Modeling of Microfluidic Devices using Scalable Algorithms

Linda Stals

Department of Computer Science, Old Dominion University, USA, stals@cs.odu.edu
& ICASE, NASA Langley Res. Ctr, USA, stals@icase.edu

ABSTRACT

We present parallel algorithms for the solution of microfluidic equations. The algorithm consists of the multigrid method in combination with adaptive finite elements. These methods give good algorithmic and parallel scalability and are thus very efficient. Model test problems are taken from Electroosmotic flow simulations.

Keywords: Numerical Methods, Finite and Boundary Element Methods, Multigrid Methods, Parallel Programming

1 Introduction

Models simulating flow through microchannels usually consist of complicated equations defined on complex domains and are thus very expensive to solve. For example, solving the equations which model reactive flow through microchannels consumes a great deal of computational resources, as pointed out by Hsing in [4]. It is this type of equation which we are aiming to solve. One way of meeting such requirements on computing power is to use parallel computers (or distributed computing), but this approach on its own is not enough. To make full use of available resources one must use state-of-the-art mathematical algorithms. We feel that this approach will allow us to include details in the model which would otherwise be impractical.

In this paper we present some scalable algorithms for use in the modeling of microfluidic devices. These algorithms are designed to show both parallel and algorithmic scalability. By algorithmic scalability we mean that the solution time should only increase linearly with the number of grid points.

2 Solution Techniques

To achieve the aims mentioned above we are using multigrid methods and/or domain decomposition methods in combination with adaptive finite elements. As the name suggests the multigrid method uses several layers of (usually) nested grids. Each grid layer is designed to remove different frequency components of the error and it has been shown that by combining the information associated with each grid level in a certain way, this method gives optimal convergence for many types of equations. See for example [1], [2], [6], [7]. In the domain decomposition method the grid is subdivided into a set of overlapping subdomains. The problem on each subdomain is solved independently so this method is well suited to a parallel environment. If a coarse grid is used to recover the global information then domain decomposition methods also show optimal convergence for different types of partial differential equations. See for example [11] or the WWW site `http://www.ddm.org/`. Another good source of information on both the multigrid and domain decomposition methods is the MGNet Home Page [3].

The nested sequence of grids needed by the multigrid method is built by repeatedly applying a refinement procedure. That is, we start with an initial coarse grid and then refine the grid to build the next finest level. This procedure is repeated until the grid is fine enough to give an accurate solution. The grids may be built by using either uniform refinement or adaptive refinement.

These algorithms have been implemented in a C^{++} program built upon a very flexible parallel data structure. The program has been used to model a variety of problems such as the flow through heterogeneous material [5] and Plasma Ion Immersion Process [10]. In the case of flow through heterogeneous material we were able to solve problems which were larger then what had been previously reported in the literature.

3 Example Runs

3.1 Model Problem

To demonstrate these ideas we have taken some equations from the cross-channel problem presented in [8] and [9]. Many examples of biological micromechanical systems consist of an injection channel and a separation channel which run perpendicular to one another. The separation of fluid components is usually carried out in the capillary (separation) channels and the flow into these channels is controlled by an electrical current. This is modeled by electroosmotic flow.

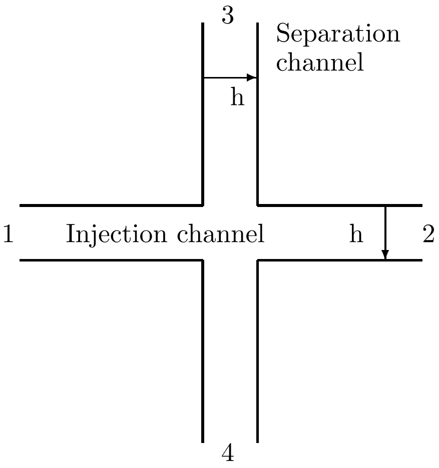
The results presented in this paper are for the two equations:

$$
\begin{aligned}
\nabla^2\psi &= \lambda\sinh(c\psi), & (1)\\
\nabla^2\phi &= 0, & (2)
\end{aligned}
$$

where ϕ is the potential due to the external electric field, and ψ is the potential due to the charge on the walls. The values for λ and c used in our current examples are $\lambda = 2 \times 10^4$ and $c = 0.02$.

The full set of equations given in [8] and [9] also include Navier Stokes equation for the velocity and pressure. We have extended our code to handle Navier Stokes equations and plan to conduct some example runs soon.

The injection channel and separation channel are represented by the following domain:



When solving for ψ, the potential is set to -1 on all of the walls with insulation-type boundary at the reservoirs. In the case of the potential ϕ the boundary is set to insulation-type boundary condition on all of the walls and $\phi = 1$ at reservoir 1 and $\phi = 0$ at reservoir 2. The value of ϕ at reservoirs 3 and 4 can be varied to simulate different experimental settings.

As pointed out in [9] there is a singularity at the four corners where the two bars cross. The spacing of the grid near these four corners needs to be reduced to ensure an accurate solution. We use an automatic adaptive refinement procedure to capture the rapid change in the solution.

3.2 L-shaped Domain

The singularity at these corners is similar to the singularity which occurs at the corner of an L-shaped domain, as shown in Figure 1. As an initial test of our algorithm we tried to solve the equation $\Delta u = 0$ on the L-shaped domain with boundary conditions such that the exact solution is $u = r^{2/3}\sin(2/3\theta)$.

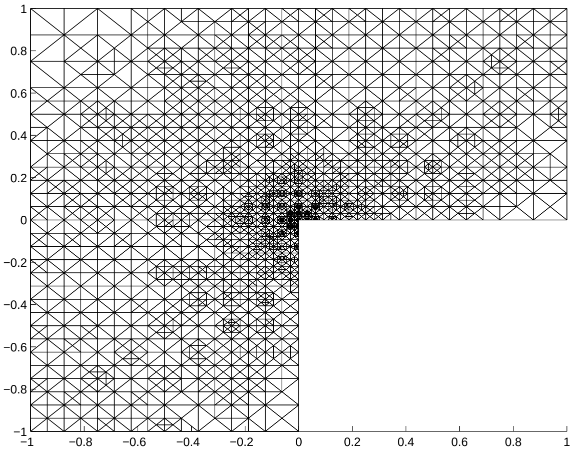


Figure 1: Fine grid after five levels of adaptive refinement.

Table 1: Table showing solution time and convergence rate with uniform refinement of the L-shaped domain.

n	$\lVert . \rVert_\infty$ $\times 10^{-5}$	Solver (sec)	Refine. (sec)	FEM (sec)
225	2000	0.24	0.034	0.035
833	1300	1.1	0.13	0.14
3201	830	4.9	0.54	0.58
12545	520	23	2.2	2.4
49665	330	100	9.1	9.8

The results for uniform refinement are given in Table 1. The first column is the number of nodes n. The second column, labeled $||.||_\infty$, shows the maximum error. Notice that the convergence rate falls far short of the ideal rate of $O(h^2)$. Table 2 gives the results for adaptive refinement. In this case the convergence rate is close to $O(h^2)$. As a consequence of the increased convergence rate we are able to obtain a higher degree of accuracy in a reduced amount of time. For example, using uniform refinement it takes approximated 6 seconds to get an error of 8.3×10^{-3}, but when using adaptive refinement an error of 8.0×10^{-3} is obtained in about 2.3 seconds.

Table 2: Table showing solution time and convergence rate with adaptive refinement of the L-shaped domain.

n	$\lVert . \rVert_\infty$ $\times 10^{-5}$	Solver (sec)	Refine. (sec)	FEM (sec)
92	2100	0.23	0.054	0.020
408	800	1.5	0.70	0.075
2068	160	4.1	1.7	0.38
9199	34	23	8.8	1.9
44420	5.6	140	47	9.5

Columns 3-5 in Tables 1 and 2 show the time spent in the major modules. The ‘Solver’ columns displays the amount of time require to solve the system of equation using the multigrid method[1]. In Table 1 we see that the solution time is linearly dependent on the number of nodes, as predicted. The linear dependence is not so clear in Table 2. Before refining a grid we must solve the system of equations on that level and the solution time shown in Table 2 includes these coarse grid solves. If we just looked at the time required to solve the problem on the finest, final, grid then it would be more easy to see the linear dependence. Finally, the column labeled ‘Refine’ shows the time spent refining the grids and the column labeled ‘FEM’ gives the time taken to form the discrete system of equations by using the finite element method.

3.3 Example Runs

We now present some example results obtained from the solution of Equations 1 and 2. When solving for ϕ we set the boundary conditions at reservoir 3 and 4 to be 0.9. The solution for ϕ and ψ are given in Figures 2 and 3 respectively. A physical interpretation of the results is given in [8] and [9], we are going to concentrate more on the properties of the algorithms.

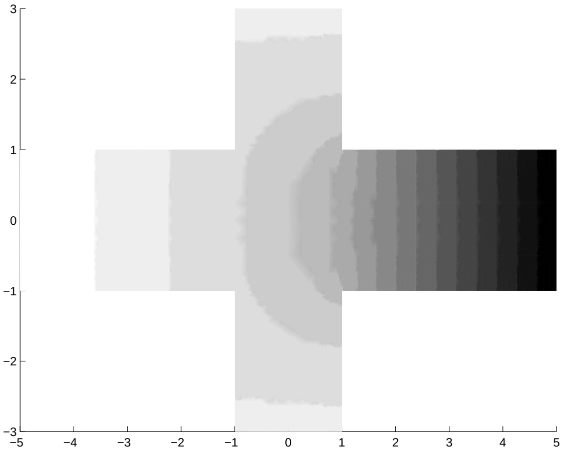


Figure 2: Potential due to external electric field.

Figure 4 shows a resulting grid if the adaptive refinement procedure is applied to Equation 2. Notice that most of the refinement is around the walls, which reflects the way that ψ changes rapidly near the walls.

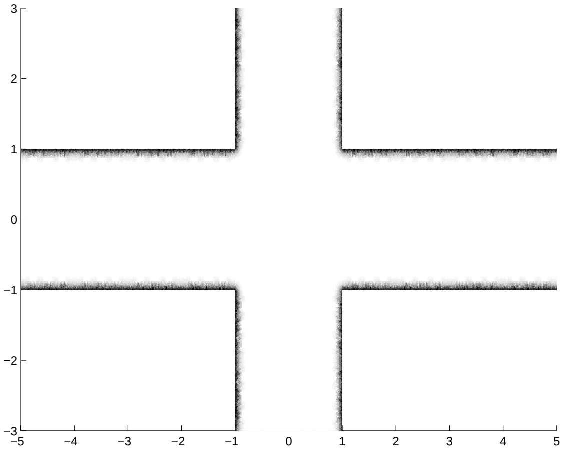


Figure 3: Potential due to charge on walls.

All of the steps used in the solution of the equations can be done in parallel, including the refinement procedure. In Figure 4 we can see what the grid looks like if it is refined on four processors. The dark region passing through the center of the cross show the regions where the grid is shared by two or more processors.

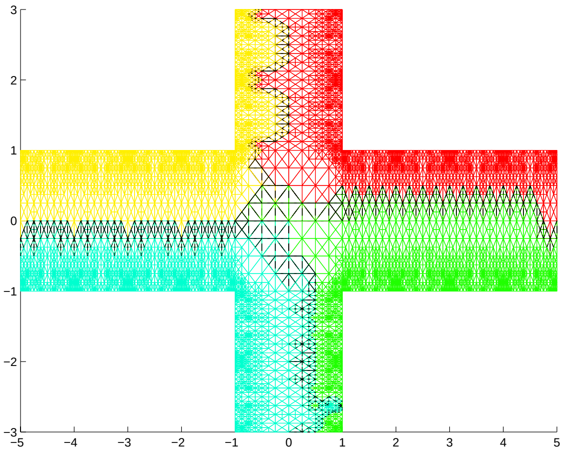


Figure 4: Fine grid after one level of uniform refinement followed by three levels of adaptive refinement. The grid has been split up over four processors.

The final property we would like to look at is the parallel scalability of the algorithms. Table 3 highlights the time taken to solve Equation 2 on different numbers of processors. Uniform refinement was used[2]. The

[1] We used six iterations of the V-scheme method. Six sweeps of the Jacobi method formed the pre and post smoothers. Interpolation operator is linear interpolation and the restriction operator is defined to be the transpose of the interpolation operator.

[2] In this case 12 pre and 12 post smoothers were used in the multigrid method.

Table 3: Table showing solution time for solve Equation 2 in parallel.

p	n	Total (sec)	Solver (sec)	Refine. (sec)	FEM (sec)	E %
1	54145	213.2	192.4	10.1	10.6	
2	54145	117.4	105.5	6.3	5.5	91
4	214785	231.3	207.8	11.8	10.8	91
8	214785	122.1	107.4	6.4	5.4	87
16	855553	249.4	221.8	13.3	10.9	84
32	3415314	599.1	437.5	28.5	21.5	67

efficiency was calculated by using the following equation

$$E = \frac{t_1}{pt_p}\frac{n_p}{n_1}$$

where t_p is the time for p processors and n_p is the number of nodes.

We feel that the results presented in Table 3 are very encouraging. It is difficult to get high efficiency results for this type of problem since we start with a very coarse grid (81 nodes) which means that we have to refine the grid several times before we have enough nodes to fill up all of the processors. Fortunately though these costs are one-off costs. When the computations for Navier Stokes equations are included we believe that we will see better efficiency results because the percentage of time spent rebalancing the load will be reduced. Note that the solver module is showing good efficiency results (87% for 32 processors).

4 Conclusion

We presented a parallel algorithm for the solution of equations arising in microfluidic models and showed that they give good parallel and algorithmic efficiency results.

REFERENCES

[1] J. H. Bramble. *Multigrid Methods.* Pitman research notes in mathematics series; 294. Longman Scientific & Technical, Essex, England, 1993.

[2] W. L. Briggs. *A Multigrid Tutorial.* SIAM, Philadelphia,1987.

[3] C. C. Douglas and M. B. Douglas. MGNet, URL `http://www.mgnet.org/mgnet/bib/mgnet.bib`, Yale University, Department of Computer Science, New Haven, CT (USA), 1991-1999.

[4] I-M. Hsing, R. Srinivasan, K. F. Jensen and M. A. Schmidt Finite Element Approach for Reactive Microfluidic Devices. *Technical Proceedings of the Second International conference on Modeling and Simulation of Microsystems*, pages 492–497. San Clara, California, Apr. 6-8, 1998.

[5] K. A. Cliffe, I. G. Graham, R. Scheichl and L. Stals. Parallel Computation of Flow in Heterogeneous Media using Mixed Finite Elements. Technical Report, University of Bath , Bath BA2 7AY, U.K. Bath Mathematics Preprint 99/16. July 1999. Submitted for publication in J. Computational Physics.

[6] W. Hackbusch. *Multigrid Methods and Applications.* Computational Mathematics; 4. Springer–Verlag, Berlin, 1985.

[7] S. F. McCormick, editor. *Multigrid Methods.* SIAM Frontiers In Applied Mathematics, Philadelphia, 1987.

[8] M. J. Mitchell and N. R. Aluru. Simulation of Electroosmosis using a Meshless Finite Point Method. *Technical Proceedings of the Second International conference on Modeling and Simulation of Microsystems*, pages 522-525. San Juan, Puerto Rico, Apr. 19-21, 1999.

[9] N. A. Patankar and H. H. Hu. Numerical Simulation of Electroosmotic Flow. *Anal. Chem*, **Vol 70, No. 9** pages 522-525, May 1998.

[10] M. Paulus, L. Stals, U. Rüde and B. Rauschenbach. Two-Dimensional Simulation of Plasma-Based Ion Implantation. *Journal of Applied Physics*, **Vol 85, No. 2**, pages 761-766, January 1999.

[11] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, New York, 1996.