III European Conference on Computational Mechanics
Solids, Structures and Coupled Problems in Engineering
C.A. Mota Soares et.al. (eds.)
Lisbon, Portugal, 5–8 June 2006

PARALLEL 3D FINITE ELEMENT ANALYSIS OF COUPLED PROBLEMS

Lee Margetts[1], Ian M. Smith[2], and Joanna M. Leng[1]

[1] Manchester Computing, University of Manchester
Kilburn Building, Oxford Road, Manchester M13 9PL
e-mail: lee.margetts@manchester.ac.uk, joanna.leng@manchester.ac.uk

[2] School of Mechanical, Aeronautical and Civil Engineering, University of Manchester,
P.O. Box 88, Manchester M60 1QD
ian.smith@manchester.ac.uk

Keywords: Parallel finite element analysis, magnetohydrodynamics, element-by-element, BiCGStab(l).

Abstract. *Steady advances in computer power have enabled researchers to consider tackling increasingly complex problems. In the academic community, current focus is on multiscale modelling and multi-physics. The aim is for simulation to be more realistically representative of real world processes. This paper considers the simulation of coupled problems involving more than one physical process, multi-physics. In particular, the authors present some ideas and experiences regarding the use of the finite element method and parallel computers to solve 3D coupled problems. In the literature, two main approaches have been used to solve coupled problems. These are sometimes referred to as (i) fully coupled modelling and (ii) uncoupled multi-physics. Both methods have their advantages and disadvantages. In the paper, the authors discuss some of the issues that should be considered when selecting a particular strategy, to ensure computational efficiency. Particular attention is given to an example from the field of magnetohydrodynamics: three dimensional steady state flow in a perfectly insulated rectangular duct. The magnetohydrodynamics example involves solving a system in which both magnetic and hydrodynamic forces influence the behaviour of the fluid. A fully coupled solution strategy is presented in which the full system is represented by a single "stiffness" matrix and solved by a single computer program. A parallel implementation of an element-by-element variant of BiCGStab(l) is used to solve the equations, demonstrating the efficient use of up to 128 processors.*

1 INTRODUCTION

In finite element analysis, parallel computing is typically employed to enable larger problems to be solved than is possible using a single processor or to drastically reduce solution times so that useful studies of a particular process can be carried out. In simple terms, the effectiveness of parallel computation in achieving these aims can be assessed with reference to the quantity scalability (a measurement of the benefit achieved by adding increasing numbers of processors). In this paper, the authors first consider some of the issues that affect scalability before describing in detail a fully coupled approach to the solution of a magnetohydrodynamics problem.

The two main characteristics of computer programs that limit scalability are (i) the ratio of the serial to the parallel fraction in the program and (ii) the ratio of computation to communication (message passing between processors).

In the first case, typified by Amdahl's Law [1], an example of a serial operation could be file input or output. The parallel fraction may be an "embarrassingly parallel" (very efficient) linear equation solver. Table 1 presents a hypothetical example to illustrate the limitations imposed by Amdahl's Law. The table shows speed-up for a hypothetical computational scenario in which serial and perfectly parallel fractions are equal on one processor. In this case, the best performance that can be expected using even a large number of processors is that the total solution time is half that when using one processor.

Processors	1	2	4	8	16	32	64	100
Time - serial fraction	50	50	50	50	50	50	50	50
Time - parallel fraction	50	25	12.5	6.25	3.13	1.56	0.78	0.5
Total solution time	100	75	62.5	56.25	53.13	51.56	50.78	50.5
Speed-up	1	1.33	1.60	1.78	1.88	1.94	1.97	1.98

Table 1 Consequences of Amdahl's Law

The second characteristic, the ratio of computation to communication, is illustrated most clearly by considering an uncoupled solution strategy. Using an uncoupled strategy, the physical processes and their interaction are dealt with separately. In magnetohydrodynamics, the fluid flow may be solved using one software application and the magnetic field solved using another. Time spent in the software packages may be considered "computation" and time spent in the transferring data from one software package to another may be considered "communication". Interaction using an uncoupled strategy is illustrated in Figure 1.

```
For each load or time step

    1. Compute the fluid flow in software package A       P
    2. Write results to intermediate storage              S
    3. Interpolate quantities to magnetic representation  S/P
    4. Compute magnetic field in software package B       P
    5. Write results to intermediate storage              S
    6. Interpolate quantities to fluid representation     S/P

Repeat
```

Figure 1 Steps in a Typical Uncoupled Solution Strategy

The benefit of the uncoupled approach is that the best available parallel solvers can be selected for the fluid and magnetic processes. One disadvantage is that these solvers may require different representations of the model (finite elements for the magnetic field and finite volumes for the fluid). Interpolation of results from one representation to another requires additional computation and may influence the final accuracy of the results. Intermediate storage is also required. For large problems, usage of disk will be necessary rather than in-core memory.

In Figure 1, the letters “P” and “S” represent parallel and serial respectively. With reference to Table 1, one can appreciate that the hypothetical uncoupled strategy will suffer with respect to Amdahl’s Law. In a parallel computing environment, total solution times may be dominated by serial activities, or communication between packages, with the consequence that parallel computation will be of little benefit to large, coupled problems.

In a fully coupled strategy, sometimes referred to as the “monolithic” approach, the physical processes may be solved as a single system of equations. One clear advantage is that the serial bottlenecks or communication overhead that arises in uncoupled strategies do not occur. Another benefit is that a scalable parallel solution strategy is more likely to exist for the fully coupled system, giving the potential to greatly reduce solution times using large numbers of processors. With little or no serial fraction, the limitation to scalability imposed by Amdahl’s Law does not occur.

Selecting a strategy and an appropriate solution algorithm for parallel computing requires considerable care. The authors have noted a tendency in the community to select an algorithm that runs well on a serial machine and try and adapt it to parallel computing. An efficient serial strategy is not necessarily efficient in parallel. Two metrics need to be taken into account when comparing solution strategies (i) the total number of operations required to find the solution and (ii) the rate at which those operations can be processed on a particular computer platform. For instance, when comparing uncoupled and fully coupled strategies, one may find that an uncoupled approach requires fewer operations to deliver a solution than a fully coupled method. However, if one takes into account the efficiency by which operations are processed and scalability on a parallel platform, an algorithm that requires more absolute work, may return results faster than a seemingly cheaper method (see Table 2).

	Absolute Cost Operations	% Use of Machine’s Peak Performance	Solution Time 1 Processor (Serial)	Solution Time 100 Processors
Algorithm A	1,000	1%	1 unit	0.5 unit
Algorithm B	10,000	10%	1 unit	0.01 units

Table 2 Comparison of Two Hypothetical Algorithms

In Table 2, Algorithm A requires fewer operations to solve the system than Algorithm B. However Algorithm A makes less efficient use of the computer hardware. Assuming that Algorithm A is the same one that is presented in Table 1 (equal serial and parallel fractions) and Algorithm B is embarrassingly parallel, an apparently costly algorithm has the potential to solve systems of equations much faster than a cheaper one when using large numbers of processors.

In summary, when considering the use of parallel computing to solve coupled physical problems, the following three observations need to be taken into account:

- Scalability is limited by the serial fraction in the application.
- Scalability is limited by communication between processors and applications.
- An expensive serial algorithm may be the most efficient on a parallel platform.

2 MAGNETOHYDRODYNAMIC DUCT FLOW

In this paper, a fully coupled approach to solving magnetohydrodynamic problems is presented. Magnetohydrodynamics is a field which concerns the study of the behaviour of electrically conducting fluids under the influence of magnetic fields. Magnetohydrodynamics has a range of diverse application areas. Examples include metal forming processes, measurement of the flow of coolants in nuclear reactors, plasma containment for fusion research and astrophysics.

The equations to be solved, for a steady state incompressible flow, are:

$$u_j \cdot \frac{\partial u_i}{\partial x_j} + \frac{1}{\rho}\frac{\partial p}{\partial x_j} - \frac{1}{\mu\rho} B_k \left(\frac{\partial B_i}{\partial x_k} - \frac{\partial B_k}{\partial x_i} \right) - \frac{\nu}{\rho} \nabla^2 u_i = 0 \tag{1}$$

$$u_j \cdot \frac{\partial u_i}{\partial x_j} + \frac{1}{\rho}\frac{\partial p}{\partial x_j} - \frac{1}{\mu\rho} B_k \left(\frac{\partial B_i}{\partial x_k} - \frac{\partial B_k}{\partial x_i} \right) - \frac{\nu}{\rho} \nabla^2 u_i = 0 \tag{2}$$

$$\frac{\partial \mathrm{u_i}}{\partial \mathrm{x_j}} = 0 \tag{3}$$

$$\frac{\partial \mathrm{B_i}}{\partial \mathrm{x_j}} = 0 \tag{4}$$

The equation relating to the fluid flow (1) and the equation relating to the magnetic field (2) both contain terms in velocity u_i and magnetic field B_i, i.e. they are coupled. Equations (3) and (4) relate to the continuity conditions for the u_i and B_i respectively. In addition to the usual fluid properties, μ (permittivity) and σ (conductivity) are required

The selected test problem concerns magnetohydrodynamic flow through a perfectly insulated rectangular duct. The geometry of the problem, together with the boundary conditions, is shown in Figure 2. The full field is applied in the region indicated by dark grey. In the light grey zone, the strength of the applied field decays to zero. Although the geometry is simple, the authors are unaware of any other work in which this system has been solved in three dimensions using the finite element method.

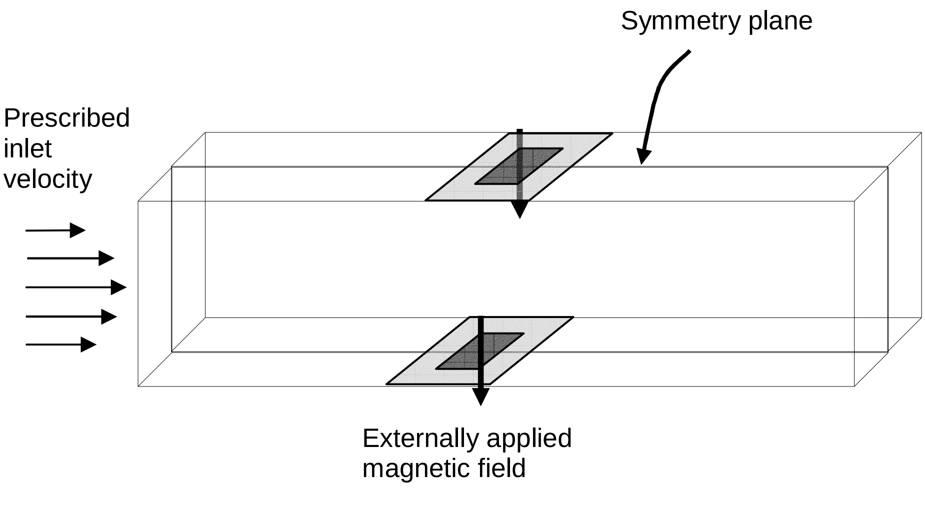


Figure 2 Geometry and Boundary Conditions for Magnetohydrodynamic Duct Flow

The problem has been discretised using hexahedral elements. Twenty noded bricks have been used to represent the fluid velocity, eight nodes for the fluid pressure and twenty nodes for the magnetic field. With three degrees of freedom for both the fluid velocity and the magnetic field and one degree of freedom for the pressure, each full element "stiffness" matrix has 128 degrees of freedom. The full reconstructed finite element mesh is shown in Figure 3. For the purposes of simulation, symmetry was taken into account and only half the model shown was used.

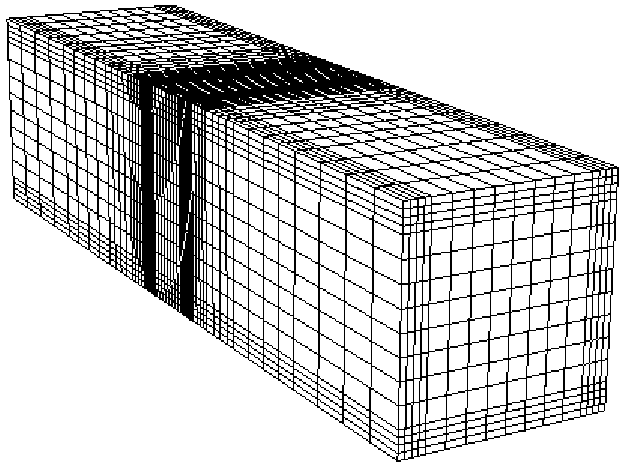


Figure 3 Full Reconstructed Finite Element Mesh

The equilibrium equation to be solved (Equation 5) is of the form [Ke](r) = (f), where Ke is the element stiffness matrix. All the entries C_{ij} are submatrices of size 60 x 60 except for C_{21}, C_{23}, C_{24}, C_{12}, C_{32} and C_{42} which are of size 8 x 60.

Navier Stokes — Coupling Terms

$$\left[\begin{array}{cccc:ccc} C_{11} & C_{12} & 0 & 0 & C_{15} & C_{16} & C_{17} \\ C_{21} & 0 & C_{23} & C_{24} & 0 & 0 & 0 \\ 0 & C_{32} & C_{11} & 0 & C_{35} & C_{36} & C_{37} \\ 0 & C_{42} & 0 & C_{11} & C_{45} & C_{46} & C_{47} \\ \hdashline C_{51} & 0 & 0 & 0 & C_{55} & 0 & 0 \\ 0 & 0 & C_{51} & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & C_{51} & 0 & 0 & C_{55} \end{array}\right] \begin{Bmatrix} u \\ p \\ v \\ w \\ B_x \\ B_y \\ B_z \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{5}$$

Coupling Terms — Magnetic Induction

The stiffness matrix has a structure that can be broken down into regions that represent the fluid properties, the magnetic properties and the interaction between the two physical processes. In an uncoupled approach, the fluid flow and magnetic field problems may be solved in separate software packages. The interaction would involve an offline interpolation between the two representations. This is in clear contrast to the fully coupled approach where the interactions are built in at the element level through the coupling terms.

3 SOLUTION STRATEGY

In three dimensions and with 128 degrees of freedom per element, a direct solver would quickly run out of core memory for this problem. Therefore a more memory efficient strategy such as an element-by-element approach with an iterative solver is essential. As the element stiffness matrices are unsymmetrical, BiCGStab(l) is an appropriate choice of iterative solver.

In previous work the authors have developed a library of routines that enable scalable parallel finite element analysis for a wide range of problem types [2]. The routines, together with example parallel programs are freely available in the form of an open source library, ParaFEM [3]. This library has been used as a basis for developing the parallel program to solve the magnetohydrodynamic problem presented here.

The problem is subdivided across processors by assigning an equal number of finite elements to each one. In domain decomposition techniques, a global matrix is assembled and later decomposed. In the element-by-element approach, a global matrix is never created, and perhaps domain composition is a more apt description of the process followed here.

At the core of the element-by-element implementation of BiCGStab(l) is a series of element loops comprising matrix-vector multiplications. This is where most of the computational effort is concentrated. One might argue that the modern commodity processors used in supercomputers are optimised for such operations. On the machine used for this study, an SGI Origin 3800, matrix-vector computations can achieve 40% of the machine's peak performance and matrix-matrix 70%. This is in contrast with direct solvers that typically achieve 1 or 2% peak.

Although the presence of matrix-vector multiplication implies that BiCGStab(l) makes efficient use of the processors, there are two computational issues that can be addressed to further improve performance. Firstly, refering to Equation (5), the element matrices are sparse. A standard matrix-vector multiplication would carry out many redundant floating point operations per element, multiplying by zero. Secondly, large element matrices rapidly fill cache memory. Most current computers use a small high speed cache memory to reduce the time for memory access during computation. Once the processor starts to retrieve data from the main core memory, performance or the efficient utilization of the processor suffers.

To counter these two inefficiencies, redundant calculations and poor cache usage, the finite element based matrix-vector computations can be optimised. The key is to operate at the submatrix level, storing by sub-matrix (C_{ij}) and looping over sub-matrices rather than elements. The advantage is a reduction in total memory usage and a reduction in the total number of floating point operations carried out per finite element. Operating on submatrices requires less data to be loaded from main memory during each stage in the matrix-vector computation – the data is more likely to fit in the faster cache memory. Furthermore, no operations are carried out involving the sparse regions of the element stiffness matrix. Reductions in solution times of up to factor of 4 have been achieved using this implementation [4].

4 RESULTS

The results presented in this section are for the solution of a steady state magnetohydrodynamic duct flow problem with 4 million equations. Figures 4 and 5 present performance data, whereas Figures 6-8 illustrate features of the fluid flow and the magnetic field. In the analyses, full account of problem symmetry was considered in order to reduce solution times. For clarity in the visualizations, the original duct geometry has been restored.

Figure 4 shows the computational speed up achieved using up to 128 processors on the SGI Origin 3800 system. The dashed line shows the ideal speed up.

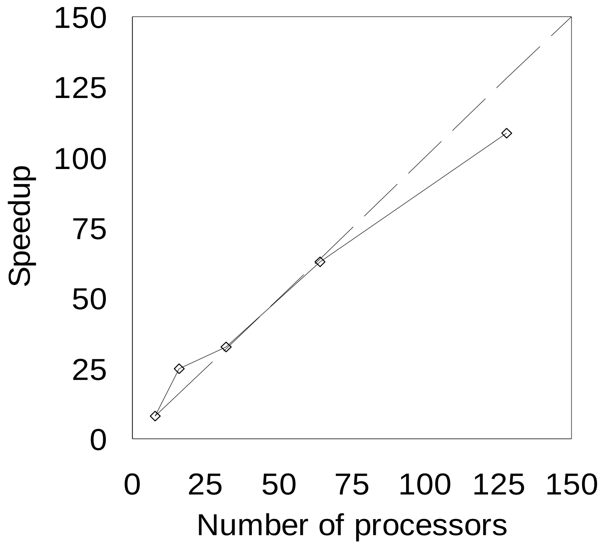


Figure 4 Speed Up For 4 Million Equation Magnetohydrodynamic Problem

Figure 5 presents peak performance. The dashed line shows the peak performance achieved in the computational kernel, the matrix-vector computations in the BiCGStab(l) algorithm. The solid line shows the peak performance for the whole program. The difference between the two is mainly attributed to communication costs. The authors note that further improvements may be made to reduce communication overhead.

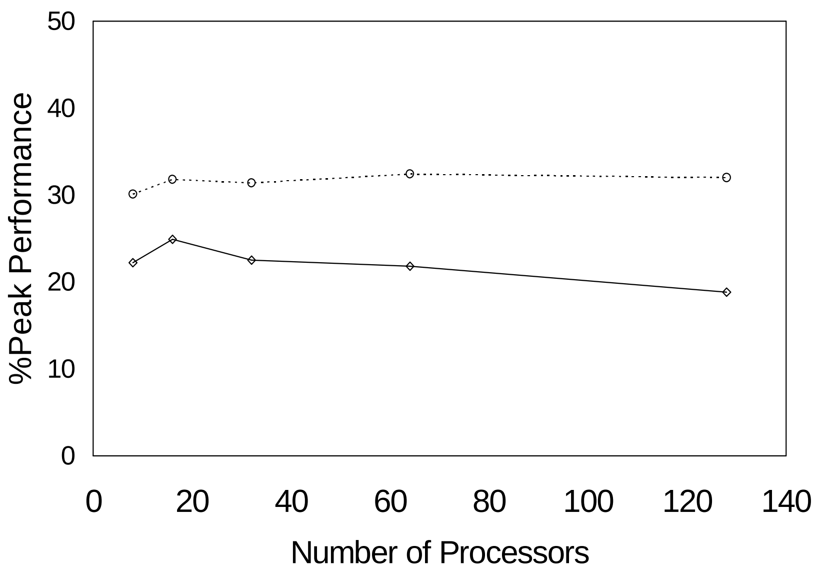


Figure 5 Percentage of Peak Performance

Figure 6 shows three images that illustrate the interaction between the fluid flow and the magnetic field. Each image shows an outline of the rectangular duct, with the fluid flowing in the direction of the arrows, from bottom to top. The position of the external magnetic field is indicated by grey squares.

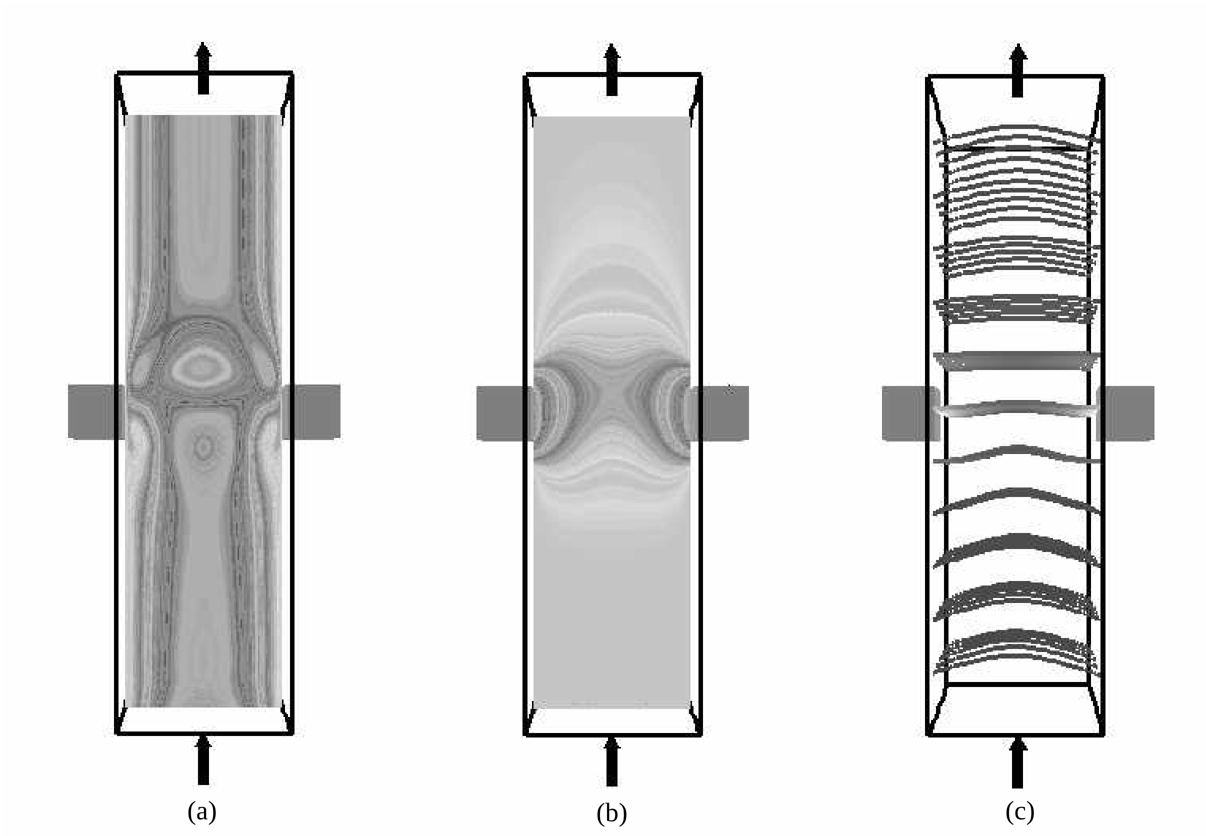


Figure 6 Interaction Between Fluid Flow and Magnetic Field

In Figures 6a and 6b, a randomized colour map of high frequency stripes is used to highlight the features of the fluid flow and magnetic field respectively. As the colour map is randomized, the transition from light to dark grey does not represent low to high values.

Figure 6a shows the magnitude of the fluid velocity and it is clear that the fluid flow is affected by the magnetic field. Figure 6b shows the magnitude of the magnetic field. The image clearly indicates that the magnetic field is dragged downstream by the fluid flow. Figure 6c shows the interaction between the two physical processes with the distortion of the magnetic field lines.

Figure 7 again uses a randomized greyscale colour map to represent the magnitude of the velocity. Slice planes are used to illustrate the flow at three locations along the duct, with the downstream direction running from left to right (Figures 7a to 7c). Figure 7a represents an approximately parabolic flow profile, as was prescribed at the duct inlet. Figure 7b shows how this is disrupted by the influence of the magnetic field. In the final image, Figure 7c, the flow takes the form of two jets. These are represented by the two circular regions. The splitting of a parabolic flow into two fluid jets by the application of a magnetic field is a well-known feature of magnetohydrodynamic flows through ducts and has been demonstrated experimentally elsewhere [5].

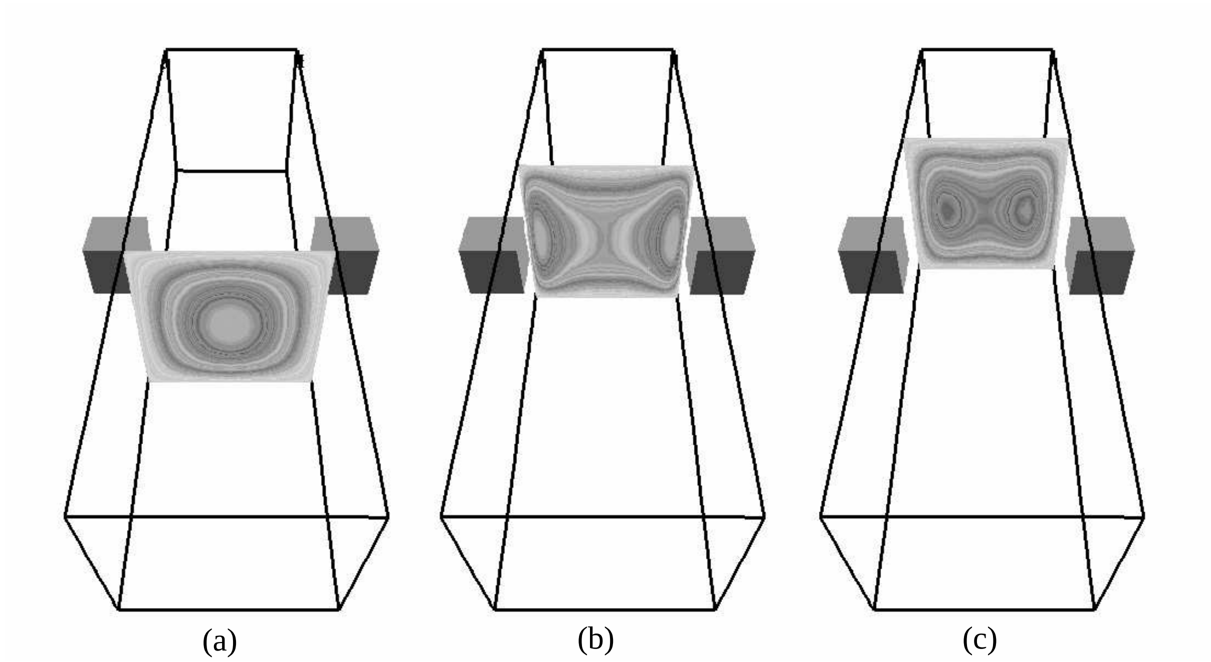
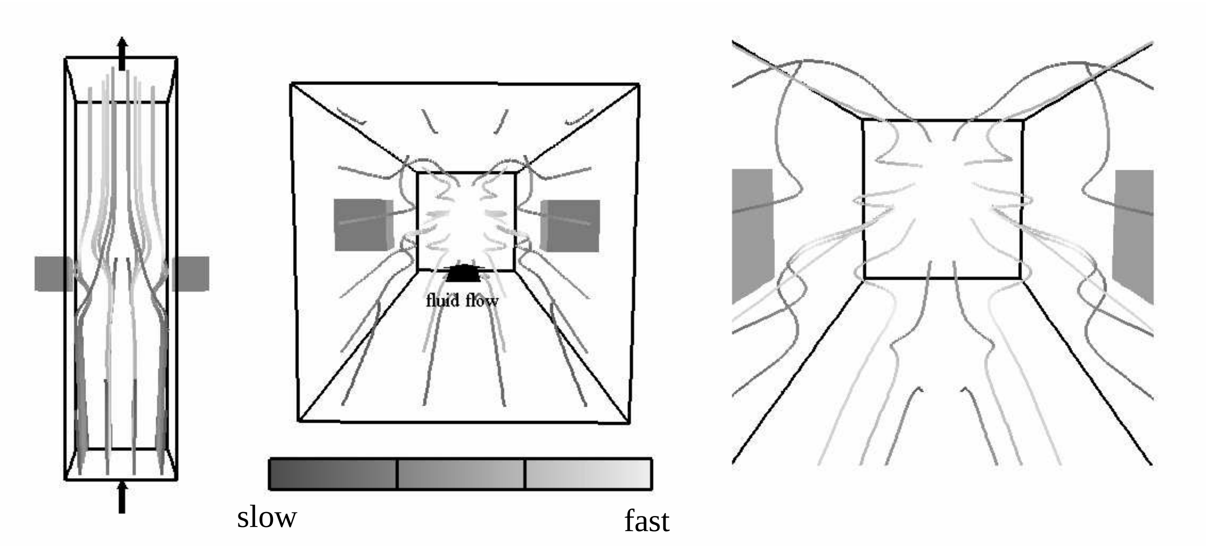


Figure 7 Variation of Velocity Magnitude Along the Duct

Figure 8a shows fluid streamlines, indicating the path of a number of selected fluid particles as they pass through the duct. In this case, a uniform greyscale colour map is used: going from grey for slow flow to white for fast flow. Figures 8b and 8c show the same streamlines with the observer viewing them from the duct inlet and from a downstream position, close to the applied magnetic field, respectively. The streamline trajectories clearly show that a three dimensional finite element model is needed to capture the full complexity of the physical processes being studied.



(a) (b) (c)

Figure 8 Velocity Streamlines From Three Different Viewpoints

5 CONCLUSIONS

Two techniques for the solution of coupled problems have been described, the uncoupled and the fully coupled approaches. Using simple hypothetical examples, it has been shown that careful consideration must be made in selecting the best strategy, especially when use is made of parallel computing.

A magnetohydrodynamics problem has been used to illustrate how the fully coupled approach lends itself well to parallel computation. In this paper, performance figures have been shown for a four million equation problem that can be solved efficiently on up to 128 processors. Specific optimizations of the element-by-element solution strategy have been described that take advantage of the sparsity of the element stiffness matrices. Finally, the physical results have been visualized for fully coupled three dimensional flow through an insulated rectangular duct. The visualizations of streamlines show that the flow is complex and can only be properly captured in a three dimensional model.

Finally, the authors note that this is not a particularly large analysis. Systems of more than half a billion equations have recently been described using up to 4088 processors [6].

REFERENCES

[1] J.J Dongarra et al., *Numerical linear algebra for high performance computers*. SIAM, 1998.

[2] I.M. Smith and D.V. Griffiths, *Programming the finite element method*, 4th Edition, Wiley, 2002.

[3] www.parafem.org.uk, 2006.

[4] L. Margetts, *Parallel finite element analysis,* PhD Thesis, University of Manchester, 2002.

[5] H. Branover, *Magnetohydrodynamic flow in ducts*, Israel Universities Press, 1978.

[6] M.F. Adams et al., Ultrascalable implicit FE analysis in solid mechanics with over half a billion degrees of freedom, *Supercomputing,* Pheonix, Arizona, 2004.