Report
on the Dagstuhl Seminar 9745

Parallel Processing in the Engineering Sciences – Methods and Applications

Organized by

G. Alefeld, Universität Karlsruhe
O. Mahrenholtz, TU HH-Harburg
R. Vollmar, Universität Karlsruhe

November, 3–7, 1997

Overview

The Dagstuhl seminar was attended by 33 participants from four countries. The intention was to bring together scientists from the fields of Numerical Analysis, Computer Science, Engineering and Natural Sciences, respectively, in order to discuss the state of the art and future developments of parallel processing in the applied sciences. The meeting provided a forum of exchange between these different research fields.

In 24 talks various parallel algorithms for different computer architectures and parallel software for mathematical modeling of real life applications e.g. in climatology, structural mechanics, tomography and acoustics were presented. In a plenary session the state of the art and the future of parallel computation in the engineering sciences was discussed focusing on technology transfer, the requirements of engineers, hardware and software. This report contains in alphabetic order the abstracts of the given talks and the summary of the plenary session.

We express our thanks to the administration and to the staff of Schloß Dagstuhl for the pleasant atmosphere and to all who contributed to the success of the conference.

Contents

Possibilities and Limits of Numerical Simulation in Regional Climatology

Gerhard Adrian, Forschungszentrum Karlsruhe / Universität Karlsruhe,
Institut für Meteorologie und Klimaforschung, D-76021 Karlsruhe,
adrian@imk.fzk.de

The first numerical models of atmospheric flow processes based on crudely approximated systems of equations derived by scale analysis from the Navier-Stokes equations for only small intervals of scales of atmospheric motions. The approximations were necessary because of the technical limitations of the available computer systems. Parallel to improvements of computer performance and of availability of memory not only the grids have been refined but also the accuracy of the physical models has been improved more and more.

Now full elastic models are developed which describe also sound waves, although sound processes are not assumed to be relevant for atmospheric flow phenomena. But the full set of the elastic Navier-Stokes equations has a quite simple structure in comparison to sound proved systems like anelastically approximated models. These structures allow to use quite simple structures of the numerical models, which can therefore implemented on parallel computer systems without too much efforts. Unfortunately these systems require very short time steps.

To describe the climate long time series of atmospheric variables of state are calculated by global circulation models. The statistical properties of the model outputs are evaluated similar to time series of observed variables. On a regional scale this approach is not practical because of the necessarily more complete models so that different methods have to be applied to estimate the state of the regional climate from a large number of simulations. Some examples of different approaches are discussed with results of the Karlsruhe Atmospheric Mesoscale Model KAMM, which has been developed at the IMK.

This model KAMM has been redesigned for parallel computing. The new possibilities of parallel computing enable the improvement of the accuracy as well of the physical as of the numerical approximations and parameterizations. The new full elastic model is written in FORTRAN-90. The necessary communication is gathered into few functions collected in a communication library, which is written using MPI for different computer architectures, for massive parallel systems, for parallel vector computers requiring long vectors, but also for monoprocessors.

Parallelization of Decision Support Applications in C++ — Practical Experiences —

Sonja Berner, Lufthansa Systems GmbH, 60556 Frankfurt/Main,
Sonja.Berner@lhsystems.com

Decision support applications become more and more important in airline business. To improve speed, these applications, written in C++, have been parallelized. Since the

applications deal with large data structures that are difficult to subdivide, a shared memory machine, a Convex SPP 1000, was used for parallelization.

The possibility of automatic parallelization of loops did not work well, thus explicit parallelization with threads was used. Since this was only supported for C, first of all C++ classes for threads, locks, guards, read-/write-locks etc. have been designed to simplify parallelization of C++ code and to take advantage of C++ features not available in C at the same time.

The first application parallelized was a market model which forecasts the passengers' connection selection. It consists of three steps which have all been parallelized explicitly handling dependencies by locks. Almost linear speedup was gained for the first two steps.

The second application, a parallel server, was designed using a master slave model. Use of a read-/write-lock allows to execute requests for reading in parallel while only the more rarely used requests for modifiying data are executed serially.

Problems which became evident during parallelization like use of bad programming styles and especially missing support and tools for parallelization of C++ code are described.

Exploiting Parallelism in Automated Differentiation

Christian Bischof, Mathematics and Computer Science Division,
Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439,
bischof@mcs.anl.gov

`http://www.mcs.anl.gov/autodiff`

Automatic versus Automated Differentiation

Automatic Differentiation (AD) is a method for augmenting algorithms with derivatives through methodological application of the chain rule of differential calculus. The ADIFOR (Automatic DIfferentiation of FORtran, see

`ftp://info.mcs.anl.gov/pub/tech_reports/reports/P481.ps.Z`)

and ADIC (Automatic DIfferentiation of C, see

`ftp://info.mcs.anl.gov/pub/tech_reports/reports/P626.ps.Z`)

augment Fortran 77 and ANSI-C codes with derivatives. They employ the so-called reverse mode of automatic differentiation at the statement level, and the so-called forward mode overall. Thus, computation of p directional derivatives $dy/dx * S$, where S is a matrix with p columns, requires $O(p)$ as much time and memory as the computation of f alone. ADIFOR and ADIC can compute arbitrary directional derivatives, allowing, for example, for derivative selection and derivative chaining, and can optionally generate code invoking the SparsLinC library to exploit sparsity in derivative computations in a transparent fashion.

ADIFOR and ADIC have been applied successfully in a variety of application domains. However, further reductions in runtime or memory savings are always desirable. Thus, we show two applications of *automated* differentiation, where (AD) tools are employed in a fashion that exploits high-level user knowledge to make optimal use of their capabilities.

FCAP and Derivative Stripmining

First, we consider differentiation of FCAP (Fact Capacitance Extraction), a code in use at Hewlett-Packard Labs in Palo Alto in the statistical simulation of circuit interconnects. Such simulation runs can take 5-10 days of CPU time on a HP9000/755 workstation, with derivative computations taking up the bulk of the time. Employing ADIC "out of the box", we achieve a factor 1.8 speedup over central divided differences, the previously employed method for computing derivatives, when computing 10 derivatives. By post-optimizing the ADIC-generated code through eliminating temporaries in critical loops and avoiding differentiation of subroutines that do not contribute to the dependent variables of interest, speedup over central differences increases to 4.3. This application is described in more detail in

`ftp://info.mcs.anl.gov/pub/tech_reports/reports/P698.ps.Z`.

However, memory requirements of FCAP.AD, the AD-generated code, are substantial – up to 208 Mbytes for the computation of 10 derivatives. Thus, we employ a stripmining technique for computing parts of the Jacobian independently and in parallel on several processors. This amounts to parallelizing the linear algebra kernels underlying automatic differentiation and results, for the "out of the box" ADIC-generated code, in a speedup of 1.8 and 3.0 on two and five processors, respectively. In contrast, the optimized code, which contains fewer vector operations, exhibits a lower speedup of 1.4 and 2.3, respectively. However, despite its inherently limited scalability, the simplicity of this technique make it an good candidate for getting around memory bottlenecks on networks of workstations.

Pseudoadjoints

Lastly, we consider a leapfrog scheme $Z(t+1) = H(Z(t), Z(t-1), W)$, where $Z(t)$ is the state of the system at time t, and W are time-independent parameters. We compute a merit function $r = R(Z(T))$, and we would like to compute $dr/d[Z(0), W]$. This scheme is employed, for example, in the MM5 mesoscale weather model developed by NCAR and Penn State. Typically, the Jacobian $dH/d[Z(t), Z(t-1), W]$ is sparse.

Straightforward use of a forward mode tool such as ADIFOR or ADIC to compute the desired derivative will require runtime and memory that are of the order $O(n+k)$ that of the function, where n is the number of state variables, and k is the number of time-independent parameters. For typical problems, where n easily is in the thousands, this is infeasible. We propose an alternative approach, where we preaccumulate the Jacobians $dH/d[Z(t), Z(t-1), W]$ for time steps $1, \ldots, T$ and then accumulate them in a backwards fashion. A straightforward recursive approach leads to a binary recursion which requires $O(2^T)$ multiplications of a sparse matrix with a dense vector, but by doubling up successive states, thus turning the leapfrog scheme into an Euler scheme, we arrive at a linear recursion that only requires $O(T)$ such multiplications. Thus, for a 2-D shallow water model on a 21x21 grid with 1323 state variables and 4 time-independent variables, the combination of exploitation of structure and parallelism results in up to a factor 70 speedup on a 16-processor IBM SP1. Details can be found in

`ftp://info.mcs.anl.gov/pub/tech_reports/reports/P639.ps.Z`.

Shared Programming on Shared Systems Truly Efficient Parallelization of Scientific Codes

Ralf Diekmann, Universität GH Paderborn,
Institut für Informatik, Fürstenallee 11, 33102 Paderborn,
diek@uni-paderborn.de
(Joint work with Jens-Michael Wierum)

With the invention of Sun's Ultra HPC systems, SGI's Origin and HP/Convex's Exemplar, powerful and at least partially scalable CC-NUMA systems become available offering GFlops of computing power, hundreds of processors and cache-coherent access to virtually shared memory. Still, these systems are programmed in a distributed style using MPI or PVM, because such message passing code is usually more efficient than compiler generated single address-space code.

We propose a thread-based model for parallelizing dynamic iterative scientific codes on CC-NUMA architectures. The basic idea is to program in a distributed style as to develop message passing code but then to take advantage of the fact that threads can access the global, shared memory of their father process. The increased locality automatically imposed by distributed programming is combined with the ease of shared models of being allowed to access arbitrary data in the global memory. While message-passing parallelizations are relatively easy for static codes, the difficulties increase a lot if dynamic, non-linear codes are being parallelized. The thread model allows a very easy re-assignment of work to processors. Therefor, load balancing is not a problem any longer provided the locality of partitions assigned to processors is not destroyed.

We suggest to use *spacefilling curves* to sort the data (nodes and elements) of an FE-mesh and to assign consecutive index ranges to individual threads. Spacefilling curves are partially locality preserving. Thus, any consecutive range of indices defines a compact partition. The load balancing can therefore be performed by simply shifting index boundaries which can be done in $O(\log P)$ steps if P is the number of processors. This very low overhead allows to re-assign the mesh every time-step of an explicit code. The resulting parallelization is very easy, causes only minimal changes in the code and delivers linear speedups on large numbers of processors.

The strategy is applied to a dynamic, explicit structural mechanics code. Due to a strong non-linear behavior of the complex material model, the code performs typically 10^6 steps on domains discretized by less than 5000 elements. Measurements show super-linear speedup on up to 16 processors (due to cache effects) and an efficiency of more than 90% on up to 28.

A Distributed Functional Framework for Recursive FE-Simulations

Ralf Ebner, Technische Universität München,
Fakultät für Informatik, D–80290 München,
ebner@informatik.tu-muenchen.de

http://www5.informatik.tu-muenchen.de/publikat/publikat.html

In contrast to standard finite element methods, the recursive substructuring technique assembles the element matrices recursively in several levels. The adaptive version results in an unbalanced tree with equation systems in each node. In order to increase efficiency and to reduce memory consumption, conventional static condensation is replaced by an iterative solver. Its convergence is enhanced by transformation to a hierarchical basis.

For automated parallelization, the method is coded using functional programming in Haskell and in the data flow language FASAN for the topmost coordinating layer (the framework), where data locality is achieved by monads and stream semantics, respectively. Adaptivity requires dynamic load balancing with strategies like bisectors, hashing techniques or the generation of threads.

Optimal Nearest Neighbor Load Balancing Schemes

Andreas Frommer, Bergische Universität GH Wuppertal, FB Mathematik,
Institut für Angewandte Informatik, Gaußstraße 20, 42097 Wuppertal,
andreas.frommer@math.uni-wuppertal.de

http://www.math.uni-wuppertal/SciComp/

We consider static diffusive load balancing schemes for local memory parallel computers. In these schemes, nearest neighbor communication is the only communication allowed. We present a new scheme which is based on an optimal polynomial which has to be precomputed once from the given processor graph by using a variant of the cg method. The resulting load balancing algorithm requires at most p (p is the number of processors) steps. Moreover, on several standard topologies like arrays, rings or hypercubes, these schemes are optimal since the number of iterations reduces to the diameter of the processor graph. Similar optimality can be achieved on more-dimensional grids and tori if one considers appropriate generalizations of the dimension exchange algorithm.

Implementation and Parallelization of Adaptive Multilevel Methods for the Solution of PDEs by Means of Hash Tables and Space Filling Curves

Michael Griebel, Universität Bonn,
Institut für Angewandte Mathematik, Wegelerstr. 6, 53115 Bonn,
griebel@iam.uni-bonn.de

http://wwwwissrech.iam.uni-bonn.de

The efficient solution of (elliptic) PDEs needs the application of fast solution methods, like multigrid and multilevel techniques, the application of adaptive refinement procedures in the discretization process to keep the number of degrees of freedom low, and the application of parallel computers to solve fast the arising large linear systems. Each of these three ingredients to an efficient overall solution method is well understood by itself, however to bring them all (adaptivity, multilevel, parallelization) together is a major challenge. Adaptive multilevel methods are usually based on tree-like data structures, which maintain the adaptively refined grid and the grid hierarchies for the multilevel method. Besides domain decomposition techniques are used for the parallelization. Here, however, the dynamic load balancing problem is very difficult to treat efficiently.

We report on our new method, discuss the hash table technique as data structure for adaptive multilevel methods, present the parallelization approach using the space filling curve idea, and give the results of numerical experiments on different parallel computers.

Advanced Solvers and Parallelism

Gundolf Haase, Johannes Kepler Universität Linz,
Institut f. Analysis und Computermathematik, Altenbergerstr. 69, A-4040 Linz,
ghaase@numa.uni-linz.ac.at

The starting point of our parallelization strategy is a non-overlapping domain decomposition (DD) used as data partitioning as well as a solution method. The location of the (FEM-) nodes on the boundary between different subdomains implies two types of vectors - accumulated and non-accumulated (distributed) ones. The same holds for the sparse (FEM-) matrices. Changing the type from distributed to accumulated requires next-neighbor communication. Whereas the parallelization of the cg- (Krylov-like) method is very easy it is in general not possible to factor the stiffness matrix without a dramatic loss in efficiency. The investigation of matrix-by-vector products with accumulated matrices leads to the possibility of special ILU factorizations requiring only one communication step per iteration.

The well-known multi-level iteration is much more efficient then an ILU preconditioned cg. Due to the data structure only the smoothing and the coarse grid solver need communication. The crucial point is the coarse grid solver including a global information transport. Therefore only a V-cycle can be implemented efficiently. For smoothing, we use the ω-Jacobi iteration, block Gauss-Seidel iterations or the special ILU factorization.

DD as a solution method is used for solving magnetic field problems by coupling of BEM and FEM. Here, multi-level solves the internal FEM-problems.

In future, the great advantage of multi-grid and multi-level methods as fast solvers for 2nd and fourth order partial differential equations will consist in the adaptive construction of the finer grids in combination with algebraic methods. Both opportunities are presented by joint projects with the industry.

Computing the Hausdorff Distance for Finite Unions of Hyperrectangles in Parallel

Karl-Udo Jahn, HTWK Leipzig,
FB Informatik, Mathematik und Naturwissenschaften,
Gustav-Freytag-Str. 42, Postfach 66, D-04251 Leipzig,
jahn@imn.htwk-leipzig.de

`http://www.imn.htwk-leipzig.de/~jahn`

In several fields such as image processing, pattern recognition, exemplar-based learning and integrated circuit design, the considered objects can be often approximated or represented by finite sets of (hyper-) rectangles resp. their unions. For comparing such sets, the Hausdorff distance is very useful. This distance can be computed directly after subdividing the (hyper-) rectangles in a suitable manner. Thus the number of (hyper-) rectangles is growing on. Nevertheless, a polynomial time algorithm (for bounded dimension n of the space in consideration) for a CREW shared memory computer is given that yields the Hausdorff distance of two unions $\bigcup_{i=1}^{r} A^i$ and $\bigcup_{i=1}^{s} B^i$ of (hyper-) rectangles A^i, $1 \leq i \leq r$ and B^j, $1 \leq j \leq s$.

Tools and Methods for Error Controlled (Parallel) Computing

Ulrich Kulisch, Universität Karlsruhe,
Institut für Angewandte Mathematik 76128 Karlsruhe
ae15@rz.uni-karlsruhe.de

Advances in computer technology are now so profound that the arithmetic capability and repertoire of computers can and should be expanded. The quality of the elementary floating-point operations should be extended to the most frequent numerical data types or mathematical spaces of computation (vectors, matrices, complex numbers and intervals over these types). A VLSI co-processor chip with integrated PCI-interface has been developed which provides these operations. The expanded capability is gained at modest hardware cost and does not implicate performance penalty. Language and programming support (the -XSC languages) are available. There the chip's functionality is directly coupled to the operator symbols for the corresponding data types. By operator overloading a long real arithmetic (array of reals) and long interval arithmetic as well as automatic differentiation arithmetic become part of the runtime system of the compiler. I. e. derivatives, Taylor-coefficients, gradients, Jacobian and Hessian matrices or

enclosures of these are directly computed out of the expression by a simple type change of the operands. Techniques are now available so that with this expand capability, the computer itself can be used to appraise the quality and the reliability of the computed results over a wide range of applications. Program packages for many standard problems of Numerical Analysis have been developed where the computer itself verifies the correctness of the computed result and proves existence and uniqueness of the solution within the computed bounds.

Many applications require that rigorous mathematics can be done with the computer using floating-point. As an example, this is essential in simulation runs (fusion reactor) or mathematical modeling where the user has to distinguish between computational artifacts and genuine reactions of the model. The model can only be developed systematically if errors entering into the computation can be essentially excluded. Automatic result verification re-integrates digital computing into real mathematics.

Efficient Orthogonal Transforms for Computing Eigenvalue and Singular Value Decompositions

Bruno Lang, Bergische Universität GH Wuppertal, FB Mathematik,
Institut für Angewandte Informatik, Gaußstraße 20, 42097 Wuppertal,
bruno.lang@math.uni-wuppertal.de

`http://www.math.uni-wuppertal.de/~lang`

We present techniques for speeding up the two major steps in eigenvalue and singular value computations, i.e., the reduction to condensed (tri- or bidiagonal) form and the QR iteration performed on the condensed matrix. Both techniques reduce the computing time on serial machines. In addition they increase the potential for exploiting parallelism.

For the reduction, using a two–phase algorithm (reduction to banded form, then tri- or bidiagonalization of the banded matrix) enables us to increase the portion of matrix–matrix operations in the overall computation, thus leading to higher performance on today's machines with their memory hierarchies. In the QR iteration the key to introducing matrix–matrix operations is a suitable reordering and grouping of the rotations.

In both techniques the performance depends crucially on an adequate setting of certain control parameters. We present a very simple performance model that allows to choose these parameters automatically.

Krylov Subspace Methods for Structural Finite Element Analysis

Ingrid Lenhardt, Universität Karlsruhe,
Institut für Angewandte Mathematik, 76128 Karlsruhe,
ingrid.lenhardt@math.uni-karlsruhe.de
(Joint work with Thomas Rottner, Institut für Mechanik, Universität Karlsruhe)

`http://ma70.rz.uni-karlsruhe.de/~ae23/`

The application of the finite element method to problems like collapse and buckling of structures leads to nonlinear problems. For the solution of these systems usually Newton-like methods are used. Therefore in each step of this method a symmetric linear system has to be solved which can be nearly singular or even indefinite, in most cases, however, ill-conditioned. In the past usually direct methods have been used.

For these problems iterative Krylov subspace methods with several preconditioning strategies (Jacobi, SSOR, different variants of incomplete factorization) for the arising ill conditioned problems were presented. For nonlinear computations with arc length methods an extended Lanczos method was developed.

For the considered problems the arising finite element grid is partitioned into non overlapping subdomains by domain decomposition methods from the METIS-library in a sequential preprocessing phase. Each processor P_i, $i = 1, \ldots, p$ works on a local grid and computes a partial stiffness matrix $\mathbf{K}_i$. The global stiffness matrix is the sum of the partial stiffness matrices: $\mathbf{K} = \sum_{i=1}^{p} \mathbf{K}_i$. The nodes on the arising inner boundaries belong to each corresponding subdomain and cause overlapping. The application of a Krylov subspace method for solving the linear system $\mathbf{Ku} = \mathbf{d}$ requires in each step vector updates, dot products, matrix-vector multiplication and preconditioning operations. For matrix-vector multiplication the local stiffness matrices $\mathbf{K}_i$ are used.

The preconditioning matrix is computed in parallel. For each partial stiffness matrix $\mathbf{K}_i$ the global entries at the overlapping components are computed. For this matrix a partial preconditioning matrix $\mathbf{M}_i$, e. g. by an incomplete factorization is computed in parallel. Thus the local preconditioning systems $\mathbf{M}_i \mathbf{t}^{j+1} = \mathbf{r}^{j+1}$ are solved in parallel. The entries at the overlapping components of $\mathbf{t}^{j+1}$ are computed more than once on different processors. These overlapping nodes are assembled in a pre- and post-weighting phase, then the resulting global preconditioning matrix $\mathbf{M}$ is symmetric:

$$\mathbf{M}^{-1} = \sum_{i=1}^{p} \mathbf{E}_i \mathbf{M}_i^{-1} \mathbf{E}_i$$

with positive definite diagonal matrices $\mathbf{E}_i$. Different weighting schemes are implemented.

Three examples from structural finite element analysis (torus, flange, tooth post crown) and speed-ups for the corresponding linear and nonlinear computations were presented.

Hyper-Systolic Parallel Computing: Theory and Applications

Thomas Lippert, Bergische Universität GH Wuppertal,
Institut für Physik, 42097 Wuppertal,
lippert@wptu21.physik.uni-wuppertal.de

In this talk I want to give an introduction to the theory of hyper-systolic computing. I will present two applications, exact N-body computations and a hyper-systolic formulation of matrix products.

Hyper-systolic algorithms define a new class of parallel computing structures. The scheme can be considered as an extension of standard systolic parallel computing on a 1-dimensional ring of systolic cells. Such a ring can be mapped onto a 1-dimensional (virtual) connectivity of the given parallel implementation machine.

The appropriate formulation of the hyper-systolic algorithm is found in the language of Additive Number Theory. The requirement for minimal interprocessor communication leads to an h-range problem which defines the shift bases for the hyper-systolic communication. In general, the determination of optimal bases is difficult. Shortest and optimal bases are produced by direct calculation, quasi-optimal bases can be found by simulated annealing methods. A sub-optimal solution is given by the so-called regular base algorithm which leads to a particularly simple hyper-systolic scheme. We distinguish between bases for 1-array and 2-array problems. The 1-array problem refers to the interaction of the elements of one systolic array, a prototype is the n-body computation, a 2-array problem is given by the interaction between two different arrays, a prototype is the convolution.

The implementation of systolic and hyper-systolic algorithms on a parallel system implies a mapping of the given systolic array (with n elements) onto the parallel computer (equipped with p processors). For 1-dimensional problems, cyclic-splitted mapping complies very well with the hyper-systolic data flow. For 2-dimensional problems, as found in matrix computations, 1-dimensional block mapping or 1-dimensional staggered mapping can be chosen.

The hyper-systolic algorithm for n-body computations with cyclic splitted mapping has been benchmarked on a Connection Machine CM5, a Cray T3D and an APE100/Quadrics. On the CM5, the theoretical scaling behavior of the hyper-systolic algorithm compared to the conventional systolic method has been proven. On the Cray T3D, super-linear scaling has been found, which is due to the reduced memory-to-cache data-transfer. On the APE100, a performance of more than 60 % of the peak performance of the machine has been achieved!

A very important application field of the hyper-systolic algorithm is the distributed matrix multiplication. The hyper-systolic matrix product is based on a 1-dimensional column-block data layout. Showing the same complexity as Cannon's algorithm with its 2-dimensional layout, it might be an alternative to existing schemes. Moreover, the data layout chosen is ideal for matrix-vector computations, therefore, in applications where a combination of matrix-matrix and matrix-vector products is required, the hyper-systolic algorithms can be advantageous.

There are many further scientific fields where hyper-systolic computations can be of use. I hope that the present talk can give some seminal ideas that will help to spread the new technique among the high-performance computing community.

Multisplitting Methods for Band Matrices

Günter Mayer, FB Mathematik, 18051 Rostock
guenter.mayer@mathematik.uni-rostock.de
(Joint work with Götz Alefeld and Ingrid Lenhardt, Karlsruhe)

Multisplitting methods are parallel iterative methods for solving $n \times n$ linear systems $Ax = b,\ A \in \mathbb{R}^{n\times n},\ b \in \mathbb{R}^n,\ \det(A) \neq 0$. They are based on p splittings $A = M_l - N_l,\ \det(M_l) \neq 0,\ l = 1, \ldots, p$, and on p weighting matrices $E_l,\ l = 1, \ldots, p$, which are diagonal matrices and which satisfy $\sum_{l=1}^{p} E_l = I$. Here, I denotes the identity matrix and p is the number of processors being involved in the computational process. Starting with $x^{(0)} \in \mathbb{R}^n$ one iterates according to

$$\left\{ \begin{array}{rcl} M_l\, y^{(k,l)} & = & N_l\, x^{(k)} + b\ , \quad l = 1, \ldots, p, \\ x^{(k+1)} & = & \sum_{l=1}^{p} E_l\, y^{(k,l)} = \sum_{l=1}^{p} E_l M_l^{-1} N_l x^{(k)} + \sum_{l=1}^{p} E_l M_l^{-1} b, \quad k = 0, 1, \ldots\ . \end{array} \right.$$

We report on some criteria of convergence for this method and on some comparison results which mainly hold for band matrices. In particular, we drop the usual restriction of E_l being nonnegative. Our theoretical results are confirmed by numerical examples.

A Systolic Implementation of the MLEM Reconstruction Algorithm for PET Images

Reinhard Möller, Bergische Universität GH Wuppertal,
FBE-Technik, Institut für Angewandte Informatik, Gaußstraße 20, 42097 Wuppertal,
rmoe@lfa.uni-wuppertal.de

This work is concerned to parallel reconstruction of 3D images from the data delivered by positron emission tomography. Due to the Poisson-nature of positron emission processes, likelihood models can be used for their description. We are investigating on an expectation- maximum algorithm, which iterates to a maximum likelihood of an estimated radiosity image.

The method called Maximum-Likelihood Expectation-Maximum (MLEM) was first introduced by Shepp and Vardy (1982). Its vast amount of necessary computation demanded several attempts to adapt the algorithm to high-performance computing systems since that time. The two main problems are the definition and storage of the huge ($> 2^{29}$ elements) and sparse transition matrix $\mathbf{p}(b, d)$, which describes the probability that an event happening at some location b is detected by detector d, and the matrix-vector multiplication.

Our parallel implementation is adapted to the modular architecture of a Quadrics DM-SIMD parallel computer, which can be tailored for special applications. We describe a "slice-wise" approach, i.e. parallel calculation on 15 slices of a 3D data set, one slice per processor. The projection calculation is based on an area weighted convolutional interpolation scheme according to Schwinger, Cool and King (1986). Further we present a systolic approach, that provides more precise calculations based on direct matrix-vector multiplication while making more efficient use of the Quadrics architecture.

The Modeling of Acoustical Phenomena

Andras Montvay, TU HH-Harburg
A.Montvay@tu-harburg.d400.de

`http://www.tu-harburg.de/~seam2104/parsecs.tar.gz`

Many applications of parallel computing pose the main task of porting existing sequential code to parallel machines. In contrast to this, the time-domain simulation of enclosed soundfields is a rather new field for computational science. Hence it is possible to choose freely among all the possible algorithms, equations etc. while considering suitability for parallelisation.

The first important decision to be taken is between time-domain and frequency-domain simulation. The main advantage of the former is the possibility to introduce arbitrarily shaped or even moving boundaries, which outweighs the disadvantages arising from possibly frequency-dependant behaviour of materials.

Acoustical phenomena can be described in the time domain either by the conservation laws of energy, momentum and mass, by the equations of motion or by the wave equation. Main aspects that have to be considered here are the memory requirements, the formulation of boundary conditions, and the amount of computation needed in each time step.

Arbitrarily shaped boundaries can be modeled in different degrees of sophistication: On a regular rectangular grid, a boundary point can be assumed to be rectangular, or to have an arbitrary angle, or an offset away from the grid point as well. The introduction of arbitrary angles implies an important improvement, while the error made by missing offsets can be compensated by refining the grid.

The given URL contains the postscript version (76 pages) of my project thesis ("Studienarbeit") entitled "Parallel Finite Difference Methods for the Simulation of Sound Fields in Arbitrary Boundaries".

Quasi-Optimal Mixed Load-Balancing of Parallel Adaptive Computations in Nonlinear 3D-Elasticity

Rainer Niekamp, Universität Hannover,
Institut für Baumechanik & Numerische Mechanik,
Appelstr. 9A, 30167 Hannover, rainer@verdi.ifbnm.uni-hannover.de
(Joint work with E. Stein)

For the effectivity of adaptive parallel FE-computations, particularly in the case of 3-dimensional problems, a load-balancing during running time becomes necessary. In p-adaptive methods as well as in the case of elasto-plastic deformations, the (element-related) effort due to the construction of the system matrix is no longer proportional to the (degree of freedom related) effort for solving the linearized system of equations (here using multilevel preconditioned Krylov-space methods). Therefore the partitionings of the FE-graph and of the system matrix are optimized separately. The repartition of the FE-graph is a nontrivial task, due to the used horizontally (neighborhood-relation) and vertically (parent-child relation) connected data structures. But, just because of these relations, it can be solved in a totally local manner, i.e. without any search operation. Therefore the amount of work is of the order O(#moved elements/#processor). The whole FE-program is coded in C++. Furthermore we present an adaptive 3-D refinement algorithm working on hexaeder elements generating regular meshes, i.e. without hanging nodes. Linear and nonlinear examples for geometrically nonlinear elasticity are given to illustrate the effectivity and robustness of these parallel algorithms.

Does Ill-Conditioned Mean Nearly Ill-Posed?

Siegfried M. Rump, TU HH-Harburg,
Arbeitsbereich Informatik III, Eißendorfer Str. 38, 21071 Hamburg,
rump@tu-harburg.de

It is well known that an ill-conditioned matrix is in a normwise sense not too far from a singular matrix. In fact, the distance is equal to the reciprocal of the condition number for matrices normed to 1. It is natural to ask whether a matrix being ill-conditioned in a componentwise sense (weighted by some nonnegative weight matrix) is also not too far from a singular matrix in the same componentwise sense. This has been conjectured by Demmel and N. Higham in 1991. The problem becomes involved due to the fact that Poljak and Rohn showed that the computation of the componentwise distance to the nearest singular matrix is NP-hard. The conjecture has been solved in the affirmative: Indeed a componentwise ill-conditioned matrix is nearly singular in the same componentwise sense. It has also been proved that this statement becomes false when restricting the perturbations to symmetric perturbations. Nevertheless, the general componentwise condition number and a condition number restricted to symmetric perturbation differs only by a small factor.

Parallel Simulation of Flows in Sewer Network Systems

Gudula Rünger, Universität Leipzig,
Institut für Informatik, Postfach 920, 04009 Leipzig,
ruenger@informatik.uni-leipzig.de
(Joint work with Ralf Ruth)

Water flow in sewer network systems can be simulated by solving the Saint-Venant equation, a coupled hyperbolic partial differential equation of first order. The parallel implementation is based on a spatial discretization according to the structure of the underlying sewerage system on which consecutive time steps simulate the water flow behavior. Two different basic algorithms are considered: a step-wise organization of the computation order and a breadth-first organization. Runtime tests are performed on an IBM SP2.

Industrial Application of an Explicit Finite Element Program on Parallel Computers

Karl Schweizerhof, Universität Karlsruhe,
Institut für Mechanik, 76128 Karlsruhe,
karl.schweizerhof@bau-verm.uni-karlsruhe.de

Die industrielle Anwendung Finiter Elemente für stark nichtlineare Probleme in der Strukturmechanik wird derzeit von den Programmen mit expliziten Zeitintegrationsverfahren dominiert. Die wesentlichen Anwendungsbereiche dieser Programme am Beispiel des FE Programms LS-DYNA sind: Crash- und Impactanalysen , Großdeformationsanalysen, Metallumformung dünner Bleche sowie Massivumformen, Innenhochdruckumformung, Insassensimulation, Airbagsimulation, Explosionsprobleme, Kopplung von strukturmechanischen Problemen mit instationärer Wärmeleitung, Kopplung mit Strömungsmechanik.

Die Vorteile der sogenannten expliziten Programme beruhen auf dem einfachen Zentralen Differenzen Verfahren, das bei Einsatz von diagonalisierten Massenmatrizen zu einem sehr einfachen Algorithmus führt, bei dem zur Gleichungsauflösung nur die Division jeder Gleichung durch einen Skalar auszuführen ist. Da mit dem Zeitintegrationsverfahren aus Stabilitätsgründen auch eine Zeitschrittbegrenzung anfällt, werden üblicherweise nur Elemente niederer Ansatzordnung eingesetzt. Elemente niederer Ansatzordnung zeigen jedoch oft "Locking" Effekte, so daß mit reduzierter Integration gearbeitet wird und auftretende Kinematiken mit "Hourglass"-Kontrolle verhindert werden. Damit wird zwar keine durchgehende Robustheit, aber sehr hohe Effizienz erreicht. Die Kontrolle auf Kinematiken und deren quantitativen Einfluß auf Ergebnisse erfolgt über die Überprüfung der Energiebilanzen, insbesondere der Hourglassenergie. Als Folge des einfachen "Forward marching"-Algorithmus ist auch die Bewältigung starker Nichtlinearitäten insbesondere infolge Kontakt gut möglich.

Explizite Programme sind wegen der einfachen Löserstrategie relativ einfach zu parallelisieren. Für die beiden wesentlichen Programmiermodelle – Virtual Shared und

Distributed Memory – liegen Programmversionen vor. Für Rechner mit Distributed Memory wird ein SPMD Modell benutzt; für die Gebietsaufteilung stehen die üblichen Zerlegungsverfahren wie Rekursive Spektrale Bisektion u.a. zur Verfügung. Da oft sehr heterogene Modelle vorliegen, werden Gewichte entsprechend der anfallenden Operationen pro Element und Material eingeführt. Auch für Kontaktgebiete kann eine separate Wichtung eingebracht und damit eine besserer Lastausgleich der Prozessoren erreicht werden.

Die Beispiele aus der Crashberechnung, Metallumformsimulation sowie der Insassenberechnung zeigen die erreichbare hohe Parallelität. Allerdings ist auch eine große Abhängigkeit von der Problemklasse insbesondere - nicht unerwartet - bei Shared Memory Rechnern erkennbar. Es werden für verschiedene Rechner Rechenzeit- und Effizienzvergleiche mit Prozessorzahlen bis 128 gezeigt.

Noch offen ist die effiziente Behandlung numerischer Sensitivitäten bei Beulproblemen, die bei Wechsel der Prozessorzahlen zu teilweise unterschiedlichen Ergebnissen führen kann. In der praktischen Anwendung wird gewünscht, auch in diesem Fall gleiche Ergebnisse wie auf seriellen Rechnern zu erhalten. Es wird daher angestrebt, auch bei einer parallelen Bearbeitung der Probleme die Summation analog zur seriellen Bearbeitung durchzuführen. Dieses Vorgehen hat sich für die Shared Memory Version in der praktischen Anwendung bewährt und wird derzeit für die Version für MIMD Rechner implementiert.

Zusammenfassend läßt sich feststellen, daß explizite FE Programme für strukturmechanische Simulationen in gut parallelisierter Form vorliegen und auch bereits in der Industriepraxis genutzt werden. Der industrielle Einsatz auf Rechnern mit verteiltem Speicher hat begonnen. Auf Grund der hohen geforderten Rechenleistungen bei stetig wachsenden Modellgrößen ist der Einsatz dieser Rechner in der nächsten Zeit unumgänglich. Im Einsatz sind sicher noch wesentliche Effizienzverbesserungen durch algorithmische Anpassungen und Verbesserungen der Parallelisierung zu erwarten.

Algorithmic Trends in Industrial Scientific Computing

Ulrich Trottenberg, GMD, Inst. f. Algorithmen & Wissenschaftl. Rechnen (SCAI),
Schloß Birlinghoven, 53757 St. Augustin,
trottenberg@gmd.de

Der Vortrag diskutiert an Hand konkreter Projekte (EUROPORT, POPINDA, CISPAR) aktuelle Trends des Wissenschaftlichen Rechnens. Dabei wird der spezielle Aspekt der industriellen Softwareentwicklung in den Vordergrund gestellt.

Durch EUROPORT ist eine Vielzahl industrieller Programme zur Simulation und Untersuchung von Einzelphänomenen (Strömungslöser, Strukturcodes und viele andere) erfolgreich parallelisiert worden.

Um außer der Parallelität auch Skalierbarkeit und Adaptivität zu garantieren, muß tiefer in die Programmstrukturen und die Algorithmik eingegriffen werden: Im POPINDA-Projekt ist der FLOWER-Code entstanden, ein 3D-Navier-Stokes-Löser für volle Flugzeugkonfigurationen auf blockstrukturierten Gittern. Der FLOWER-Code ist das Pro-

duktionsprogramm der Airbus-Entwicklung. Die zugrundeliegende 3D-Kommunikationsbibliothek garantiert skalierbare Parallelität und Adaptivität.

Noch einen Schritt weiter, auch über die EUROPORT-Ergebnisse hinaus, geht die Anforderung der Industrie, für die gekoppelte Simulation verschiedener physikalischer Effekte (Strömungs-Struktur-Wechselwirkung usw.) bessere Design- und Simulationstools zur Verfügung zu haben. Diese Kopplungsproblematik steht im EU-Projekt CISPAR im Vordergrund. Die algorithmische und softwaretechnische Lösung dieser Problematik wird durch die Kopplungsbibliothek COCOLIB gewährleistet.

Modal Masters in Improved Parallel Condensation Methods for Symmetric Eigenvalue Problems

Heinrich Voss, TU HH-Harburg,
Arbeitsbereich Mathematik, Kasernenstraße 12, D–20173 Hamburg,
voss @ tu-harburg.d400.de

In the analysis of the dynamic response of a linear structure using finite element methods very often prohibitively many degrees of freedom are needed to model the behaviour of the system sufficiently accurate. In this situation static condensation methods are employed to economize the computation of a selected group of eigenvalues and eigenvectors. These methods choose from the degrees of freedom a small number of master variables which appear to be representative. The rest of the variables (termed slaves) is eliminated leaving a much smaller problem for the master variables only.

It has frequently been noted that the quality of the eigenvalue and eigenvector approximations produced by static condensation is satisfactory only for a very small part of the lower end of the spectrum. Several attempts have been made to improve the quality most of them being very time consuming since an iterative process is involved. Exploiting properties of the exactly condensed problem (which is a nonlinear eigenvalue problem of the same dimension as the statically condensed problem and which has the same eigenvalues as the original problem) the eigenvalue approximations can be improved considerably.

Partitioning the structure under consideration into substructures and choosing the degrees of freedom on the interfaces of the substructures as masters leads to data structures and formulae which are well suited to be implemented on distributed memory MIMD parallel computers.

The part of the spectrum which can be approximated accurately enough by the enhanced condensation method depends crucially on the size of the minimum slave eigenvalue. To extend the region of eigenvalues which can be obtained with sufficient accuracy a method at hand is to incorporate additional master variables which are degrees of freedom at interior nodes of the substructures. In this case the data structure essentially is preserved and the parallel method can be carried over in a straightforward way.

For a fixed number $\tilde{m}$ of additional master variables for each substructure the minimum slave eigenvalue gets maximal if we choose those modal degrees of freedom corresponding to the $\tilde{m}$ smallest slave eigenvalues of each substructure. To apply the approach directly a

transformation of variables to a new orthonormal basis of the space spanned by the slave eigenvectors under consideration and its orthogonal complement has to be performed for each substructure which is very time consuming. In this lecture we discuss the application of the nonnodal condensation method which circumvents the use of the orthonormal basis of the orthogonal complement. It is shown that the nonnodal condensation using interface masters, interior masters, and modal masters can be performed completely in parallel. Examples demonstrate the favourable properties of the method.

Comparison of Parallelization Strategies for Finite Element Calculations

Astrid Watermann, Forschungszentrum Jülich GmbH,
Institut für Sicherheitsforschung und Reaktortechnik (ISR), 52425 Jülich,
a.watermann @ fz-juelich.de

The finite element method is a powerful tool to solve partial differential equations, which occur in many engineering and natural science problems. The popularity of this method grew during the recent years with the increase of the performance of available computers. Even so computing times for applications of practical interest, for example crash test simulations with complete car bodies, are still far away from real times - a reasonable improvement of computing times has been achieved by parallel processing. Because there are still no general purpose codes for parallel machines available, which have been developed from scratch, existing serial finite element codes have been parallelized. Two such codes are SMART and PERMAS. In this paper the parallelization strategies for both codes, a data parallel and a host node concept are introduced and compared. Pros and Cons of both strategies are discussed.

In this paper parallelization strategies for two large finite element systems are described: the SMART-code, a scientific code, which is used at universities and research centers and the PERMAS code - a commercial system. Both codes are general purpose systems and consist of comparable program and data structures, because the predecessor of both programs was the well known code ASKA, one of the first finite element codes, which has been developed at the University of Stuttgart, Germany [2].

During the last years both systems have been parallelized, SMART based on a data parallel concept, PERMAS on a host node concept [1]. The strategy for the SMART code is a static concept [1], based on a geometrical mesh decomposition. This means only few additional management statements are necessary for parallel program execution, changes in the original code are few and the performance is very good compared with the parallelization effort. The host node concept is a dynamic strategy, which needs a lot of modifications within the serial code and high additional amount of data management and communication for parallel processing. However, this concept is much more flexible than the data parallel one, because of a dynamic load balancing.

References

[1] A. Watermann, Parallelisierung des Finite-Element-Codes SMART-Konvektion, Institut für Sicherheitsforschung und Reaktortechnik, Forschungszentrum Jülich GmbH, Jül-Bericht 3122, ISSN 0944–2952, 1995.

[2] J. Argyris und H.P. Mlejnek, Die Methode der Finiten Elemente, Band I, II, III, Friedrich Vieweg & Sohn, Braunschweig/Wiesbaden, 1986, 87, 88.

Adaptivity versus Parallelism

Helmut Weberpals, TU HH-Harburg,
Technische Informatik II, Harburger Schloßstr. 20, D–21071 Hamburg,
weberpals@tu-harburg.de

Adaptivity usually degrades parallel efficiency considerably. This is why it is hard to find adaptivity in parallel production codes. We present two examples which demonstrate the need for adaptivity.

The first example deals with the head-on collision of vortices in plasma physics. By splitting the model equation into an elliptic equation and into a time dependent equation we are able to solve these equations using a multigrid method in space and a leap-frog method in time. In this case, the multi-level approach makes an adaptive grid superfluous.

The second example concerns the drift-diffusion model of semiconductor devices in electrical engineering. Adaptivity is needed for modeling regions of strongly varying electron density as well as 3-dimensional effects in submicron semiconductors. Numerical experiments clearly show that data access is more critical than computation (therefore, we preserve regular data structures and data affinity) and that many lightweight threads per processor are needed to implement adaptivity efficiently.

From these examples we may conclude that the acceptance of parallel computers requires a programming model that supports massive parallelism and that the software parallelism must exceed the hardware parallelism by far for adaptivity to become efficient.

Plenary Session: Parallelverarbeitung in den Ingenieurwissenschaften – Methoden und Anwendungen

Benedikt Grosser (Protokoll), Andreas Frommer (Diskussionsleitung)

Diskussionspunkte

1) Wer nutzt Parallelrechner?

2) Wer liefert die Technologie (Hardware und Software)?

3) Architekturen vs. Anforderungen

4) Wirtschaftliche und universitäre Perspektiven

5) Technologietransfer

zu 1) Es herrschte Einigkeit darüber, daß Parallelrechner auch in weiterer Zukunft die einzige Technologie darstellen werden, mit der die notwendigen Kapazitäten in Spitzenbereichen befriedigt werden können. Für eine breite Akzeptanz der Parallelverarbeitung in den Ingenieurwissenschaften muß ein Kompromiß zwischen der Leistungsfähigkeit und der einfachen Handhabung der entwickelten Werkzeuge gefunden werden. Auf der einen Seite steht die Vision, daß die Parallelität in Hard- und Software für den Anwender transparent ist. So kann er sich voll auf die Lösung seines Problems (Modellbildung, Simulation, etc.) konzentrieren, ohne sich um anwendungsfremde Fragen wie Auswahl geeigneter Architekturen, Bibliotheken und Compiler kümmern zu müssen. Um aber gleichzeitig die Anwendung mit einer vertretbaren Effizienz auf Parallelrechnern betreiben zu können, macht die hierfür erforderliche Beschleunigung der Verfahren oft einen großen Anteil aus. Beklagt wurde in diesem Zusammenhang, daß sich Optimierungen häufig auf spezifische Hardware- und Compilergegebenheiten beziehen und dabei die Algorithmik vernachlässigt wird. Parallelverarbeitung als Gegenstand der Forschung kann hier nützlich sein, benötigt aber Rückmeldung seitens der ,,mündigen" Anwender.

zu 2) Die Diskussion in diesem Punkt konzentrierte sich auf den Softwarebereich. Staatliche und EU-Förderung für Software-Portierungen auf Parallelrechner sind wesentlich. Beispiele zeigen auch, daß von industrieller Seite die Parallelisierung kommerzieller Codes zunehmend verlangt wird. Es wurde darauf abgehoben, daß sich für den Transfer von paralleler Technologie Projekte anbieten, in denen unter Einbeziehung der Universitäten, Hersteller und Anwender kommerzielle serielle Software parallelisiert wird.

zu 3) Das Spannungsfeld Architekturen vs. Anforderungen führt zu deutlich unterschiedlichen Kompromißlösungen. Anwender mit enormem Rechenbedarf sind z.B. in der Theoretischen Elementarteilchenphysik dazu bereit, auf Special Purpose Rechnern zu vergleichsweise billigem Hardwarepreis vergleichsweise hohe Manpower zur Entwicklung von Software und deren Optimierung zu investieren. In den Ingenieurwissenschaften stehen dagegen typischerweise Fragen nach Portabilität und Bedienungskomfort im Vordergrund. Hier ist es Aufgabe der Informatik, Schnittstellen zu schaffen, die von der Parallelrechnerarchitektur abstrahieren. Umstritten bleibt dabei die Frage nach der notwendigen oder erreichbaren Effizienz.

zu 4),5) Die Diskussion zu diesen beiden Fragen vertiefte sich auf die Rolle der Universitäten als Lieferanten von Absolventen mit Parallelrechner Know-How. Es wurde deutlich, daß die Parallelverarbeitung in der Lehre stärker betont werden sollte. Möglichkeiten hierzu sind sowohl Vorlesungen als auch Praktika. Problematisch ist die Einbettung in die Lehrplangestaltung. In vielen Vorlesungen bietet es sich jedoch an, Parallelisierung bestimmter Verfahren stärker zu thematisieren. Ergänzend dazu können Programmierpraktika angeboten werden. Wichtig für die Attraktivität bei den Studenten sind zeitlich überschaubare Projekte, inhaltlich sinnvolle Anwendungen und eine stabile Entwicklungsumgebung. In diesem Zusammenhang wurde erwähnt, daß Qualifikationen in Parallelverarbeitung von Unternehmen zunehmend konkret nachgefragt werden.