47

Analysis of Substructuring in a Metal Forming Process

A.Müller[1], P.Adamidis[2]

Introduction

Many metal forming processes are solved by the Finite Element method. Due to the high demands in computing time and storage these methods should be parallelized especially for nonlinear use. The substructure method is an integrated approach which allows one to parallelize on the one hand the element loop and on the other hand the solution process. By decomposition of the element graph, the work is distributed homogeneously between the processors. The use of a proper numbering scheme for the nodes gives the system matrix the shape of an arrow matrix which is well suited for parallel solution.

For this representation we regard the process of twin–roll casting [ABBL+98]. Getting results in a reasonable response time is essential to make progress in understanding complex processes which are, for example, needed in reducing the number of production steps of manufacturing steel strips [Jes95].

Sequential Algorithm

The metal forming process is analysed by using the Finite Element system Larstran. This solution system has originally been developed at the University of Stuttgart

1 University of Stuttgart, Institute of Parallel and Distributed Systems, Breitwiesenstrasse 20–22, D–70565 Stuttgart, Germany, current affiliation: LBNL, NERSC, One Cyclotron Road; Berkeley, 94720 California, USA, email: Amueller@lbl.gov
2 same as above, current affiliation: HLRS, Allmandring 30, 70550 Stuttgart, Germany
Eleventh International Conference on Domain Decomposition Methods
Editors Choi-Hong Lai, Petter E. Bjørstad, Mark Cross and Olof B. Widlund

especially for the simulation of nonlinear behaviour of material (in strain–stress and strain–displacement relations) and geometry (i.e. major displacements and rotations). The system is now maintained and distributed by Lasso Consulting Engineers. Further development is also conducted by the Institute for Metal Forming at the RWTH Aachen. The system has proved successful in practical use and is installed at a number of industrial customer locations.

Larstran is an implicit code. The calculations are executed in small increments. Equilibrium of loads and (nonlinear) material reaction is traced by an iteration in each increment. For the simulation of rolling, a thermomechanical algorithm for slow viscous motion problems, coupled with transient heat flow processes is used. Both the mechanical and the thermal problems are iterated to equilibrium in a staggered solution for each increment. The flow–chart is shown in figure 3.

In this application, the two–dimensional elements are quadrilateral plane strain elements. For the constitutive law, a superplastic approach is applied. The contact problem is solved by penalty techniques, i.e. boundary nodes of the workpiece are traced against penetration into the roll's surface, and in the case of penetration, correction terms of high order are introduced into the stiffness matrix and the force vector. The direct solution of the linear equation systems is performed by an LU-decomposition with a skyline storage scheme of the coefficient matrix [Ort88]. All involved matrices are symmetric and positive definite.

Parallelization

Parallelization Strategies

An analysis of the computation time of the Finite Element package shows that most of the computational time is spent in the calculation of the entries in the system matrices and in the solving modules (more than 95%). It is clearly seen, by Amdahl's law, that the parallelization of the solver of the linear system is not sufficient even for large problems. For a parallelization strategy, the element calculation for the system matrices has to be accounted for.

Two elementary strategies can be regarded to parallelize the Finite Element system. One strategy is to parallelize the modules independently from one another, offering the advantage of a possibly independent modular software development for parallelization. Another approach is the coordinated parallelization of the two computationally time consuming parts. The coordination of the element–loop and the solving of the linear system can be realized by the substructuring method [SBG96].

Substructuring

We start with an unstructured grid and a global numbering as usually used in finite element programs. Such grids provide information about the coordinates of the nodes, the topology of the nodes and the assignment of the nodes to elements. A principle example is illustrated in figure 4.

From this information, an element graph is created whose nodes are the elements and whose edges are between nodes of the element graph iff the elements have one or

more nodes in common in the original grid of nodes.

In the second step, each processor decomposes the element graph in the same way, using a partitioning method, as given by the software package *metis* [KK97] for example. The partitioning tools aim at a homogeneous distribution of the given graph with the constraint of a heuristic minimization of the number of interface grid points (boundary points) between the different partitions. Because of the homogeneous distribution, the load is evenly balanced, and the minimization aims at a possibly low cost for the explicit communication using a parallel machine with distributed memory. After this, all the information about the element graph and its decomposition between different processes is available to each processor. The chosen number of partitions is the same as the number of involved processors. With this element partitioning, we can easily assign the nodes belonging to a processor. One node can belong to more than one partition.

Each processor now separates the grid points belonging to its partition into inner grid points – lying only in its partition – and into interface (boundary) grid points – belonging also to another partition. After this, each processor defines its local numbering for its inner grid points, to which it has been assigned to, starting from local number 1. The interface grid points are numbered globally starting from the maximum of all inner grid points of all partitions plus 1 up to the number of the interface grid points. The numbering of the boundary grid points is done globally because of the common use of these grid points by different processors.

Next, the calculation of the contributions of the elements is done in parallel. Each processor calculates the elements it is responsible for. We get the following local parts for the system matrix for each processor:

$$\begin{pmatrix} A_{II}^i & A_{IB}^i \\ A_{BI}^i & A_{BB}^i \end{pmatrix}, i = 1, \ldots, n. \tag{1}$$

In (1) I stands for inner and B for boundary, and n is the number of partitions, which is also the number of processors.

The global system matrices have the form of arrow matrices given in (2)

$$A = \begin{pmatrix} A_{II}^1 & & & & A_{IB}^1 \\ & A_{II}^2 & & & A_{IB}^2 \\ \cdots & & \ddots & & \vdots \\ & & & A_{II}^n & A_{IB}^n \\ A_{BI}^1 & \cdots & \cdots & A_{BI}^n & A_{BB} \end{pmatrix} \tag{2}$$

with

$$A_{BB} = \sum_{i=1}^{n} A_{BB}^i. \tag{3}$$

This global form does not exist explicitly on the parallel system, but rather in parts, as given in (1).

The compilation of the matrices A_{II}^i, A_{IB}^i, A_{BI}^i and A_{BB}^i can be calculated completely independent of each other in parallel. So, with the exception of A_{BB}, the other submatrices of (1) are calculated locally and are, consequently, available in

the local memory – no communication is needed. The sum in (3) has to be generated by a reduce–operation.

The system (2) can be also written in the form

$$\begin{pmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{pmatrix}.$$

So the solution of the system

$$A \begin{pmatrix} x_I \\ x_B \end{pmatrix} = \begin{pmatrix} b_I \\ b_B \end{pmatrix} \tag{4}$$

decomposes into the two parts

$$A_{II}x_I + A_{IB}x_B = b_I \tag{5}$$

$$A_{BI}x_I + A_{BB}x_B = b_B. \tag{6}$$

Under the assumption of regularity of the matrices A_{II}, we form (6) - $A_{BI}A_{II}^{-1}$ (5) and get

$$\hat{A}x_B = \hat{b} \tag{7}$$

with the Schur complement $\hat{A}$

$$\hat{A} = A_{BB} - A_{BI}A_{II}^{-1}A_{IB} \tag{8}$$

$$\hat{b} = b_{BB} - A_{BI}A_{II}^{-1}b_I. \tag{9}$$

Due to the shape of the arrow matrix the calculation of the inner degrees of freedoms can be parallelized in a straight forward way, which is described in [MA98]. Here we will present comparisons of sequential and parallel solution of the Schur complement system. For the parallel solution of the Schur complement we have used the general direct solver of the public domain package ScaLAPACK [EEE+97]. The matrices have high conditioning numbers. For the solution of such systems the use of direct methods is more suitable.

Results

The described substructuring method applied to the nonlinear Finite Element Method has been implemented on the parallel computer IBM RS/6000 SP with local memories. A software–technical demand was a far reaching use of existing software of Larstran and the use of existing standard libraries. For basic linear algebra operations, the BLAS libraries, which are available on all major platforms, have been used. The standard message passing library MPI [SHLWD96] has been used. The LU–decomposition and the serial solution of the Schur complement have been executed by the skyline–solver of Larstran. For the parallel solution of the Schur complement the solver for general matrices of ScaLAPACK with a block cyclic distribution has been chosen [EEE+97].

A comparably low degree of freedom is established by the nonlinearity of the simulated process of metal forming. Within each increment, several thermomechanical coupled iterations are executed. Each thermomechanically coupled iteration consists of up to several hundreds of thermal and mechanical iterations. Real world simulation studies need days of computation time at a workstation.

For the analysis of the parallelization, a two–dimensional model has been chosen. In the thermal calculation, to each node one degree of freedom is assigned. In the mechanical calculation, two degrees of freedom are assigned to most of the nodes. Some nodes at the boundary use suppressed values in the mechanical part, i.e. the number of degrees of freedom for the mechanical calculation is lower than twice the number of nodes. Due to the vertical symmetry of the problem, one half of the strip in figure (2) is modeled. So no horizontal displacement of the grid points laying at the symmetric axis is allowed. The model size is given in table 1.

Table 1 model size

elements	nodes	degree of freedom mechanical/thermal
3600	4207	7813/4207

Due to the fixed topology of the grid, the computation of both, the mechanical and the thermal iteration is constant. Therefore we only compare single iterations between different runs parameterized by the number of processors.

The main computational portions are the calculation of the entries for the system matrix (element calculation and assembly) and the solution of the linear systems. In table 2, time portions of a sequential calculated coupled iteration are listed.

Table 2 Sequential time portions (in seconds) of the time consuming parts

mechanical iteration		thermal iteration	
solver	0.17	solver	0.05
element	4.7	element	4.3

In the two–dimensional example under consideration, most of the time is spent in the element loops. In figure 5, the reduction of time to build up the system matrices is shown. Because of faster access to smaller data sets a super–linear speed–up was reached.

Results of time–measurement for solving the linear systems in a mechanical, and thermal, iteration are demonstrated in figure 6 for different numbers of processors. In figure 7 the total computational time of the simulation is shown. The comparison of a sequential and a parallel solver show that the use of a parallel solver for the Schur complement solution is for a moderate number of processors, slightly better than the sequential solution.

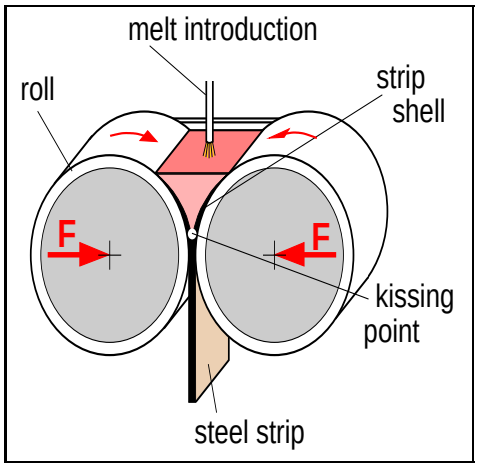

Figure 1 Principle of twin–roll casting process

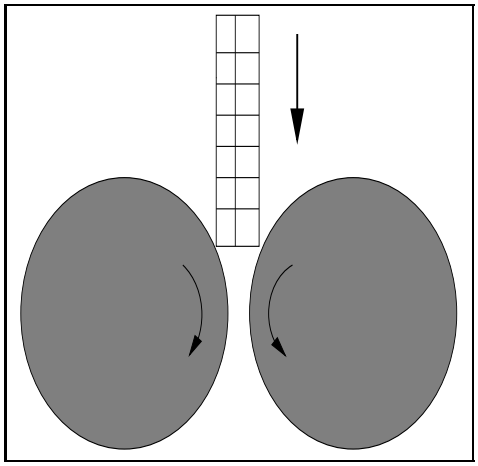

Figure 2 Initial grid

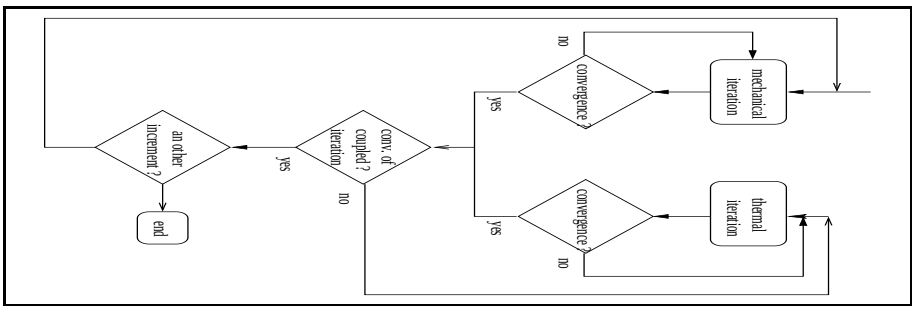

Figure 3 Flow chart of the coupled algorithm

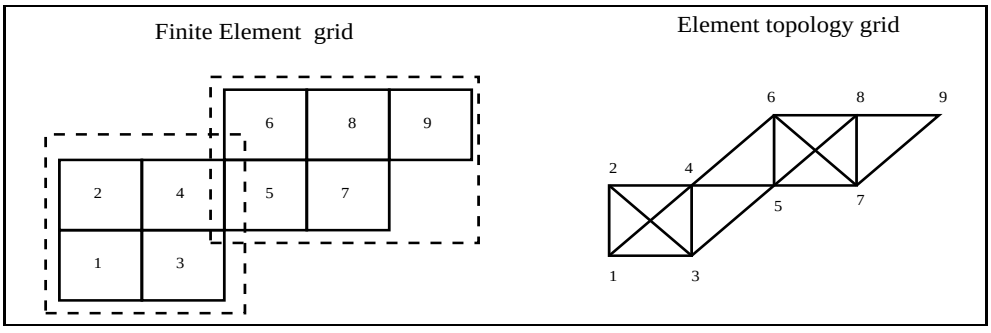

Figure 4 Unstructured element grid and associated element topology grid

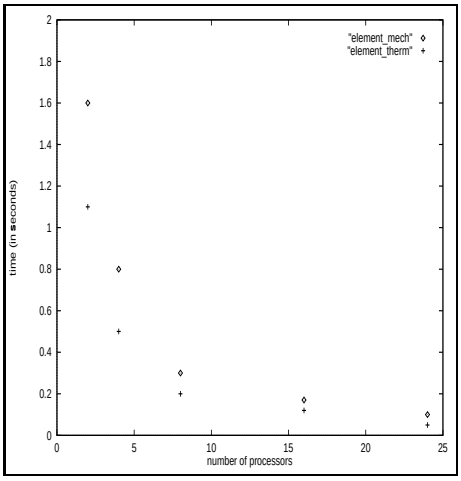

Figure 5 Time for calculation of the entries in the mechanical/thermal system matrices

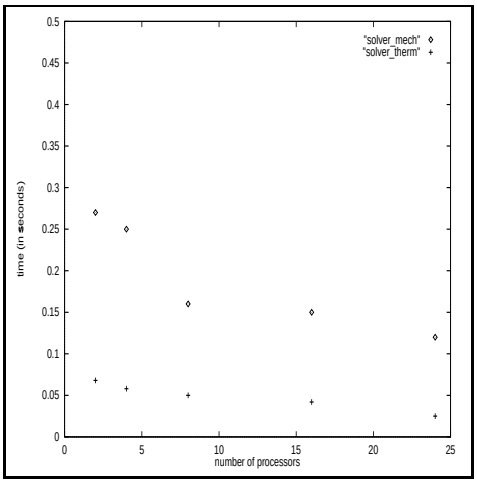

Figure 6 Time for the solution of linear systems in a mechanical/thermal iteration

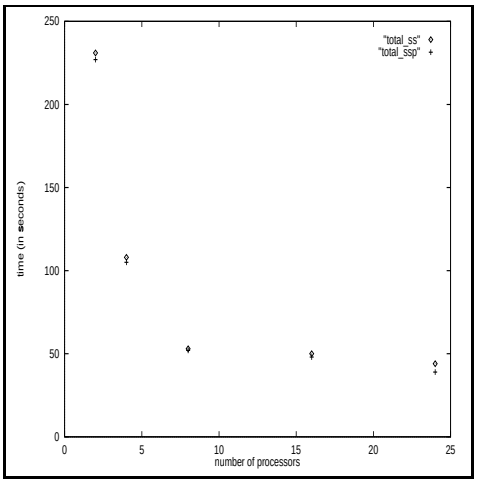

Figure 7 Comparison of the total runtime with a sequential and parallel solution of the Schur complement system

Further Work

Parallelization of three–dimensional modeling of this problem is currently under consideration, for this and other modeling tasks. Furthermore, a comparison with overlapping domain decomposition (Schwarz–methods) is in progress.

Acknowledgements

We are grateful to our colleagues Achim Beck, Yun Ding, Manfred Münch, University of Stuttgart; Matthias Laux, Christoph Pospiech, IBM and Artur Kurz, Lasso Consulting Engineers for their kind support. We thank the Scientific Supercomputing Center Karlsruhe for the possibility of using the IBM RS/6000 SP. The work has been performed in the GRISSLi project, which was sponsored by the German Federal Ministry of Education, Science, Research and Technology under contract No. 01 IS 512 A-C/GRISSLi.

REFERENCES

[ABBL+98] Adamidis P., Beck A., Becker-Lemgau U., Ding Y., Franzke M., Müller H. H. M. L. A., Münch M., Reuter A., Steckel B., and Tilch R. (September 1–3 1998) Steel strip production - a pilot application for coupled simulation with several calculation systems. *Metal Forming 98* .

[EEE+97] (Editor) L. S. B., (Editor) G. H., (Editor) D. W., (Editor) J. D., (Editor) J. C., (Editor) A. C., (Editor) E. D., (Editor) J. D., (Editor) I. D., , (Editor) S. H., (Editor) A. P., (Editor) K. S., and (Editor) R. C. W. (1997) *ScaLAPACK User's Guide.*

[Jes95] Jestrabek J. (1995) *Stahlbandherstellung nach dem Zweirollenverfahren – Modellierung des Strömungs- und Temperaturfeldes.* PhD thesis, RWTH Aachen.

[KK97] Karypis G. and Kumar V. (1997) METIS:A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Technical report, University of Minnesota,Department of Computer Science / Army HPC Research Center.

[MA98] Müller A. and Adamidis P. (July 13-16 1998) Parallelization of a metal forming process by substructuring. *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'98)* .

[Ort88] Ortega J. M. (1988) *Introduction to parallel and vector solution of linear systems.* Plenum Press.

[SBG96] Smith B., Bjørstad P., and Gropp W. (1996) *Domain decomposition : parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press.

[SHLWD96] Snir S. W., Huss-Ledermann S., Walker D. W., and Dongarra J. (1996) *MPI The Complete Reference.* The MIT Press.