

Review of FELIPE version 3

Rob Beale
Oxford Brookes
University

rgbeale@brookes.ac.uk



FELIPE (Finite Element learning Package) is a software package designed to enable students of engineering and mathematics understand the finite element method. It consists of a suite of executable programs together with the Fortran 77 source code to solve a variety of finite element problems. FELIPE was developed by Dr Martin Reed of the Department of Biological Sciences, Brunel University. The price of the program is £69+VAT for a licence for a named user on a single pc. A department licence allowing up to 30 pcs in a single location costs £699+VAT. This package is one of the cheapest introductory systems for finite element analysis, providing source code as well as operational programs.

A demonstration program can be down-loaded from the website <http://www.brunel.ac.uk/~blstmbr/felipe>. The demonstration contains pre- and post-processors with a fixed lifespan together with three basic programs for two-dimensional elasticity and Poisson problems and a frame analysis. Online manuals are available at the website. Both the demonstration and the full program consist of self-executing zip-files which expand to occupy less than 4 Mbytes. The installation is straightforward and the files can be stored in a user-defined directory. The full package consists of:

- an interactive graphics pre-processor to enable the generation, refinement and renumbering of two-dimensional finite element meshes. The reviewer was pleased and surprised to see in amongst the standard element types a number of semi-infinite elements as these are often not included in basic packages.
- Nine executable programs to perform a range of mathematical analyses – Poisson's equation, plane stress/plane strain, beam/frame, elasto-plasticity, viscoplasticity, thermoelasticity and soil consolidation.
- A post-processor to enable stress contours, temperature contours, displacement plots to be produced.
- The Fortran 77 source code of the nine executable programs.
- Sample input and output files.
- A manual describing the operation of the program together with a summary of the theory used in developing the Fortran code.

Supplier's contact details

Dr Martin B Reed
Brunel University
Kingston Lane
Uxbridge, UB8 3PH

E: martin.reed@brunel.ac.uk
W: <http://www.felipe.co.uk/>

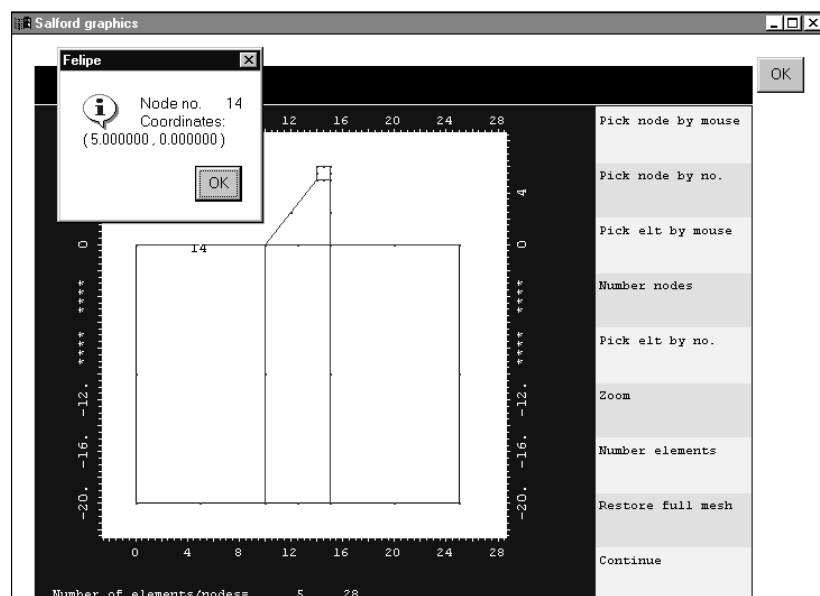


Figure 1 Initial mesh, with node 14 selected

The Pre-processor

To enable finite element models to be constructed it is essential to be able to see the mesh created. The pre-processor, PREFEL, used in the Felipe suite is easy to use, containing a mixture of text and graphics input. Figures 1 and 2 show the development and refinement of a simple mesh for an analysis of a dam created by using PREFEL. A major criticism the reviewer initially had with regard to the pre-processor was that due to its design as an instructional program the user is unable to undo an incorrect action. Following discussions with the program author the ability to save and restart models has now been introduced. The author, commendably, is very quick to take on board comments which enable the program to be improved. The program has the following excellent features:

- When a single element is subdivided using one of the mesh refinement utilities the subdivision is continued into adjacent elements so that inter-element nodal continuity is enforced. Boundary restraints are automatically included in the new elements.
- A single operation enables the whole mesh to be doubled so that students can investigate convergence of the finite element model.
- The ability to reorder element and node numbers starting from an element selected by the user using screen input.

This last facility enables the student to understand the influences of element and nodal numbering on the frontwidth and bandwidth of the resulting system of equations. At the same time as renumbering there is an option of displaying graphically the structure of the global stiffness matrix during the renumbering process. Figure 3 shows a combination of the global structure matrix during the renumbering system and also indicates the relationship between local and global element nodal numbering for element 21.

The pre-processor operates in two modes – basic and advanced. The basic mode restricts the elements to

eight noded quadrilaterals for elasticity problems and three noded triangles for Poisson problems. The program will enable four noded quadrilaterals to be used for ease of data generation for the latter case but these quadrilaterals must then be triangulated before the pre-processor produces a data set. The advanced mode enables all the different elements to be generated, including the infinite elements. The basic-level mode generates files which are used with the Poisson and two-dimensional elasticity programs provided with the demonstration version.

Having obtained a correct mesh with appropriate boundary conditions (such as nodes fixed in the x-direction) the final part of the pre-processor enables loading data (point loads, surface tractions, prescribed displacements and body forces for elasticity problems and Dirichlet or Neumann conditions for thermal analyses) to be applied.

As the program is designed for teaching purposes the maximum number of nodes is 3000 and the maximum number of elements is 900. The reviewer would have liked to

have seen higher limits as real problems drawn from engineering often require a couple of thousand elements for convergence studies to be properly undertaken. The author has stated that for a nominal charge he will provide versions of the programs which have extended element and nodal sizes. The program was written using the Salford Fortran compiler which uses proprietary extended memory management utilities. Early versions of the pre-processor crashed whenever a second program was run simultaneously. The reviewer was impressed with the response of the program author when this problem was reported. He immediately contacted Salford and revised run-time libraries were sent which corrected the fault.

Analysis programs

Having created a data file the user of the suite has available a set of nine standard executable programs to

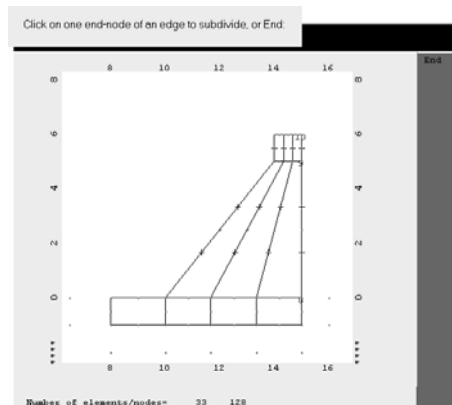


Figure 2 Mesh refinement in progress

Figure 3 Display of global stiffness matrix structure during assembly process



perform various analyses. The author's objective is to get students to understand the theory and implementation into useable programs of the finite element method. Therefore, as well as having the executable program, in the user manual of the suite, a brief overview of the relevant finite element theory is given. The author expects the student to have available a standard text book, such as *The Finite Element Method in Engineering Science* by O C Zienkiewicz (McGraw Hill) or *Concepts and Applications of Finite Element Analysis* by R D Cook et al (Wiley 2002) to enhance the brief descriptions. A full description is then given of the detailed structure of each program describing the Fortran subroutines used in its code.

Included with the suite is the Fortran 77 source code of each program. As it is relatively straightforward to convert from Fortran 77 to Fortran 90/95 the reviewer would have liked to have seen this source code in Fortran 90/95. The source code requires the user to link a second file containing library routines to perform input/output. Unfortunately the need for this library is not made clear in the documentation and the reviewer initially had trouble compiling the source. After the library was included, however, the source program was easy to compile using the Compaq Fortran compiler.

The reviewer found the programs to be extremely well documented. They are designed so that students can take the code and modify it for their own problems. The reviewer was impressed with the number of different algorithms coded into the system. Amongst the nine pre-supplied programs there are four different storage techniques combined with five solution algorithms! This obviously means that a student who is coding his or her own finite element program has a wide variety of different methods to use. The combined storage and solution techniques are:

- Symmetric banded storage in conjunction with

Choleski decomposition which is used in the programs for Poisson analyses, the standard elasticity program and an elasto-plastic program.

- Non-symmetric banded storage in conjunction with the standard Gauss elimination technique which is used in the coupled thermo-elastic analysis program.
 - A skyline analysis (a modified bandwidth approach where the position of the start of non-zero elements in each row is used to reduce storage requirements) with decomposition which is used in the two-dimensional consolidation analysis.
 - Frontal solution which is used in the full elastic analysis program containing all the available element types in

combination with all loading types. It is also used in the elasto-viscoplastic program.

- The frame analysis program uses the diagonal-preconditioned conjugate gradient algorithm.
- Finally, the extended elasto-plastic program contains several of the above solution and storage algorithms as well as an algorithm incorporating Incomplete Choleski preconditioning in combination with the conjugate gradient solution technique. This enables the student to investigate the effects different solution strategies have on the resulting finite element model.

The user manual describes the theory behind the techniques with reference to further reading if required.

The post-processor

The post-processor, FELVUE, takes the results of the analysis programs and produces graphic plots. It is capable of plotting contours or vectors of deformed meshes, stress contours of individual stress components, principal stresses and the stress average and stress deviator stress.

For thermal problems the post-processor can contour temperatures and contours of flow magnitudes or flowlines. Figures 4 and 5 show samples of displacement profiles and stress contours. The analysis programs output stress data at the Gauss points and when plotting, these stress values are interpolated to the boundaries of

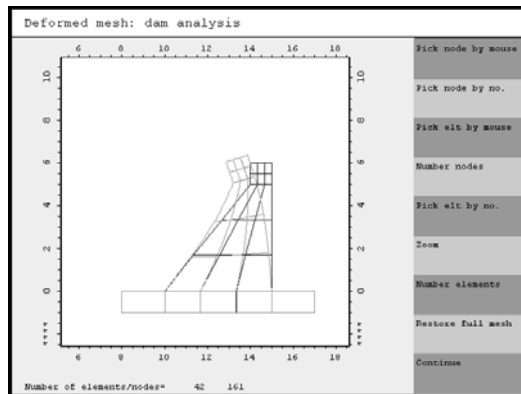


Figure 4 Deformed mesh (Uniform normal pressure on vertical face of dam)

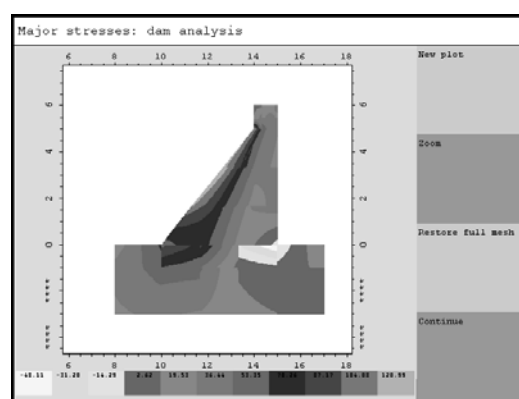


Figure 5 Contours of major principal stress (compression positive sign convention)

elements so that the discontinuities in the stress fields can be seen. The post-processor operates in the same way as the pre-processor and is very easy to use.

The reviewer is concerned that when students encounter finite element methods for the first time that emphasis is given to determining the accuracy of the model produced. This, on a first course in finite element theory, is often simply performed by investigating the numerical values of the inter-element nodal discontinuities in derived quantities such as temperature fluxes and stresses. Although plots can be obtained showing these discontinuities (as seen in Figure 5) it would have been useful to have the option, without having to write the specific Fortran code, of obtaining printouts of these quantities at element nodes.

Conclusions

The reviewer feels that for the reasons given above the package is currently not suitable for students undertaking

a first course in finite element analysis, particularly engineering students. The reviewer feels that commercial packages such as ANSYS or LUSAS, which although significantly more expensive than Felipe do have full modelling and analysis capabilities are more appropriate for first courses, particularly for Engineering students. However, for students who already have an overview of the finite element method and wish to get a deeper understanding of the underlying concepts of finite element theory and implementation Felipe is an excellent vehicle for developing that understanding. The reviewer feels that as currently presented the package is suitable for final year and post-graduate students who expect to have to program their own finite element algorithms. It should certainly be seriously considered for all courses which require finite element programming. Further development of the package to deal with some of the deficiencies above would also make the package suitable for initial finite element courses.

Supplier Comments from Martin Reed

I am very grateful to the Reviewer for his many positive comments about the FELIPE concept and package, which combines source-code FEM modules with powerful graphics pre- and post-processing at a highly affordable price. His valuable suggestions have helped to improve the package significantly.

There are two points on which I was reluctant to follow the Reviewer's recommendations fully. One concerns the maximum size of mesh handled by the pre- and post-processors. There is a danger that if a large amount of virtual memory is allocated in these Fortran programs, users with low-specification PC's may experience poor performance even while constructing small meshes. However, recognizing the different needs and resources of users, I am now offering the package in either Standard (maximum 900 elements), Large (2,500 elements) or Giant (9,999 elements) installations. As the source code is provided for all the main FEM programs, users can redimension these according to their needs, and re-compile.

There are two main reasons why I have not taken up the Reviewer's suggestion of outputting stress/flux data at nodes:

- Stresses evaluated away from the element Gauss points using the shape functions are inherently less accurate, and may be wildly wrong even with a good discretization (e.g. in nearly incompressible materials). They are best recovered by fitting a lower-degree surface through the Gauss-point values (which is the technique used in the contouring routines);
- As well as being an instructional package, FELIPE has proved popular with practising engineers as a cheap 2D

'black box' analysis tool. These users would not appreciate being provided with inaccurate and confusing stress output.

Moreover, the FELIPE concept means that an instructor requiring this output can easily program its evaluation into the elasticity source code (or, better still, set it as a student exercise!) for printing in the preferred format, element-by-element or node-by-node, and re-compile.

I hope that this does adequately address the two residual concerns of the Reviewer, and would claim that FELIPE is a valuable learning aid for students meeting finite elements for the first time. There are some unique features, such as the interactive demonstration of the element assembly process, which are particularly helpful for beginners.

I am very keen to hear from instructors interested in using FELIPE in their courses, after trying out the basic package from the download at the FELIPE website (also reachable at www.felipe.co.uk). A FELIPE-based FEM textbook would be a logical development, and publishers are welcome to contact me with proposals!

I would like to acknowledge here the helpful comments of Dr John Appleby, who worked on a draft review of FELIPE, but was unable to complete it because of other commitments.