CONFERENTIE VAN NUMERIEK WISKUNDIGEN

2 oktober - 4 oktober 1989

CONFERENTIEOORD WOUDSCHOTEN
ZEIST

Werkgemeenschap Numerieke Wiskunde

CONFERENTIE VAN NUMERIEK WISKUNDIGEN

Werkgemeenschap Numerieke Wiskunde

VEERTIENDE CONFERENTIE NUMERIEKE WISKUNDE

Doel van de conferentie

De Conferentie Numerieke Wiskunde wordt eenmaal per jaar gehouden onder auspiciën van de Werkgemeenschap Numerieke Wiskunde. Het doel van de conferentie is om kennis te nemen van recente ontwikkelingen binnen de numerieke wiskunde. Hiertoe worden jaarlijks twee thema's vastgesteld. Lezingen over deze thema's worden normaliter verzorgd door uitgenodigde buitenlandse en binnenlandse deskundigen.

Thema's

1. *Parallel rekenen (met bijzondere aandacht voor de thans gangbare computer architekturen).*

2 *Aspecten van de eindige elementen methode (met bijzondere aandacht voor gemengde eindige elementen).*

Organisatie

De organisatie is in handen van de voorbereidingscommissie bestaande uit A. van der Sluis (voorzitter), M.H.C. Paardekooper (KUB), M.N. Spijker (RUL) en W.H. Hundsdorfer (CWI) (secretaris). Ondersteuning is verleend door L. Verdonk (CWI) en het Centrum voor Wiskunde en Informatica.

Sprekers

Thema 1. R. Hockney, Reading, U.K.
S. McCormick, University of Colorado at Denver, U.S.A.
H.A. van der Vorst, Technische Universiteit Delft.

Thema 2. L.D. Marini, CNR, Pavia, Italië.
R. Verfürth, Universität Zurich, Zwitserland.
M.F. Wheeler, Rice University, Houston, U.S.A.

Korte voordrachten zullen worden gegeven door:

E.F. Kaasschieter, Dienst Grondwaterverkenning TNO, Delft (thema 2)
P.H. Michielse, Technische Universiteit Delft (thema 1)
G. Mur, Technische Universiteit Delft (thema 2)
H.J. Stam, Technische Universiteit Delft (thema 2).

Programma

Maandag 2 oktober

10.00 - 11.15	aankomst, koffie	14.15 - 15.15	Marini
11.15 - 12.15	opening, Hockney	15.15 - 15.45	thee
12.30	lunch	15.45 - 16.45	McCormick
		16.50 - 17.20	Kaasschieter
		18.00	diner

Dinsdag 3 oktober

8.00	ontbijt	14.15 - 15.15	Hockney
9.00	Verfürth	15.15 - 15.45	thee
10.00 - 10.30	koffie	15.45 - 16.45	Marini
10.30 - 11.30	van der Vorst	16.50 - 17.20	Michielse
11.35 - 12.35	Wheeler	17.20 - 17.30	VergaderingWerkgemeenschap Numerieke Wiskunde
12.45	lunch		
		18.00	diner

Woensdag 4 oktober

8.00	ontbijt	13.45 - 14.45	van der Vorst
9.00 - 10.00	McCormick	14.50 - 15.50	Wheeler
10.00 - 10.30	koffie	15.50	sluiting, thee, vertrek
10.30 - 11.00	Stam		
11.00 - 11.20	Mur		
11.25 - 12.25	Verfürth		
12.35	lunch		

De bar is geopend van 17.00-18.00 uur en van 21.00-24.00 uur.

Titels en Samenvattingen Voordrachten

Maandag 2 oktober

11.15	R. Hockney:	*Current parallel computers, I.*
14.15	L.D. Marini:	*Mixed approximations of second order elliptic problems.*
15.45	S. McCormick:	*Asynchronous multilevel adaptive methods on multiprocessors, I.*
16.50	E.F. Kaasschieter:	*Mixed-hybrid finite elements and streamline computation for the potential flow problem.*

Dinsdag 3 oktober

9.00	R. Verfürth:	*The design of stable mixed finite element schemes for flow problems.*
10.30	H.A. van der Vorst:	*Parallel algorithms for large linear systems, I.*
11.35	M.F. Wheeler:	*Mixed finite elements for porous media problems*
14.15	R. Hockney:	*Performance of parallel computers and algorithms.*
15.45	L.D. Marini:	*Mixed exponential fitting schemes for current continuity equations.*
16.50	P. Michielse:	*Parallel multigrid in adaptive reservoir simulation.*

Woensdag 4 oktober

9.00	S. McCormick:	*Asynchronous multilevel adaptive methods on multiprocessors, II.*
10.30	H.J. Stam:	*Theoretical considerations on a finite-element method for the computation of three-dimensional space-time acoustic wave fields.*
11.00	G. Mur:	*Mixed and irreducible formulations for three-dimensional electromagnetic fields in strongly inhomogeneous media.*

12.25	R. Verfürth:	*Aposteriori error estimators and adaptive mesh-refinement for flow problems.*
13.45	H.A. van der Vorst:	*Parallel algorithms for large linear systems, II.*
14.50	M.F. Wheeler:	*Domain decomposition, mixed finite element methods and multi-level procedures.*

Deelnemers

1.	Aarden, J.M.C.	Katholieke Universiteit	Nijmegen
2.	Axelsson, A.O.H.	Katholieke Universiteit	Nijmegen
3.	Bosch, P. van den	Waterloopkundig Lab.	Emmeloord
4.	Botta, E.F.F.	Rijksuniversiteit	Groningen
5.	McCormick, S.	University of Colorado	Denver (U.S.A.)
6.	Dekker, K.	Technische Universiteit	Delft
7.	Dekker, Th.J.	Universiteit van	Amsterdam
8.	Dijke, M.C.A. van	Technische Universiteit	Delft
9.	Dijkzeul, J.C.M.	Rijkswaterstaat, D.I.V.	Rijswijk
10.	Eijkeren, J.C.H. van	RIVM	Bilthoven
11.	Eijkhout, V	Katholieke Universiteit	Nijmegen
12.	Gerritsen, M.G.	Technische Universiteit	Delft
13.	Gmelig Meyling, R.H.J.	KSEPL/Shell	Rijswijk
14.	Goede, E.D. de	CWI	Amsterdam
15.	Griend, J.A. van de	Rijksuniversiteit	Leiden
16.	Groen, P. de	Vrije Universiteit	Brussel (B)
17.	Groot, Y. de	Nat. Lab. Philips	Eindhoven
18.	Hemker, P.W.	CWI	Amsterdam
19.	Hockney, R.	Reading	U.K.
20.	Hoogwerf, M.	Katholieke Universiteit	Nijmegen
21.	Hout, K.J. in 't	Rijksuniversiteit	Leiden
22.	Houwen, P.J. van der	CWI	Amsterdam
23.	Hundsdorfer, W.H.	CWI	Amsterdam
24.	Jansen, J.K.M.	Technische Universiteit	Eindhoven
25.	Kaasschieter, E.F.	Dienst Grondwaterverk. TNO	Delft
26.	Koren, B.	CWI	Amsterdam
27.	Kraaijevanger, J.F.B.M.	Rijksuniversiteit	Leiden
28.	Kramer, M.E.	Technische Universiteit	Eindhoven
29.	Lioen, W.	CWI	Amsterdam
30.	Louter-Nool, M.	CWI	Amsterdam
31.	Maarel, H.T.M. van der	CWI	Amsterdam
32.	Marini, L.D.	CNR	Pavia (Italië)
33.	Maubach, J.M.L.	Katholieke Universiteit	Nijmegen
34.	Michielse, P.H.	Technische Universiteit	Delft
35.	Mol, W.J.A.	Waterloopkundig Lab.	Emmeloord
36.	Molenaar, J.	CWI	Amsterdam
37.	Mur, G.	Technische Universiteit	Delft
38.	Nooyen, R.R.P. van	CWI	Amsterdam
39.	Oosterlee, C.W.	Technische Universiteit	Delft
40.	Paardekooper, M.H.C.	Katholieke Universiteit Brabant	Tilburg
41.	Piers, W.J.	NLR	Amsterdam
42.	Ploeg, A. v.d.	Rijksuniversiteit	Groningen
43.	Polak, S.J.	Ned. Philips Bedrijven B.V.	Eindhoven
44.	Reusken, A.	Technische Universiteit	Eindhoven
45.	Riele, H.J.J. te	CWI	Amsterdam
46.	Rusch, J.J.	Nat. Lab. Philips	Eindhoven
47.	Rijk, P.P.M. de	Ac. Computer Centrum	Utrecht
48.	Schmidt, G.H.	KSEPL/Shell	Rijswijk

49.	Schulkes, R.M.S.M.	Technische Universiteit	Delft
50.	Sluis, A. van der	Rijksuniversiteit	Utrecht
51.	Sommeijer, B.P.	CWI	Amsterdam
52.	Spekreijse, S.P.	NLR	Emmeloord
53.	Spijker, M.N.	Rijksuniversiteit	Leiden
54.	Stam, H.J.	Technische Universiteit	Delft
55.	Steen, A.J. van der	Ac. Computer Centrum	Utrecht
56.	Stevenson, R.P.	Rijksuniversiteit	Utrecht
57.	Stijn, Th.L. van	RWS/Dienst Informatieverwerk.	Rijswijk
58.	Timmermans, L.J.P.	Technische Universiteit	Eindhoven
59.	Traas, C.R.	Universiteit Twente	Enschede
60.	Trompert, R.A.	CWI	Amsterdam
61.	Veldhuizen, M. van	Vrije Universiteit	Amsterdam
62.	Verfürth, R.	Universiteit Zürich	Zürich (Zwitserland)
63.	Verwer, J.G.	CWI	Amsterdam
64.	Vorst, H.A. van der	Technische Universiteit	Delft
65.	Vuik, C.	Technische Universiteit	Delft
66.	Wesseling, P.	Technische Universiteit	Delft
67.	Wheeler, M.F.	Rice University	Houston (U.S.A.)
68.	Wiel, M.C.J.	Nat. Lab. Philips	Eindhoven
69.	Wubs, F.W.	Rijksuniversiteit	Groningen
70.	Wuytack, L.	Universiteit Antwerpen	Wilrijk (B)
71.	Zeeuw, P.M. de	CWI	Amsterdam
72.	Zegeling, P.A.	CWI	Amsterdam

CURRENT PARALLEL COMPUTERS

R. Hockney
23 Hillside, Hardwick Road
Whitchurch on Thames
Reading RG8 7HL, U.K.

In contrast to traditional serial computation the timing characteristics of parallel computers are quite complex, and consequentlty it is necessary for numerical analysts to have some understanding of the design (or architecture) of the computers that they use. To this end this first lecture is a descriptive survey of the currently available parallel computers that numerical analysts may find that they are using. Before the advent of parallel computers it was not necessary for the numerical analyst to know anything about the architecture of the computers that they used, indeed many took the cavalier view that such matters were beneath their concern. However the performance of algorithms now depends so much on a good match between the algorithm and the architecture of the computer on which it is used, that such a view is no longer tenable. In order to be able to relate the performance of an algorithm to the architecture of the computer, it is necessary to use some relatively simple timing equations that characterise the behaviour of the computer hardware. This is the topic of the second lecture entitled "Performance of Parallel Computers and Algorithms". The characterisation introduced deliberately uses the minimum of parameters, and leads to a method of parallel algorithm analysis which can replace that traditionally used for judging serial algorithms. Most of the material presented in the two lectures is given in "Parallel Computers 2" by Hockney Jesshope (second edition 1988; Adam Hilger, Bristol, UK: ISBN 0-85274-812-4) which is available at a reasonable price in paperback. This book is the recommended seminar reading material (references $n.n, and Chapter references to PC2, made below, refer to sections and chapters in this book).

The computer architectures described include both SIMD and MIMD architectures with, respectively, a single or multiple instruction streams that may be programmed by the user. Both of these may be implemented by pipelining or replication. The pipelined SIMD computers include most of the well known supercomputers (e.g. the Cray X-MP, Y-MP, Cray-2, Cyber 205 and ETA-10, IBM 3090-VF, and the Japanese computers from Fujitsu, Hitachi and NEC) which are often referred to as vector computers (see Chapter 2 of PC2). The similarity and differences between these designs will be pointed out. Replicated SIMD include the best known of the massively parallel systems that have recently had a wide exposure, namely the Distributed Array Processor (or DAP) from AMT, and the Connection Machine from Thinking Machines (see Chapter 3 of PC2). These two types of SIMD computer have widely different timing characteristics, and must be programmed in quite different ways with often quite different algorithms (see Chapter 5 of PC2).

The numerical analyst is today presented with a bewildering variety of MIMD computers to consider. The first commercial MIMD computer, the Denelcor HEP, employed the pipelining of many instruction streams through a single instruction unit. This machine will be described both to show that pipelined MIMD is a practical option, and because these ideas are likely to reappear in future machines from the same designer (Burton Smith of Tera Computer with the Horizon computer). However the HEP was not a commercial success, and the currently available commercial MIMD computers fall into two types: modestly parallel shared memory systems, and massively parallel distributed memory networks. In shared memory systems a relatively small number of processors (say up to 50, but usually between 2 and 8) are connected to a large common memory by a switch. The most common switch is a shared bus, but multi-level switches based on the Omega switch appear in many of the experimental designs presently nearing completion (e.g. the U. Illinois Cedar machine).

From the numerical analysts and programmers point of view, the main point to appreciate about shared memory systems is that, as the name implies, all the processors share the same common memory address space, both physically and logically. This raises questions of protecting the integrity of data from unintended alteration by other processors, which fall generally into the realm of correct parallel program synchronisation. These are issues that were not present in serial computation, and greatly increase the difficulty of parallel programming. Nevertheless programming within the shared memory model of computation, seems to be the simplest parallel step to take for those familiar with serial computation. Partly for this reason shared memory systems are currently the most saleable to the user community, and the most likely to be found. Furthermore, parallelising compilers exist for such computers that can take an existing serial program (usually referred to, for historical reasons, as a 'dusty deck') and run it on the shared memory parallel system, without any intervention from the programmer. This is usually achieved by distributing the iterations of Fortran DO-loops within the program amongst the multiple processors. Thus numerical analysts are able to run their existing serial codes on such machines with some speed gain obtained from the multiple processors. However, the timing characteristics of a DO-loop spread over many processors is generally speaking much greater than the overhead of a DO-loop executed on a single processor, so that the existing algorithm may not be the best one to use. But, again speaking generally, it is likely that algorithms developed for serial computers will perform acceptably well on shared memory systems with a modest number of processors.

The second important class of MIMD computer is the network of processors in which the whole memory of the system is distributed, usually equally, as local memory to a large number (usually hundreds or thousands) of processors. There is now no global common address space, as each processor can only address directly the portion of the total memory that it has as its own local memory. If one processor requires data from another then the data must be transferred across the network from processor to processor as a message, and the communicating processors must be programmed to receive and acknowledge such messages. The most common example of such a Distributed Memory Network is the Meiko Computing Surface in which up to about 100 INMOS transputers are connected in a network whose connections can be changed electronically under program control. The INMOS transputer itself is designed to be a node in these distributed memory networks, and contains a floating-point microprocessor (T800), 4K Bytes of RAM, and four I/O links to other processors in the network, all on a single chip (see Chapter 3 in PC2). Thus any four connected (or in chemical terms four valent) network can be assembled by connecting together directly the pins of a number of Transputer chips. Of course, usually much more memory is required at each node, and several MBytes of off-chip memory is currently usually provided. The great attraction of such networks is that they can, in principle, be extended to a very large size containing many thousands of processors, whereas the shared memory bus connected system cannot extend above a few tens of processors because the bus would saturate. For this reason the use of a distributed network usually implies that one is using a massively parallel system of many thousands of processors, and the numerical analyst needs to think of ways of problem decomposition that can make effective use of all these processors.

A particularly interesting example of a reconfigurable distributed memory network is the PARSYS Supernode that can expand to 1024 Transputers, using several levels of switching. In this system the Transputers can be connected under program control in the way demanded by the data relationships in the algorithm to be evaluated (a feature that was never available in serial computers), and the data pipelined through this hardwired algorithm. In addition the problem to be solved must be partitioned (or decomposed) into separate problems to be computed by each processor of the network: the key to a successful decomposition being that the communication required between different nodes of the network is kept to a minimum. The reason for this is that the time required to send a number between processors on the network is much longer than the time taken to perform arithmetic on that number when it is received. Thus minimising communication is usually the key criterion when optimising algorithms on distibuted memory networks, rather than minimising arithmetic which is the criterion used in serial computation. It goes now without saying that programming such networks requires quite different thought processes than those used in serial computation, and most programmers with serial experience find this very difficult at first. Indeed it is probably true that it is easier for programmers with little previous experience to user such systems, than people with a lot of previous experience with serial computation. Thus most current use of such systems is thus by graduate students in Universities or small startup companies producing turnkey systems for specialist markets. Currently there are no parallelising compilers capable of converting dusty decks into code running tolerably efficiently on such massively parallel networks. At the moment, these systems are definitely only suitable in situations where it is practicable and worth while to reprogram and reanalyse a problem from scratch, or when entirely new work is being started. At present this fact severely limits the attraction of distributed memory networks. It is therefore the main challenge to the coming generation of numerical analysts to solve these problems, and there are many who believe that massively parallel networks will become the pattern for computing in the future.

PERFORMANCE OF PARALLEL COMPUTERS AND ALGORITHMS

R. Hockney
23 Hillside, Hardwick Road
Whitchurch on Thames
Reading RG8 7HL, U.K.

The principal difference between the timing behaviour of a pipelined vector computer and a serial computer, is that the performance depends on the vector length of the vector instruction being executed, whereas the performance of the equivalent serial DO-loop is approximately a constant independent of the loop length. This effect can be quantified by defining the half-performance length, $n_{\frac{1}{2}}$, which is the vector length required to achieve half of the maximum performance, r_∞. The performance for other vector lengths is given by (see §1.3.5)

$$r = r_\infty \; / \; (1 + n_{\frac{1}{2}} / n)$$

Values for the parameter pair (r_∞ , $n_{\frac{1}{2}}$) will be given for a number of computers. When the value of $n_{\frac{1}{2}}$ exceeds 10, the degradation of performance due to inadequate vector length becomes important in many problems, and disappointing performance may be seen on vector supercomputers. In this case new algorithms must be devised that use longer vectors. Unfortunately it is much easier to increase the arithmetic speed of a processor than to reduce its $n_{\frac{1}{2}}$, and the latest generation of vector supercomputers tend to have values of $n_{\frac{1}{2}}$ in the hundreds, due to the greater complexity of memory organisation.

The half-performance length is a measure of the startup overhead of a vector instruction in terms of a currency known to an algorithm designer, namely the vector lengths occurring in his algorithm. A method of algorithm analysis is derived, "The $n_{\frac{1}{2}}$ method of vector algorithm analysis", that derives the performance of a vector algorithm in terms of the average vector length in the algorithm and the half-performance length of the computer (see §5.1.6, and throughout Chapter 5 in PC2).

Parallel computation on a multi-processor computer introduces new problems and overheads that are not present in traditional serial computation. These are scheduling, synchronisation and communication which may be defined thus:

(1) Scheduling - the decomposition of the algorithm into roughly equal parts for execution by each processor in the system.

(2) Communication - the transfer of data between processors so that data is in the right place for future computation.

(3) Synchronisation - the control necessary to ensure that the computation takes place in the correct time order.

Scheduling may be characterised by the efficiency parameter E_p which is 100 percent if the scheduling is perfect, and all processors have exactly the same amount of work to perform. In this case if all processors start executing at the same time, they will all finish together, and none will become idle waiting for others to finish. If the scheduling (or load balancing) is imperfect, some processors have less work to do than others and must become idle waiting for others to finish. In this case the efficiency is less than 100 percent.

A simple model of a parallel program is as a sequence of parallel sections separated by synchronisation points, which have the property that all work must finish in one parallel section before any work can start in the next. Even if the scheduling is perfect and there are no idle processors, it still takes time to perform the above synchronisation. This overhead may be measured by asking how much arithmetic could have been done during the time it took to perform the actions required at the synchronisation point. This number (measured in floating-point operations) is called $s_{\frac{1}{2}}$ or the half-performance grain size. As the name implies, it is the number of floating-point operation in a parallel section (called the grain sizes of the section) that are necessary to achieve half of the maximum performation. When the grain size equals $s_{\frac{1}{2}}$, half the time is spent performing useful work and half the time is spent on synchronisation. Clearly efficient calculation only takes place if the grain size is greater than $s_{\frac{1}{2}}$, and it is therefore an important number for the numerical analyst to know about the computer he is using. Methods of measurement will be described and values given for some well known computers (e.g. Cray-2 Cray X-MP, IBM 3090, see §1.3.6 in PC2).

The communication overhead is determined by the ratio of arithmetic speed in the processor (measured in Mflop/s) to the communication bandwidth between processors (measured in Mword/s). If this ratio is low then the overhead for communication is small, whereas if it is high then the communication overheads are likely to dominate any calculation. We have developed a benchmark to measure this effect which transfers 'mref' words into a processor, and performs 'flop' floating point operations upon each word transferred. The ratio flop/mref is called the computational intensity, f, and we determine the value of f required to achieve half of the peak speed of the arithmetic unit ($\hat{r}_\infty$). This value is called the half-performance intensity, or $f_{\frac{1}{2}}$. The average rate f computation is then fitted to:

$$r = \hat{r}_\infty \; / \; (1 + f_{\frac{1}{2}} \; / \; f \;)$$

Values of $f_{\frac{1}{2}}$ are given for a number of computers, and it is noticeable that these are significantly larger for network computers compared to vector supercomputers. This shows that reducing communication is the main problem facing algorithm designers using current network computers.

MIXED-HYBRID FINITE ELEMENTS AND STREAMLINE COMPUTATION FOR THE POTENTIAL FLOW PROBLEM

E.F. Kaasschieter
Dienst Grondwaterverkenning TNO
Postbus 285
2600 AG Delft

An important class of problems in mathematical physics involves equations of the form

(1a) $\mathbf{u} = -\mathbf{A}\,\nabla\phi,$

(1b) $\nabla\cdot\mathbf{u} = f,$

where $\mathbf{A}$ is a symmetric and uniformly positive definite second rank tensor. The equations (1) are fundamental in the theory of heat conduction, electrostatics and groundwater hydraulics.
For instance, in modelling the flow of an incompressible fluid in a porous medium, the piezometric head (potential) ϕ and the specific discharge (Darcy velocity) $\mathbf{u}$ are related by Darcy's law (equation (1a)), where $\mathbf{A}$ is the tensor of hydraulic conductivity (permeability), and $\mathbf{u}$ has to fulfill the continuity equation (equation (1b)). The function f is used to represent sources and sinks.
An accurate approximation of the specific discharge is crucial in the numerical solution of a variety of groundwater flow problems. In approximating $\mathbf{u}$ from (1) with standard finite difference or finite element techniques, ϕ is first determined as a set of cell averages, nodal values or piecewise smooth functions. This ϕ is then differenced or differentiated and multiplied by an often rough tensor $\mathbf{A}$ to obtain $\mathbf{u}$. In many cases an inaccurate specific discharge results from this approach, i.e. the thus obtained approximation of $\mathbf{u}$ does not necessarily fulfill the continuity equation rather accurately.
From physical considerations it is wishful to obtain an approximation of $\mathbf{u}$, that fulfills (1b) as good as possible with

respect to the finite difference grid or finite element mesh. Such an accurate approximation can be determined by the mixed finite element method according to Raviart, Thomas and Nedelec, that will be discussed in this lecture for general two- and three-dimensional problems. This lecture will be limited to the lowest order mixed method, because, firstly, higher order methods result in some conceptual complications and, secondly, because the lowest order method is rather easy and straightforward to use for practical problems.

The mixed finite element method results in a large system of linear equations. The choice of a numerical method to solve this system is restricted by the fact that its coefficient matrix is indefinite. This drawback can be circumvented by an implementation technique, called hybridization, which leads to a symmetric positive definite system of linear equations. Because this system is sparse, it can be solved efficiently by the preconditioned conjugate gradient method.

Finally it will be indicated how streamlines and residence times induced by the specific discharge field can be computed very efficiently and without additional numerical errors.

MIXED APPROXIMATIONS OF SECOND ORDER ELLIPTIC PROBLEMS

Louisa Donatella Marini
Instituto di Analisi Numerica del CNR
Corso C. Alberto 5
27100 Pavia, Italy.

Consider the following boundary value problem:

$$(1) \qquad \begin{cases} -\, div(a\underline{\nabla} u) \;+\; \nu\, u = f & in \quad \Omega \\ \qquad\qquad\qquad\quad u = 0 & on \quad \Gamma \;=\; \partial\Omega \end{cases}$$

where Ω is a convex polygonal domain in $\mathbf{R}^2$, $f \in L^2(\Omega)$, $a = a(x) \geq a_0 > 0$ is a smooth function on the closure of Ω, and ν is a non negative constant. When dealing with problems of the type (1), often the most important physical quantity is not u itself, but the vector field $\underline{\sigma} = -a(x)\underline{\nabla} u$. If a classical discretization method is used for (1), the vector field can be recovered from u. However, its computation requires the derivatives of u which can be, from the numerical point of view, a loss of precision. If instead non-standard finite element discretizations are used (mixed or hybrid), the vector field is introduced, and then approximated, as an independent variable. Let us recall the mixed formulation of problem (1). Define the spaces

$$(2) \qquad V \;=\; H(div;\Omega) = \{\underline{\tau} \in [L^2(\Omega)]^2 \mid div\ \underline{\tau} \in L^2(\Omega)\},$$

$$(3) \qquad W \;=\; L^2(\Omega),$$

with the norms

$$(4) \qquad \|\underline{\tau}\|_V^2 \;=\; \|\underline{\tau}\|_{L^2(\Omega)}^2 \;+\; \|div\ \underline{\tau}\|_{L^2(\Omega)}^2 \;,\quad \|\phi\|_W \;=\; \|\phi\|_{L^2(\Omega)} \;,$$

and the bilinear forms

$$(1.5) \qquad a(\underline{\sigma},\underline{\tau}) \;=\; \int_\Omega a^{-1}\underline{\sigma}\cdot\underline{\tau}\; dx \;, \qquad \underline{\sigma},\ \underline{\tau} \in V,$$

$$(6) \qquad b(\underline{\tau},\phi) \;=\; \int_\Omega div\ \underline{\tau}\ \phi\; dx \;, \qquad \underline{\tau} \in V,\ \ \phi \in W.$$

The usual mixed formulation of (1) is:

$$(7) \qquad \begin{cases} find\ (\underline{\sigma},\ u) \in V \times W\ such\ that \\ a(\underline{\sigma},\underline{\tau}) \;-\; b(\underline{\tau},u) \;=\; 0 & \forall \underline{\tau} \in V \\ b(\underline{\sigma},\phi) \;+\; \nu(u,\phi) \;=\; (f,\phi) & \forall \phi \in W \;. \end{cases}$$

where $(.,.)$ is the scalar product in $L^2(\Omega)$. Existence and uniqueness of the solution of (7) follow from the abstract theory of [2]. When going to the approximation of (7), one has to choose finite dimensional spaces $W_h \subset W$, $V_h \subset V$ properly related in order to guarantee uniqueness of the discrete solution and optimal error bounds [2]. While the requirement $W_h \subset W$ does not represent any problem, the inclusion $V_h \subset V$ implies some regularity on the elements of V_h: more precisely, the normal components of a vector in V_h must be continuous across the interelement boundaries. In the literature there are four families of mixed finite elements [3], [13], [8] satisfying all the requirements and giving optimal error bounds. The key properties of these families are the internal approximation property (i.e. $W_h \subset W, V_h \subset V$) and the commutativity of the *divergence* operator with suitably defined "projections" from the continuous spaces onto the discrete ones [8], [9], [3], [6]. These families of discretizations lead to a linear algebraic system whose matrix is not positive definite. A way to circumvent this inconvenience is the use of Lagrange multipliers. The idea, introduced by Fraeijis de Veubeke [10], is to relax the continuity requirement on the normal components and to enforce it back via the use of Lagrange multipliers. This leads to the introduction of a third discrete space, say Λ_h, properly related to W_h and V_h (see e.g. [1],[3],[6]). Using this procedure, both scalar and vector variables can be eliminated element by element by static condensation with virtually no cost, and the problem reduces to the solution of a linear system in the Lagrange multipliers only. This Lagrange multiplier formulation, often called hybridization of the mixed formulation, leads to a final system whose matrix is symmetric and positive definite.

In many applications, such as in the drift-diffusion models for semiconductor device simulation, a discrete maximum principle is desirable (i.e., the final matrix should be an M-matrix). None of the elements of the four known families has this property for $\nu > 0$, and only the lowest-order Raviart-Thomas triangular element produces an M-matrix for $\nu = 0$. For that, two new elements have been recently introduced and discussed in [11],[12]. They have been proved to produce the M-matrix property for all non negative ν, if the triangulation is of weakly acute type (Every angle of every triangle is $\leq \frac{\pi}{2}$).

References

[1] D.N.Arnold - F.Brezzi: Mixed and non-conformir 'inite element methods: implementation, post-processing and error estimates. M^2AN **19** , 7-32, 1985.

[2] F.Brezzi: On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. R.A.I.R.O., **8-R2**, 129-151, 1974.

[3] F.Brezzi - J.Douglas,Jr. - L.D.Marini: Two families of mixed finite elements for second order elliptic problems. Numer. Math., **47**, 217-235, 1985.

[4] F.Brezzi - L.D.Marini - P.Pietra: Two-dimensional exponential fitting and applications to drift-diffusion models. (To appear in SIAM J.Numer.Anal.)

[5] F.Brezzi - L.D.Marini - P.Pietra: Numerical simulation of semiconductor devices. (To appear in Comput.Meths.Appl.Mech. and Engr.)

[6] F.Brezzi - M.Fortin: Mixed and hybrid finite element methods. Book to appear.

[7] P.G.Ciarlet: *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.

[8] J.Douglas,Jr. - J.E.Roberts: Mixed finite element methods for second order elliptic problems. Mat. Apl. e Comput., **1**, 91-103, 1982.

[9] J.Douglas,Jr. - J.E.Roberts: Global estimates for mixed methods for second order elliptic equations. Math. Comp., **44**, 39-52, 1985.

[10] B.X. Fraeijs de Veubeke: Displacement and equilibrium models in the finite element method. *Stress Analysis*, O.C.Zienkiewicz - G.Hollister eds, New York, 1965.

[11] L.D.Marini - P.Pietra: An abstract theory for mixed approximations of second order elliptic problems. (To appear in Matem. Aplic. e Comput.).

[12] L.D.Marini - P.Pietra: New mixed finite element schemes for current continuity equations. (Submitted to COMPEL).

[13] P.A.Raviart - J.M.Thomas: A mixed finite element method for second order elliptic problems. In *Mathematical aspects of the finite element method.* Lecture Notes in Math. **606**, 292-315, Springer, Berlin, 1977.

MIXED EXPONENTIAL FITTING SCHEMES FOR CURRENT CONTINUITY EQUATIONS

Luisa Donatella Marini
Instituto di Analisi Numerica del CNR
Corso C. Alberto 5
27100 Pavia, Italy

The aim of this presentation is to summarize the results recently obtained for the numerical treatment of the current continuity equations arising in semiconductor device simulation problems. For simplicity, we consider the current continuity equation for positive charge densities only. After a suitable scaling [7] this equation has the form

$$\begin{cases} \textit{Find } p \in H^1(\Omega) \textit{ such that} & & \\ -div(\underline{\nabla} p + p\underline{\nabla}\psi) = R(p,n,\underline{\nabla}\psi) & \textit{in } \Omega \subset R^2 & \\ p = g & \textit{on } \Gamma_0 \subset \partial\Omega & \\ \dfrac{\partial p}{\partial n} + p\dfrac{\partial \psi}{\partial n} = 0 & \textit{on } \Gamma_1 = \partial\Omega \backslash \Gamma_0 & \end{cases} \tag{1}$$

where p and n are positive and negative charge densities respectively, ψ is the electrical potential, R is the recombination-generation term, and the current density $\underline{J}$ is given by

$$\underline{J} = -\underline{\nabla} p - p\underline{\nabla}\psi. \tag{2}$$

In most applications, $|\underline{\nabla}\psi|$ can be quite large in some parts of Ω. Hence, (1) becomes an advection dominated equation, and classical discretization schemes do not work. Moreover, since the current is the most important physical quantity of the problem, ad-hoc methods which are current preserving are needed. In the 1 dimensional case, the Scharfetter-Gummel scheme [14] is very effective, both with respect to the accuracy in the computation of the approximate p and with respect to the current preserving property. Many attempts in generalizing the Scharfetter-Gummel scheme to 2-d problems have been done. We recall for instance [10],[11], [1], [8], [4], [9]. Here we describe some schemes, recently introduced and analyzed in [2],[3],[5],[6], which are based on the mixed exponential fitting. These methods reproduce the Scharfetter-Gummel scheme when applied to the 1-D case, and, moreover, are current preserving. Here we describe 2-dimensional schemes, which are based on the mixed exponential fitting. These

methods reproduce the Scharfetter-Gummel scheme when applied to the 1-D case, and, moreover, are current preserving.

We shall deal with discretizations of (1) assuming that ψ and n are known. Moreover, ψ is assumed piecewice linear. Using the classical change of variable from the charge density p to the Slotboom variable ρ

$$p = \rho e^{-\psi} , \tag{3}$$

equation (1) can be written in the symmetric form

$$\begin{cases} \textit{Find } \rho \in H^1(\Omega) \textit{ such that} & \\ -div(e^{-\psi}\underline{\nabla}\rho) = R(\rho e^{-\psi}, n, \underline{\nabla}\psi) & \textit{in } \Omega \\ \rho = \chi := e^{\psi} g & \textit{on } \Gamma_0 \\ \dfrac{\partial \rho}{\partial n} = 0 & \textit{on } \Gamma_1 \ , \end{cases} \tag{4}$$

and the current is given by

$$\underline{J} = -e^{-\psi}\underline{\nabla}\rho . \tag{5}$$

During the iterative solution process of the complete system, equation (4) is usually linearized in such a way that (4) becomes [7],[12]

$$\begin{cases} \textit{Find } \rho \in H^1(\Omega) \textit{ such that} & \\ -div(e^{-\psi}\underline{\nabla}\rho) + c\rho = f & \textit{in } \Omega \\ \rho = \chi := e^{\psi} g & \textit{on } \Gamma_0 \\ \dfrac{\partial \rho}{\partial n} = 0 & \textit{on } \Gamma_1 \end{cases} \tag{6}$$

The idea is to discretize equation (6) with mixed finite element methods, go back to the original variable p by using a discrete version of the transformation (3), and then solve for p.

From the algebraic point of view, this class of schemes produces, in the unknown ρ, a symmetric positive definite matrix. When the inverse transformation (3) is applied for going back to the original unknown p, the corresponding matrix,in general, is not symmetric anymore (nor, in general, positive definite). The only structure property of the matrix which could be preserved when going from the unknown ρ to the unknown

p is the M-matrix property. Moreover, this property guarantees a discrete maximum principle and, in particular, a non-negative solution if the boundary data are non-negative. Then, when the entire van Roosbroeck system is solved, instabilities are not introduced. The main feature of mixed finite element schemes is to treat the current $\underline{J}$ as an independent variable. This fact allows on one hand to derive optimal error bounds for the current, and on the other hand to guarantee current preservation. When $c = 0$, the lowest order Raviart-Thomas element provides an M-matrix if the decomposition is of weakly acute type. Unfortunately, this is not the case for $c > 0$. For that, two new elements have been recently introduced in [5],[6]. These elements guarantee the M-matrix property for all $c \geq 0$, under the usual assumption of a triangulation of weakly acute type.

References

[1] R.E.Bank - D.J.Rose - W.Fichtner: Numerical methods for semiconductor device simulation. *IEEE Trans. El. Dev.*, **30** (9), 1031-1041, 1983.

[2] F.Brezzi - L.D.Marini - P.Pietra: Two-dimensional exponential fitting and applications to drift-diffusion models. (To appear in SIAM J.Numer.Anal.)

[3] F.Brezzi - L.D.Marini - P.Pietra: Numerical simulation of semiconductor devices. (To appear in Comput.Meths.Appl.Mech. and Engr.)

[4] J.F.Burgler - R.E.Bank - W.Fichtner - R.Kent Smith: A New Discretization Scheme for the Semiconductor Current Continuity Equations. (To appear).

[5] L.D.Marini - P.Pietra: An abstract theory for mixed approximations of second order elliptic problems. (To appear in Matem. Aplic. e Comput.).

[6] L.D.Marini - P.Pietra: New mixed finite element schemes for current continuity equations. (Submitted to COMPEL).

[7] P.A.Markowich: *The Stationary Semiconductor Device Equations.* Springer, 1986.

[8] P.A.Markowich - M.Zlamal: Inverse-average-type finite element discretizations of self-adjoint second order elliptic problems. *Math. of Comp.*, **51**, 431-449, 1988.

[9] J.J.H.Miller - S.Wang - C.H.Wu: A mixed finite element method for the stationary semiconductor continuity equations. *Engineering Computations*, **5**, 285-288, 1988.

[10] M.S.Mock: *Analysis of Mathematical Models of Semiconductor Devices*, Dublin, Boole Press, 1983.

[11] M.S.Mock: Analysis of a discretisation algorithm for stationary continuity equations in semiconductor device models II. *COMPEL* **3**, 137-149, 1984.

[12] S.J.Polak - C.Den Heijer - W.H.A.Schilders - P.A.Markowich: Semiconductor device modelling from the numerical point of view, Int. J. Numer. Methods Eng. **24**, 763-838, 1987.

[13] P.A.Raviart - J.M.Thomas: A mixed finite element method for second order elliptic problems. In *Mathematical aspects of the finite element method.* Lecture Notes in Math. **606**, 292-315, Springer, Berlin, 1977.

[14] D.Scharfetter - H.Gummel: Large-signal analysis of a silicon Read diode oscillator. *IEEE Trans.El.Dev.*, **ED-16**, 64-77, 1969.

ASYNCHRONOUS MULTILEVEL ADAPTIVE METHODS ON MULTIPROCESSORS

S. Mc Cormick
Computational Mathematics Group
Campus box 170, University of Colorado at Denver
1100 14th Street
Denver, Colorado 80202, USA

1. Introduction.

The recently developed fast adaptive composite grid method, FAC [1,2], is a discretization and solution method for PDEs designed to achieve efficient local resolution by systematically constructing the discretization based on various regular grids and using them as a basis for fast solution. Using multigrid as the individual grid solver, FAC has been applied to a variety of fluid flow problems, including incompressible Navier-Stokes equations [3], in both two and three dimensions. A more recently developed variation of FAC designed for parallel computers, AFAC [4], allows for independent processing of the individual uniform grids that constitute the composite grid. This means that the cost of solving the adapted equations may be fully shared by independent processing units.

Because of the way local grids add computational load irregularly to processors, an efficient load balancer is an essential ingredient for implementing adaptive methods on distributed memory machines. The complexity of this process, as well as the overall algorithm itself, depends intimately on how the processors are assigned to the computational tasks. The independence of the various refinement levels in the AFAC process allows these assignments to be made by level (in contrast to the usual domain decomposition approach). This greatly simplifies the load balancing method (MLB) since the levels may be ordered in a list and the partition of such a list is inherently one-dimensional.

An important aspect of adaptive refinement is the discretization process. An especially effective technique is the finite volume element method, FVE [3], that effectively combines finite elements (FE) and finite volumes (FV). It is especially convenient for use with AFAC for fluid flow applications.

This talk is in two parts. The first part is an introduction to AFAC, MLB and FVE, with special emphasis on performance results on an iPSC/1 and iPSC/2. The second part will develop theoretical foundations for these techniques and report on their application to fluid flow simulation.

2. AFAC Algorithm.

The asynchronous fast adaptive composite grid method (AFAC [4]) is a multilevel method for adaptive grid discretization and solution of partial differential equations. AFAC appears to have optimal complexity in a parallel computing environment because it allows for simultaneous processing of all levels of refinement. Coupled with multigrid (MG) processing of each level and nested iteration on the composite grids, AFAC is usually able to solve the composite grid equations in a time proportional to what it would take to solve the global grid alone. See Hart and McCormick [4] for further details.

A basic code for AFAC was developed to solve Poisson's equation on the unit square, using the finite volume element technique (FVE [3]) for composite grid discretization and MG [5] for solution of each grid level problem. This algorithm was also coupled with a relatively simple load balancer (MLB) based on assignment of the processors by levels; that is, instead of assignment by domain decomposition, processors are assigned to levels without regard to the location of these levels. Since AFAC requires little interlevel communication, such an effective load assignment and balancing approach is possible.

Loosely speaking, AFAC consists of two basic steps: Step 1. Given the solution approximation and composite grid residuals on each level, use MG to compute a correction local to that level. Step 2: Combine the local corrections with the global solution approximation, compute the global composite grid residual, and transfer the local components of the approximation and residual to each level.

To be more specific (see Hart and McCormick [4] for motivation and further detail), let Ω^k be a sequence of uniform *locally nested grids* associated with the *nested regions* $\Omega_k, 1 \leq k \leq l$. (By nested regions we mean $\Omega_{k+1} \subset \Omega_k$; by locally nested grids we mean $\Omega^k \cap \Omega_{k+1} \subset \Omega^{k+1}$.) Let $\Omega^0 = \cup_{k=1}^{l} \Omega^k$ be the *composite* grid. Now suppose H^k are the vector function spaces associated with the grids $\Omega^k, 0 \leq k \leq l$. Let $L^k : H^k \rightarrow H^k$ be the discrete operator on grid space H^k. Note that L^0 is an approximation to the differential operator on the global region $\Omega_0 = \Omega_1$. Let $I_k^0 : H^k \rightarrow H^0$ and $I_0^k : H^0 \rightarrow H^k$ be given intergrid transfers operators (interpolation and restriction, respectively). Finally, consider the intermediate grids $\tilde{\Omega}^k = \Omega^{k-1} \cap \Omega^k$ and their associated spaces $\tilde{H}^k$ and operators $\tilde{L}^k, \tilde{I}_k^0$, and $\tilde{I}_0^k, 2 \leq k \leq l$. (These intermediate grids are designed to remove error components that prevent simultaneous processing of the levels.) Then one iteration of AFAC, given the composite grid right-hand side f^0 and inital approximation u^0, is defined as follows:

Step 1. Set $u^k = (L^k)^{-1} I_0^k (f^0 - L^0 u^0)$ and $\tilde{u}^k = (\tilde{L}^k)^{-1} \tilde{I}_0^k (f^0 - L^0 u^0)$.

Step 2. Set $u^0 \leftarrow u^0 + I_1^0 u^1 + \sum_{k=2}^{l} (I_k^0 u^k - \tilde{I}_k^0 \tilde{u}^k)$.

The processing of each step of AFAC is fully parallelizable. For example, the levels in step 1 can be processed simultaneously by an MG solver, which is itself parallelizable [5]. The present version of the code uses synchronization of these steps, although asynchronous processing would be allowed here. (With an efficient load balancing scheme, asynchronous processing of the levels provides little real advantage.)

3. MLB.

MLB is responsible for the dynamic readjustment of the evolving composite grid. As new grids are built, the composite grid is modified, its tree is updated in all processors, and the partitions are adjusted to balance the loads. Given the tree data structures used for the storage of the matrices, MLB can adjust a partition at a cost commensurate with the amount of data transferred between processors. Additionally, MLB assures that the data transferred between processors during partitioning follows a minimum length path. Further, since the cost of determining if a multiprocessor system requires balancing (a global communication) is approximately the cost of MLB when no partitions are changed, there is a negligible penalty for the frequent load balancing required in dynamic refinement.

4. FVE.

To determine the L^k from a given differential operator, an effective discretization process for composite grids must be chosen. The basic idea behind FVE is to start from the differential equation, which we write as $Lu = f$, and convert it to its primitive form:

$$\int_V Lu \, dV = \int_V f \, dV.$$

Here, V is a given 'control volume' in Ω_0. Speaking loosely, since V is nearly arbitrary, the primitive equation represents an infinite number of equations; the space H of the admissible u is infinite dimensional, so this equation has an infinite number of unknowns. To discretize it, FVE first discretizes the number of equations by selecting a finite number of volumes V that typically partition Ω_0. To discretize the unknowns, the space H is replaced by a finite element subspace, where the element nodes typically have a one-to-one correspondence with the discrete volumes. In many fluid flow applications, finite element functions are inadmissible for the primitive equation. For example, L may contain second order derivatives, so simple continuity at element interfaces is not enough to make Lu integrable. In such cases, $\int_V Lu \, dV$ is usually transformed by the Gauss Divergence Theorem to a surface integral involving lower-order derivatives. Actually, the 'resulting' form is often the original form of the physical conservative law that was used to derive the differential equation in the first place.

FVE guides not only the choice of the L^k, but the construction of the intergrid transfers as well. For example, I_k^{k+1} and I_k^0 naturally correspond to the identity on the conforming finite element spaces associated with the corresponding grids. The restriction operators are naturally determined by the relationships between the volumes on these grids. Advantages of these constructs include the Galerkin relationship: $L^k = I_0^k L^0 I_k^0$. This means that L^k and L^0 agree at points of Ω_k where there is no further refinement. Other properties depend on the choice of elements and volumes, but typical stencils are simple (e.g., 5-point), autonomous (i.e., they do not need to be assembled but can be implemented algebraically in the code instead), and fairly regular at grid interfaces (i.e., because the interfaces are between uniform grids, the composite grid stencils there have a regular pattern). These properties simplify AFAC and ensure its optimal complexity.

5. Comments.

The combined AFAC-MLB-FVE method has been applied successfully (theoretically and numerically) to a wide range of problems that include general features like nonlinearities, high Reynolds number of times, transonic regimes, systems, and fully adaptive strategies. Of critical interest is the method's natural ability to maintain conservation without special mechanisms to account for grid interfaces. This combined scheme seems to adopt many of the best features of finite difference approaches, especially simplicity, and of finite element approaches, especially robustness.

References.

[1] S. McCormick, "Fast adaptive composite grid (FAC) methods: theory for the variational case," in *Proc. Error Asymptotics and Defect Correction*, (K. Böhmer and H. J. Stetter, eds.), Computing Supplementum **5**, Springer-Verlag, Wien, 1984.

[2] S. McCormick and J. Thomas, "The fast adaptive composite grid method for elliptic boundary value problems," *Math. Comp* **46** (1986), pp. 439–456.

[3] C. Liu and S. McCormick, "The finite volume element method (FVE) for planar cavity flow," *Proc. of the 11th International Conf. in Computational Fluid Dynamics,* Williamsburg, Virginia, 1988.

[4] L. Hart and S. McCormick, "Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors," *Lecture Notes in Pure and Applied Mathematics/110,* Marcel-Dekker, May, 1988.

[5] W. Briggs, L. Hart, S. McCormick and D. Quinlan, "Multigrid methods on a hypercube," *Lecture Notes in Pure and Applied Mathematics/110,* Marcel-Dekker, May, 1988.

PARALLEL MULTIGRID IN ADAPTIVE RESERVOIR SIMULATION

Peter Michielse
Technische Universiteit Delft
Technische Wiskunde en Informatica
Postbus 356, 2600 AJ Delft

In numerical oil–reservoir simulation, adaptive local grid refinement may be applied to model fluid–flows in an oil–reservoir in an efficient way. In [1], a multigrid method is used at each time–step to solve the elliptic part of the reservoir simulation equations for the primary unknowns p (pressure) and u (total flow), on a locally refined grid. After a short introduction to the sequential multigrid–algorithm, we will discuss a parallelization method, based upon domain decomposition techniques, for a parallel computer with a modest number of processors. Techniques to handle very coarse grids are considered. Also a load–balancing procedure [2], to balance work–loads equally over the processors, is discussed. The parallel multigrid–method has been implemented on local memory machines (NCube) as well as shared memory machines (Alliant). Experimental results (timings and convergence–rates), for uniform and locally refined grids, obtained on both types of computers will be shown.

[1] G.H. Schmidt and F.J. Jacobs, Adaptive local grid refinement and multigrid in numerical reservoir simulation, Journal of Comp. Physics, vol. 77, 140–165, 1988.

[2] P.H. Michielse, Domain decomposition in parallel reservoir simulation, report 87–63, Delft University of Technology, Faculty of Mathematics and Informatics, The Netherlands.

MIXED AND IRREDUCIBLE FORMULATIONS FOR THREE-DIMENSIONAL ELECTROMAGNETIC FIELDS IN STRONGLY INHOMOGENEOUS MEDIA

G. Mur
Technische Universiteit Delft
Faculteit der Elektrotechniek
Postbus 5031
2600 GA Delft

Abstract:

Both mixed and irreducible formulations for the computation of three-dimensional electromagnetic fields are discussed. In particular a method is presented for the efficient and accurate modeling of the electromagnetic field when "strong" inhomogeneities are present.

THEORETICAL CONSIDERATIONS ON A FINITE-ELEMENT METHOD FOR THE COMPUTATIONS OF THREE-DIMENSIONAL SPACE-TIME ACOUSTIC WAVE FIELDS

H.J. Stam
Technische Universiteit Delft
Faculteit der Elektrotechniek
Postbus 5031
2600 GA Delft

ABSTRACT

The theory of a mixed space-time finite-element method for the numerical solution of acoustic wave problems in bounded time-invariant subdomains of three-dimensional space is presented. The configuration may consist of fluid and solid parts. The wave fields in the fluid parts are characterized by their particle velocity and scalar traction; the wave fields in the solid parts are characterized by their particle velocity and stress; these quantities are regarded as the state variables. In our mixed finite-element method we approximate the different state quantities simultaneously.

It is shown how the finite-element method can be regarded as to be based on a space-time elastodynamic reciprocity theorem of the time-correlation type. In its turn this theorem is shown to be equivalent to the mixed formulation that follows from a certain weighting procedure applied to the equation of motion and the deformation rate equation that govern the wave motion.

The fluids and solids in the configuration are taken to be linear, locally and instantaneously reacting, and time-invariant in their elastic behaviour. Arbitrary inhomogeneity and anisotropy are taken into account.

Particular local representations for the acoustic wave field quantities are introduced that can handle strongly inhomogeneous structures, in which solid/solid, fluid/fluid and fluid/solid interfaces may be present.

THE DESIGN OF STABLE MIXED FINITE ELEMENT SCHEMES FOR FLOW PROBLEMS

R. Verfürth
Institut für Angewandte Mathematik
Universität Zürich
Rämistrasse 74
8001 Zürich, Switzerland

A key-point in almost all algorithms for numerical flow simulation is an efficient discretization of the Stokes problem. A main difficulty here is the continuity equation $\nabla \cdot \mathbf{u} = 0$ for the velocity field since there are no low-order finite element spaces which satisfy this condition exactly and are conforming, i.e. continuous, at the same time. This difficulty can be circumvented by either relaxing the continuity equation or the conformity condition. The first alternative leads to so-called mixed finite element schemes.

In this talk we first give an abstract stability and convergence result for mixed finite element schemes. The crucial point here is the so-called inf-sup or Babuska-Brezzi condition [2,7,10] which couples the finite element spaces discretizing the velocity and pressure, resp. We give an heuristical argument showing the necessity of this condition and examples illustrating the effects of its violation [5,6].

Then we present a framework for constructing mixed finite element schemes of arbitrary order satisfying the inf-sup condition [4,10]. The main tool here is a localization technique which allows to check the inf-sup condition element-wise or on suitably constructed macro-elements [4,11].

Finally, we discuss the most commonly used schemes (Mini element [1], Taylor-Hood element [3,12], Crouzeix-Raviart element [9,10]) and shortly discuss their advantages and disadvantages.

References

1 **D. N. Arnold, F. Brezzi , M. Fortin** : A stable finite element for the Stokes equations. *Calcolo* 21, 337-344 (1984)

2 **I. Babuska** : The finite element method with Lagrange multipliers. *Numer. Math.* 20, 179-192 (1973)

3 **M. Bercovier, O. Pironneau** : Error estimates for finite element method solution of the Stokes problem in the primitive variables. *Numer. Math.* 33, 211-224 (1979)

4 **J. M. Boland, R. A. Nicolaides** : Stability of finite elements under divergence constraints. *SIAM J. Numer. Anal.* 20, 722-731 (1983)

5 **J. M. Boland, R. A. Nicolaides** : On the stability of bilinear-constant velocity pressure finite elements. *Numer. Math.* 44, 219-222 (1984)

6 **J. M. Boland, R. A. Nicolaides** : Stable and semistable low order finite elements for viscous flows. *SIAM J. Numer. Anal.* 22, 474-492 (1985)

7 **F. Brezzi** : On the existence, uniqueness, and approximation of saddle-point problems arising from Lagrangian multipliers. *RAIRO Anal Numér.* 8, 129-151 (1974)

8 **F. Brezzi, J. Pitkäranta** : On the stabilization of finite element approximations of the Stokes equations. *Report MAT-A 219*, Helsinki University of Technology, 1984

9 **M. Crouzeix, P. A. Raviart** : Conforming and non-conforming finite element methods for solving the stationary Stokes equations. *RAIRO Anal. Numér.* 7, 33-76 (1973)

10 **V. Girault, P. A. Raviart** : Finite element approximation of the Navier-Stokes equations. *Computational Methods in Physics, Springer*, Berlin, 2nd edition, 1986

11 **R. Stenberg** : Analysis of mixed finite element methods for the Stokes problem: a uniform approach. *Math. Comput.* 42, 9-23 (1984)

12 **R. Verfürth** : Error estimates for a mixed finite element approximation of the Stokes equations. *RAIRO Anal. Numér.* 18, 175-182 (1984)

13 **R. Verfürth** : Finite element approximation of incompressible Navier-Stokes equations with slip boundary condsition. *Numer. Math.* 50, 697-721 (1987)

14 **R. Verfürth** : Mixed finite element approximation of the vector potential. *Numer. Math.* 50, 685-695 (1987)

A POSTERIORI ERROR ESTIMATORS AND ADAPTIVE MESH-REFINEMENT FOR FLOW PROBLEMS

R. Verfürth
Institut für Angewandte Mathematik
Universität Zürich
Rämistrasse 74
8001 Zürich, Switserland

In the numerical approximation of partial differential equations, one often encounters the problem that the overall accuracy of the numerical solution is degraded by local singularities of the solution of the continuous problem such as ,e.g., the singularity near a re-entrant corner. An obvious remedy is to refine the discretization in the critical regions, i.e., to place more gridpoints where the solution is less regular. The question is how to identify these regions automatically and how to determine a good balance of the number of gridpoints in the refined and unrefined regions such that the overall accuracy is optimal. This problem can be solved by trying to estimate locally the error of the numerical solution from the given data and the numerical solution itself.

In [2,3,5,6] automatic, self-adaptive mesh-refinement techniques based on a posteriori error estimators are investigated for finite element discretizations of 2nd order elliptic PDEs. The main idea is to solve local problems, which have the same structure as the original one and which involve the residual of the calculated numerical solution, using higher order finite elements. The error is then estimated by suitable norms of the solution of the local problems. Typically, one can prove that one obtains locally lower and globally upper bounds for the error of the numerical solution.

In this talk, we introduce two a posteriori error estimators for mixed finite element approximations of the Stokes equations. The first one simply computes suitable norms of the residual with respect to the strong form of the differential equation; the second one requires the solution of local Stokes problems. For both estimators, we prove that they yield locally lower and globally upper bounds for the error of the numerical solution. In contrast to [5], we do not need a super-approximation assumption for the finite elements used for the estimator with respect to those used for the numerical solution. On the other hand, we have in our estimates additional higher order terms involving the right-hand side. Our estimators can easily be extended to the Navier-Stokes equations and to other approximation schemes. We present numerical examples which demonstrate the efficiency of the error estimators.

References

1 **E. M. Abdalass** : Resolution performante du problème de Stokes par mini-éléments, maillages auto-adaptifs et méthodes multigrilles - applications. Thèse de 3me cycle, *Ecole Centrale de Lyon*, 1987

2 **I. Babuska, W. C. Rheinboldt** : A posteriori error estimates for the finite element method. *Int. J. Numer. Methods in Engrg.* 12, 1597-1615 (1978)

3 **I. Babuska, W. C. Rheinboldt** : Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.* 15, 736-754 (1978)

4 **R. E. Bank, T. Dupont, H. Yserentant** : The hierarchical basis multigrid method. *Konrad Zuse Zentrum,* Berlin, Preprint SC-87-2 (1987)

5 **R. E. Bank, A. Weiser** : Some a posteriori error estimators for elliptic partial differential equations. *Math. Comput.* 44, 283-301 (1985)

6 **K. Eriksson, C. Johnson** : An adaptive finite element method for linear elliptic problems. *Math. Comput.* 50, 361-383 (1988)

7 **R. Verfürth** : A posteriori error estimators for the Stokes equations. *Numer. Math.* 55, 309-325 (1989)

8 **R. Verfürth** : A posteriori error estimators and adaptive mesh-refinement for a mixed finite element discretization of the Navier-Stokes equations. to appear in: *Proc. 5th GAMM Conf. on Numerical Methods for the Navier-Stokes Equations* (W. Hackbusch, R. Rannacher; eds.) Vieweg, Braunschweig 1989

PARALLEL ALGORITHMS FOR LARGE LINEAR SYSTEMS

Henk van der Vorst
Technische Universiteit Delft
Technische Wiskunde en Informatica
Postbus 356, 2600 AJ Delft

PART I

The present supercomputers achieve their high computational speeds through the parallel use of vector CPU's. In order to deliver data to the CPU's at the required speed, some memory hierarchy is introduced with vector registers and/or cache. Examples of commercially available parallel supercomputers include CRAY X-MP (4 processors), CRAY Y-MP (8 processors), NEC-SX3 (4 processors), IBM 3090-VF (6 processors), Alliant FX18 (8 processors) and CONVEX C240 (4 processors). Supercomputers in the high Gflops range with as much as 64 vectorprocessors are under design (CRAY-4, Chen?).

Vector CPU's, parallelism and memory hierarchy have a tremendous impact on the performance and hence on the choice and the design of linear algebra algorithms. This is nicely reflected by the three generations of Basic Linear Algebra Subprograms, which have been designed for problems with dense matrices.

For sparse linear problems it is much less clear whether suitable basic kernels can be defined such that frequently used algorithms, when expressed in terms of those kernels, can be executed at near optimal speed. The design of suitable parallel and vectorizable algorithms for this class of problems is still subject of intensive research. Some of these algorithms are based upon ideas developed for very simple problems, like, e.g., the solutions of bidiagonal or tridiagonal linear systems.

Literature:

1. James M. Ortega, Introduction to parallel and vector solutions of linear systems, Plenum Press, New York, 1988.
2. Henk van der Vorst, Parallel rekenen en supercomputers, Academic Service, Schoonhoven, 1988.

PARALLEL ALGORITHMS FOR LARGE LINEAR SYSTEMS

Henk van der Vorst
Technische Universiteit Delft
Technische Wiskunde en Informatica
Postbus 356, 2600 AJ Delft

PART II

Iterative solution methods, like CG, GMRES and CGS, are usually only attractive for solving certain sparse linear systems, when used in combination with a suitable preconditioning matrix. Even without any preconditioning it is often quite difficult to achieve near-optimal computational speeds for these methods and for many problems the existence of effective proeconditioners is still an open problem. Hence, one might easily get the impression that the modern supercomputer architectures in this respect form an extra handicap instead of offering clear advantages.

Several approaches for precondtioned iterative methods with a suitable degree of parallel vectorizablility have been suggested. Some of them lead, indeed, to rather high performances but their numerical properties are less favourable. Other approaches seem to be more attractive but their effect is not yet fully understood, nor is it clear how to extend them for more than a modest number of processors. The more attractive parallel precondtioners, at least from a numerical point of view, have a strong relationship with domain decomposition methods.

A somewhat embarassing conclusion from experiments is that this type of parallel approach has different effects on, e.g., ICCG, MICCG and preconditioned CGS. Careful examination of the eigenvalues and the convergence histroy of the Ritz values, for the preconditioned matrices, seems to indicate an explanation.

Literature:

1. Henk van der Vorst, High performance preconditioning, SIAM J. on Sc. & St. Comput., to appear.
2. Henk van der Vorst, Large tridiagonal and block tridiagonal linear systems on vector and parallel computers, Parallel Computing, 5, p. 45-54, 1987.
3. Henk van der Vorst, The convergence behaviour of some iterative solution methods, Report 89-19, Fac. of Math. and Inf., Delft University of Techn., 1989.

MIXED FINITE ELEMENTS FOR POROUS MEDIA PROBLEMS

M.F. Wheeler
Mathematical Sciences Department
Rice University
Houston, Texas 77251, U.S.A.

Mixed finite element methods for computing an approximation to the pressure equation and Darcy velocity are formulated.
Both theoretical and computational results are discussed.

DOMAIN DECOMPOSITION, MIXED FINITE ELEMENT METHODS AND MULTI-LEVEL PROCEDURES

M.F. Wheeler
Mathematical Sciences Department
Rice University
Houston, Texas 77251, U.S.A.

We consider the numerical solution of elliptic partial differential equations by a combination of domain decomposition algorithms, mixed finite element methods and multi-level procedures.
Here the multi-level procedure is used to accelerate convergence of the algorithm, which iteratively adjusts the matching condition at the interfaces of the subdomains.
Numerical results are presented.