Accepted for Publication in CNME
August 2001

On a Multilevel Preconditioning Module for Unstructured Mesh Krylov Solvers: Two -level Schwarz[1]

R. S. Tuminaro[2], C. H. Tong[3], J.N. Shadid[2], K.D. Devine[2], D.M. Day[2]

ABSTRACT: Multilevel methods offer the best promise to attain both fast convergence and parallel efficiency in the numerical solution of parabolic and elliptic partial differential equations. Unfortunately, they have not been widely used in part because of implementation difficulties for unstructured mesh solvers. To facilitate use, a multilevel preconditioner software module, **ML,** has been constructed. Several methods are provided requiring relatively modest programming effort on the part of the application developer. This report discusses the implementation of one method in the module: a two-level Krylov-Schwarz preconditioner. To illustrate the use of these methods in computational fluid dynamics (CFD) engineering applications, we present results for 2D and 3D CFD benchmark problems.

1. INTRODUCTION

The numerical solution of partial differential equations (PDEs) often requires the solution of linear systems that arise from discrete approximations to PDE operators. Among iterative methods to solve these linear systems, preconditioned Krylov schemes are the methods of choice. Unlike the currently popular incomplete factorization preconditioners, multigrid methods are relatively suitable for parallel implementation [1] and their convergence rates do not deteriorate as grids are refined [2]. The idea of multigrid is to capture errors at different frequencies by using a hierarchy of grids with different amounts of resolution. By traversing different grids, optimal convergence rates are frequently observed with overall solution times that are much faster than methods using incomplete factorizations. Unfortunately, multigrid methods are difficult to implement and incorporate into applications, and are consequently rarely found in large complex application codes. While algebraic multigrid strategies address some of these difficulties, additional research is required to make the techniques general enough for widespread use.

To address the complexities of multigrid research, our goal has been to design a multilevel code module (**ML**) incorporating a wide variety of multilevel methods. A preliminary discussion of the ML interface can be found in [3]. In this communication, we briefly discuss the status of the two-level Schwarz domain decomposition capability.

2. MULTILEVEL METHODS

Multilevel methods (e.g., multigrid) are used to solve systems of equations. In general, multigrid methods approximate the original PDE problem of interest on a hierarchy of grids and use 'solutions' from coarse grids to accelerate the convergence on the finest grid. In Figure 1, we illustrate a simple multilevel iteration. The $S_k()$'s are approximate solvers (or more popularly called smoothers). These usually correspond to basic iterative methods such as Gauss-Seidel. Within domain decomposition (DD) methods, the subdomain direct solve or incomplete factorization can be viewed as an approximate solver. The P_k's and R_k's are grid transfer operators that interpolate and restrict solutions between grids (denoted G_k) in a hierarchy. Finally, the A_k's are discrete approximations to the PDEs on each grid within the hierarchy. Often the same discretization scheme can be used for each grid although this is not always appropriate. A more robust choice is

1. This work was partially funded by the U.S. Department of Energy's Mathematical, Information and Computational Sciences Division, and was carried out at Sandia National Laboratories operated for the U.S. Department of Energy under contract no. DE-ACO4-94AL85000.

2. Sandia National Laboratories, Albuquerque, NM, 87185-1111, jnshadi@sandia.gov

3. Lawrence Livermore National Laboratory, Livermore, CA, 94550-9234, chtong@appleton.llnl.gov.

the Galerkin approximation $A_{k-1} = R_k A_k P_k$. The precise definition of these operators depends on the application and the multigrid method. Our ML software module supports several ways of creating these operators including a number of algebraic multigrid methods, a way of importing user-defined operators, and a two-level domain decomposition scheme that uses the application's finite element basis functions to transfer between grids in the hierarchy.

```
proc multilevel(A_k,b,u,k)
   /* Solve A_k u = b  (where k is the current grid level). */
{
        if (k == 0) then u = A_0^{-1} b;
        else {S_k(A_k ,b,u);
              residual = b - A_k u ; v = 0;
              /* solve on coarser grid. */
              multilevel(A_{k-1}, R_k * residual,v,k-1)
              u = u + P_k v
        }
}
```

Figure 1. Pseudo-code for a generic multi-level solver

2.1. A Two-level Preconditioner

In this brief discussion, we present a two-level preconditioner based on the popular one-level Schwarz (overlapping domain decomposition) preconditioner[4]. In the one-level Schwarz method, each processor is assigned a subdomain of the entire grid which can overlap with subdomains of other processors. The preconditioner essentially consists of each processor approximately solving its local subdomain problem and combining its results with those of the neighboring processors in the overlapped regions (e.g. averaging values). The subdomain solve might be a direct solve, an incomplete solve, or an iterative solve. The Schwarz preconditioner is fairly popular because it is easy to implement (e.g. it does not require grid information) and gives good results on a variety of problems. A number of publicly available parallel iterative software packages contain Schwarz preconditioners [5,6]. Unfortunately, the convergence rate of the Schwarz preconditioner deteriorates as the number of subdomains (processors) increases. The is due primarily to global decoupling in the preconditioner; i.e., coupling between distant subdomains is ignored. A coarse grid solution step (very coarse compared to the fine grid) can be added to the basic Schwarz method to approximate the global coupling in the PDE operator. A proper two-level Schwarz method can significantly reduce solution time. Furthermore, the convergence degradation of the resulting preconditioner is provably quite modest and depends only weakly on the size of subdomains. Unfortunately, library software to do this coarse solution step is generally not publicly available. To address this problem, we have implemented a coarse grid solution step within the multilevel module. This coarse solve is based on both the serial and parallel versions of SuperLU[7] which is a direct sparse linear system solver. Since the local subdomain 'solvers' or smoothers can be obtained from many iterative libraries, these are not implemented. Instead, the multilevel module is designed to be used with an iterative solver package that supplies the outer iterative scheme as well as the subdomain 'solves.' To fully specify the two-level Schwarz preconditioner, we refer to the multilevel algorithm given in Figure 1. Specifically, take $k = 1$ and let $S_1()$ correspond to the one-level Schwarz preconditioner. We require that G_0 and G_1 be supplied by application developers. While this is inconvenient, it is important to note that G_1 is generally available and the one additional grid can often be created with the same grid generator used to generate G_1. ML can

generate A_0 via the Galerkin formula or it can be supplied by the application. All results in this paper were generated with A_0 obtained by explicit discretization on G_0. P_1 is defined implicitly by the application's elemental basis functions. That is, users supply simple routines indicating how information is interpolated within a specific finite element. The multilevel module then uses these routines to perform the tedious task (especially in a parallel environment) of constructing P_1. In this construction ML allows for the coarse grid and fine grid to be partioned differently for parallel execution. This flexibility is required to produce a general interface for FE application codes that do not use mesh adaptivity to produce the fine grid from an initial coarse grid. In practice the cost of this possible imbalance and non-optimal communication structure is small since the coarse grid is usually significantly smaller than the corresponding fine grid. Finally, R_1 is taken as the transpose of P_1.[1] The interaction between the application and the multilevel module is summarized in Figure 2.

3. The Governing Equations and the Numerical Formulation

To illustrate the usefulness of the multilevel approach we briefly present a comparison of the one- and two-level DD preconditioners applied to the solution of two thermal convection problems and a Stokes flow problem. The governing elliptic PDEs (Navier-Stokes with the thermal energy equations) are shown in Table 1. These equations are discretized by a Galerkin least squares (GLS)

Momentum	$\rho(\mathbf{u} \bullet \nabla\mathbf{u}) - \nabla\bullet\mathbf{T} - \rho\mathbf{g} = 0$
Total Mass	$\nabla\bullet\mathbf{u} = 0$
Thermal Energy	$\rho\hat{C}_p\left[\mathbf{u} \bullet \nabla T\right] + \nabla\bullet\mathbf{q}_c = 0$

Table 1. Governing PDEs

finite element formulation which allows for equal order of interpolation of velocity and pressure[8]. The resulting nonlinear system of equations is solved using a fully-coupled inexact Newton method along with Krylov solvers as implemented in our Aztec Krylov solver library [5]. The details of the parallel finite element implementation, convergence of the fully-coupled inexact Newton methods, and the parallel performance and convergence of various standard preconditioning strategies can be found in [9,10,11].

4. Preliminary results

To illustrate the use of the two-level DD preconditioners as implemented in the ML library we present characteristic results for a standard 2D benchmark thermal convection flow problem along with a generalization to a 3D problem. In this standard 2D benchmark problem [11], a thermal convection (or buoyancy-driven) flow in a differentially heated square box in the presence of gravity is modeled. The momentum transport, energy transport and total mass conservation equations defined in Table 1 are solved on a unit square. No-slip boundary conditions are applied on all walls. The temperature on the heated wall and other parameters are chosen so that the Rayleigh number, Ra,

1. Other techniques exist to construct grid transfer operators. However, these are not discussed here.

can be varied. The 3D problem adds two no-slip insulated walls in the third dimension to form a 1x1x1 cube; a solution for this problem with $Ra = 1000$ is shown in Figure 3. These simple geometries facilitate algorithmic/parallel studies as different mesh sizes can be easily generated. The results were obtained on the ASCI-Red Tflop computer at Sandia National Laboratories.
In Table 2 and Table 3 results are presented for the algorithmic scaling of the one- and two-level

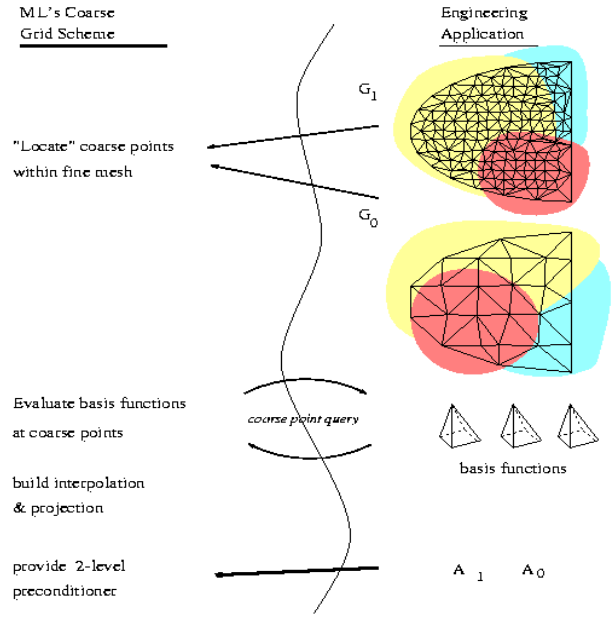

Figure 2. Schematic representation of interaction between application and multilevel module for 2 - level Schwarz method

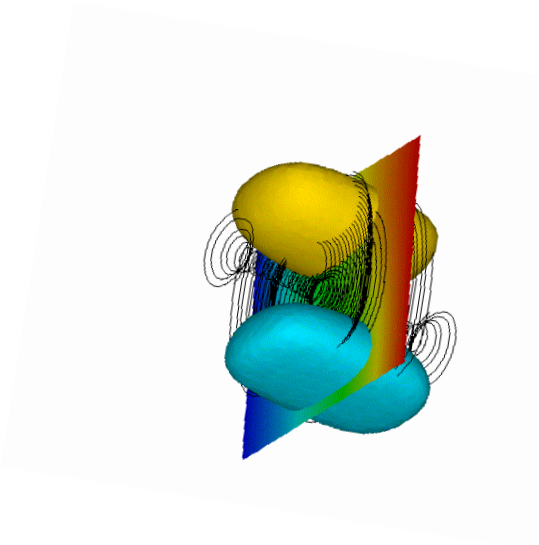

Figure 3. Constant x-velocity iso-surfaces with streamlines and temperature contours on slice plane. Ra = 1,000. Pr = 1

schemes. Clearly as the number of unknowns, N, is increased the number of iterations to convergence for the one-level schemes increases significantly. This increase is roughly proportional to $N^{2/3}$ in 2D and $N^{1/2}$ in 3D. The two-level schemes are shown to be optimally convergent for the given fine-to-coarse grid ratio of 64 in both cases. The CPU time comparison indicates that while the two-level scheme can be faster, careful attention needs to be directed to the coarse grid solve times. In the 2D cases the serial version of SuperLU is replicated to solve exactly the coarse grid problem and on the 3D case a parallel version of SuperLU was invoked. Since using all processors to factor this small system is not efficient, groups of approximately $P^{1/2}$ are utilized to solve a partially replicated linear system. Since the fine grid smoother is highly parallel [10] and the fine grid work per processor is fixed, it is the SuperLU performance on the increasingly larger coarse grid that causes unfavorable performance on the larger problems. In the 3D case, the larger bandwidth or fill-in of the direct factorization is apparent even at moderate sized coarse grid problems. However, by using even coarser coarse grids or approximate solves instead of direct solves, it is possible to overcome the computational bottleneck associated with this direct solve. One option is to apply an iterative method such as a 1-level DD ILU factorization on the coarse grid. These results shown in the lower entries of Table 3 indicate that even this inexact coarse grid solve provides a suitable correction to the fine grid problem to accelerate convergence. However for this case the optimal convergence property is not obtained and a modest increase in the number of iterations is evident.

			One-Level Method		Two-Level Method		
No. of Processors	**Fine Grid Size**	**Total Unknowns**	**Avg Its per Newton Step**	**Total Time (sec.)**	**Coarse Grid Size**	**Avg Its per Newton Step**	**Total Time (sec.)**
1	**32x32**	**4,096**	**47**	**28**	**4x4**	**38**	**36**
4	**64x64**	**16,384**	**110**	**64**	**8x8**	**42**	**49**
16	**128x128**	**65,536**	**258**	**237**	**16x16**	**44**	**93**
64	**256x256**	**262,144**	**622**	**1483**	**32x32**	**41**	**181**
256	**512x512**	**1,048,576**	**1591**	**9198**	**64x64**	**40**	**920**

Table 2. Comparison of one- and two-level Schemes for 2D Thermal Convection Problem, Ra=1000, Pr=1., non-restarted GMRES, one-level - DD ILU, two-level Gauss-Siedel Smoother, SuperLU Coarse Grid Solver

			One-Level Method		Two-Level Method		
No. of Processors	**Fine Grid Size**	**Total Unknowns**	**Avg Its per Newton Step**	**Total Time (sec.)**	**Coarse Grid Size**	**Avg Its per Newton Step**	**Total Time (sec.)**
1	**8x8x8**	**2560**	**18**	**44**	**2x2x2**	**14**	**35**
8	**16x16x16**	**20,480**	**47**	**70**	**4x4x4**	**19**	**56**
64	**32x32x32**	**163,840**	**102**	**145**	**8x8x8**	**19**	**143**
512	**64x64x64**	**1,310,720**	**308**	**532**	**16x16x16**	**19**	**807**
1	**8x8x8**	**2560**	**18**	**44**	**2x2x2**	**15***	**30***
8	**16x16x16**	**20,480**	**47**	**70**	**4x4x4**	**25***	**60***
64	**32x32x32**	**163,840**	**102**	**145**	**8x8x8**	**39***	**179***
512	**64x64x64**	**1,310,720**	**308**	**532**	**16x16x16**	**67***	**315***

Table 3. Comparison of One and Two Level Schemes for 3D Thermal Convection Problem, Ra=1000, Pr=1., non-restarted GMRES, one level - DD ILU, two- level Gauss-Siedel Smoother, Superlu Coarse Grid Solver.

*. Coarse grid solve corresponds to a DD ILU factorization/backsolve in parallel with two levels of overlap.

Our last example of Stokes flow in a channel with an obstruction demonstrates the two-level Schwarz capability on an unstructured mesh problem for which the fine mesh is not a refinement of the coarse mesh. In this study the meshes were independently generated and therefore totally unrelated in structure (see Figure 4). As well since each mesh is partitioned separately by an automatic tool, Chaco [12], the resulting mesh partitions are not aligned in any way. In this example the one level solver uses a domain decomposition ILUT preconditioner with roughly 2 times as many nonzeros (fill-in=2.0) as the original matrix and two levels of subdomain overlap [5]. The two level solver uses the standard DD-ILU solver as a smoother with one level of overlap. As is evident from the convergence results presented in Table 4, optimal convergence rates are obtained along with faster solution times.

5. CONCLUSIONS

Multigrid technology has received considerable attention in recent years due to its fast convergence rates compared to classical preconditioning methods. The implementation complexities and the knowledge required to make it efficient for a given application, however, hinder its widespread use. In an effort to increase the usefulness of these methods for various applications, we have begun multilevel preconditioning research and development in two areas: (1) design of an objected-oriented user interface for finite element applications, and (2) research, development, and implementation of new multilevel techniques. A framework encompassing the two-level technique

with a user-supplied coarse grid and coarse grid operator was developed and tested on two thermal convection problems and a Stokes flow example. These results demonstrated optimal convergence for the required number of iterations-to-solution and very encouraging decreases in CPU times over the one-level domain decomposition methods for these test problems.

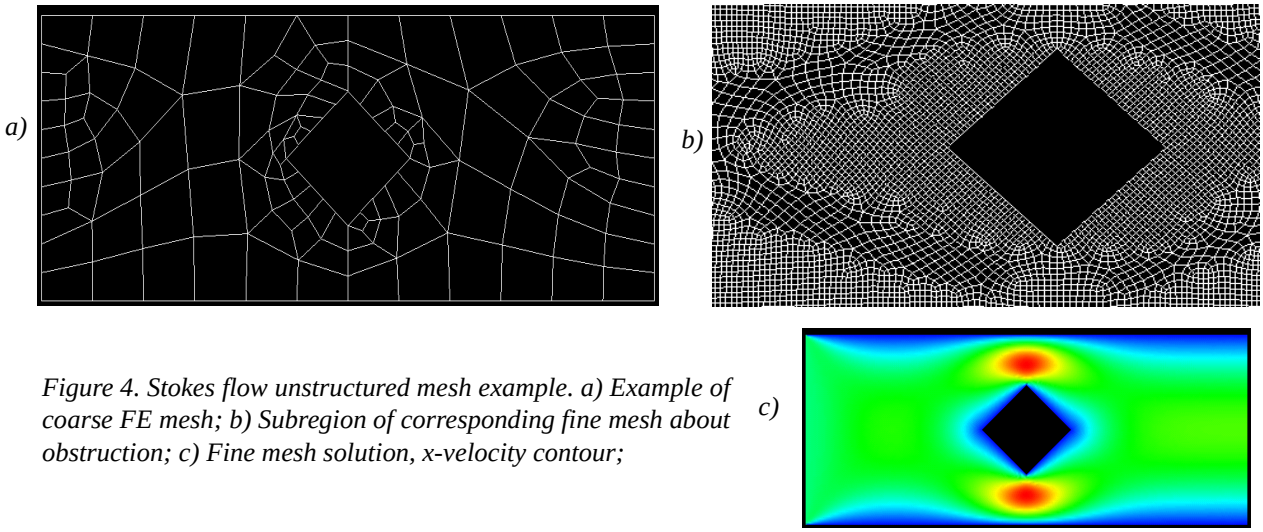

Figure 4. Stokes flow unstructured mesh example. a) Example of coarse FE mesh; b) Subregion of corresponding fine mesh about obstruction; c) Fine mesh solution, x-velocity contour;

		One-Level		Two-Level	
Procs	Unknowns	Iterations	Time (sec.)	Iterations	Time (sec.)
16	4704	32	5.9	25	6.0
64	13,008	73	9.9	33	6.6
256	55,008	316	48.3	34	14.1

Table 4. Stokes flow problem shown in Figure 6, GMRES solver. one-level - DD ILUT(2.,.2), two-level DD ILU smoother, coarse solver SuperLU, fine/coarse mesh ratio is approximately 64.

6. ACKNOWLEDGMENTS

The authors would like to thank Gary Hennigan, Scott Hutchinson, Harry Moffat and Andrew Salinger for collaboration in developing MPSalsa which was used to generate the results in this study. We also want to thank Colin O'Rourke for help with making a number of computational runs that were used in this study.

7. REFERENCES

1. J.N. Shadid and R.S. Tuminaro, “A Comparison of Preconditioned Nonsymmetric Krylov Methods on a Large-Scale MIMD Machine,” *SIAM J. Sci. Comput*., Vol 15, No. 2, pp 440-459, March 1994
2. W.L., Briggs, V.E. Henson, and S.F., McCormick, “A Multigrid Tutorial”, SIAM, (2000)
3. C.H. Tong, and R.S. Tuminaro, “ML 2.0 Smoothed Aggregation User’s Guide”, Sandia National Laboratories Technical Report, SAND2001-8028, (2000)
4. B. Smith, P. Bjorstad, and B. Gropp, “Domain Decomposition: Parallel Multilevel Methods for Partial Differential Equations”, Cambridge Univ. Press, NY, (1996)
5. S. A. Hutchinson, L., Prevost, J. N. Shadid, C. Tong, and R. S. Tuminaro, “Aztec User’s Guide Version 2.0", Sandia National Laboratories Technical Report, Sand99-8801J 1999
6. W. D. Gropp and B. F. Smith, Scalable, extensible, and portable numerical libraries, Proceedings of the Scalable Parallel Libraries Conference, IEEE 1994, pp. 87-93
7. J. Demmel, S. Eisenstat, J. Gilbert, X. Li, and J. Liu, “A Supernodal Approach to Sparse Partial Pivoting”,. to appear in SIAM J. Mat. Anal. Appl.
8. Hughes, J. R., Franca, L. P., Balestra, M., “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. and Eng*., 59 (1986) 85–99
9. J. N. Shadid, H. K. Moffat, S. A. Hutchinson, G. L. Hennigan, K. D. Devine, and A. G. Salinger. “MPSalsa: A Finite Element Computer Program for Reacting Flow Problems: Part 1 - Theory Document.” Sandia National Laboratories Technical Report, SAND95–2752, Albuquerque, NM, (1996).
10. J.N., Shadid, S.A.,Hutchinson, G.L., Hennigan, H.K., Moffat, K.D., Devine, A. G. Salinger, “Efficient Parallel Computation of Unstructured Finite Element Reacting Flow Solutions”, *Parallel Computing* 23, 1307-1325, 1997
11. J.N. Shadid, R.S. Tuminaro and H.F. Walker. “An Inexact Newton Method for Fully-Coupled Solution of the Navier-Stokes Equations with Heat and Mass Transport.” *J. Comput. Phys*., 137, 155-185 (1997)
12. B. Hendrickson and R. Leland. “A user’s guide to Chaco, Version 1.0.” Sandia National Laboratories Technical Report, SAND93-2339, Albuquerque, NM, (1993).

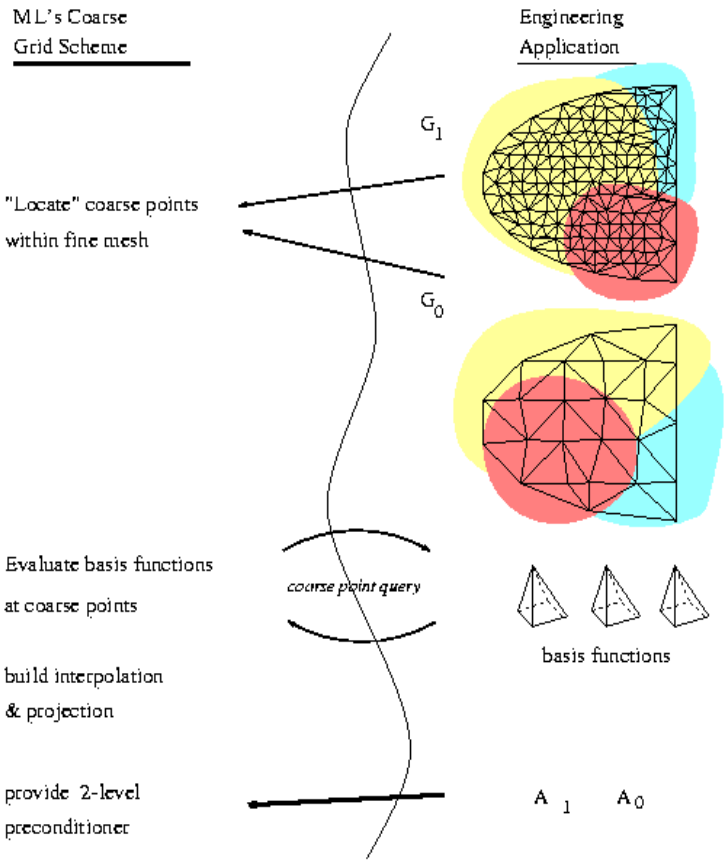

Figure 2. Schematic representation of interaction between application and multilevel module for 2 - level Schwarz method

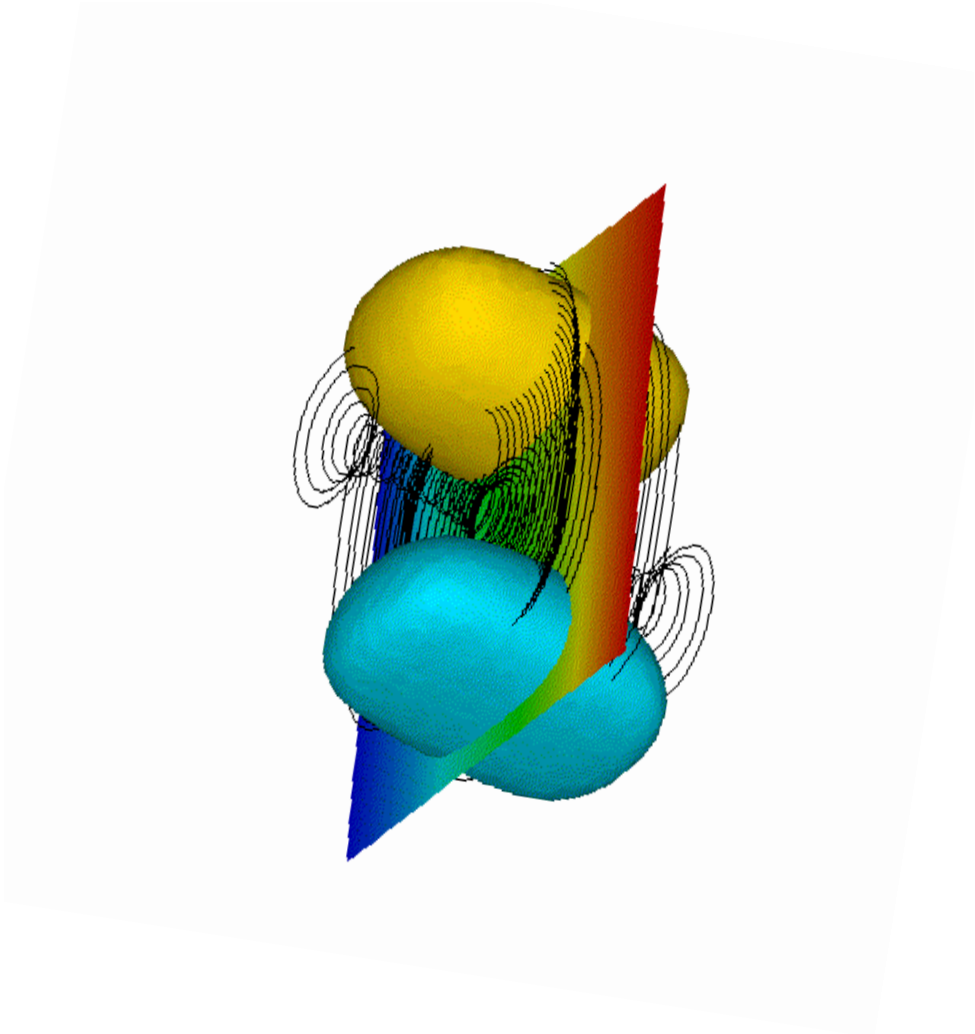

Figure 3. Constant x-velocity iso-surfaces with streamlines and temperature contours on slice plane. Ra = 1,000. Pr = 1

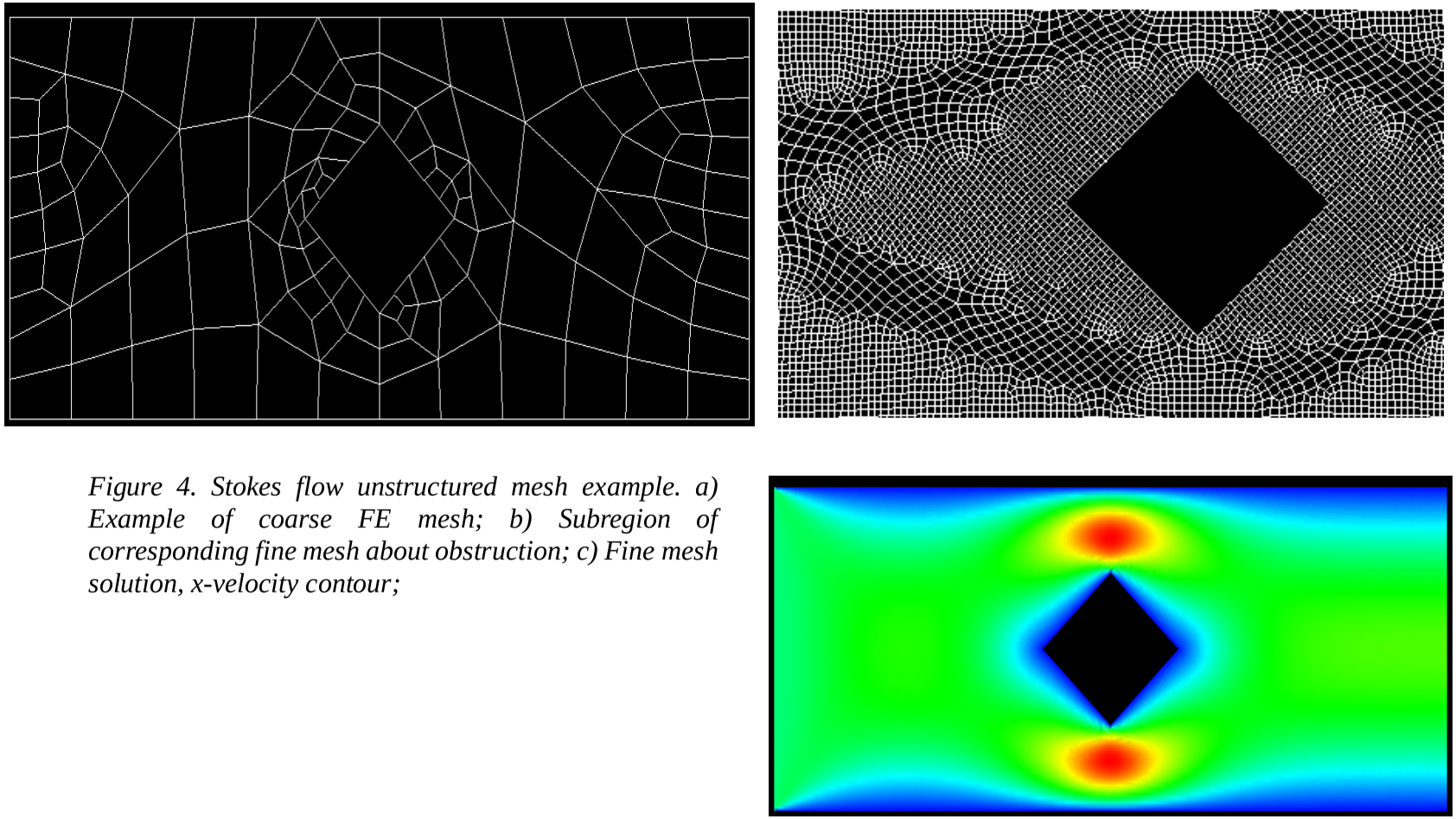

Figure 4. Stokes flow unstructured mesh example. a) Example of coarse FE mesh; b) Subregion of corresponding fine mesh about obstruction; c) Fine mesh solution, x-velocity contour;