Int. J. Appl. Engng Ed. Vol. 7, No. 2, pp. 140–150, 1991
Printed in Great Britain.

0742–0269/91 $3.00+0.00

FESTAL, a Finite Element Scheme for Thermal Analysis*

K. JAMBUNATHAN
F. SHEMIRANI
B. L. BUTTON
Mechanical Engineering Department, Faculty of Engineering, Nottingham Polytechnic, Nottingham, NG1 4BU U.K.

A finite element scheme is described for the analysis of conjugate fluid flow and heat transfer based on the Galerkin formulation. The general structure of the algorithm is presented and its capabilities are highlighted. Results have been predicted for laminar flow over a backward facing step and laminar natural convection in a square cavity and these compare well against those obtained from a commercially available finite volume package and benchmark results. The present scheme has been designed to be user-friendly with minimal data input so that it can be used effectively to introduce and demonstrate the finite element technique to students as well as others with limited knowledge of the method.

Nomenclature

c_p	specific heat capacity
D	width of square cavity
F_x, F_y	body forces in x and y directions
g_x, g_y	components of gravity in x and y directions
Gr	Grashof number
H	step height
k	thermal conductivity
l_x	x direction cosine
l_y	y direction cosine
M_i	shape function for linear elements
N_i	shape function for quadrilateral elements
p	pressure
Pr	Prandtl number, ν/α
Re	Reynolds number, $U_{avrg}H/\nu$
Ra	Rayleigh number, $g_x\beta D^3\Delta T/\nu\alpha$
T	temperature
T_c, T_h	cold and hot wall temperatures
T_∞	fluid temperature at infinity
u	x-component of velocity
U_{avrg}	average inlet velocity
v	y-component of velocity
x	x-coordinate
y	y-coordinate.

Greek symbols

α	thermal diffusivity
β	volumetric thermal expansion coefficient
ΔT	temperature difference, $T - T_\infty$
η, ξ	local coordinates for quadrilateral regions
η_i, ξ_i	nodal values of local coordinates
μ	dynamic viscosity
ν	kinematic viscosity, μ/ρ
ρ	mass density
φ	general variable.

* Paper accepted 18 September 1990.

INTRODUCTION

FESTAL is a finite element package for solving the differential equations which describe the transfer of momentum and energy in a fluid flow environment. The set of equations considered are the Navier-Stokes and the energy equations limited to the two-dimensional Cartesian coordinate and axisymmetric systems. The package is capable of tackling laminar flows with or without heat transfer, heat conduction in solids, and conjugate heat transfer-fluid flow problems where the domain of interest has both solid and fluid regions. The theoretical background on which FESTAL is based, is first explained and the general structure of the package is then illustrated. FESTAL is developed in such a way that any future improvements, such as code enhancement and subsequent alterations or modifications can be performed with ease. This is essential with any finite element coding, simply because the great majority of researchers are now dedicating their efforts to finding better techniques for modelling fluid flow problems using finite element methodology.

The capabilities of FESTAL including the domain geometry definition, types of problems and output of results are outlined. Only two types of elements are used to define the geometry of the domain, namely the biquadratic eight-noded shapes for defining regions and variations of velocity components and temperature and bilinear four-noded shapes for defining the pressure variations. The output of results from an analysis are available as both numerical values and pictorial representation either on screen display or hard copy so that these may be compared or assessed easily.

As mentioned previously the coding is under development. The current areas of the program that are receiving attention are listed and possible improvements that will be carried out are highlighted.

Numerical examples of laminar flows over a backward facing step and in a square cavity are provided. The results are compared with those obtained from a commercially available finite volume package in the first case and with previous benchmark results in the latter.

The paper concludes by assessing the current state of the program and pinpoints the areas where further work needs to be carried out.

THEORETICAL BACKGROUND

The governing partial differential equations for steady state two-dimensional laminar compressible flow used by FESTAL are [1]:

Continuity $\partial(\rho u)/\partial x+\partial(\rho v)/\partial y=0$ (1a)

X-momentum $\rho u\partial u/\partial x+\rho v\partial u/\partial y= -\partial p/\partial x+F_x+\mu(\partial^2 u/\partial x^2+\partial^2 u/\partial y^2)$ (1b)

Y-momentum $\rho u\partial v/\partial x+\rho v\partial v/\partial y= -\partial p/\partial y+F_y+\mu(\partial^2 v/\partial x^2+\partial^2 v/\partial y^2)$ (1c)

Energy $\rho u c_\rho \partial T/\partial x+\rho v c_\rho \partial T/\partial y= k(\partial^2 T/\partial x^2+\partial^2 T/\partial y^2)$ (1d)

In addition, for flows with varying density, the equation of state is

$\rho=f(p,T)$ (1e)

which relates the density to pressure and temperature. For axisymmetric analysis appropriate terms are added automatically by the program.

The Galerkin weighted residual approach [2] is used to reduce the above non-linear simultaneous equations into the assembled matrix form:

$[A]\{\lambda\}=\{F\}+\{B\}$ (2)

where A is the global coefficient matrix, F is the global force vector, B is the global boundary condition vector and λ is the variables (unknown) vector. The above set is solved by an iterative process to yield components of velocity, pressure and temperature. A convergence sequence and a method of variable updating is employed. The algorithm contains seven major steps:

1—User input data is read.
2—Computational domain is generated.
3—Boundary conditions are set.
4—Initial values of the primitive variables are assumed.
5—Updated values are solved for.
6—Changes in all variables are calculated throughout the domain. If these fall within the specified tolerance, the computation is complete.
7—If the differences evaluated in step 6 do not fall below the required tolerance, all variables are updated. The steps 5 to 7 are repeated until the required tolerance is satisfied at all points within the domain.

The domain of interest is divided into eight-noded isoparametric two-dimensional quadrilateral regions (Fig. 1). The shape function for these regions is [3]:

For corner nodes,

$N=0.25\,(1+\xi_i\xi)(1+\eta_i\eta)(\xi_i\xi+\eta_i\eta-1),\quad i=1,3,5,7$ (3a)

and for the midside nodes,

$N_i=0.5\,(1-\xi^2)(1+\eta_i\eta),\ \xi_i=0,\ i=4,8$ (3b)

$N_i=0.5\,(1+\xi_i\xi)(1-\eta^2),\ \eta_i=0,\ i=2,6$ (3c)

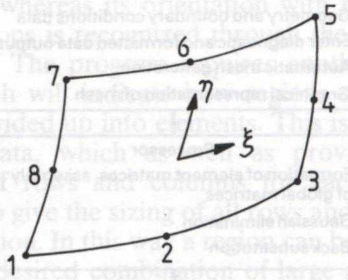


Fig. 1. Isoparametric biquadratic eight-noded element for velocity and temperature.

Each region is further divided into eight-noded isoparametric elements with the same shape function as above. The above shape function is also used for u, v, T. Hence

$$\phi=\sum_{i=1}^{8} N_i\phi_i \quad (4)$$

where ϕ_i is the nodal value of velocity components or temperature.

The four corner nodes of each eight-noded element is used to construct the linear element for p to avoid spurious pressures generated [4] (Fig. 2). The shape function for this bilinear element is

$M_i=0.25\,(1+\xi_i\xi)(1+\eta_i\eta),\ i=1,2,3,4$ (5)

$$p=\sum_{i=1}^{4} M_i p_i \quad (6)$$

where p_i is the nodal value of pressure.

STRUCTURE OF FESTAL

Figure 3 shows the general structure of FESTAL. It consists of three main parts:

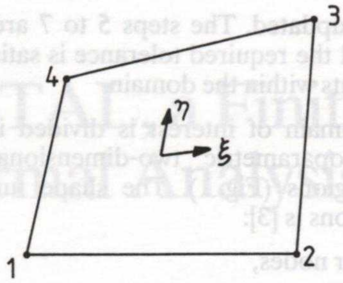


Fig. 2. Isoparametric bilinear four-noded element for pressure.

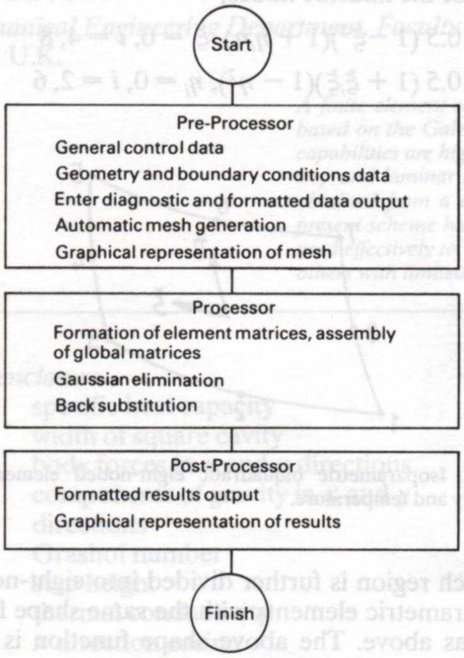


Fig. 3. Schematic structure of FESTAL.

Pre-processor

At this level the necessary data regarding the geometry and the imposed boundary conditions are read from an already prepared data file. The data is then sectioned, formatted and written into the output file. The data is checked and if any errors are detected, relevant error messages are initiated and the program is then terminated. With the correct input data the domain is automatically divided into quadrilateral regions. The generated mesh can then be inspected visually and if satisfactory the analysis may be continued. The pre-processor contains the following parts:

(1) General control data
All the parameters controlling the next stage of the data input phase are first read. These provide the information regarding the type of flow, total number and tolerance for iterations, type of boundary conditions and dimensionless parameters (Re, Gr, Pr, l_x, l_y).

(2) Geometry and boundary conditions data
The domain of interest is defined in terms of quadrilateral regions and the size of each region. The location of each region is defined through its eight boundary points and their x and y coordinates. Information regarding regions' connectivity and number of rows and columns are then read. Prescription of boundary conditions is performed through integer identifiers. These specify the type of imposed condition.

(3) Error diagnostic and formatted data output
Once all the data is read, it is checked for correctness. If any error is encountered, the relevant error message is sent to both the terminal and the output file. Any error would result in the termination of the program. As the data is checked it is formatted, sectioned and written to the output file for later inspection by the user.

(4) Automatic mesh generation
The program uses the information provided in the last section to divide each region into the desired number of rows and columns. The global nodal and elemental numbering is then performed. Once this task is completed, the information on the geometry based on regions provided by the user becomes irrelevant and the program recognizes the shape of the domain only through the nodal and elemental information, which has been generated automatically in this section.

(5) Graphical representation of mesh
The generated mesh can be inspected for either accidental or logical errors, which could not be detected in section (3). Here the connectivity data for regions and elements as well as nodal and elemental numbering can be interrogated. At this stage the program may be terminated if any errors are detected. A hard copy of the mesh can be obtained.

Processor

At this level the global coefficient matrix, A, global force vector, F, and global boundary condition vector, B, are assembled (cf. equation (2)). The iterative solution procedure is based on the direct elimination frontal solution method [5]. However, the frontal solution is modified slightly to allow for the unsymmetric nature of the global matrices [6]. The results of each iteration are written to the output file. The calculations end if the solution converges to the required tolerance or if the specified iteration number is exceeded. This part of the program consists of the following sections:

(1) Formation of elemental matrices, assembly of global matrices
After the generation of elements, each element is visited in turn and the coefficient matrix, force vector are set up and the boundary conditions incorporated. These are then placed in the appropriate places within the global matrices. This process is repeated until the contributions of all elements to a particular node are considered. Therefore, the global matrices are never completely formed.

(2) Gaussian elimination
Once a pre-assigned global core area is filled from contributions from adjacent elements, the direct Gaussian elimination process begins. The largest diagonal entry in the core is found and used as a pivot to eliminate the maximum pre-determined number of equations. The reduced equations are then written onto disk and Section (1) is repeated until all elements have been included.

(3) Back substitution
Upon completion of the last part, unknown variables are determined. The elimination process will have terminated with one coefficient and a corresponding term on the right hand side of the equations. The variable associated with this coefficient is thus evaluated. Each back substitution introduces an equation with one or more unknown variable and all values are found by simple arithmetic. The end product is an array of variable magnitudes in ascending order of node numbers.

(4) Tolerance check and variable updating
After all the variables have been evaluated a tolerance check is initiated. The tolerance limit is set by the user. If the differences between the recently evaluated values and the previous values of the variables throughout the domain fall below the tolerance limit, the calculation procedure is stopped. However if this condition is not satisfied the variables are updated and calculations recommence. Updating takes the simple form of finding the arithmetic mean for a variable by considering the two above mentioned values. The accuracy of the solution is dependent on the tolerance set for the termination of the iterative solution procedure.

Post-processor

Once a solution set is arrived at, the results are available as numerical values or for visual inspection. This level of the program consists of:

(1) Formatted results output
The results are in the form of nodal values of velocity components, pressure and temperature. After each iteration these are written onto disk for subsequent inspection. Also at the final iteraton, the results are sent to a separate file, which may be used as starting values for further future use of the program.

(2) Pictorial representation of results
In this part the results can be interrogated visually in order to assess their credibility. The mesh may be drawn with velocity vectors, isobars and isotherms superimposed.

CAPABILITIES OF FESTAL

FESAL allows modelling of complicated geometries with relative ease. The prescription of boundary conditions forms an essential aspect of any computer modelling. This aspect is dealt with comprehensively so that the user can readily impose any type of desired boundary conditions. The interactive graphical facility both before and after the computations enables the user to inspect the mesh and boundary conditions and perform the necessary alterations.

The detailed description of the capabilities of FESTAL are as follows:

Domain geometry and division

The domain is divided into quadrilateral eight-noded regions (Fig. 1). Any complex geometry, with straight or curved boundaries or a combination, can be defined quite readily. The program would require the coordinates of all boundary points making up the domain as well as the topology of regions and connectivity data. The shape of each region is recognized through its topology, whereas its orientation with respect to other regions is recognized through the connectivity data. The program requires another set of data, which will indicate how regions should be further divided up into elements. This is the row/column data, which as well as providing the number of rows and columns for each region, would also give the sizing of all rows and columns for the region. In this way a region can be made to have any desired combination of large and small sized elements. The division of the domain of interest into quadrilateral regions enables the user to accurately define any complex two-dimensional shape with a minimum of data input.

Once the above sets of data are read the automatic mesh generation routine commences. Each region is further sub-divided into quadrilateral elements. This part of the program ends after generating global node numbers and element numbers.

Prescription of boundary conditions

In order to yield a unique solution, boundary conditions for all variables must be specified. These conditions prescribe values for velocity components, pressure and temperature. FESTAL enables the user to specify boundary conditions in five different ways:

(1) Throughout the domain
In cases where the energy equation is separated from the rest of the equation set (cf. equations (1a)–(1d)), either the flow field or the temperature field need to be solved. The former embraces a whole range of isothermal problems where the temperature remains unchanged, whereas the latter describes a non-flow situation (e.g. pure conduction). Therefore, any variable can be set to have a fixed value throughout the domain of interest. Here the type of the variable and its value must be provided.

(2) Within a region
Any variable can be assigned to have a fixed value within any region of the domain. This situation can for example arise where a solid region is in contact

with a fluid region. Then within the solid region, velocity components and pressure would be set to zero. This option requires the region number, type of the variable and its value.

(3) Along a side of a region
This is the most useful type of boundary condition, whereby any variable may be assigned a constant value along a side for the region. Isothermal surfaces and no-slip conditions can be modelled in this way. Then the side number of the region together with the type of the variable and its value must be supplied.

(4) At a point
This option can be used in places where the nodal value of a variable is to be fixed. This situation may arise as a result of trying to define a section of the domain where a known variation of a variable exists, such as a parabolic inlet velocity profile. The nodal number, type of the variable and its value may be stated.

(5) Gradient type boundary conditions
This type of boundary conditions are encountered usually on axes of symmetry or where there would not be any further changes in a particular variable with respect to a spatial coordinate. In this case the element number and its side number together with the type of the gradiant boundary condition and its value must be provided.

Types of problems

The classes of problems that can be tackled by FESTAL are as follows:

(1) Conduction in solids
FESTAL may be used to analyse heat conduction problems in solids. This is possible by defining the regions to be of solid type and providing the thermal conductivities of the regions together with any heat generation that may exist within them. The analysis would yield the governing temperature field in the domain of interest by providing nodal values of temperature. Here only a modified version of the energy equation (1d), i.e. for heat conduction in solids, is solved for. This facility is similar to FIESTA [7], and is included within FESTAL.

(2) Laminar flow
FESTAL can be used to analyse laminar flow fields. It would provide the two components of velocity as well as pressure field. In this case the energy equation is ignored altogether and only equations (1a) to (1c) are considered.

(3) Conjugate solid/fluid situations
Here all the equations (1a) to (1d) are considered simultaneously.

(4) 2-dimensional, axisymmetric problems
Through an integer identifier in the 'control data' section (Fig. 3), the program can switch between two-dimensional or axisymmetric analysis. The use of the axisymmetric option requires some modifications to equations (1a) to (1d), which is performed automatically.

(5) Irrotational flow
Again through an integer identifier at the 'control data' section FESTAL can switch to irrotational fluid analysis, where it would yield nodal values of velocity potential.

Error diagnostics

As data is read from an already prepared input data file it is checked for correctness and accuracy by the program. Error messages are initiated and directed to both the terminal and the output file, if any part of the input file contains erroneous data. This check is carried out for both geometry and boundary conditions data. Initiation of any error message would cause the termination of the program.

Graphical outputs

At the pre-processor level the geometry data may be inspected visually. Incorrect geometry data input would result in program failure. Hence by interactively checking the created mesh, the program can be terminated at this early stage. The geometrical output of FESTAL may contain:

(1) Generated grid,
(2) region boundaries,
(3) domain boundaries, and
(4) region, element and node numbers.

Graphical outputs may also be obtained at the post-processor level. The additional information that can be requested are:

(1) lines of constant property, for pressure and temperature, and
(2) velocity vector plot, where the option of zooming onto a particular area is available.

The graphical outputs can both be directed to the terminal or hard copying facilities.

FUTURE DEVELOPMENTS

FESTAL is currently under development in order to make it more efficient in problem specification, solution algorithm and in post-processing and representation of results. The areas which are being looked at are in the:

Pre-processor
The correct specification of a problem is the first and most important step towards the successful analysis of that problem. All the necessary information regarding the geometry and boundary conditions must be provided by the user, if the analysis is to yield a unique solution set. At present the data is read from an external file which has already been prepared by the user. It is desired to develop the pre-processor so that the necessary data can be requested by FESTAL in an interactive manner.

The data can then be checked and error messages initiated if incorrect data is input.

Processor

The solution algorithm is based on the Galerkin weighted residual method. This requires the use of mixed-order velocity and pressure elements so as to avoid spurious pressure modes. This to some extent requires a large computer storage and long solution times. Other recently developed techniques where equal-order pressure and velocity elements are used [8–14], seem promising. The computer storage requirement and run-time are compatible with those of finite difference schemes. Inclusion of the equal-order method in the processor will make FESTAL more efficient and applicable to a wider class of problems.

Post-processor

Results obtained at this level are in terms of nodal values of velocity components, pressure and temperature. However, other parameters of interest such as stream function and vorticity may be evaluated once the velocity field is determined. This requires the inclusion of additional subroutines in the post-processor to extract the above mentioned parameters. Also, further development for graphical representation of results is needed.

EXAMPLES

Backward facing step

The first example is that of isothermal laminar flow over a backward facing step. In such a situation flow separation occupies a large portion of the flow domain. Indeed separation has been shown to occur at very low values of the Reynolds number. Here the advection terms in the momentum equations are dominant and for high Reynolds numbers special techniques are required to ensure that the solution does not oscillate. These techniques were not necessary for the current low Reynolds number case. The extent of the domain together with the boundary conditions are shown in Fig. 4. The fluid is air with its physical constants taken at 300 K. A parabolic velocity profile was assumed at the inlet.

The problem was analysed both by FESTAL and PHOENICS (Parabolic, Hyperbolic, and Elliptic Numerical Integration Code Series [15]).

The computational grid for both cases are shown in Fig. 5. The mesh was chosen so that the results were grid independent. FESTAL used 49 elements and the grid for PHOENICS consisted of 30 × 40

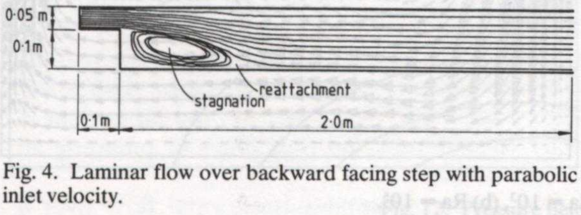


Fig. 4. Laminar flow over backward facing step with parabolic inlet velocity.

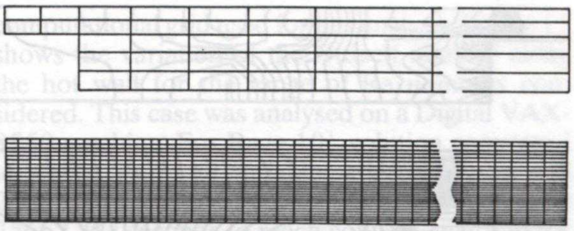


Fig. 5. Computational grid used by (a) FESTAL—49 elements, (b) PHOENICS—30 × 45 cells.

cells. The Reynolds number, based on the average velocity at the inlet and the step height, is 100. The computations were terminated when the difference between two successive iterations fell below 5% throughout the domain.

Figure 6 shows the vector plot of velocity. As can be seen the predicted flow patterns are similar and they both show that the stagnation point is about 1.5 × step height downstream of the step. Also the reattachment point is 3 × step height downstream of the step. Closer inspection of the numerical values in the recirculation region reveals that FESTAL slightly overestimates the velocity magnitude there.

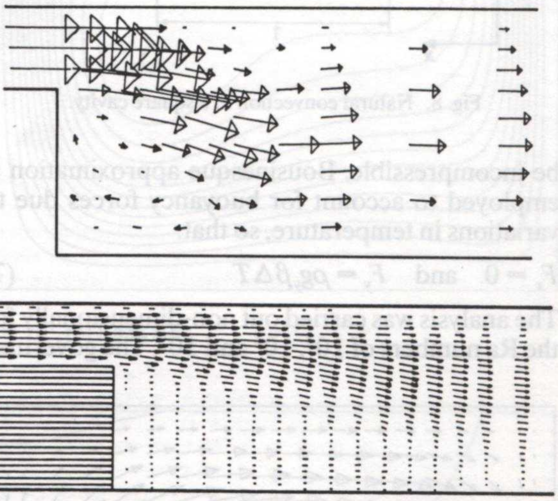


Fig. 6. Velocity field up to the reattachment zone, (a) FESTAL, (b) PHOENICS.

The pressure fields are different as shown in Fig. 7. FESTAL predicts a more volatile fluctuation in pressure field near the stagnation point. This analysis was performed on a Digital VAX-8550 machine. PHOENICS took 156 cycles and 1864 CPU seconds and FESTAL took 26 cycles and 1065 CPU seconds.

Square cavity

The second example is that of a laminar natural convection in a square cavity as shown in Fig. 8. This problem was analysed for a range of Rayleigh numbers and the results are compared with the benchmark exercise [16]. The flow is considered to

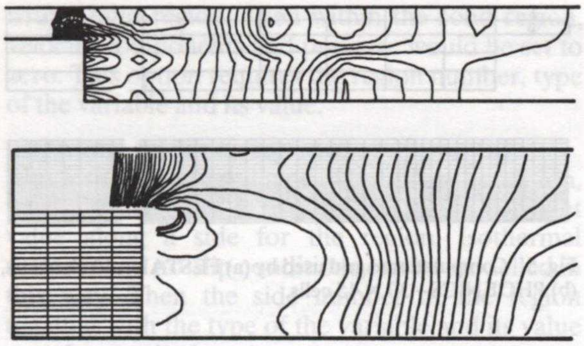


Fig. 7. Pressure field up to the reattachment zone, (a) FESTAL, (b) PHOENICS.

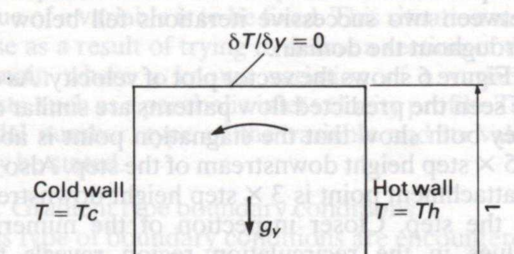


Fig. 8. Natural convection in a square cavity.

be incompressible. Bousinesque approximation is employed to account for buoyancy forces due to variations in temperature, so that:

$$F_x = 0 \quad \text{and} \quad F_y = \rho g_y \beta \Delta T \tag{7}$$

The analysis was carried out non-dimensionally for the Ra numbers of 10^3, 10^4 and 10^5. The governing equations (1a to 1d) were solved simultaneously and solutions were considered to have converged when the difference between two successive iterations became less than 5% throughout the flow domain. The computations were started with zero initial conditions for all variables in all three cases.

Figure 9 shows the computational grid employed for the case of $Ra = 10^5$ with 20×20 elements. The grid is typical of the type of meshing employed for this analysis. Finer grid lines are used near the side walls in order to obtain accurate temperature predictions there. The relative widths of the elements in the first 6 columns on each side of the cavity are 1:3:5:7:10. The computational grid for $Ra = 10^3$ and 10^4 consisted of 12×12 and 16×16 elements respectively. The graphical

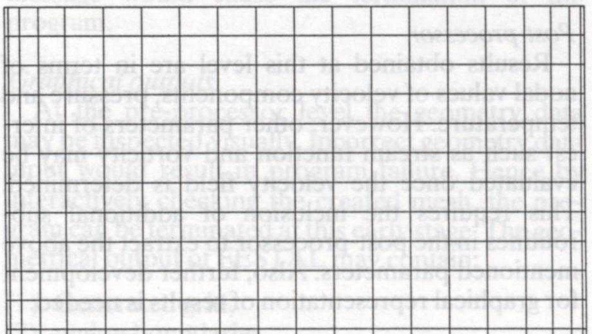
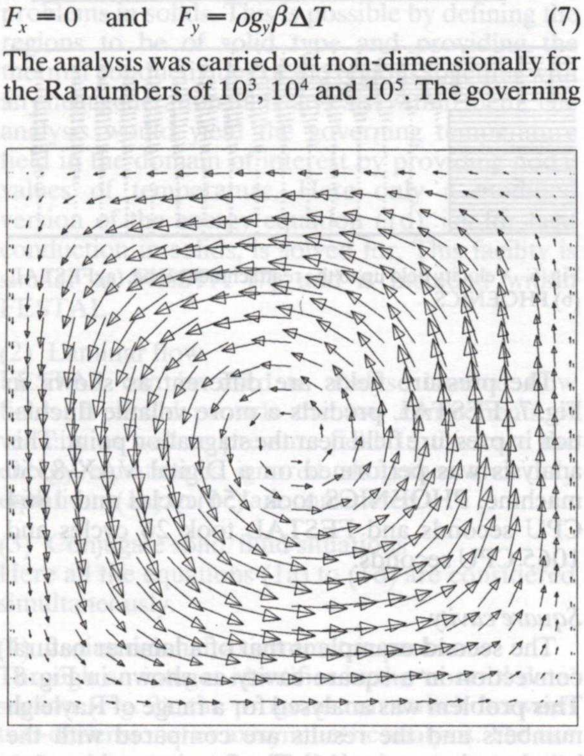


Fig. 9. Typical computational grid for natural convection in square cavity, $Ra = 10^5$, 20×20 elements.



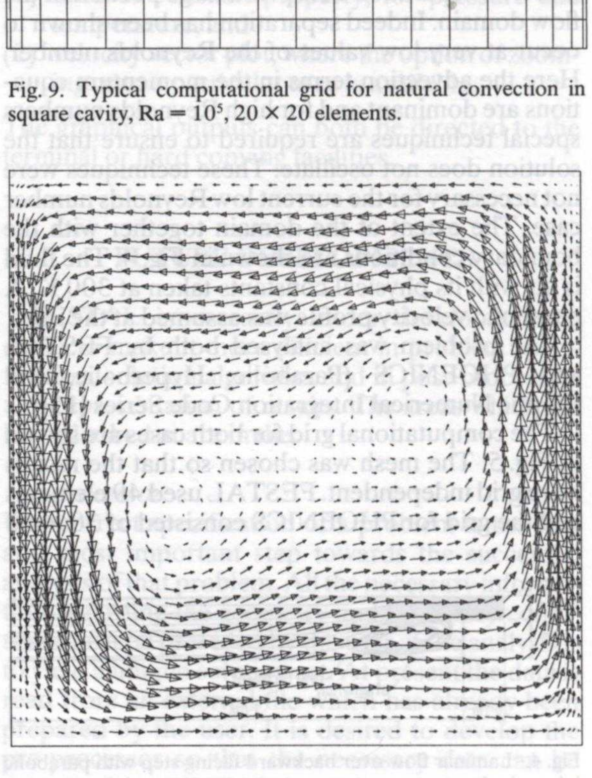


Fig. 10. Velocity field, (a) $Ra = 10^3$, (b) $Ra = 10^5$.

representation of the results for the two cases of Ra = 10^3 and 10^5 are shown in the following figures. Figure 10 shows the vectorial representation of the computed flow field. Streamline plots for the flow fields are shown in Fig. 11. The recirculation region exists at the centre of the cavity for Ra = 10^3, whereas for Ra = 10^5 two distinct circulation regions are apparent. Figure 12 shows the contour plot of the pressure fields. The isotherms for the two cases are shown in Fig. 13. Figure 14 shows the heat flux vectors in the cavity. Table 1 presents the numerical comparisons of selected variables at key locations within the domain with those of the benchmark [16]. The values are in very good agreement with the benchmark results considering the coarseness of the computational grid used for this analysis. Figure 15 shows the variation of the Nusselt number along the hot wall for the range of Ra numbers considered. This case was analysed on a Digital VAX-8550 machine. For Ra = 10^3, solution converged to within 5% in all variables after 43 cycles and 2408 CPU seconds. At Ra = 10^4, 76 cycles took 7566 CPU seconds to reach convergence, and for Ra = 10^5, 112 cycles took 17422 CPU seconds to yield the final results.

CONCLUSIONS

FESTAL is a viable finite element package for laminar fluid flow and heat transfer analysis. The

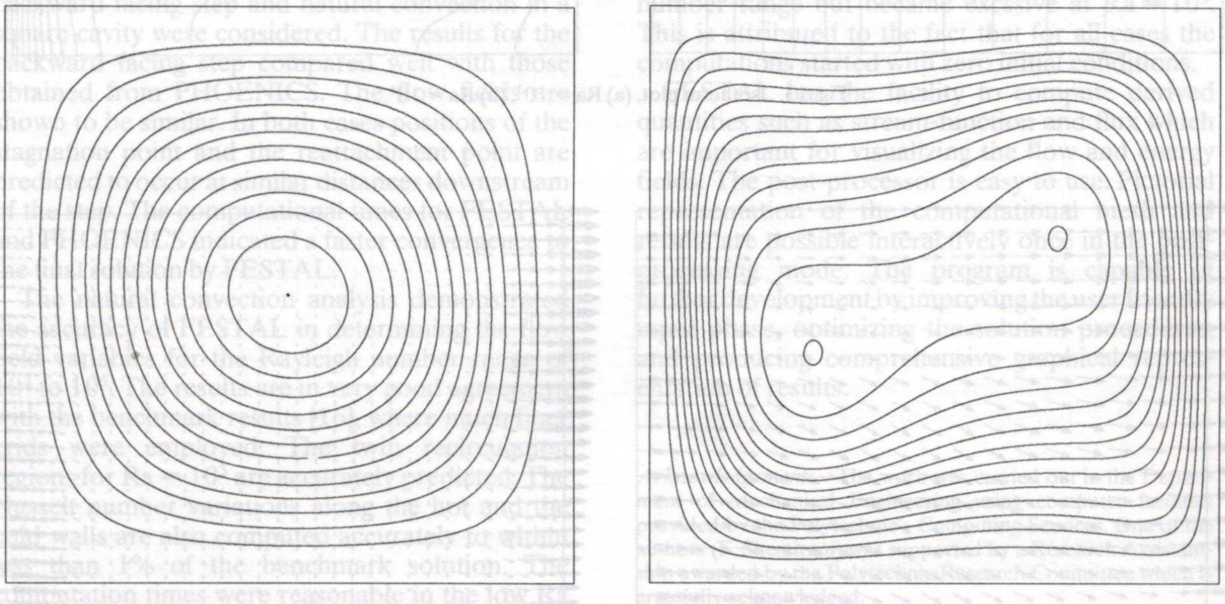


Fig. 11. Streamline plot, (a) Ra = 10^3, (b) Ra = 10^5.

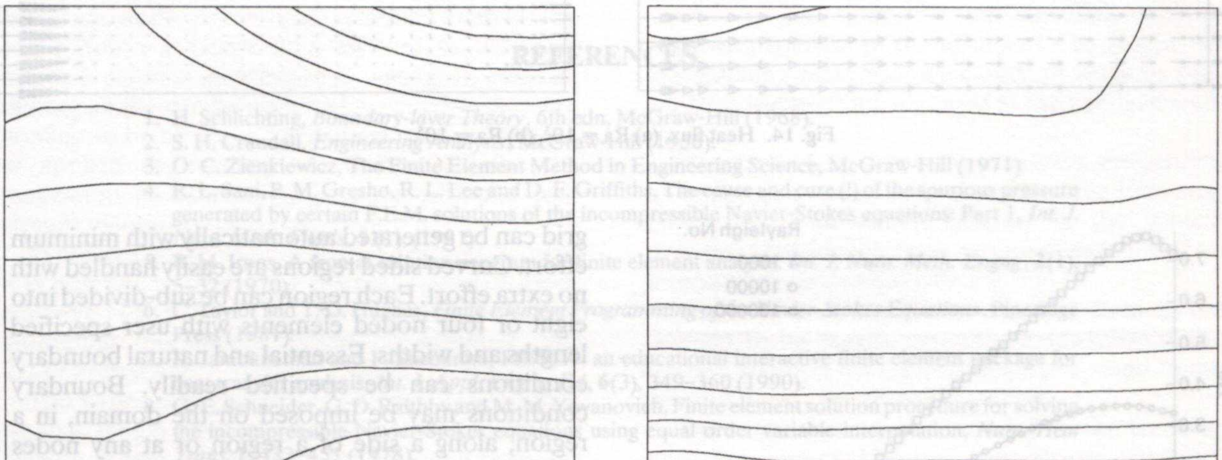


Fig. 12. Pressure field, (a) Ra = 10^3, (b) Ra = 10^5.

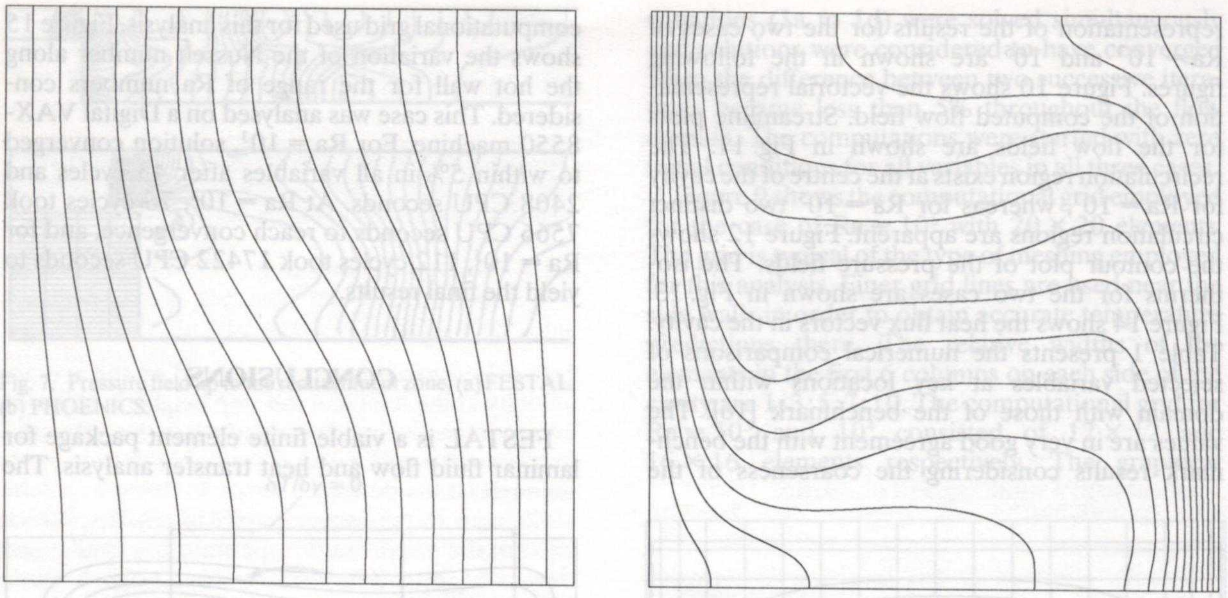


Fig. 13. Isotherm plot, (a) Ra = 10^3, (b) Ra = 10^5.

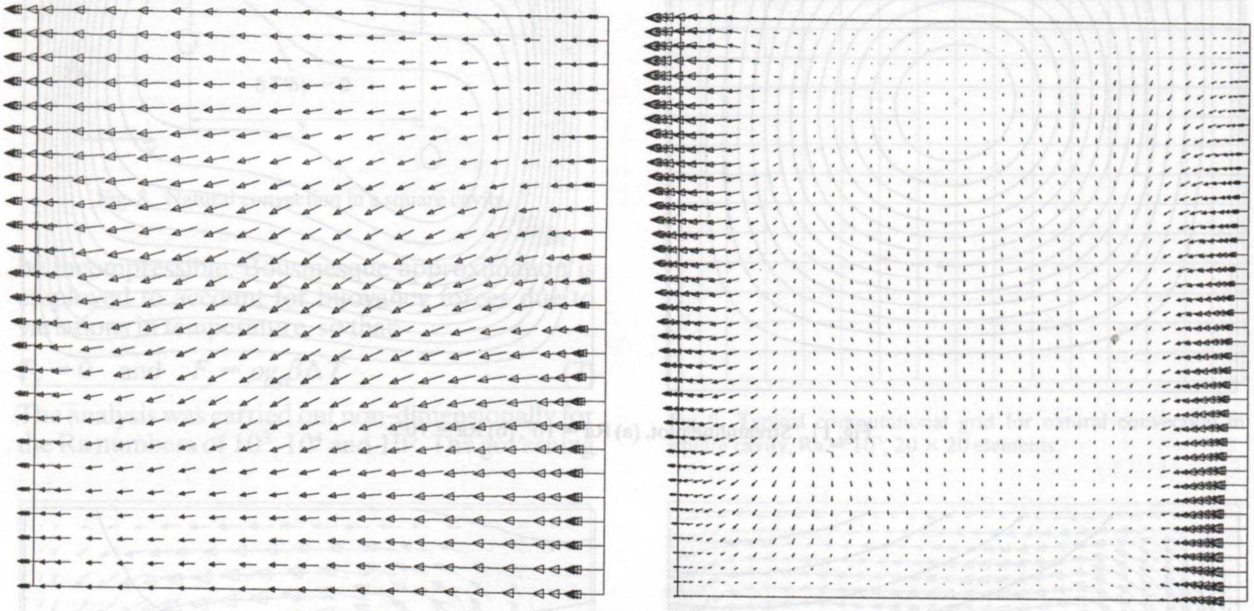


Fig. 14. Heat flux, (a) Ra = 10^3, (b) Ra = 10^5.

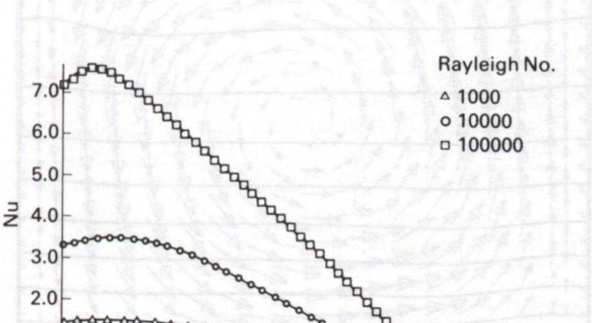


Fig. 15. Nusselt number variation along the hot wall.

grid can be generated automatically with minimum effort. Curved sided regions are easily handled with no extra effort. Each region can be sub-divided into eight or four noded elements with user specified lengths and widths. Essential and natural boundary conditions can be specified readily. Boundary conditions may be imposed on the domain, in a region, along a side of a region or at any nodes within the domain. The data input stage may be both interactive and/or in batch mode. Also at this stage, checks are performed on the input data in order to minimize the possibility of initiating an erroneous run.

Two cases of laminar isothermal flow over a

Table 1. Comparison of predicted results with those of the benchmark exercise[16]

	Ra = 10^3		Ra = 10^4		Ra = 10^5	
	benchmark	present work	benchmark	present work	benchmark	present work
ψ mid	1.174	1.174	5.071	5.072	9.111	9.135
ψ max	1.174	1.174	5.071	5.072	9.612	9.724
x, y	0.5, 0.5	0.5, 0.5	0.5, 0.5	0.5, 0.5	0.285, 0.601	0.290, 0.600
U max	3.649	3.630	16.178	16.126	34.730	43.518
y	0.813	0.950	0.823	0.813	0.855	0.847
V max	3.697	3.649	19.617	19.598	68.59	68.22
x	0.178	0.153	0.119	0.112	0.066	0.053
Nu avrg	1.118	1.118	2.243	2.244	4.519	4.561
Nu max	1.505	1.506	3,528	3,522	7.717	7.719
y	0.092	0.100	0.143	0.156	0.081	0.075
Nu min	0.692	0.691	0.586	0.584	0.729	0.727
x	1.0	1.0	10.0	1.0	1.0	1.0

backward facing step and natural convection in a square cavity were considered. The results for the backward facing step compared well with those obtained from PHOENICS. The flow fields are shown to be similar. In both cases positions of the stagnation point and the reattachment point are predicted to occur at similar distances downstream of the step. The computational times for FESTAL and PHOENICS indicated a faster convergence to the final solution by FESTAL.

The natural convection analysis demonstrated the accuracy of FESTAL in determining the flow field variables for the Rayleigh number range of 10^3 to 10^5. The results are in very good agreement with the benchmark results [16], where much finer grids were employed. The twin recirculation regions for Ra = 10^5 are accurately predicted. The Nusselt number variations along the hot and the cold walls are also computed accurately to within less than 1% of the benchmark solution. The computation times were reasonable in the low Ra number range but became excssive at Ra = 10^5. This is attributed to the fact that for all cases the computations started with zero initial conditions.

FESTAL has the facility to compute derived quantities such as stream-function and flux which are important for visualizing the flow and energy fields. The post-processor is easy to use. Pictorial representation of the computational mesh and results are possible interactively once in the post-processing mode. The program is capable of further development by improving the user friendly input phase, optimizing the solution procedures, and producing comprehensive graphical representation of results.

Acknowledgements—The work was carried out in the Department of Mechanical Engineering using computer facilities provided by the Polytechnic's Computing Services. One of the authors (F. Shemirani) was supported by a Research Assistantship awarded by the Polytechnic Research Committee which is gratefully acknowledged.

REFERENCES

1. H. Schlichting, *Boundary-layer Theory*, 6th edn, McGraw-Hill (1968).
2. S. H. Crandall, *Engineering Analysis*, McGraw-Hill (1956).
3. O. C. Zienkiewicz, The Finite Element Method in Engineering Science, McGraw-Hill (1971).
4. R. L. Sani, P. M. Gresho, R. L. Lee and D. F. Griffiths, The cause and cure (!) of the spurious pressure generated by certain F.E.M. solutions of the incompressible Navier-Stokes equations: Part 1, *Int. J. Num. Meth. Fluids*, **1** (1), (1981).
5. B. M. Irons, A frontal solution program for finite element analysis. *Int. J. Num. Meth. Engng*, **2**(1), 5–32 (1970).
6. C. Taylor and T. G. Hughes, *Finite Element Programming of the Navier-Stokes Equations*, Pineridge Press (1981).
7. K. Jambunathan and F. Shemirani, FIESTA, an educational interactive finite element package for heat transfer analysis, *Int. J. Appl. Engng. Ed*, **6**(3), 349–360 (1990).
8. G. E. Schneider, G. D. Raithby and M. M. Yovanovich, Finite element solution procedure for solving the incompressible Navier-Stokes equations using equal order variable interpolation, *Num. Heat Trans.* **1**, 433–451 (1978).
9. R. J. Schnipke and J. G. Rice, Examination of a new finite element applied to convection heat transfer, *FE in Analysis &Design*, **1**, 227–239 (1985).
10. J. G. Rice and R. J. Schnipke, An equal order velocity-pressure formulation that does not exhibit spurious pressure modes, *Comp. Meth. App. Mech. Engng*, **58**, 135–149 (1986).
11. T. J. R. Hughes, L. P. Franca and M. Balestera, A new finite element formulation for computational fluid dynamics II—Beyond SUPG, *Comp. Meth. App. Mech. Engng*, **54**, 341–355 (1986).
12. T. J. R. Hughes and M. Mallet, A new finite element formulation for computational fluid dynamics

III—The generalised streamline operator for multidimensional advective-diffusive systems, *Comp. Meth. App. Mech. Engng*, **58**, 305–328 (1986).

13. T. J. R. Hughes, L. P. Franca and M. Balestera, A new finite element formulation for computational fluid dynamics V—Circumventing The Babuska—Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations, *Comp. Meth. App. Mech. Engng*, **59**, 85–99 (1986).
14. R. J. Schnipke and J. G. Rice, A finite element method for free and forced convection heat transfer, *Int. J. Num. Meth. Engng*, **24**, 117–128 (1987).
15. D. B. Spalding, A general purpose computer program for multidimensional one and two phase flows, *Math. Comput. Simul.* **23**, 267–278 (1981).
16. S. De Vahl Davis and I. P. Jones, Natural convection in a square cavity: A comparison exercise, *Int. J. Num. Meth. Fluids*, **3**, 227–248 (1983).