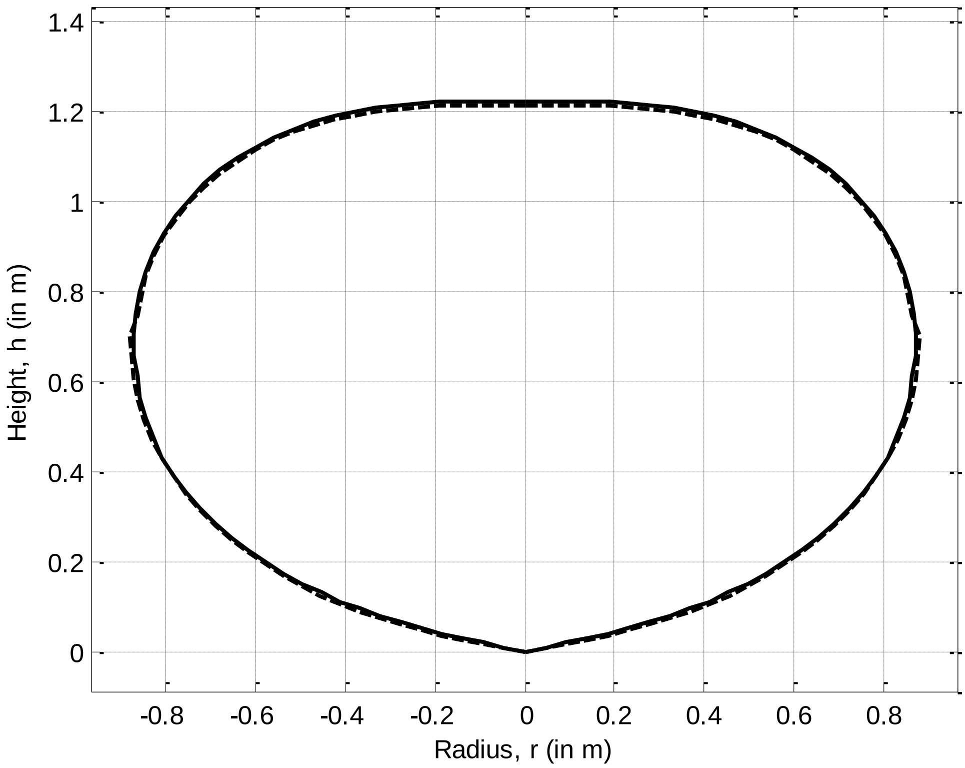


Fig. 4.6 Natural shape bag with p_0 = 0.2373bar. Dashed line: $\phi = 5.2429\times10^{-5}$. Solid line: $\phi = 1$

Once full load has been applied (i.e. once ϕ = 1), the internal pressure is then reduced to the target pressure in small steps, updating the deformed shape of the bag at each step. One approach to finding the deformed shapes of underpressurised bags without needing to begin with an overpressurised bag would be to include bending stiffness between the elements, so effectively modelling the bag as a shell, and then gradually reduce this stiffness to zero. If bending stiffness is included then load incrementation may become redundant.

When finding natural shapes, we start with the undeformed configuration being that of the meridian lying on the arc of a circle (assuming that the meridional length and base radius have been specified), as shown later in figure 4.8(a). This is done because the deformed shape of an overpressurised natural shape bag is relatively similar to an arc of a circle.

4.5.1 Constraints Matrix

The force vector, load vector, stiffness matrix and load stiffness matrix must be constrained before being used in the Newton-Raphson method. This can be accomplished by simply removing rows and columns of the vectors and matrices that correspond to forces and displacements in the constrained degrees of freedom. However, it is not so straightforward to constrain nodes in directions other than *x*, *y*, and *z*, and so a constraints matrix is used. Although in this axisymmetric FE model we only need to constrain nodes in the *x* and *y* directions (e.g. keeping the bottom node stationary by constraining it in *x* and *y* and only allowing the top node to move vertically by constraining it in *x*), the three-dimensional FE described later on is not so simple and we may wish to do things such as constrain some nodes to remain on a plane that does not coincide with any of *x*-*y*, *y*-*z*, or *z*-*x*. We call the constraints matrix *T* and use it to transform the force vectors and stiffness matrices as follows:

$$F' = TF \tag{4.58}$$

$$K' = TKT^T \tag{4.59}$$

T is formed using another matrix *C*, which is used to define the constraints. In this two-dimensional model, *C* has two columns, and the number of rows is equal to the number of degrees of freedom in the unconstrained model (so twice the number of nodes). C is a set of vertically concatenated 2×2 matrices, one for each node. *T* is formed by putting the submatrices comprising *C* into the appropriate places in a sparse matrix. It is necessary to remove rows of *T* that contain all zeros otherwise the constrained stiffness matrix and constrained load stiffness matrix would be singular.

As an example of how to form *C*, consider the three-noded meridian shown in figure 4.7, with top node constrained in *x* and bottom node constrained in *x* and *y*.

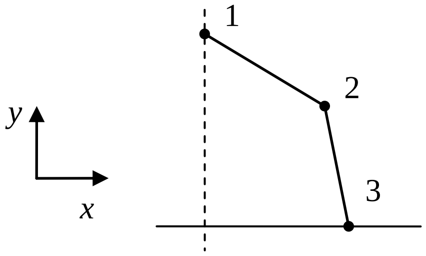


Fig. 4.7 Three-noded meridian

In this case, *C* should be

$$C = \left[\begin{array}{cc} 0 & 0 \\ 0 & 1 \\ \hline 1 & 0 \\ 0 & 1 \\ \hline 0 & 0 \\ 0 & 0 \end{array}\right]. \tag{4.60}$$

So once *C* has been rearranged to form *T*, rows 1, 5, and 6 of *T* must be removed as they will contain all zeros. In this case the resulting constraints matrix is

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.61}$$

Using *T* as shown in equations (4.58) and (4.59), the elements of the force residuals vector ($f_{reac} - f_{load}$) and residuals stiffness matrix ($K - R$) corresponding to the constrained degrees of freedom (x_1, x_3, and y_3) are removed. Nodes can be constrained in other directions that do not coincide with *x* or *y* by making use of rotation matrices. Including the constraints matrix, equation (4.55) becomes

$$u_{i+1} = u_i - T^T\left(T\left(K_i - R_i\right)T^T\right)^{-1} T\left(f_{reac,i} - f_{load,i}\right). \tag{4.62}$$

Note that the first instance of T^T in this equation is necessary to include the constrained degrees of freedom in the calculated displacements.

4.5.2 Seabed Resistance Forces

When finding partially-inflated shapes of a bag with a wide sealed base, it is necessary to ensure that the nodes do not cross into the seabed, and this is accomplished by adding a vector of vertical resistance forces to the load vector. Resistance inversely proportional to the distance of the node from the seabed is appropriate, and so a suitable element seabed resistance vector for a 2-node element with 4 degrees of freedom is

$$F_{sb} = 0.5k\begin{bmatrix} 0 & \dfrac{1}{y_1+v_1} & 0 & \dfrac{1}{y_2+v_2} \end{bmatrix}^T \tag{4.63}$$

where *k* is a value such that $k/(y+v)$ is equal to the differential pressure force at a small height, λ, above the seabed when $y+v=\lambda$. The appropriate form for *k* is thus

$$k = \lambda|p(\lambda)|A, \tag{4.64}$$

where *A* is the swept area of the element, as given in equation (4.46).

Note that the seabed resistance force on a node under the seabed will be vertically downwards. However, this is not a problem because in each iteration of the Newton-Raphson solver, the deformed node positions are checked to ensure that they have not crossed into the seabed, and if they have then the displacement vector is multiplied by a suitable fraction so that the nodes no longer cross into the seabed. Of course, the seabed resistance forces must also be included in the load stiffness matrix, and this is accomplished by adding the seabed resistance stiffness matrix to the load stiffness matrix. The element seabed resistance stiffness matrix used here is given by

$$\frac{\partial F_{sb}}{\partial u} = 0.5\left(k\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\dfrac{1}{(y_1+v_1)^2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\dfrac{1}{(y_2+v_2)^2} \end{bmatrix} + \begin{bmatrix} 0 & \dfrac{1}{y_1+v_1} & 0 & \dfrac{1}{y_2+v_2} \end{bmatrix}^T \frac{\partial k}{\partial u} \right), \tag{4.65}$$

where

$$\frac{\partial k}{\partial u} = \lambda|p(\lambda)|\left(\frac{\pi L}{n}\begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} + \frac{\pi}{n}(x_1+u_1+x_2+u_2)\frac{1}{L}\begin{bmatrix} -a_x & -a_y & a_x & a_y \end{bmatrix} \right). \tag{4.66}$$

F_{sb} and $\partial F_{sb}/\partial u$ are added to the element load vector and element load stiffness matrix respectively, and multiplied by the load fraction along with the other loads and load stiffnesses.

Seabed resistance forces should only be introduced when necessary, as they can have an effect on the shape of sections of the membrane that are only subjected to low forces.

This is described in more detail in section 4.6.3. If the mass of the bag materials is being ignored, seabed resistance forces need only be introduced if $p_0 < 0$ and the base of the bag has nonzero radius ($r > 0$) – only in such instances could the base of such a bag be forced against the seabed. However, a heavy bag could be forced against the seabed even with $p_0 > 0$ and $r = 0$, so care should be taken with when seabed resistance forces are introduced.

A simpler approach to resistance forces is to allow nodes to pass into the seabed and treat the seabed as an elastic body with stiffness k_{sb}. This method has been successfully used (with damping) to simulate bottom interaction in various dynamic models of ocean cable structures [117],[118]. F_{sb} is just a vertical force and can be calculated for each node (rather than for each element). The simplest seabed resistance force used in this way is

$$\begin{aligned} F_{sb} &= -k_{sb}y \text{ if } y < 0 \\ F_{sb} &= 0 \text{ if } y \geq 0 . \end{aligned} \tag{4.67}$$

However, the derivative of the vertical load vector would not be continuous at $y = 0$ (the seabed), so a modified approach which is continuous at the seabed would be

$$\begin{aligned} F_{sb} &= k_{sb}y^2 \text{ if } y < 0 \\ F_{sb} &= 0 \text{ if } y \geq 0 . \end{aligned} \tag{4.68}$$

Unfortunately, neither of equations (4.67) or (4.68) can be used to successfully find the shapes of highly deflated bags, having tried a variety of stiffnesses. We also tried fixing nodes with $y < 0$ to the seabed using the constraints matrix – in this way, if $y < 0$ then we set $y = 0$ and remove the appropriate row of the constraints matrix. However, this approach does not work because at the pressure level where the first node becomes anchored to the seabed, the bag becomes unstable and will not solve, and gradually all of the nodes become anchored to the seabed. This also happens when seabed-crossing nodes are constrained in x as well as y. Not too much time was invested in trying to solve this problem because we do not consider the need to find accurate highly deflated shapes to be great enough; the maximum stresses in the tendons will occur when the bag is highly inflated. In any case, a bag with a wide base would probably not be an ideal design because deployment would not be straightforward, the anchors (e.g. ballast or piles) would need to withstand sideways loads as well as vertical loads (especially when partially inflated), and if it were to be sealed at the base then it would require a large plate at the base or a rigid hoop at the base with reinforced fabric across the hoop.

4.5.3 Convergence Criteria

After each iteration of the Newton-Raphson method, the force residuals and nodal displacements must be compared with some criteria to determine whether a solution has been reached. The maximum absolute value of all the components of the constrained force residuals is compared with some small value, the maximum absolute value of the constrained dimensionless residuals (explained further on) is compared with another small value, and the maximum absolute value of the nodal displacements is compared with another small value (which can be a small fraction of some dimension of the undeformed bag, e.g. the meridional length of the undeformed tendon). The convergence criteria are met when either both of the residuals criteria are met or when the displacement criterion is met.

The dimensionless residual ($\hat{r}$) at each node is calculated by taking the magnitude of the force residual (r) at the node and dividing it by the sum of the absolute values of the cable tensions and seabed resistance force at the node. For each node,

$$r = F_{reac} - F_{load} \tag{4.69}$$

$$\hat{r} = \frac{r}{T_{m,up} + T_{m,lo} + T_c + F_{sb}}. \tag{4.70}$$

$T_{m,up}$ is the tension in the meridional cable above the node, $T_{m,lo}$ is the tension in the meridional cable below the node, T_c is the circumferential tension at the node and F_{sb} is the seabed force at the node. The vectors of force residuals and dimensionless force residuals are formed and the constraints matrix is used to remove the elements of the vectors that correspond to constrained displacements. The maximum absolute value of all the constrained force residuals is found, along with the maximum absolute value of all the constrained dimensionless force residuals, and these are compared with the residuals criteria.

We also keep track of the number of iterations that have been performed for a given load fraction, and stop the solution process if this iteration count gets too high. This causes the program to stop trying to find a solution when one can’t be found, though in such instances it may be that before this maximum number of iterations is reached, the nodes work themselves into positions whereby the residual stiffness matrix becomes singular – Matlab automatically stops the code from running and presents an error message when this happens.

4.6 Results

In all of the following examples, the bottom node of the bag is constrained so that it cannot move in any direction (as if fixed to the seabed), and the top node is constrained so that it can only move in the *y*-direction (so vertically up and down).

4.6.1 Natural Shape Energy Bags

Figure 4.8 shows the starting configuration and deformed shape of an overpressurised single point of anchorage (SPA) natural shape bag with meridional length L = 2.36m and differential pressure at base of p_0 = 100kPa. For reference, the prototype bag has 36 tendons, and in the work presented here, each is given E = 200GPa and A = 3.142x10^{-4}m^2. The bag is able to deform so much from the starting configuration because there is no circumferential reinforcement in a natural shape bag – natural shape bags are designed so that at all stages of inflation there is no part of the fabric where there is a shortage of length in the circumferential direction. Therefore no circumferential stiffness is included in the model. No material mass or bulkheads have been included. The bag clearly takes on the pumpkin shape of superpressure natural shape balloons, as seen in the literature review.

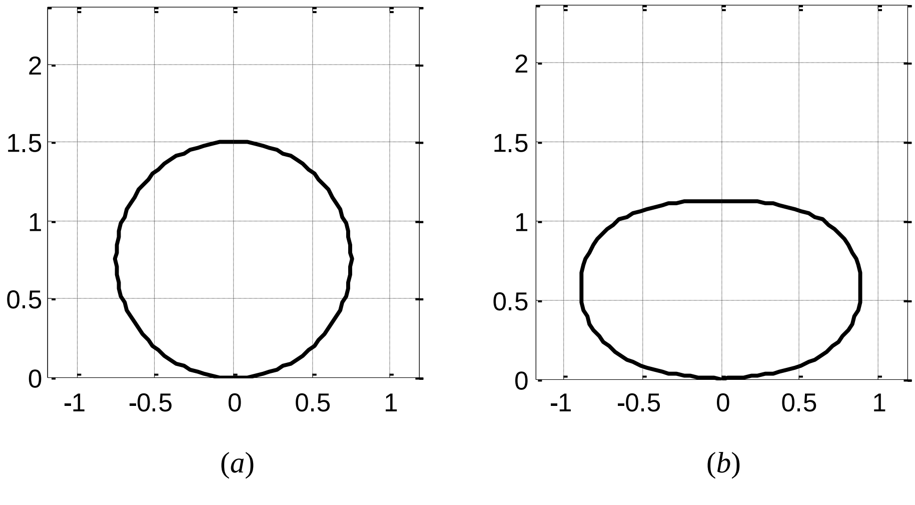


Fig. 4.8 Modelling a natural shape bag with L = 2.36m, (*a*) Starting configuration; (*b*) Deformed configuration with p_0 = 100kPa

Figure 4.9 shows partially inflated shapes of the massless bag shown in figure 4.8, with p_0 ranging between 20kPa and -20kPa in 10kPa increments. The shape with p_0 = 0Pa is the ZPNS for this bag, and is very similar to the shape of hot air balloons. The lower portion of the most deflated bag has what is known as a “rope section” – the bag has collapsed in against itself due to the large negative *p* in this section. Note that because the load is only carried meridionally, the point of inflexion in the meridian is at the level of zero differential pressure (as is to be expected from the Young-Laplace equation). This can be seen by looking at the three most deflated shapes; the level of zero differential pressure for the ZPNS is at the seabed (h = 0m) and at h = 0.995m and h = 1.989m for the other two shapes. These levels were calculated using g = 9.81m/s^2 and ρ_w = 1,025kg/m^3, but if we were to use g = 10m/s^2 and ρ_w = 1,000kg/m^3 then these levels would be at 0m, 1m, and 2m. It is anticipated that the air hose connection would need to be located at the top of an Energy Bag whose sides meet during deflation, otherwise the air may not escape after the sides of the bag meet.

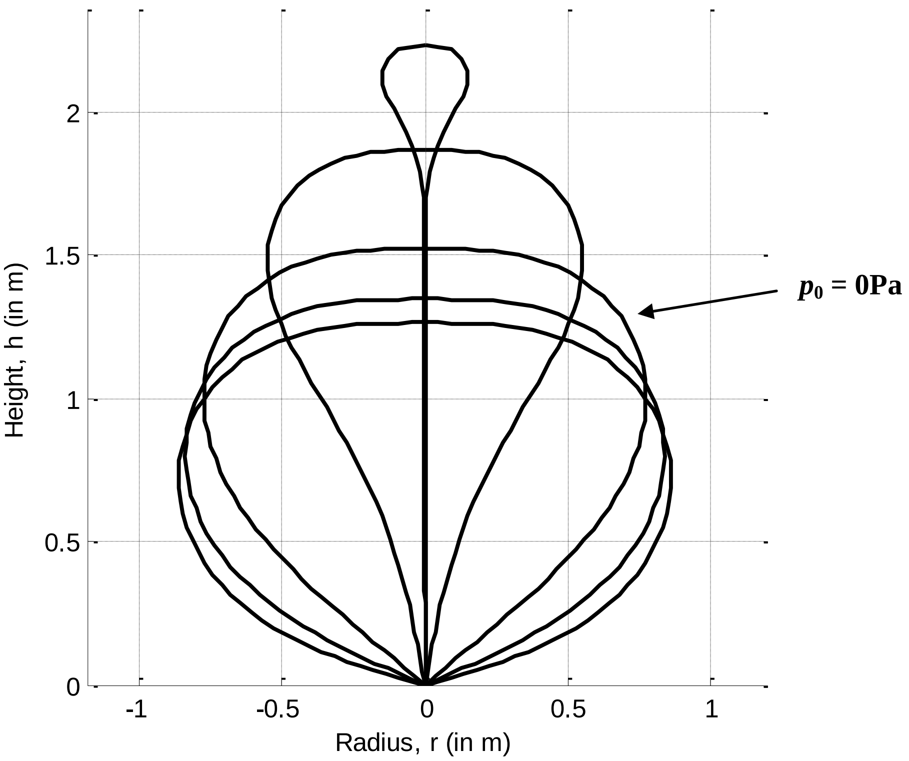


Fig. 4.9 Partially inflated shapes of a natural shape bag from p_0 = 20kPa to p_0 = -20kPa in 10kPa increments

The pressure-volume (PV) curve for this bag when the base of the bag is anchored at a depth of 2.4m is shown in figure 4.10. The effect of water level rise due to displacement of water is not included – it may have a noticeable effect on the PV curve for this prototype when it is being tested in a small tank, but the effect when the bag is anchored in a larger body of water, such as the sea, will be negligible.

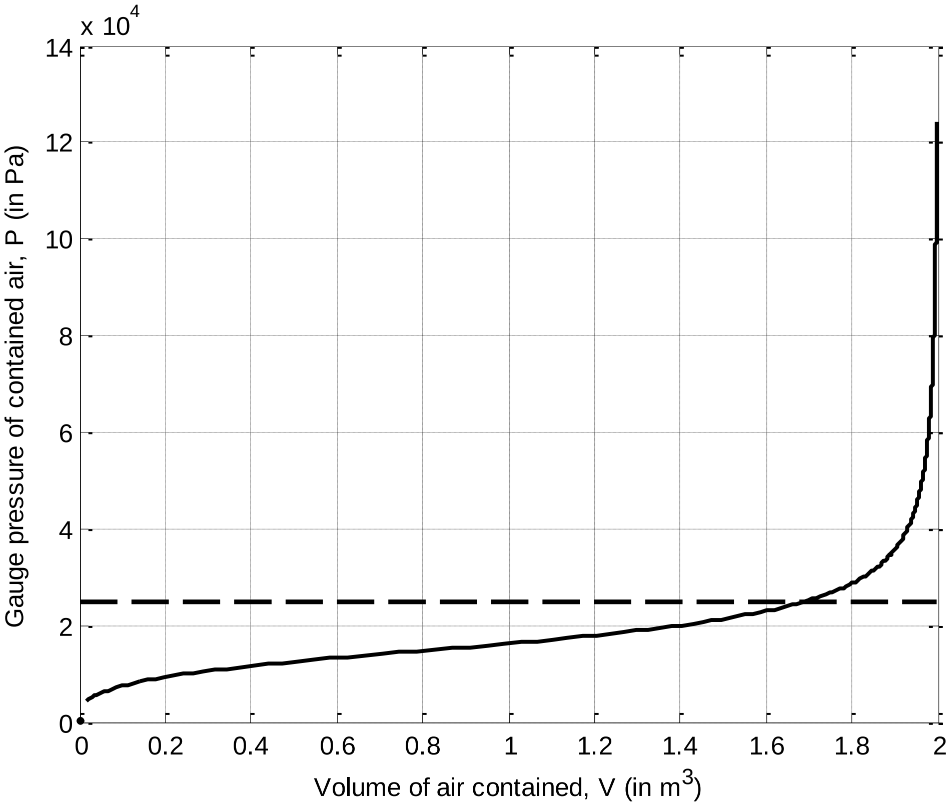


Fig. 4.10 Pressure-volume curve for the prototype bag anchored at 2.4m depth, inflated up to p_0 = 100kPa. Dashed line indicates hydrostatic pressure at 2.4m depth (the base of the bag)

The pressure at V = 0m^3 is marked on figure 4.10 as a dot (at $P = 0.04\times10^4$ Pa), and is the gauge pressure of the air at which the level of zero differential pressure (p = 0Pa) is at the top of the fully deflated bag, or simply the hydrostatic pressure at the depth of the top of the fully deflated bag. As the base radius and upper bulkhead radius are set to zero, this is the pressure at which p = 0Pa is at $h = L$ = 2.36m, and is given by

$$\begin{aligned} P &= \rho_w g(depth - h) \\ &= 1025\times9.81\times(2.40-2.36) \\ &= 0.04\times10^4\,\text{Pa}. \end{aligned} \tag{4.71}$$

It should be noted that if material mass were included, this level would not be reached because there will come a point where the volume and so buoyancy will be less than the mass of the materials, and the bag will drop towards the seabed. It will not be possible to find any further solutions when this happens, and in any case it is not necessary to find any because of the low stresses in a highly deflated bag.

The PV curve is clearly quite shallow up to approximately 1.9m^3. This is because of the ability of a flexible fabric structure to deform and change its volume, allowing more efficient turbomachinery to be used than with a fixed volume CAES scheme. Evidently when the bag is nearly full, the volume doesn't increase with pressure as it does at lower pressures, as at these high pressures the pumpkin shape has been reached and the only possible further increase in volume is by extension of the tendons.

We now introduce material mass. Each of the 36 tendons has a diameter of 20mm, and is given the density of steel (approximately 7,800kg/m^3). The membrane material is 1mm thick and made of rubber with density of 1,522kg/m^3. When looking at the strains in the partially-inflated shapes of bags with material mass, it was found that the highly deflated bag had negative strains at the very base – the elements had begun to go into compression and were being supported by the very small compressive stiffness that they are given in the wrinkling model. This occurs because the bottom of the bag sees the weight of the whole bag, and the buoyancy force has been reduced to such an extent that it is less than the weight of the whole bag. In reality the bag will drop slightly until enough of the bottom of the bag is resting against the seabed. Note that as the bag drops, it moves into an area of higher pressure, and so the contained air will be compressed even more (becoming slightly heavier) but the volume and net buoyancy of the bag will reduce. This can be viewed as a good thing because it will serve to flatten the PV curve at the lower volumes, however it may not be desirable for the bag to drop in case it rubs or catches against the ballast and seabed. A float could always be attached to the upper bulkhead to ensure that it never drops to the seabed, or the air control system could be designed so that the volume of air in the bag never gets so low that the bag will drop.

Figure 4.11 shows the shape of the bag just before it will start to drop, and the distribution of strain in the tendon. The phenomenon of the bag slowly dropping towards the seabed as it empties is also encountered with helium-filled Mylar celebration balloons; as they leak helium into the environment their buoyancy decreases, and after a few days they begin to very gradually drop towards the floor. The FEA code was modified to check if any meridional elements go into compression as p_0 is reduced, and to stop reducing p_0 if so. We assume that the lower bulkhead is fixed in place – in reality it may be attached by cable to the ballast/anchor, in which case the whole bag may drop towards the seabed before the strain at the bottom of the tendons becomes zero.

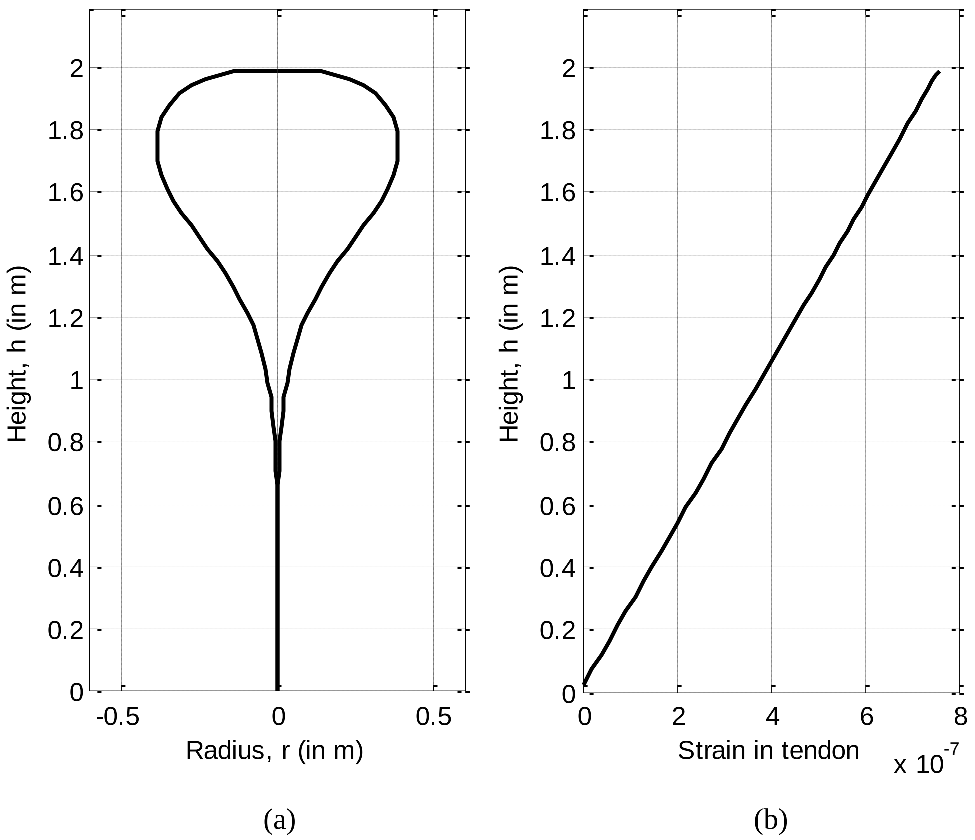


Fig. 4.11 (a) The shape of a deflating ZPNS bag with material mass just before it starts to drop towards the seabed. (b) Distribution of strain in the tendon

We have found that the shape of a ZPNS bag is independent of the depth at which it is anchored. Therefore anchoring the bag at greater depths has very little effect on the shape of the PV curve, simply translating the curve up to greater pressures. The density of the stored air is increased but not by such a large amount that there is a noticeable effect on the shape of the curve, and a plot of p_0 against V for the bag at both 2.4m depth and 240m depth on the same axes is not worth showing as the curves are so similar that there is no way to distinguish between the two. Similarly, for a given p_0, depth has little effect on the loading on the tendons, meaning that the cost of bag materials is independent of the depth at which the bag is located.

While discussing the effects of anchorage depth, it should be noted that in shallow water the bag deflates slightly further before beginning to drop to the seabed than the same bag (with the same material mass) in deeper water. This is because for a given volume, the density of the contained air in the deep water bag will be higher, and so the mass of the air will be greater. For the bag shown in figure 4.11, in 2.4m deep water it can deflate to $-p_0/\rho_w g(L\text{-}r)$ = 0.6326 before starting to drop, but in 500m deep water it only deflates to $-p_0/\rho_w g(L\text{-}r)$ = 0.5864 before it starts to drop.

Using this FE model, we can show that for any given ratio of p_0/L, the natural shape is independent of the size of the bag, assuming small strains. This can be shown by setting a value of p_0/L, and then plotting the normalised natural shapes found with various values of L on the same axes. Figure 4.12 shows normalised plots (to give bag height = 1m in each case) of the natural shape with $p_0/L = \rho_w g$ for bags with L = 1m (solid line) and L = 100m (dashed line). In each case, the Young's modulus of the tendon is multiplied by L^3, so that all the bags have the same ratio of cable stiffness to buoyancy. (Note that the buoyancy force is proportional to bag volume V, which is proportional to L^3.)

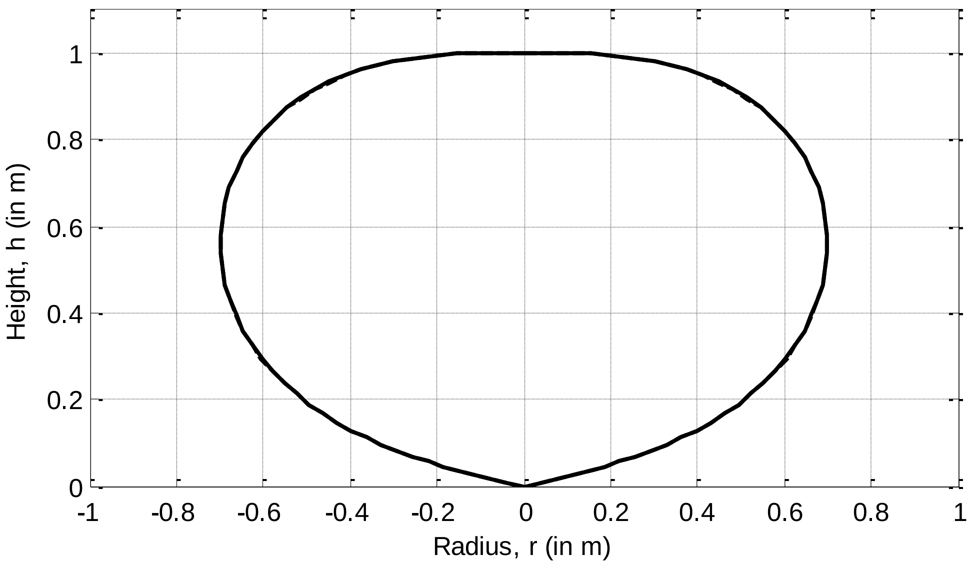


Fig. 4.12 Normalised plots of natural shape bag with L = 1m (solid line) and L = 100m (dashed line), $p_0/L = \rho_w g$

Clearly the two curves are so close that there is no visible difference between the two. Figure 4.13 shows normalised plots of the same two bags, this time with $p_0/L = -0.6\rho_w g$. The dashed red line indicates the level at which p = 0. The differences between the two shapes are greater than with $p_0/L = \rho_w g$, but they are still so close that it is hard to see the dashed black line.

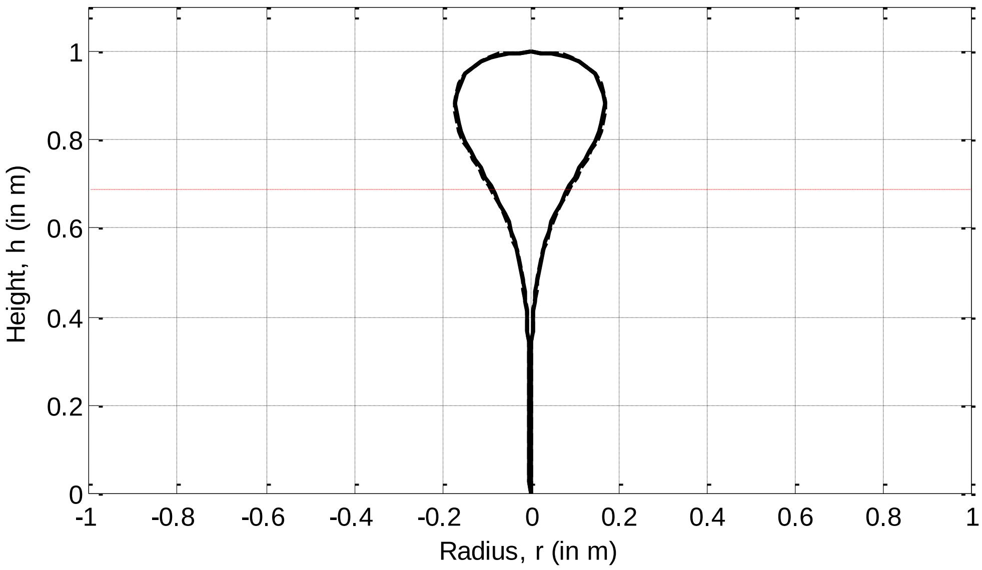


Fig. 4.13 Normalised plots of a partially deflated natural bag with L = 1m (solid black line) and L = 100m (dashed black line), $p_0/L = -0.6\rho_w g$. Dashed red line indicates level at which p = 0

We have also found that PV curves for larger bags are stretched relative to smaller bags in both the x and y directions, because larger bags have larger pressure and volume ranges. However, the shape of the normalised PV curve is independent of the size of the bag; this is to be expected given that we have just shown that the shape of the bag is independent of its size. The fact that there is a larger range of contained air pressures for larger bags makes using a small number of large bags slightly less attractive than using a large number of smaller bags because the more constant the pressure of the air entering the turbomachinery, the more efficient the turbomachinery will be. However, it may be that cost per unit storage capacity of the bag will have a greater impact on the optimum size than the efficiency of the turbomachinery, and future work could investigate this tradeoff.

4.6.2 Energy Bags with Circumferential Reinforcement

Figure 4.14 shows the stages of inflation of a 1.5632m diameter sphere (so having a fully-inflated volume of 2m^3) with meridional length L = 2.4554m, with 36 tendons and 49 circumferential cables (evenly spaced along the meridian), all with Young's modulus of 200GPa and cross-sectional area of 3.142×10^{-4}m^2. Again, we assume that the bag is anchored at a depth of 2.4m – this depth is just less than L, so it would not be possible to fully deflate this bag without exposing some of the top.

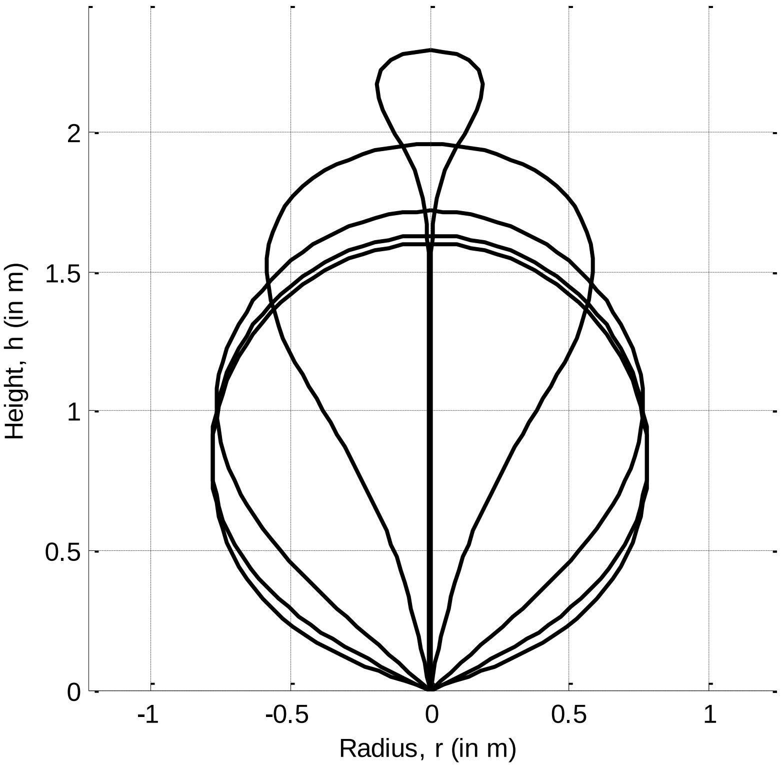


Fig. 4.14 Partially inflated shapes of a sphere from p_0 = 20kPa to p_0 = -20kPa in 10kPa increments

The shapes of the partially-inflated spherical bag clearly look quite similar to those of a partially-inflated natural shape bag; compare figure 4.14 with figure 4.9. However, the top of the partially-inflated spherical bag is not as flat as the top of the partially-inflated natural shape bag, because the spherical bag's circumferential restraint is active at the top of the bag at all fill levels. The lower parts of a spherical bag become stressed circumferentially as the bag is inflated.

Figure 4.15 shows the stages of inflation of a vertical cylinder capped with a hemisphere at each end. The undeformed cylinder has a diameter of 2m and total height (including endcaps) of 7m. It has 36 tendons and 29 circumferential cables (all with E = 200GPa and A = 3.142x10^{-4}m^{2}), and is anchored at 500m depth.

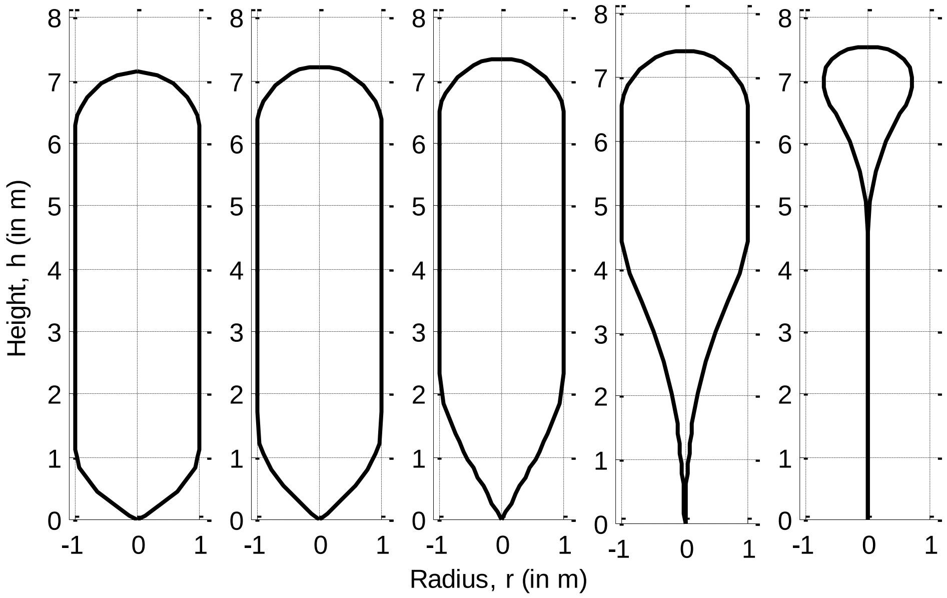


Fig. 4.15 Partially inflated shapes of a cylinder with hemispherical caps from p_0 = 50kPa to p_0 = -60kPa in 27.5kPa increments

For a large part of the deflation, the top section of the deflating cylinder maintains its full diameter. The lower sections of the bag collapse in as the internal pressure is reduced, and circumferential wrinkling occurs in these sections as they take on the natural shape. Again, the most deflated shape shown has the classic bubble and rope section.

4.6.3 Wide Base Natural Shape Energy Bags

This section looks at the shape of a natural shape bag with a wide base. The shapes shown are only valid if the bag is sealed against the seabed. If the bag is not sealed at the base, any overpressure at the base ($p_0 > 0$) will cause air to leak from the base of the bag and rise to the surface in bubbles until $p_0 = 0$, and any underpressure at the base will allow water to leak into the base of the bag, compressing the air trapped inside. Such an unsealed bag could still be useful if the air hose connection is located at the top of the bag, and is discussed further on in section 4.6.4.

Figure 4.16 shows partially inflated shapes for a bag with L = 2.36m and base radius r = 0.472m (= L/5), from p_0 = 5kPa down to p_0 = -15kPa in 5kPa increments. As expected, the base of the bag is pressed against the seabed when p_0 is low. The PV curve for this bag is shown in figure 4.17. The curve is flatter than that for a SPA bag because the collapsing in of the lower parts of the bag against the seabed means that the bag remains quite close to the seabed at all times.

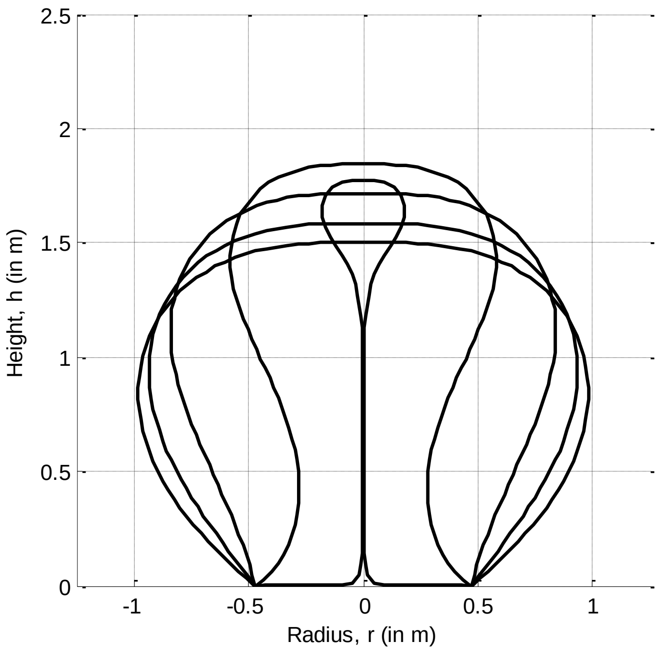


Fig. 4.16 Partially inflated shapes of a wide base natural shape bag from p_0 = 5kPa to p_0 = -15kPa in 5kPa increments

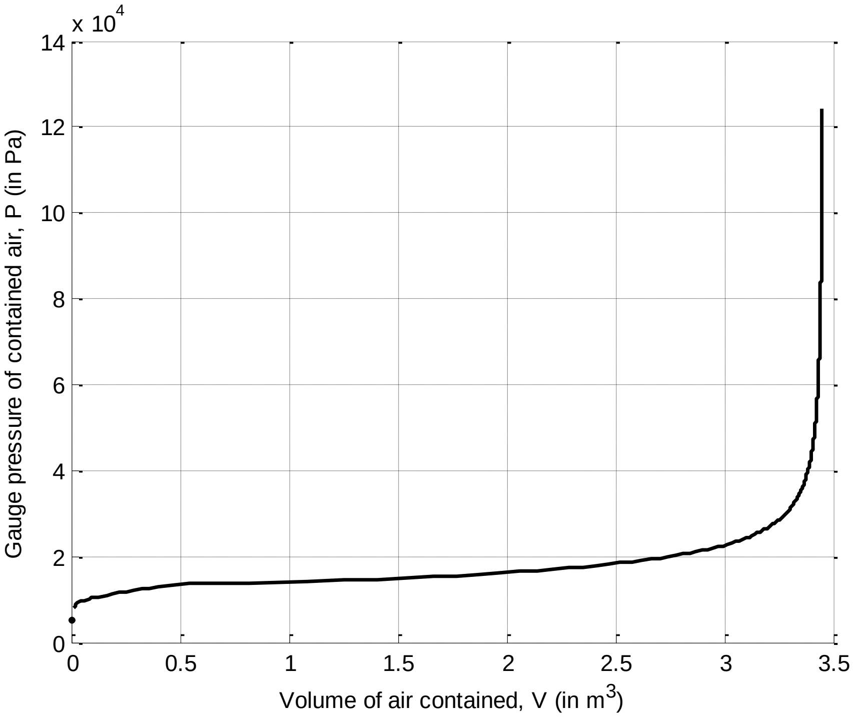


Fig. 4.17 Pressure-volume curve for the bag shown in figure 4.16, anchored at 2.4m depth, inflated up to p_0 = 100kPa

4.6.4 Energy Bags with an Unsealed Base

A bag that is not sealed at the base will allow water into the lower part of the bag as it deflates, with a pocket of air trapped above. For static equilibrium, this air must have a pressure that is equal to the hydrostatic pressure at the bottom of the air pocket, so the differential pressure across the membrane at the bottom of the air pocket will be zero (see figure 4.18). The differential pressure across the membrane below the air pocket must also be zero. If the water level inside the bag is at a height *h* = *w* above the base of the bag, then

$$p_w = 0 \tag{4.72}$$

and the gauge pressure of the air is given by

$$P_a = \rho_w g(d - w), \tag{4.73}$$

where *d* is the depth of the base of the bag. If the gauge pressure of the air inside the pocket is known, the water level inside the bag can be calculated by rearranging to get

$$w = d - \frac{P_a}{\rho_w g}. \tag{4.74}$$

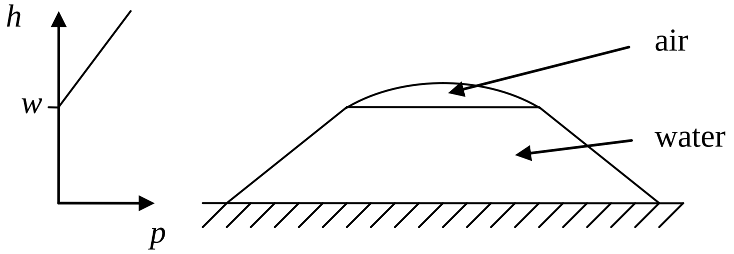


Fig. 4.18 Wide base unsealed bag with plot of height above base against differential pressure across the membrane

If the bags shown in figures 4.9 and 4.14 – 4.16 are open at the base, the deformed shapes are as shown in figures 4.19 – 4.22. Note that to find the starting (so most highly pressurised) solution for the bag shown in figure 4.19, we cannot start with a small load fraction and so must start with a high load fraction (e.g. 1), and prestress is not required. However, the opposite is true for the bag shown in figure 4.22, where in order to find the starting solution we have to start with a small load fraction and must include prestress in the meridional elements.

The unsealed bags shown in figures 4.19 – 4.22 have almost exactly the same shape as inverted water load test weights (detailed in section 2.2.2) would have if made with the same initial shape. In each case the water level would be at the level of the red dashed line. The only differences between the shape are because of: the direction of the material weight relative to the direction of the weight of contained water/buoyancy of contained air (though the plots shown here are for massless bags), and the added mass of the compressed air. Modifying the FE code to account for these differences and more accurately model water load test weights would take very little effort.

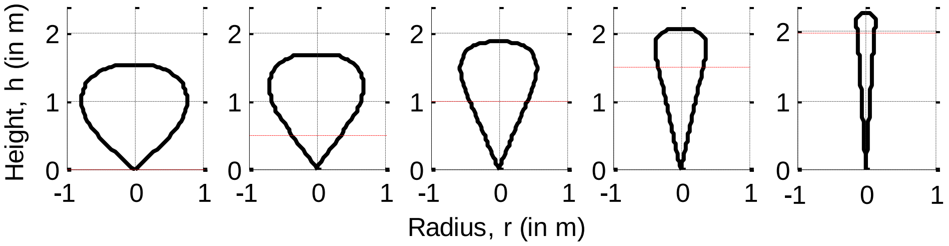


Fig. 4.19 Partially inflated shapes of a natural shape bag with unsealed base; red dashed line indicates water level inside the bag

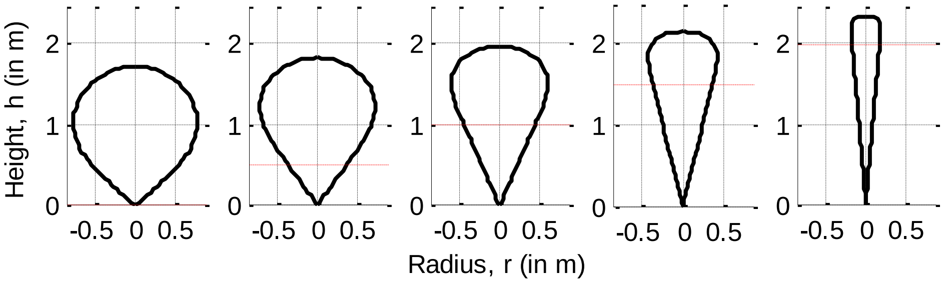


Fig. 4.20 Partially inflated shapes of a spherical bag with unsealed base; red dashed line indicates water level inside the bag

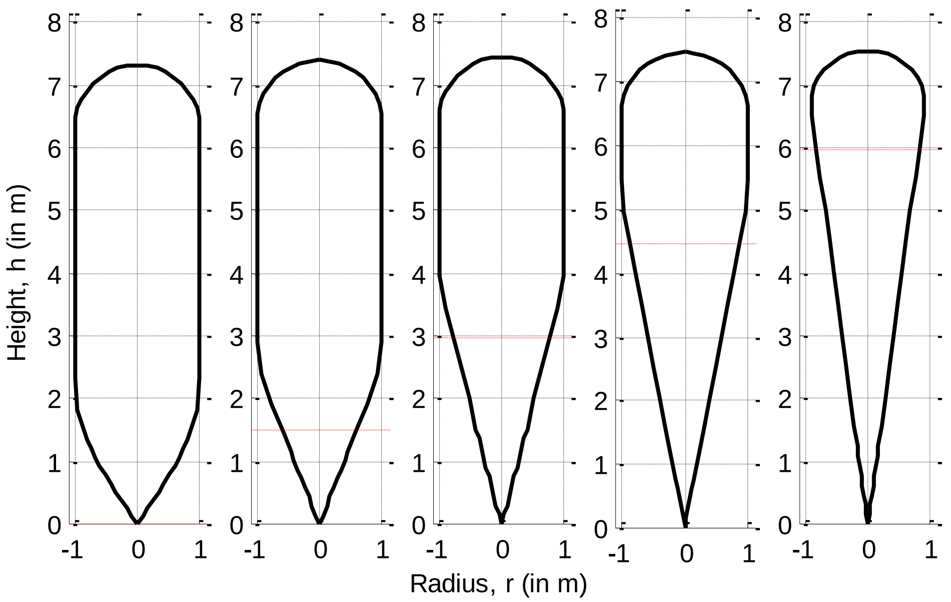


Fig. 4.21 Partially inflated shapes of a cylindrical bag with hemispherical caps and unsealed base; red dashed line indicates water level inside the bag

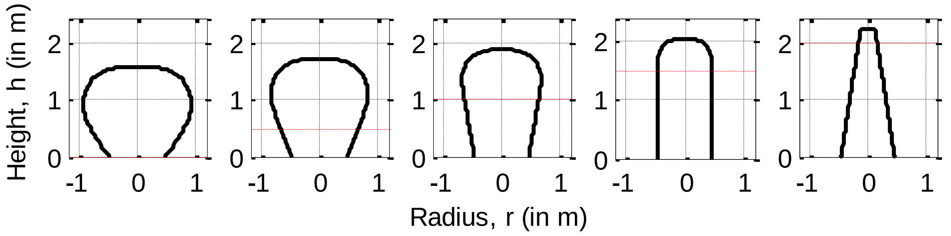


Fig. 4.22 Partially inflated shapes of a wide base natural shape bag with unsealed base; red dashed line indicates water level inside the bag

As expected from the Young-Laplace equation, the sections of the bag that are below the air pocket are straight because they have no differential pressure force across them (bearing in mind that these sections are wrinkled circumferentially and so T_c = 0). It is necessary to ensure that seabed resistance forces are not included when modelling an unsealed bag, as they cause these sections to have a curved profile rather than straight. Note that including membrane and/or tendon mass will cause these sections to sag slightly anyway. Again, there will come a point during deflation where material mass causes the bag to drop towards the seabed. The air connection should be located at the top of a bag with an unsealed base, so that all of the air (and so stored energy) is vented before the hose becomes blocked with water. It may be worth attaching a float to the top of an unsealed Energy Bag with buoyancy at least equal to the mass of the empty bag and the attached hose, to ensure that the deflating bag doesn't drop towards the seabed and allow water into the hose before all of the air has left the bag (as well as for the reasons mentioned in section 4.6.1).

Figure 4.23 shows a comparison of the PV curves for massless SPA natural shape bags with sealed base and unsealed base. The two curves are very similar, but it can be seen that at lower pressures, the bag with sealed base has a slightly lower volume than the bag with unsealed base for a given contained air pressure. This is because the sealed bag collapses in on itself.

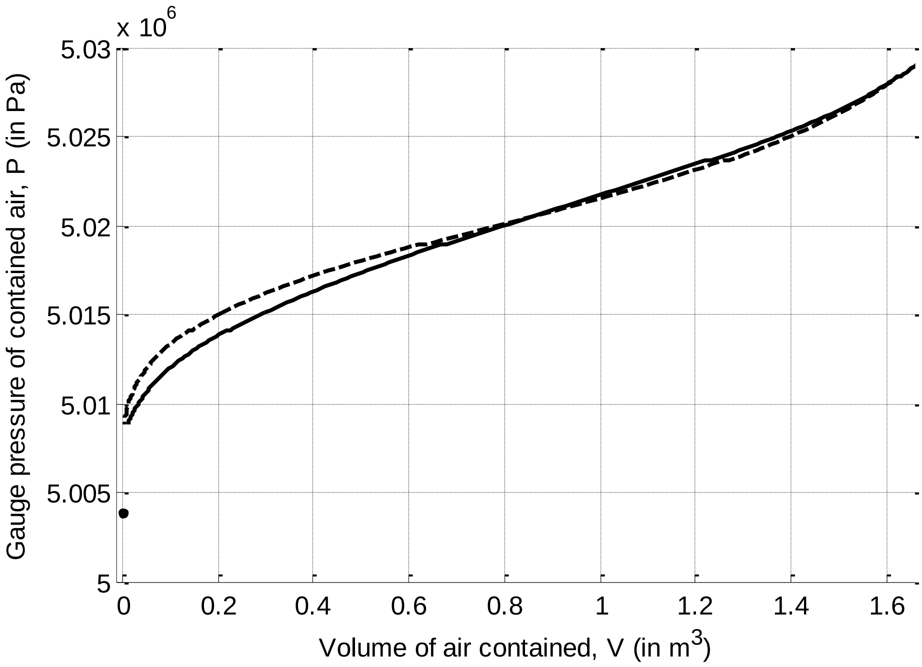


Fig. 4.23 PV curves for massless SPA natural shape bags with *L* = 2.36m: solid line – unsealed base, dashed line – sealed base

Chapter 5

Lobed Fabric Structures

Lobing in fabric structures is used to lower the membrane stresses, because we know from the Young-Laplace equation that stress is inversely proportional to the curvature of the membrane. Lobed structures can be assembled by welding lobes together at the seams (using a technique such as RF welding) and incorporating a stiff tendon along the boundaries between lobes. In manufacturing a rotationally symmetric balloon or Energy Bag, a simpler approach is to simply use two identical sheets of fabric (e.g. circles) welded together at the edges, incorporating stiff tendons enclosed in pockets. The tendons are shortened to form lobes (otherwise the only lobing would be due to stretching of the membrane); if they are shortened until no meridional stress is carried in the fabric then all of the meridional loads are carried in the tendons. In this way, only circumferential stresses are carried in the membrane, and the high curvature of the lobes in the circumferential direction ensures that these stresses are not very high. A good description of lobes is given in [119].

5.1 Constant Angle and Constant Radius Lobes

Pagitz [120] shows that in lobed balloons with large numbers of lobes (e.g. 200), the cross-sections of the lobes are essentially circular arcs (as in figure 5.1), and lobes are typically designed to have one of two different types of cross-section: constant angle (CA) and constant radius (CR). As the names suggest, a constant angle lobe subtends a constant angle at all points along the meridian, and a constant radius lobe has a constant radius at all points along the meridian. In figure 5.1, α is the subtended angle (constant in a CA lobe), r_l is the lobe radius (constant in a CR lobe), r_b is the bag radius (measured from the centreline of the bag to the tendon, in the horizontal plane), w is the arc length of the lobe (or width of the lobe cutting pattern), and W is the width of the lobe in the deformed configuration. The arc length of the lobe (w) is required to set up the model, so we calculate w for a CA lobe and then for a CR lobe.

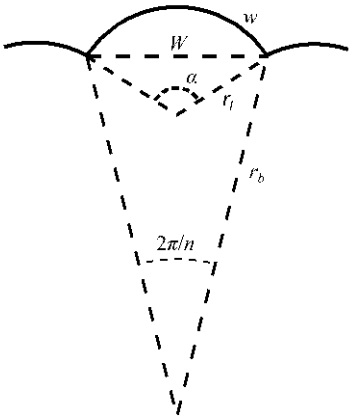


Fig. 5.1 Cross-section of a lobe (in the horizontal plane)

If the subtended angle of a lobe is given by α, then the local width of the lobe cutting pattern at a distance s along the centre of the lobe from the base is given by

$$w(s) = \alpha(s) r_l(s). \tag{5.1}$$

For a CA lobe, we set α and must derive an expression for the lobe radius $r_l(s)$. We know that

$$\sin(\alpha/2) = \frac{W(s)/2}{r_l(s)} \tag{5.2}$$

and rearrange for the lobe radius,

$$r_l(s) = \frac{W(s)}{2\sin(\alpha/2)}. \tag{5.3}$$

The lobe width in the deformed configuration, W, is given by

$$W(s) = 2r_b(s)\sin(\pi/n), \tag{5.4}$$

so the radius of a CA lobe is

$$r_l(s) = \frac{r_b(s)\sin(\pi/n)}{\sin(\alpha/2)} \tag{5.5}$$

and the cutting pattern width of a CA lobe is

$$w(s) = \alpha\,\frac{r_b(s)\sin(\pi/n)}{\sin(\alpha/2)}. \tag{5.6}$$

For a CR lobe we set r_l and must derive an expression for the subtended angle, $\alpha(s)$. Rearranging equation (5.5) we obtain the subtended angle for a CR lobe,

$$\alpha(s) = 2\sin^{-1}\left(\frac{r_b(s)}{r_l}\sin(\pi/n)\right). \tag{5.7}$$

Therefore the cutting pattern width of a CR lobe is

$$w(s) = 2r_l\sin^{-1}\left(\frac{r_b(s)}{r_l}\sin(\pi/n)\right). \tag{5.8}$$

The cutting patterns of both CA and CR lobes depend upon the tendon profile.

5.2 Ultra High Performance Vessel Architecture

Two 1.8m diameter prototype Energy Bags were manufactured for us by Thin Red Line Aerospace Ltd. (TRL), a Canadian company that manufactures deployable fabric structures for use in space. The design of the prototype Energy Bags is based on TRL's rotationally symmetric lobed habitation structures and balloons which have their "Ultra High Performance Vessel" (UHPV) architecture. A UHPV is essentially a pumpkin balloon formed from two circles of fabric welded together at the edges, with meridional tendons running through sleeves attached to the surface of the fabric. Two circles of fabric joined at the edges must form a natural shape bag because the radii of the fabric at all points must reduce as the natural shape is formed, no matter how inflated the bag is (assuming realistic levels of meridional strain). This reduction in radius causes circumferential wrinkling (meridional wrinkles), as there is excess fabric around the circumference.

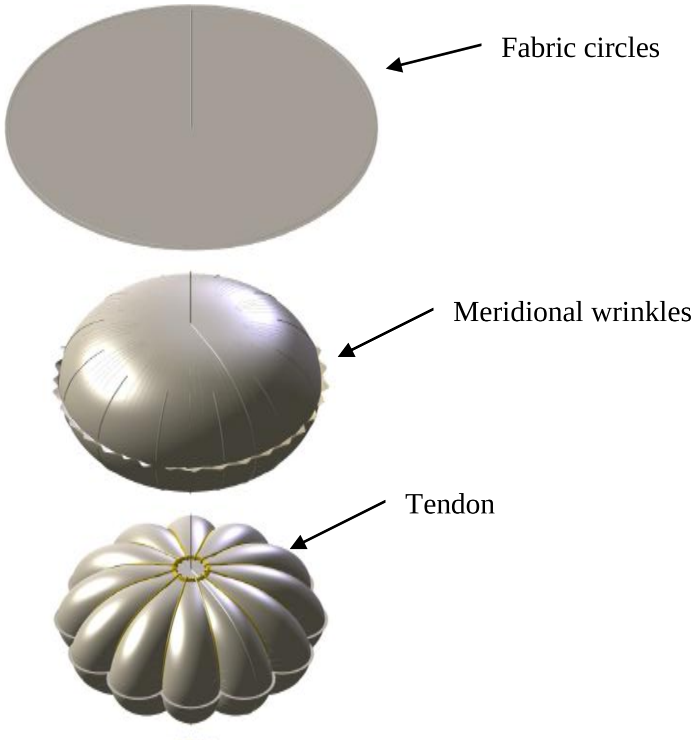


Fig. 5.2 A UHPV formed from two circles of fabric must have circumferential wrinkling at all points when inflated, thus forming a natural shape bag [121]

In a UHPV, the tendons are shortened relative to the fabric along which they run in order that this excess fabric in the circumferential direction is gathered up. With increased tendon shortening, the meridional stress in the fabric is reduced, and in a UHPV the tendons are shortened to such an extent that all meridional stress is removed from the fabric. Consequently the fabric only carries circumferential stress, the load path is deterministic, and the small radius in the circumferential direction (relative to the radius of the fabric in the meridional direction) ensures that the stresses in the fabric are low. “Bellows”, essentially just extra fabric folded up, can also be included around the equator to ensure that there is no meridional stress in the fabric. Bellows can be formed by just making the fabric circles larger than necessary and putting a circumferential fold around the edge before attaching the tendons.

The fabric of a UHPV can be laid completely flat on the ground even after the circles of fabric have been welded together (though they cannot be laid flat after shortened tendons have been included, as seen in figure 5.3). This is because the edges of the flat “lobes” are straight (figure 5.4), and differs from a vessel with separate, tailored lobes (e.g. CA, CR, and CT lobes), which have curved lobe edges. Deflated vessels with curved lobe edges cannot be laid flat on the ground without folding the excess fabric between each lobe.

UHPVs require no more than two pieces of fabric – an upper circle and a lower circle – but in larger bags each of these circles may have to comprise several pieces of fabric (if the diameter of the circle is larger than the width of the manufactured roll of fabric). Also, using several pieces of fabric to form each circle is likely to lower the amount of wasted fabric and allows the fabric to be aligned so that the highest stresses are carried in the strongest direction. (Fabric is generally weaved with fibres lying in two perpendicular directions, warp and fill. The direction at an angle of 45° between the warp and fill directions is known as the bias. Usually the warp direction is the strongest, then bias, then fill.)

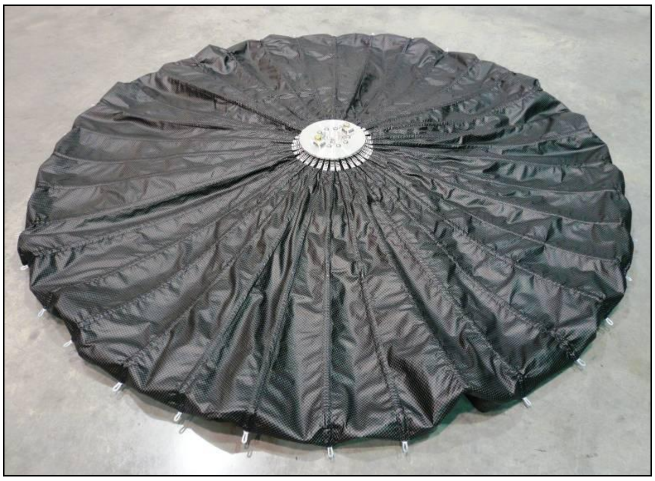


Fig. 5.3 A prototype Energy Bag with the UHPV architecture laid out on the ground. Wrinkling due to tendon shortening is visible

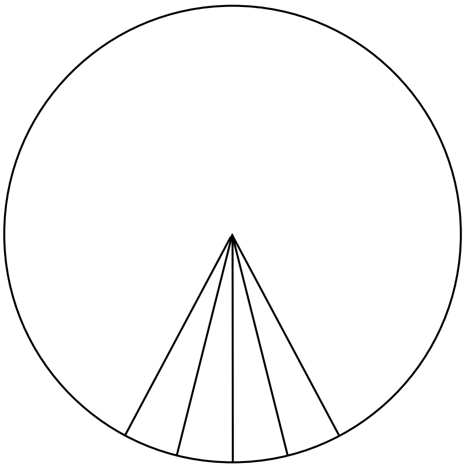


Fig. 5.4 Top view of a deflated UHPV laid flat, showing 4 "lobes". Note the flat edges of the "lobes". TRL's UHPV has circular sheets as shown here, but the sheets could take any other shape, as in figure 5.5

One of the advantages of UHPVs is that they are very simple to manufacture. It is shown later on that the stresses in the fabric of a pressurised lobed Energy Bag or balloon are inversely proportional to the number of lobes in the structure. A large number of lobes is not a problem for a UHPV, which requires no more than one weld, but if a bag or balloon is formed from separate lobes that require welding at the seams, the number of welds is equal to the number of lobes. Another advantage to having a low number of seams is that the chance of seam failure (which causes leaking and can lead to total failure of the bag) is reduced.

One of the disadvantages of UHPVs is that, because they may simply be formed from two sheets of fabric welded together along the perimeters, the edges of the "lobes" must be straight, and cannot be tailored to minimise stresses or improve stability as they could if the lobes were all separate pieces of fabric (though they could feasibly be shaped by pinching parts of the fabric and welding/sewing the pinch in place). The pieces of fabric forming a UHPV do not necessarily have to be circular, nor do they even have to be rotationally symmetric, but a rotationally symmetric pattern might be beneficial (as in figure 5.5). However, such a pattern may well be harder to cut than circles. The tendons may also be shortened by any desired amount.

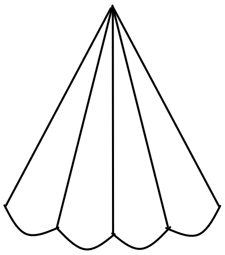


Fig. 5.5 The two sheets of fabric that form a UHPV may take any desired shape. The four lobes shown here have rotational symmetry

Thin Red Line have proposed using their UHPVs for space habitation, in which case straight sides may be included between the upper half of the vessel and the lower half (as shown in figure 5.6).

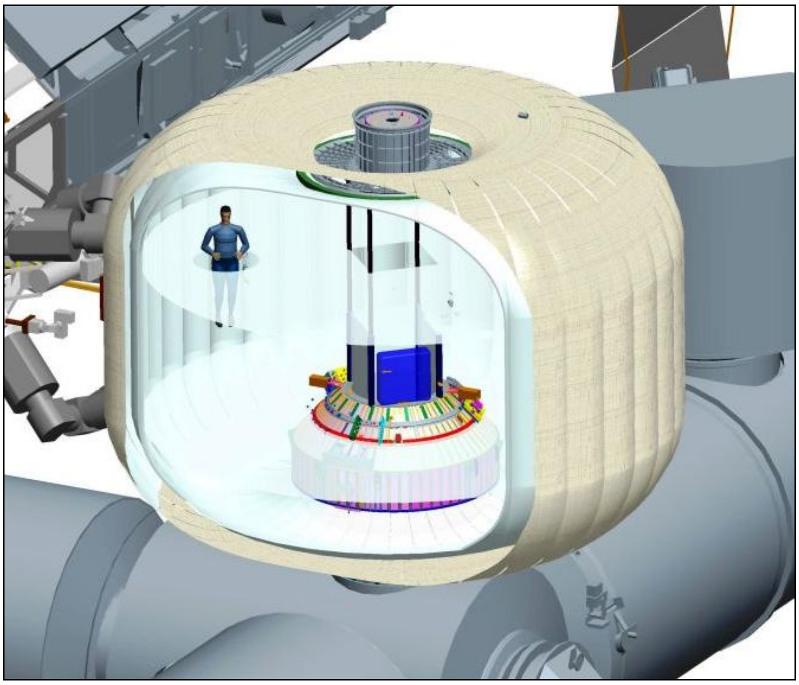


Fig. 5.6 A proposed space habitation vessel using the UHPV architecture, designed to be attached to the International Space Station

5.3 Membrane Stresses

If the meridional membrane stress has been removed using tendon shortening then the circumferential tension in a lobe (per unit meridional length) depends upon the local radius of curvature r_l, and is expressed as

$$T_c(s) = r_l(s)p(s). \tag{5.9}$$

Supposing that we desire to minimise the circumferential stress at all points in a lobe, then at all points the lobe's circumferential curvature should be as large as possible. In order to maximise this curvature, the lobe's radius should be as small as possible, and assuming a circular cross-section, such a lobe would be a CA lobe with constant subtended angle of π radians, as shown in figure 5.7.

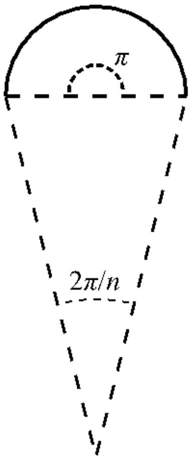


Fig. 5.7 Cross-section of a lobe which subtends π radians

For the bag in figure 5.7, the local lobe radius is given by

$$r_l(s) = r_b(s)\sin(\pi/n). \tag{5.10}$$

However, a CA lobe will have lower radii at the top and bottom of the bag than at the middle, and so the circumferential tension at the top and bottom of the bag will be lower than the circumferential tension at the middle. Assuming that the strength and thickness of the membrane used in an Energy Bag will be constant at all points in the bag, it is unnecessary to minimise the tension at the top and bottom of the lobe when the material must be strong enough to withstand the maximum tension (encountered slightly above the middle of the lobe) anyway. Balloon designers often use CR lobes – this is likely to be because designers of superpressure balloons often assume a uniform differential pressure at all heights in the balloon, in which case a constant lobe radius will give constant circumferential tension. As expected, CR lobes have smaller widths than CA lobes (see figure 5.8), and CR lobes are more uniformly stressed (figure 5.9).

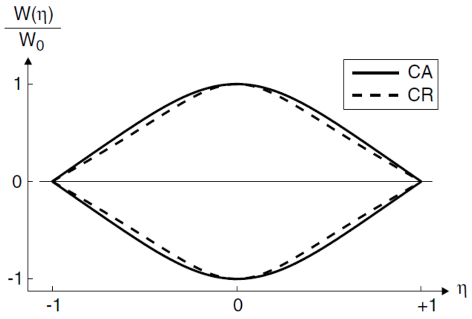


Fig. 5.8 Constant angle (CA) and constant radius (CR) lobe cutting patterns (with equal width at the equator) for a pumpkin balloon [92]. η is the dimensionless distance along the lobe from the equator ($\eta = 0$) to the top ($\eta = 1$) and bottom ($\eta = -1$). $W(\eta)$ is the lobe width at η and W_0 is the lobe width at the equator

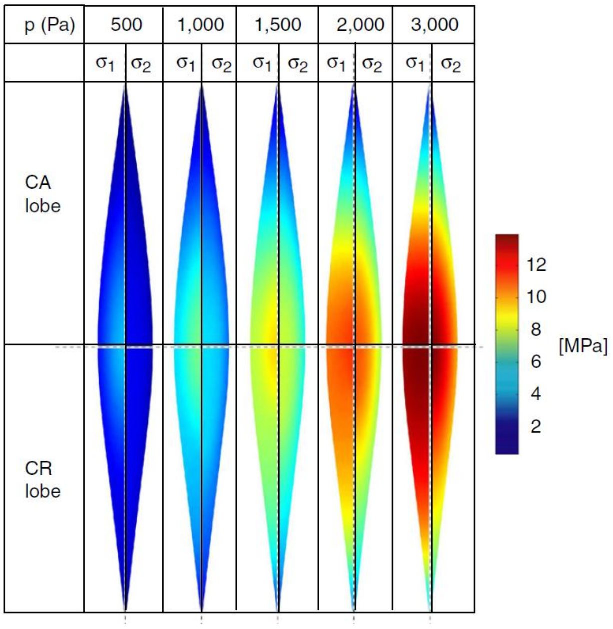


Fig. 5.9 Major and minor principal stresses for a CA and CR lobe in a 10m diameter balloon with 145 lobes, plotted on undeformed lobe cutting pattern (width magnified 5 times) [92]

In the next section, we introduce the constant tension lobe, a new type of lobe cutting pattern with a smaller area than both CA and CR lobes. If for now we proceed with the idea of using a CA lobe with subtended angle $\alpha = \pi$ rads (giving a lower bound on the circumferential tension at all points a lobe), we arrive at an interesting result. The following equation, found by substituting equation (5.10) into equation (5.9), gives the circumferential tension per unit meridional length in such a lobe,

$$T_c = pr_b \sin(\pi/n). \tag{5.11}$$

Taking small angle approximations for a vessel with a large number of lobes, equation (5.10) becomes

$$r_l = \frac{\pi r_b}{n}. \tag{5.12}$$

To manufacture larger balloons or Energy Bags, the number of tendons is increased. The buoyancy force, and so total tension in the tendons, is proportional to the bag radius cubed,

$$T_{t,tot} \propto r_b^{\ 3}. \tag{5.13}$$

The tension in each tendon is given by

$$T_t \propto \frac{r_b^{\ 3}}{n}. \tag{5.14}$$

If *n* is proportional to the bag radius r_b then the tension in each tendon is proportional to the bag radius squared,

$$T_t \propto \frac{r_b^{\ 3}}{r_b} \propto r_b^{\ 2}. \tag{5.15}$$

If the cross-sectional diameter of each tendon is kept proportional to bag radius, then the stress in each tendon is independent of the bag radius,

$$\sigma_t \propto \frac{T_t}{A_t} \propto \frac{r_b^{\ 2}}{r_b^{\ 2}} = \text{constant}. \tag{5.16}$$

Therefore the number of tendons can be kept proportional to the bag radius without an increase in tendon stress, as long as the cross-sectional diameter of each tendon is also kept proportional to the bag radius. Looking back at equation (5.12), we can see that if the number of tendons is proportional to the bag radius, the local lobe radius is independent of the size of the bag. Looking further back to equation (5.9), we then see that if the differential pressure p remains unchanged with bag size (as is roughly the case for superpressure balloons), the circumferential tension is also independent of the size of the bag. This differs from unlobed vessels and is the reason why the addition of lobes dramatically increases the pressure resistance of large balloons. It is interesting to now think back to the reason Graham Brading gave for why Seaflex don't make underwater lifting bags with lifting capacity any greater than 35t (in section 2.2.3), bearing in mind that Seaflex's lifting bags don't have any lobing: "The hoop stress in the fabric at the largest diameter of the bag limits the size. We could look at higher tensile strength fabric if needs be."

For an Energy Bag, p (at a given height in the bag as a fraction of the bag's meridional length) is proportional to the size of the bag, so the circumferential tension is proportional to the bag's size. The circumferential tension in an Energy Bag can therefore be made independent of the size of the bag if we make the number of tendons proportional to the bag radius squared (in which case the cross-sectional area of the tendons need only be proportional to the bag radius).

Unfortunately, superpressure balloons can have problems with stability, and it has been shown that the buckling pressure of a superpressure balloon reduces as the number of lobes and the subtended angle of each lobe increase. Therefore it may be worthwhile to assess the stability of Energy Bags and seek the lobe cutting pattern that minimises membrane stress while the bag remains stable. Of course, stability may only be an issue if Energy Bags are designed to be overpressurised, and so the stability of less highly pressurised (e.g. zero pressure) bags should also be investigated.

5.4 The Constant Tension Lobe

Balloon designers often neglect the pressure gradient and assume a uniform differential pressure across the balloon at all heights. However, it is not strictly correct to assume a uniform differential pressure across an inflated vessel, particularly underwater where the pressure gradient is much higher than that in air, so we now propose a new lobe cutting pattern that takes the pressure gradient into account, known as the constant tension (CT) lobe. Such a lobe will have constant circumferential tension at all points in the lobe (equal to the maximum circumferential tension encountered in a CA lobe with constant subtended angle of π rads) and because p depends upon the height within the bag, neither α nor r_l will be constant. A CT lobe cutting pattern will minimise the width of the lobe while meeting the maximum stress condition, so minimising the area of the lobe cutting pattern. This has two advantages over other lobe designs: 1) the cost of the lobe will be as low as possible, and 2) the bag stability will be increased (it has been shown that the stability of a superpressure balloon is maximised by minimising the cutting pattern area [104]). The top of a CT lobe will be slightly wider than the top of a CR lobe (because *p* is maximum at the top of the bag), and the bottom of a CT lobe will be slightly thinner than the bottom of a CR lobe (because *p* is minimum at the bottom of the bag, see figure 5.15 for confirmation).

The procedure for generating a CT cutting pattern is as follows:

1. Start with a CA lobe with $\alpha = \pi$ rads (so minimising T_c at all points along the lobe) and find the maximum value of $T_c(s)$ (which we now call $T_{c,max}$). It is anticipated that this will be located at a point slightly above the maximum radius of the bag.
2. Calculate the lobe radius at every other point along the lobe using $r_l(s) = T_{c,max}/p(s)$.
3. Calculate the subtended angle $\alpha(s)$ at all points using the local lobe radius ($r_l(s)$ from step 2) and equation (5.7).
4. Calculate the cutting pattern width $w(s)$ at all points using $r_l(s)$ from step 2, $\alpha(s)$ from step 3, and equation (5.1).

The CT lobe cutting pattern for a massless ZPNS bag with L = 1m and 36 lobes is shown in figure 5.10. Interestingly the lobe edges of a CT lobe are very straight, with slight curvature at the point of maximum width. The point of maximum width is slightly above the point on the lobe at which the bag has maximum diameter, as anticipated. The point on the lobe at which the bag has maximum diameter depends upon the maximum fill level of the bag; for a massless bag this point will always be above halfway up the lobe and will be closest to halfway up the lobe when the maximum fill level is high (because a massless bag only has a symmetrical shape if the differential pressure is infinite). Figure 5.11 shows a cross-section of the ZPNS bag with a dashed line to indicate the level of maximum lobe width for the CT lobe.

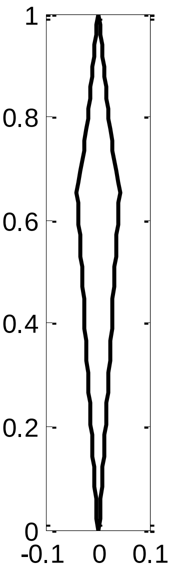


Fig. 5.10 CT lobe cutting pattern for a massless ZPNS bag with 36 lobes and *L* = 1m (shown in figure 5.11)

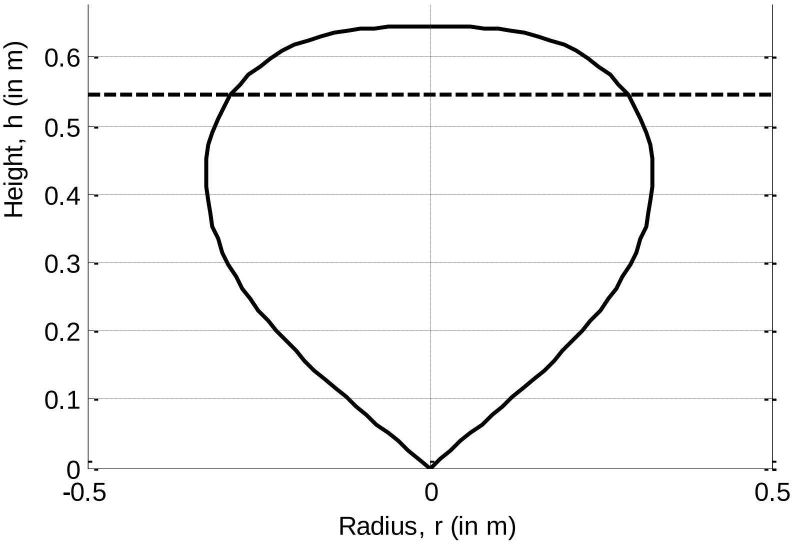


Fig. 5.11 Massless ZPNS bag with dashed line indicating the level at which the CT lobe cutting pattern width is maximum

It should be noted that the CT lobe cutting pattern with $\alpha_{max} = \pi$ rads is not necessarily the lobe cutting pattern of lowest cost, because in order to minimise the maximum tension, lobes with large bulges are used, which require lots of material. If we assume that cost of surface is proportional to the product of total cutting pattern width and circumferential tension, then the cost of surface is

$$C_{su} \propto T_c wn\,. \tag{5.17}$$

Using equations (5.17), (5.9), (5.5), and (5.6), we obtain

$$C_{su} \propto pn\left(r_b \sin\left(\pi/n\right)\right)^2 \frac{\alpha}{\sin^2\left(\alpha/2\right)}\,. \tag{5.18}$$

For a sufficiently large number of lobes, $C_{su} \propto 1/n$. Differentiating equation (5.18) with respect to α and setting equal to zero, we can obtain the optimum subtended angle in a particular element.

$$\frac{\partial C_{su}}{\partial \alpha} = 0 \tag{5.19}$$

$$\alpha = \tan\left(\alpha/2\right) \tag{5.20}$$

$$\alpha = 2.3309\,\text{rads} \tag{5.21}$$

So the subtended angle that minimises cost of surface in a particular element is 2.3309rads, or 133.5°. It should be noted that this calculation ignores the effect of the size of the lobe's bulge on the enclosed volume of the bag (and so the amount of stored energy). However, it has been found that taking the enclosed volume of the bag into account has very little effect on the optimum lobe angle; figure 5.12 shows a plot of C_{su}/A, where A is the total enclosed area of the lobe (which is proportional to the amount of stored energy), for n = 36, r_b = 0.9m, and p = 49810Pa. The subtended angle α spans the range $2\pi/n$ (so giving a circular bag cross-section) to $\pi + 2\pi/n$ (the largest lobe bulge possible), and clearly there is a marked drop in surface cost per unit of energy stored as lobing is introduced. The minimum is at α = 2.3rads, very close to the optimum angle calculated without taking the effect of the lobe bulge on the amount of stored energy into account. The number of lobes, bag radius, and differential pressure have very little effect on the shape of the curve.

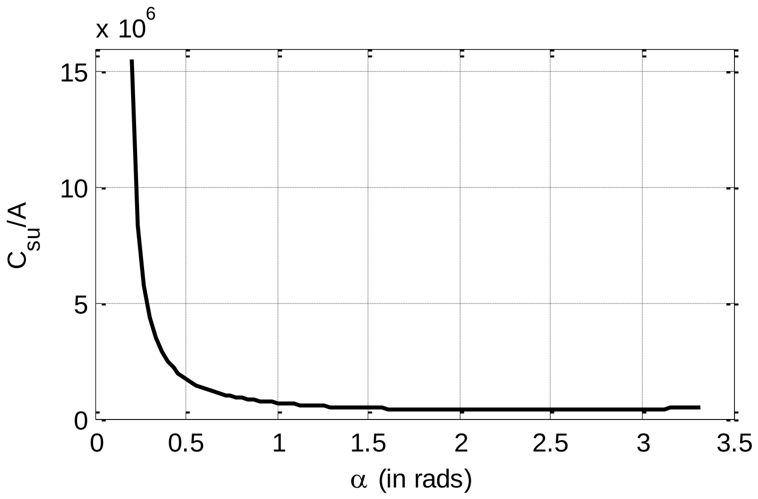


Fig. 5.12 Plot of cost of surface per unit of contained area (which is proportional to the amount of stored energy) against the lobe angle. n = 36, r_b = 0.9m, and p = 49,810Pa.

As mentioned before, the fabric used for the lobe will have constant thickness and so in order to maximise material utilisation and minimise the cutting pattern area, the circumferential tension should be constant throughout the lobe. A constant angle lobe does not have constant tension. Setting the constant tension to the maximum tension found with α = 2.3309rads does not necessarily give the optimum cutting pattern either; the optimum cutting pattern should be found using an optimisation routine. For a ZPNS bag with L = 2.36m and 36 lobes it was found that the optimum CT lobe has maximum subtended angle of α = 2.9615rads, or 170°.

Figure 5.13 shows CT lobe cutting patterns for a superpressure bag with L = 1m, p_0 = 1bar, and 36 lobes. Figure 5.13(a) shows the cutting pattern for a lobe with constant tension equal to the maximum tension found with $\alpha = \pi$ rads, and figure 5.13(b) shows the cutting pattern for a lobe with constant tension equal to the maximum tension found with α = 2.3309rads. Clearly the larger the maximum subtended angle, the straighter the edges of the CT lobe cutting pattern.

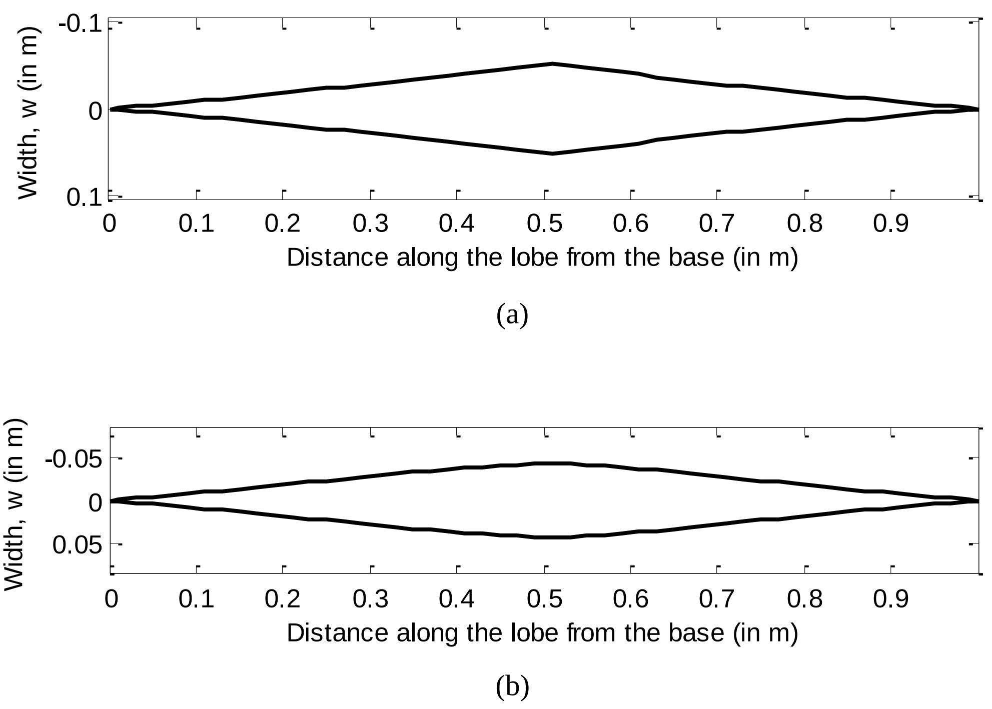


Fig. 5.13 CT lobe cutting patterns for a superpressure bag with L = 1m, p_0 = 1bar, and 36 lobes (a) constant tension equal to the maximum tension found with $\alpha = \pi$ rads, (b) constant tension equal to the maximum tension found with α =2.3309rads

In figure 5.14 we compare the CT lobe cutting pattern from figure 5.13(a) with a CA lobe cutting pattern for a lobe with constant angle of $\alpha = \pi$ rads. This shows that the CT lobe has equal width to the CA lobe at the point where the tension in the CA lobe is at a maximum, and is thinner than the CA lobe at all other points.

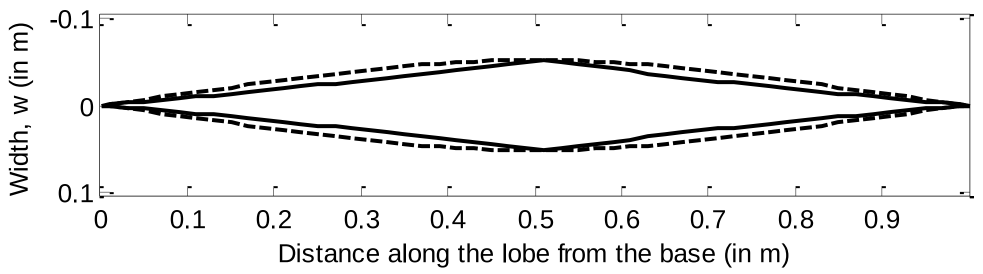


Fig. 5.14 Solid line: CT lobe cutting pattern from figure 5.13(a). Dashed line: CA cutting pattern for bag with constant angle of $\alpha = \pi$ rads

A CR lobe with lobe radius r_l equal to the smallest possible radius (half the straight-line distance between tendons at the point at which the bag has its largest radius, i.e.

$r_l = 0.5\max(W)$) has very similar shape to the CT lobe cutting pattern from figure 5.13(a) and figure 5.14. Figure 5.15 shows a plot of the relative differences in width between the CT lobe cutting pattern and the CR lobe cutting pattern. We see that the CT lobe is thinner than the CR lobe beneath the equator of the bag, and the CT lobe is wider than the CR lobe above the equator of the bag. The overall area of a CT lobe is slightly less than that of a CR lobe with maximum tension equal to the constant tension in the CT lobe.

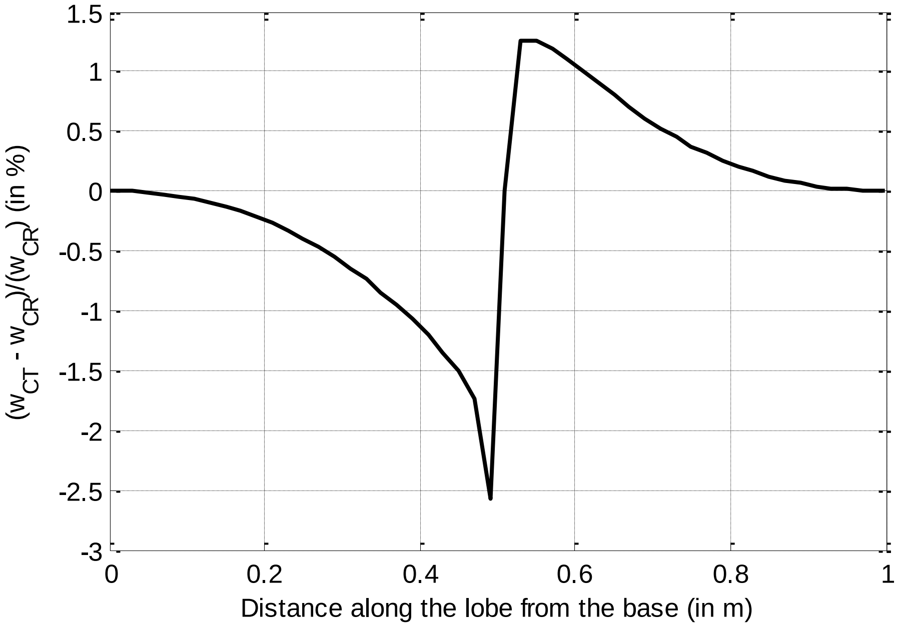


Fig. 5.15 Relative differences (in %) between the lobe width of a CT lobe and CR lobe for a bag with *L* = 1m, p_0 = 1bar, and 36 lobes

Chapter 6

Three-Dimensional Finite Element Analysis

In this chapter a three-dimensional finite element analysis of a cable reinforced membrane is presented, using membrane and cable elements. The cable element force vector and stiffness matrix have been derived previously in sections 4.1.1 and 4.1.2 respectively, and cable wrinkling was handled in section 4.1.3.

6.1 The Membrane Element

Membrane elements are two-dimensional elements which have no bending stiffness and so can only carry in-plane forces. They model solids of a specified (small) thickness which exhibit no stress normal to the thickness, and the constitutive relations are modified to ensure that this normal stress is zero.

6.1.1 Force Vector

The following derivations of the membrane element force vector and stiffness matrix are taken from [93] and [120], and given here for completeness. The element is derived in a local coordinate system, shown in figure 6.1, with the *x*-axis aligned along side 12 and the origin located such that the *y*-axis passes through node 3.

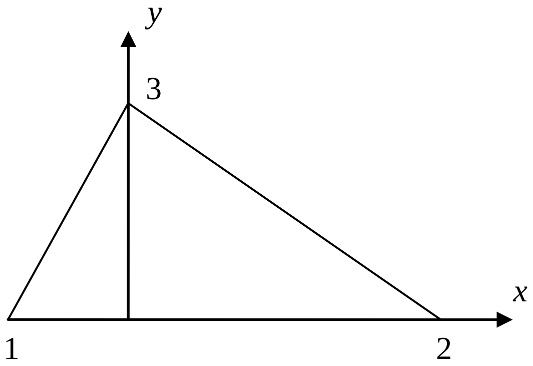


Fig. 6.1 A 3-node, 9 degree of freedom triangular element

The displacement at any point on the element is given by

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y\,,$$

$$v = \alpha_4 + \alpha_5 x + \alpha_6 y\,,$$

$$w = \alpha_7 + \alpha_8 x + \alpha_9 y\,. \tag{6.1}$$

Converting from the α coefficients to the nine nodal displacements

$$u = [u_1 \quad v_1 \quad w_1 \quad u_2 \quad v_2 \quad w_2 \quad u_3 \quad v_3 \quad w_3]^T \tag{6.2}$$

and solving for α by evaluating the displacements at each of the three nodes, we have the element shape functions,

$$u(x,y) = (a_1 + b_1 x + c_1 y)u_1 + (a_2 + b_2 x + c_2 y)u_2 + (a_3 + b_3 x + c_3 y)u_3\,,$$

$$v(x,y) = (a_1 + b_1 x + c_1 y)v_1 + (a_2 + b_2 x + c_2 y)v_2 + (a_3 + b_3 x + c_3 y)v_3\,,$$

$$w(x,y) = (a_1 + b_1 x + c_1 y)w_1 + (a_2 + b_2 x + c_2 y)w_2 + (a_3 + b_3 x + c_3 y)w_3\,, \tag{6.3}$$

where

$$a_1 = (x_2 y_3 - x_3 y_2)/2S\,,\; b_1 = (y_2 - y_3)/2S\,,\; c_1 = (x_3 - x_2)/2S\,,$$

$$a_2 = (x_3 y_1 - x_1 y_3)/2S\,,\; b_2 = (y_3 - y_1)/2S\,,\; c_2 = (x_1 - x_3)/2S\,,$$

$$a_3 = (x_1 y_2 - x_2 y_1)/2S\,,\; b_3 = (y_1 - y_2)/2S\,,\; c_3 = (x_2 - x_1)/2S\,. \qquad (6.4)$$

S is the undeformed area of the element.

The nonlinear displacement-strain relations in a two-dimensional element are given by

$$\varepsilon_x = \frac{\partial u}{\partial x} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right],$$

$$\varepsilon_y = \frac{\partial v}{\partial y} + \frac{1}{2}\left[\left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2\right],$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \left[\frac{\partial u}{\partial x}\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\frac{\partial v}{\partial y} + \frac{\partial w}{\partial x}\frac{\partial w}{\partial y}\right]. \qquad (6.5)$$

Writing the strain vector in matrix form,

$$\varepsilon = B_0 u + \frac{1}{2} A\theta\,, \qquad (6.6)$$

where

$$B_0 = \begin{bmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 \\ c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 \end{bmatrix}, \qquad (6.7)$$

$$A = \begin{bmatrix} \partial u/\partial x & \partial v/\partial x & \partial w/\partial x & 0 & 0 & 0 \\ 0 & 0 & 0 & \partial u/\partial y & \partial v/\partial y & \partial w/\partial y \\ \partial u/\partial y & \partial v/\partial y & \partial w/\partial y & \partial u/\partial x & \partial v/\partial x & \partial w/\partial x \end{bmatrix}, \qquad (6.8)$$

$$\theta = \left[\partial u/\partial x \quad \partial v/\partial x \quad \partial w/\partial x \quad \partial u/\partial y \quad \partial v/\partial y \quad \partial w/\partial y\right]^T . \tag{6.9}$$

The matrices *A* and θ can be formed using

$$A = \begin{bmatrix} (Pu)^T & 0_{1\times 3} \\ 0_{1\times 3} & (Qu)^T \\ (Qu)^T & (Pu)^T \end{bmatrix}, \tag{6.10}$$

$$\theta = Gu , \tag{6.11}$$

where

$$P = \begin{bmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 \\ 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 \\ 0 & 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 \end{bmatrix}, \tag{6.12}$$

$$Q = \begin{bmatrix} c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 \\ 0 & 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 \end{bmatrix}, \tag{6.13}$$

$$G = \begin{bmatrix} P \\ Q \end{bmatrix}. \tag{6.14}$$

A set of virtual displacements δu gives rise to virtual strains of

$$\delta\varepsilon = \left(B_0 + AG\right)\delta u . \tag{6.15}$$

The stress vector is found using the constitutive relations for linear elastic plane stress analysis,

$$\sigma = D\varepsilon + \sigma_0 \tag{6.16}$$

where σ_0 is the initial stress vector (prestress) and *D* is the elastic matrix, defined as

$$D = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{6.17}$$

where *E* and ν are the Young's modulus and Poisson's ratio of the material, respectively.

Using the principle of virtual work, the work done on a deformable body by a set of virtual displacements δu is given by

$$W = \int_V \delta\varepsilon^T \sigma dV \tag{6.18}$$

where *V* is the volume of the body. Substituting equations (6.15)-(6.16) into equation (6.18) and eliminating δu^T gives the element force vector,

$$F = \int_V \left(B_0 + AG\right)^T \left(D\left(B_0 u + \frac{1}{2} A\theta \right) + \sigma_0 \right) dV\,. \tag{6.19}$$

Integrating over the volume of the element is equivalent to multiplying by its volume because the interpolating polynomial is linear, and so for the element force vector we have

$$F = tS\left(B_0 + AG\right)^T \left(D\left(B_0 u + \frac{1}{2} A\theta \right) + \sigma_0 \right) \tag{6.20}$$

where *t* and *S* are the thickness and undeformed area of the element, respectively.

6.1.2 Stiffness Matrix

Differentiating *F* with respect to *u* to obtain the element stiffness matrix,

$$K = \frac{\partial F}{\partial u}$$

$$= \int_V (B_0 + AG)^T \frac{\partial}{\partial u}\left(D\left(B_0 u + \frac{1}{2} A\theta \right) + \sigma_0 \right) dV$$

$$+ \int_V \frac{\partial}{\partial u}\left((B_0 + AG)^T \right) \times \left(D\left(B_0 u + \frac{1}{2} A\theta \right) + \sigma_0 \right) dV$$

$$= K_e + K_g \,. \tag{6.21}$$

The elastic and geometric stiffness matrices are given by

$$K_e = \int_V (B_0 + AG)^T D(B_0 + AG) dV \,, \tag{6.22}$$

$$K_g = \int_V G^T \frac{\partial (A)^T}{\partial u} \sigma dV = \int_V G^T MG dV \,, \tag{6.23}$$

where

$$M = \begin{bmatrix} \sigma_x & 0 & 0 & \tau_{xy} & 0 & 0 \\ 0 & \sigma_x & 0 & 0 & \tau_{xy} & 0 \\ 0 & 0 & \sigma_x & 0 & 0 & \tau_{xy} \\ \tau_{xy} & 0 & 0 & \sigma_y & 0 & 0 \\ 0 & \tau_{xy} & 0 & 0 & \sigma_y & 0 \\ 0 & 0 & \tau_{xy} & 0 & 0 & \sigma_y \end{bmatrix} . \tag{6.24}$$

The element force vector and stiffness matrix must be rotated from the local coordinate system into the global coordinate system using standard transformation matrices, details of which can be found in many FEA textbooks including [122].

6.1.3 Wrinkling

A membrane is unable to sustain load in compression, and so a membrane element under compression must be given very low stiffness in the direction of compression. This can be accomplished by modifying the elastic matrix, *D*. It is necessary to begin with a description of coordinate transformations for stresses and strains, and then move onto the definition of the modified elastic matrix for a wrinkling element, and how this is used to form the element force vector and element stiffness matrix. A new method of smoothly transitioning between the taut, uniaxial wrinkling and biaxial wrinkling stress states is derived in this work.

The stress vector can be transformed from one coordinate system to another coordinate system rotated about the *z*-axis through an angle θ, using

$$\sigma' = T\sigma \, , \tag{6.25}$$

where

$$T = \begin{bmatrix} c^2 & s^2 & 2sc \\ s^2 & c^2 & -2sc \\ -sc & sc & c^2 - s^2 \end{bmatrix} \tag{6.26}$$

and $c = \cos\theta$ and $s = \sin\theta$. To rotate the strains (or virtual strains) in this way we must use

$$\varepsilon' = RTR^{-1}\varepsilon \tag{6.27}$$

where the Reuter's matrix

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \tag{6.28}$$

The principal stresses are initially found by simply rotating the stress vector as in equation (6.25), with

$$\theta_p = \frac{1}{2}\tan^{-1}\left(\frac{2\tau_{xy}}{\sigma_x - \sigma_y}\right). \tag{6.29}$$

However, the principal stresses in a wrinkling element must be recalculated using

$$\sigma' = D_c{}' \varepsilon' + \sigma_0{}' \, , \tag{6.30}$$

where $D_c^{\prime}$ is the modified elastic matrix, given by

$$D_c^{\prime} = \begin{bmatrix} a & b & 0 \\ b & c & 0 \\ 0 & 0 & d \end{bmatrix}. \tag{6.31}$$

Remember that, in a taut element ($\sigma_2 > 0$), we set

$$a = c = \frac{E}{1-\nu^2},$$

$$b = \frac{\nu E}{1-\nu^2},$$

$$d = \frac{E(1-\nu)}{2(1-\nu^2)}. \tag{6.32}$$

For an element in uniaxial wrinkling ($\sigma_1 > 0$ and $\sigma_2 \leq 0$), we smoothly reduce *b* to zero and *c* and *d* to a small value (α) using the cos function. For an element in biaxial wrinkling ($\sigma_1 \leq 0$), we also smoothly reduce *a* to α using the cos function.

Using equations (6.18), (6.25), (6.27), and (6.30), and again eliminating δu^T, we get the element force vector in local coordinates,

$$F = \int_V \left(B_0 + AG\right)^T \left(T^{-1} D_c^{\prime} R T R^{-1} \left(B_0 u + \frac{1}{2} A\theta \right) + \sigma_0 \right) dV. \tag{6.33}$$

Note that the strains and virtual strains are unchanged. Differentiating *F* with respect to the nodal displacements gives

$$K = \frac{\partial F}{\partial u}$$

$$
\begin{aligned}
&= \int_V \left(B_0 + AG\right)^T \frac{\partial}{\partial u}\left(T^{-1}D_c{}' RTR^{-1}\left(B_0 u + \frac{1}{2}A\theta\right) + \sigma_0\right) dV \\
&\quad + \int_V \frac{\partial}{\partial u}\left(\left(B_0 + AG\right)^T\right) \times \left(T^{-1}D_c{}' RTR^{-1}\left(B_0 u + \frac{1}{2}A\theta\right) + \sigma_0\right) dV \\
&= K_e + K_g \,. \qquad (6.34)
\end{aligned}
$$

In finding K_e it is necessary to differentiate R, R^{-1}, T, T^{-1}, and $D_c{}'$ with respect to u. This is possible without too much difficulty, but the workings are not displayed here due to their length.

6.2 Loading

The load on the elements is a combination of differential pressure, material mass, and resistance forces to stop nodes from crossing into the seabed and the bulkheads. The following sections contain derivations of the differential pressure force vector, the material mass force vectors, and the corresponding load stiffnesses.

6.2.1 Load Vector

The differential pressure force vector on the deformed element is calculated in the global coordinate system using the differential pressure at the centroid of the element. The deformed element is shown in figure 6.2, with the axes of the global coordinate system being *x*,*y*,*z*. The *z*-axis is perpendicular to a flat seabed.

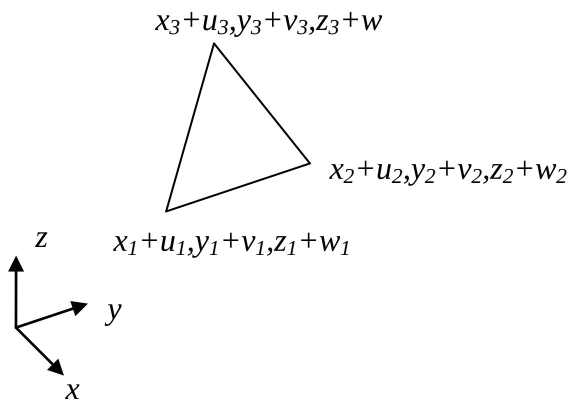


Fig. 6.2 A 3-node, 9 degree of freedom triangular element

First, the three sides of the deformed element are defined as vectors,

$$a = \left[x_{21} + u_{21} \quad y_{21} + v_{21} \quad z_{21} + w_{21}\right]^T ,$$

$$b = \left[x_{31} + u_{31} \quad y_{31} + v_{31} \quad z_{31} + w_{31}\right]^T ,$$

$$c = \left[x_{32} + u_{32} \quad y_{32} + v_{32} \quad z_{32} + w_{32}\right]^T . \qquad (6.35)$$

As before, $x_{21} = x_2 - x_1$, etc. Note that in calculating the differential pressure force across an element, the ordering of the nodes is important. In this derivation, the nodes are ordered anticlockwise when viewed from the outside of the bag, so that the vector $a \times b$ (and so the unit normal to the element, $\hat{n}$) is directed outside the bag.

The vector of differential pressure force on the element is given by

$$F_{dp} = \frac{1}{3} pA\hat{n} , \qquad (6.36)$$

where the unit normal to and area of the deformed element are

$$\hat{n} = \frac{1}{\left\| a \times b \right\|} \left[(a \times b)^T \quad (a \times b)^T \quad (a \times b)^T \right]^T \qquad (6.37)$$

and

$$A = \frac{\|a \times b\|}{2}, \tag{6.38}$$

and the differential pressure at the centroid of the deformed element is

$$p = p_0 + (\rho_w - \rho_a) g \frac{1}{3} (z_1 + w_1 + z_2 + w_2 + z_3 + w_3). \tag{6.39}$$

The density of seawater is approximately 1,025kg/m^3 and the density of the contained air is calculated using equation (3.16).

Substituting equations (6.37) and (6.38) into equation (6.36) and cancelling $\|a \times b\|$, the element differential pressure force vector becomes

$$F_{dp} = \frac{1}{6} p \left[(a \times b)^T \quad (a \times b)^T \quad (a \times b)^T \right]^T, \tag{6.40}$$

where

$$a \times b = \left[a_y b_z - a_z b_y \quad a_z b_x - a_x b_z \quad a_x b_y - a_y b_x \right]^T. \tag{6.41}$$

The membrane mass element force vector is given by

$$F_m = \frac{1}{3} \rho_m t S \left[0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \right]^T \tag{6.42}$$

where ρ_m is the density of the membrane material, and t and S are the thickness and area of the undeformed membrane element, respectively. The mass of cables can be included using

$$F_c = \frac{1}{2} \rho_c A_c L_0 \left[0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \right]^T \tag{6.43}$$

where ρ_c is the density of the cable material, and A_c and L_0 are the cross-sectional area and length of the undeformed cable element, respectively. The material mass element force vectors must be added to the element differential pressure force vector to obtain the element load vector.

6.2.2 Load Stiffness Matrix

The cable mass and membrane mass force vectors have no dependence upon nodal displacements, and so have zero load stiffness. We differentiate the element differential pressure force vector with respect to nodal displacement to obtain the element load stiffness matrix,

$$\begin{aligned} R &= \frac{\partial F}{\partial u} \\ &= \frac{1}{3}\left(A\hat{n}\frac{\partial p}{\partial u} + p\frac{\partial A\hat{n}}{\partial u} \right). \end{aligned} \tag{6.44}$$

Differentiating the differential pressure at the centroid of the element gives

$$\frac{\partial p}{\partial u} = \frac{1}{3}(\rho_w - \rho_a)g[0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1]. \tag{6.45}$$

Differentiating the product of element area and unit normal to the element gives

$$\frac{\partial A\hat{n}}{\partial u} = \frac{1}{2}\left[\left(\frac{\partial (a \times b)}{\partial u} \right)^T \quad \left(\frac{\partial (a \times b)}{\partial u} \right)^T \quad \left(\frac{\partial (a \times b)}{\partial u} \right)^T \right]^T, \tag{6.46}$$

where the differentiated cross product of *a* and *b* is

$$\frac{\partial (a \times b)}{\partial u} = \left[\left(\frac{\partial (a \times b)_x}{\partial u} \right)^T \quad \left(\frac{\partial (a \times b)_y}{\partial u} \right)^T \quad \left(\frac{\partial (a \times b)_z}{\partial u} \right)^T \right]^T. \tag{6.47}$$

The differentiated components of the cross product are given by

$$\frac{\partial (a \times b)_x}{\partial u} = [0 \quad -c_z \quad c_y \quad 0 \quad b_z \quad -b_y \quad 0 \quad -a_z \quad a_y],$$

$$\frac{\partial (a \times b)_y}{\partial u} = [c_z \quad 0 \quad -c_x \quad -b_z \quad 0 \quad b_x \quad a_z \quad 0 \quad -a_x],$$

$$\frac{\partial(a\times b)_z}{\partial u}=\begin{bmatrix}-c_y & c_x & 0 & b_y & -b_x & 0 & -a_y & a_x & 0\end{bmatrix}. \tag{6.48}$$

6.3 Solving the Equilibrium Equation

As with the axisymmetric FEA, the force equilibrium equation is solved using the Newton-Raphson method, described in sections 2.4 and 4.5, and the structure is constrained with the use of a constraints matrix, described in section 4.5.1. If the bag is rotationally symmetric with *n* identical lobes, and each lobe is symmetric about the plane cutting the lobe in half and containing the central axis of the bag, it is only necessary to model half of a single lobe; this is generally the case with balloons. When modelling the lobes of such a bag, an initial bag configuration may be found using the axisymmetric FEA. Balloon designers also take into account the symmetry of pumpkin balloons about the equator; however, Energy Bags will not necessarily be designed to be overpressurised in the way that pumpkin balloons are, and so will not be symmetrical in the same way.

It has been found that if displacement control is used then load incrementation is not required. The displacement factor used in this work is calculated to ensure that node displacements are never greater than a certain fraction (in this case 1/100th) of the distance of the furthest node from the origin. So the displacement factor is given by

$$\begin{aligned}&\gamma=\max(\|u_n\|)/(100\max(\|\Delta u_n\|)) && \text{if } \max(\|\Delta u\|)>\max(\|u_n\|)/100,\\ &\gamma=1 && \text{if } \max(\|\Delta u\|)\le\max(\|u_n\|)/100,\end{aligned} \tag{6.49}$$

where

$$\Delta u=-T^T\left(TJ_r(u_n)T^T\right)^{-1}Tr(u_n) \tag{6.50}$$

are the node displacements originally calculated by the Newton-Raphson method ($u_{n+1}=u_n+\Delta u$), $\|\Delta u_n\|$ is the vector of the magnitudes of these displacements, and $\|u_n\|$ is the vector of the distance of each node from the origin.

6.4 Results

First we tried to use the 3D FEA to model Energy Bags with the UHPV architecture, described in Chapter 5, taking advantage of the lobed bag's rotational symmetry (and the symmetry of each lobe) by only modelling half of one lobe. In particular, we tried to model the 1.8m prototype bags that have been installed in a water tank housed in one of the university's laboratories, in order to find the stresses in the tendons and bladder and assess their stability. These bags are described in more detail in Chapter 8, and the details of the bags are given in table 6.1. It should be noted that the tendons are shortened relative to the length of bladder along which they run by 3.86%, so as to ensure that the fabric isn't subjected to stresses in the meridional direction. Cables with the same cross-section and Young's modulus as the tendons were included around the top and bottom of the lobe, to ensure that the lobes remain quite flat at the top and bottom. It would be more accurate to use a constraints matrix to ensure that all nodes along the top and bottom of the lobe move as one (as if clamped into the bulkheads) but the form of such a constraints matrix was not obvious, and including stiff cables around the top and bottom serves the same purpose.

Diameter	1.8 m
Volume (excluding lobes)	2.002 m^3
Number of tendons	36
Safety factor	5
Meridional length (pole to pole)	2.36 m
Bulkhead material	Aluminium
Bulkhead diameter	0.19686 m
Bulkhead mass	5.502 kg
Tendon material	Spectra®-Nylon
Tendon width	13.5 mm
Tendon thickness	2.8 mm
Tendon Young's modulus	2.34 GPa
Tendon density	1,400 kg/m^3
Tendon failure stress	374 MPa
Carrier material	420d polyurethane coated Nylon
Carrier thickness	0.4089 mm
Carrier Young's modulus (fill direction)	0.29364 GPa
Carrier Poisson's ratio	~0.4
Carrier density	1,440 kg/m^3
Minimum carrier failure stress (fill direction)	86 MPa

Table 6.1 Specifications of the 1.8m diameter prototype

It was found that it is only possible to obtain converged solutions (where the maximum dimensionless force residual drops to less than 1×10^{-9}) for bags inflated to pressures higher than a certain threshold. This threshold, to the nearest 10kPa (0.1bar), is p_0 = 90kPa. At a range of pressures below p_0 = 90kPa (80kPa down to 40kPa inclusive, to the nearest 10kPa) the solution routine converges to an extent that the maximum node displacement becomes less than a certain amount (0.01% of the meridional length of the bag, so 0.236mm in this case), but the force residuals don't fully converge. (In many parts of the bag the force residuals become very small but towards the bottom of the bag the force residuals do not converge.) In this pressure range, the maximum dimensionless force residual is observed to end up jumping between two different values, rather than dropping continuously as it would if the solution routine was fully converging.

Once full inflation has been achieved, we calculate the von Mises stress (σ_{vM}) in each membrane element. For taut elements (i.e. no wrinkling), this is calculated in the usual way. If the continuous elastic matrix approach is being used, the von Mises stress for a wrinkling element must be handled carefully. For elements that are undergoing uniaxial wrinkling, we set the von Mises stress to be equal to the maximum principal stress (the maximum stress in such an element), and for elements that are undergoing biaxial wrinkling (i.e. slack), we set the von Mises stress to be zero.

$$
\begin{aligned}
&\sigma_{vM} = \sqrt{\sigma_1{}^2 + \sigma_2{}^2 - \sigma_1\sigma_2} && \text{if } \sigma_2 > 0 \text{ (no wrinkling)} \\
&\sigma_{vM} = \sigma_1 && \text{if } \sigma_1 > 0 \text{ and } \sigma_2 \le 0 \quad \text{(uniaxial wrinkling)} \\
&\sigma_{vM} = 0 && \text{if } \sigma_1 \le 0 \text{ (biaxial wrinkling)} \qquad (6.51)
\end{aligned}
$$

The lobe cutting pattern with von Mises stresses for the bag at p_0 = 100kPa and p_0 = 40kPa is shown in figure 6.3. Slight gaps can be seen between the top and bottom halves of the lobe, as is to be expected for a UHPV lobe. An oblique view of the full bag with p_0 = 40kPa is shown in figure 6.4.

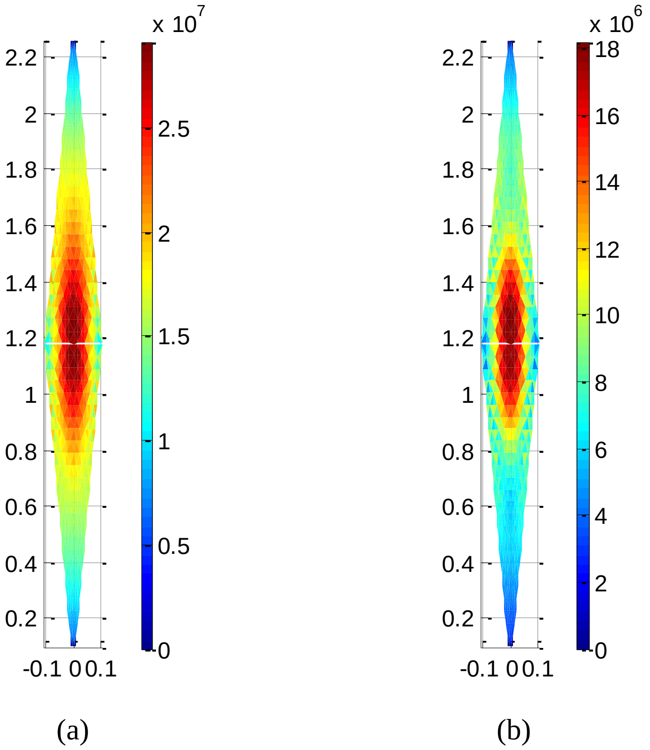


Fig. 6.3 Lobe cutting pattern for the 1.8m diameter prototype bag with von Mises membrane stresses, (a) p_0 = 100kPa, (b) p_0 = 40kPa

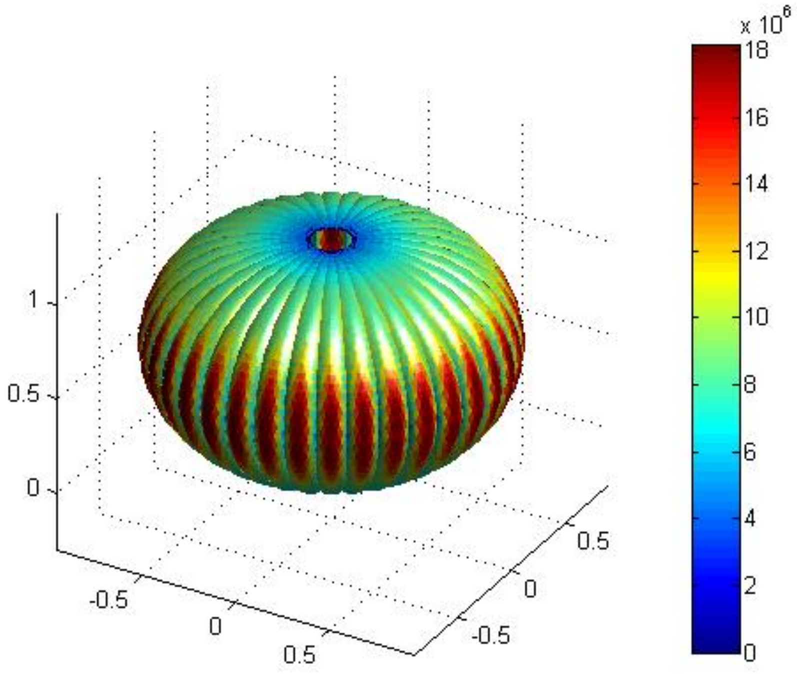


Fig. 6.4 Oblique view of the full 1.8m diameter prototype bag with p_0 = 40kPa, showing von Mises membrane stresses. Half of a single lobe was modelled and this image was formed by mirroring the deformed half-lobe and then repeating the resulting lobe

The maximum von Mises stress for the bag with p_0 = 40kPa is 18.20MPa, slightly higher than the allowable bladder stress of 86/5 = 17.2MPa (dividing the minimum bladder failure stress of 86MPa by the safety factor of 5). As noted above, the maximum node displacements do not settle to an acceptable level for pressures below about p_0 = 40kPa, however colour plots of the residuals and displacements show the slightly larger residuals and displacements to be found towards the bottom of the bag, and for a range of pressures below p_0 = 40kPa, the force residuals and node displacements reach acceptable levels in all other parts of the bag (most importantly around the areas of maximum stress). If the bag is only inflated to p_0 = 35kPa, the maximum von Mises stress is 17.27MPa, and with p_0 = 34kPa this reduces to 17.09MPa. At p_0 = 34kPa, the maximum tendon stress is 36.40MPa. With a safety factor of 5, the tendon failure stress of 374MPa reduces to an allowable tendon stress of 74.8MPa, so the tendon stress is also within acceptable levels.

In light of this work, for the 1.8m diameter prototype we would recommend a maximum differential pressure at the base of p_0 = 34kPa (0.34bar). In the tank-based Energy Bag testing, the base of the bag is beneath no less than 2m of fresh water, so the maximum gauge pressure of the contained air should be $1{,}000\times9.81\times2 + 34\times10^3$ = 53.62kPa (0.5362bar).

We now give an example of how tendon shortening reduces membrane stress while increasing tendon stress. With increased tendon shortening (4.86% rather than 3.86%), and p_0 = 40kPa, the maximum membrane stress reduces from 18.20MPa to 16.24MPa (now lower than the allowable bladder stress), and the maximum tendon stress rises from 40.40MPa to 42.42MPa (still lower than the allowable tendon stress).

It should be noted that in this work it has been assumed that the membrane is attached to the tendons, whereas in reality the membrane will be able to slide against the tendons (though frictional forces will provide some resistance to this sliding).

As well as not being able to successfully model less pressurised bags (as detailed above), we also have problems trying to model bags with increased or decreased tendon shortening. Figure 6.5 shows a side view of a single lobe after a randomly chosen iteration during the attempted solution for a lobe whose tendons are 6% shorter than the fabric along which they run. The nodes at the top and bottom of the bag do not settle into positions (as can be seen in the figure) and the force residuals do not reduce in these areas.

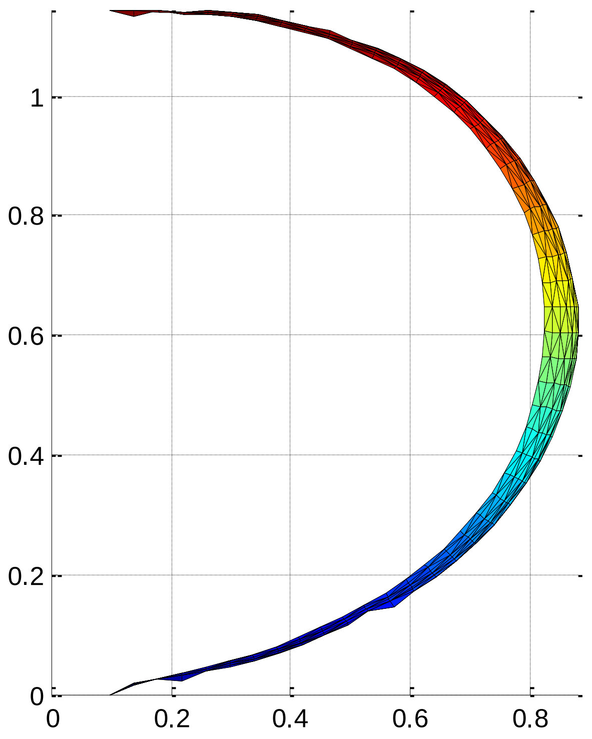


Fig. 6.5 Side view of a single lobe during the attempted solution for p_0 = 100kPa with 6% tendon shortening

It is anticipated that areas of wrinkling membrane mean that sometimes the deformed bag does not have *n*-fold rotational symmetry for certain fill levels, and the fact that it is not possible to find a converged solution while only modelling a single lobe (or half of a single lobe as we have been doing) serves to back up this belief. Part of the reason for this belief is the fact that with some bags, deformed solutions with *n*-fold rotational symmetry can only be found if the bag is highly pressurised or if the bag has low membrane stiffness, i.e. if the bag has reduced levels of wrinkling. In tank testing of prototype bags (described in Chapter 8) we find substantial wrinkling at the bottom of partially inflated bags, and the *n*-fold rotational symmetry is no longer present. In future we plan to model full bags (rather than simply half of a single lobe) to see if converged solutions can be found. The possibility of rounding error and cancellation error being the cause of the problem will also be assessed.

We now show another example of a deformed solution that can be found when only modelling half of a single lobe. Figures 6.6 and 6.7 show the undeformed and deformed configurations of a single lobe in a wide-based bag with 36 lobes, at a depth of 500m and subjected to p_0 = 10kPa. The membrane has thickness t = 1mm and is made of rubber, so has a low Young's modulus of E = 50MPa and a Poisson's ratio of v = 0.4. There is meridional reinforcement along the lobe edge of 10mm diameter steel cable (E = 200GPa), and cable and element mass is included: we use ρ_{steel} = 7,800kg/m^3, and ρ_{rubber} = 1,522kg/m^3. The bag has a base radius of 1m. Looking at the undeformed lobe in figure 6.6 it should be clear that the bag could not be laid flat on the ground after all the lobes have been welded together without significant circumferential wrinkling towards the centre of the bag.

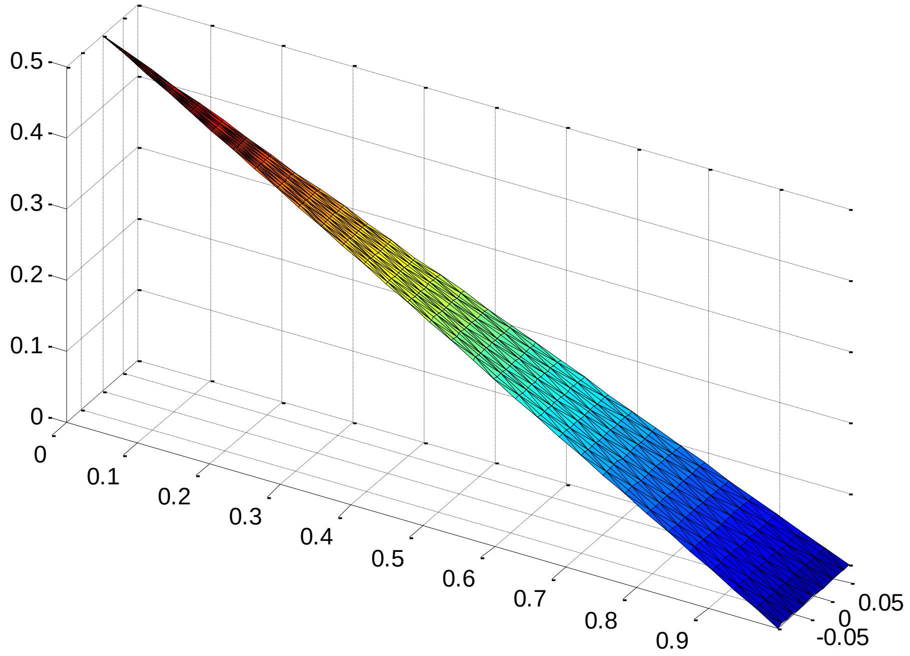


Fig. 6.6 Undeformed lobe

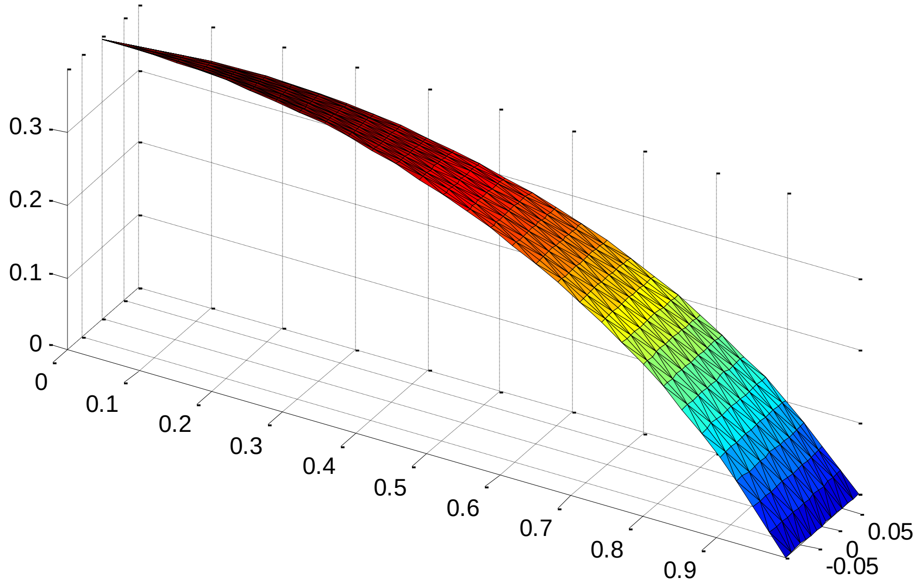


Fig. 6.7 Lobe of figure 6.6 subjected to p_0 = 10kPa

Interestingly, the mass of the concentration of cables causes the top of the bag subjected to p_0 = 10kPa to hang down slightly. This is better illustrated in figure 6.8, a cross-sectional view of a lobe with a slightly lower base pressure difference of p_0 = 7kPa. Note that to reach this low pressure, it was necessary to start with large overpressure (e.g. 1MPa as shown in figure 6.9) and then gradually reduce the pressure in stages.

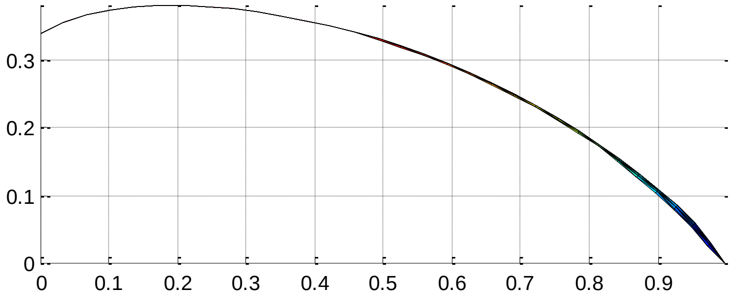


Fig. 6.8 Cross-sectional view of lobe with p_0 = 7kPa

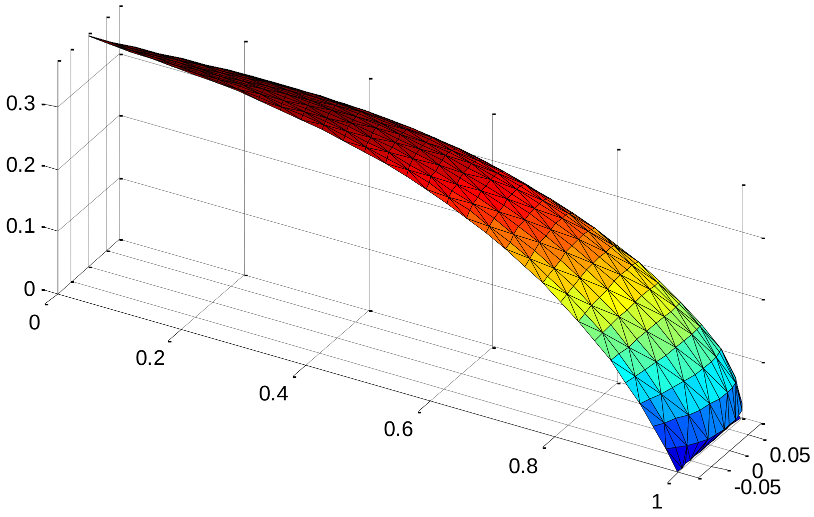


Fig. 6.9 Lobe of figure 6.6 subjected to p_0 = 1MPa

Figure 6.10 shows the same lobe with p_0 = 2.65kPa – at this low pressure, the centre of the bag is about to meet the seabed. No bulkhead mass has been included so it is simply the mass of the concentration of 36 10mm diameter steel cables and 1mm thick rubber membrane that is causing the centre of the bag to hang so low. It is not possible to find a meaningful solution for much lower air pressures without the use of seabed resistance forces.

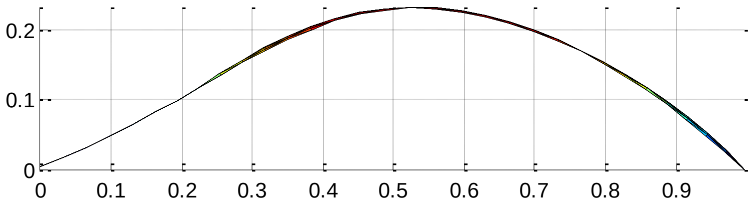


Fig. 6.10 Cross-sectional view of lobe with p_0 = 2.65kPa

Chapter 7

Optimisation

In this chapter, the analysis methods presented in the previous chapters are used in optimisation routines to minimise the cost of an Energy Bag per unit of energy stored. In section 7.1, we begin with a simple analytical optimisation of an axisymmetric natural shape bag. Section 7.2 presents a study of the optimum maximum fill level using the axisymmetric FEA, and section 7.3 presents a numerical optimisation using the axisymmetric inextensible model derived in Chapter 3. It should be noted that within this chapter we make the assumption that contained air will be expanded to atmospheric temperature and pressure and that the expansion process will be isothermal. This is a conservative approach because isothermal expansion at atmospheric temperature is the least efficient expansion process.

Essentially the energy available in a store of compressed air is the product of pressure and volume, and the most conservative calculation of the energy available would be the product of the maximum change in bag volume (so just the fully inflated volume if the bag may be emptied completely) and the minimum possible air pressure within the store. This minimum possible air pressure is the hydrostatic pressure at the greatest height above the seabed that the top of the bag will reach. A bag with mass will drop towards the seabed, and the greatest height that the top will reach is dependent upon the total mass of the bag. However, a massless bag will not drop to the seabed, and if such a bag is sealed at the base and has meridional length *L*, lower bulkhead radius r_{lo}, and upper bulkhead radius r_{up}, then the height of the top of the empty bag above the seabed is given by

$$h = L - \left(r_{lo} + r_{up}\right). \tag{7.1}$$

The absolute hydrostatic pressure at a height *h* above the seabed is given by

$$P = \rho_w g\left(d - h\right) + P_{atm}, \tag{7.2}$$

where P_{atm} is atmospheric pressure and d is the depth of the base of the bag. Therefore the minimum possible absolute air pressure in a bag is

$$P = \rho_w g\left(d - L + \left(r_{lo} + r_{up}\right)\right) + P_{atm}. \tag{7.3}$$

We should also explain a simplification made in calculating the cost of materials. For simplicity, in this chapter we assume that the cost of carrying a force over a certain distance is proportional to the force-distance product, which we call structural capacity (γ). An explanation of this simplification follows.

Cost of materials is generally proportional to the product of volume of material required (V_{reqd}) and price of the material (per unit of volume).

$$\text{Cost} \propto \text{price}.V_{\text{reqd}} \tag{7.4}$$

V_{reqd} is the distance over which the force is transmitted (x) multiplied by the required cross-sectional area of the material (A_{reqd}), and A_{reqd} is the force transmitted (F) divided by the material's yield stress (σ_y) upon the factor of safety (FoS).

$$\text{Cost} \propto \text{price}.xA_{\text{reqd}} = \text{price}\frac{xF.FoS}{\sigma_y} \tag{7.5}$$

Assuming that the price of the material scales linearly with yield stress, then

$$\text{Cost} \propto xF \ (= \gamma). \tag{7.6}$$

7.1 Analytical Cost Minimisation of a Natural Shape Bag

By considering the three components of materials cost for an axisymmetric natural shape Energy Bag (ballast, meridional reinforcement, and surface costs) and the energy stored, and then carrying out a simple optimisation, it is possible to find the bag diameter that minimises the cost of bag materials per unit of energy stored.

The volume, and so buoyancy, of a natural shape Energy Bag is proportional to any geometric dimension (e.g. maximum diameter, D_{max}) cubed.

$$F_{\text{buoyancy}} \propto D_{\max}{}^{3} \tag{7.7}$$

The cost of ballast (to hold the bag to the seabed) depends upon the mass of ballast required (and so upon the buoyancy of the bag), and so the cost of ballast (C_{ba}) must be proportional to $D_{\max}{}^{3}$.

$$C_{ba} \propto D_{\max}{}^{3} \tag{7.8}$$

In considering the cost of meridional reinforcement it is first necessary to show that, for a natural shape bag subjected to a given differential pressure at the base, the angle between the seabed and the base of the bag is independent of the bag's size. This has been effectively shown in Chapter 4 by demonstrating that the natural shape is independent of the size of the bag, but further proof is given here. The solid line in figure 7.1 is a plot of the angle β between the base of a zero pressure natural shape (ZPNS) bag with base radius $r_0 = 0$ (found using the axisymmetric FE model) and the seabed, against the meridional length of the bag, L. The angle of 40° between the seabed and the base of the bag is equal to that found in [115]. The dashed line is a plot for a ZPNS bag with base radius $r_0 = L/2$. The meridional length of the bag clearly has no effect on β in either case, which is important because it means that the tension in the meridian scales linearly with the volume of the bag. This is clear when looking at the following equation relating tension in the meridian to the buoyancy of the bag and β.

$$T = \frac{F_{\text{buoyancy}}}{\sin\beta} \tag{7.9}$$

Note that the cross-sectional area of the meridian has been scaled with L^3 (so scaled with the volume of the bag) to ensure that the strains do not become too great. If the cross-sectional area is not scaled up in this way then the lines in figure 7.1 would remain straight up to a point and then curve downwards – this is because the strains become very large if the reinforcement is not scaled up, and the stresses at the point where the line begins to curve are already greater than the yield stress of mild steel (250MPa). Figure 7.2 shows a ZPNS bag with $h(0)$ = 2m and another ZPNS bag with $h(0)$ = 10m – both bags clearly have the same shape.

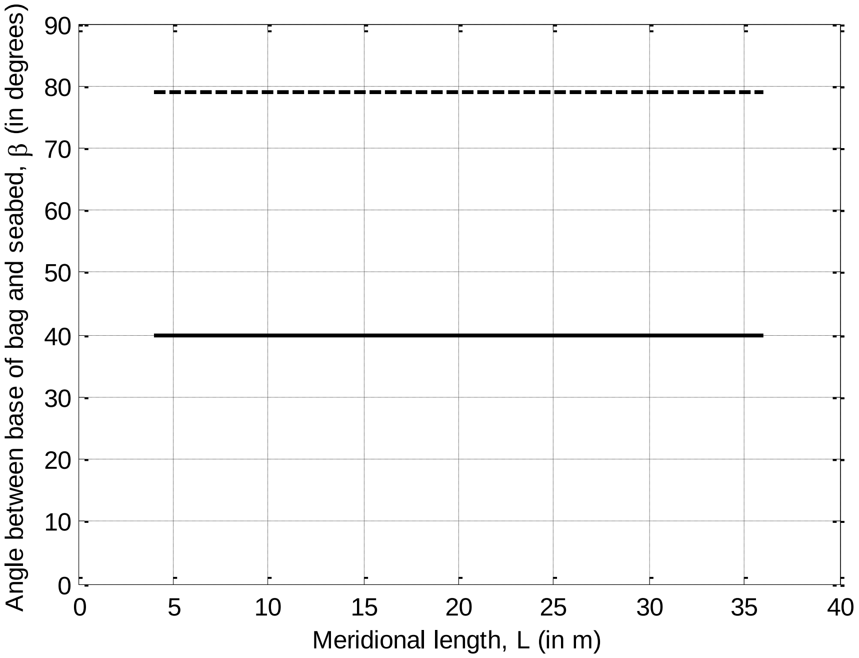


Fig. 7.1 Plot of β against L for a ZPNS bag. Solid line – base radius, r_0 = 0m; Dashed line – $r_0 = L/2$

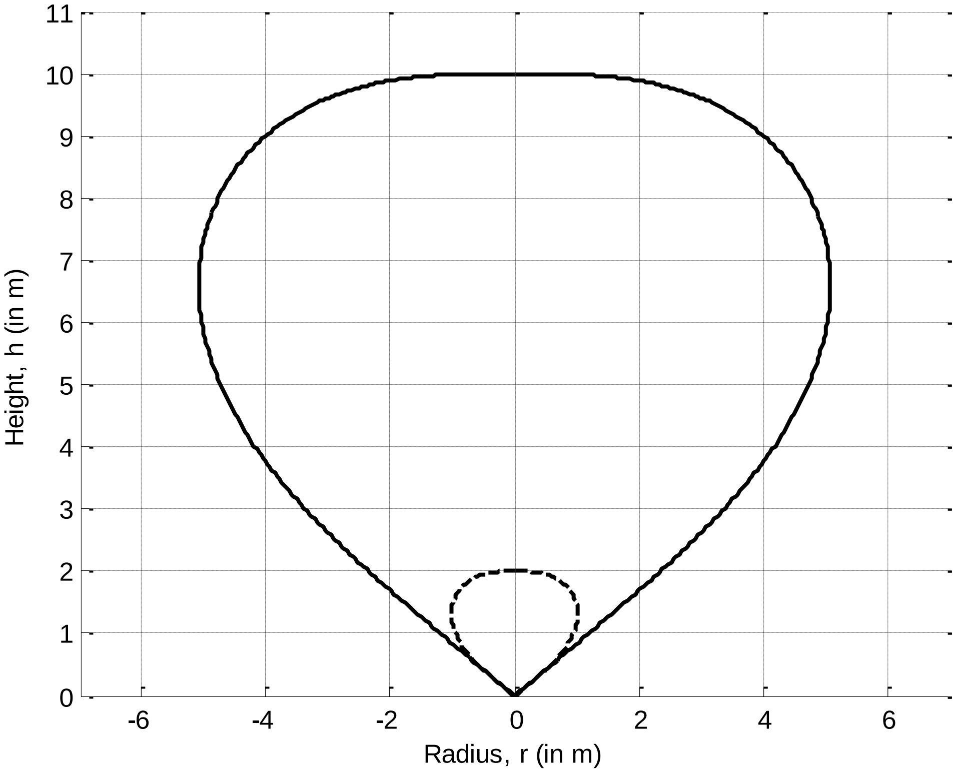


Fig. 7.2 Plot of two natural shape bags with p_0 = 0Pa; both bags have the same shape

As the meridians of a ZPNS bag always leave the seabed at the same angle (assuming the strains are reasonable), the tension in a meridian is also proportional to $D_{\max}^3$.

$$T \propto D_{\max}^{\ 3} \tag{7.10}$$

In this case, the tension is constant over the length of the meridian because there is no circumferential stress or hanging mass. The meridians each have length *L*, which is proportional to $D_{\max}$, and so the force-distance product in the meridians, and therefore cost of meridional reinforcement (C_{mr}), is proportional to $D_{\max}^4$.

$$C_{mr} \propto TL \propto D_{\max}^{\ 4} \tag{7.11}$$

This can be seen in figure 7.3: in this case, the constant of proportionality is 2.577.

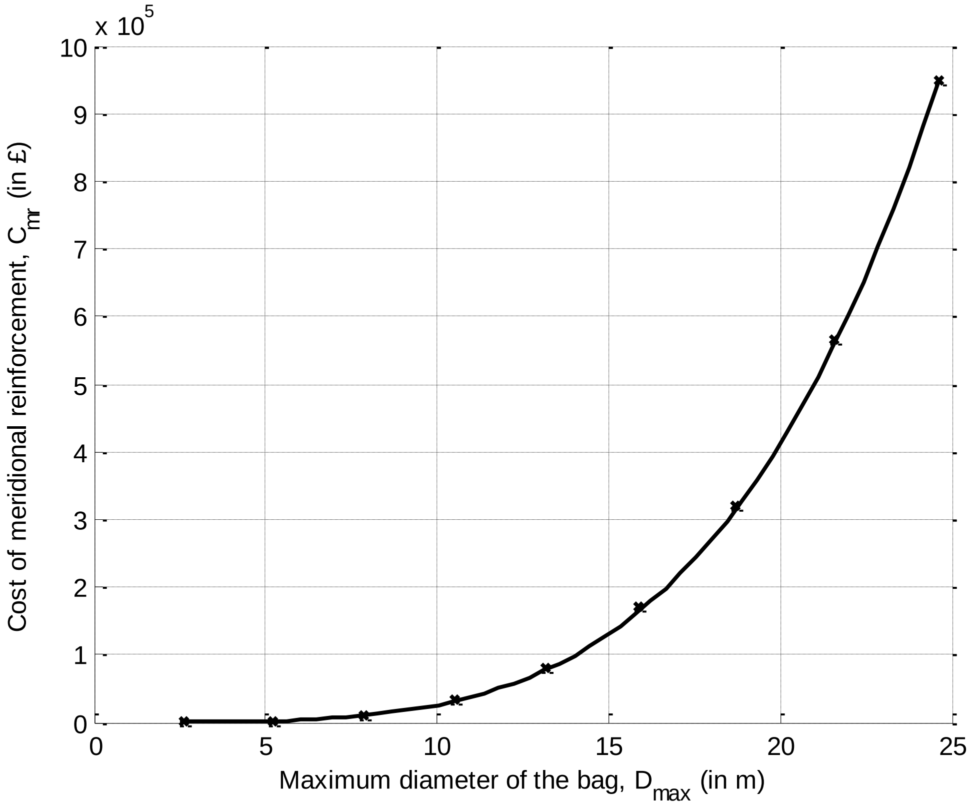


Fig. 7.3 Plot of cost of meridional reinforcement against $D_{\max}$. Solid line – C_{mr} = $2.577D_{\max}^4$; Crosses – costs found using FE model

The surface area and so cost of surface (C_{su}) is approximately proportional to $D_{\max}{}^2$.

$$A \propto D_{\max}{}^2 \tag{7.12}$$

$$C_{su} \propto D_{\max}{}^2 \tag{7.13}$$

The cost of bag materials (C_{bm}) is equal to the sum of the costs of reinforcement, surface, and ballast, where

$$C_{mr} = aD_{\max}{}^4,\ C_{su} = bD_{\max}{}^2,\text{ and } C_{ba} = cD_{\max}{}^3, \tag{7.14}$$

and a, b, and c are constants. Therefore we write

$$\begin{aligned} C_{bm} &= C_{mr} + C_{su} + C_{ba} \\ &= aD_{\max}{}^4 + bD_{\max}{}^2 + cD_{\max}{}^3 \end{aligned} \tag{7.15}$$

We are trying to minimise the cost per unit of energy stored (which is proportional to volume), where the amount of energy stored is given by

$$E = kD_{\max}{}^3, \tag{7.16}$$

and k is a constant. Therefore it is necessary to minimise

$$\frac{C_{bm}}{E} = \frac{aD_{\max}{}^4 + bD_{\max}{}^2 + cD_{\max}{}^3}{kD_{\max}{}^3}, \tag{7.17}$$

which, after simplification, is equivalent to minimising

$$\frac{C_{bm}}{E} = \frac{1}{k}\left(aD_{\max} + \frac{b}{D_{\max}} + c\right). \tag{7.18}$$

To find the bag diameter that minimises this, it must be differentiated with respect to $D_{\max}$ and set equal to zero, giving

$$\frac{1}{k}\left(a-\frac{b}{D_{\max}{}^{2}}\right)=0\,. \tag{7.19}$$

Rearranging for the optimum bag diameter we obtain

$$D^{*}_{\max} = \sqrt{\frac{b}{a}}\,. \tag{7.20}$$

Note that the asterisk indicates an optimum value. Substituting this into equation (7.14) gives equal values for optimum cost of reinforcement and optimum cost of surface.

$$C^{*}_{mr} = C^{*}_{su} = \frac{b^{2}}{a} \tag{7.21}$$

The optimum cost of ballast is given by

$$C^{*}_{ba} = c\left(\frac{b}{a}\right)^{\frac{3}{2}}. \tag{7.22}$$

It should be noted that this simple optimisation did not include costs for manufacture, transport, and installation, which are likely to increase the optimum bag diameter. On the other hand, considering the vulnerability to being holed motivates the designer to contemplate using more bags of a smaller diameter.

7.2 Optimisation of a SPA Natural Shape Bag

The cost of a bag will depend upon its maximum fill level and the meridional length of the bag. In this study we use the axisymmetric FEA, and look at each component of cost in turn. We begin by removing the cost of ballast and surface material from the equation so that we can concentrate on how the cost of required reinforcement varies with size and maximum fill level. In this study the bag is assumed to be sealed at the base, bag mass is not taken into account and bulkheads are not included.

7.2.1 Meridional Reinforcement

For a bag with meridional length *L*, and assuming isothermal expansion, the ratio of reinforcement cost to stored energy is given by

$$\frac{C_{mr}}{E} \propto \frac{TL}{PV \ln\left(P/P_{atm}\right)} \tag{7.23}$$

The total tension, *T*, in the tendons of a natural shape bag is given by

$$T = \frac{F_{buoyancy}}{\sin\beta}, \tag{7.24}$$

where β is the angle between the seabed and the tendon at the base of the bag. The buoyancy force is given by

$$F_{buoyancy} = \left(\rho_w - \rho_a\right)gV, \tag{7.25}$$

so (7.23) can be rewritten as

$$\frac{C_{mr}}{E} \propto \frac{\left(\rho_w - \rho_a\right)L}{P \ln\left(P/P_{atm}\right)\sin\beta}. \tag{7.26}$$

Note that *g* has been removed because it is a constant.

A surface plot of the function given in equation (7.26) against $-p_0/\rho_w gL$ (dimensionless fill level) and *L* for a bag anchored at 500m depth is shown in figure 7.4. We see that in order to minimise cost of meridional reinforcement per unit of energy stored, Energy Bags should be small and underpressurised at the base (for the bag with *L* = 1m the optimum maximum fill level is $-p_0/\rho_w gL$ = 0.54). Essentially, cost of meridional reinforcement per unit of energy stored increases with the amount of energy stored inside a bag.

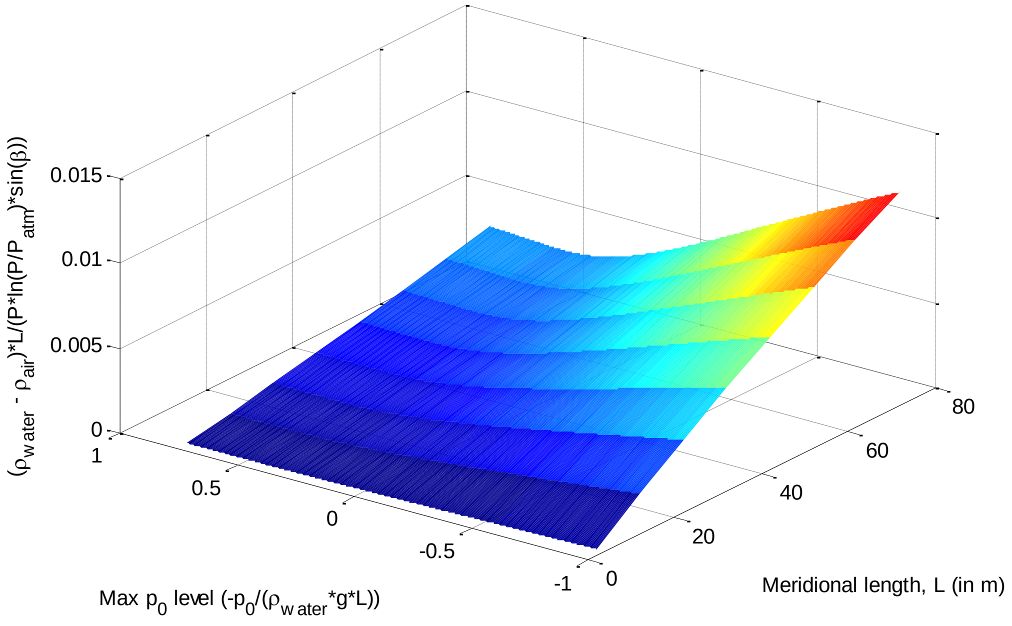


Fig. 7.4 Surface plot of a function proportional to cost of meridional reinforcement against meridional length and maximum fill level

Plots of the function given in equation (7.26) against p_0 for bags with particular meridional lengths at various depths will show the effects of depth on the optimum fill level in a natural shape bag. Figures 7.5 and 7.6 show this plot for two bags with L = 2.36m and r = 0m (figure 7.5: bag anchored at 500m depth; figure 7.6: bag anchored at 2.4m depth). Figure 7.7 shows the shape of the bags at the optimum fill levels.

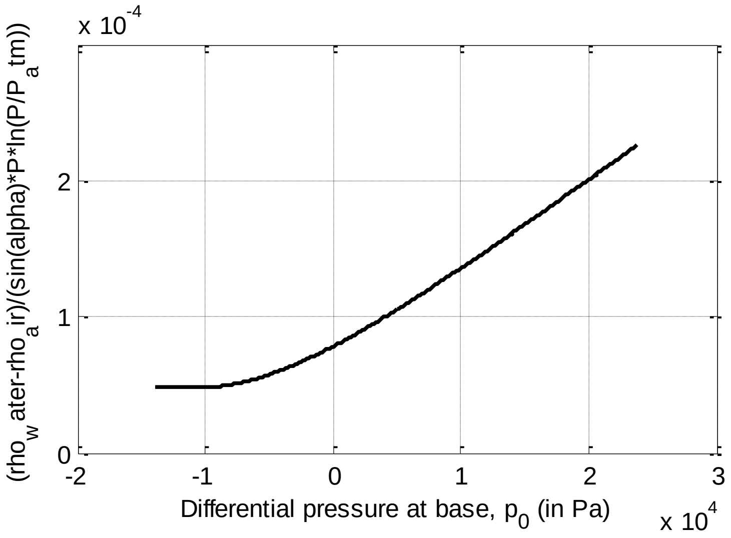


Fig. 7.5 Plot of a function proportional to cost of meridional reinforcement per unit energy stored against differential pressure across the base of the bag at full inflation. Bag anchored at 500m depth

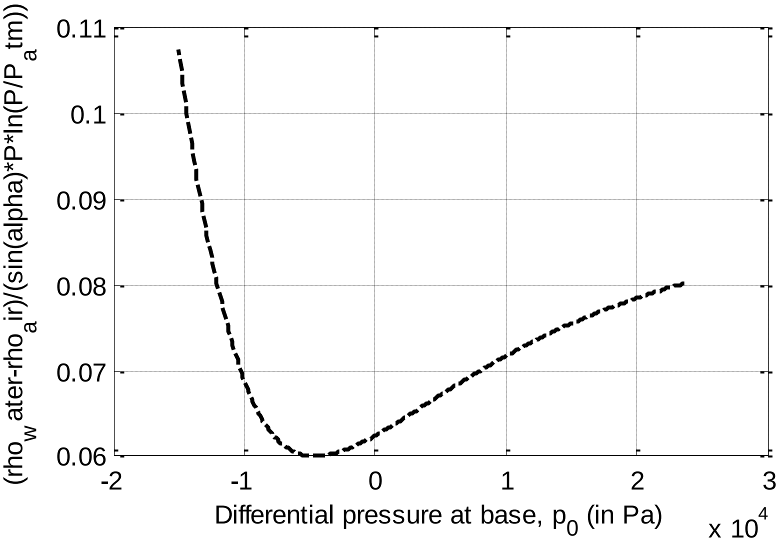


Fig. 7.6 Same plot as figure 7.5 but for a bag anchored at 2.4m depth

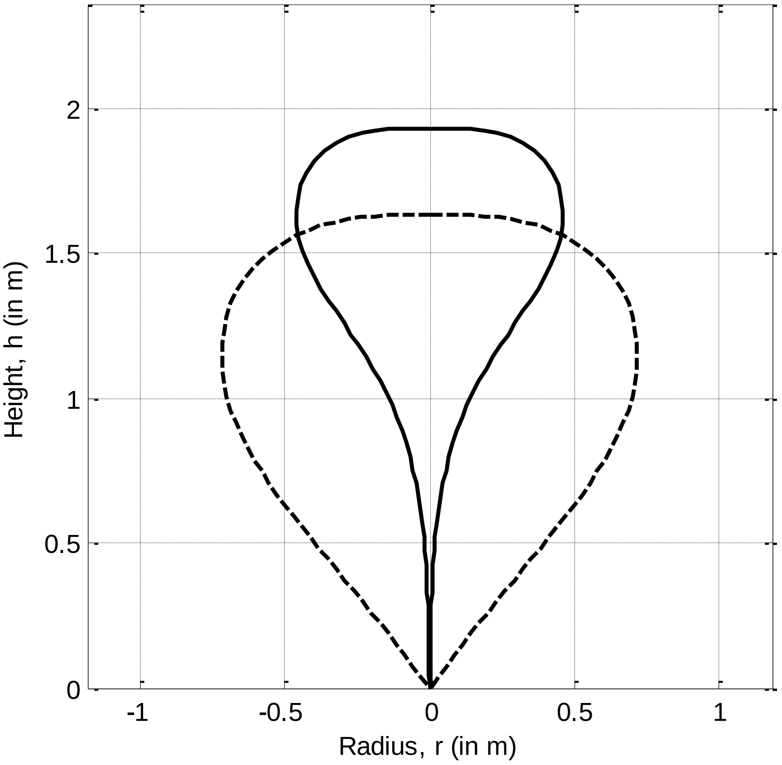


Fig. 7.7 Plot of the shape of the bags at the maximum fill level that minimises cost of meridional reinforcement per unit of energy stored. Solid line: bag anchored at 500m depth. Dashed line: bag anchored at 2.4m depth

Clearly the cost of meridional reinforcement in both bags is minimised by underpressurising the base. However in practice, the costs of installation and manufacture must be taken into account, and these may act to increase the optimum fill level, so reducing the number of bags required for a plant with given storage capacity (assuming that the cost of manufacturing and installing a bag are fairly independent of the maximum fill level of the bag). Costs of manufacture and installation depend upon the scale of the operation, as material costs do to an extent, but they are not as easy to obtain as materials costs so we press on with a study of the materials costs. It should also be noted that the effects of currents on the stresses in the fully inflated bag may also influence the optimum maximum fill level.

Though it can't be seen without zooming in, there is a minimum cost for the bag anchored at 500m depth, at p_0 = -1.2×10^{4}Pa (or a fill level of $-p_0/\rho_w g(L-r) = 0.51$). The optimum bag at this depth meets the seabed at a very sharp angle (almost 90°), minimising the tension in the tendons per unit volume (see equation (7.24)). Clearly, further reducing the maximum fill level hardly increases the cost of meridional reinforcement at all. The optimum bag at 2.4m depth enters the seabed at a shallower angle. This is because the relative change in stored energy resulting from a change in fill level is much greater at shallow depths than at greater depths, so the increase in tension in the tendons is offset by the increase in stored energy. From here on we will concentrate on the analysis of a bag anchored at 500m depth.

7.2.2 Ballast

We now look at the cost of ballast. The volume of ballast required is directly proportional to the volume of the bag, V. Therefore for a bag with a given meridional length, and assuming isothermal expansion, the ratio of ballast cost to stored energy is given by

$$\frac{C_{ba}}{E} \propto \frac{V}{PV \ln\left(P/P_{atm}\right)}, \tag{7.27}$$

so

$$\frac{C_{ba}}{E} \propto \frac{1}{P \ln\left(P/P_{atm}\right)}. \tag{7.28}$$

Clearly this ratio is independent of the shape of the bag. A plot of the ratio against the absolute pressure of the contained air is shown in figure 7.8.

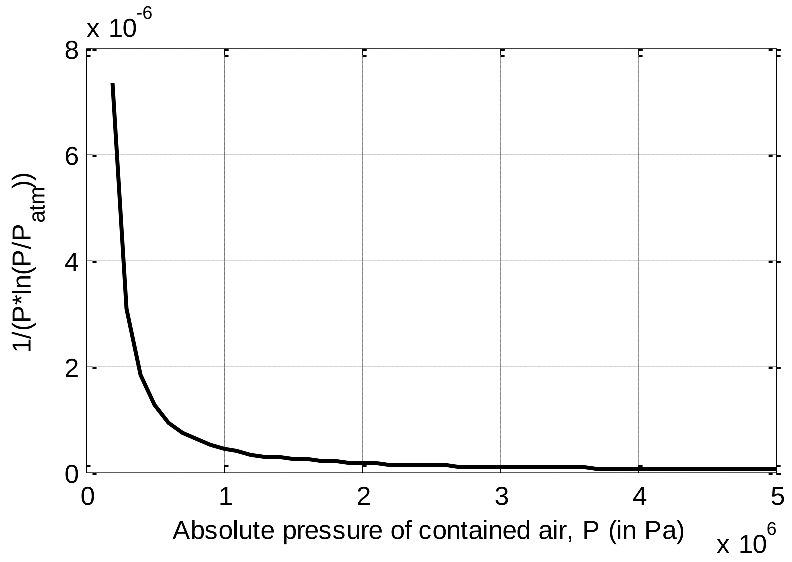


Fig. 7.8 Plot of a function proportional to the cost of ballast per unit of energy stored against absolute pressure of contained air

So the best option for minimising cost of ballast per unit of energy stored is to increase the maximum fill level, i.e. increase $p_{0,\max}$ for a bag at a given depth. Whether it is worth increasing the maximum fill level in order to reduce the cost of ballast per unit of energy stored will ultimately depend upon the effects of maximum fill level on the costs of meridional reinforcement and membrane.

7.2.3 Surface

Finally we look at the cost of surface material. First we should note that while bags with separate lobes can be tailored to provide the optimum lobe shape, cutting separate lobes for such bags may in fact require more material usage (including waste) than a bag comprising a fewer number of separate pieces of fabric (e.g. one piece for more than one lobe, or even just two circles of fabric forming the bag as in the prototypes). The total material required depends upon the width and length of the roll of fabric that is used, but for now we assume that no material is wasted.

In Chapter 5 we showed how to generate a constant tension (CT) lobe cutting pattern that has the minimum possible circumferential tension in the lobe, and has constant circumferential tension at all points along the lobe so making the best use of materials. We assume that the cost of surface material is proportional to the product of maximum circumferential tension and the distance along which the tension is transmitted (*x*), integrated along the lobe. In the axisymmetric FE model, the circumferential lobe tension in an element is given by T_cL, where T_c is the circumferential tension per unit meridional length and *L* is the length of the element. Therefore

$$C_{su} \propto \max(T_c L) \sum x . \tag{7.29}$$

The distance along which the circumferential tension is transmitted is given by *wn*, where *w* is the width of the lobe cutting pattern (as shown in figure 5.1 and given in equation (5.1)). *w* and T_c are found for CT lobes of various meridional lengths and maximum fill levels using the procedure given in Chapter 5, and the function proportional to C_{su} (per unit of energy stored) is plotted against $-p_0/\rho_w gL$ and *L* in figure 7.9. (Note that the axes are the other way round relative to those in figure 7.4.) Importantly, the number of lobes is kept proportional to the maximum radius of the bag at each fill level.

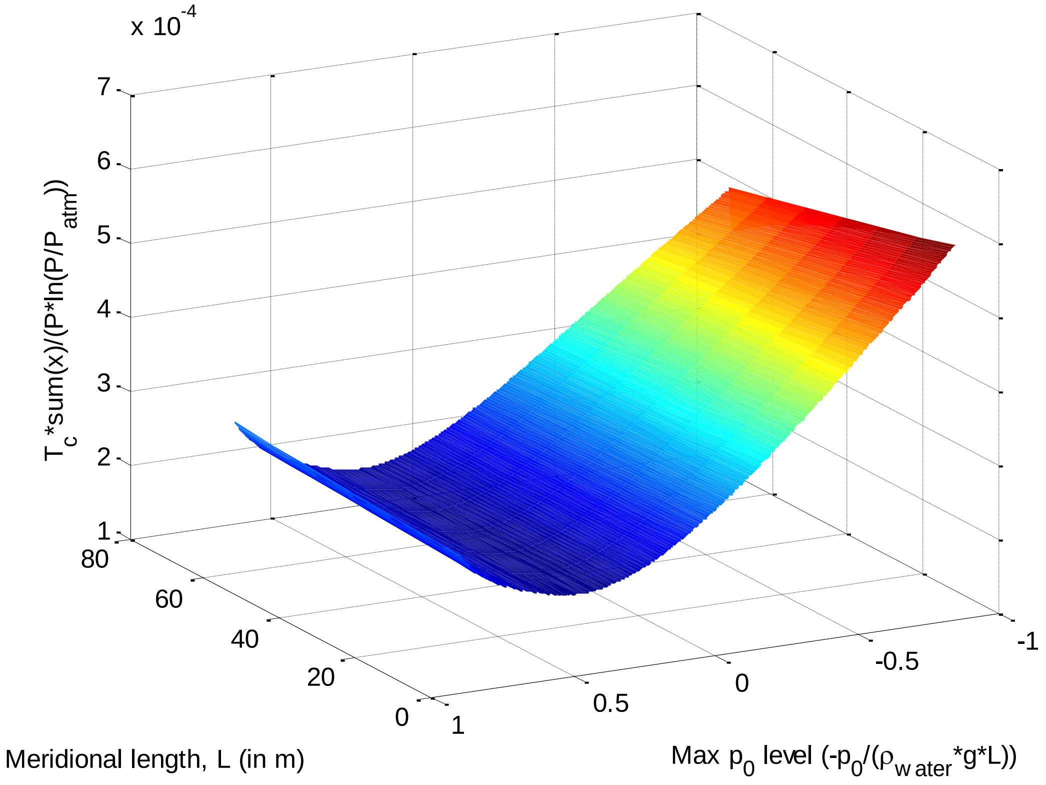


Fig. 7.9 Plot of a function proportional to cost of surface per unit of energy stored against meridional length and maximum fill level for a bag at 500m depth

We see that if the number of lobes is kept proportional to the maximum radius of the bag, the bag (with CT lobes) that minimises cost of surface is underpressurised at the base and is as small as possible. Though the lowest surface cost is obtained by making the bag small, there is little difference in surface cost between a small bag and a large bag. The maximum fill level has much more of an effect on cost, and the lowest surface cost is obtained by designing the bag such that maximum contained air pressure is equal to the surrounding water pressure at a height of 0.35*L* above the seabed (assuming no bulkheads).

From Chapter 5, we know that circumferential tension (per unit meridional length) in a CA lobe with subtended angle of π rads is given by

$$T_c = pr_b \sin(\pi/n) \tag{7.30}$$

where r_b is the local bag radius, p is the local differential pressure, and n is the number of lobes. We also know that the circumferential tension in an Energy Bag can be made independent of the size of the bag if we make the number of tendons proportional to the bag radius squared. It should be noted that changing the number of tendons will have little effect on the cost of meridional reinforcement, which is simply proportional to the bag volume and the angle at which the tendons leave the seabed.

Again, the cost of surface material is proportional to the product of maximum circumferential tension and distance along which the tension is transmitted (x). For an element in the axisymmetric FE model, the circumferential lobe tension is given by T_cL, where T_c is the circumferential tension per unit meridional length and L is the length of the element. So

$$C_{su} \propto \max(T_c L)\sum x. \tag{7.31}$$

For the lobe with the lowest possible circumferential tension (a constant angle lobe with constant subtended angle of π radians), T_c is given in equation (7.30) and the cutting pattern width is given by

$$w = \pi r_b \sin(\pi/n). \tag{7.32}$$

Therefore

$$\max(T_c L) = \max(r_b pL \sin(\pi/n)), \tag{7.33}$$

$$x = wn = \pi r_b n \sin(\pi/n). \tag{7.34}$$

The cost of surface is given by

$$C_{su} \propto \max(T_c L)\sum x \propto \pi n \sin^2(\pi/n)\max(r_b pL)\sum r_b\ . \tag{7.35}$$

For a large number of lobes we can take small angle approximations, and see that, for a bag with a given size and fill level (i.e. if r_b and p at any given height in the bag are constant and independent of n), the cost of surface is almost exactly inversely proportional to the number of lobes; a plot of $\pi n \sin^2(\pi/n)$ against n is shown in figure 7.10.

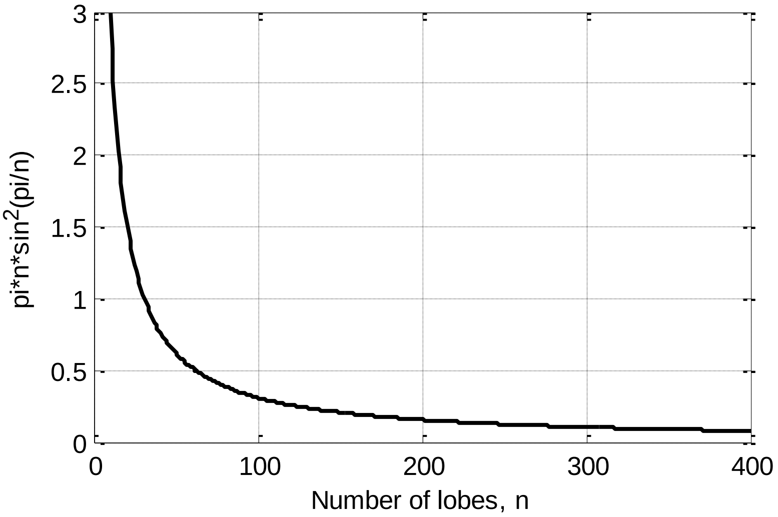


Fig. 7.10 Plot of a function proportional to cost of surface against the number of lobes for a bag with a given size and fill level

7.2.4 Summary

To summarise, we have found that the costs (per unit of energy stored) of meridional reinforcement (tendons) and surface material (in constant tension lobes and assuming no wastage of material) in a sealed Energy Bag are reduced if the maximum pressure of the contained air is less than the hydrostatic pressure at the base of the bag, i.e. if the bag is underpressurised at the base (p_0 < 0). For a bag anchored at 500m depth, the cost of meridional reinforcement is minimised if the maximum pressure of the contained air is equal to the hydrostatic pressure at a height of approximately 0.5*L* above the seabed (where *L* is the meridional length of the bag), and the cost of surface material is minimised if the maximum pressure of the contained air is equal to the hydrostatic pressure at a height of approximately 0.35*L* above the seabed. However, the cost of ballast per unit of energy stored is increased as the stored air pressure is reduced. We have also found that, for a bag with a given size and fill level, the cost of surface is inversely proportional to the number of lobes in the bag, while the costs of meridional reinforcement and ballast remain approximately independent of the number of lobes.

We should also remember that the pressure-volume characteristic of the store will affect the cost of the expansion machinery. In Chapter 4 we found that the PV curve for an axisymmetric SPA natural shape bag remains relatively flat up to a small overpressure and then becomes much steeper with increased volume (see figures 4.10 and 4.17), indicating another advantage to limiting the maximum contained air pressure to around the hydrostatic pressure at the base of the bag: more efficient expansion machinery can be used because the air in the bag remains at a reasonably constant pressure at all fill levels.

Therefore, the optimum single point of anchorage natural shape bag should have as many lobes as possible, and if costs for manufacture and installation are not taken into account and the bag has a sealed base, the optimum bag should be designed to have a maximum contained air pressure equal to the hydrostatic pressure at a height somewhere between about 0.35*L* and 0.54*L* above the seabed. Bag cost would be further reduced by making the bag as small as possible. However, the optimum size and maximum fill level depend upon the relative prices of surface, meridional reinforcement, ballast and manufacture and installation. The costs of manufacture and installation will cause the optimum size of the bag to be greater than zero.

Since balloon designers have found that the buckling pressure of superpressure pumpkin balloons decreases as the number of lobes and the bulge formed by the lobes are increased, in the future we should check if Energy Bag designs could also be unstable. Unfortunately, increasing the number of lobes and the bulge formed by the lobes are both methods of decreasing the circumferential stress in the fabric.

7.3 Optimisation of a Wide Base Natural Shape Bag

In this section we carry out an optimisation of natural shape bags that may have wide bases. The axisymmetric ODEs method presented in Chapter 3 is used (because the optimisation was carried out before the axisymmetric FE method was developed), and some extra coupled ODEs are derived that must be solved to calculate costs and the amount of energy stored. Circumferential reinforcement and distributed hanging masses are included, as are the effects of setting a maximum fill level.

7.3.1 Description of the Method

We seek to minimise the objective function, $f(x) =$cost of a bag/stored energy, subject to inequality and equality constraints of the form $g(x) \leq 0$ and $h(x) = 0$. While the total cost of a bag will depend upon the cost of manufacture and deployment, these costs are not taken into account in this study; only the cost of materials is included. As before, this cost is broken down into the sum of the cost of reinforcement, the cost of surface, and the cost of ballast. To calculate these costs in the ODEs method, it is necessary to derive some extra quantities. The volume of air contained (*V*) is required because the stored energy in the bag depends upon *V*. As justified at the start of this chapter, for simplicity we assume that the cost of carrying a force over a certain distance is proportional to the force-distance product, which we call structural capacity (γ). The cost of the bag materials depends upon the surface area (*A*), and the required structural capacities (meridional, circumferential, and vertical). These are found by simultaneously solving five more differential equations alongside (3.1), (3.2), (3.9), (3.12), and taking the final values.

$$\frac{dV}{ds} = 2\pi rh\cos\alpha \tag{7.36}$$

$$\frac{dA}{ds} = 2\pi r \tag{7.37}$$

$$\frac{d\gamma_m}{ds} = 2\pi T_m \tag{7.38}$$

$$\frac{d\gamma_c}{ds} = 2\pi r T_c \tag{7.39}$$

$$\frac{d\gamma_w}{ds} = \pi r h n_w \cos\alpha \tag{7.40}$$

In equation (7.40), the structural capacity in the vertical direction (γ_w) has been divided by 2: this is done because, in suspending a series of masses evenly along a cable hanging between the membrane and the base of the bag, it is possible to use weaker/thinner (and so cheaper) cable lower down in the series of masses, and the average tension in the cable is only half the tension at the very top of the cable (assuming an equal distribution of mass along the cable).

Equations (3.1), (3.2), (3.9), (3.12), and (7.36)-(7.40) are integrated numerically, in this case as an initial value problem. We always set $V(0)$, $A(0)$, $\gamma_m(0)$, $\gamma_c(0)$ and $\gamma_w(0)$ to zero (0 in brackets indicating an initial value at the top of the bag). An upper bulkhead at the top of the bag requires nonzero $r(0)$ (to represent the radius of the bulkhead) and nonzero $\alpha(0)$ (so the weight of the bulkhead is reacted by the tension in the meridian), but no bulkheads are used here and so $r(0)$ and $\alpha(0)$ are also set to zero. Once again, the average density of seawater, ρ_w = 1,025kg/m^3, and the density of the compressed air is calculated using the equation of state for an ideal gas, as shown in equation (3.16).

The cost of reinforcement is the sum of each of the meridional, circumferential, and vertical structural capacities multiplied by the price per unit structural capacity of each. The cost of surface is the surface area multiplied by the price of surface per unit area. The cost of ballast is equal to the mass of ballast required multiplied by the price of ballast per unit mass (p_{bal}). The mass of ballast required is given by the product of required ballast volume (V_{bal}) and the density of the ballast material (ρ_{bal}), so

$$C_{bal} = p_{bal} \times \rho_{bal} V_{bal}\,. \tag{7.41}$$

If we ignore the mass of the surface and reinforcements, the required ballast volume is found by balancing the net buoyancy of the bag, given by

$$\left(\rho_w - \rho_a\right) g V_{bag} \tag{7.42}$$

and the difference between the weight of the ballast and the buoyancy of the ballast, given by

$$(\rho_{bal} - \rho_w) g V_{bal} . \tag{7.43}$$

Setting these equal, rearranging for V_{bal} and cancelling g, we obtain

$$V_{bal} = V_{bag} \frac{(\rho_w - \rho_a)}{\rho_{bal} - \rho_w} . \tag{7.44}$$

This is substituted into (7.41) to find the cost of ballast.

The energy available in the compressed air store depends upon how the compressed air is expanded. It is conservatively assumed that the air is an ideal gas and will be expanded isothermally. The work done in the isothermal expansion from stored volume V_A (with absolute pressure P_A) to volume V_B (with absolute pressure P_B – in our case, atmospheric pressure) is

$$W_{A \to B} = \int_{V_A}^{V_B} P dV = \int_{V_A}^{V_B} \frac{nRT}{V} dV = nRT \ln \frac{V_B}{V_A} . \tag{7.45}$$

For an ideal gas, the product PV remains unchanged in an isothermal process, so $\frac{V_B}{V_A} = \frac{P_A}{P_B}$ and

$$W_{A \to B} = P_A V_A \ln \frac{P_A}{P_B} . \tag{7.46}$$

Exterior penalty functions [123] are used to transform the constrained problem into a single unconstrained problem. In order to minimise $f(x)$ subject to the constraints $g_i(x) \le 0$ for $i = 1, ..., l$ and $h_i(x) = 0$ for $i = 1, ..., m$, we minimise the auxiliary function $f(x) + \mu\alpha(x)$, where μ is a large positive penalty parameter and α is a penalty function that is zero for feasible points and increasingly positive for increasingly infeasible points. A suitable form for α is

$$\alpha(x) = \sum_{i=1}^{l} [\max\{0, g_i(x)\}]^2 + \sum_{i=1}^{m} |h_i(x)|^2 . \tag{7.47}$$

By squaring both terms, differentiability is ensured at both $g(x)=0$ and $h(x)=0$. It is possible to get arbitrarily close to a minimum of $f(x)$ by minimising the auxiliary function for a sufficiently large μ. In this work an upward turn of the meridian and compressive stresses are both penalised, and in future a maximum stress constraint could also be imposed using penalty functions. As explained in Chapter 3, an upward turn of the meridian leads to undesired looping and very often to a solution which does not reach the seabed (i.e. does not reach $h = 0$).

For a given p_0, energy density increases with depth while the forces on the bag (and so materials costs) remain almost unchanged. Therefore the objective function (materials costs only) will decrease with depth. It would be hard to put costs to the extra problems that installing bags at great depths would bring (e.g. installation and piping) without further research, so the effect of depth is not studied here.

The objective function is minimised over $h(0)$, V, T_c, n_w, and p_0 using a tiered line search. This is a multidimensional optimisation procedure in which a local stationary point is found by simply tiering line searches, so minimising a series of minima. As an example, in minimising the objective function $f(x_1, x_2)$, we would perform a line search in x_1 for a set value of x_2 to find the minimising x_1 (call this minimising value $\bar{x}_1$) for the certain x_2. $\bar{x}_1, \bar{x}_2$ is found by performing a line search adjusting x_2 (and finding $\bar{x}_1$ for each value of x_2 tried) until the minimum of $f(\bar{x}_1, x_2)$ is found. $\bar{x}_1, \bar{x}_2$ is then a local minimum of $f(x_1, x_2)$. To minimise for more variables, another tier of the same procedure is added. Though the bag volume cannot be set directly, the meridional stress at the top of the bag ($T_m(0)$) which gives a shape that encloses a certain required volume can be found using a root finding algorithm. All of the one-dimensional line searches are carried out using Brent’s method: a combination of the golden section search and parabolic interpolation [124]. Other multidimensional search procedures which could be used include Rosenbrock’s method and methods that use derivatives in determining the search direction, such as the method of steepest descent and the method of feasible directions [125], as used by Pagitz and Pellegrino in their cutting pattern optimisation of lobed superpressure pumpkin balloons [104].

7.3.2 Results

In the following analyses, all of the meridional, circumferential, and vertical (hanging cable) stresses are carried through steel at its yield strength (250MPa), with an estimate at the cost of steel of £0.5/kg. The cost of surface is estimated at £2/m^2 and the cost of ballast estimated at £4 per ton of net weight (after subtracting the ballast's buoyancy). Of course all of the optimum shapes are sensitive to these costs. The optima are all found for a depth of 500m.

Zero Pressure Natural Shape

An optimum ZPNS bag was found and is shown in figure 7.11. It stores 289m^3 of air compressed to an absolute pressure of 51.28bar, which is 1.62MWh of energy. The value of the objective function at this optimum is £1,104/MWh. As mentioned before, this figure only accounts for the costs of reinforcement, surface, and ballast materials.

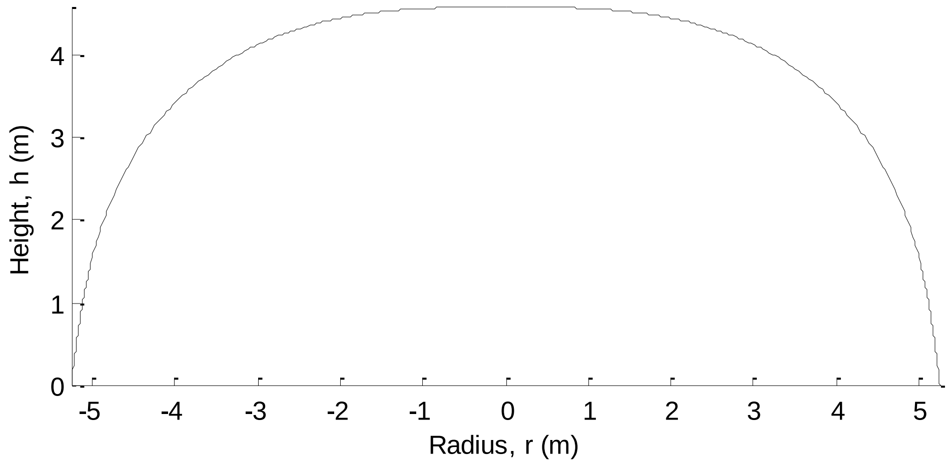


Fig. 7.11 Optimum zero pressure natural shape bag at 500m depth

It is interesting to note that the optimum ZPNS bag has quite a wide base radius. This is because the differential pressure at the top of a wide, low bag is not as great as it would be at the top of a taller bag with the same p_0. The sides of the bag meet the seabed at angles close to 90° because, as shown in equation (7.24), the smallest possible meridional tension for a given stored volume is that which gives an entry angle of 90° (assuming no ballast mass hangs from the inside of the bag and no component of the bag materials has mass). At angles away from 90°, larger meridional tension is necessary to balance the buoyancy force.

Zero Pressure, Nonzero Circumferential Stress and Hanging Ballast

The effects of constant nonzero circumferential stress and constant nonzero hanging ballast on a zero pressure bag are now included in the optimisation, and the optimum bag is shown in figure 7.12. This has zero circumferential stress and hanging ballast of 32kN/m^2, and costs £962/MWh; a 13% reduction in cost when compared to the ZPNS bag with no hanging masses. It stores 1,314m^3 of air compressed to an absolute pressure of 51.28bar, which is 7.36MWh of energy. The inclusion of positive circumferential stress increases the value of the objective function and so as compressive circumferential stress is not an option, the optimum circumferential stress is zero.

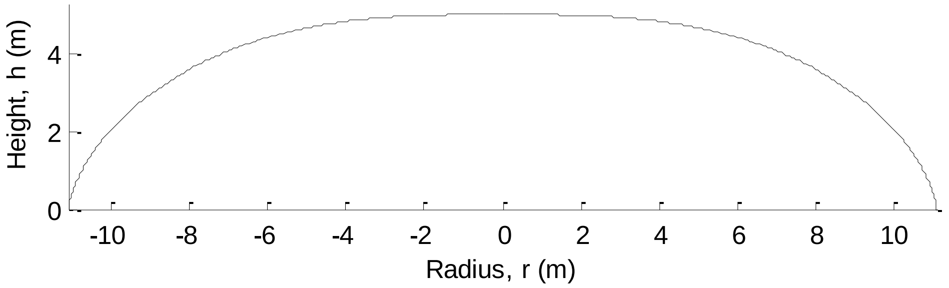


Fig. 7.12 Optimum zero pressure bag with nonzero hanging mass at 500m depth

It is clear that the inclusion of hanging masses has led to an optimum bag shape which has a greater stored volume and lower profile than the ZPNS bag. Hanging mass counteracts the differential pressure force, particularly in horizontal sections (low α) like at the top of the bag, and so the required net restoring force is reduced. This net restoring force is balanced by curvature of the membrane, and so the required curvature is reduced, and the bag can be wider and store more energy for a given centre height (and so given maximum differential pressure). However, it should be noted that the model does not yet account for the volume taken up by the hanging masses. Once again the sides of the bag meet the seabed at approximately right-angles. It is anticipated that hanging masses will flatten the PV curve, allowing more efficient turbomachinery to be used.

Nonzero Pressure, Nonzero Circumferential Stress and Hanging Ballast

Now the effect of nonzero p_0 in a sealed base bag is included. If the base of the bag is sealed, differential pressure at the base may be positive (like a superpressure balloon) or negative (subpressure), though subpressure balloons have only been considered when studying partial inflation and ascent shapes [96],[126]. The optimum bag is shown in figure 7.13. This is a subpressure bag with p_0 = -32kPa, zero circumferential stress, and hanging ballast of 10kN/m^2, and costs £907 per MWh. It stores 2,249m^3 of air compressed to an absolute pressure of 50.96bar, which is 12.52MWh of energy.

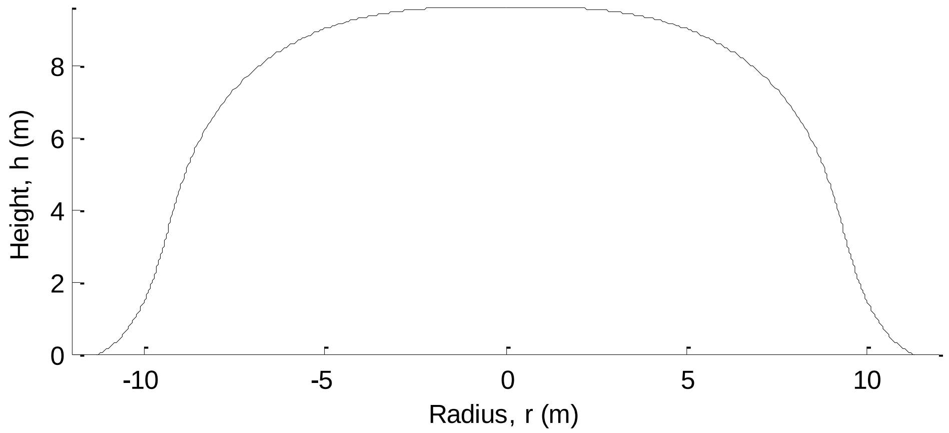


Fig. 7.13 Optimum nonzero pressure bag with nonzero hanging mass at 500m depth

With the inclusion of nonzero differential pressure at base, the base of the optimum bag no longer meets the seabed at right angles because it must take on negative curvature at the base to balance the negative pressure gradient whilst maintaining a fairly low centre height (and so maximum differential pressure).

7.4 Specification for a 0.2GWh Energy Bag

Assuming isothermal expansion, the volume required to store 0.2GWh of energy can be found by rearranging

$$E = PV\ln\left(P/P_{atm}\right) \tag{7.48}$$

to get

$$V = \frac{E}{P\ln\left(P/P_{atm}\right)}, \tag{7.49}$$

where E is the energy stored, P_{atm} is atmospheric pressure (101.325kPa), and P is the absolute pressure of the compressed gas, given by

$$P = p_0 + \rho_w g d + P_{atm} \qquad (7.50)$$

where p_0 is the differential pressure at the base of the bag, ρ_w is the density of the surrounding seawater (1,025kg/m^3), g is standard gravity (9.81m/s^2), and d is the depth of the base of the bag. We will assume that the bag is anchored at 500m depth (so d = 500m) and has a maximum of zero differential pressure at the base (so p_0 = 0Pa when fully inflated). Using equations (7.49) and (7.50), we calculate that the maximum volume of air required to store 0.2GWh of energy in a zero pressure bag anchored at 500m depth is 35,705m^3 (required volume would be less if some heat storage were incorporated).

From earlier work, we know that the angle β between the base of a zero pressure natural shape (ZPNS) bag and the seabed is approximately 40°. Using equations (7.24), (7.25), (3.16) and (3.17), we can calculate the total tension in the meridians of a 35,705m^3 ZPNS bag.

$$\begin{aligned} T_{tot} &= \frac{(\rho_w - \rho_a)gV}{\sin\beta} \\ &= \frac{(1025 - MP/RT)\times 9.81 \times 35705}{\sin(40\pi/180)} \\ &= 5.235\times10^8\text{N} \end{aligned} \qquad (7.51)$$

The required tendon diameters (assuming the tendon is steel with σ_y = 400MPa and Factor of Safety = 2) for ZPNS bags with various numbers of tendons are given in table 7.1.

Number of tendons	Diameter (in mm)
100	182.6
200	129.1
300	105.4
400	91.28
500	81.64

Table 7.1 Required tendon diameter for various numbers of tendons (tendons stressed at 200MPa)

The ZPNS bag that will store 0.2GWh of energy is found using the axisymmetric FEA, and shown in figure 7.14. The mass of steel tendons (ρ = 7,800kg/m^3) is 1,304 tons, almost 4% of the total required ballast mass, and the mass of membrane (assuming 1mm thick material with density of rubber, ρ = 1,522kg/m^3) is 1.3 tons (assuming no excess material is included for lobing or further inflation) – approximately 1/1,000th of the total mass of the tendons. Excess membrane material would be included for lobes, but as membrane mass/cost is such a small proportion of the total bag cost, this is not included here.

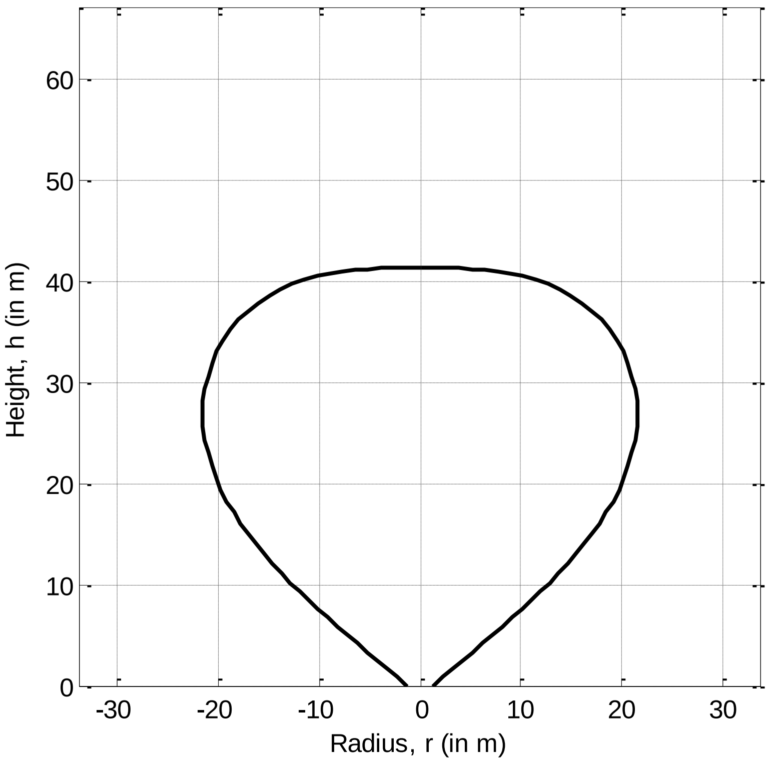


Fig. 7.14 Cross-section of a 0.2GWh ZPNS Energy Bag

This bag has a meridional length of 64m and lower bulkhead radius of 1.3m – this bulkhead radius was chosen by scaling up the size of the lower bulkhead used in the 1.8m diameter prototypes. It is 41.3m tall at full inflation with a maximum diameter of 43m. In calculating the costs of the different components of the bag, we assume the following prices:

- Membrane material costs £10/m^2
- Meridional reinforcement costs £2/kg (steel stressed at 200MPa with ρ = 7,800kg/m^3)
- Extra ballast costs £40 per ton.

Clearly these prices are higher than those used in section 7.3, and were initial prices suggested by Maxim De Jong of Thin Red Line Aerospace. The materials costs break down as follows:

- Membrane - £53,300
- Reinforcement - £2.36m
- Extra ballast - £1.37m

- Total materials cost - £3.78m.

The total cost of materials per unit of energy stored is approximately £18,900/MWh. Note that the cost of meridional reinforcement is approximately 44 times the cost of surface, but it has been found in section 7.1 that the optimum-sized bag will have cost of surface equal to cost of meridional reinforcement. That previous work neglected the mass of membrane and tendons but it remains quite valid because it has been found that in single point of anchorage bags (as this bag almost is), material mass does not have a significant effect on the bag's shape (see figure 7.15) or the cost of ballast mass (~4% reduction).

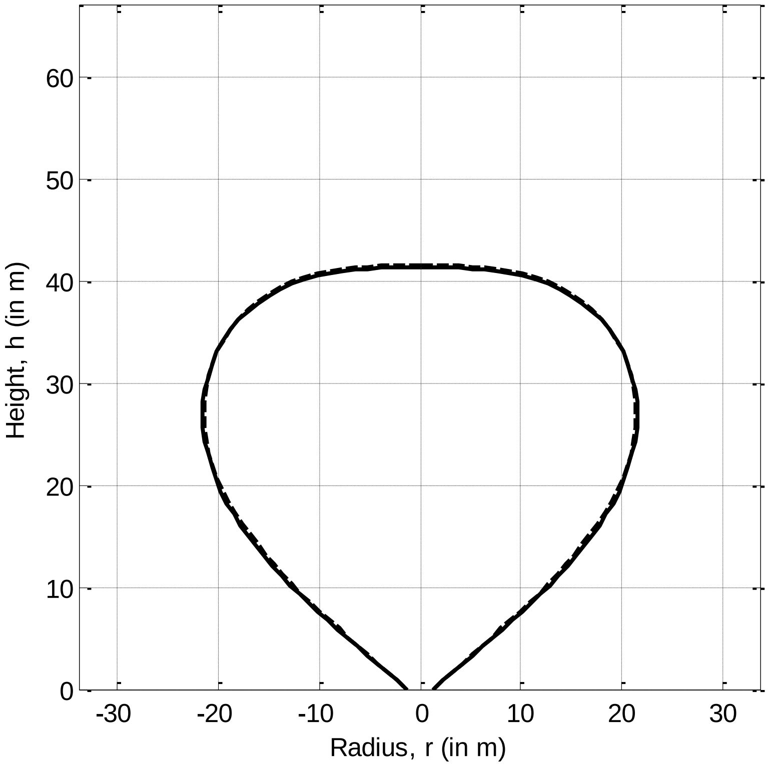


Fig. 7.15 Effect of material mass on the shape of a 0.2GWh ZPNS Energy Bag. Solid line: with material mass; Dashed line: without material mass

Chapter 8

Testing

We contracted Thin Red Line Aerospace Ltd (TRL), a Canadian company that specialises in the design and manufacture of deployable fabric structures for use in space, to design and build three Energy Bag prototypes for underwater testing. Two of these bags are 1.8m diameter (designated ~1/10th scale) when fully inflated, and the other is 5m diameter (~1/4th scale). The bags have the design of pumpkin balloons but, being used to contain air underwater, these bags are stronger than pumpkin balloons which are traditionally made from linear low-density polyethylene (LLDPE). The two 1.8m bags have been rigged for testing in a tank which was built at the university and installed in the university laboratories. Each of the 1.8m bags has a quoted volume of 2.002m^3 when fully inflated, not including the volume of the lobes. The 5m bag will soon be tested at sea, and has a volume (excluding lobes) of 39.850m^3. All of the bags have been designed by TRL with a safety factor of at least 5 and with testing in mind; the bulkheads have multiple ports for sensors and pneumatics and have external threaded attachments for axial compression test equipment. TRL based the prototypes on their Ultra High Performance Vessel (UHPV) architecture.

For a given fill level, the forces in the bag and anchorage for an Energy Bag anchored in shallow waters are almost identical to those in the bag and anchorage for a bag anchored in much deeper waters. In fact in some ways, testing in calm shallow waters is more conservative than testing in deep waters, because in deep water the air is at a higher pressure than the air in the bag in shallow water, and so is denser. This means that the net buoyancy of a bag in deeper waters is less than the net buoyancy of the same bag at the same fill level in shallow waters. (Testing in water with waves is a different matter.)

It was found that one of the bags leaked around the seams at the top and sides of the bag, while the other bag has remained sealed to date. It was decided that in future the airline fitting should be attached to the top of an Energy Bag, because if water leaks into a bag its weight causes it to rest at the bottom of the bag, and to run into an airline fitted at the bottom. It then blocks the passage of air out of the bag, rendering the bag useless. A bag with an airline attached to the top will remain useful even if the bag is completely open at the base.

8.1 Design of the 1/10th Scale Prototypes

The ~1/10th scale prototypes, shown in figure 8.1, are adapted from the design of lobed superpressure pumpkin balloons used for long duration ballooning. The main differences between the prototypes and pumpkin balloons are: the prototypes have thicker membranes (because of the higher differential pressure across an underwater Energy Bag than across a balloon of equivalent size and because Energy Bag material mass is not an issue), and the prototypes have three layers of membrane (an airtight internal bladder/carrier made of polyurethane coated Nylon, and two layers of Nylon external envelopes) whereas a balloon typically only has one layer of film. The internal bladder also forms the pressure restraint shell which carries the local pressure load, while the external envelopes are neither airtight or load carrying, and are simply used to locate the tendons.

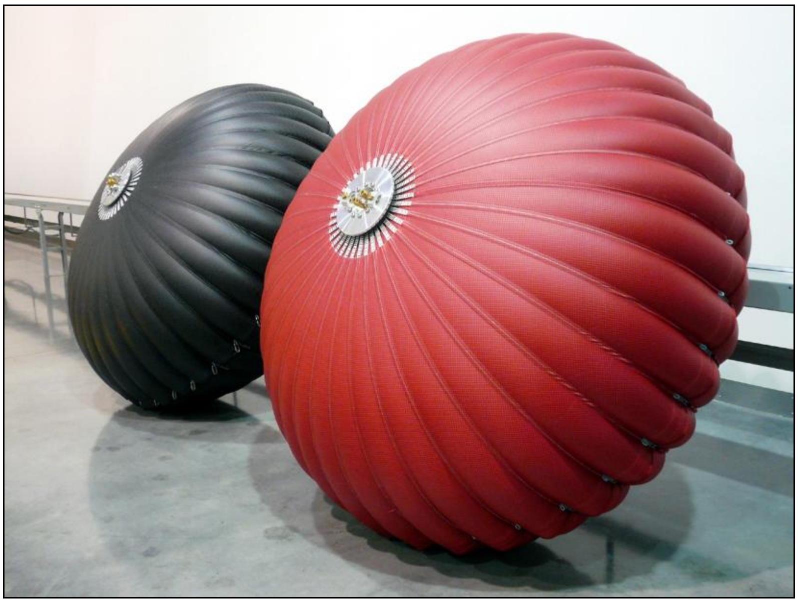


Fig. 8.1 The two 1/10th scale prototypes undergoing a test inflation before installation in the tank

The bag design is based on that of TRL's Ultra High Performance Vessel (UHPV). As explained in Chapter 5, the UHPV design differs from that of conventional superpressure balloons in that a UHPV is simply formed of two circles of fabric welded along their common edge, rather than being formed from *n* separate pieces of fabric, one for each lobe (where *n* is the number of lobes in the bag). Simply requiring two circles of fabric greatly reduces the complexity of manufacture: complex cutting patterns need not be followed, and only one weld is required, rather than *n* welds as with a conventional balloon. Reducing the number of seams also lowers the chances of defects (and so leaks) during manufacture.

The specifications for the bag are given in table 8.1. There are 36 Spectra®-Nylon tendons, which are contained in sleeves and are approximately 4% shorter than the fabric along which they run. As previously explained, this shortening is required to ensure that the fabric forming the lobes does not carry meridional stresses, and so all the membrane stress is transmitted circumferentially. The circumferential curvature is much higher than the meridional curvature, so transmitting the membrane stresses circumferentially allows for weaker materials to be used compared with transmitting the stresses meridionally. However, the effective lobe cutting pattern of a lobe in a UHPV is set and can only be adjusted through shortening of the tendons, whereas the cutting pattern of lobes in a conventional bag can be set freely. The wrinkling in the deflated bag caused by the tendon shortening can be seen in figure 8.2.

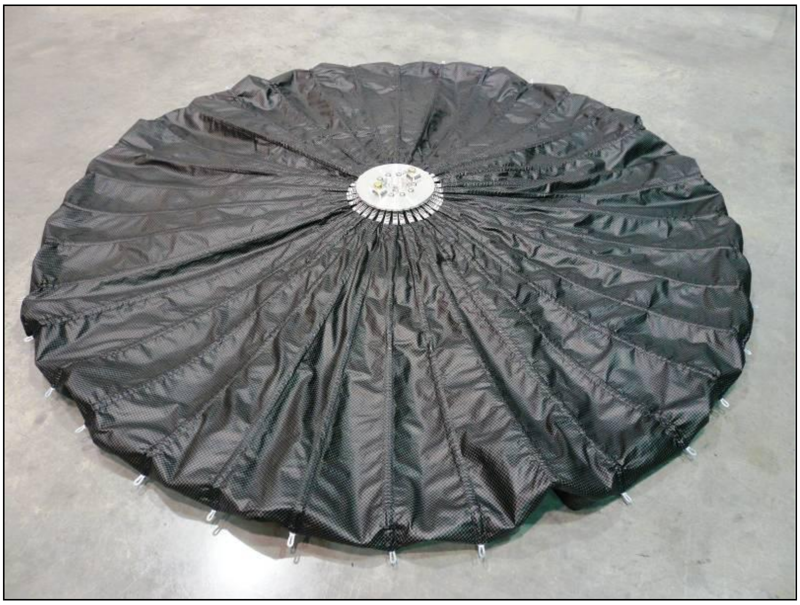


Fig. 8.2 One of the deflated 1/10th scale prototypes laid flat

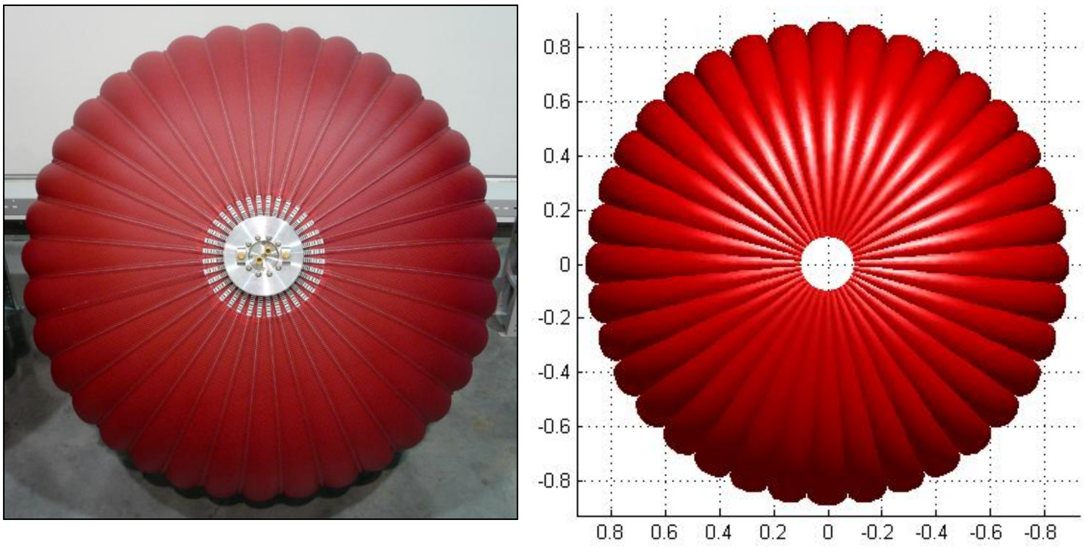
The upper and lower bulkheads are identical, manufactured of aluminium, and have a radius of 98mm. The bags are designed to be 1.8m in diameter (at the widest point, i.e. at the centre of a lobe) when symmetrical (so if the differential pressure could be infinite). In reality a symmetrical bag is not possible because of buoyancy, so the bag diameter will never quite be 1.8m and tends to 1.8m as the differential pressure tends to infinity. The fully inflated volume, excluding lobes, is 2.002m^3, and the bags were designed with a safety factor of 5. The meridional length of the bag (distance along the centre of a lobe from the top dead centre of the bag to the bottom dead centre) is 2.36m. The total mass of materials in each bag is 12.10kg.

Diameter	1.8 m
Volume (excluding lobes)	2.002 m^3
Number of tendons	36
Safety factor	5
Meridional length (pole to pole)	2.36 m
Bulkhead material	Aluminium
Bulkhead diameter	0.19686 m
Bulkhead mass	5.502 kg
Tendon material	Spectra®-Nylon
Tendon width	13.5 mm
Tendon thickness	2.8 mm
Tendon Young's modulus	2.34 GPa
Tendon density	1,400 kg/m^3
Tendon failure stress	374 MPa
Carrier material	420d polyurethane coated Nylon
Carrier thickness	0.4089 mm
Carrier Young's modulus (fill direction)	0.29364 GPa
Carrier Poisson's ratio	~0.4
Carrier density	1,440 kg/m^3
Minimum carrier failure stress (fill direction)	86 MPa

Table 8.1 Specifications of the 1.8m diameter prototype

The bulkheads are sealed against the bladder by compressing the bladder between two parts of the bulkhead ("clamp ring" and "pressure plate"). The clamp ring and pressure plate are bolted together with the bladder between them acting as a gasket. A top view of the inflated prototype (in air rather than water) is shown in figure 8.3(a), along with a top view of the same bag modelled using the 3D FEA tool presented in Chapter 6 (with a differential pressure at the base of p_0 = 0.4bar – the internal pressure used in figure 8.3(a) is unknown). Note that the hole in the FE model is smaller than the bulkhead seen on the prototype, but the hole is the correct size – the top plate of the bulkhead is wider than the clamp ring, against which the tendons terminate.

Thin Red Line specified the design burst pressure for the prototype as 21.2psi (1.46bar) overpressure, which they reduce to an allowable overpressure of 4.2psi (0.29bar) after including a safety factor of 5. Using the 3D FE and the materials properties supplied by TRL (and given in table 8.1), we calculated that the maximum allowable pressure, after including a safety factor of 5, should be 0.34bar. Clearly, Thin Red Line's allowable overpressure has been calculated by simply dividing the design burst pressure by the safety factor, however this is not best practice; the safety factor should be used with the failure *stresses* of the materials to obtain maximum allowable *stresses*. Using the 3D FEA it has been found that the differential pressure at the base of the bag (p_0) is not directly proportional to the maximum von Mises stress in the fabric, or to the maximum stress in the tendons. As an example, the p_0 = 0.34bar allowable pressure calculated using the 3D FEA tool gives a membrane stress equal to the failure stress of the membrane divided by the safety factor (the membrane would fail before the tendons if no bellows are included and the bag were being pressurised to destruction). However, multiplying this 0.34bar by the safety factor of 5 to obtain p_0 = 1.7bar and then using the 3D FEA tool again, we find that the maximum membrane stress has only increased by a factor of approximately 2.46 (so remaining less than half the failure stress) and the maximum tendon stress has only increased by a factor of approximately 4.10.



(a) (b)

Fig. 8.3 (a) End view of one of the inflated 1/10th scale prototypes; (b) End view of the bag modelled using the 3D FEA, underwater with p_0 = 40kPa

8.2 Installation and Control of the 1/10th Scale Prototypes

The tank in which the bags were installed was designed and manufactured at the university, and can be seen from above in figure 8.4. It is 3.6m long, 1.8m across, and 2.4m high, and comprises a frame of welded steel box-section, 25mm thick transparent acrylic walls sealed against the frame, and a steel exoskeleton welded to the frame. The exoskeleton comprises two horizontal crossbars and two vertical restraining bars against each sheet of acrylic, of which there are 6: one at each end and two along each side. The vertical restraining bars are pressed up against the acrylic walls by means of bolts threaded through the crossbars. The exoskeleton can also be seen later in figure 8.6. The bags are attached to eyebolts screwed into the base of the tank. The anchor rigging, shown in figure 8.5, comprises two Vectran® anchor straps attached to bridge anchors bolted against the lower bulkhead, and to an anchor link which attaches to the eyebolt.

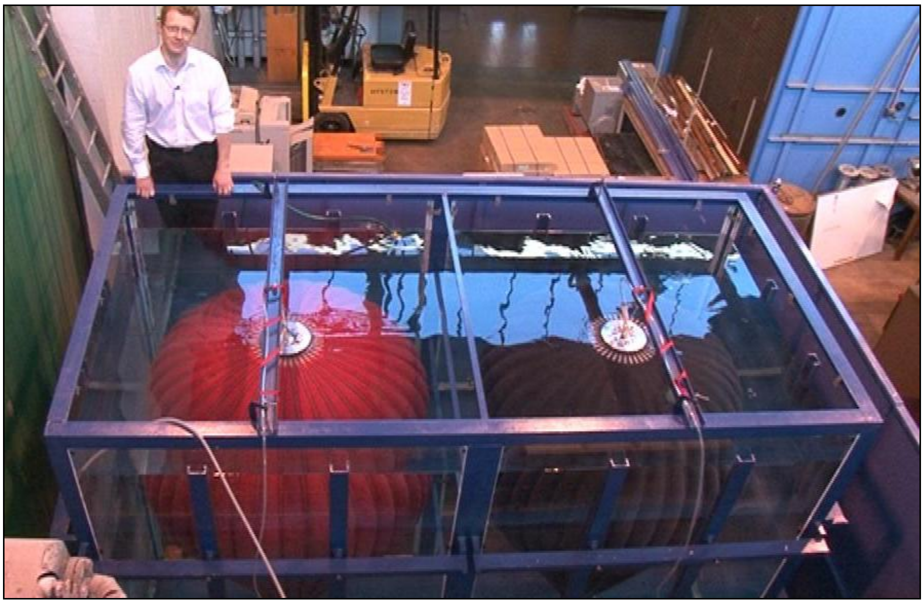


Fig. 8.4 Top view of the two 1/10th scale prototypes installed in the tank

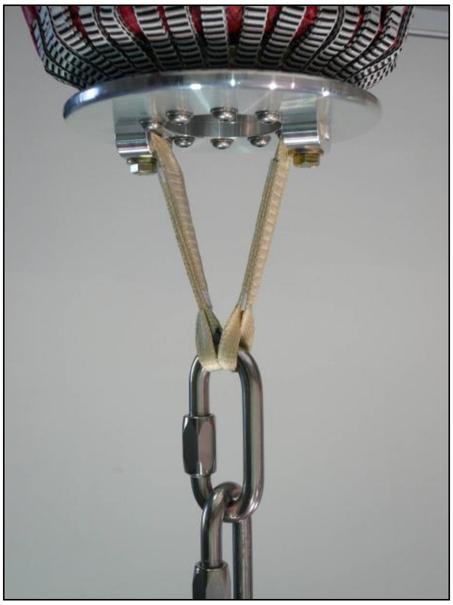


Fig. 8.5 Anchor rigging at the base of the bag

Initially there were several leaks in the sealing between the acrylic walls and the steel frame, so some resealing was necessary. After resealing it was found that the tank still leaked slightly during the filling process, but the leaking stopped by the time the tank was completely full of water.

Figure 8.6 shows the two bags partially inflated underwater with $p_0 \approx 0$. Clearly their shapes are very similar to that found for a ZPNS bag using the models presented earlier in this thesis.

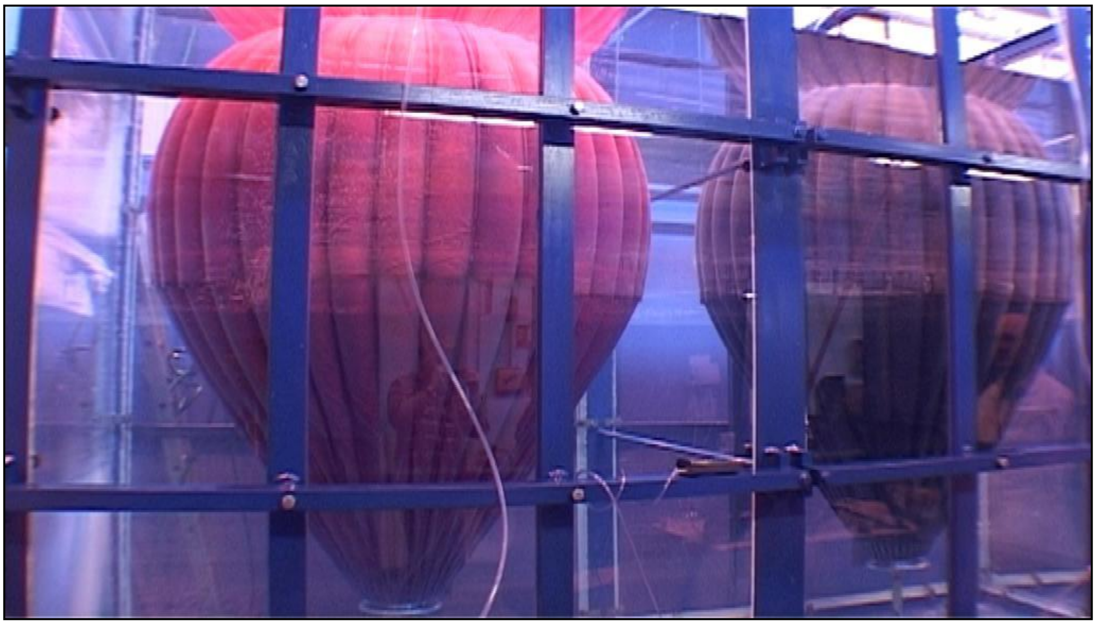


Fig. 8.6 The two 1/10th scale prototype Energy Bags partially inflated underwater, with $p_0 \approx 0$

The valve layout is shown in figure 8.7. The compressor is designated C and the two bags designated B1 and B2. The four 3/2 valves (V1-V4) can provide every flow path that may be desired. These are: single inflation of either bag (from atmosphere), single deflation of either bag (to atmosphere), dual inflation (from atmosphere), dual deflation (to atmosphere), and cycling in either direction. Initially a 4/2 valve was going to be used in place of valves V1 and V2, but this would not have provided the ability to dual inflate/deflate and was not as straightforward to source as two more 3/2 valves. Two pressure transducers (Druck PMP1400, 0-1bar) and two pressure relief valves were also included, one of each being attached into the airline connecting to each bag, as close to the bags as possible (without being in the water). The pressure transducers provide feedback to the control system (to control switching to a different mode) and the pressure relief valves ensure that the bags do not get over-inflated if there is a problem with the control system.

The compressor was also specifically chosen so that its pressure capacity is not higher than the maximum pressure of the bags, in case of a problem with the air control system. As a further level of security, valves V3 and V4 were fitted so that in their de-energised states they are venting to/taking from atmosphere (in case the power supply to the valves fails while power continues to be supplied to the compressor).

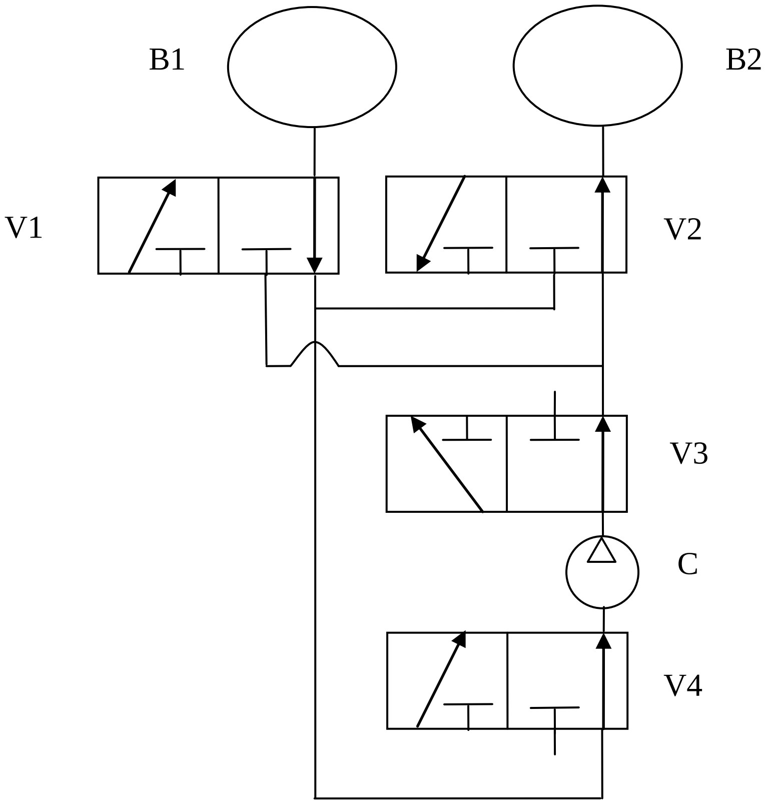


Fig. 8.7 Schematic of the air control system for cycling the 1/10th scale Energy Bag prototypes

The valves and pressure transducers are powered by a Calex power supply unit (32024E/10) providing a 24V DC output, and are connected to a Measurement Computing USB data acquisition (DAQ) board (USB-1408FS) which is connected to a PC running control code written using the data acquisition toolbox in Matlab. Relays are used to switch on or off the 24V supply to the valves. There are four relays in all, one in series with each valve, and the relays are triggered using the 5V output signal from the DAQ board.

The bags are being cycled, so as one bag inflates the other deflates, and air is sent from the deflating bag to the other using the compressor. This ensures a fast flow of air and doesn’t waste energy in the same way that repeatedly transferring air between the bags and the atmosphere would do. The control code splits into two distinct sections: one that should be run at the start of the day, which gets one bag to the specified upper pressure level and gets the other bag to the specified lower pressure level, and one that is run to carry out cycling. The cycling code includes measures to ensure that at the end of each half-cycle, one bag is at the upper pressure level and the other bag is at the lower pressure level, and air is transferred between the bags and atmosphere when necessary to ensure that this is the case (e.g. to compensate for leakage).

Initially air hose with 8mm ID (internal diameter) was used in all places, but it was found that the compressor cut out during inflation because it was overheating. The hose connecting valves V1 and V2 to the bags was replaced with hose with 19mm ID, and the compressor stopped cutting out.

The hose was initially connected to the bottom of each bag, but it was found that the black bag leaked slightly. This meant that a pool of water formed at the bottom of the bag, and entered the air hose. As a result, it was not possible to suck air out of the bag because water was sucked into the compressor and it cut out. Also, once a substantial amount of water had entered the hose it was not even possible to simply open the hose to the atmosphere and let the contained air pressure drive the water out – nothing moved because the water in the hose was trapping the air inside the bag. We came up with two ways of dealing with this problem: install a water trap, or attach the hose to the top of the bag. We decided to attach the hose to the top of the bag, but there is still the problem that water may enter the bag, and then remain at the bottom, slowly filling the bag with water. Some kind of water relief valve would be required to let water out of the bottom, otherwise the bag could be left open at the bottom, however an open bottom bag could not be overpressurised at the base.

The leaks in the black bag were evident because of streams of very small bubbles rising from various points in the inflated bag. These were at the seam around the equator of the bag (where the two circles of bladder material are welded together), and near the top of the bag. It can be seen on figure 8.3(a) that there is another section of material at the top of the bag, so we believe that the upper leak was coming from the seal between this section of material and the upper circle of bladder.

While the hose was connected to the bottom of the bag it was found that it was prone to kinking at the point of maximum curvature (just near the attachment to the bag). This could also trap air inside the bag, so we learned that large curvature of a flexible hose should be avoided for this reason. Instead, rigid fittings (e.g. L-fittings) should be used, or the hose should be reinforced where it curves (e.g. with spiral hose reinforcement).

8.3 Performance of the 1/10th Scale Prototypes

Testing of the 1/10th scale prototypes has only recently commenced but the bags have been performing as expected. Substantial wrinkling is visible at the bottom of the bags when they are partially inflated; the bag loses its rotational symmetry, potentially explaining why the shape of a partially inflated bag cannot be found using symmetry adapted 3D FEA.

The tank testing of the two 1.8m prototypes serves two main purposes: 1) Proof of concept, showing that air can be stored underwater in low-cost bags for extended periods of time without leakage, and; 2) To expose and deal with problems before the larger prototypes are installed at sea. The deformed shapes of the bags are not easy to measure because they are so large and located underwater. Photogrammetry was considered but it was decided that refraction caused by the acrylic walls and water would make accurate measurements difficult to achieve. If photogrammetry were to be used then it would be best if there was a scale (i.e. a grid) marked on a thin sheet of clear polyethylene hanging inside the tank, as close to the bag as possible, so that the scale is also refracted by the acrylic wall and some of the water.

We also generated a concept for a device used to measure the tension in the tendons. This would take the form of a three-pronged fork – the two outer prongs would be fixed to a bar held by the user and touched against the tendon, while the third prong (located midway between the two fixed prongs) would be used to apply a certain deflection to the tendon (e.g. a few millimetres) while the force required to give this deflection is measured by means of a spring with known stiffness. The deflection and force required could then be used to calculate the tension in the tendon. This device could also be used to measure the curvature of the tendon. However, the downside to using such a device is that it would have to be operated underwater and would require the user to either lean over the side of the tank (so the user's reach would limit the depth to which measurements can be taken), or put on a wetsuit and underwater breathing apparatus and climb into the tank!

8.4 Plans for the 1/4th Scale Prototypes

One 5m diameter (~1/4th scale) prototype has been manufactured for us by TRL and will soon be installed at sea, in reasonably sheltered and shallow waters. It will be attached to a buoy-mounted compressor, and cycled (i.e. it will be filled by the compressor, then once full it will automatically dump the air to the environment and filling will recommence). The 5m prototype has a very similar design to the 1.8m prototypes, based on TRL's UHPV architecture. It will have a fully inflated volume (excluding lobes) of 39.850m^3 and a total mass of 75.4kg. It will have 48 tendons and in some ways will be simpler than the 1.8m prototypes, having no bulkheads and only one layer of fabric – exactly the same material that is used as the bladder in the 1.8m prototypes, with the same thickness, Young's modulus, and strength. The 1.125in wide tendons in the 5m prototype are larger than the 9/16in tendons used in the 1.8m prototype, and are made of Vectran® rather than Spectra®-Nylon. The specification for the 5m prototypes is given in table 8.2.

Diameter	5 m
Volume (excluding lobes)	39.850 m^3
Number of tendons	48
Safety factor	6
Meridional length (pole to pole)	6.87 m
Tendon material	Vectran®
Tendon width	28.6 mm
Tendon thickness	2.6 mm
Tendon Young's modulus	10.95 GPa
Tendon density	1,400 kg/m^3
Tendon failure stress	808 MPa
Carrier material	420d polyurethane coated Nylon
Carrier thickness	0.4089 mm
Carrier Young's modulus (fill direction)	0.29364 GPa
Carrier Poisson's ratio	~0.4
Carrier density	1,440 kg/m^3
Minimum carrier failure stress (fill direction)	86 MPa

Table 8.2 Specifications of the 5m diameter prototype

A flow control valve will be used to switch between sending air from the compressor to the bag and sending air from the bag to atmosphere, and will be switched using a pressure switch with adjustable hysteresis/deadband.

Chapter 9

Conclusions and Future Work

9.1 Contributions of Present Research

In this thesis three computational fabric structure analysis tools have been presented, two of which are used to model axisymmetric structures and the third of which is used to model a full cable-reinforced membrane. The work was focused on the design of Energy Bags, inflatable fabric structures that will be anchored to the seabed and used for subsea compressed air energy storage. In the first of the two procedures used to model axisymmetric bags, the membrane is assumed to be inextensible and a system of four coupled ordinary differential equations is derived and then integrated numerically to find a solution. Because of the need to set boundary values and difficulties in finding the shapes of partially inflated bags, the second axisymmetric procedure was created. This is a finite element model of a single extensible meridional reinforcing cable ("meridian") using cable elements, and it proved to be much easier to use. Seabed resistance forces were included when using this procedure to look at the shapes of partially inflated bags with wide sealed bases, as without them the meridian would just be forced below the seabed.

Finally, three-dimensional finite element analysis has been used to model a full cable-reinforced membrane. It is necessary to take membrane wrinkling into account, and so a membrane element's elastic matrix is appropriately modified if the element undergoes uniaxial or biaxial wrinkling. It was hoped that the rotational symmetry of balloons and lobed Energy Bags would allow the size of the model to be reduced, by only modelling half of one lobe. However, it was found that converged solutions for some bags which take symmetry into account could only be reached if the bag was heavily pressurised, or if the membrane stiffness is not very high. In future, full Energy Bags will be modelled using the 3D FEA and it is hoped that converged solutions will be found.

A new lobe cutting pattern has been generated which minimises the maximum circumferential tension in a lobe of an Energy Bag which has had all meridional lobe stresses removed (by means of tendon shortening or a bellows arrangement providing excess fabric in the meridional direction). This cutting pattern has constant circumferential tension at all points along the lobe, and so is known as a Constant Tension (CT) lobe. A CT lobe has the smallest possible area that a lobe may have if the maximum circumferential tension is also being minimised. As well as reducing material usage (and so cost), it is also anticipated that the CT lobe will increase the stability of balloons and Energy Bags; it was shown in [104] that maximising the stability of a balloon is equivalent to minimising the area of the lobe cutting pattern.

By examining scaling laws, it has been shown that in order to minimise the cost of materials per unit of energy stored for an axisymmetric, natural shape bag, the cost of reinforcement should be equal to the cost of surface, and the bag diameter which accomplishes this has been found in terms of the price of reinforcement materials and the price of surface materials. This optimisation and the three analysis tools presented in this thesis have been documented in a paper which has been published in Proceedings of the IMechE, Part C: Journal of Mechanical Engineering Science [127].

The optimum shape (minimising cost of materials per unit of energy stored) of an inextensible, axisymmetric bag which may have nonzero base radius has been found using tiered line searches, Brent's method, and penalty functions to solve the constrained multidimensional optimisation problem. The optimum axisymmetric Energy Bag only has stresses in the meridional direction (so zero circumferential stress), a wide base radius, masses hanging from the inside, and is underpressurised at the base. At 500m depth it costs less than £1,000/MWh, though estimated costs have so far only been attributed to materials. This cost is much less than the costs of all other competing large-scale energy storage technologies – modern pumped hydro plants, for example, cost at least £50,000/MWh, and Dinorwig pumped hydro plant in North Wales cost over £140,000/MWh (in 2010 prices) when it was constructed in 1980. This work was presented at the IoP's 7th International Conference on Modern Practice in Stress and Vibration Analysis in Cambridge, UK, in September 2009, and an accompanying paper was published in the conference proceedings [116].

Investigations were carried out into the effects of bag size and maximum fill level on the three components of materials costs (reinforcement, surface, and ballast) for an enclosed bag with a single point of anchorage, using the axisymmetric FE model. It was found that costs (per unit of energy stored, assuming isothermal expansion) of meridional reinforcement and surface materials are minimised by making the bag as small as possible, while bag size has no effect on the cost of ballast per unit of energy stored. For a bag anchored at significant depth (e.g. 500m), costs of meridional reinforcement are minimised by underpressurising the base of the bag. The effect of maximum fill level on the cost of surface materials depends upon the type of bag being used; if each lobe in the bag is a separate panel (so that the flat lobe cutting patterns can take any shape that the designer wishes) then the optimum bag in terms of surface materials is underpressurised at the base, and the lobe cutting patterns should be designed appropriately. However, if the bag is formed from two circles of fabric (as the prototype bags are) then the bag should be as pressurised as possible. This also minimises ballast costs.

A specification for a 0.2GWh Energy Bag has been drawn up. If this is to be anchored at 500m depth and filled to a maximum differential pressure at the base of zero, and if the air contained will be expanded isothermally (quite a conservative approach), the volume of the bag must be 35,705m^3. Such a bag has a height of 41.3m, and an estimated cost of materials (including ballast mass) of £3.89m, working out at a cost of £19,400/MWh. Meridional reinforcement costs (£2.5m) make up the largest component of the total cost (64%). Membrane only accounts for about 1.4% of the total cost and the rest (34.6%) is the cost of ballast.

Two 1.8m diameter prototype Energy Bags have been manufactured for us by Thin Red Line Aerospace Ltd. They have been installed in a tank on campus and testing has commenced – they will be cycled over 1,000 times, acting as a proof of concept and informing us about the problems that must be overcome.

It has been found that it is best to always locate airline fittings at the top of a bag, even if the bag is sealed at the base, just in case the sealing fails – water entering the bag will naturally sit at the base, and could be evacuated from the base of a sealed bag through a valve located at the base. The airline fittings on the test bags were at the base and, when one of the bags leaked during the testing, it was found that the water ran down into the pipe (and up part of the pipe that came out of the top of the tank) and blocked the air from escaping when the pipe was opened to the atmosphere. If the hose is attached to the top of the bag, air can still escape if there is water in the bag, and so a sealed bag remains useful even if there is a leak (as long as there is some means of removing the water from the bottom of the bag, e.g. a valve in the base). However, it may be the case that at more substantial depths (the test bags were only in 2.4m of water) the pressure of the air in the bag would have forced the water out of the bottom. It may be the case that we want to suck leaked water out of the bag by creating a vacuum in the pipe – this would only be possible if the pipe were attached to the base of the bag, where the water rests.

Finally, water ingress would not be an issue if the bag is open-bottomed and the airline is attached to the top. An open-bottomed bag could not be overpressurised, but this may not be a problem as it has been shown in this thesis that the cost per unit energy stored of meridional reinforcement increases as pressurisation increases, as does the cost per unit energy stored of fabric in a bag with separate lobes. Also, an open bottom brings with it an important safety feature: the bag will automatically vent air from the base when the pressure of the contained air tries to rise above the hydrostatic pressure of the water outside the base, so the bag cannot be damaged through overpressurisation (as long as the size of the hole in the base is not so small that air is being pumped in at a higher rate than it can vent through the hole).

9.2 Future Work

During the course of this work, a number of future areas of research were envisaged. In terms of developing the analysis tools, it is important to be able to

- Try to model a full Energy Bag using the 3D FEA, rather than just half of a single lobe.

It will also be wise to

- Carry out benchmarking of the 3D FEA tool by creating some simple physical models of inflated cable-reinforced envelopes and comparing the stresses in the models with the stresses found using the 3D FEA.

In reality the tendons of an Energy Bag will be located in sleeves on the fabric, and not attached to the fabric. Also, bellows are included in a UHPV to ensure that there is no meridional stress in the fabric. Therefore we will

- Extend the 3D FEA so that the tendons are separate from the fabric but constrained to lie along the seams between lobes (as if free to slide in sleeves), include friction between the tendons and fabric, and include bellows around the equator.

Fabric is often manufactured with different strengths and stiffnesses in the warp and fill directions, and so it would be useful to

- Extend the 3D FEA to be able to set different stiffnesses for the warp and fill directions.

Problems with stability have been encountered with superpressure pumpkin balloons, so the 3D FEA tool should also be used to

- Check the stability of Energy Bags, particularly those designed to have a maximum of zero differential pressure at the base (e.g. open-bottomed bags).

It would also be useful to use the 3D FEA tool to

- Study the effects of current on the shape of a bag, the stresses within the materials, and the stability of the ballast.

If it were decided to manufacture Energy Bags with separate lobes (rather than a small number of separate sheets, e.g. just two circles of fabric as in the prototypes), it would be important to

- Carry out lobe cutting pattern optimisation using the 3D FEA tool and insights gained from analytical considerations in this thesis (e.g. the Constant Tension lobe).

It would be advisable to couple cutting pattern optimisation with a study of stability, to ensure that the bag formed using the optimum lobe cutting pattern is stable when fully inflated.

There are also a number of other areas that should be investigated. The materials used in an Energy Bag will be critical to its life, so it is necessary to

- Test sample Energy Bag materials, focusing on their fatigue life, ability to remain airtight, and resistance to corrosion and the marine environment (seawater, sand, and marine life such as barnacles).

Some of these issues will also be studied when we

- Carry out sea-based testing of a larger 5m diameter prototype that will be automatically filled then emptied many times over the course of several months.

A 5m diameter prototype has already been manufactured for us for this very purpose by Thin Red Line Aerospace, the company that we contracted to manufacture the 1.8m diameter prototypes. Finally, before moving onto larger scales and commercialisation it will be necessary to

- Carry out a wider reliability and risk assessment, including generating appropriate safety factors and understanding the consequences of sudden failure of an Energy Bag on marine life, boats on the surface, and any bags attached to the same pipeline/manifold. Also carry out fatigue calculations and lifetime assessment.

9.3 Future of ICARES

Over the course of this research, Energy Bags have progressed from concept to prototype and been shown to be cheaper than all other forms of large-scale energy storage. The analysis tools presented in this thesis have been useful in understanding the deformed shapes of Energy Bags, and will allow us to be sure of required dimensions and materials strengths when designing bags in future.

In the very near future, a 5m diameter prototype will be installed at sea and attached to a buoy-mounted compressor. This will be tested for several months and then removed for inspection. The 5m diameter bag has been manufactured for us by Thin Red Line Aerospace; this is based on Thin Red Line’s Ultra High Performance Vessel architecture (very much like the shape of a pumpkin balloon), and is very similar in design to the 1.8m diameter prototypes (also manufactured by Thin Red Line). Along with building the scale of the prototypes, creating procedures for deployment of the bag and ballast in deep sea, and addressing the future areas of Energy Bag research detailed above, it may now be useful to work on progressing integral compression wind turbines and wave energy converters, thermal energy storage and heat exchangers, and to look at options for siting the thermal energy stores, heat exchangers, and expander-generators at sea.

It should be remembered that even in the event that integral compression renewable energy harvesters do not come to commercial reality, there is still a future for Energy Bags as a low-cost means of storing large amounts of electricity from the grid. However, ICARES is a promising technology which has emerged as a natural consequence of two major current considerations taken together: large-scale energy storage and the need to achieve substantial cost reduction in offshore wind turbines. The potential market for ICARES is extremely large and, if it is successfully commercialised, ICARES could deal with both of these issues while providing significant benefits to investors and the local population and kick-starting a renewable energy industry in the UK and Ireland.

References

[1] EIA, EIA 2010 International Energy Outlook, http://www.eia.doe.gov/oiaf/ieo/highlights.html. Last accessed 16/06/2010

[2] IPCC, *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change* [Solomon S, Qin D, Manning M, Chen Z, Marquis M, Averyt KB, Tignor M and Miller HL (Eds.)], 2007, Cambridge University Press: Cambridge, UK and New York, NY, USA, Ch.1 p97

[3] *Ibid.*, SPM p2-3

[4] *Ibid.*, Ch.7 p511

[5] *Ibid.*, Ch.2 p137

[6] *Ibid.*, Ch.7 p512

[7] *Ibid.*, Ch.2 p131

[8] *Ibid.*, SPM p5

[9] *Ibid.*, SPM p13

[10] Orr JC *et al*, Anthropogenic ocean acidification over the twenty-first century and its impact on calcifying organisms, *Nature*, 437, 2005, pp.681-686

[11] EIA, EIA Coal Reserves Information Sheet: Reserves, http://www.eia.doe.gov/neic/infosheets/coalreserves.html. Last accessed 16/06/2010

[12] EIA, *EIA 2007 International Energy Outlook*, Ch.5 p49

[13] CIA, The World Factbook, https://www.cia.gov/library/publications/the-world-factbook/index.html. Last accessed 16/06/2010

[14] OPEC, OPEC Share of World Crude Oil Reserves 2009, http://www.opec.org/opec_web/en/data_graphs/330.htm. Last accessed 19/10/2009

[15] Directive 2009/28/EC, Official Journal of the European Union, L 140, 2009, pp.16-62

[16] HM Government, The UK Renewable Energy Strategy 2009, http://www.decc.gov.uk/en/content/cms/what_we_do/uk_supply/energy_mix/renewable/res/res.aspx. Last accessed 17/06/2010

[17] Ofgem, The Renewables Obligation buy-out price and mutualisation ceiling 2010-11, http://www.ofgem.gov.uk/Media/PressRel/Documents1/RO%20Buy-Out%20price%202010%2011%20FINAL%20FINAL.pdf. Last accessed 14/10/2010

[18] Ofgem, Renewables Obligation, http://www.ofgem.gov.uk/Sustainability/Environment/RenewablObl/Pages/RenewablObl.aspx. Last accessed 15/09/2010

[19] The Renewables Obligation (Amendment) Order 2010, http://www.legislation.gov.uk/uksi/2010/1107/contents/made. Last accessed 14/10/2010

[20] Ernst & Young, Cost of and financial support for offshore wind, 2009, http://webarchive.nationalarchives.gov.uk/+/http://www.berr.gov.uk/files/file51142.pdf. Last accessed 14/10/2010

[21] Feed-In Tariffs, Tariff Levels Table, http://www.fitariffs.co.uk/eligible/levels. Last accessed 05/10/2010

[22] Ofgem, Factsheet: Introducing the Feed-in Tariff scheme, http://www.ofgem.gov.uk/Media/FactSheets/Documents1/fitfs_energy%20prices%20update%20FS.pdf. Last accessed 19/10/2010

[23] MacKay DJC, *Sustainable Energy – without the hot air*, 2009, UIT Cambridge: Cambridge, UK, p186

[24] Sinden G, Characteristics of the UK wind resource: long term patterns and relationship to electricity demand, *Energy Policy*, 35, 2007, pp.112-127

[25] MacKay, *op. cit.*, p188

[26] BWEA, Actions for 33GW, http://www.bwea.com/pdf/publications/33GW_08.pdf. Last accessed 28/07/2010

[27] Chen GZ, Seawater Supercapattery, presentation at The University of Nottingham on 01/12/2010

[28] Andrews J and Jelley N, *Energy Science: Principles, Technologies, and Impacts*, 2007, Oxford University Press: Oxford, UK, p70

[29] Sharman H, Why wind power works for Denmark, *Proceedings of ICE – Civil Engineering*, 158, 2005, pp.66-72

[30] Safalra, Historical UK Inflation And Price Conversion, http://safalra.com/other/historical-uk-inflation-price-conversion. Last accessed 05/08/2010

[31] MacKay, *op. cit*., p191

[32] MacKay, *op. cit*., p193

[33] Andrews and Jelley, *op. cit*., p284

[34] New Energy News, Energy Storage Gets Hot 1, http://newenergynews.blogspot.com/2009/11/energy-storage-gets-hot-1.html. Last accessed 26/07/2011

[35] Garvey SD and Pimm AJ, Thermal Performance of an Integrated Compressed Air Renewable Energy System, *Energy Storage Congress at the HUSUM International Wind Energy Fair, Husum, Germany, 20-24 Sep 2010*

[36] BBC News, Gas storage plan for Preesall salt caverns rejected, http://news.bbc.co.uk/1/hi/england/lancashire/8485060.stm. Last accessed 01/12/2010

[37] Siemens, Hywind: Siemens and StatoilHydro install first floating wind turbine, http://www.siemens.com/press/en/pressrelease/?press=/en/pressrelease/2009/renewable_energy/ere200906064.htm. Last accessed 06/10/2010

[38] DTI, *Atlas of UK Marine Renewable Energy Resources*, 2004, p27 with scale taken from p1

[39] Wikipedia, Tensile structure, http://en.wikipedia.org/wiki/Tensile_structure. Last accessed 06/07/2010

[40] Sorensen B, *Renewable Energy Conversion, Transmission and Storage*, 2007, Elsevier: USA

[41] Sandia National Laboratories, Solution to some of country's energy woes might be little more than hot air, http://www.sandia.gov/media/NewsRel/NR2001/norton.htm. Last accessed 25/02/2009

[42] Andrews and Jelley, *op. cit*.

[43] Meyer F (Ed.), Compressed air energy storage power plants, BINE Projekt-Info 05/07, FIZ Karlsruhe GmbH, Bonn, Germany, 2007

[44] CAES Development Co. & Sandia National Laboratories, CAES Monitoring to Support RMRCT, DE-FC26-01NT40868, 2004

[45] Doetsch C and Wolf D, Druckluftspeicherkraftwerke – Bewertung und Charakterisierung von netzintegrierten druckluftbasierten

Energiespeicherkonzepten. *Fraunhofer Institut Umwelt-, Sicherheits-, Energietechnik UMSICHT, Fulda, Germany, 25-26 Mar 2009*

[46] Perry RH, *Perry's Chemical engineers' handbook*, 6th Ed., 1984, McGraw-Hill: New York, USA, Table 3-162

[47] Seymour RJ, Undersea Pumped Storage for Load Leveling, *Proc. of California and the World's Oceans '97, 1, San Diego, CA, USA, 1997*, pp.158-163

[48] Seymour RJ, Ocean energy on-demand using underocean compressed air storage, *26th International Conference on Offshore Mechanics and Arctic Engineering, San Diego, CA, USA, 10-15 Jun 2007*

[49] Cheung KYC, Cheung STH, De Silva RGN, Juvonen MPT, Singh R and Woo JJ, Large-Scale Energy Storage Systems, 2003, http://q-m.org/academic/ise2grp/energystorage_report/node6.html. Last accessed 25/02/2009

[50] Ter-Gazarian A, *Energy Storage for Power Systems*, 1994, Peter Peregrinus Ltd. on behalf of the IEE

[51] Hauff C, Pumped storage technology – new milestones for European pumped storage, *Int. J. Hydropower and Dams*, 9 (5), 2003, pp.101-103

[52] Berry G, Present and Future Electricity Storage for Intermittent Renewables, *From workshop proceedings, "The 10-50 Solution: Technologies and Policies for a Low-Carbon Future.", 2004*

[53] Tennessee Valley Authority, http://www.tva.gov/power/pumpstorart.htm. Last accessed 03/03/2009

[54] Denholm P and Kulcinski GL, Life cycle energy requirements and greenhouse gas emissions from large scale energy storage systems, *Energy Conversion and Management*, 45 (13-14), 2004, pp.2153-2172

[55] ESA, Technologies: Pumped Hydro Storage, http://www.electricitystorage.org/tech/technologies_technologies_pumpedhydro.htm. Last accessed 02/03/2009

[56] Connolly D, *An investigation into the energy storage technologies available, for the integration of alternative generation techniques*, 2007, University of Limerick

[57] Fujihara T, Imano H and Oshima K, Development of Pump Turbine for Seawater Pumped-Storage Power Plant, *Hitachi Review*, 47 (5), 1998, pp.199-202

[58] Chen H, Cong NC, Yang W, Tan C, Li Y and Ding Y, Progress in electrical energy storage system: A critical review, *Progress in Natural Science*, 19, 2009, pp.291-312

[59] Gonzalez A, Ó Gallachóir B, McKeogh E and Lynch K, *Study of Electricity Storage Technologies and Their Potential to Address Wind Energy Intermittency in Ireland*, 2004, University College Cork: Cork, Ireland

[60] Van der Linden S, Bulk energy storage potential in the USA, current developments and future prospects, *J. Energy*, 31 (15), 2006, pp.3446-3457

[61] Marsh G, RE Storage – The Missing Link, *Refocus*, 3 (10), 2002, pp.38-41

[62] Ibrahim H, Ilinca A and Perron J, Energy storage systems – Characteristics and comparisons, *Renewable and Sustainable Energy Reviews*, 12, 2008, pp.1221-1250

[63] Taylor JB, Alderson JEA, Kalyanam KM, Lyle AB and Phillips LA, Technical and economic assessment of methods for the storage of large quantities of hydrogen, *Int. J. Hydrogen Energy*, 11, 1986, pp.5-22

[64] Stone HBJ, Veldhuis I and Richardson RN, An Investigation into Large-Scale Hydrogen Storage in the UK, *Proc. International Hydrogen Energy Congress and Exhibition (IHEC) 2005, Istanbul, Turkey, 13-15 Jul 2005*

[65] Sandia National Laboratories, National Solar Thermal Test Facility, http://www.sandia.gov/Renewable_Energy/solarthermal/NSTTF/salt.htm. Last accessed 31/01/2011

[66] Gil A, Medrano M, Martorell I, Lázaro A, Dolado P, Zalba B and Cabeza LF, State of the art on high temperature thermal energy storage for power generation. Part 1 – concepts, materials and modellization, *Renewable & Sustainable Energy Reviews*, 14 (1), 2010, pp.31-55

[67] Isentropic, Storage, http://www.isentropic.co.uk/index.php?page=storage. Last accessed 14/02/2011

[68] Seaflex, Project Overview: Vigdis Template Stabilisation, http://www.seaflex.co.uk/Information/Vigdis_Template.pdf. Last accessed 29/09/2010

[69] IMCA D 016, Underwater air lift bags, 2007

[70] Seaflex, Product Overview: Air Lift Bag, http://www.seaflex.co.uk/Information/ALB_Overview.pdf. Last accessed 30/09/2010

[71] Seaflex, Product Specification – 35t Air Lift Bag (ALB): Parts & Features, email communication from Graham Brading of Seaflex to Andrew Pimm on 01/10/2010

[72] Offshore Technology, JW Automarine, http://www.offshore-technology.com/contractors/lifting/jwa. Last accessed 29/09/2010

[73] JW Automarine, Premier Parachute Totally Enclosed Liftbag datasheet, http://www.jwautomarine.co.uk/dts_pdf/dt_lbs_prte.pdf. Last accessed 30/09/2010

[74] JW Automarine, Planning, Mobilisation and Implementation, http://www.jwautomarine.co.uk/download/lb_man4.pdf. Last accessed 30/09/2010

[75] JW Automarine, Premier Parachute Totally Enclosed Liftbag, http://www.jwautomarine.co.uk/pr_lbs_pr_prte.htm. Last accessed 29/09/2010

[76] Seaflex, Product Specification – 35t Inflatable Buoyancy Unit (IBU): Parts & Features, http://www.sea-landtech.com.sg/pdf/35t%20IBU%20Spec.pdf. Last accessed 01/10/2010

[77] JW Automarine, Products, http://www.jwautomarine.co.uk/products.htm#lbs. Last accessed 30/09/2010

[78] Canflex Inc., Lift Bags brochure, http://www.canflexinc.com/pdf/Lift%20bags%20flyer_en.pdf. Last accessed 30/09/2010

[79] Email communication from Alan Milne of Water Weights to Andrew Pimm on 18/10/2010

[80] Water Weights, Brochure 2009, http://www.waterweights.com/files/files/downloads/Water_Weights_brochure2009.pdf. Last accessed 05/10/2010

[81] Water Weights, Case Study: Great Yarmouth Outer Harbour, http://www.waterweights.com/files/files/2010_Seajacks__Great_Yarmouth_Outer_Harbour.pdf. Last accessed 05/10/2010

[82] Wikipedia, Balloon (aircraft), http://en.wikipedia.org/wiki/Balloon_%28aircraft%29. Last accessed 01/07/2010

[83] BBC News, Special report: Aviation's last great challenge, http://news.bbc.co.uk/1/hi/special_report/1998/11/98/great_balloon_challenge/223362.stm. Last accessed 01/07/2010

[84] German Federal Archive, Bild 102-13738, http://www.bild.bundesarchiv.de/archives/barchpic/search/_1278071294/?search[view]=detail&search[focus]=1. Last accessed 02/07/2010

[85] Encyclopaedia Britannica Online, Balloon (aircraft), http://www.britannica.com/EBchecked/topic/50696/balloon#. Last accessed 05/07/2010

[86] Parsch A, Directory of US Military Rockets and Missiles, Appendix 4: Undesignated Vehicles, WS-119L / WS-461L, http://www.designation-systems.net/dusrm/app4/ws-119l.html. Last accessed 05/07/2010

[87] NASA Wallops Flight Facility, Super Pressure vs Zero Pressure, http://sites.wff.nasa.gov/code820/pumpkinvssphere.html. Last accessed 17/03/2009

[88] Pagitz M, The future of scientific ballooning, *Phil. Trans. R. Soc. A*, 365, 2007, pp.3003-3017

[89] Yamagami T, Saito Y, Matsuzaka Y, Namiki M, Toriumi M, Yokota R, Hirosawa H and Matsushima K, Development of the highest altitude balloon, *Advances in Space Research*, 33, 2004, pp.1653-1659

[90] Anon., Research development in the field of high altitude plastic balloons, NONR-710(01a) reports, Department of Physics, University of Minnesota, MN, USA, 1951-1956

[91] Cathey Jr. HM, The NASA super pressure balloon – A path to flight, *Advances in Space Research*, 44, 2009, pp.23-38

[92] Pagitz M and Pellegrino S, Buckling pressure of “pumpkin” balloons, *Int. J. Solids and Structures*, 44, 2007, pp.6963–6986

[93] Tabarrok B and Qin Z, Nonlinear analysis of tension structures, *Computers and Structures*, 45 (5/6), 1992, pp.973–984

[94] Smalley JH, Determination of the Shape of a Free Balloon – Balloons with Superpressure, Subpressure and Circumferential Stress and Capped Balloons, AFCRL-65-72, Air Force Cambridge Research Labs, Bedford, MA, USA, 1964

[95] Baginski F and Winker J, The natural shape balloon and related models, *Advances in Space Research*, 33, 2004, pp.1617-1622

[96] Baginski F, Collier W and Williams T, A Parallel Shooting Method for Determining the Natural Shape of a Large Scientific Balloon, *SIAM J. Appl. Math.*, 58 (3), 1998, pp.961-974

[97] Taylor GI, On the Shapes of Parachutes, 1919, Paper written for the Advisory Committee for Aeronautics. In: Batchelor GK (Ed.), *The Scientific Papers of G.I. Taylor, Volume III: Aerodynamics & the Mechanics of Projectiles & Explosions*, 1963, Cambridge University Press: Cambridge, UK, pp.26-37

[98] The National Guard, Rhode Island National Guard Holds 25th Annual International Military Parachute Competition, http://www.ng.mil/news/archives/2007/09/091907-RI_NG.aspx. Last accessed 06/10/2010

[99] Baginski F, Brakke KA and Schur WW, Unstable cyclically symmetric and stable asymmetric pumpkin balloon configurations, *Journal of Aircraft*, 44 (3), 2006, pp.764-773

[100] Calladine CR, Stability of the Endeavour balloon, 1988. In: Elishakoff I, Arbocz J, Babcock CD and Libai A (Eds.), *Buckling of Structures – Theory and Experiment*, 1988, Elsevier Science Publishers: Amsterdam, The Netherlands, pp.133-149

[101] Lennon BA and Pellegrino S, Stability of lobed inflatable structures, *41st AIAA/ASME/ASCE/AHS/ASC SDM Conference, AIAA-2000-1728, 2000*

[102] Wakefield D, Numerical Modelling of Pumpkin Balloon Instability, *AIAA 5th ATIO and 16th Lighter-Than-Air Systems Technology and Balloon Systems Conferences, AIAA-2005-7445, 2005*

[103] Pagitz M and James J, Symmetry Transformation Matrices for Structures, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2007

[104] Pagitz M and Pellegrino S, Shape Optimization of “Pumpkin” Balloons, *AIAA Balloon Systems Conference, Williamsburg, VA, USA, 21-24 May 2007*

[105] Wesley Wong Y and Pellegrino S, Wrinkled Membranes Part III: Numerical Simulations, *Journal of Mechanics of Materials and Structures*, 1 (1), 2006, pp.61-93

[106] Miller RK and Hedgepeth JM, An algorithm for finite element analysis of partly wrinkled membranes, *AIAA Journal*, 20, 1982, pp.1761-1763

[107] Miller RK, Hedgepeth JM, Weingarten VI, Das P and Kahyai S, Finite element analysis of partly wrinkled membranes, *Computers and Structures*, 20, 1985, pp.631-639

[108] Adler AL, *Finite element approaches for static and dynamic analysis of partially wrinkled membrane structures*, 2000, PhD thesis, University of Colorado at Boulder

[109] Contri P and Schrefler BA, A geometrically nonlinear finite element analysis of wrinkled membrane surfaces by a no-compression material model, *Commun. Appl. Numer. Methods*, 4, 1988, pp.5-15

[110] Nishima T, Tosaka N and Honma T, Membrane structure analysis using the finite element technique, Shells, Membranes and Space Frames, *Proceedings of the IASS Symposium, 2, Osaka, Japan, 1986*, pp.9-16

[111] Wesley Wong Y and Pellegrino S, Wrinkled Membranes Part I: Experiments, *Journal of Mechanics of Materials and Structures*, 1 (1), 2006, pp.3-25

[112] Wikiversity, The Newton-Raphson method, http://en.wikiversity.org/wiki/Nonlinear_finite_elements/Newton_Raphson_method. Last accessed 04/01/2011

[113] Becker AA, *An Introductory Guide to Finite Element Analysis*, 2003, Professional Engineering Publishing: London, UK

[114] Spyrakos CC and Raftoyiannis J, *Linear and Nonlinear Finite Element Analysis in Engineering Practice*, 1997, Algor Publishing: Pittsburgh, PA, USA

[115] Yajima N, Izutsu N, Imamura T and Abe T, *Scientific Ballooning: Technology and Applications of Exploration Balloons Floating in the Stratosphere and the Atmospheres of Other Planets*, 2009, Springer: Berlin, Germany

[116] Pimm AJ and Garvey SD, Analysis of flexible fabric structures for large-scale subsea compressed air energy storage, *J. Phys.: Conf. Ser.*, 181 (1), 2009

[117] Gatti-Bono C and Perkins NC, Numerical Simulations of Cable/Seabed Interaction, *Proc. 13th Offshore and Polar Eng. Conf., Honolulu, Hawaii, USA, 25-30 May 2003*, pp.120-126

[118] Gobat JI and Grosenbaugh MA, Time-domain numerical simulation of ocean cable structures, *Ocean Engineering*, 33, 2006, pp.1373-1400

[119] Lennon A and Pellegrino S, Structural Mechanics of Lobed Inflatable Structures, *European Conference on Spacecraft Structures, Materials & Mechanical Testing 2005, Noordwijk, The Netherlands, 10-12 May 2005* (ESA SP-581, Aug 2005)

[120] Pagitz M, *Analytical and Numerical Studies of Superpressure Balloons*, 2007, PhD thesis, University of Cambridge, UK

[121] Email correspondence from Maxim De Jong of Thin Red Line Aerospace to Andrew Pimm on 07/01/2011

[122] Rao SS, *The Finite Element Method in Engineering*, 4th Ed., 2005, Elsevier: Oxford, UK

[123] Bazaraa MS, Sherali HD and Shetty CM, *Nonlinear Programming: Theory and Algorithms*, 2nd Ed., 1993, John Wiley & Sons: Hoboken, NJ, USA, p361

[124] Brent RP, *Algorithms for Minimization without Derivatives*, 1973, Prentice-Hall: Englewood Cliffs, NJ, USA, Ch.5

[125] Zoutendijk G, *Methods of Feasible Directions*, 1960, Elsevier: Amsterdam, Netherlands

[126] Smalley JH, Balloon Shapes and Stresses below the Design Altitude, NCAR-TN-25, National Center for Atmospheric Research, Boulder, CO, USA, 1966

[127] Pimm AJ, Garvey SD and Drew RJ, Shape and cost analysis of pressurized fabric structures for subsea compressed air energy storage, *Proc. IMechE Part C: Journal of Mechanical Engineering Science*, 225 (5), 2011, pp.1027-1043

Appendices

A. Matlab Code for Axisymmetric ODE Method

```
function [opt_fun,Y,T,costs] =
optim_vol(reqd_vol,p_zero,sigma_c,sigma_w)
% Finds the centre height that will give an optimum bag of volume
given by reqd_vol.

[z_zero,opt_fun,exitflag] = fminbnd(@(z_zero)
optim_height(z_zero,reqd_vol,p_zero,sigma_c,sigma_w),0*(reqd_vol/pi)
^(1/3),5*(reqd_vol/pi)^(1/3));
[opt_fun,Y,T,costs] =
optim_height(z_zero,reqd_vol,p_zero,sigma_c,sigma_w);
```

```
function [opt_fun,Y,T,costs] =
optim_height(z_zero,reqd_vol,p_zero,sigma_c,sigma_w)
% Finds the meridional stress that gives a bag with the volume
specified in
% reqd_vol.  Used in a line search to find the optimum centre
height.

% Set constants.
g = 9.81;
M_molar_mass = 28.97e-3;
R_gas_const = 8.314472;

% Set gas temperature and depth of bag's base.
temp_gas = 273.15 + 5;
depth = 500;

% Set seawater density and calculate gas density.
rho_water = 1025;
rho_gas = M_molar_mass*(p_zero + rho_water*g*depth +
1e5)/(R_gas_const*temp_gas);

% Calculate buoyancy, b.
b = g*(rho_water - rho_gas);

% Guess at a close value of meridional stress.
T_guess = reqd_vol*b/(2*pi);

% Find the tension that gives the required volume with a bag of the
% required height in the centre.
T = fzero(@(T)
run_baggy_odes(T,z_zero,reqd_vol,b,sigma_c,sigma_w,p_zero,rho_water,
g,depth),T_guess);

% Find the value of the optimisation function, the shape data, and
the cost
% breakdown for the bag with the required volume and centre height.
[V_residual,opt_fun,Y,costs] =
run_baggy_odes(T,z_zero,reqd_vol,b,sigma_c,sigma_w,p_zero,rho_water,
g,depth);
```

```
function [V_residual,opt_fun,Y,costs] =
run_baggy_odes(T_zero,z_zero,...
    reqd_vol,b,sigma_c,sigma_w,p_zero,rho_water,g,depth)
% Sets the intial conditions and runs the ODE solver.  Used in a
root
% finder to find the meridional stress that gives a bag of the
required
% volume.

% Set the initial conditions for the ODEs.
alpha_zero = (0/180)*pi;
r_zero = 0;
A_zero = 0;
V_zero = 0;
Strength_merid_zero = 0;
Strength_circum_zero = 0;
Strength_ballast_zero = 0;
Total_hanging_force_zero = 0;

% Form the vector of initial conditions.
init_conds = [alpha_zero,z_zero,r_zero,A_zero,V_zero,T_zero,...

Strength_merid_zero,Strength_circum_zero,Strength_ballast_zero,...
    Total_hanging_force_zero];

% Set the range of values for s.
s_span = [0 100];

% Increase the number of steps in the integration.
options = odeset('Refine',50,'Events',@events);

% Use ode45 to solve the ODEs over s_span or until a terminating
event
% function is reached.
[s,Y,SE,YE,IE] = ode45(@(s,inp)
baggy_odes(s,inp,b,p_zero,sigma_c,...
    sigma_w,T_zero),s_span,init_conds,options);

% Calculate the difference between the volume enclosed and reqd_vol.
V = Y(end,5);
V_residual = V - reqd_vol;

% Set yield stress of reinforcement, cost of reinforcement per kg,
and
% density of reinforcement, and calculate the cost of reinforcement
per
% m^3.
yield_s = 250e6;
cost_reinf_per_kg = 0.5;
density_reinf = 7800;
cost_reinf_per_mcub = cost_reinf_per_kg*density_reinf;

% Calculate the cost of meridional reinforcement.
cost_reinf_merid = Y(end,7)*cost_reinf_per_mcub/yield_s; % Seamus's
price

% Calculate the cost of circumferential reinforcement.
cost_reinf_circum = Y(end,8)*cost_reinf_per_mcub/yield_s;
```

```
% Calculate the total cost of meridional and circumferential
reinforcement.
cost_reinf = cost_reinf_merid + cost_reinf_circum;

% Set the cost of surface per m^2.
%cost_surface_per_msq = 2; % Seamus's price
cost_surface_per_msq = 4; % Seamus's new price (May 09)
%cost_surface_per_msq = 27.5*0.64; % Maxim's price

% Calculate the cost of surface.
cost_surface = cost_surface_per_msq*Y(end,4);

% Set the cost of ballast per ton.
%cost_ballast_per_ton = 4; % Seamus's price
cost_ballast_per_ton = 20; % Seamus's new price (May 09)

% Calculate the cost of ballast mass.
cost_ballast_mass = cost_ballast_per_ton*b*V/(g*1000);

% Calculate the cost of vertical cables.
cost_vert_cables = 0.5*Y(end,9)*cost_reinf_per_mcub/yield_s;

% Calculate the total cost of ballast.
cost_ballast = cost_ballast_mass + cost_vert_cables;

% Form the vector of costs.
costs = [cost_reinf cost_surface cost_ballast];

% Calculate the cost of stored energy in million £ per GWh.
cost_per_en_stored = (cost_reinf + cost_surface + cost_ballast)/...
    ((1/(3600*1e3))*Y(end,5)*(p_zero + rho_water*g*depth + 1e5)*...
    log((p_zero + rho_water*g*depth + 1e5)/1e5));

% Set the optimisation function to the cost of stored energy.
opt_fun = cost_per_en_stored;



pen_factor = 1e6;

% Penalise an upward turn of the meridian.
opt_fun = opt_fun + pen_factor*max(0,max(diff(Y(:,2))));
if max(diff(Y(:,2))) > 0
    %disp('Upward turn of the meridian')
end

% Penalise a compressive meridional stress.
opt_fun = opt_fun + pen_factor*max(0,-min(Y(:,6)));
if -min(Y(:,6)) > 0
    %disp('Compressive meridional stress')
end

% Penalise a compressive circumferential stress.
opt_fun = opt_fun + pen_factor*max(0,-sigma_c);
if -sigma_c > 0
    %disp('Compressive circumferential stress')
end
```

```
% Penalise a compressive vertical stress.
opt_fun = opt_fun + pen_factor*max(0,-sigma_w);
if -sigma_w > 0
    %disp('Compressive vertical stress')
end

% Penalise compressive vertical stress in bag.
opt_fun = opt_fun + pen_factor*max(0,sigma_w*(max(Y(:,1))-pi/2));
if sigma_w*(max(Y(:,1))-pi/2) > 0
    %disp('Compressive vertical stress in bag')
end

% Penalise more hanging mass than total ballast required.
opt_fun = opt_fun + pen_factor*max(0,Y(end,10)-b*V);
if Y(end,10)-b*V > 0
    %disp('More hanging ballast than total ballast required')
end

% Penalise a negative depth.
opt_fun = opt_fun + pen_factor*max(0,-depth);
if -depth > 0
    %disp('Depth is negative')
end

% Penalise a negative centre height.
opt_fun = opt_fun + pen_factor*max(0,-z_zero);
if -z_zero > 0
    %disp('Centre height is negative')
end

% Penalise a crossing of the centreline.
opt_fun = opt_fun + pen_factor*max(0,-min(Y(:,3)));
if -min(Y(:,3)) > 0
    %disp('Crossing of the centreline')
end

%{
% Make a single point bag.
opt_fun = opt_fun + pen_factor*abs(Y(end,2));
if abs(Y(end,2)) > 0
    %disp('No single point at the seabed')
end
%}



function [value,isterminal,direction] = events(t,y)
% Locate the time when height passes through zero in a decreasing
direction
% and stop integration.

value = y(2);      % detect height = 0
isterminal = 1;    % stop the integration
direction = -1;    % negative direction


value(2) = y(3);      % detect radius = 0
isterminal(2) = 1;    % stop the integration
direction(2) = -1;    % negative direction
```

```
function outp = baggy_odes(t,inp,b,p_zero,sigma_c,sigma_w,T_zero)
% Sets the ODEs.  Used in an ODE solver to find bag shapes.

alpha = inp(1);
z = inp(2);
r = inp(3);
T = inp(6);

% Calculate the differential pressure, p, at the current height, z.
p = b*z + p_zero;

% Calculate the values of the differentials.
dalpha = (r*p - sigma_c*sin(alpha) - sigma_w*r*(cos(alpha))^2)/T;
dz = -sin(alpha);
dr = cos(alpha);
dA = 2*pi*abs(r);
dV = 2*pi*r*z*cos(alpha);
dT = sigma_c*cos(alpha) - sigma_w*r*cos(alpha)*sin(alpha);
dStrength_merid = 2*pi*T;
dStrength_circum = 2*pi*r*sigma_c;
dStrength_ballast = pi*r*z*cos(alpha)*sigma_w;
dTotal_hanging_force = 2*pi*r*cos(alpha)*sigma_w;

% Form the vector of outputs.
outp = [dalpha;dz;dr;dA;dV;dT;dStrength_merid;dStrength_circum;...
    dStrength_ballast;dTotal_hanging_force];
```

B. Matlab Code for Axisymmetric FEA

```
clear

global g
g = 9.81; % g.
rho_water = 1025; % Density of surrounding water.

prompt = {'Enter meridional length (m):','Enter base radius
(m):',...
    'Enter number of tendons:','Enter depth of base (m):',...
    'Enter final level of p = 0, as a fraction of the collapsed bag
height:',...
    'Enter ballast density (kg/m^3):','Unsealed base? (1 = Unsealed,
0 = Sealed):'};
dlg_title = 'Input for bag details';
num_lines = 1;
def = {'2.36','0','36','500','0.75','5100','0'};
answer = inputdlg(prompt,dlg_title,num_lines,def);

L = str2num(answer{1});
r = str2num(answer{2});
num_meridians = str2num(answer{3});
depth = str2num(answer{4});
p_fac = str2num(answer{5});
rho_ball = str2num(answer{6});
us_flag = str2num(answer{7});

prompt = {'Enter number of meridional elements:','Enter stiffness of
meridional elements (GPa):',...
    'Enter stiffness of circumferential elements (GPa):',...
    'Enter diameter of meridional elements (m):',...
    'Enter diameter of circumferential elements (m):',...
    'Enter membrane thickness (m):',...
    'Enter density of meridional elements (kg/m^3):','Enter density
of membrane (kg/m^3):'};
dlg_title = 'Input for bag details';
num_lines = 1;
def = {'50','200','0','0.02','0.02','0.001','7800','1522'};
answer = inputdlg(prompt,dlg_title,num_lines,def);

num_links = str2num(answer{1});
Em = str2num(answer{2})*1e9*ones(num_links,1);
Ec = str2num(answer{3})*1e9*ones(num_links+1,1);
Am = str2num(answer{4})^2*(pi/4)*ones(num_links,1);
Ac = str2num(answer{5})^2*(pi/4)*ones(num_links+1,1);
t = str2num(answer{6});
rho_ten = str2num(answer{7});
rho_mem = str2num(answer{8});

init_node_posns = gen_circle(r,L,num_links); % Create a circular
arc.
stress_pre = 1000*ones(num_links,1); % Set prestresses.
k = 0; % Set bending stiffness between meridional elements.
p_zero_min = p_fac*(-rho_water*g*(L-r)); % Set the final value for
p0.

% Find the shape of the pressurised bag.
```

```
[node_posns,pv_register,node_posns_register,alpha_register,C_su_regi
ster] =
main(num_links,L,r,stress_pre,Em,Am,Ec,Ac,num_meridians,depth,rho_wa
ter,p_zero_min,init_node_posns,k,rho_ten,rho_mem,t,rho_ball,us_flag)
;

% Plot the cross-section of the pressurised bag.
plot(node_posns(:,1),node_posns(:,2),'k','LineWidth',2)
hold on
plot(-node_posns(:,1),node_posns(:,2),'k','LineWidth',2)
xlabel('Radius, r (in m)')
ylabel('Height, h (in m)')
axis equal; v = axis; v = 1.1*v; axis(v)
grid on
```

```
function
[node_posns,pv_register,node_posns_register,alpha_register,C_su_regi
ster] =
main(num_links,L,r,stress_pre,Em,Am,Ec,Ac,num_meridians,depth,rho_wa
ter,p_zero_min,init_node_posns,k,rho_ten,rho_mem,t,rho_ball,us_flag,
alp)
% In this program, three variables may be incremented/decremented to
find a
% solution: load fraction, prestress, and base pressure difference.
% Due to the lack of stiffness in bending, even a very small initial
% load fraction can deform the structure quite considerably, and
% typically the structure then deforms very little from this initial
% deformed shape as the load fraction is increased, due to high
cable
% stiffness.

global g % Declare the global variables.

node_posns = init_node_posns; % Set current node positions.

% Calculate the angles between the elements.
link_lengths = ((init_node_posns(2:end,1)-init_node_posns(1:end-
1,1)).^2 + (init_node_posns(2:end,2)-init_node_posns(1:end-
1,2)).^2).^0.5;
alpha = acos((init_node_posns(2:end,1) - init_node_posns(1:end-
1,1))./link_lengths);
alpha = [pi/2 - alpha(1,1);alpha(1:end-1,1) + pi -
alpha(2:end,1);alpha(end,1)];

% Constraints matrix for a meridian with a top node only allowed to
move
% vertically and bottom node held in place.
constraints_mat = sparse(size(node_posns,1)*2,size(node_posns,1)*2-
3);
constraints_mat(2:end-2,:) = speye(2*size(node_posns,1)-3);

original_link_lengths = Em.*((init_node_posns(2:end,1)-
init_node_posns(1:end-1,1)).^2 + (init_node_posns(2:end,2)-
init_node_posns(1:end-1,2)).^2).^0.5./(stress_pre + Em);

R = init_node_posns(:,1); % Set the initial circumferential radii.

E_comp = 1e-6.*Em; % Set Young's modulus of links in compression.

% Assemble matrices of material properties.
mat_props_m = [stress_pre,Em,E_comp,Am];
mat_props_c = [Ec,Ac];

vol_ten = Am.*original_link_lengths; % Volume of meridional
elements.
vol_mem = original_link_lengths.*(init_node_posns(1:end-
1,1)+init_node_posns(2:end,1))/(2*num_meridians).*t; % Volume of
membrane section associated with each meridional element.
w_ten = rho_ten.*vol_ten*g; % Weight of meridional elements.
w_mem = rho_mem.*vol_mem*g; % Weight of membrane section associated
with each meridional element.

p_zero = rho_water*g*(L-r); % Initial differential pressure at base
(set to be high).
```

```
conv_crit = 1e-7; % Force residual convergence criterion.
disp_crit = 1e-3*sum(original_link_lengths); % Displacement
convergence criterion.


%----------------------------------------------
% Find deformed shape.

% Set initial fraction of full load to a very small value and then
find the
% load fraction that gives a max. change in angle from the current
set of
% element orientations to the new set of less than a certain amount
(e.g. 1
% degree).
init_load_fraction = 1e-11;
load_fraction =
find_load_frac(node_posns,p_zero,init_load_fraction,num_meridians,or
iginal_link_lengths,mat_props_m,constraints_mat,depth,rho_water,R,ma
t_props_c,alpha,k,w_ten,w_mem,us_flag);

% Loop to find equilibrium positions for increasing load fractions.
while load_fraction < 1
    % Display the current load fraction.
    disp('Load fraction = '); disp(load_fraction)

    % Find the equilibrium positions for the current load fraction.
    node_posns =
find_eqlbrm_posns(node_posns,constraints_mat,original_link_lengths,p
_zero,mat_props_m,load_fraction,conv_crit,num_meridians,depth,rho_wa
ter,R,mat_props_c,disp_crit,alpha,k,w_ten,w_mem,us_flag);

    % Increase the load fraction.
    load_fraction =
find_load_frac(node_posns,p_zero,load_fraction,num_meridians,origina
l_link_lengths,mat_props_m,constraints_mat,depth,rho_water,R,mat_pro
ps_c,alpha,k,w_ten,w_mem,us_flag);
end

load_fraction = 1; % Full load.

% Display the current load fraction.
disp('Load fraction = '); disp(load_fraction)

% Find the equilibrium positions with full load.
node_posns =
find_eqlbrm_posns(node_posns,constraints_mat,original_link_lengths,p
_zero,mat_props_m,load_fraction,conv_crit,num_meridians,depth,rho_wa
ter,R,mat_props_c,disp_crit,alpha,k,w_ten,w_mem,us_flag);



% Ensure that the final force residuals are correct by setting the
% displacement criterion to be very small.
disp_crit = 1e-9*sum(original_link_lengths);

% Find equilibrium positions.
node_posns =
find_eqlbrm_posns(node_posns,constraints_mat,original_link_lengths,p
_zero,mat_props_m,load_fraction,conv_crit,num_meridians,depth,rho_wa
ter,R,mat_props_c,disp_crit,alpha,k,w_ten,w_mem,us_flag);


%-----------------------------------------------
% This section can be used to find final equilibrium positions with
zero
% prestress.

stress_pre = 0*ones(num_links,1); % Set prestresses to zero.

% Calculate original element lengths.
original_link_lengths = Em.*((init_node_posns(2:end,1)-
init_node_posns(1:end-1,1)).^2 + (init_node_posns(2:end,2)-
init_node_posns(1:end-1,2)).^2).^0.5./(stress_pre + Em);

% Assemble matrix of revised meridian material properties.
mat_props_m = [stress_pre,Em,E_comp,Am];

% Find equilibrium positions.
node_posns =
find_eqlbrm_posns(node_posns,constraints_mat,original_link_lengths,p
_zero,mat_props_m,load_fraction,conv_crit,num_meridians,depth,rho_wa
ter,R,mat_props_c,disp_crit,alpha,k,w_ten,w_mem,us_flag);

% Set price of materials.
p_surf = 4; % Surface per m^2.
p_ball = 20/1000; % Ballast per kg.
p_reinf_kg = 2; % Reinforcement per kg.
sig_y = 400e6; % Yield stress of the reinforcement.
FoS = 2; % Factor of safety for the reinforcement.
p_reinf_Nm = (rho_ten*p_reinf_kg)/(sig_y/FoS);

% Calculate the materials cost per unit of stored energy (in £/MWh).
[obj,cost_mat,energy_Wh] =
obj_func(node_posns,num_meridians,L,p_zero,original_link_lengths,mat
_props_m,p_reinf_Nm,p_surf,p_ball,depth,rho_water,R,mat_props_c,alph
a,k,rho_ball);


%-----------------------------------------------
% This section can be used to find deformed shapes for decreasing
base
% pressure differences.

% Initialise data collection matrices.
pv_register = zeros(0,3);
node_posns_register = node_posns;
sv_register = zeros(0,2);
alpha_register = zeros(0,1);
C_su_register = zeros(0,2);

p_zero_max = p_zero; % Set the current p_zero to be the initial
p_zero.
p_zero_vec = linspace(p_zero_max,p_zero_min,401);
i = 1;

% Loop to find equilibrium positions for decreasing base pressure
% difference.
```

```
exitflag = 0;
while (i <= size(p_zero_vec,2)) && (exitflag == 0)

    p_zero = p_zero_vec(i);

    % Find the equilibrium positions for the current load fraction.
    [node_posns,constraints_mat] =
find_eqlbrm_posns(node_posns,constraints_mat,original_link_lengths,p
_zero,mat_props_m,load_fraction,conv_crit,num_meridians,depth,rho_wa
ter,R,mat_props_c,disp_crit,alpha,k,w_ten,w_mem,us_flag);

    link_lengths = sqrt(diff(node_posns(:,1)).^2 +
diff(node_posns(:,2)).^2);
    strains = (link_lengths-
original_link_lengths)./original_link_lengths;
    stress = Em.*strains + stress_pre;

    if min(stress) <= 0
        exitflag = 1; % Prepare to exit loop.
    else
        % Plot the shape found for current base pressure difference.
        figure(1)
        %hold on
        plot(node_posns(:,1),node_posns(:,2),'k','LineWidth',2)
        hold on
        plot(-node_posns(:,1),node_posns(:,2),'k','LineWidth',2)
        %plot([-1000;1000],(-p_zero/(rho_water*g))*ones(2,1),'r-.')
        hold off
        axis equal
        ymax = (r^2+L^2)^0.5;
        %axis([-0.75*ymax,0.75*ymax,0,1.5*ymax])
        axis([-ymax,ymax,0,ymax])
        grid on
        xlabel('Radius, r (in m)')
        ylabel('Height, h (in m)')
        pause(0.001)

        %mov(i+1) = getframe(gcf); % Capture a movie frame.

        % Volume of air contained in the bag.
        vol = sum(0.5*(node_posns(2:end,2) + node_posns(1:end-
1,2)).*(node_posns(2:end,1) - node_posns(1:end-
1,1))*2*pi*0.5.*(node_posns(2:end,1) + node_posns(1:end-1,1)));

        % Absolute pressure of the contained air.
        p_int = rho_water*g*depth + p_zero + 101.325e3;

        if us_flag == 1 && p_zero < 0
            vol = 0;
            j = 2;
            while (node_posns(j,2) > depth - (p_int-
101.325e3)/(rho_water*g))
                vol = vol + (node_posns(j-1,2)-
node_posns(j,2))*pi*(0.5*(node_posns(j,1)+node_posns(j-
1,1)))^2*sign(0.5*(node_posns(j,1)+node_posns(j-1,1)));
                j = j+1;
            end

            % Interpolate the final element.
            vol = vol + ((node_posns(j-1,2) - (depth - (p_int-
101.325e3)/(rho_water*g)))/(node_posns(j-1,2)-
node_posns(j,2)))*(node_posns(j-1,2)-
node_posns(j,2))*pi*(0.5*(node_posns(j,1)+node_posns(j-
1,1)))^2*sign(0.5*(node_posns(j,1)+node_posns(j-1,1)));
        end

        % Add to the pressure-volume register.
        pv_register =
[pv_register;vol,p_int,2*max(node_posns(:,1))];
        node_posns_register = [node_posns_register,node_posns(:,1)];
        alpha_register = [alpha_register;atan((node_posns(end-1,2)-
node_posns(end,2))/(node_posns(end-1,1)-node_posns(end,1)))];

        i = i+1; % Update counter.
    end
end

% Make sure p_zero is for the last pressure above the wrinkling.
if exitflag == 1
    disp('Negative stress encountered at next pressure.')

    % Find the equilibrium positions for the previous pressure.
    p_zero = p_zero_vec(i-1);
    [node_posns,constraints_mat] =
find_eqlbrm_posns(node_posns,constraints_mat,original_link_lengths,p
_zero,mat_props_m,load_fraction,conv_crit,num_meridians,depth,rho_wa
ter,R,mat_props_c,disp_crit,alpha,k,w_ten,w_mem,us_flag);
end
```

```
function [node_posns] = gen_circle(r,L,num_links)
% Generates num_links+1 nodes on a circle with base radius r and
meridional
% length L.

% Check the number of links is at least one.
if num_links < 1
    error('Number of links must be greater than or equal to one')
end

% Check that r >= 0.
if r < 0
    error('Base radius must be greater than or equal to zero')
end

% Check that L > 0.
if L <= 0
    error('Meridional length must be greater than zero')
end

% Check the number of links is an integer.
if num_links - floor(num_links) > 0
    error('Number of links must be an integer')
end

% Check the meridional length is greater than the base radius.
if L <= r
    error('Meridional length cannot be less than the base radius')
end

% Calculate the angle subtended by a circular meridian and the
radius of
% the meridian.
if r == 0
    alpha = 0;
    merid_ang = pi;
    R = L/pi;
else
    f = @(alpha)alpha - pi + L*sin(alpha)/r;
    alpha = fzero(f,[0,pi]);
    merid_ang = pi - alpha;
    R = r/sin(alpha);
end

% Create nodes along a circular meridian.
theta = (0:merid_ang/num_links:merid_ang)';
x = R*sin(theta);
y = R*cos(theta) + R*cos(alpha);

node_posns = [x,y];
```

```
function [load_frac_new] =
find_load_frac(node_posns,p_zero,load_frac,num_meridians,original_li
nk_lengths,mat_props,constraints_mat,depth,rho_water,R,mat_props_c,a
lpha,k,w_ten,w_mem,us_flag)
% Finds a new, increased load fraction that gives a maximum change
in angle
% of the elements just less than the specified angle criterion after
a
% single iteration of the Newton-Raphson method.

% Set the angle criterion (currently at 3 degrees).
ang_crit = (1/180)*pi;

% Begin with a new load fraction of twice the old one (when a = 1).
a = 1;
load_frac_new = (1 + a)*load_frac;

% Find the new node positions with the old load fraction.
posns1 =
find_new_posns(node_posns,p_zero,load_frac,num_meridians,original_li
nk_lengths,mat_props,constraints_mat,depth,rho_water,R,mat_props_c,a
lpha,k,w_ten,w_mem,us_flag);

% Find the new node positions with the new load fraction.
posns2 =
find_new_posns(node_posns,p_zero,load_frac_new,num_meridians,origina
l_link_lengths,mat_props,constraints_mat,depth,rho_water,R,mat_props
_c,alpha,k,w_ten,w_mem,us_flag);

% Calculate the difference in angle between the two sets of
positions.
angle_diff = atan((posns2(1:end-1,2) -
posns2(2:end,2))./(posns2(2:end,1) - posns2(1:end-1,1)))...
    - atan((posns1(1:end-1,2) - posns1(2:end,2))./(posns1(2:end,1) -
posns1(1:end-1,1)));

% If the maximum change in angle is greater than the angle
criterion.
if max(abs(angle_diff)) > ang_crit
    % Decrease the load fraction until the max. change in angle is
lower
    % than the criterion.
    while max(abs(angle_diff)) > ang_crit
        % Decrease the load fraction.
        a = 0.5*a;
        load_frac_new = (1 + a)*load_frac;

        % Find the new node positions with the new load fraction.
        posns2 =
find_new_posns(node_posns,p_zero,load_frac_new,num_meridians,origina
l_link_lengths,mat_props,constraints_mat,depth,rho_water,R,mat_props
_c,alpha,k,w_ten,w_mem,us_flag);

        % Calculate the difference in angle between the two sets of
        % positions.
        angle_diff = atan((posns2(1:end-1,2) -
posns2(2:end,2))./(posns2(2:end,1) - posns2(1:end-1,1)))...
            - atan((posns1(1:end-1,2) -
posns1(2:end,2))./(posns1(2:end,1) - posns1(1:end-1,1)));
    end
```

```
% If the maximum change in angle is less than or equal to the angle
% criterion.
else
    % Increase the load fraction until the max. change in angle is
greater
    % than the criterion.
    while max(abs(angle_diff)) <= ang_crit
        % Increase the load fraction.
        a = 2*a;
        load_frac_new = (1 + a)*load_frac;

        % Find the new node positions with the new load fraction.
        posns2 =
find_new_posns(node_posns,p_zero,load_frac_new,num_meridians,origina
l_link_lengths,mat_props,constraints_mat,depth,rho_water,R,mat_props
_c,alpha,k,w_ten,w_mem,us_flag);

        % Calculate the difference in angle between the two sets of
        % positions.
        angle_diff = atan((posns2(1:end-1,2) -
posns2(2:end,2))./(posns2(2:end,1) - posns2(1:end-1,1)))...
            - atan((posns1(1:end-1,2) -
posns1(2:end,2))./(posns1(2:end,1) - posns1(1:end-1,1)));
    end

    % Go back to the last load fraction that gave a max. change in
angle
    % lower than the criterion.
    a = 0.5*a;
    load_frac_new = (1 + a)*load_frac;
end
```

```
function [node_posns] =
find_new_posns(node_posns,p_zero,load_fraction,num_meridians,origina
l_link_lengths,mat_props_m,constraints_mat,depth,rho_water,R,mat_pro
ps_c,alpha,k,w_ten,w_mem,us_flag)
% Finds a new set of node positions using the Newton-Raphson method.

% Find differential pressure forces.
[F_dp,link_lengths,F_seabed] =
find_F_dp(node_posns,p_zero,load_fraction,num_meridians,depth,rho_wa
ter,w_ten,w_mem,us_flag);

% Find reaction forces.
[F_reac,T,Tc,Fb_tot] =
find_F_reac(link_lengths,original_link_lengths,mat_props_m,node_posn
s,R,mat_props_c,alpha,k);

% Find the force residuals.
residuals = F_dp - F_reac;
residuals_vec = reshape(residuals',[],1);
residuals_vec_short = constraints_mat'*residuals_vec;

% Find dimensionless residuals by dividing by the sum of the
tensions at
% each node.
%     dimless_residuals =
reshape((residuals./[([T(1:end);0]+[0;T(1:end)]),([T(1:end);0]+[0;T(
1:end)])])',[],1);
%     dimless_residuals_short = constraints_mat'*dimless_residuals;

% Find a new set of node positions and the nodal displacement vector
% by carrying out a single iteration of the Newton-Raphson solution
% method.
[node_posns,disps] = NR_iteration(node_posns,link_lengths,...

load_fraction,mat_props_m,original_link_lengths,constraints_mat,...

residuals_vec_short,num_meridians,p_zero,rho_water,R,mat_props_c,alp
ha,k,depth,us_flag);
```

```
function [node_posns,constraints_mat] =
find_eqlbrm_posns(node_posns,constraints_mat,original_link_lengths,p
_zero,mat_props_m,load_fraction,conv_crit,num_meridians,depth,rho_wa
ter,R,mat_props_c,disp_crit,alpha,k,w_ten,w_mem,us_flag)
% Finds the equilibrium position of the structure.

% Find differential pressure forces.
[F_dp,link_lengths,F_seabed] =
find_F_dp(node_posns,p_zero,load_fraction,num_meridians,depth,rho_wa
ter,w_ten,w_mem,us_flag);

% Find reaction forces.
[F_reac,T,Tc,Fb_tot] =
find_F_reac(link_lengths,original_link_lengths,mat_props_m,node_posn
s,R,mat_props_c,alpha,k);

%----------------------------------------------
% Find the force residuals.
residuals = F_dp - F_reac;
residuals_vec = reshape(residuals',[],1);
residuals_vec_short = constraints_mat'*residuals_vec;

% Find dimensionless residuals by dividing by the sum of the
tensions at
% each node - TAKE ABSOLUTE VALUES!!!!!
dimless_residuals =
reshape((residuals./[([T(1:end);0]+[0;T(1:end)]+Tc+F_seabed+Fb_tot),
([T(1:end);0]+[0;T(1:end)]+Tc+F_seabed+Fb_tot)])',[],1);
dimless_residuals_short = constraints_mat'*dimless_residuals;

% Initially set the nodal displacement vector to be a very large
scalar.
disps = 1e6*sum(original_link_lengths);

%----------------------------------------------
% Loop to find new node positions.
while ((max(abs(residuals_vec_short)) >= 1) |
(max(abs(dimless_residuals_short)) >= conv_crit)) & (max(abs(disps))
> disp_crit)
%blah = 0; while blah == 0
    % Find a new set of node positions and the nodal displacement
vector
    % by carrying out a single iteration of the Newton-Raphson
solution
    % method.
    [node_posns,disps] = NR_iteration(node_posns,link_lengths,...

load_fraction,mat_props_m,original_link_lengths,constraints_mat,...

residuals_vec_short,num_meridians,p_zero,rho_water,R,mat_props_c,alp
ha,k,depth,us_flag);
    %{
    % Constrain nodes against the seabed if necessary.
    for i = 1:size(node_posns,1)-1
        if node_posns(i,2) < 0 && node_posns(i+1,2) == 0
            node_posns(i,2) = 0; % Move node to the seabed.
            %constraints_mat(:,2*i-2:2*i-1) = []; % Constrain in y
and x.
            constraints_mat(:,2*i-1) = []; % Constrain in y.
            %keyboard
        end
    end
    %}
    % Find differential pressure forces.
    [F_dp,link_lengths,F_seabed] =
find_F_dp(node_posns,p_zero,load_fraction,num_meridians,depth,rho_wa
ter,w_ten,w_mem,us_flag);

    % Find reaction forces.
    [F_reac,T,Tc,Fb_tot] =
find_F_reac(link_lengths,original_link_lengths,mat_props_m,node_posn
s,R,mat_props_c,alpha,k);

    % Find the force residuals.
    residuals = F_dp - F_reac;
    residuals_vec = reshape(residuals',[],1);
    residuals_vec_short = constraints_mat'*residuals_vec;

    % Find dimensionless residuals by dividing by the sum of the
tensions at
    % each node.
    dimless_residuals =
reshape((residuals./[([T(1:end);0]+[0;T(1:end)]+Tc+F_seabed+Fb_tot),
([T(1:end);0]+[0;T(1:end)]+Tc+F_seabed+Fb_tot)])',[],1);
    dimless_residuals_short = constraints_mat'*dimless_residuals;
end
```

```
function [F_dp,link_lengths,Fsb_glo] =
find_F_dp(node_posns,p_zero,load_fraction,num_meridians,depth,rho_wa
ter,w_ten,w_mem,us_flag)
% Find pressure difference forces.

% Declare the global variables.
global g

F_dp = zeros(2*size(node_posns,1),1);
Fsb_glo = zeros(size(node_posns,1),1);

% Find link lengths.
link_lengths = ((node_posns(2:end,1)-node_posns(1:end-1,1)).^2 +
(node_posns(2:end,2)-node_posns(1:end-1,2)).^2).^0.5;

for i = 1:size(link_lengths,1)

    % Assign names to element node positions and length.
    x1 = node_posns(i,1);
    y1 = node_posns(i,2);
    x2 = node_posns(i+1,1);
    y2 = node_posns(i+1,2);
    l = link_lengths(i,1);

    % Find density of contained air stored at T degrees Kelvin.
    p_int = rho_water*g*depth + 1e5 + p_zero;
    M_air = 0.02897;
    R = 8.314472;
    T = 5 + 273.15;
    rho_air = p_int*M_air/(R*T);

    % Unit normal.
    n = (1/l)*[y1-y2;x2-x1];

    % Differential pressure at centroid.
    p = p_zero + (rho_water-rho_air)*g*0.5*(y1+y2);

    if us_flag == 1
        % Set any negative p to zero, for open base.
        p = max(p,0);
    end

    % Set dp to p0 if centroid is below the seabed.
%     if 0.5*(y1+y2) < 0
%         p = p_zero;
%     end

    % Swept area.
    Asw = pi*(x1+x2)*l/num_meridians;

    % Element load vector.
    F = 0.5*[n;n]*p*Asw + [0;-w_ten(i,1)/2;0;-w_ten(i,1)/2] + [0;-
w_mem(i,1)/2;0;-w_mem(i,1)/2];

    % Add the element load vector to the global load vector.
    F_dp(2*(i-1)+1:2*(i-1)+4,1) = F_dp(2*(i-1)+1:2*(i-1)+4,1) + F;

%     % Add the seabed forces.
%      k = -0.001*(p_zero+(rho_water-rho_air)*g*0.001)*0.5*Asw;
%
%      % Set the seabed force on the last node to zero.
%      if i == size(link_lengths,1)
%          Fsb = k*[0;1/y1;0;0];
%      % All other nodes.
%      else
%          Fsb = k*[0;1/y1;0;1/y2];
%      end
%
%      Fsb_glo(2*(i-1)+1:2*(i-1)+4,1) = Fsb_glo(2*(i-1)+1:2*(i-
1)+4,1) + Fsb;
    %F_dp(2*(i-1)+1:2*(i-1)+4,1) = F_dp(2*(i-1)+1:2*(i-1)+4,1) +
Fsb;

end

k_ce = 0; % TEMPORARY - MOVE TO RUN_MAIN OR MAIN!!!
Fce = zeros(size(node_posns,1),1);
for i = 1:size(node_posns,1)
    if node_posns(i,1) < 0
        Fce(i,1) = k_ce*(node_posns(i,1)^2); % Calculate central
reaction.
    end
end
F_dp(1:2:end-1,1) = F_dp(1:2:end-1,1) + Fce; % Add seabed forces.

% Add seabed force proportional to the depth of the node below the
seabed.
k_sb = 1e9; % TEMPORARY - MOVE TO RUN_MAIN OR MAIN!!!
for i = 1:size(node_posns,1)
    if node_posns(i,2) < 0
        Fsb_glo(i,1) = k_sb*(node_posns(i,2)^2); % Calculate seabed
force.

%          % Make sure the seabed force isn't greater than the
downward load.
%          if Fsb_glo(i,1) > -F_dp(2*(i-1)+2,1)
%              Fsb_glo(i,1) = -F_dp(2*(i-1)+2,1);
%          end
    end
end
F_dp(2:2:end,1) = F_dp(2:2:end,1) + Fsb_glo; % Add seabed forces.


% Multiply by the load fraction.
F_dp = F_dp*load_fraction;

% Reshape into a matrix.
F_dp = reshape(F_dp,2,[])';
% Fsb_glo = Fsb_glo(2:2:end,1);
% Fsb_glo = zeros(size(node_posns,1),1);

%{
% Add a vertically upwards force to each node inversely proportional
to the
% distance of the node from the seabed.
%k = 0.001*(p_zero + g*(rho_water -
rho_air)*0.001)*0.5*([swept_areas(:,1);0] + [0;swept_areas(:,1)]);
%F_seabed = load_fraction*k(1:end-1,:)./abs(node_posns(1:end-1,2));
k = -0.001*(p_zero + g*(rho_water -
rho_air)*0.001)*0.5*([swept_areas(:,1);0] + [0;swept_areas(:,1)]);
F_seabed = load_fraction*k(1:end-1,:)./node_posns(1:end-1,2);
F_seabed = [F_seabed;0];
%F_seabed = zeros(size(F_seabed,1)+1,1);
F_dp(:,2) = F_dp(:,2) + F_seabed;
%}

% Add the mass of the top bulkhead.
%F_dp(1,2) = F_dp(1,2) - load_fraction*m_upp*g/num_meridians;

% Add the mass of the bottom bulkhead.
%F_dp(end,2) = F_dp(end,2) - m_low*g/num_meridians;
```

```
function [F_reac,Tm,Tc,Fb_tot] =
find_F_reac(link_lengths,original_link_lengths,mat_props_m,node_posn
s,R,mat_props_c,alpha,k);
% Find reaction forces.

% Calculate cable strains.
strains = (link_lengths-
original_link_lengths)./original_link_lengths;

% Unpack the material properties matrix.
stress_pre = mat_props_m(:,1);
E = mat_props_m(:,2);
E_comp = mat_props_m(:,3);
A = mat_props_m(:,4);

stress = E.*strains + stress_pre;

% Initialise the Jacobian as an empty sparse matrix.
F_reac = zeros(2*(size(link_lengths,1)+1),1);
Tm = zeros(size(link_lengths,1),1);

for i = 1:size(link_lengths,1)
    x1 = node_posns(i,1);
    y1 = node_posns(i,2);
    x2 = node_posns(i+1,1);
    y2 = node_posns(i+1,2);
    l = link_lengths(i,1);
    L = original_link_lengths(i,1);

    dl_du = (1/l)*((x1-x2)*[1 0 -1 0] + (y1-y2)*[0 1 0 -1]);
    dstrain_du = (1/L)*dl_du;

    if stress(i,1) >= 0
        Tm(i,1) = A(i,1)*stress(i,1);
    else
        limit = -1e-6*E(i,1);

        a = limit/(-pi/2);
        b = E(i,1)/a;

        % Sets b to zero if E = 0.
        if E(i,1) == 0
            b = 0;
        end

        Tm(i,1) = A(i,1)*a*atan(b*strains(i,1));
    end

    p = (1/l)*[x1-x2;y1-y2;x2-x1;y2-y1];

    F = Tm(i,1)*p;

    F_reac((i-1)*2+1:(i-1)*2+4,1) = F_reac((i-1)*2+1:(i-1)*2+4,1) +
F;
end

F_reac = reshape(F_reac,2,[])';
```

```
% ONLY HERE TO SATISFY OUTPUT ARGUMENT LIST.
% Find the unit vector parallel to each link from node 2 to node 1.
unit_parallels = [(node_posns(1:end-1,1)-
node_posns(2:end,1))./link_lengths,...
    (node_posns(1:end-1,2)-node_posns(2:end,2))./link_lengths];


%-----------------------------------------------------------------
% Circumferential forces.
% Unpack the circumferential material properties matrix.
Ec = mat_props_c(:,1);
Ac = mat_props_c(:,2);

% Initialise the circumferential force vector as zeros.
Fc = zeros(size(Ec,1),1);
Tc = zeros(size(Ec,1),1);

% Use a small fraction of the Young's modulus if the circumferential
force
% is negative.
for i = 1:size(Fc,1)
    if R(i,1) == 0
        Tc(i,1) = 0;
        Fc(i,1) = 0;
    else
        if node_posns(i,1) > R(i,1)
            Tc(i,1) = Ec(i,1)*Ac(i,1)*(node_posns(i,1)-
R(i,1))/R(i,1);
            Fc(i,1) = 2*pi*Tc(i,1);
        else
            %Tc(i,1) = 0;
            %Fc(i,1) = 2*pi*Tc(i,1);

            limit = -1e-12*Ec(i,1);

            a = limit/(-pi/2);
            b = Ec(i,1)/a;

            % Sets b to zero if Ec = 0.
            if Ec(i,1) == 0
                b = 0;
            end

            Tc(i,1) = Ac(i,1).*a*atan(b*(node_posns(i,1)-
R(i,1))/R(i,1));
            Fc(i,1) = 2*pi*Tc(i,1);
        end
    end
end

% Add the circumferential forces to the nodal reaction force matrix.
F_reac = F_reac + [Fc,zeros(size(Fc,1),1)];


%-----------------------------------------------------------------
% Bending forces.

% Initialise the matrix of bending forces and vector of sum of
absolute
```

```
% values of bending forces as zeros.
%Fb = zeros(size(node_posns,1),2);
Fb_tot = zeros(size(node_posns,1),1);
%{
for i = 1:size(link_lengths,1)
    % Deal with the first element.
    if i == 1
        % Assign names to the node positions.
        x1 = node_posns(i,1);
        y1 = node_posns(i,2);
        x2 = node_posns(i+1,1);
        y2 = node_posns(i+1,2);
        x3 = node_posns(i+2,1);
        y3 = node_posns(i+2,2);

        % Calculate the angle between the element and the vertical
        % (downwards) and the angle between the element and the next
        % element.
        %theta1 = pi/2 - acos((x2 - x1)/link_lengths(i,1));
        %theta2 = acos((x2 - x1)/link_lengths(i,1)) + pi - acos((x3
- x2)/link_lengths(i+1,1));
        theta1 = pi/2 - sign(y1-y2)*acos((x2 -
x1)/link_lengths(i,1));
        theta2 = sign(y1-y2)*acos((x2 - x1)/link_lengths(i,1)) + pi
- sign(y2-y3)*acos((x3 - x2)/link_lengths(i+1,1));

    % Deal with the last element.
    elseif i == size(link_lengths,1)
        % Assign names to the node positions.
        x0 = node_posns(i-1,1);
        y0 = node_posns(i-1,2);
        x1 = node_posns(i,1);
        y1 = node_posns(i,2);
        x2 = node_posns(i+1,1);
        y2 = node_posns(i+1,2);

        % Calculate the angle between the element and the previous
element
        % and the angle between the element and the horizontal.
        %theta1 = acos((x1 - x0)/link_lengths(i-1,1)) + pi -
acos((x2 - x1)/link_lengths(i,1));
        %theta2 = acos((x2 - x1)/link_lengths(i,1));
        theta1 = sign(y0-y1)*acos((x1 - x0)/link_lengths(i-1,1)) +
pi - sign(y1-y2)*acos((x2 - x1)/link_lengths(i,1));
        theta2 = sign(y1-y2)*acos((x2 - x1)/link_lengths(i,1));

    % Deal with all other elements.
    else
        % Assign names to the node positions.
        x0 = node_posns(i-1,1);
        y0 = node_posns(i-1,2);
        x1 = node_posns(i,1);
        y1 = node_posns(i,2);
        x2 = node_posns(i+1,1);
        y2 = node_posns(i+1,2);
        x3 = node_posns(i+2,1);
        y3 = node_posns(i+2,2);

        % Calculate the angles between the element and the previous
and
```

```
        % next elements.
        %theta1 = acos((x1 - x0)/link_lengths(i-1,1)) + pi -
acos((x2 - x1)/link_lengths(i,1));
        %theta2 = acos((x2 - x1)/link_lengths(i,1)) + pi - acos((x3
- x2)/link_lengths(i+1,1));
        theta1 = sign(y0-y1)*acos((x1 - x0)/link_lengths(i-1,1)) +
pi - sign(y1-y2)*acos((x2 - x1)/link_lengths(i,1));
        theta2 = sign(y1-y2)*acos((x2 - x1)/link_lengths(i,1)) + pi
- sign(y2-y3)*acos((x3 - x2)/link_lengths(i+1,1));
    end

    alpha1 = alpha(i,1);
    alpha2 = alpha(i+1,1);

    % Calculate the bending force required at each end of the
element.
    F1 = k*(alpha2 - theta2)/link_lengths(i,1);
    F2 = k*(alpha1 - theta1)/link_lengths(i,1);

    % Add the element bending force to the bending force matrix.
    Fb(i,1) = Fb(i,1) - F1*(y1 - y2)/link_lengths(i,1);
    Fb(i,2) = Fb(i,2) - F1*(x2 - x1)/link_lengths(i,1);

    Fb(i+1,1) = Fb(i+1,1) - F2*(y1 - y2)/link_lengths(i,1);
    Fb(i+1,2) = Fb(i+1,2) - F2*(x2 - x1)/link_lengths(i,1);

    Fb_tot(i,1) = Fb_tot(i,1) + abs(F1);
    Fb_tot(i+1,1) = Fb_tot(i+1,1) + abs(F2);
end

% Add the bending force matrix to the reaction force matrix.
F_reac = F_reac + Fb;
%}


% Reshape the matrix of nodal reaction forces into a vector of the
form
% Fx1;Fy1;Fx2;Fy2;...
F_reac_vec = reshape(F_reac',[],1);
```

```
function [node_posns,disps] =
NR_iteration(node_posns,link_lengths,...

load_fraction,mat_props,original_link_lengths,constraints_mat,...

residuals_vec_short,num_meridians,p_zero,rho_water,R,mat_props_c,alp
ha,k,depth,us_flag);
% Carries out a single iteration of the Newton-Raphson solution
method.

% Find the Jacobian of the differential pressure forces.
J_F_dp = find_J_F_dp(node_posns,link_lengths,...
    load_fraction,num_meridians,p_zero,rho_water,depth,us_flag);

% Find the Jacobian of the reaction forces.
J_F_reac = find_J_F_reac(mat_props,node_posns,link_lengths,...
    original_link_lengths,R,mat_props_c,alpha,k);

% Find the Jacobian of the force residuals.
J_resid = J_F_dp - J_F_reac;

% Constrain the Jacobian of the residuals.
J_resid_constr = constraints_mat'*(J_resid)*constraints_mat;

% Find the displacement factor (less than or equal to 1) that
ensures that
% nodes don't move into the seabed.
node_disps = reshape(constraints_mat*(J_resid_constr\(-
residuals_vec_short)),2,[])';

disp_factor = 1;

% while min(node_posns(:,2) + disp_factor*node_disps(:,2)) < 0
%    disp_factor = disp_factor/2;
% end
%disp(disp_factor)

disps = disp_factor*reshape(constraints_mat*(J_resid_constr\(-
residuals_vec_short)),2,[])';

% Update the node positions using the Newton-Raphson method.
node_posns = node_posns + disps;
```

```
function J_F_dp = find_J_F_dp(node_posns,link_lengths,...
    pres_fraction,num_meridians,p_zero,rho_water,depth,us_flag)
% Finds the Jacobian of the differential pressure forces.

% Declare the global variables.
global g

J_F_dp = sparse(2*size(node_posns,1),2*size(node_posns,1));

for i = 1:size(link_lengths,1)

    % Assign names to element node positions and length.
    x1 = node_posns(i,1);
    y1 = node_posns(i,2);
    x2 = node_posns(i+1,1);
    y2 = node_posns(i+1,2);
    l = link_lengths(i,1);

    % Find density of contained air stored at T degrees Kelvin.
    p_int = rho_water*g*depth + 1e5 + p_zero;
    M_air = 0.02897;
    R = 8.314472;
    T = 5 + 273.15;
    rho_air = p_int*M_air/(R*T);

    % Element length.
    dl_du = (1/l)*((x2-x1)*[-1 0 1 0] + (y2-y1)*[0 -1 0 1]);

    % Unit normal.
    n = (1/l)*[y1-y2;x2-x1];
    dn_du = [y1-y2;x2-x1]*(-1/l^2)*dl_du + (1/l)*[0 1 0 -1;-1 0 1
0];

    % Differential pressure at centroid.
    p = p_zero + (rho_water-rho_air)*g*0.5*(y1+y2);
    dp_du = (rho_water-rho_air)*g*0.5*[0 1 0 1];

    if us_flag == 1
        % Set any negative p to zero, for open base.
        p = max(p,0);

        if p == 0
            dp_du = [0 0 0 0];
        end
    end

%     % Set dp to p0 if centroid is below seabed.
%     if 0.5*(y1+y2) < 0
%         p = p_zero;
%         dp_du = zeros(1,4);
%     end

    % Swept area.
    Asw = pi*(x1+x2)*l/num_meridians;
    dAsw_du = (pi*(x1+x2)*dl_du + pi*l*[1 0 1 0])/num_meridians;

    % Element load stiffness matrix.
    dF_du = 0.5*p*Asw*[dn_du;dn_du] + 0.5*[n;n]*Asw*dp_du +
0.5*[n;n]*p*dAsw_du;
```

```
    % Add the element load stiffness matrix to the global load 
stiffness
    % matrix.
    J_F_dp(2*(i-1)+1:2*(i+1),2*(i-1)+1:2*(i+1)) = ...
        J_F_dp(2*(i-1)+1:2*(i+1),2*(i-1)+1:2*(i+1)) + dF_du;

    % Include effect of upwards force on nodes to keep them from 
crossing into
    % the seabed.
    k = -0.001*(p_zero+(rho_water-rho_air)*g*0.001)*0.5*Asw;
    dk_du = -0.001*(p_zero+(rho_water-rho_air)*g*0.001)*0.5*dAsw_du;

    % Set change in force at last node to zero.
    if i == size(link_lengths,1)
        dFsb_du = k*[0 0 0 0;0 -1/y1^2 0 0;0 0 0 0;0 0 0 0]...
            + [0;1/y1;0;0]*dk_du;
    % All other nodes.
    else
        dFsb_du = k*[0 0 0 0;0 -1/y1^2 0 0;0 0 0 0;0 0 0 -1/y2^2]...
            + [0;1/y1;0;1/y2]*dk_du;
    end

%     J_F_dp(2*(i-1)+1:2*(i+1),2*(i-1)+1:2*(i+1)) = ...
%         J_F_dp(2*(i-1)+1:2*(i+1),2*(i-1)+1:2*(i+1)) + dFsb_du;

end

% Add stiffness of bag (if meeting in the middle).
k_ce = 0; % TEMPORARY - MOVE TO RUN_MAIN OR MAIN!!!
for i = 1:size(node_posns,1)
    if node_posns(i,1) < 0
        J_F_dp(2*(i-1)+1,2*(i-1)+1) = J_F_dp(2*(i-1)+1,2*(i-1)+1) + 
k_ce*2*node_posns(i,1); % Calculate central reaction.
    end
end

% Add stiffness of seabed to nodes currently below the seabed.
k_sb = 1e9; % TEMPORARY!!! MOVE TO MAIN OR RUN_MAIN.
for i = 1:size(node_posns,1)
    if node_posns(i,2) < 0
        J_F_dp(2*(i-1)+2,2*(i-1)+2) = J_F_dp(2*(i-1)+2,2*(i-1)+2) + 
k_sb*2*node_posns(i,2);
    end
end

% Multiply by the load fraction.
J_F_dp = J_F_dp*pres_fraction;
```

```
function J_F_reac =
find_J_F_reac(mat_props,node_posns,link_lengths,...
    original_link_lengths,R,mat_props_c,alpha,k)
% Finds the Jacobian of the reaction forces.

% Unpack the material properties matrix.
stress_pre = mat_props(:,1);
E = mat_props(:,2);
E_comp = mat_props(:,3);
A = mat_props(:,4);

% Calculate cable strains.
strains = (link_lengths-
original_link_lengths)./original_link_lengths;

% Calculate the stress vector.
stress = E.*strains + stress_pre;

% Initialise the Jacobian as an empty sparse matrix.
J_F_reac =
sparse(2*(size(link_lengths,1)+1),2*(size(link_lengths,1)+1));

for i = 1:size(link_lengths,1)
    x1 = node_posns(i,1);
    y1 = node_posns(i,2);
    x2 = node_posns(i+1,1);
    y2 = node_posns(i+1,2);
    l = link_lengths(i,1);
    L = original_link_lengths(i,1);

    dl_du = (1/l)*((x1-x2)*[1 0 -1 0] + (y1-y2)*[0 1 0 -1]);
    dstrain_du = (1/L)*dl_du;

    if stress(i,1) >= 0
        T = A(i,1)*stress(i,1);
        dstress_du = E(i,1)*dstrain_du;
        dT_du = A(i,1)*dstress_du;
    else
        limit = -1e-6*E(i,1);

        a = limit/(-pi/2);
        b = E(i,1)/a;

        % Sets b to zero if E = 0.
        if E(i,1) == 0
            b = 0;
        end

        T = A(i,1)*a*atan(b*strains(i,1));
        dT_du = (A(i,1)*a*b/(1+(b*strains(i,1))^2))*dstrain_du;
    end

    p = (1/l)*[x1-x2;y1-y2;x2-x1;y2-y1];
    dp_du = (1/l)*[1 0 -1 0;0 1 0 -1;-1 0 1 0;0 -1 0 1]...
        + [x1-x2;y1-y2;x2-x1;y2-y1]*(-1/l^2)*dl_du;

    dF_du = p*dT_du + T*dp_du;

    J_F_reac((i-1)*2+1:(i-1)*2+4,(i-1)*2+1:(i-1)*2+4) = ...
```

```
        J_F_reac((i-1)*2+1:(i-1)*2+4,(i-1)*2+1:(i-1)*2+4) + dF_du;
end



% CIRCUMFERENTIAL STRESS

% Unpack the circumferential material properties matrix.
Ec = mat_props_c(:,1);
Ac = mat_props_c(:,2);

% Initialise the circumferential force vector as zeros.
dFc_dx = zeros(size(Ec,1),1);

% Use a small fraction of the Young's modulus if the circumferential
force
% is negative.
for i = 1:size(dFc_dx,1)
    if R(i,1) == 0
        dFc_dx(i,1) = 0;
    else
        if node_posns(i,1) > R(i,1)
            dFc_dx(i,1) = 2*pi*Ec(i,1)*Ac(i,1)/R(i,1);
        else
            %dFc_dx(i,1) = 0;

            limit = -1e-12*Ec(i,1);

            a = limit/(-pi/2);
            b = Ec(i,1)/a;

            % Sets b to zero if Ec = 0.
            if Ec(i,1) == 0
                b = 0;
            end

            dstrains_dx = 1/R(i,1);
            dFc_dx(i,1) =
2*pi*b*dstrains_dx*Ac(i,1)*a/(1+(b*(node_posns(i,1)-
R(i,1))/R(i,1))^2);
        end
    end
end

% Differentiate the vertical component of circumferential force.
% dFc_dy = zeros(size(dFc_dx,1),1);

% Add the differentiated circumferential forces to the stiffness
matrix.
for i = 1:2:size(J_F_reac,1)
    J_F_reac(i,i) = J_F_reac(i,i) + dFc_dx(1+(i-1)/2,1);
end

% for i = 2:2:size(J_F_reac,1)
%     J_F_reac(i,i) = J_F_reac(i,i) + dFc_dy(i/2,1);
% end
```

```
%{
% BENDING
for i = 1:size(link_lengths,1)
    % Assign names to the bending stiffness at each end of the
element.
    k1 = k;
    k2 = k;

    % Assign names to the undeformed angles between the element and
the
    % previous and next elements.
    alpha1 = alpha(i,1);
    alpha2 = alpha(i+1,1);

    % Deal with the first element.
    if i == 1
        % Assign names to the node positions.
        x1 = node_posns(i,1);
        y1 = node_posns(i,2);
        x2 = node_posns(i+1,1);
        y2 = node_posns(i+1,2);
        x3 = node_posns(i+2,1);
        y3 = node_posns(i+2,2);

        % Assign names to the element lengths.
        l = link_lengths(i,1);
        l2 = link_lengths(i+1,1);

        % Differentiate the element lengths.
        dl_du = (1/l)*((x2-x1)*[-1 0 1 0 0 0] + (y2-y1)*[0 -1 0 1 0
0]);
        dl2_du = (1/l2)*((x3-x2)*[0 0 -1 0 1 0] + (y3-y2)*[0 0 0 -1
0 1]);

        % Calculate theta, the angles between the element and the
previous
        % and next elements.
        %theta1 = pi/2 - acos((x2 - x1)/l);
        %theta2 = acos((x2 - x1)/l) + pi - acos((x3 - x2)/l2);
        theta1 = pi/2 - sign(y1-y2)*acos((x2 - x1)/l);
        theta2 = sign(y1-y2)*acos((x2 - x1)/l) + pi - sign(y2-
y3)*acos((x3 - x2)/l2);

        % Differentiate theta.
        %dtheta1_du = (1/sqrt(1-((x2-x1)/l)^2))*((x2-x1)*(-
1/l^2)*dl_du + (1/l)*[-1 0 1 0 0 0]);

        %dtheta2_du = (-1/sqrt(1-((x2-x1)/l)^2))*((x2-x1)*(-
1/l^2)*dl_du + (1/l)*[-1 0 1 0 0 0])...
        %    + (1/sqrt(1-((x3-x2)/l2)^2))*((x3-x2)*(-1/l2^2)*dl2_du
+ (1/l2)*[0 0 -1 0 1 0]);

        dtheta1_du = sign(y1-y2)*(1/sqrt(1-((x2-x1)/l)^2))*((x2-
x1)*(-1/l^2)*dl_du + (1/l)*[-1 0 1 0 0 0]);

        dtheta2_du = sign(y1-y2)*(-1/sqrt(1-((x2-x1)/l)^2))*((x2-
x1)*(-1/l^2)*dl_du + (1/l)*[-1 0 1 0 0 0])...
            + sign(y2-y3)*(1/sqrt(1-((x3-x2)/l2)^2))*((x3-x2)*(-
1/l2^2)*dl2_du + (1/l2)*[0 0 -1 0 1 0]);
```

```
        % Find the unit normal to the element.
        n = (1/l)*[y1-y2;x2-x1];

        % Differentiate the unit normal with respect to the node
positions.
        dn_du = [y1-y2;x2-x1]*(-1/l^2)*dl_du + (1/l)*[0 1 0 -1 0 0;-
1 0 1 0 0 0];

        % Differentiate the bending force vector.
        dFb_du = (1/l)*[k2*(theta2 - alpha2)*dn_du;k1*(theta1 -
alpha1)*dn_du]...
            + [n*k2*(theta2 - alpha2);n*k1*(theta1 - alpha1)]*(-
1/l^2)*dl_du...
            + (1/l)*[n*k2*dtheta2_du;n*k1*dtheta1_du];

        % Add the differentiated bending forces to the stiffness
matrix.
        J_F_reac(2*(i-1)+1:2*(i-1)+4,2*(i-1)+1:2*(i-1)+6) =
J_F_reac(2*(i-1)+1:2*(i-1)+4,2*(i-1)+1:2*(i-1)+6) + dFb_du;

    % Deal with the last element.
    elseif i == size(link_lengths,1)
        % Assign names to the node positions.
        x0 = node_posns(i-1,1);
        y0 = node_posns(i-1,2);
        x1 = node_posns(i,1);
        y1 = node_posns(i,2);
        x2 = node_posns(i+1,1);
        y2 = node_posns(i+1,2);

        % Assign names to the element lengths.
        l0 = link_lengths(i-1,1);
        l = link_lengths(i,1);

        % Differentiate the element lengths.
        dl_du = (1/l)*((x2-x1)*[0 0 -1 0 1 0] + (y2-y1)*[0 0 0 -1 0
1]);
        dl0_du = (1/l0)*((x1-x0)*[-1 0 1 0 0 0] + (y1-y0)*[0 -1 0 1
0 0]);

        % Calculate theta, the angles between the element and the
previous
        % and next elements.
        %theta1 = acos((x1 - x0)/l0) + pi - acos((x2 - x1)/l);
        %theta2 = acos((x2 - x1)/l);

        theta1 = sign(y0-y1)*acos((x1 - x0)/l0) + pi - sign(y1-
y2)*acos((x2 - x1)/l);
        theta2 = sign(y1-y2)*acos((x2 - x1)/l);

        % Differentiate theta.
        %dtheta1_du = (-1/sqrt(1-((x1-x0)/l0)^2))*((x1-x0)*(-
1/l0^2)*dl0_du + (1/l0)*[-1 0 1 0 0 0])...
        %    + (1/sqrt(1-((x2-x1)/l)^2))*((x2-x1)*(-1/l^2)*dl_du +
(1/l)*[0 0 -1 0 1 0]);

        %dtheta2_du = (-1/sqrt(1-((x2-x1)/l)^2))*((x2-x1)*(-
1/l^2)*dl_du + (1/l)*[0 0 -1 0 1 0]);
```

```
        dtheta1_du = sign(y0-y1)*(-1/sqrt(1-((x1-x0)/l0)^2))*((x1-
x0)*(-1/l0^2)*dl0_du + (1/l0)*[-1 0 1 0 0 0])...
            + sign(y1-y2)*(1/sqrt(1-((x2-x1)/l)^2))*((x2-x1)*(-
1/l^2)*dl_du + (1/l)*[0 0 -1 0 1 0]);

        dtheta2_du = sign(y1-y2)*(-1/sqrt(1-((x2-x1)/l)^2))*((x2-
x1)*(-1/l^2)*dl_du + (1/l)*[0 0 -1 0 1 0]);

        % Find the unit normal to the element.
        n = (1/l)*[y1-y2;x2-x1];

        % Differentiate the unit normal with respect to the node
positions.
        dn_du = [y1-y2;x2-x1]*(-1/l^2)*dl_du + (1/l)*[0 0 0 1 0 -1;0
0 -1 0 1 0];

        % Differentiate the bending force vector.
        dFb_du = (1/l)*[k2*(theta2 - alpha2)*dn_du;k1*(theta1 -
alpha1)*dn_du]...
            + [n*k2*(theta2 - alpha2);n*k1*(theta1 - alpha1)]*(-
1/l^2)*dl_du...
            + (1/l)*[n*k2*dtheta2_du;n*k1*dtheta1_du];

        % Add the differentiated bending forces to the stiffness
matrix.
        J_F_reac(2*(i-1)+1:2*(i-1)+4,2*(i-1)-1:2*(i-1)+4) =
J_F_reac(2*(i-1)+1:2*(i-1)+4,2*(i-1)-1:2*(i-1)+4) + dFb_du;

    % Deal with all other elements.
    else
        % Assign names to the node positions.
        x0 = node_posns(i-1,1);
        y0 = node_posns(i-1,2);
        x1 = node_posns(i,1);
        y1 = node_posns(i,2);
        x2 = node_posns(i+1,1);
        y2 = node_posns(i+1,2);
        x3 = node_posns(i+2,1);
        y3 = node_posns(i+2,2);

        % Assign names to the element lengths.
        l0 = link_lengths(i-1,1);
        l = link_lengths(i,1);
        l2 = link_lengths(i+1,1);

        % Differentiate the element lengths.
        dl_du = (1/l)*((x2-x1)*[0 0 -1 0 1 0 0 0] + (y2-y1)*[0 0 0 -
1 0 1 0 0]);
        dl0_du = (1/l0)*((x1-x0)*[-1 0 1 0 0 0 0 0] + (y1-y0)*[0 -1
0 1 0 0 0 0]);
        dl2_du = (1/l2)*((x3-x2)*[0 0 0 0 -1 0 1 0] + (y3-y2)*[0 0 0
0 0 -1 0 1]);

        % Calculate theta, the angles between the element and the
previous
        % and next elements.
        %theta1 = acos((x1 - x0)/l0) + pi - acos((x2 - x1)/l);
        %theta2 = acos((x2 - x1)/l) + pi - acos((x3 - x2)/l2);
        theta1 = sign(y0-y1)*acos((x1 - x0)/l0) + pi - sign(y1-
y2)*acos((x2 - x1)/l);
        theta2 = sign(y1-y2)*acos((x2 - x1)/l) + pi - sign(y2-
y3)*acos((x3 - x2)/l2);

        % Differentiate theta.
        %dtheta1_du = (-1/sqrt(1-((x1-x0)/l0)^2))*((x1-x0)*(-
1/l0^2)*dl0_du + (1/l0)*[-1 0 1 0 0 0 0 0])...
        %    + (1/sqrt(1-((x2-x1)/l)^2))*((x2-x1)*(-1/l^2)*dl_du +
(1/l)*[0 0 -1 0 1 0 0 0]);

        %dtheta2_du = (-1/sqrt(1-((x2-x1)/l)^2))*((x2-x1)*(-
1/l^2)*dl_du + (1/l)*[0 0 -1 0 1 0 0 0])...
        %    + (1/sqrt(1-((x3-x2)/l2)^2))*((x3-x2)*(-1/l2^2)*dl2_du
+ (1/l2)*[0 0 0 0 -1 0 1 0]);

        dtheta1_du = sign(y0-y1)*(-1/sqrt(1-((x1-x0)/l0)^2))*((x1-
x0)*(-1/l0^2)*dl0_du + (1/l0)*[-1 0 1 0 0 0 0 0])...
            + sign(y1-y2)*(1/sqrt(1-((x2-x1)/l)^2))*((x2-x1)*(-
1/l^2)*dl_du + (1/l)*[0 0 -1 0 1 0 0 0]);

        dtheta2_du = sign(y1-y2)*(-1/sqrt(1-((x2-x1)/l)^2))*((x2-
x1)*(-1/l^2)*dl_du + (1/l)*[0 0 -1 0 1 0 0 0])...
            + sign(y2-y3)*(1/sqrt(1-((x3-x2)/l2)^2))*((x3-x2)*(-
1/l2^2)*dl2_du + (1/l2)*[0 0 0 0 -1 0 1 0]);

        % Find the unit normal to the element.
        n = (1/l)*[y1-y2;x2-x1];

        % Differentiate the unit normal with respect to the node
positions.
        dn_du = [y1-y2;x2-x1]*(-1/l^2)*dl_du + (1/l)*[0 0 0 1 0 -1 0
0;0 0 -1 0 1 0 0 0];

        % Differentiate the bending force vector.
        dFb_du = (1/l)*[k2*(theta2 - alpha2)*dn_du;k1*(theta1 -
alpha1)*dn_du]...
            + [n*k2*(theta2 - alpha2);n*k1*(theta1 - alpha1)]*(-
1/l^2)*dl_du...
            + (1/l)*[n*k2*dtheta2_du;n*k1*dtheta1_du];

        % Add the differentiated bending forces to the stiffness
matrix.
        J_F_reac(2*(i-1)+1:2*(i-1)+4,2*(i-1)-1:2*(i-1)+6) =
J_F_reac(2*(i-1)+1:2*(i-1)+4,2*(i-1)-1:2*(i-1)+6) + dFb_du;
    end
end
%}
```

```
function [obj,cost_mat,energy_Wh,c_reinf] =
obj_func(node_posns,num_meridians,L,p_zero,original_link_lengths,mat
_props_m,p_reinf_Nm,p_surf,p_ball,depth,rho_water,R,mat_props_c,alph
a,k,rho_ball)
% Finds the value of the objective function.

% Declare the global variables.
global g

% STORED ENERGY

% Find link lengths.
link_lengths = ((node_posns(2:end,1)-node_posns(1:end-1,1)).^2 +
(node_posns(2:end,2)-node_posns(1:end-1,2)).^2).^0.5;

% Find positions of link centrepoints.
centres = 0.5*(node_posns(2:end,:)+node_posns(1:end-1,:));

% Find enclosed volume.
Vbag = sum(2*pi*centres(:,1).*centres(:,2).*(node_posns(2:end,1)-
node_posns(1:end-1,1)));

% Calculate the contained air pressure (absolute).
p_air = p_zero + rho_water*g*depth + 1e5;

% Calculate the stored energy in Joules and Wh, assuming isothermal
% expansion to atmospheric pressure.
energy_J = Vbag*p_air*log(p_air/1e5);
energy_Wh = energy_J/3600;

% MATERIAL COST

% Find surface area and cost of surface material - deformed lengths
are
% used because the surface material should not stretch before the
% reinforcements do.
A_surf = 2*pi*centres(:,1)'*link_lengths;
c_surf = A_surf*p_surf;

% Find mass of ballast required and cost of ballast.
M_air = 0.02897;
R_const = 8.314472;
T_K = 5 + 273.15;
rho_air = p_air*M_air/(R_const*T_K);
V_ball = (rho_water-rho_air)*Vbag/(rho_ball-rho_water);
m_ball = rho_ball*V_ball;
c_ball = m_ball*p_ball;

% Find cost of structural reinforcement.
[F_reac,Tm,Tc,Fb_tot] =
find_F_reac(link_lengths,original_link_lengths,mat_props_m,node_posn
s,R,mat_props_c,alpha,k);
Fd = sum(Tm.*link_lengths)*num_meridians +
sum(2*pi*node_posns(:,1).*Tc);
c_reinf = Fd*p_reinf_Nm;

% Sum the costs to find the total cost of materials.
cost_mat = c_reinf + c_surf + c_ball;
```

```
% OBJECTIVE FUNCTION

% Calculate the objective function value, cost of bag materials per
unit of
% energy stored, in £/MWh.
obj = (cost_mat)/(energy_Wh/1e6);
```

C. Matlab Code for 3D FEA

```
clear

% Declare the global variables.
global depth T M rho_ext p0;
global R g p_int;

M = 0.02897; % Molar mass of stored gas.
R = 8.314472; % Universal gas constant.
rho_ext = 1025; % Density of external fluid.
g = 9.81; % Standard gravity.

prompt = {'Meridional length (m):'
    'Lower bulkhead radius (m):'
    'Upper bulkhead radius (m):'
    'Upper bulkhead mass (kg):'
    'Number of tendons:'
    'Depth of base (m):'
    'Differential pressure at base (Pa):'
    'Stored air temperature (C):'};
dlg_title = 'Input for bag details';
num_lines = 1;
def = {'2.1628','0.09843','0.09843','5.502','36','500','1e5','5'}; %
1.8m prototype defaults
%def = {'6.5556','0','0','0','48','500','2.78e5','5'}; % 5m
prototype defaults
answer = inputdlg(prompt,dlg_title,num_lines,def);

L_m = str2num(answer{1});
r_lo = str2num(answer{2});
r_up = str2num(answer{3});
m_bh = str2num(answer{4});
no_of_lobes = str2num(answer{5});
depth = str2num(answer{6});
p0 = str2num(answer{7});
T = 273.15 + str2num(answer{7});

prompt = {'Stiffness of tendons (GPa):'
    'Tendon diameter (mm):'
    'Tendon density (kg/m^3):'
    'Tendon shortening percentage:'
    'Siffness of membrane (GPa):'
    'Poissons ratio of membrane:'
    'Membrane thickness (mm):'
    'Membrane density (kg/m^3):'
    'Round (0) or straight (1) lobe ends:'};
dlg_title = 'Input for material details';
num_lines = 1;
def =
{'2.5532','7','1400','96.46','0.29367','0.4','0.4089','1440','0'}; %
1.8m prototype defaults
%def =
{'0.38936','9.7','1400','96.46','0.29367','0.4','0.4089','1440','0'}
; % 5m prototype defaults (Tendon strength : 808MPa)
answer = inputdlg(prompt,dlg_title,num_lines,def);

E_ten = 1e9*str2num(answer{1});
```

```
D_ten = 1e-3*str2num(answer{2});
rho_ten = str2num(answer{3});
fac = 1e-2*str2num(answer{4});
E_mem = 1e9*str2num(answer{5});
poi = str2num(answer{6});
t = 1e-3*str2num(answer{7});
rho_mem = str2num(answer{8});
str_ends = str2num(answer{9});


prompt = {'Number of meridional nodes in a lobe'
    'Number of circumferential nodes in a half-lobe'};
dlg_title = 'Input for mesh details:';
num_lines = 1;
def = {'24','8'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
nn_m = str2num(answer{1});
no_of_radii = str2num(answer{2});




% start_p = [0.01,0.5];
% end_p = [1,0];
% int_points = zeros(nn_m,2);
% for i = 1:nn_m
%     int_points(i,:) = start_p + i*(end_p-start_p)/(nn_m+1);
% end
% line_definition = [start_p;int_points;end_p];



h0 = 0;
start_p = [r_up,h0];
end_p = [r_up+(L_m/2)*cos(asin(h0/L_m)),h0/2];
int_points = zeros(nn_m,2);
for i = 1:nn_m
    int_points(i,:) = start_p + i*(end_p-start_p)/(nn_m+1);
end
line_definition = [start_p;int_points;end_p];

% Add the lower half of the meridian.
int_points = zeros(nn_m,2);
for i = 1:nn_m
    int_points(i,:) = end_p + i*([r_lo,0]-end_p)/(nn_m+1);
end
line_definition = [line_definition;int_points;[r_lo,0]];

node_posns = line_definition;


[q0,e_ind,f_ind,num_ax_links,num_circ_links,num_dia_links] = ...

flat_half_lobe_gen(node_posns,no_of_lobes,no_of_radii,str_ends);

% Generate the constraints matrix.
constraints_mat = gen_T(no_of_radii,line_definition,no_of_lobes,q0);

% if r_up == 0
%     [q0,e_ind,f_ind,num_ax_links,num_circ_links,num_dia_links] =
...
```

```
%
flat_half_lobe_gen(node_posns,no_of_lobes,no_of_radii,str_ends);
%
%     centre = [0;0;line_definition(1,2)];
%     line_definition(1,:) = [];
%
%     % Generate the constraints matrix.
%     constraints_mat =
T_constrain_end_and_lobe_edge_nodes(no_of_radii,line_definition,no_o
f_lobes);
% elseif r_up > 0
%     [q0,e_ind,f_ind,num_ax_links,num_circ_links,num_dia_links] =
...
%
flat_half_lobe_gen_upp_bh(node_posns,no_of_lobes,no_of_radii);
%
%     % Generate the constraints matrix.
%     [constraints_mat,Tee_unf] =
T_upp_bh(no_of_radii,line_definition,no_of_lobes);
% else
%     error('Upper radius cannot be negative.')
% end

% Let all faces be membrane elements.
mem_ind = f_ind;

% Set the current node positions (q) to the initial node positions
(q0).
q = q0;


%h0 = (L_m+r_up+r_lo)/2;
h0 = 0.001;
start_p = [r_up,h0];
end_p = [r_up+(L_m/2)*cos(asin(h0/L_m)),h0/2];
int_points = zeros(nn_m,2);
for i = 1:nn_m
    int_points(i,:) = start_p + i*(end_p-start_p)/(nn_m+1);
end
line_definition = [start_p;int_points;end_p];

% Add the lower half of the meridian.
int_points = zeros(nn_m,2);
for i = 1:nn_m
    int_points(i,:) = end_p + i*([r_lo,0]-end_p)/(nn_m+1);
end
line_definition = [line_definition;int_points;[r_lo,0]];

node_posns = line_definition;


[q,e_ind,f_ind,num_ax_links,num_circ_links,num_dia_links] = ...

flat_half_lobe_gen(node_posns,no_of_lobes,no_of_radii,str_ends);

% if r_up == 0 && r_lo == 0
%     [q,e_ind,f_ind,num_ax_links,num_circ_links,num_dia_links] =
...
%         flat_half_lobe_gen(node_posns,no_of_lobes,no_of_radii);
% elseif r_up > 0 && r_lo == 0
%     [q,e_ind,f_ind,num_ax_links,num_circ_links,num_dia_links] =
...
%
flat_half_lobe_gen_upp_bh(node_posns,no_of_lobes,no_of_radii);
% elseif r_up == 0 && r_lo > 0
%     [q,e_ind,f_ind,num_ax_links,num_circ_links,num_dia_links] =
...
%
flat_half_lobe_gen_low_bh(node_posns,no_of_lobes,no_of_radii);
% elseif r_up > 0 && r_lo > 0
%     [q,e_ind,f_ind,num_ax_links,num_circ_links,num_dia_links] =
...
%
flat_half_lobe_gen_2_bh(node_posns,no_of_lobes,no_of_radii);
% else
%     error('A bulkhead radius cannot be negative.')
% end

% Calculate the internal fluid (gauge) pressure.
p_int = (p0 + rho_ext*g*depth)*ones(size(f_ind,1),1);

% Plot a full lobe in the undeformed configuration.
posns = q0;
trisurf(f_ind,posns(:,2),posns(:,3),posns(:,4))
hold on
trisurf(f_ind,posns(:,2),-posns(:,3),posns(:,4))
hold off
axis equal

pause(0.0001)

% Set prestress, Young's Modulus (E), and diameter (d) of axial
links.
stress_pre_ax = 0*ones(num_ax_links,1);

% Upper and lower bulkhead.
if r_up > 0 && r_lo > 0
    E_ax = [zeros((no_of_radii-1)*(size(line_definition,1)-
1),1);E_ten*ones(size(line_definition,1)-1,1)];
    d_ax = [zeros((no_of_radii-1)*(size(line_definition,1)-
1),1);D_ten*ones(size(line_definition,1)-1,1)];

% No bulkhead.
elseif r_up == 0 && r_lo == 0
    E_ax = [zeros((no_of_radii-1)*(size(line_definition,1)-
3),1);E_ten*ones((size(line_definition,1)-3),1);zeros(no_of_radii-
1,1);E_ten;zeros(no_of_radii-1,1);E_ten];
    d_ax = [zeros((no_of_radii-1)*(size(line_definition,1)-
3),1);D_ten*ones((size(line_definition,1)-3),1);zeros(no_of_radii-
1,1);D_ten;zeros(no_of_radii-1,1);D_ten];

% Just one bulkhead.
elseif (r_up == 0 && r_lo > 0) || (r_up > 0 && r_lo == 0)
    E_ax = [zeros((no_of_radii-1)*(size(line_definition,1)-
2),1);E_ten*ones((size(line_definition,1)-2),1);zeros(no_of_radii-
1,1);E_ten];
    d_ax = [zeros((no_of_radii-1)*(size(line_definition,1)-
2),1);D_ten*ones((size(line_definition,1)-2),1);zeros(no_of_radii-
1,1);D_ten];
```

```
end

% Set prestress, E, and d of circumferential links.
stress_pre_circ = 0*ones(num_circ_links,1);
E_circ = 0*ones(num_circ_links,1);
d_circ = 0*ones(num_circ_links,1);
if r_up > 0
    for i = 1:size(line_definition,1):(no_of_radii-
1)*size(line_definition,1)
        E_circ(i,1) = E_ten; % Model a cable running along the top
edge of the lobe.
        d_circ(i,1) = D_ten;
    end
end
if r_lo > 0
    for i =
size(line_definition,1):size(line_definition,1):(no_of_radii-
1)*size(line_definition,1)
        E_circ(i,1) = E_ten; % Model a cable running along the
bottom edge of the lobe.
        d_circ(i,1) = D_ten;
    end
end
%d_circ = D_ten*ones(num_circ_links,1);

% Set prestress, E, and d of diagonal links.
stress_pre_dia = 0*ones(num_dia_links,1);
E_dia = 0*ones(num_dia_links,1);
d_dia = zeros(num_dia_links,1);
%d_dia = D_ten*ones(num_dia_links,1);

% Concatenate prestress, E, and d for all the links.
stress_pre = [stress_pre_ax;stress_pre_circ;stress_pre_dia];
E = [E_ax;E_circ;E_dia];
d = [d_ax;d_circ;d_dia];

% Calculate cross-sectional areas of links.
A = pi.*(d./2).^2;

% Set Young's Modulus of links in compression.
E_comp = 1e-6*E;

% Initialise L as an empty matrix.
L = zeros(size(e_ind,1),1);

% Find the original length (L) of each cable.
for i = 1:size(e_ind,1)
    % Assign node 1 to q1.
    q1 = q0(e_ind(i,1),2:4);

    % Assign node 2 to q2.
    q2 = q0(e_ind(i,2),2:4);

    % Calculate the length of each cable.
    L(i,1) = norm(q1-q2);
end

% Shorten cables as appropriate.
fac = (fac*(L_m+r_up+r_lo)-(r_up+r_lo))/L_m;
```

```
%fac = (0.90*(L_m+r_up+r_lo)-(r_up+r_lo))/L_m;

if r_up > 0 && r_lo > 0
    top = (no_of_radii-1)*(size(line_definition,1)-1)+1;
    bot = no_of_radii*(size(line_definition,1)-1);
    L(top:bot,1) = fac*L(top:bot,1);
elseif r_up == 0 && r_lo == 0
    top = (no_of_radii-1)*(size(line_definition,1)-3)+1;
    bot = no_of_radii*(size(line_definition,1)-3);
    L(top:bot,1) = fac*L(top:bot,1);
    L(bot+no_of_radii,1) = fac*L(bot+no_of_radii,1);
    L(bot+2*no_of_radii,1) = fac*L(bot+2*no_of_radii,1);
elseif (r_up > 0 && r_lo == 0) || (r_up == 0 && r_lo > 0)
    top = (no_of_radii-1)*(size(line_definition,1)-2)+1;
    bot = no_of_radii*(size(line_definition,1)-2);
    L(top:bot,1) = fac*L(top:bot,1);
    L(bot+no_of_radii,1) = fac*L(bot+no_of_radii,1);
end

% Set the thickness of the membrane elements.
t = t*ones(size(mem_ind,1),1);

% Set the Young's Modulus of the membrane elements.
%E_mem = 0.05e9*ones(size(mem_ind,1),1); % Rubber
E_mem = E_mem*ones(size(mem_ind,1),1); % UHPV material

% Set the Poisson's Ratio of the membrane elements.
poi = poi*ones(size(mem_ind,1),1); % Rubber, UHPV material not yet
known

% Set the prestress in the membrane elements.
stress_pre_mem =
[1*ones(size(mem_ind,1),1),0*ones(size(mem_ind,1),2)];

% Calculate the areas of the membrane elements.
S = zeros(size(mem_ind,1),1);
for i = 1:size(mem_ind,1)
    a = q0(mem_ind(i,2),2:4)' - q0(mem_ind(i,1),2:4)';
    b = q0(mem_ind(i,3),2:4)' - q0(mem_ind(i,1),2:4)';
    S(i,1) = 0.5*norm(cross(a,b));
end

% Calculate the mass of the cable and membrane elements.
cab_mass = rho_ten.*A.*L;
mem_mass = rho_mem.*S.*t;

% Form the matrix of cable properties.
mat_props = [L,E,E_comp,A,stress_pre];


%-------------------------------------------------------------------
--
% SOLVE

% Set the convergence criterion.
conv_crit = 1e-6;

% Set the load fraction to 1 (full load) and solve.
load_frac = 1;
```

```
q =
find_eqlbrm_posns(q,f_ind,e_ind,mat_props,load_frac,constraints_mat,
conv_crit,mem_ind,cab_mass,mem_mass,E_mem,poi,stress_pre_mem,t,q0,Te
e_unf,m_bh,line_definition,no_of_lobes,no_of_radii);

% Set the prestress in the membrane elements to zero.
stress_pre_mem = zeros(size(mem_ind,1),3);



%--------------------------------------------------------------------
--
% GENERATE PLOTS

% Surface plot a full lobe (and show Von-Mises stresses if correct
code is used).
posns = q;
[~,~,~,~,~,stress_prin] =
find_F_reac(q,e_ind,mat_props,mem_ind,E_mem,poi,stress_pre_mem,t,q0)
;
stress_vm = zeros(size(stress_prin,1),1);
for i = 1:size(stress_prin,1)
    if (stress_prin(i,2) <= 0) && (stress_prin(i,1) > 0) % Uniaxial
wrinkling.
        stress_vm(i,1) = stress_prin(i,1);
    elseif stress_prin(i,1) <= 0 % Biaxial wrinkling.
        stress_vm(i,1) = 0;
    else % No wrinkling.
        stress_vm(i,1) = (stress_prin(i,1)^2-
stress_prin(i,1)*stress_prin(i,2)+stress_prin(i,2)^2)^0.5;
    end
end
trisurf(f_ind,posns(:,2),posns(:,3),posns(:,4),stress_vm(:,1))%,'Edg
eColor','none')
%trisurf(f_ind,posns(:,2),posns(:,3),posns(:,4),'EdgeColor','none')
hold on
trisurf(f_ind,posns(:,2),-
posns(:,3),posns(:,4),stress_vm(:,1))%,'EdgeColor','none')
%trisurf(f_ind,posns(:,2),-posns(:,3),posns(:,4),'EdgeColor','none')
hold off
axis equal
view(30,30)
colorbar
```

```
function [undeformed_node_posns,links_indices,patches_indices,...
    num_ax_links,num_circ_links,num_dia_links]...
    =
flat_half_lobe_gen(node_posns,no_of_lobes,no_of_radii,str_ends)

meridian = node_posns;
line_definition = node_posns;
x = meridian(:,1);
z = meridian(:,2);

% Calculate alpha, the angle subtended by one set of meridional
nodes.
alpha = (pi/no_of_lobes)/(no_of_radii-1);

% Create the list of undeformed node positions.
undeformed_node_posns = zeros(0,4);

r_up = meridian(1,1);
r_lo = meridian(end,1);

% Set the upper node coordinates if there's no upper bulkhead.
if r_up == 0
    up = [1000,0,0,z(1)];
end

% Set the lower node coordinates if there's no lower bulkhead.
if r_lo == 0
    lo = [9999,0,0,z(end)];
end

% Upper and lower bulkheads.
if r_up > 0 && r_lo > 0
    for i = 0:no_of_radii-1
        if str_ends == 0
            top =
[1000*(i+1)+1,x(1)*cos(i*alpha),x(1)*sin(i*alpha),z(1)];
            bottom =
[1000*(i+1)+size(meridian,1),x(end)*cos(i*alpha),x(end)*sin(i*alpha)
,z(end)];
            undeformed_node_posns =
[undeformed_node_posns;top;1000*(i+1)+(2:size(meridian,1)-
1)',x(2:end-1)*cos(i*alpha),x(2:end-1)*sin(i*alpha),z(2:end-
1);bottom]; % Curved end
        else
            top = [1000*(i+1)+1,x(1),x(1)*tan(i*alpha),z(1)];
            bottom =
[1000*(i+1)+size(meridian,1),x(end),x(end)*tan(i*alpha),z(end)];
            undeformed_node_posns =
[undeformed_node_posns;top;1000*(i+1)+(2:size(meridian,1)-
1)',x(2:end-1),x(2:end-1)*tan(i*alpha),z(2:end-1);bottom]; %
Straight end
        end
    end

% No bulkheads.
elseif r_up == 0 && r_lo == 0
    for i = 0:no_of_radii-1
        undeformed_node_posns =
[undeformed_node_posns;1000*(i+1)+(1:size(meridian,1)-2)',x(2:end-
1)*cos(i*alpha),x(2:end-1)*sin(i*alpha),z(2:end-1)];
```

```
    end

    undeformed_node_posns = [undeformed_node_posns;up;lo];

% Upper bulkhead only.
elseif r_up > 0 && r_lo == 0
    for i = 0:no_of_radii-1
        if str_ends == 0
            top =
[1000*(i+1)+1,x(1)*cos(i*alpha),x(1)*sin(i*alpha),z(1)];
            undeformed_node_posns =
[undeformed_node_posns;top;1000*(i+1)+(2:size(meridian,1)-
1)',x(2:end-1)*cos(i*alpha),x(2:end-1)*sin(i*alpha),z(2:end-1)]; %
Curved end
        else
            top = [1000*(i+1)+1,x(1),x(1)*tan(i*alpha),z(1)];
            undeformed_node_posns =
[undeformed_node_posns;top;1000*(i+1)+(2:size(meridian,1)-
1)',x(2:end-1),x(2:end-1)*tan(i*alpha),z(2:end-1)]; % Straight end
        end
    end

    undeformed_node_posns = [undeformed_node_posns;lo];

% Lower bulkhead only.
elseif r_up == 0 && r_lo > 0
    for i = 0:no_of_radii-1
        if str_ends == 0
            bottom = [1000*(i+1)+size(meridian,1)-
1,x(end)*cos(i*alpha),x(end)*sin(i*alpha),z(end)];
            undeformed_node_posns =
[undeformed_node_posns;1000*(i+1)+(1:size(meridian,1)-2)',x(2:end-
1)*cos(i*alpha),x(2:end-1)*sin(i*alpha),z(2:end-1);bottom]; % Curved
end
        else
            bottom = [1000*(i+1)+size(meridian,1)-
1,x(end),x(end)*tan(i*alpha),z(end)];
            undeformed_node_posns =
[undeformed_node_posns;1000*(i+1)+(1:size(meridian,1)-2)',x(2:end-
1),x(2:end-1)*tan(i*alpha),z(2:end-1);bottom]; % Straight end
        end
    end

    undeformed_node_posns = [undeformed_node_posns;up];

% Error message if a bulkhead radius is negative.
else
    error('Bulkhead radii cannot be negative.')
end

% Create the mesh.
[links_indices,patches_indices,num_ax_links,num_circ_links,num_dia_l
inks] =
create_lobe_mesh(no_of_radii,line_definition,no_of_lobes,r_up,r_lo,u
ndeformed_node_posns);
```

```
function
[links_indices,patches_indices,num_ax_links,num_circ_links,num_dia_l
inks] =
create_lobe_mesh(no_of_radii,line_definition,no_of_lobes,r_up,r_lo,u
ndeformed_node_posns)


if r_up == 0
    line_definition(1,:) = [];
end
if r_lo == 0
    line_definition(end,:) = [];
end


%  LINKS

% Create axial links.
ax_node_1 = zeros(0,1);
ax_node_2 = zeros(0,1);
for i = 1:no_of_radii
    ax_node_1 = [ax_node_1;ones(size(line_definition,1)-1,1)*(i-
1)*size(line_definition,1) + (1:size(line_definition,1)-1)'];
    ax_node_2 = [ax_node_2;ones(size(line_definition,1)-1,1)*(i-
1)*size(line_definition,1) + (2:size(line_definition,1))'];
end

% Axial links connecting to the centre (node_1 here).
if r_up == 0 && r_lo == 0
    ax_node_1 =
[ax_node_1;ones(no_of_radii,1)*(size(undeformed_node_posns,1)-1)];
    ax_node_2 =
[ax_node_2;(1:size(line_definition,1):no_of_radii*size(line_definiti
on,1))'];

    ax_node_1 =
[ax_node_1;ones(no_of_radii,1)*(size(undeformed_node_posns,1))];
    ax_node_2 =
[ax_node_2;(size(line_definition,1):size(line_definition,1):no_of_ra
dii*size(line_definition,1))'];
end
if r_up == 0 && r_lo > 0
    ax_node_1 =
[ax_node_1;ones(no_of_radii,1)*size(undeformed_node_posns,1)];
    ax_node_2 =
[ax_node_2;(1:size(line_definition,1):no_of_radii*size(line_definiti
on,1))'];
end
if r_up > 0 && r_lo == 0
    ax_node_1 =
[ax_node_1;ones(no_of_radii,1)*(size(undeformed_node_posns,1))];
    ax_node_2 =
[ax_node_2;(size(line_definition,1):size(line_definition,1):no_of_ra
dii*size(line_definition,1))'];
end

ax_nodes = [ax_node_1,ax_node_2];
num_ax_links = size(ax_nodes,1);
```

```
% Create circumferential links.
circ_node_1 = zeros(0,1);
circ_node_2 = zeros(0,1);
for i = 1:no_of_radii-1
    circ_node_1 = [circ_node_1;(1+(i-
1)*size(line_definition,1):i*size(line_definition,1))'];
    circ_node_2 =
[circ_node_2;(1+i*size(line_definition,1):(i+1)*size(line_definition
,1))'];
end

circ_nodes = [circ_node_1,circ_node_2];
num_circ_links = size(circ_nodes,1);


% Create diagonal links (i.e. triangulate the quads).
dia_node_1 = zeros(0,1);
dia_node_2 = zeros(0,1);
for i = 1:2:no_of_radii-1
    for j = 1:2:size(line_definition,1)-1
        dia_node_1 = [dia_node_1;j+(i-1)*size(line_definition,1)];
        dia_node_2 = [dia_node_2;(j+1)+i*size(line_definition,1)];
    end

    for j = 2:2:size(line_definition,1)-1
        dia_node_1 = [dia_node_1;j+i*size(line_definition,1)];
        dia_node_2 = [dia_node_2;(j+1)+(i-
1)*size(line_definition,1)];
    end
end

for i = 2:2:no_of_radii-1
    for j = 2:2:size(line_definition,1)-1
        dia_node_1 = [dia_node_1;j+(i-1)*size(line_definition,1)];
        dia_node_2 = [dia_node_2;(j+1)+i*size(line_definition,1)];
    end

    for j = 1:2:size(line_definition,1)-1
        dia_node_1 = [dia_node_1;j+i*size(line_definition,1)];
        dia_node_2 = [dia_node_2;(j+1)+(i-
1)*size(line_definition,1)];
    end
end

dia_nodes = [dia_node_1,dia_node_2];
num_dia_links = size(dia_nodes,1);

links_indices = [ax_nodes;circ_nodes;dia_nodes];



% PATCHES

patches_node_1 = zeros(0,1);
patches_node_2 = zeros(0,1);
patches_node_3 = zeros(0,1);
```

```
for i = 1:2:no_of_radii-1
    for j = 1:2:size(line_definition,1)-1
        patches_node_1 = [patches_node_1;j+(i-
1)*size(line_definition,1)];
        patches_node_2 = [patches_node_2;(j+1)+(i-
1)*size(line_definition,1)];
        patches_node_3 =
[patches_node_3;(j+1)+i*size(line_definition,1)];

        patches_node_1 = [patches_node_1;j+(i-
1)*size(line_definition,1)];
        patches_node_2 =
[patches_node_2;(j+1)+i*size(line_definition,1)];
        patches_node_3 =
[patches_node_3;j+i*size(line_definition,1)];
    end

    for j = 2:2:size(line_definition,1)-1
        patches_node_1 =
[patches_node_1;j+i*size(line_definition,1)];
        patches_node_2 = [patches_node_2;j+(i-
1)*size(line_definition,1)];
        patches_node_3 = [patches_node_3;(j+1)+(i-
1)*size(line_definition,1)];

        patches_node_1 =
[patches_node_1;j+i*size(line_definition,1)];
        patches_node_2 = [patches_node_2;(j+1)+(i-
1)*size(line_definition,1)];
        patches_node_3 =
[patches_node_3;(j+1)+i*size(line_definition,1)];
    end
end

for i = 2:2:no_of_radii-1
    for j = 2:2:size(line_definition,1)-1
        patches_node_1 = [patches_node_1;j+(i-
1)*size(line_definition,1)];
        patches_node_2 = [patches_node_2;(j+1)+(i-
1)*size(line_definition,1)];
        patches_node_3 =
[patches_node_3;(j+1)+i*size(line_definition,1)];

        patches_node_1 = [patches_node_1;j+(i-
1)*size(line_definition,1)];
        patches_node_2 =
[patches_node_2;(j+1)+i*size(line_definition,1)];
        patches_node_3 =
[patches_node_3;j+i*size(line_definition,1)];

    end

    for j = 1:2:size(line_definition,1)-1
        patches_node_1 =
[patches_node_1;j+i*size(line_definition,1)];
        patches_node_2 = [patches_node_2;j+(i-
1)*size(line_definition,1)];
        patches_node_3 = [patches_node_3;(j+1)+(i-
1)*size(line_definition,1)];
```

```
        patches_node_1 =
[patches_node_1;j+i*size(line_definition,1)];
        patches_node_2 = [patches_node_2;(j+1)+(i-
1)*size(line_definition,1)];
        patches_node_3 =
[patches_node_3;(j+1)+i*size(line_definition,1)];
    end
end

% Central patches at the top and bottom of the bag.
if r_up == 0 && r_lo == 0
    % Top.
    patches_node_1 = [patches_node_1;ones(no_of_radii-
1,1)*(size(undeformed_node_posns,1)-1)];
    patches_node_2 =
[patches_node_2;(1:size(line_definition,1):(no_of_radii-
1)*size(line_definition,1))'];
    patches_node_3 =
[patches_node_3;(1+size(line_definition,1):size(line_definition,1):n
o_of_radii*size(line_definition,1))'];

    % Bottom.
    patches_node_1 = [patches_node_1;ones(no_of_radii-
1,1)*size(undeformed_node_posns,1)];
    patches_node_2 =
[patches_node_2;(2*size(line_definition,1):size(line_definition,1):n
o_of_radii*size(line_definition,1))'];
    patches_node_3 =
[patches_node_3;(size(line_definition,1):size(line_definition,1):(no
_of_radii-1)*size(line_definition,1))'];
end

% Central patches at the top of the bag.
if r_up == 0 && r_lo > 0
    patches_node_1 = [patches_node_1;ones(no_of_radii-
1,1)*(size(undeformed_node_posns,1))];
    patches_node_2 =
[patches_node_2;(1:size(line_definition,1):(no_of_radii-
1)*size(line_definition,1))'];
    patches_node_3 =
[patches_node_3;(1+size(line_definition,1):size(line_definition,1):n
o_of_radii*size(line_definition,1))'];
end

% Central patches at the bottom of the bag.
if r_up > 0 && r_lo == 0
    patches_node_1 = [patches_node_1;ones(no_of_radii-
1,1)*size(undeformed_node_posns,1)];
    patches_node_2 =
[patches_node_2;(2*size(line_definition,1):size(line_definition,1):n
o_of_radii*size(line_definition,1))'];
    patches_node_3 =
[patches_node_3;(size(line_definition,1):size(line_definition,1):(no
_of_radii-1)*size(line_definition,1))'];
end

patches_indices = [patches_node_1,patches_node_2,patches_node_3];
```

```
function [Tee] =
gen_T(no_of_radii,line_definition,no_of_lobes,undeformed_node_posns)

% Initialise C as a matrix of zeros.
C = zeros(3*size(undeformed_node_posns,1),3);

% Calculate the angle subtended by half of one lobe.
theta = pi/no_of_lobes;

r_up = line_definition(1,1);
r_lo = line_definition(end,1);

% 2 bulkheads.
if r_up > 0 && r_lo > 0
    % Allow movement of nodes on x-axis in x and z.
    C(4:3:3*size(line_definition,1)-5,1) = 1; % Allow movement in x.
    C(3:3:3*size(line_definition,1)-3,3) = 1; % Allow movement in z.

    % Only allow movement of nodes on upper bulkhead in z.
    C(3:3*size(line_definition,1):3*(no_of_radii-
1)*size(line_definition,1)+3,3) = 1;

    % Allow movement of all interior nodes in x, y, and z.
    for i = 2:no_of_radii-1
        C(3*(i-
1)*size(line_definition,1)+4:3:3*i*size(line_definition,1)-5,1) = 1;
        C(3*(i-
1)*size(line_definition,1)+5:3:3*i*size(line_definition,1)-4,2) = 1;
        C(3*(i-
1)*size(line_definition,1)+6:3:3*i*size(line_definition,1)-3,3) = 1;
    end

    % Allow in-plane movement of nodes on tendon apart from top and
    % bottom nodes.
    C(3*(no_of_radii-
1)*size(line_definition,1)+4:3:3*no_of_radii*size(line_definition,1)
-5,1) = cos(theta);
    C(3*(no_of_radii-
1)*size(line_definition,1)+4:3:3*no_of_radii*size(line_definition,1)
-5,2) = sin(theta);
    C(3*(no_of_radii-
1)*size(line_definition,1)+6:3:3*no_of_radii*size(line_definition,1)
-3,3) = 1;

% No bulkheads.
elseif r_up == 0 && r_lo == 0
    % Allow movement of top node in z.
    C(end-3,3) = 1;

    line_definition(1,:) = [];
    line_definition(end,:) = [];

    % Allow movement of nodes on x-axis in x and z.
    C(1:3:3*size(line_definition,1)-2,1) = 1; % Allow movement in x.
    C(3:3:3*size(line_definition,1),3) = 1; % Allow movement in z.

    % Allow movement of all interior nodes in x, y, and z.
    for i = 2:no_of_radii-1
```

```
        C(3*(i-
1)*size(line_definition,1)+1:3:3*i*size(line_definition,1)-2,1) = 1;
        C(3*(i-
1)*size(line_definition,1)+2:3:3*i*size(line_definition,1)-1,2) = 1;
        C(3*(i-
1)*size(line_definition,1)+3:3:3*i*size(line_definition,1),3) = 1;
    end

    % Allow in-plane movement of nodes on tendon apart from top and
    % bottom nodes.
    C(3*(no_of_radii-
1)*size(line_definition,1)+1:3:3*no_of_radii*size(line_definition,1)
-2,1) = cos(theta);
    C(3*(no_of_radii-
1)*size(line_definition,1)+1:3:3*no_of_radii*size(line_definition,1)
-2,2) = sin(theta);
    C(3*(no_of_radii-
1)*size(line_definition,1)+3:3:3*no_of_radii*size(line_definition,1)
,3) = 1;

% Only a lower bulkhead.
elseif r_up == 0 && r_lo > 0
    % Allow movement of top node in z.
    C(end,3) = 1;

    % Remove the top node from line_definition.
    line_definition(1,:) = [];

    % Allow movement of nodes on x-axis in x and z.
    C(1:3:3*size(line_definition,1)-5,1) = 1; % Allow movement in x.
    C(3:3:3*size(line_definition,1)-3,3) = 1; % Allow movement in z.

    % Allow movement of all interior nodes in x, y, and z.
    for i = 2:no_of_radii-1
        C(3*(i-
1)*size(line_definition,1)+1:3:3*i*size(line_definition,1)-5,1) = 1;
        C(3*(i-
1)*size(line_definition,1)+2:3:3*i*size(line_definition,1)-4,2) = 1;
        C(3*(i-
1)*size(line_definition,1)+3:3:3*i*size(line_definition,1)-3,3) = 1;
    end

    % Allow in-plane movement of nodes on tendon apart from bottom
node.
    C(3*(no_of_radii-
1)*size(line_definition,1)+1:3:3*no_of_radii*size(line_definition,1)
-5,1) = cos(theta);
    C(3*(no_of_radii-
1)*size(line_definition,1)+1:3:3*no_of_radii*size(line_definition,1)
-5,2) = sin(theta);
    C(3*(no_of_radii-
1)*size(line_definition,1)+3:3:3*no_of_radii*size(line_definition,1)
-3,3) = 1;

% Only an upper bulkhead.
elseif r_up > 0 && r_lo == 0
    % Remove the bottom node from line_definition.
    line_definition(end,:) = [];

    % Allow movement of nodes on x-axis in x and z.
    C(4:3:3*size(line_definition,1)-2,1) = 1; % Allow movement in x.
    C(3:3:3*size(line_definition,1),3) = 1; % Allow movement in z.

    % Only allow movement of nodes on upper bulkhead in z.
    C(3:3*size(line_definition,1):3*(no_of_radii-
1)*size(line_definition,1)+3,3) = 1;

    % Allow movement of all interior nodes in x, y, and z.
    for i = 2:no_of_radii-1
        C(3*(i-
1)*size(line_definition,1)+4:3:3*i*size(line_definition,1)-2,1) = 1;
        C(3*(i-
1)*size(line_definition,1)+5:3:3*i*size(line_definition,1)-1,2) = 1;
        C(3*(i-
1)*size(line_definition,1)+6:3:3*i*size(line_definition,1),3) = 1;
    end

    % Allow in-plane movement of nodes on tendon apart from top
node.
    C(3*(no_of_radii-
1)*size(line_definition,1)+4:3:3*no_of_radii*size(line_definition,1)
-2,1) = cos(theta);
    C(3*(no_of_radii-
1)*size(line_definition,1)+4:3:3*no_of_radii*size(line_definition,1)
-2,2) = sin(theta);
    C(3*(no_of_radii-
1)*size(line_definition,1)+6:3:3*no_of_radii*size(line_definition,1)
,3) = 1;
end



% Form the constraints matrix Tee.
Tee = sparse(size(C,1),size(C,1));

for i = 1:size(undeformed_node_posns,1)
    Tee(3*(i-1)+1:3*(i-1)+3,3*(i-1)+1:3*(i-1)+3) = C(3*(i-1)+1:3*(i-
1)+3,:);
end

% Remove rows of zeros.
for i = size(Tee,1):-1:1
    if max(abs(Tee(i,:))) == 0
        Tee(i,:) = [];
    end
end

Tee = Tee';
```

```
function [q] =
find_eqlbrm_posns(q,f_ind,e_ind,mat_props,load_frac,constraints_mat,
conv_crit,mem_ind,cab_mass,mem_mass,E_mem,poi,stress_pre_mem,t,q0,Te
e_unf,m_bh,line_definition,no_of_lobes,no_of_radii)
% Find the equilibrium positions.

%--------------------------------------------------------------------
--
% FIND FORCE VECTORS AND CALCULATE RESIDUALS AND DIMENSIONLESS
RESIDUALS.

% Find the applied forces.
[F_app,p,rho_int] =
find_F_app(q,f_ind,e_ind,mem_ind,cab_mass,mem_mass,m_bh,no_of_lobes,
no_of_radii,line_definition);

% Find the reaction forces.
[F_reac,sum_T,l,stress_cab,strain_cab] =
find_F_reac_cont(q,e_ind,mat_props,mem_ind,E_mem,poi,stress_pre_mem,
t,q0);

% Calculate the force residuals.
r = load_frac*F_app - F_reac;

% Make the residuals dimensionless.
r_nd = r./(1+[sum_T,sum_T,sum_T]);

% Vectorise and constrain r.
r_vec_constr = constraints_mat'*reshape(r',[],1);

% Vectorise and constrain r_nd.
r_nd_vec_constr = constraints_mat'*reshape(r_nd',[],1);


%--------------------------------------------------------------------
--
% RUN SOLVER IF NECESSARY - FIND JACOBIAN OF RESIDUALS AND UPDATE
THE NODE
% POSITIONS.

% Initially set disps to be a very large number.
sum_L = sum(mat_props(1:size(line_definition,1)-1,1));
disps = 1e6*sum_L;
loop_count = 0;

% Until either both the maximum residual and max dimensionless
residual are
% below their criteria, or the max displacement is below a
criterion, carry
% on with solution routine.
while ((max(abs(r_nd_vec_constr)) > conv_crit) ||
(max(abs(r_vec_constr)) > 1)) && (max(max(abs(disps))) > 1e-4*sum_L)

    loop_count = loop_count + 1;

    if loop_count == 2
            stress_pre_mem = zeros(size(mem_ind,1),3);
    end
    if loop_count == 1
        % Carry out one iteration of the solver.
        [q,disps] =
NR_iter_cont(q,f_ind,e_ind,mat_props,p,l,stress_cab,r_vec_constr,loa
d_frac,constraints_mat,rho_int,strain_cab,mem_ind,E_mem,poi,stress_p
re_mem,t,q0);
    else
        % Carry out one iteration of the solver.
        [q,disps] =
NR_iter(q,f_ind,e_ind,mat_props,p,l,stress_cab,r_vec_constr,load_fra
c,constraints_mat,rho_int,strain_cab,mem_ind,E_mem,poi,stress_pre_me
m,t,q0);
    end

    %----------------------------------------------------------------
------
    % FIND FORCE VECTORS AND CALCULATE RESIDUALS AND DIMENSIONLESS
RESIDUALS.

    % Find the applied forces.
    [F_app,p,rho_int] =
find_F_app(q,f_ind,e_ind,mem_ind,cab_mass,mem_mass,m_bh,no_of_lobes,
no_of_radii,line_definition);

    % Find the reaction forces.
    [F_reac,sum_T,l,stress_cab,strain_cab] =
find_F_reac(q,e_ind,mat_props,mem_ind,E_mem,poi,stress_pre_mem,t,q0)
;
%     [F_reac,sum_T,l,stress_cab,strain_cab] =
find_F_reac_cont(q,e_ind,mat_props,mem_ind,E_mem,poi,stress_pre_mem,
t,q0);

    % Calculate the force residuals.
    r = load_frac*F_app - F_reac;

    % Make the residuals dimensionless.
    r_nd = r./(1+[sum_T,sum_T,sum_T]);

    % Vectorise and constrain r.
    r_vec_constr = constraints_mat'*reshape(r',[],1);

    % Vectorise and constrain r_nd.
    r_nd_vec_constr = constraints_mat'*reshape(r_nd',[],1);

    disp(max(abs(r_nd_vec_constr)))

    % Let posns = q, to plot the current node positions.
    posns = q;

    % Surface plot a full lobe.
    trisurf(f_ind,posns(:,2),posns(:,3),posns(:,4))%,r_avg)
    hold on
    trisurf(f_ind,posns(:,2),-posns(:,3),posns(:,4))%,r_avg)
    hold off
    axis equal
    v = axis;
    v(1) = 0;
    axis(v)
    %colorbar

    pause(0.0001)
end
```

```
function [F_app,p,rho_int] =
find_F_app(q,f_ind,e_ind,mem_ind,cab_mass,mem_mass,m_bh,no_of_lobes,
no_of_radii,line_definition)
% Finds the applied forces.

% Declare the global variables.
global T M rho_ext R g p0 p_int;

% Find the centroid of each face.
centroids = zeros(0,3);
for i = 1:size(f_ind,1)
    centroids(i,1) =
mean([q(f_ind(i,1),2),q(f_ind(i,2),2),q(f_ind(i,3),2)]);
    centroids(i,2) =
mean([q(f_ind(i,1),3),q(f_ind(i,2),3),q(f_ind(i,3),3)]);
    centroids(i,3) =
mean([q(f_ind(i,1),4),q(f_ind(i,2),4),q(f_ind(i,3),4)]);
end

% Calculate the density of the fluid inside the bag/balloon.
rho_int = p_int(1,1)*M/(R*T);

% Calculate the differential pressure at each centroid.
p = p0 + (rho_ext - rho_int)*g.*centroids(:,3);

% Initialise Sn (the product of area and unit normal for each
element) as
% an empty matrix.
Sn = zeros(size(f_ind,1),3);

% Find the area and unit normal of each element.
for i = 1:size(f_ind,1)
    % Find vector along side B (n2 - n1).
    Bx = q(f_ind(i,2),2) - q(f_ind(i,1),2);
    By = q(f_ind(i,2),3) - q(f_ind(i,1),3);
    Bz = q(f_ind(i,2),4) - q(f_ind(i,1),4);
    B = [Bx,By,Bz];

    % Find vector along side A (n3 - n1).
    Ax = q(f_ind(i,3),2) - q(f_ind(i,1),2);
    Ay = q(f_ind(i,3),3) - q(f_ind(i,1),3);
    Az = q(f_ind(i,3),4) - q(f_ind(i,1),4);
    A = [Ax,Ay,Az];

    % Calculate the product of the area and unit normal of the
element.
    Sn(i,1:3) = cross(B,A)/2;
end

% Calculate the force on each element.
F_el = [p,p,p].*Sn;

% Calculate the force at each node on each element.
F_n = F_el/3;

% Sum the applied forces at each node.
F_app = zeros(size(q,1),3);
for i = 1:size(F_n,1)
    F_app(f_ind(i,1),1) = F_app(f_ind(i,1),1) + F_n(i,1);
    F_app(f_ind(i,2),1) = F_app(f_ind(i,2),1) + F_n(i,1);
    F_app(f_ind(i,3),1) = F_app(f_ind(i,3),1) + F_n(i,1);

    F_app(f_ind(i,1),2) = F_app(f_ind(i,1),2) + F_n(i,2);
    F_app(f_ind(i,2),2) = F_app(f_ind(i,2),2) + F_n(i,2);
    F_app(f_ind(i,3),2) = F_app(f_ind(i,3),2) + F_n(i,2);

    F_app(f_ind(i,1),3) = F_app(f_ind(i,1),3) + F_n(i,3);
    F_app(f_ind(i,2),3) = F_app(f_ind(i,2),3) + F_n(i,3);
    F_app(f_ind(i,3),3) = F_app(f_ind(i,3),3) + F_n(i,3);
end

% Include weight of cable elements.
for i = 1:size(e_ind,1)
    F_app(e_ind(i,1),3) = F_app(e_ind(i,1),3) - 0.5*g*cab_mass(i,1);
    F_app(e_ind(i,2),3) = F_app(e_ind(i,2),3) - 0.5*g*cab_mass(i,1);
end

% Include weight of membrane elements.
for i = 1:size(mem_ind,1)
    F_app(mem_ind(i,1),3) = F_app(mem_ind(i,1),3) -
(1/3)*g*mem_mass(i,1);
    F_app(mem_ind(i,2),3) = F_app(mem_ind(i,2),3) -
(1/3)*g*mem_mass(i,1);
    F_app(mem_ind(i,3),3) = F_app(mem_ind(i,3),3) -
(1/3)*g*mem_mass(i,1);
end

% Include weight of the bulkhead.
for i = 1:size(line_definition,1):1+(no_of_radii-
1)*size(line_definition,1)
    if i == 1
        F_app(i,3) = F_app(i,3) -
0.5*g*m_bh/(no_of_lobes*2*(no_of_radii - 1));
    elseif i == 1+(no_of_radii-1)*size(line_definition,1)
        F_app(i,3) = F_app(i,3) -
0.5*g*m_bh/(no_of_lobes*2*(no_of_radii - 1));
    else
        F_app(i,3) = F_app(i,3) - g*m_bh/(no_of_lobes*2*(no_of_radii
- 1));
    end
end
```

```
function [F_reac,sum_T,l,stress_cab,strain_cab,stress_prin_reg] =
find_F_reac(q,e_ind,mat_props,mem_ind,E_mem,poi,stress_pre_mem,t,q0)
% Finds the reaction forces.

% Split the matrix of material properties into its constituent
parts.
L = mat_props(:,1);
E = mat_props(:,2);
E_comp = mat_props(:,3);
A = mat_props(:,4);
stress_pre = mat_props(:,5);

% Initialise l and unit_par_to_1 as empty matrices.
l = zeros(size(e_ind,1),1);
unit_par_to_1 = zeros(size(e_ind,1),3);

% Find the length of each cable and the unit parallel of each cable.
for i = 1:size(e_ind,1)
    % Assign node 1 to q1.
    q1 = q(e_ind(i,1),2:4);

    % Assign node 2 to q2.
    q2 = q(e_ind(i,2),2:4);

    q2_to_q1 = q1-q2;

    % Calculate the length of each cable.
    l(i,1) = norm(q2_to_q1);

    % Calculate the unit parallel to each cable in the direction of
node 1.
    unit_par_to_1(i,:) = q2_to_q1./[l(i,1),l(i,1),l(i,1)];
end

% Calculate the strain in each cable.
strain_cab = (l-L)./L;

% Calculate the stress in each cable.
stress_cab = E.*strain_cab + stress_pre;

% Calculate the stress in any compressed cables / set stress equal
to zero
% in compressed cables.
for i = 1:size(stress_cab,1)
    if stress_cab(i,1) < 0
        stress_cab(i,1) = 0;
    end
end

% Calculate the tension in each cable.
T = stress_cab.*A;

% Calculate the force at each node on each cable.
nodal_force_1 = unit_par_to_1.*[T,T,T];
nodal_force_2 = -nodal_force_1;

% Initialise F_reac and sum_T as empty matrices.
F_reac = zeros(size(q,1),3);
sum_T = zeros(size(q,1),1);
```

```
% Sum the nodal forces and tension at each node.
for i = 1:size(e_ind,1)
    F_reac(e_ind(i,1),:) = F_reac(e_ind(i,1),:) +
nodal_force_1(i,:);
    F_reac(e_ind(i,2),:) = F_reac(e_ind(i,2),:) +
nodal_force_2(i,:);

    sum_T(e_ind(i,1),1) = sum_T(e_ind(i,1),1) + abs(T(i,1));
    sum_T(e_ind(i,2),1) = sum_T(e_ind(i,2),1) + abs(T(i,1));
end

stress_reg = zeros(size(mem_ind,1),3);
stress_prin_reg = zeros(size(mem_ind,1),3);

for i = 1:size(mem_ind,1)
    % Form the rotation matrix R.
    a = q0(mem_ind(i,2),2:4)' - q0(mem_ind(i,1),2:4)';
    b = q0(mem_ind(i,3),2:4)' - q0(mem_ind(i,1),2:4)';
    axb = cross(a,b);

    R = [a/norm(a),cross(axb/norm(axb),a/norm(a)),axb/norm(axb)];

    % Set u using the nodal displacements.
    disps = q(:,2:4) - q0(:,2:4);
    u =
[R,zeros(3,3),zeros(3,3);zeros(3,3),R,zeros(3,3);zeros(3,3),zeros(3,
3),R]'...

*[disps(mem_ind(i,1),:)';disps(mem_ind(i,2),:)';disps(mem_ind(i,3),:
)'];

    % Find the local node coordinates with origin at node 1.
    q_loc1 = R'*(q0(mem_ind(i,1),2:4)' - q0(mem_ind(i,1),2:4)');
    q_loc2 = R'*(q0(mem_ind(i,2),2:4)' - q0(mem_ind(i,1),2:4)');
    q_loc3 = R'*(q0(mem_ind(i,3),2:4)' - q0(mem_ind(i,1),2:4)');

    % Move the local origin to the point on side a where a line
    % perpendicular to a will pass through node 3.
    q_loc1 = q_loc1 - [q_loc3(1,1);0;0];
    q_loc2 = q_loc2 - [q_loc3(1,1);0;0];
    q_loc3 = q_loc3 - [q_loc3(1,1);0;0];

    % Assign names to 3 components of local node coordinates.
    x1 = q_loc1(1,1); y1 = q_loc1(2,1); %z1 = q_loc1(3,1);
    x2 = q_loc2(1,1); y2 = q_loc2(2,1); %z2 = q_loc2(3,1);
    x3 = q_loc3(1,1); y3 = q_loc3(2,1); %z3 = q_loc3(3,1);

    % Find the area of the element.
    S = 0.5*norm(axb);

    % Double the area of the element.
    twoS = 2*S;

    % Find the coefficients of the shape functions.
    %a1 = (x2*y3-x3*y2)/twoS;
    b1 = (y2-y3)/twoS;
    c1 = -(x2-x3)/twoS;
    %a2 = (x3*y1-x1*y3)/twoS;
    b2 = (y3-y1)/twoS;
    c2 = -(x3-x1)/twoS;

    %a3 = (x1*y2-x2*y1)/twoS;
    b3 = (y1-y2)/twoS;
    c3 = -(x1-x2)/twoS;

    % Calculate the matrix of linear derivative coefficients.
    B0 = [b1 0  0  b2 0  0  b3 0  0;
          0  c1 0  0  c2 0  0  c3 0;
          c1 b1 0  c2 b2 0  c3 b3 0];

    % Find G, theta, and A, to form matrix of nonlinear derivative
    % coefficients.
    P = [b1 0  0  b2 0  0  b3 0  0;
         0  b1 0  0  b2 0  0  b3 0;
         0  0  b1 0  0  b2 0  0  b3];

    Q = [c1 0  0  c2 0  0  c3 0  0;
         0  c1 0  0  c2 0  0  c3 0;
         0  0  c1 0  0  c2 0  0  c3];

    Pu = P*u;
    Qu = Q*u;

    A = [Pu'        zeros(1,3);
         zeros(1,3) Qu';
         Qu'        Pu'];

    G = [P;Q];

    theta = [Pu;Qu];

    % Calculate the strain in the element.
    strain = B0*u + 0.5*A*theta;

    % Create the elastic matrix for the element in plane stress.
    D = (E_mem(i,1)/(1-poi(i,1)^2))*[1   poi(i,1) 0;
                                     poi(i,1) 1   0;
                                     0   0   (1-poi(i,1))/2];

    % Calculate the stress in the element.
    stress = D*strain + R'*stress_pre_mem(i,:)';
    % STRESS_PRE_MEM SHOULD BE ROTATED TO LOCAL!!!
pre_loc = R'*stress_pre_mem(i,:)';
    % Calculate the principal stresses and strains.
    prin_max = 0.5*(stress(1,1) + stress(2,1)) + ((0.5*(stress(1,1)
- stress(2,1)))^2 + stress(3,1)^2)^(0.5);
    prin_min = 0.5*(stress(1,1) + stress(2,1)) - ((0.5*(stress(1,1)
- stress(2,1)))^2 + stress(3,1)^2)^(0.5);


    prin_max_pre = 0.5*(stress_pre_mem(i,1) + stress_pre_mem(i,2)) +
((0.5*(stress_pre_mem(i,1) - stress_pre_mem(i,2)))^2 +
stress_pre_mem(i,3)^2)^(0.5);
    prin_min_pre = 0.5*(stress_pre_mem(i,1) + stress_pre_mem(i,2)) -
((0.5*(stress_pre_mem(i,1) - stress_pre_mem(i,2)))^2 +
stress_pre_mem(i,3)^2)^(0.5);
```

```
    prin_strain_max = 0.5*(strain(1,1) + strain(2,1)) +
((0.5*(strain(1,1) - strain(2,1)))^2 + strain(3,1)^2)^(0.5);
    prin_strain_min = 0.5*(strain(1,1) + strain(2,1)) -
((0.5*(strain(1,1) - strain(2,1)))^2 + strain(3,1)^2)^(0.5);

    % Calculate the angle between the principal coordinates and
local
    % coordinates.
    prin_theta = 0.5*atan(2*stress(3,1)/(stress(1,1)-stress(2,1)));
    tan2theta = 2*stress(3,1)/(stress(1,1)-stress(2,1));

    % Create Reuter's matrix and the pseudo-vector matrix form of
the
    % rotation matrix.
    Reut = [1,0,0;0,1,0;0,0,2];
    c = cos(prin_theta);
    s = sin(prin_theta);
    T = [c^2,s^2,2*s*c;s^2,c^2,-2*s*c;-s*c,s*c,c^2-s^2];

    % No wrinkling.
    if prin_min > 0
        % Calculate the reaction force at each node on the element.
        F_mem_loc = t(i,1)*S*(B0+A*G)'*stress;

        stress_reg(i,:) = stress';
        stress_prin_reg(i,:) = [prin_max,prin_min,0];

    % Wrinkling.
    else
        b = 1e-6*E_mem(i,1); % THESE MUST BE THE SAME IN THIS AND K!

        if prin_strain_max > 0
            % Changing Poisson's ratio method.
            P = (strain(1)-strain(2))/(prin_strain_max-
prin_strain_min);
            Q = strain(3)/(prin_strain_max-prin_strain_min);

            D = (E_mem(i,1)/4)*[2*(1+P) 0 Q;
                                      0 2*(1-P) Q;
                                      Q Q 1];
        else
            D = [0 0 0;
                 0 0 0;
                 0 0 0];
        end

        % Calculate the stress in the element.
        stress = D*strain + R'*stress_pre_mem(i,:)';

        % Recalculate principal stresses and store in stress_reg.
        prin_max = 0.5*(stress(1,1) + stress(2,1)) +
((0.5*(stress(1,1) - stress(2,1)))^2 + stress(3,1)^2)^0.5;
        prin_min = 0.5*(stress(1,1) + stress(2,1)) -
((0.5*(stress(1,1) - stress(2,1)))^2 + stress(3,1)^2)^0.5;
        stress_prin_reg(i,:) = [prin_max,prin_min,0];

        % Calculate the reaction force at each node on the element.
        F_mem_loc = t(i,1)*S*(B0+A*G)'*stress;
    end
    % Rotate the reaction forces into the global coordinate system.
    F_mem = 
[R,zeros(3,3),zeros(3,3);zeros(3,3),R,zeros(3,3);zeros(3,3),zeros(3,
3),R]*F_mem_loc;

    % Add membrane reaction forces to the cable reaction forces.
    F_reac(mem_ind(i,1),:) = F_reac(mem_ind(i,1),:) + F_mem(1:3,1)';
    F_reac(mem_ind(i,2),:) = F_reac(mem_ind(i,2),:) + F_mem(4:6,1)';
    F_reac(mem_ind(i,3),:) = F_reac(mem_ind(i,3),:) + F_mem(7:9,1)';

    % Add magnitudes of membrane reaction forces to sum_T.
    sum_T(mem_ind(i,1),1) = sum_T(mem_ind(i,1),1) + 
norm(F_mem(1:3,1));
    sum_T(mem_ind(i,2),1) = sum_T(mem_ind(i,2),1) + 
norm(F_mem(4:6,1));
    sum_T(mem_ind(i,3),1) = sum_T(mem_ind(i,3),1) + 
norm(F_mem(7:9,1));
end
```

```
function [q,disps_mod] =
NR_iter(q,f_ind,e_ind,mat_props,p,l,stress_cab,r_vec_constr,load_fra
c,constraints_mat,rho_int,strain_cab,mem_ind,E_mem,poi,stress_pre_me
m,t,q0)
% Carries out one iteration of the Newton-Raphson solver.

% Find the Jacobian of the applied forces.
J_F_app = find_J_F_app(q,f_ind,p,rho_int);

% Find the Jacobian of the reaction forces.
J_F_reac =
find_J_F_reac(q,e_ind,mat_props,l,stress_cab,strain_cab,mem_ind,E_me
m,poi,stress_pre_mem,t,q0);

% Calculate the Jacobian of the force residuals.
J_r = load_frac*J_F_app - J_F_reac;

% Constrain the Jacobian of the force residuals.
J_r_constr = constraints_mat'*J_r*constraints_mat;

% Calculate a set of nodal displacements using the Newton-Raphson
method.
disps = reshape(constraints_mat*(J_r_constr\(-r_vec_constr)),3,[])';

% Multiply the displacement vector by a certain fraction in order to
limit
% the maximum displacements.
if max(sqrt(disps(:,1).^2 + disps(:,2).^2 + disps(:,3).^2)) > 1e-
2*max(sqrt(q(:,2).^2 + q(:,3).^2 + q(:,4).^2))
    disps_mod = disps*1e-2*max(sqrt(q(:,2).^2 + q(:,3).^2 +
q(:,4).^2))/max(sqrt(disps(:,1).^2 + disps(:,2).^2 +
disps(:,3).^2));
else
    disps_mod = disps;
end

% Update the node positions.
q(:,2:4) = q(:,2:4) + disps_mod;
```

```
function J_F_app = find_J_F_app(q,f_ind,p,rho_int)
% Finds the Jacobian of the applied forces.

% Declare the global variables.
global rho_ext g;

% Initialise J_F_app as an empty matrix.
J_F_app = sparse(3*size(q,1),3*size(q,1));

% Calculate dp_dq outside the loop as it is the same for every
element.
dp_dq = (1/3)*(rho_ext - rho_int)*g*[0 0 1 0 0 1 0 0 1];

% Calculate the element Jacobian for each element and add to the
full
% Jacobian.
for i = 1:size(f_ind,1)
    % Assign the node positions.
    x1 = q(f_ind(i,1),2);
    y1 = q(f_ind(i,1),3);
    z1 = q(f_ind(i,1),4);

    x2 = q(f_ind(i,2),2);
    y2 = q(f_ind(i,2),3);
    z2 = q(f_ind(i,2),4);

    x3 = q(f_ind(i,3),2);
    y3 = q(f_ind(i,3),3);
    z3 = q(f_ind(i,3),4);

    % Calculate the cross product of two sides of the element and
use this
    % to calculate the product of the element's area and unit
normal.
    a = [x2-x1;y2-y1;z2-z1];
    b = [x3-x1;y3-y1;z3-z1];
    axb = cross(a,b);

    Sn = 0.5*[axb;axb;axb];

    % Differentiate the cross product with respect to position of
each
    % node, finding the components in x, y, and z.
    daxb_x = [0 z2-z3 y3-y2 0 z3-z1 y1-y3 0 z1-z2 y2-y1];

    daxb_y = [z3-z2 0 x2-x3 z1-z3 0 x3-x1 z2-z1 0 x1-x2];

    daxb_z = [y2-y3 x3-x2 0 y3-y1 x1-x3 0 y1-y2 x2-x1 0];

    % Form the matrix of differentiated area-unit normal products
with
    % respect to position of each node.
    dSn =
0.5*[daxb_x;daxb_y;daxb_z;daxb_x;daxb_y;daxb_z;daxb_x;daxb_y;...
        daxb_z];

    % Form the element Jacobian.
    dFe = (1/3)*(p(i,1)*dSn + Sn*dp_dq);
    % Add the element Jacobian to the system Jacobian.
    % Add to dF1_dq1.
    J_F_app(3*f_ind(i,1)-2:3*f_ind(i,1),3*f_ind(i,1)-2:3*f_ind(i,1))
= ...
        J_F_app(3*f_ind(i,1)-2:3*f_ind(i,1),3*f_ind(i,1)-
2:3*f_ind(i,1))...
        + dFe(1:3,1:3);

    % Add to dF1_dq2.
    J_F_app(3*f_ind(i,1)-2:3*f_ind(i,1),3*f_ind(i,2)-2:3*f_ind(i,2))
= ...
        J_F_app(3*f_ind(i,1)-2:3*f_ind(i,1),3*f_ind(i,2)-
2:3*f_ind(i,2))...
        + dFe(1:3,4:6);

    % Add to dF1_dq3.
    J_F_app(3*f_ind(i,1)-2:3*f_ind(i,1),3*f_ind(i,3)-2:3*f_ind(i,3))
= ...
        J_F_app(3*f_ind(i,1)-2:3*f_ind(i,1),3*f_ind(i,3)-
2:3*f_ind(i,3))...
        + dFe(1:3,7:9);


    % Add to dF2_dq1.
    J_F_app(3*f_ind(i,2)-2:3*f_ind(i,2),3*f_ind(i,1)-2:3*f_ind(i,1))
= ...
        J_F_app(3*f_ind(i,2)-2:3*f_ind(i,2),3*f_ind(i,1)-
2:3*f_ind(i,1))...
        + dFe(4:6,1:3);

    % Add to dF2_dq2.
    J_F_app(3*f_ind(i,2)-2:3*f_ind(i,2),3*f_ind(i,2)-2:3*f_ind(i,2))
= ...
        J_F_app(3*f_ind(i,2)-2:3*f_ind(i,2),3*f_ind(i,2)-
2:3*f_ind(i,2))...
        + dFe(4:6,4:6);

    % Add to dF2_dq3.
    J_F_app(3*f_ind(i,2)-2:3*f_ind(i,2),3*f_ind(i,3)-2:3*f_ind(i,3))
= ...
        J_F_app(3*f_ind(i,2)-2:3*f_ind(i,2),3*f_ind(i,3)-
2:3*f_ind(i,3))...
        + dFe(4:6,7:9);


    % Add to dF3_dq1.
    J_F_app(3*f_ind(i,3)-2:3*f_ind(i,3),3*f_ind(i,1)-2:3*f_ind(i,1))
= ...
        J_F_app(3*f_ind(i,3)-2:3*f_ind(i,3),3*f_ind(i,1)-
2:3*f_ind(i,1))...
        + dFe(7:9,1:3);

    % Add to dF3_dq2.
    J_F_app(3*f_ind(i,3)-2:3*f_ind(i,3),3*f_ind(i,2)-2:3*f_ind(i,2))
= ...
        J_F_app(3*f_ind(i,3)-2:3*f_ind(i,3),3*f_ind(i,2)-
2:3*f_ind(i,2))...
        + dFe(7:9,4:6);
    % Add to dF3_dq3.
    J_F_app(3*f_ind(i,3)-2:3*f_ind(i,3),3*f_ind(i,3)-2:3*f_ind(i,3))
= ...
        J_F_app(3*f_ind(i,3)-2:3*f_ind(i,3),3*f_ind(i,3)-
2:3*f_ind(i,3))...
        + dFe(7:9,7:9);
end
```

```
function [J_F_reac,stress,dstress_du] =
find_J_F_reac(q,e_ind,mat_props,l,stress,strain,mem_ind,E_mem,poi,st
ress_pre_mem,t,q0)
% Finds the Jacobian of the reaction forces.

% Split the matrix of material properties into its constituent
parts.
L = mat_props(:,1);
E = mat_props(:,2);
E_comp = mat_props(:,3);
A = mat_props(:,4);
stress_pre = mat_props(:,5);

% Initialise J_F_reac as an empty matrix.
J_F_reac = sparse(3*size(q,1),3*size(q,1));

% Calculate the element Jacobian for each cable element and add it
% to the full Jacobian.
for i = 1:size(e_ind,1)
    % Assign the node positions for the two nodes on edge i.
    x1 = q(e_ind(i,1),2);
    y1 = q(e_ind(i,1),3);
    z1 = q(e_ind(i,1),4);

    x2 = q(e_ind(i,2),2);
    y2 = q(e_ind(i,2),3);
    z2 = q(e_ind(i,2),4);

    % Calculate the directional cosines of the rotation from local
    % coordinates to global coordinates.
    lx = (x2 - x1)/l(i,1);
    mx = (y2 - y1)/l(i,1);
    nx = (z2 - z1)/l(i,1);

    % Form part of the element Jacobian.
    k = [lx^2 lx*mx lx*nx;
         lx*mx mx^2 mx*nx;
         lx*nx mx*nx nx^2];

    % Form another part of the element Jacobian.
    C = [mx^2+nx^2 -lx*mx -lx*nx;-mx*lx lx^2+nx^2 -mx*nx;-nx*lx -
nx*mx lx^2+mx^2];

    % Form the elastic part of the full element Jacobian in global
    % coordinates.
    if stress(i,1) > 0
        ke = (E(i,1)*A(i,1)/L(i,1))*[k -k;-k k];
    else
        ke = zeros(6,6);
    end

    % Add the geometric part to the full element Jacobian in global
    % coordinates.
    ke = ke + (A(i,1)*stress(i,1)/l(i,1))*[C -C;-C C];

    % Add to dF1_dq1.
    J_F_reac(3*e_ind(i,1)-2:3*e_ind(i,1),3*e_ind(i,1)-
2:3*e_ind(i,1)) = ...
```

```
        J_F_reac(3*e_ind(i,1)-2:3*e_ind(i,1),3*e_ind(i,1)-
2:3*e_ind(i,1)) + ...
        ke(1:3,1:3);

    % Add to dF1_dq2.
    J_F_reac(3*e_ind(i,1)-2:3*e_ind(i,1),3*e_ind(i,2)-
2:3*e_ind(i,2)) = ...
        J_F_reac(3*e_ind(i,1)-2:3*e_ind(i,1),3*e_ind(i,2)-
2:3*e_ind(i,2)) + ...
        ke(1:3,4:6);

    % Add to dF2_dq1.
    J_F_reac(3*e_ind(i,2)-2:3*e_ind(i,2),3*e_ind(i,1)-
2:3*e_ind(i,1)) = ...
        J_F_reac(3*e_ind(i,2)-2:3*e_ind(i,2),3*e_ind(i,1)-
2:3*e_ind(i,1)) + ...
        ke(4:6,1:3);

    % Add to dF2_dq2.
    J_F_reac(3*e_ind(i,2)-2:3*e_ind(i,2),3*e_ind(i,2)-
2:3*e_ind(i,2)) = ...
        J_F_reac(3*e_ind(i,2)-2:3*e_ind(i,2),3*e_ind(i,2)-
2:3*e_ind(i,2)) + ...
        ke(4:6,4:6);
end


for i = 1:size(mem_ind,1)
    % Form the rotation matrix R.
    a = q0(mem_ind(i,2),2:4)' - q0(mem_ind(i,1),2:4)';
    b = q0(mem_ind(i,3),2:4)' - q0(mem_ind(i,1),2:4)';
    axb = cross(a,b);

    R = [a/norm(a),cross(axb/norm(axb),a/norm(a)),axb/norm(axb)];

    % Set u using the nodal displacements.
    disps = q(:,2:4) - q0(:,2:4);
    u =
[R,zeros(3,3),zeros(3,3);zeros(3,3),R,zeros(3,3);zeros(3,3),zeros(3,
3),R]'...

*[disps(mem_ind(i,1),:)';disps(mem_ind(i,2),:)';disps(mem_ind(i,3),:
)'];

    % Find the local node coordinates with origin at node 1.
    q_loc1 = R'*(q0(mem_ind(i,1),2:4)' - q0(mem_ind(i,1),2:4)');
    q_loc2 = R'*(q0(mem_ind(i,2),2:4)' - q0(mem_ind(i,1),2:4)');
    q_loc3 = R'*(q0(mem_ind(i,3),2:4)' - q0(mem_ind(i,1),2:4)');

    % Move the local origin to the point on side a where a line
    % perpendicular to a will pass through node 3.
    q_loc1 = q_loc1 - [q_loc3(1,1);0;0];
    q_loc2 = q_loc2 - [q_loc3(1,1);0;0];
    q_loc3 = q_loc3 - [q_loc3(1,1);0;0];

    % Assign names to 3 components of local node coordinates.
    x1 = q_loc1(1,1); y1 = q_loc1(2,1); %z1 = q_loc1(3,1);
    x2 = q_loc2(1,1); y2 = q_loc2(2,1); %z2 = q_loc2(3,1);
    x3 = q_loc3(1,1); y3 = q_loc3(2,1); %z3 = q_loc3(3,1);
```

```
% Find the area of the element.
S = 0.5*norm(axb);

% Double the area of the element.
twoS = 2*S;

% Find the coefficients of the shape functions.
%a1 = (x2*y3-x3*y2)/twoS;
b1 = (y2-y3)/twoS;
c1 = -(x2-x3)/twoS;

%a2 = (x3*y1-x1*y3)/twoS;
b2 = (y3-y1)/twoS;
c2 = -(x3-x1)/twoS;

%a3 = (x1*y2-x2*y1)/twoS;
b3 = (y1-y2)/twoS;
c3 = -(x1-x2)/twoS;

% Calculate the matrix of linear derivative coefficients.
B0 = [b1 0  0  b2 0  0  b3 0  0;
      0  c1 0  0  c2 0  0  c3 0;
      c1 b1 0  c2 b2 0  c3 b3 0];

% Find G, theta, and A, to form matrix of nonlinear derivative
% coefficients.
P = [b1 0  0  b2 0  0  b3 0  0;
     0  b1 0  0  b2 0  0  b3 0;
     0  0  b1 0  0  b2 0  0  b3];

Q = [c1 0  0  c2 0  0  c3 0  0;
     0  c1 0  0  c2 0  0  c3 0;
     0  0  c1 0  0  c2 0  0  c3];

Pu = P*u;
Qu = Q*u;

A = [Pu'        zeros(1,3);
     zeros(1,3) Qu';
     Qu'        Pu'];

G = [P;Q];

theta = [Pu;Qu];

% Calculate the strain in the element.
strain = B0*u + 0.5*A*theta;

% Create the elastic matrix for the element in plane stress.
D = (E_mem(i,1)/(1-poi(i,1)^2))*[1   poi(i,1) 0;
                                 poi(i,1) 1   0;
                                 0   0   (1-poi(i,1))/2];

% Calculate the stress in the element.
stress = D*strain + R'*stress_pre_mem(i,:)';

% Calculate the principal stresses and strains.
```

```
    prin_max = 0.5*(stress(1,1) + stress(2,1)) + ((0.5*(stress(1,1)
- stress(2,1)))^2 + stress(3,1)^2)^(0.5);
    prin_min = 0.5*(stress(1,1) + stress(2,1)) - ((0.5*(stress(1,1)
- stress(2,1)))^2 + stress(3,1)^2)^(0.5);

    prin_max_pre = 0.5*(stress_pre_mem(i,1) + stress_pre_mem(i,2)) +
((0.5*(stress_pre_mem(i,1) - stress_pre_mem(i,2)))^2 +
stress_pre_mem(i,3)^2)^(0.5);
    prin_min_pre = 0.5*(stress_pre_mem(i,1) + stress_pre_mem(i,2)) -
((0.5*(stress_pre_mem(i,1) - stress_pre_mem(i,2)))^2 +
stress_pre_mem(i,3)^2)^(0.5);

    prin_strain_max = 0.5*(strain(1,1) + strain(2,1)) +
((0.5*(strain(1,1) - strain(2,1)))^2 + strain(3,1)^2)^(0.5);
    prin_strain_min = 0.5*(strain(1,1) + strain(2,1)) -
((0.5*(strain(1,1) - strain(2,1)))^2 + strain(3,1)^2)^(0.5);

    % Calculate the angle between the principal coordinates and
local
    % coordinates.
    prin_theta = 0.5*atan(2*stress(3,1)/(stress(1,1)-stress(2,1)));

    % Create Reuter's matrix and the pseudo-vector matrix form of
the
    % rotation matrix.
    Reut = [1,0,0;0,1,0;0,0,2];
    c = cos(prin_theta);
    s = sin(prin_theta);
    T = [c^2,s^2,2*s*c;s^2,c^2,-2*s*c;-s*c,s*c,c^2-s^2];

    % No wrinkling.
    if prin_min > 0
        % Form a matrix of stresses.
        M = [stress(1,1) 0 0 stress(3,1) 0 0;
             0 stress(1,1) 0 0 stress(3,1) 0;
             0 0 stress(1,1) 0 0 stress(3,1);
             stress(3,1) 0 0 stress(2,1) 0 0;
             0 stress(3,1) 0 0 stress(2,1) 0;
             0 0 stress(3,1) 0 0 stress(2,1)];

        % Calculate the elastic and geometric element stiffness
matrices.
        Ke = t(i,1)*S*(B0 + A*G)'*D*(B0 + A*G);
        Kg = t(i,1)*S*G'*M*G;
        dstress_du = D*(B0 + A*G);

        % Form the full element stiffness matrix.
        Kloc = Ke + Kg;

    % Wrinkling.
    else
        % Differentiate tan(2*theta) wrt u.
        dstress_du = D*(B0 + A*G);
        dtan_2theta_du = (2*(stress(1,1)-
stress(2,1))*dstress_du(3,:) - 2*stress(3,1)*(dstress_du(1,:)-
dstress_du(2,:)))/(stress(1,1)-stress(2,1))^2;

        % Differentiate theta wrt u.
        dtheta_du = (1/(2*(1+(2*stress(3,1)/(stress(1,1)-
stress(2,1)))^2)))*dtan_2theta_du;

        % Differentiate the matrix T wrt u.
        dT_dtheta = [-2*c*s,2*c*s,2*(c^2-s^2);2*c*s,-2*c*s,2*(s^2-
c^2);s^2-c^2,c^2-s^2,-4*c*s];
        dT_du1 = dT_dtheta*dtheta_du(1,1);
        dT_dv1 = dT_dtheta*dtheta_du(1,2);
        dT_dw1 = dT_dtheta*dtheta_du(1,3);
        dT_du2 = dT_dtheta*dtheta_du(1,4);
        dT_dv2 = dT_dtheta*dtheta_du(1,5);
        dT_dw2 = dT_dtheta*dtheta_du(1,6);
        dT_du3 = dT_dtheta*dtheta_du(1,7);
        dT_dv3 = dT_dtheta*dtheta_du(1,8);
        dT_dw3 = dT_dtheta*dtheta_du(1,9);

        % Initialise the dD matrices.
        dD_du1 = zeros(3,3);
        dD_dv1 = zeros(3,3);
        dD_dw1 = zeros(3,3);
        dD_du2 = zeros(3,3);
        dD_dv2 = zeros(3,3);
        dD_dw2 = zeros(3,3);
        dD_du3 = zeros(3,3);
        dD_dv3 = zeros(3,3);
        dD_dw3 = zeros(3,3);

        % Uniaxial.
        if prin_strain_max > 0
            % Changing Poisson's ratio method.
            P = (strain(1)-strain(2))/(prin_strain_max-
prin_strain_min);
            Q = strain(3)/(prin_strain_max-prin_strain_min);

            D = (E_mem(i,1)/4)*[2*(1+P) 0 Q;
                                0 2*(1-P) Q;
                                Q Q 1];

            dstrain_du = B0 + A*G;

            dprin_strain_du = T*dstrain_du +
[dT_du1*strain,dT_dv1*strain,...

dT_dw1*strain,dT_du2*strain,dT_dv2*strain,dT_dw2*strain,...
                dT_du3*strain,dT_dv3*strain,dT_dw3*strain];

            dP_du = ((prin_strain_max -
prin_strain_min)*(dstrain_du(1,:) - dstrain_du(2,:)) - (strain(1,1)
- strain(2,1))*(dprin_strain_du(1,:) -
dprin_strain_du(2,:)))/(prin_strain_max - prin_strain_min)^2;
            dQ_du = ((prin_strain_max -
prin_strain_min)*dstrain_du(3,:) - strain(3,1)*(dprin_strain_du(1,:)
- dprin_strain_du(2,:)))/(prin_strain_max - prin_strain_min)^2;

            dD_du1 = (E_mem(i,1)/4)*[2*dP_du(1,1) 0 dQ_du(1,1)
                                          0 -2*dP_du(1,1) dQ_du(1,1)
                                          dQ_du(1,1) dQ_du(1,1) 0];

            dD_dv1 = (E_mem(i,1)/4)*[2*dP_du(1,2) 0 dQ_du(1,2)
```

```
                                          0 -2*dP_du(1,2) dQ_du(1,2)
                                          dQ_du(1,2) dQ_du(1,2) 0];

            dD_dw1 = (E_mem(i,1)/4)*[2*dP_du(1,3) 0 dQ_du(1,3)
                                          0 -2*dP_du(1,3) dQ_du(1,3)
                                          dQ_du(1,3) dQ_du(1,3) 0];

            dD_du2 = (E_mem(i,1)/4)*[2*dP_du(1,4) 0 dQ_du(1,4)
                                          0 -2*dP_du(1,4) dQ_du(1,4)
                                          dQ_du(1,4) dQ_du(1,4) 0];

            dD_dv2 = (E_mem(i,1)/4)*[2*dP_du(1,5) 0 dQ_du(1,5)
                                          0 -2*dP_du(1,5) dQ_du(1,5)
                                          dQ_du(1,5) dQ_du(1,5) 0];

            dD_dw2 = (E_mem(i,1)/4)*[2*dP_du(1,6) 0 dQ_du(1,6)
                                          0 -2*dP_du(1,6) dQ_du(1,6)
                                          dQ_du(1,6) dQ_du(1,6) 0];

            dD_du3 = (E_mem(i,1)/4)*[2*dP_du(1,7) 0 dQ_du(1,7)
                                          0 -2*dP_du(1,7) dQ_du(1,7)
                                          dQ_du(1,7) dQ_du(1,7) 0];

            dD_dv3 = (E_mem(i,1)/4)*[2*dP_du(1,8) 0 dQ_du(1,8)
                                          0 -2*dP_du(1,8) dQ_du(1,8)
                                          dQ_du(1,8) dQ_du(1,8) 0];

            dD_dw3 = (E_mem(i,1)/4)*[2*dP_du(1,9) 0 dQ_du(1,9)
                                          0 -2*dP_du(1,9) dQ_du(1,9)
                                          dQ_du(1,9) dQ_du(1,9) 0];
        % Biaxial.
        else
            b = 1e-6*E_mem(i,1); % THESE MUST BE THE SAME IN THIS
AND K!

            D = [0 0 0;
                 0 0 0;
                 0 0 0];

        end

        % Calculate the product of dD_du and strain.
        dD_du_strain =
[dD_du1*strain,dD_dv1*strain,dD_dw1*strain,...
            dD_du2*strain,dD_dv2*strain,dD_dw2*strain,...
            dD_du3*strain,dD_dv3*strain,dD_dw3*strain];

        % Form the local element elastic stiffness matrix.
        Ke = t(i,1)*S*(B0 + A*G)'*(dD_du_strain + D*(B0 + A*G));

        % Calculate the stresses using the new elastic matrix.
        stress = D*strain + R'*stress_pre_mem(i,:)';

        % Form a matrix of stresses.
        M = [stress(1,1) 0 0 stress(3,1) 0 0;
             0 stress(1,1) 0 0 stress(3,1) 0;
             0 0 stress(1,1) 0 0 stress(3,1);
             stress(3,1) 0 0 stress(2,1) 0 0;
             0 stress(3,1) 0 0 stress(2,1) 0;
             0 0 stress(3,1) 0 0 stress(2,1)];

        % Calculate the local element geometric stiffness matrices.
        Kg = t(i,1)*S*G'*M*G;

        % Form the full element stiffness matrix.
        Kloc = Ke + Kg;
    end


    % Rotate the element stiffness matrix into the global coordinate
    % system.
    K =
[R,zeros(3,3),zeros(3,3);zeros(3,3),R,zeros(3,3);zeros(3,3),zeros(3,
3),R]*Kloc...

*[R,zeros(3,3),zeros(3,3);zeros(3,3),R,zeros(3,3);zeros(3,3),zeros(3
,3),R]';


    % Add to dF1_dq1.
    J_F_reac(3*mem_ind(i,1)-2:3*mem_ind(i,1),3*mem_ind(i,1)-
2:3*mem_ind(i,1)) = ...
        J_F_reac(3*mem_ind(i,1)-2:3*mem_ind(i,1),3*mem_ind(i,1)-
2:3*mem_ind(i,1)) + ...
        K(1:3,1:3);

    % Add to dF1_dq2.
    J_F_reac(3*mem_ind(i,1)-2:3*mem_ind(i,1),3*mem_ind(i,2)-
2:3*mem_ind(i,2)) = ...
        J_F_reac(3*mem_ind(i,1)-2:3*mem_ind(i,1),3*mem_ind(i,2)-
2:3*mem_ind(i,2)) + ...
        K(1:3,4:6);

    % Add to dF1_dq3.
    J_F_reac(3*mem_ind(i,1)-2:3*mem_ind(i,1),3*mem_ind(i,3)-
2:3*mem_ind(i,3)) = ...
        J_F_reac(3*mem_ind(i,1)-2:3*mem_ind(i,1),3*mem_ind(i,3)-
2:3*mem_ind(i,3)) + ...
        K(1:3,7:9);


    % Add to dF2_dq1.
    J_F_reac(3*mem_ind(i,2)-2:3*mem_ind(i,2),3*mem_ind(i,1)-
2:3*mem_ind(i,1)) = ...
        J_F_reac(3*mem_ind(i,2)-2:3*mem_ind(i,2),3*mem_ind(i,1)-
2:3*mem_ind(i,1)) + ...
        K(4:6,1:3);

    % Add to dF2_dq2.
    J_F_reac(3*mem_ind(i,2)-2:3*mem_ind(i,2),3*mem_ind(i,2)-
2:3*mem_ind(i,2)) = ...
        J_F_reac(3*mem_ind(i,2)-2:3*mem_ind(i,2),3*mem_ind(i,2)-
2:3*mem_ind(i,2)) + ...
        K(4:6,4:6);

    % Add to dF2_dq3.
    J_F_reac(3*mem_ind(i,2)-2:3*mem_ind(i,2),3*mem_ind(i,3)-
2:3*mem_ind(i,3)) = ...
```

```
        J_F_reac(3*mem_ind(i,2)-2:3*mem_ind(i,2),3*mem_ind(i,3)-
2:3*mem_ind(i,3)) + ...
        K(4:6,7:9);



    % Add to dF3_dq1.
    J_F_reac(3*mem_ind(i,3)-2:3*mem_ind(i,3),3*mem_ind(i,1)-
2:3*mem_ind(i,1)) = ...
        J_F_reac(3*mem_ind(i,3)-2:3*mem_ind(i,3),3*mem_ind(i,1)-
2:3*mem_ind(i,1)) + ...
        K(7:9,1:3);

    % Add to dF3_dq2.
    J_F_reac(3*mem_ind(i,3)-2:3*mem_ind(i,3),3*mem_ind(i,2)-
2:3*mem_ind(i,2)) = ...
        J_F_reac(3*mem_ind(i,3)-2:3*mem_ind(i,3),3*mem_ind(i,2)-
2:3*mem_ind(i,2)) + ...
        K(7:9,4:6);

    % Add to dF3_dq3.
    J_F_reac(3*mem_ind(i,3)-2:3*mem_ind(i,3),3*mem_ind(i,3)-
2:3*mem_ind(i,3)) = ...
        J_F_reac(3*mem_ind(i,3)-2:3*mem_ind(i,3),3*mem_ind(i,3)-
2:3*mem_ind(i,3)) + ...
        K(7:9,7:9);
end
```

D. Matlab Code for Valve Control

```
function tank_valve_control
% AT ALL POINTS, INCLUDE CHECKS FOR DANGEROUSLY HIGH/LOW PRESSURES
AND USE
% FLAGS AND BREAK STATEMENTS IF NECESSARY TO MOVE TO A SAFE STATE
AND SEND
% EMAIL ALERTS (IF POSSIBLE).

% Set the upper and lower pressure thresholds (p_hi and p_lo) and
calculate
% upper and lower starting pressures (p_start_hi and p_start_lo).
p_hi = 0.35e5;
p_lo = 0.1e5;
p_start_hi = p_hi - (p_hi - p_lo)/10;
p_start_lo = p_lo + (p_hi - p_lo)/10;

dio = digitalio('mcc','0'); % Create digital i/o object
addline(dio,0:3, 0,'out');  % Add 4 do lines, one for each valve.

ai = analoginput('mcc','0'); % Create analog input object.
addchannel(ai,0:1); % Add 2 ai channels, one for each pressure
transducer.



% START OF DAY:
% (run this before commencing cycling)

disp('Commencing start routine')

p_reg = zeros(0,2);
hold on

tic

% Get first pressure sample.
[pL,pR] = readP(ai);

% Check if the left bag has a lower pressure than the right bag.
if pL < pR
    disp('Start: right is fuller than left')

    if pL < p_start_lo
        % Fill left until above p_start_lo.
        [T,dispStr] = setValves('a_to_L');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pL < p_start_lo
            [pL,pR] = readP(ai);
%             T = toc;
%             if rem(fix(T),10) == 0
%                 p_reg = [p_reg;pL,pR];
%                 plot(fix(T),p_reg(end,1),'.r');
plot(fix(T),p_reg(end,2),'.k'); drawnow
%             end
        end
    else
```

```
        % Empty left until less than or equal to p_start_lo.
        [T,dispStr] = setValves('L_to_a');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pL >= p_start_lo
            [pL,pR] = readP(ai);
        end
    end

    if pR < p_start_hi
        % Fill right until above p_start_hi.
        [T,dispStr] = setValves('a_to_R');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pR < p_start_hi
            [pL,pR] = readP(ai);
        end
    else
        % Empty right until less than or equal to p_start_hi.
        [T,dispStr] = setValves('R_to_a');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pR >= p_start_hi
            [pL,pR] = readP(ai);
        end
    end
else
    disp('Start: left is fuller than right')

    if pL < p_start_hi
        % Fill left until above p_start_hi.
        [T,dispStr] = setValves('a_to_L');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pL < p_start_hi
            [pL,pR] = readP(ai);
        end
    else
        % Empty left until less than or equal to p_start_hi.
        [T,dispStr] = setValves('L_to_a');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pL >= p_start_hi
            [pL,pR] = readP(ai);
        end
    end

    if pR < p_start_lo
        % Fill right until above p_start_lo.
        [T,dispStr] = setValves('a_to_R');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pR < p_start_lo
            [pL,pR] = readP(ai);
        end
    else
        % Empty right until less than or equal to p_start_lo.
        [T,dispStr] = setValves('R_to_a');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pR >= p_start_lo
            [pL,pR] = readP(ai);
        end
    end
end

% Set the valves to idle (no air being pumped to/from either bag).
```

```
setValves('idle')

T_start = toc;
T = fix(clock); hour = num2str(T(4)); min = num2str(T(5));
disp('Start routine complete')
disp('Start routine took (in m,s):')
disp([floor(T_start/60),round(rem(T_start,60))])



% BEGIN CYCLING:
% (only works properly if start routine has recently been completed)

T = fix(clock); hour = num2str(T(4)); min = num2str(T(5));
disp('Commencing cycle routine')

cycle_count = 0;
break_loop = 0;
while break_loop == 0

    tic

    [pL,pR] = readP(ai);

    if pL < pR
        % Send right to left until either threshold is reached.
        [T,dispStr] = setValves('R_to_L');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pR > p_lo && pL < p_hi
            [pL,pR] = readP(ai);
        end

        % Ensure the bag that did not cross the threshold is
        % inflated/deflated (using atm.) until it does.
        if pR <= p_lo
            [T,dispStr] = setValves('a_to_L'); % Continue to fill
left bag from atm.
            disp([T{1} ':' T{2} ' ' dispStr])
            while pL < p_hi
                [pL,pR] = readP(ai);
            end
        else
            [T,dispStr] = setValves('R_to_a'); % Continue to empty
right bag to atm.
            disp([T{1} ':' T{2} ' ' dispStr])
            while pR > p_lo
                [pL,pR] = readP(ai);
            end
        end
    else
        % Send left to right until either threshold is reached.
        [T,dispStr] = setValves('L_to_R');
        disp([T{1} ':' T{2} ' ' dispStr])
        while pR < p_hi && pL > p_lo
            [pL,pR] = readP(ai);
        end

        % Ensure the bag that did not cross the threshold is
        % inflated/deflated (using atm.) until it does.
        if pL <= p_lo
            [T,dispStr] = setValves('a_to_R'); % Continue to fill
right bag from atm.
            disp([T{1} ':' T{2} ' ' dispStr])
            while pR < p_hi
                [pL,pR] = readP(ai);
            end
        else
            [T,dispStr] = setValves('L_to_a'); % Continue to empty
left bag to atm.
            disp([T{1} ':' T{2} ' ' dispStr])
            while pL > p_lo
                [pL,pR] = readP(ai);
            end
        end
    end

    % Set the valves to idle (no air being pumped to/from either
bag).
    setValves('idle')

    T_cyc = toc;
    cycle_count = cycle_count + 0.5;
    T = fix(clock); hour = num2str(T(4)); min = num2str(T(5));
    disp('Cycle: completed cycles:')
    disp(cycle_count)

    disp('Cycle: last inflation took (in m,s):')
    disp([floor(T_cyc/60),round(rem(T_cyc,60))])

    % Check any stop conditions and set break_loop = 1 if any are
met, then
    % send email using the following (making sure the lines of code
are not
    % in a loop!!!)
%     setpref('Internet','E_mail','eaxajp@nottingham.ac.uk')
%
setpref('Internet','SMTP_Server','EXCHANGE1.ad.nottingham.ac.uk')
%     setpref('Internet','SMTP_Username','eaxajp@nottingham.ac.uk')
%     setpref('Internet','SMTP_Password','ENTER PASSWORD HERE')
%
sendmail({'andrewpimm@gmail.com','eaxajp@nottingham.ac.uk'},'Thresho
ld reached')
end

T = fix(clock); hour = num2str(T(4)); min = num2str(T(5));
disp('End of cycling')

% Set the valves to idle (no air being pumped to/from either bag).
setValves('idle')

delete(dio)
delete(ai)
clear dio ai


function dispTime
T = fix(clock);
hour = num2str(T(4)); min = num2str(T(5)); sec = num2str(T(6));
```

```
disp([hour ':' min ':' sec ' Blah blah'])

function [pL,pR] = readP(ai)
    sample = getsample(ai); pL = 1e5*sample(1)/5; pR =
1e5*sample(2)/5;

function [T,dispStr] = setValves(vMode)
% Set the valves to positions specified in vMode.

    switch vMode
        case 'idle'
            lineSet = [0,0,0,0];
            dispStr = 'idle';
        case 'a_to_L'
            lineSet = [0,1,0,0];
            dispStr = 'filling left from atm.';
        case 'a_to_R'
            lineSet = [0,1,1,1];
            dispStr = 'filling right from atm.';
        case 'a_to_both'
            lineSet = [0,1,0,1];
            dispStr = 'filling both from atm.';
        case 'L_to_a'
            lineSet = [1,0,0,0];
            dispStr = 'emptying left to atm.';
        case 'R_to_a'
            lineSet = [1,0,1,1];
            dispStr = 'emptying right to atm.';
        case 'both_to_a'
            lineSet = [1,0,1,0];
            dispStr = 'emptying both to atm.';
        case 'L_to_R'
            lineSet = [1,1,0,0];
            dispStr = 'filling right from left';
        case 'R_to_L'
            lineSet = [1,1,1,1];
            dispStr = 'filling left from right';
        otherwise
            putvalue(dio.Line(1:4),[0,0,0,0])
            error('Valve setting string not recognised.  Set to
idle.')
    end

    T = fix(clock); hour = num2str(T(4)); min = num2str(T(5));
    T = {hour,min};
    putvalue(dio.Line(1:4),lineSet)
```