Echinodome Design Curves using a Task Farm

Richard Boyd

EPCC–SS94–14

Echinodome Design Curves using a Task Farm

Richard Boyd

EPCC–SS94–14

September 1994

Abstract

The purpose of this project is to use parallel computing in the field of structural engineering by incorporating message passing into a serial program written to predict the shape of an echinodome — an underwater storage structure under development at the University of Edinburgh. Its name is in deference to the family of shapes adopted by the underwater marine animal — the sea urchin. The design of such a structure is complex and all aspects of construction, from material requirements in compliance with British Standards, to building and maintenance costs, are of great importance.

The aim of this project is to predict the most optimum shape for the structure under consideration of various boundary conditions and design criteria and to provide the designer with a method of assessing the validity of the optimum shape in accordance with a clients' requirements. The quantity of data is sufficiently large for the program to benefit from the advantages of parallel computing.

1 Introduction

In Civil and Structural Engineering the design of structures involves both a conceptual and an analytical/numerical approach. It is usually the analytical approach which is the most time consuming, hence there are many aids to assist the engineer in shortening this part of the process. This report is written both for those with and without knowledge of structural engineering, so following a description of the projects' aims, a background to design aids is given, followed by background information to the structure under consideration in this project (Section 2). Industrial applications are described in section 2.4. Details of the original serial program written and its need for parallelisation are given in sections 3 and 4, followed by the format of parallelisation (Section 5). Section 6 compares and analyses results from both the original and modified code and presents all the results from the project graphically. Future developments proposed for the project are discussed in section 7 and section 8 highlights the conclusions as to the benefits of parallelising the program.

1.1 Aim of the Project

The aim of this project is to produce a suite of design curves, using parallel computing, for the echinodome, a spheroidal underwater storage structure for liquids such as oil and liquid natural gas (LNG), currently under experimental investigation at the Department of Civil Engineering and Building Science, University of Edinburgh. A shape prediction program exists which calculates the desired shape of the structure and various design features, e.g. the dimensions, or the capacity of the structure, for given material properties and boundary conditions, and the results are presented in the form of design curves. The large number of possible design criteria and resulting suites of curves makes this an ideal candidate for parallelisation using a task farm.

1.2 Aids to Design

Computers have recently become more prominent in assisting in the design of structures. Computer Aided Design (CAD) facilities allow visualisation of structural frameworks and architectural features. Structural analysis software is also available, with which the time consuming task of repeated hand calculations is removed and a number of possible solutions for the design is achieved rapidly. This is the most efficient method of producing a safe, cost effective design without the normal paperwork usually involved.

Non-computing aids upon which the structural analysis software is based includes the British Standards for 'Construction and Maintenance of Structures', e.g., BS8110 (concrete), BS5790 (steel), BS5268 (timber). Within these are the basic concepts of the numerical design process with information on safety factors, material strengths and design charts/curves.

Design charts/curves are a graphical representation of resultant design features for a particular type of structure in consideraton of varying material strengths and boundary conditions. Typical design features include amount of steel required for a certain loading condition on reinfored concrete, diameter of a concrete shell, or condition of stress at various points within a steel girder. These all vary for different material strengths, hence several curves are usually graphed on one plot — from which the behaviour of the material can be comparatively examined.

All these design aids for the engineer assist in ensuring that a design is neither under nor overdesigned. An underdesigned structure is in danger of catastrophic collapse, an overdesigned structure is unnecessarily costly to construct.

2 The Echinodome

2.1 The Need For Underwater Storage

Within the latter half of the twentieth century concern has arisen due to the depletion of land based resources such as coal, timber, peat, etc., and the harnessing of marine resources, e.g. oil, liquid natural gas, has been seen as a viable alternative solution to the land resource problem. As the rate of exploitation of these marine resources continues to increase, so there is an apparent need for underwater and sea bed chambers for localised storage purposes, or within which related operations, such as water desalination, could take place.

The consequent change in environmental conditions from the demanding air-sea interface to the less demanding submerged environment lead to the theory of submerging such a structure, despite the fact that a surface structure would be more accessible in calm weather. Such conditions and needs are apparent in the North Sea where there are vast reserves of oil and liquid natural gas still to be harnessed.

2.2 Basis for Development of such Structures

The shape adopted is based on the membrane theory for shells of uniform strength by Laplace [Royles, 1980] and takes the form of a drop of liquid on a plane surface (Figure 1). Since the surface tension in the skin of a drop of liquid helps sustain the drop shape, this implies that there must be uniform strength in the skin. The pressure head at the apex and the stress in the membrane determines how spheroidal the shape is. For a given stress and head there is only one shell shape of uniform strength.

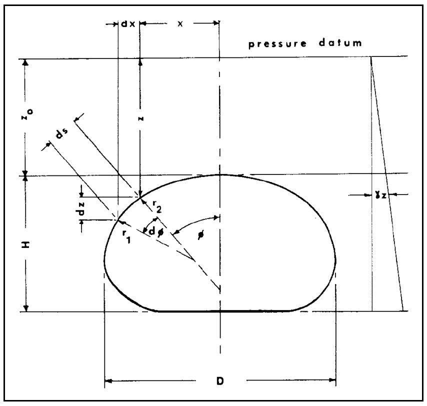


Figure 1: Drop shape on a plane surface

[Royles, 1980]

Through studies in the present marine life, observations on the sea urchin, a free flooding animal of the echinodermata (Figure 2), suggested that its shell conformed with the shape predicted by Laplace. An explanation for the shape of the shell, in accordance with recent evolutionary theory, is that it is formed in response to a nominally linear increase in external pressure over the depth of the shell caused by the tube feet (Figure 3) when the animal attaches itself to a surface[Royles, 1980]. It was also noted that sea urchins living at a greater depth tend to be more spheroidal than those living in shallower waters, but are smaller in size. This is consistent with reduced effort required in attachment to a surface in deeper waters having less demanding environmental conditions,but with a greater pressure head being exerted.

This drop shape was adopted by the Chicago Bridge and Iron Company in the construction of steel containers developed for use in the oil industry in 1928 (see Figure 4).

The stress distribution in a shell of uniform strength is not constant around the entire surface of the shell. As shown in Figure 5 the stress increases linearly towards the base of the shell. Hence the point under highest stress is the most critical. In designing the structure, the thickness of the shell skin must be able

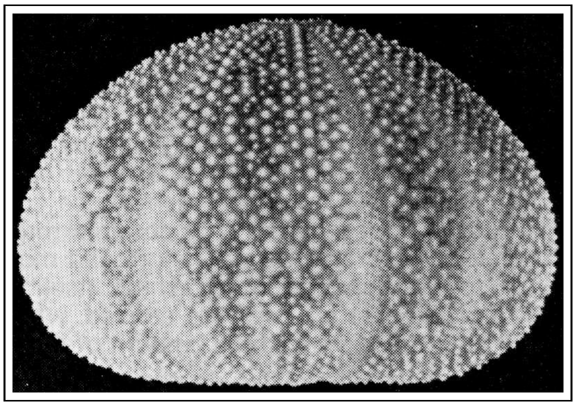


Figure 2: Elevation of a Sea Urchin

[Royles, 1980]

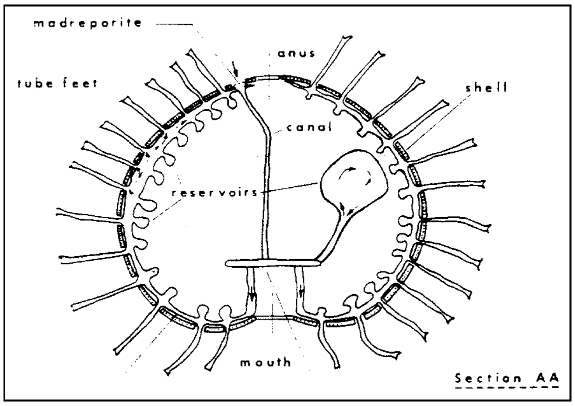


Figure 3: Cross–section through a Sea Urchin

[Royles, 1980]

to resist the highest stress — hence the thickness can be uniform all over the shell surface — or it can be made to increase as it approaches the base in order to resist buckling.

2.3 Development of Prototype Structure

Experimental investigation has been undertaken on a prototype structure at the University of Edinburgh [Royles, 1980]. A test shell was constructed using fibre glass and had a height of 380mm and a maximum horizontal diameter of 480mm. It was designed for a pressure head of 1.525m and a design stress of 0.46MN/m^2 with a shell thickness of 3.8mm. Strain gauges were connected at various points on the shell and connected to a computer. The strain values were logged for a series of pressure heads, and deformations assessed. The maximum head achieved was 2.135m without any signs of distress. This conforms to the British Standards saftey factor of 1.4, i.e. the maximum allowable head is 1.4 x design head. Collapse was not predicted below 9 x design head which may show an optimistic design or too conservative a design. This will be verified in further tests. Examination of the deformed shell after the completion of the experiment, (Figure 6), indicated that the zone of greatest distortion was near the base of the shell, i.e. the region under highest tensile stress, and the higher levels deformed axisymmetrically. This is because base of the shell is acting in shear. This complies with the theory on stress distribution.

The similarity of the shape of the test shell to the shape of the sea urchin led to the generic term “echinodome” for this structure [Royles, 1980].

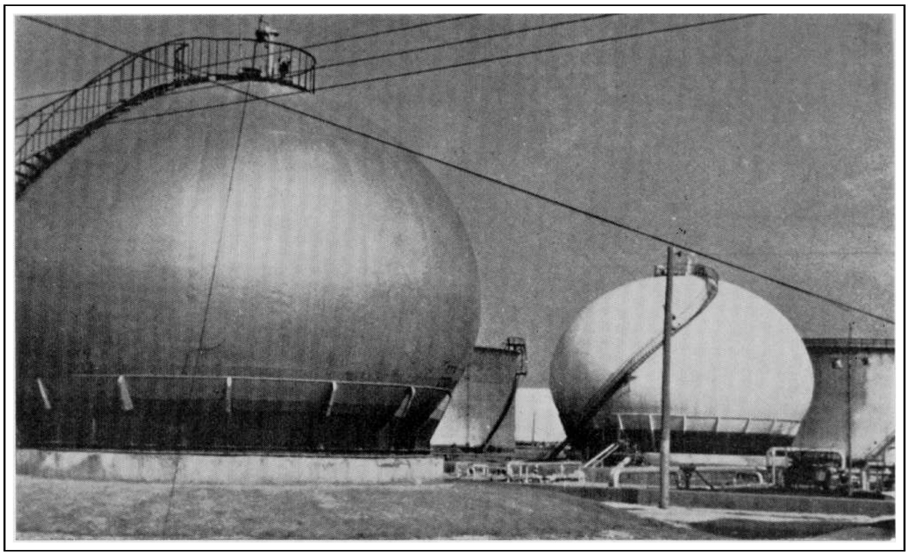


Figure 4: Tanks of Constant Strength – foreground 765 m^3, background 3970 m^3 [Royles, 1980]

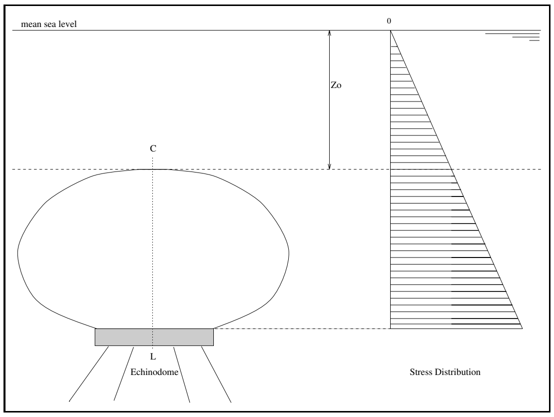


Figure 5: Stress Distribution throughout echinodome

2.4 Applications in Industry

2.4.1 The Oil Industry

Its use in the oil industry has already been mentioned. The production of oil is an ongoing process, since it proves expensive to even stop production for one day. However unpredictability of poor weather conditions, particularly in the North Sea, may lead to inaccessible conditions for the oil tankers to approach the oil production platform. Hence, currently, a storage tanker is required on site, to store extracted oil until conditions are calm enough for tankers to come and take the oil off the field. These resident storage tankers are very costly and the demanding air–sea interface can prove critical to its stability, despite its capacity. In this situation, the echinodome, tethered to the sea bed, would prove much cheaper, and safer, in the submerged environment. The production platform would be piped up to the echinodome in the same way as the permanent storage tanker. The oil tanker taking the oil off site could then be connected to the echinodome to empty it of its contents. The average size of a resident storage tanker is 300000 tonnes which could be catered for by one large echinodome or a field of smaller domes.

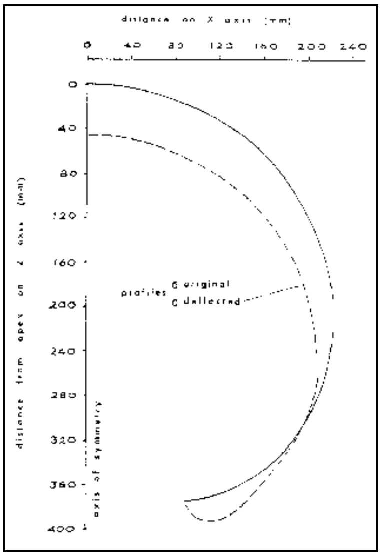


Figure 6: Deflected shape of Test Structure

[Royles, 1980]

2.4.2 Pollution Control Structures

The concept of a pollution control structure, for the storage of storm flow, is by no means new. However there are many problems associated with current methods of treatment such as above ground tanks, separate systems, on-line tanks, all of which produce land, aesthetic and odour problems and do not stop the discharge of storm flow into natural watercourses. Storing the storm flow underwater removes all of these problems and the echinodome provides a suitable solution [Innes, 1994].

2.4.3 Fresh Water Storage

Fresh water storage systems currently take the form of reservoirs, above ground tanks and wells, among others. Depending on the geographical situation these can have numerous problems. Southern England suffers almost every year with periodic drought during the summer months due to evaporation of water from reservoirs resulting in water bans disrupting industry and domestic users alike. Third World countries located near or on the equator experience prolonged drought periods, sometimes years, drying up wells and water tables which the people depend on with the result that famine of a catastrophic scale is a familiar report for our media (Ethiopia 1983-84). Islands such as Singapore have not enough land area to support water storage and as a result they are forced to import their water from Malaysia. Underwater storage in the echinodome would greatly reduce the cost associated with this. Above ground tanks are visually unattractive, but are necessary in areas where reservoirs are not possible due to extensive flat plains and underwater storage is impractical in areas remote from coastal regions. Methods for covering reservoirs using floating balls have been tried, unsuccessfully. Underwater storage, as main supplies would remove the problem of evaporation of water from wells, boreholes, and would provide back-up supplies, e.g. for parts of Southern England.

The use of the echinodome as a storage structure is an obvious application requiring the need for services. However, there is the possibility of a practical structure requiring means of entry and exit for personnel as well as services.

Further experimental work is to be carried out on the echinodome to examine its collapse mechanism under all types of loading and the influence of earthquakes, explosions, tidal waves, etc. Theoretical studies have recently been carried out to compare the dynamic and static buckling behaviour of such a structure under excited concentrated loads applied axisymmetrically and symmetrically [Eldeeb K. M. M., 1993] (see Section 7).

2.5 Relevance of this Project

In coalition with the experimental and theoretical studies being done on the echinodome, it is necessary to examine what influences the dimensions of the structure. A program has been written to calculate a predicted shape taking account of the relevant factors concerned [Royles, 1980]. This program will now be examined in detail, studying the algorithms involved and the problems associated with the serial version, proving parallelisation to be a suitable solution.

3 The Shape Prediction Program

The shape of the echinodome is calculated using a serial program written in FORTRAN 77. This program originates from a comparison of numerical and graphical solutions for the shape corresponding to a particular pressure head, which both gave similar results [Royles, 1980]. From this program, from hereon referred to as `shapepp`, the coordinates of the meridional shape and of the inner and outer surfaces of the tank are obtained along with some geometric properties, namely, total surface area, cross sectional area and shell capacity.

The shape function depends on the design head above the shell apex, z_0, and the thickness, t, and allowable stress, σ, or ds, of the material in question. The product of $\sigma \times t$ indicates the strength of the material.

3.1 Equations of Equilibrium

The basic equations of the solutions to the drop shape are derived from analysis of the equilibrium of an element of a loaded shell. Their derivation is given in appendix A.

The basic shape function equations are,

$$\frac{du}{dx} + \frac{u}{x} = \frac{z}{K} \tag{1}$$

and

$$\frac{dz}{dx} = \frac{u}{\sqrt{1-u^2}} \tag{2}$$

The solution to these equations is carried out in three stages, using explicitly improved Euler in the original code, however the algorithm was changed to fourth order Runge Kutta for the purposes of this project and will probably remain so (see Section 3.2)

The initial stage is to integrate eqns. 1 and 2 over dx until $u \geq \sin^{-1} 45\circ$. To further increase in dx over dz would mean the curve would not return to the z-axis hence, for the second stage, the variable u is replaced by $w = \sqrt{1-u^2}$, so that,

$$\frac{dw}{dz} = \frac{\sqrt{1-w^2}}{x} - \frac{z}{K} \tag{3}$$

$$\frac{dx}{dz} = \frac{w}{\sqrt{1-w^2}} \tag{4}$$

Equations 3 and 4 are integrated over dz until $w \leq \cos^{-1} 135\circ$. The final stage reverts back to eqns. 1 and 2 until $\frac{dz}{dx} = 0$, thus ensuring that the curve returns to the z-axis.

Figure 7 shows a sketch of the shape predicted `shapepp`, A, B, and C representing each stage respectively.

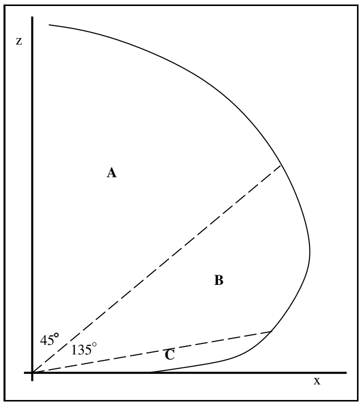


Figure 7: Sketch of Shape Predicted by Serial Code

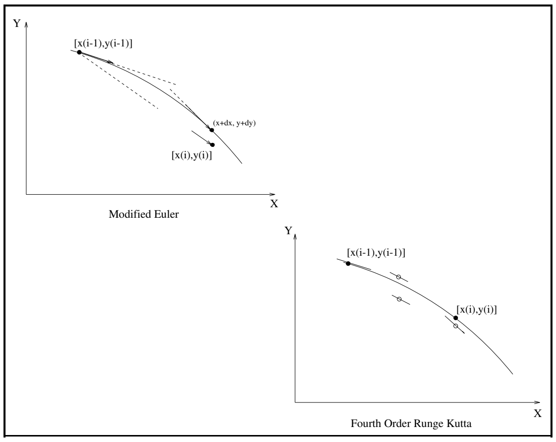


Figure 8: Algorithms used to solve Equations

3.2 Fourth Order Runge Kutta vs. Explicitly Improved Euler

The code written by the Civil Engineering department solved the equations using explicitly improved Euler. This was converted to fourth order Runge Kutta. Fourth order Runge Kutta holds a number of advantages over explicitly improved Euler. From figure 8, it is thought that the Runge Kutta algorithm is more accurate, in calculating the position of the next point on the curve than the Euler method, since it is dependent on the gradients of three intermediate points as opposed to the average of the gradients of the current point and the previous point, as is the case with Euler. Also it is certainly more robust than Euler, i.e. it is insensitive to small departures from the idealized position. The result was found to be the same, however, showing that, in this case, Euler provided sufficient robustness.

The proof of both the explicitly improved Euler and fourth order Runge Kutta algorithms is outside the confines of this report. Should the reader wish to know more details about these algorithms, they are referred to [Press, 1992].

3.3 The Sequential Code

The original code, called `shapepp`, consists of three main subroutines, within the main program, namely, `rk4`, `coor` and `iocoor`. To distinguish between the main program and these subroutines, they have been separated into four files – `main.F`, `rk4.F`, `coor.F` and `iocoor.F` – each of which are compiled and linked using a make file (see Section B). These files are described below in more detail.

Parameter	Range of Data	Increment
ds	20-40 MN/m^2	5MN/m^2
z_0	15-200 m	15 m
t	100-300 mm	25 mm

Table 1: Range of primary input parameters

3.3.1 The Main Program – `main.F`

This part of the code has been kept as close to the original as possible, with some minor adjustments accounted for by the changing of the algorithm, the isolation of the subroutines to separate files, and the adaptations of parallelism. The purpose of `main.F` is to read in the initial parameters from `input.spp`, to set the boundary conditions for the variables x, z, u and w, and to send those values to `rk4.F`, for the solution of the differential equations, and to `coor.F` and `iocoor.F` for the printout of coordinates. It also controls the increment of x, z, u and w. Its other main function is to calculate the geometric properties of the shell attributed to the initial parameters.

3.3.2 The Maths – `rk4.F`

This file contains the fourth order Runge Kutta algorithm, used to solve the differential equations described in section 3.1, and the subroutines `DERIVS1`, `DERIVS2` and `DERIVS3` which contain the equations. Each value of u and z, or w and x depending on the stage of the shape (see Figure 7), pass through this subroutine and the solutions are returned to `main,F`.

3.3.3 The Coordinate Printouts – `coor.F` and `iocoor.F`

`coor.F` calculates the coordinates (x and z) of the inner and outer surfaces (in millimetres) taking account of a varying thickness of the shell skin down the meridian. Currently these are printed out to the screen on request from the user (Section B). The subroutine `THKNSS` controls the points at which the thickness changes, depending on the value of z. This can be altered if so desired. This aids in the subsequent loading analysis (see Section 7).

Should the coordinates of the nodes – the points at which the thickness changes – be required, `iocoor.F` will print these out to `fort.7` and `fort.8`.

The output from the serial code provides the required geometric data in the form of a table. It gives an overview of the dimensions of the predicted shape and the initial parameters associated with the shape.

4 Limitations of Sequential Code

The parameters for the sequential code are read from a file, `input.spp`, the format of which is given in the user guide (Section B). The limitations of the sequential code emphasize the need for automated input in that, the program can only calculate the predicted shape for one value of design stress and thickness of a material, and for one value of design head, through one execution — which takes approximately twenty secs. on a PC. Table 1 shows the range of ds, t and z_0 for concrete. A serial run, done manually, over these parameters would take approximately four hours.

To execute `shapepp` on these parameters all on one processor uses up a lot of processing time and memory. It would be advantageous to use a mutiple number of processors, which could replicate `shapepp` doing the same calculation on each processor, but with different data. This would reduce the load on any one processor, with the bonus of cutting down the time for performing the calculations. Also since the data is read in from a file and requires storage for future reference, separate processors could specifically do this task. This is an ideal situation for the task farm, a common feature of parallel computing.

5 Overview of the Parallelisation

A task farm is a construct of processors being used for the purpose of parallel computing. It consists of a source, a sink/collator and workers (Figure 9.

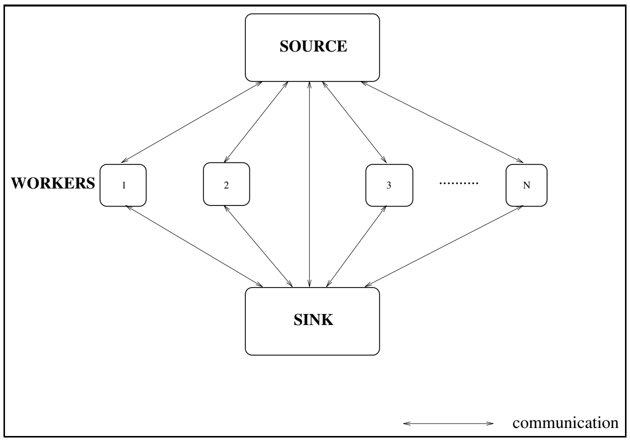


Figure 9: The Task Farm

5.1 The Use of MPI in the Task Farm

5.1.1 Initialising and Controlling the Environment

Communication between processors is carried out using Message Passing Interface (MPI), an international standard for message passing developed at the MPI Forum. MPI is a series of subroutine calls to an MPI library containing associated commands. For this project the MPI environment for the task farm is controlled by a setup file, `setup.F`. This file initiates MPI with `CALL MPI_INIT` and controls the assignment of processor(s) to the source, workers and sink. The source and sink are assigned to the first and last processors respectively and the workers are assigned the remaining processors (five in this case which can be extended if so desired). Intercommunication is then set up between source, workers and sink. Those processors used as workers are not required to communicate with each other (see Figure 9). The three subroutines — `source.F, worker.F` and `sink.F` — are then initiated depending on the active processor. `CALL MPI_FINALIZE` closes the MPI environment. The details of the communication are not relevant to this report, however readers interested in this are referred to [Forum, April 1994]. The task farm and each job of the source, workers and sink, is explained in more detail below.

5.1.2 The Source

For the purposes of the task farm it is more convenient to have access to the input parameters before the original code is called, hence `source.F` now reads in from `input.spp`as opposed to `main.F`. Data from `input.spp` is read in and stored in an array (`parameters`). The initial parameters required by `shapepp` and hence required by all workers (since each worker executes a replica of `shapepp`) are sent to each of the workers in turn. Upon receipt, the work process is initiated. A further two parameters are sent to the workers — the current values of σ, or ds, and z_0, which are looped over limits set by the user — until the respective maximum values are reached. Once completed, the source sends a “die” message to each of the workers to signify the end of the work. This message is the value -1 assigned to the DS parameter. Once finished, this processor lies idle.

5.1.3 The Workers

The worker subroutine implements `shapepp` into the parallel program. Having received the initial parameters and the ds and z_0 values, it firstly checks the value of DS. If $ds = -1$, then it passes the "die" message to the sink and exits the subroutine. Otherwise it passes these parameters to `shapepp` enabling it to execute the serial code over the ds and z_0 range. Once completing each run,`shapepp` returns the results i.e., the geometric properties, to a sink buffer array (`sinkbuf` and they are passed to the sink for storage. The workers are also used to simultaneously write out the x and z coordinates i.e., the predicted shape, associated with the current value of ds and z_0 which it holds. This gives the sink less work to do and avoids placing too much load on the sink all at once, allowing the sink to carry out its fucntion without being inundated with results. The format of the filename for the predicted shape corresponds to the value of ds and z_0. A sample from the code is shown below.

```
*****************************************
* Writing out shapes to associated file *
*****************************************

      character*12 shape
      WRITE(shape,40)DS,ZO/1000
40    FORMAT('ds',I2.2,'zo',I3.3,'.shp')
```

Once completed, these processors lie idle.

5.1.4 The Sink

The sink processor, apart from receiving the ds and z_0 boundary limits, does no work until it receives results from the worker. By using the ds and z_0 limits, the results are stored in a three dimensional array (`results(i,ZOp,DSp)`), where i is a geometric property, and ZOp and DSp correspond to the number of times the loop has been executed. This allows the results to be written out in a sensible, orderly manner. The results form the basis for the design curves. The files are in the form of data sets of the corresponding values of z_0 and the geometric property. These design curve files are appropriately named for ease of identification.

6 Results

6.1 Presentation and Comparison

For a particular material and thickness design curves can be drawn aiding the designer in his selection of an echinodome. These curves take the form of

1. volume v operating depth
2. volume v design head
3. maximum diameter v design head
4. shell height/maximum diameter v design head, and
5. volume of shell material v design head

Figures 10- 14 present these curves from the parallel code and their use is explained in section 6.2. The geometric results from the serial code and the parallel code are compared in table 2 for concrete of thickness 150mm.

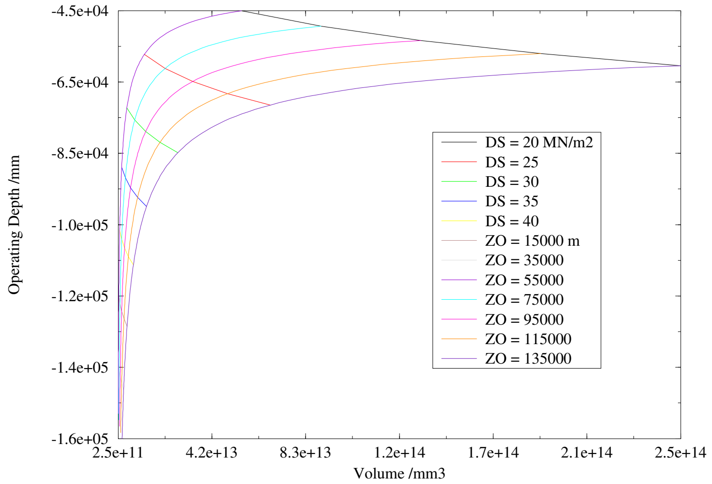


Figure 10: Operating Depth v Volume

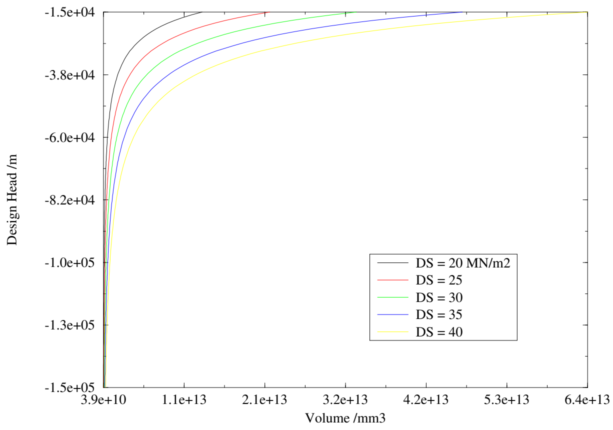


Figure 11: Design Head v Volume

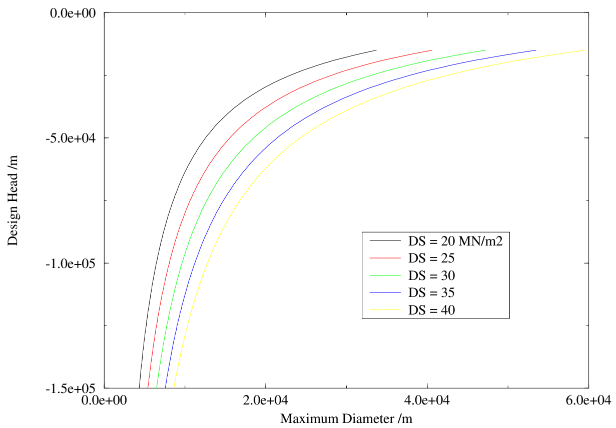


Figure 12: Design Head v Maximum Diameter

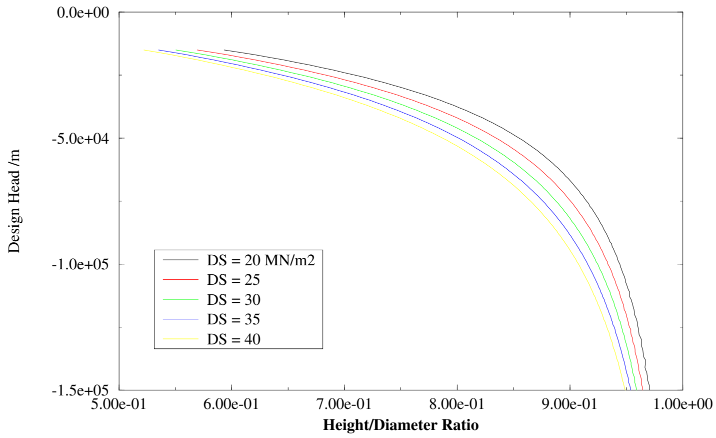


Figure 13: Design Head v Height to Diameter Ratio

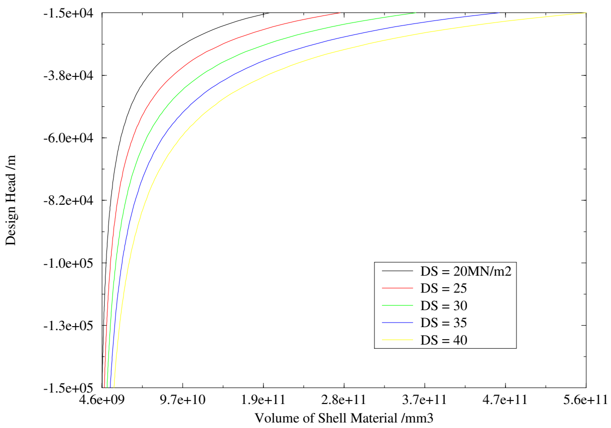


Figure 14: Design Head v Volume of Shell Material

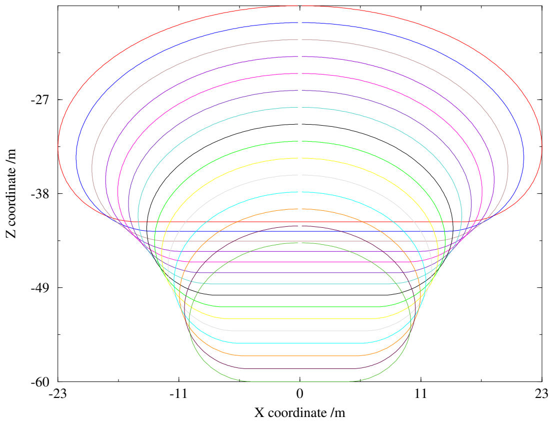


Figure 15: Family of Shapes for $ds = 30\text{MN/m}^2$

Geometric Property	ds /MN/m^2	z_0 /m	Serial Code	Parallel Code
Volume (m^3)	25	50	9530	9530
		100	1582	1582
Diameter (m)	40	50	43.4	43.4
		100	23.8	23.8

Table 2: Comparison of Serial Code to Parallel Code

6.2 Results Analysis

As expected, the comparison of the geometric results from the serial and parallel code shows that there is no variation, ignoring rounding errors.

The design curves were also as expected and a brief discussion on their usage will now follow.

The initial parameter the designer needs to know from the client is the volume of tank required. It is up to the designer to choose the material, and it is one of his aims to achieve the minimum thickness possible. From figure 10, a value for z_0 is obtained. If cost is the deciding factor for the client, the designer could choose to estimate it from the volume of material required, using figure 14, knowing the cost of the material. Should this be too expensive, the designer could input a smaller thickness and repeat the process using the same volume. The other geometric parameters of interest can be obtained from the other curves using the value of z_0.

The code does not currently cater for all thicknesses, as it does for ds and z_0, but this can be altered if so desired. Ideally, the designer should have all the design curves with all possible thicknesses available in a design manual — since this would prove quicker than calculating it each time.

Other results of interest that have been made available through this project include a plot of a range of predicted shapes for a particular design stress of concrete and various z_0 (Figure 15). It should be noted from this plot the reduction in capacity of the echinodome as z_0 increases, yet the more spherical it becomes. This was also found to be the case with the family of shapes adopted by the sea urchins (section 2.2).

7 Further Developments

The data sets from `shapepp` concerned with the coordinates of the predicted shape, the coordinates of the inner and outer surfaces, and the coordinates of the nodes are of paramount importance for future developments on the echinodome. Now that these can be achieved for many values of ds and z_0 very rapidly, each shape must be analysed for design and construction. Part of this analysis will include a theoretical investigation into the dynamic behaviour of the echinodome under various loading conditions, in particular the buckling behaviour due to statics and dynamic loads.

Buckling occurs when a structure undergoes large deformations leading to collapse. The axisymmetric shell of the echinodome exhibits unstable postbuckling behaviour with the possibility of rapid unloading. This behaviour is very sensitive to geometrical imperfections such as out of roundness and wall thickness variations. The critical buckling load, i.e. the load at which deformations start to occur, is dependent on structural geometry, boundary conditions, type of loading and material characteristics (strength, thickness, etc.).

Due to the above factors involved, theoretical predictions of the critical buckling load are not straightforward, and experimental determinations would prove costly to perform. A finite element approach was adopted using a finite element software package on a VAX. During this analysis the displacements for each time step, in accordance with the nodal forces, at each node over the entire structure, were computed for several iterations, to achieve dynamic equilibrium under both static and dynamic loads [Eldeeb K. M. M., 1993]. To perform this analysis involves intense computation – a recent analysis took thirty six hours for one shell – being due to the thousands of equilibrium equations generated. It is planned to carry out this analysis using the Cray T3D supercomputer based at Edinburgh University, to cut down the time taken for each analysis. It should be pointed out that there are only ten nodes assumed for the shells produced from `shapepp` (see section 3.3.3). A more realistic analysis would involve one hundred nodes or more.

To date the analysis has only considered axisymmetric loading due to the design head. Other loadings that may theoretically occur on the structure for which the response needs to be analysed include asymmetric and symmetric concentrated loadings due to trawlers and falling objects, axisymmetric pressure loading, quasistatic loading, explosive laoding and seismic loading, i.e. earthquakes and fault ruptures.

8 Conclusions

This project has successfully implemented a piece of serial code, designed to predict the shape of the echinodome, onto parallel machines. The result of this was to enhance the ability of the code to produce numerous shapes for various design parameters, and in so doing, produce sufficient data to plot design curves to represent the geometric results in graphical form. In the process, the original algorithm used has been changed and the original format of output has been modified to produce the design curves.

In view of the future investigations that are planned for the development of the echinodome as an underwater structure, this parallelisation process of the shape prediction program will prove extremely beneficial. Further parallel operations will enable the incorporation of this program, with a few minor alterations, and allow a complete parallel analysis of the structure from scratch.

In general, parallel computing in the field of structural engineering would prove very favourable in view of the time spent on design of elements in structures. For example, concrete floors, composite beams, steel girders, etc. could all be analysed in one execution of a relevant parallel program, providing sufficient data for choosing the most optimum design.

9 Acknowledgements

Could I take this opportunity to thank:

Rob Baxter, of EPCC for his continuous support and endless assistance which he devoted to this project.

To Colin Brough, for his support and encouragement, and endurance of my numerous queries.

To Dr Royles, of the Department of Civil Engineering and Building Science, for his aiding and abetting of the civil engineering aspect of the project.

10 Appendices

A Derivation of Differential Equations

A.1 Notation

A list of the symbols used in this work is given below:

D	maximum diameter of shell
H	height of shell
K	constant, $\frac{N}{\gamma}$
N	stress resultant at design head
N_ϕ, N_θ	stress resultants in meridional and circumferential directions.
R	resultant of external forces above a parallel circle defined by ϕ
r_1, r_2	meridional and circumferential radii of curvature
r'	parallel circle radius of curvature
t	thickness of shell
x, z	cartesian coordinates
z_0	design head above shell apex
γ	unit weight of fluid
ϕ	angle defining position on meridian
θ	angle defining azimuth position

In order to obtain an approximate shape of the shell of constant strength, two basic differential equations can be derived from examining the membrane forces on an element of an axisymmetrically loaded shell (Figure 16). From this element, two general equations of equilibrium can be obtained,

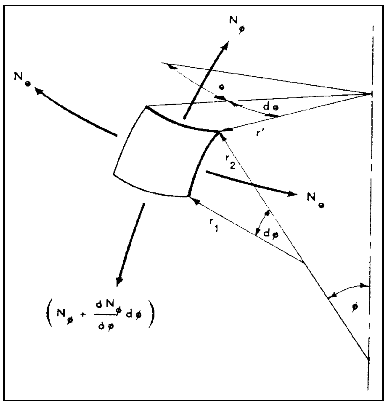


Figure 16: Forces acting on an axisymmetrically loaded Shell

[Royles, 1980]

$$2\pi r' N_\phi \sin\phi + R = 0 \tag{5}$$

$$\frac{N\phi}{r_1} + \frac{N\phi}{r_2} + z = 0 \tag{6}$$

Under a linearly varying external pressure $N_\phi = N_\theta = N$. Hence at some depth z, eqn. 6 gives

$$N(\frac{1}{r_1} + \frac{1}{r_2}) = \gamma z \tag{7}$$

where N is negative, i.e. compressive.

From Figure 16,

$$r_1 d\phi = ds = \frac{dx}{\cos\phi} \quad \text{and} \quad r_2 = \frac{x}{\sin\phi} \tag{8}$$

thus,

$$\frac{1}{r_1} = \frac{cos\phi d\phi}{dx} = \frac{d(sin\phi)}{dx} \tag{9}$$

Substituting into eqn. 7 gives

$$\frac{d(sin\phi)}{dx} + \frac{sin\phi}{x} = \frac{\gamma z}{N} \tag{10}$$

Letting $u = \sin\phi$ we get

$$\frac{du}{dx} + \frac{u}{x} = \frac{\gamma z}{N} = \frac{z}{K} \qquad \text{where} \quad K = \frac{N}{\gamma} \tag{11}$$

and since

$$\tan\phi = \frac{dz}{dx} \quad \Rightarrow \quad \frac{dz}{dx} = \frac{u}{(1-u^2)^{\frac{1}{2}}} \tag{12}$$

The coordinates of the apex are $x = x_0 = 0$, $z = z_0$, with $u = u_0 = 0$ rendering eqn. 11 indeterminate. Hence to overcome this, a small spherical cap is assumed of radius $\frac{2K}{z_0}$ and defined by a half chord Δx.

The coordinates of the apex then become, $x = x_1 = \Delta x$, $z = z_1 = z_0[1 + \frac{(\Delta x)^2}{4K}]$, and $u = u_1 = z_0\frac{\Delta x}{2K}$. These are used as the initial conditions for integrating eqns. 11 and 12.

The bounds of Δx and the associated errors are described in [Royles, 1980].

B User Guide for Shape Prediction Program

The following user guide describes the structure and format of all the parameters involved in the input, a brief description of the output files and their format, and an explanation of the execution of the parallel version of `shapepp`.

B.1 Input

The input parameters are stored in `input.spp`. This file has been commented to assist the user when inputting design data for the echinodome, however a desciption of the file and the parameters therein follows.

The file is set up as a list of data which has been placed in a specific order for the purposes of the program. The file comprises of the parameters listed below.

- T – thickness (mm) – initial thickness of shell skin starting from the apex of the tank and working down the meridian.

- DX – increment in x (mm) – controls increment of x coordinate.
- DZ – increment in z (mm) – controls increment of z coordinate.
- G – gamma (N/mm^2) – the unit weight of the fluid in which the tank is placed. Default is 9.81 MN/m^2 (unit weight of water)
- IND – an index allowing the user to control the printout of coordinates. If IND = 1 : coordinates are given for the centreline. If IND = 2 : coordinates are given for the inner and outer surfaces.
- DX1 – initial increment in x – Due to the behaviour of the equations of the shape, the initial value of dx cannot equal zero, hence a small spherical cap is left at the apex of the tank. This can be of use for a connection to the tank through the apex.
- DSMIN – the lower limit of the range of design strengths, ds, available for the material.
- DSMAX – the upper limit of the range of ds.
- DSINC – the ds increment over the range.
- ZOMIN – the lower limit of the range of design heads, z_0, above the apex.
- ZOMAX – the upper limit of the range of z_0.
- ZOINC – the z_0 increment over the range.
- ISEG – an index controlling the nodes down the meridian, i.e. the number of times the thickness changes. If no change in thickness is required, this value should be set to 1.
- THICKNESS – the value(s) of the new thickness at each node. This list should include the initial value given under T.

B.1.1 Points to Note concerning Input

1. The user is asked to adhere strictly to the above conditions laid down in the format of `input.spp`, both with the order of the parameters and the comments accompanying each and also to ensure that the filename remains as the original (since this is where `shapepp` obtains its parameters from).
2. Should a value other than 1 or 2 be input for the parameter IND, the program will only print the coordinates of the meridian.
3. For design purposes, a minimum value of 15 m is recommended for z_0. This the maximum proposed fifty year wave height, i.e. 30 m from trough to crest.
4. The angles at which the equations change over (see section 3.1 are fixed at 45° and 135°. Should the user wish to alter this, the parameters P and Q, within `main.F` should be amended accordingly, to the $\sin$ and $\cos$ of the desired angle in each case.
5. Should any such alterations be made to the actual code, as opposed to the input file, in order to execute the revised version, the user must recompile the code. This is done simply by typing `make` at the command prompt within the directory of `shapepp`.

B.2 Output

The output files are written to the current working directory. The design curves are named according to the results contained therein, for example, design head v volume is named `ZO_V.out`, etc. The predicted shape files are named according to the value of ds and z_0 for that shape, for example, for ds = 30 MN/m^2 and z_0 = 50 m the named file is of the form `ds30zo50.shp`.

These files contain XY data sets and should need no amendments for plotting purposes, unless the user wishes to do so, in which case they can be made using an editor. The curves and shapes can be visualised using any graph plotting package, for example, xvgr or xgraph (UNIX).

B.3 Executing `shapepp`

In order to execute `shapepp` successfully the user must first ensure the following,

- access to a number of processors/hosts either via a PC network or a cluster of workstations.
- an installed copy of an MPI library.
- a configuration file see below

Within the directory containing `shapepp` there is a file called `shapepp.cfg`. In order to run an MPI program a configuration file is required. This file is the configuration file for `shapepp`. It specifies which executables (or processes) the loader should load onto which processors (or hosts).

Should the reader wish to alter the structure of the configuration file, they are referred to their MPI manual.

To run `shapepp`, the user should first either remove or rename any existing output files from the program that may already be in the directory. Any files which the user does not rename and wishes to refer to at a later date will be overwritten by further executions of the program. At the command prompt, from the directory containing `shapepp`, type the following,

```
mpirun shapepp.cfg <n>
```

where `n` is the number of processors that are available to the user. An ideal number is six to eight.

Upon execution the user will find all the output files in the current working directory.

References

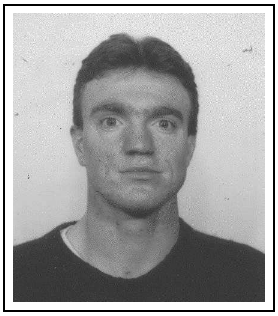
[Eldeeb K. M. M., 1993] Royles R. Eldeeb K. M. M. Dynamic and Static Buckling Behaviour of an Echinodome. *Computers and Structures Journal*, 1993.

[Forum, April 1994] MPI Forum. MPI: A Message Passing Interface Standard. *Manual*, April, 1994.

[Innes, 1994] Fraser Innes. Pollution Control Structures. *Honours Thesis*, 1994.

[Press, 1992] et. al. Press, William H. Numerical Recipes in FORTRAN: the art of scientific computing. *Book*, 1992.

[Royles, 1980] et. al. Royles. Strain. *BSSM Journal*, volume 16, 1980.



Richard Boyd is originally from Northern Ireland and is going into fourth year of a four year Civil Engineering degree at the University of Edinburgh. After graduation in 1995 he hopes to obtain chartered status and pursue a career in a computing apsect of Civil Engineering.

This project was supervised by Dr R. M. Baxter (EPCC) and Dr R. R. Royles (Department of Civil Engineering and Building Science, University of Edinburgh).