Efficient solution of large sparse linear systems

Diss. ETH No. 11105

Efficient Solution of Large Sparse Linear Systems

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Technical Sciences

presented by
ARNO LIEGMANN
Diplom-Informatiker, RWTH Aachen, Germany
born 22 March 1962
citizen of Germany

accepted on the recommendation of
Prof. Dr. W. Fichtner, examiner
Prof. Dr. W. Gander, co-examiner

1995

Zusammenfassung

In vielen numerischen Anwendungen in Wissenschaft und Technik ist die Lösung schwach besetzter linearer Gleichungssysteme eine Kernaufgabe, weil dort der weitaus grösste Teil der Rechnerressourcen benutzt wird, die das Applikationsprogramm benötigt. Daraus folgt, dass eine sorgfältige Wahl und Implementation der Lösungsmethoden getroffen werden muss, damit das Applikationsprogramm effizient ausgeführt werden kann.

Zur Lösung schwach besetzter linearer Systeme stehen iterative und direkte Methoden zur Verfügung. Iterative Methoden haben den Vorteil, dass sie in vielen Fällen weitaus weniger Speicherplatz benötigen als direkte Verfahren. Aus diesem Grund werden iterative Methoden oft vorgezogen. Auf der anderen Seite sind iterative Methoden nur nützlich, wenn sie auch konvergieren. Zudem ist es aus Effizienzgründen wünschenswert, dass die Konvergenz schnell erreicht wird, d.h. die Anzahl der Iterationen klein ist.

In den sogenannten schlecht konditionierten Fällen, in denen iterative Methoden gar nicht oder nur sehr langsam konvergieren, muss auf direkte Verfahren zurückgegriffen werden. Bei direkten Verfahren, die in der Regel auf Gauss-Elimination basieren, wird ein lineares System durch faktorisieren der Koeffizientenmatrix mit anschliessender Vorwärts- und Rückwärts-Substitution gelöst. Allerdings kann bei der Faktorisierung der Koeffizientenmatrix eine beträchtliche Anzahl zusätzlicher Nicht-Null Elemente (der sog. fill-in) erzeugt werden. Um diesen fill-in zu minimieren wird vor der Faktorisierung eine Permutation der Zeilen und Spalten der Matrix vorgenommen. In der Praxis haben sich Umordnungsalgorithmen wie der Minimum Degree oder der Multiple Minimum Degree Algorithmus als sehr effektiv erwiesen.

Ein effizientes direktes Lösungsverfahren muss aber neben der Reduzierung des fill-in auch noch schnell sein bei der Berechnung der Faktoren. In letzter Zeit wurden in dieser Richtung Fortschritte gemacht durch sogenannte Supernode-Techniken. Supernodes sind aufeinander folgende Zeilen und Spalten in den Faktoren, die dieselbe Nicht-Null Struktur aufweisen; sie werden während der symbolischen Faktorisierung erkannt. Supernodes erlauben die Verwendung von Kernroutinen, die ohne indirekte Adressierung auskommen. Dadurch kann die Anzahl der Speicherzugriffe mit indirekter Adressierung erheblich reduziert werden, was wesentlich zur Effizienz der Faktorisierung beiträgt.

Diese Arbeit konzentriert sich ausschliesslich auf die Verwendung von Supernode-Techniken zur Lösung schwach besetzter struktur-symmetrischer linearer Gleichungssysteme. Im ersten Teil werden unterschiedliche Faktorisierungsalgorithmen vorgestellt, die entweder spalten-orientiert oder block-orientiert sind. Bei spalten-orientierten Faktorisierungsalgorithmen wird jeweils eine Spalte (oder Zeile) eines Faktors mit Hilfe von Supernodes berechnet. Im Gegensatz dazu wird bei block-orientierten Faktorisierungsalgorithmen jeweils ein ganzer Supernode berechnet. Block Supernode Methoden führen zu einer weiteren Reduktion der indirekten Speicherzugriffe.

Im zweiten Teil der Arbeit wird die parallele Lösung schwach besetzter linearer Gleichungssysteme auf Rechnerumgebungen mit verteiltem Speicher betrachtet. Der Grund, warum eine Rechnerumgebung mit verteiltem Speicher gewählt wurde, ist der, dass die meisten Benutzer keinen Zugriff auf Multicomputer besitzen, wohl aber auf Workstation Cluster. Die Arbeit beschreibt einen parallelen Supernode-Löser, der als Master-Slave Modell implementiert ist. Zur Kommunikation wird PVM benutzt. Am Anfang wird die Matrix durch den Masterprozess in Blöcke zerlegt und auf die Slave-Prozessoren verteilt. Diese lösen das lineare System parallel unter Verwendung von Supernode-Techniken auf den lokalen Daten. Schliesslich sendet jeder Slave-Prozess sein Teilergebnis zum Master, der daraus den endgültigen Lösungsvektor erzeugt.

Im letzten Teil der Arbeit wird ein Hybridansatz zur Lösung mehrerer linearer Systeme betrachtet. Unter einem Hybridansatz ist in diesem Zusammenhang die Kombination von direkten Supernode-Techniken und iterativen Methoden zu verstehen. Währen eines numerischen Simulationsprozesses werden in der Regel viele lineare Systeme gelöst. Wenn iterative Methoden nicht angewendet werden können, müssen

in diesem Fall direkte Verfahren benutzt werden, wobei jeweils eine Faktorisierung für jedes lineare System erforderlich ist. Der Hybridansatz versucht durch Verwendung von iterativen Methoden so viele dieser rechenintensiven Faktorisierungen wie möglich einzusparen. Um die Erfolgswahrscheinlichkeit der iterativen Methoden zu erhöhen, wird jeweils die letzte Faktorisierung als Vorkonditionierer verwendet. Mit diesem Ansatz konnten in manchen Fällen bis zu 90% aller Faktorisierungen eingespart werden.

Abstract

In many numerical applications stemming from science and engineering the solution of sparse linear systems is a core task because it consumes the largest portion of the overall computing resources required by the application. Consequently, the solution techniques for sparse linear systems have to be carefully chosen and implemented in order to achieve an efficient application program.

Generally, for the solution of sparse linear systems one can use iterative or direct methods. Iterative methods have the advantage to require far less memory than direct methods in many cases. Thus, they are preferred in most applications. Nevertheless, iterative methods are only useful when they converge towards the solution. Furthermore, for efficiency reasons convergence is desired to be fast, i.e. the number of iterations needed should be small.

In those so-called ill-conditioned cases where iterative techniques fail to converge or convergence is very slow direct methods have to be used. Direct methods, mostly based on Gaussian elimination, solve a linear system by first factoring the coefficient matrix and then using these factors for forward and backward substitution. Unfortunately, when the coefficient matrix is sparse its factors can suffer from fill-in and their number of non-zero entries can grow rapidly. Thus, row and column permutations within the coefficient matrix are performed such that the fill-in is minimized during factorization. In practice, heuristic algorithms like the minimum degree and the multiple minimum degree algorithm proved to be very effective in this respect.

Besides reducing the fill-in an efficient direct method must be able to compute the factors as fast as possible. Recent progress in this direc-

tion was made by exploiting so-called supernodal techniques. During symbolic factorization, supernodes are identified as sets of consecutive rows or columns of the factors which share the same sparsity structure. This property allows to use dense linear algebra kernel routines during factorization. By this means, the amount of indirect memory references can be reduced significantly which makes sparse numerical factorization more efficient.

This thesis focuses exclusively on supernodal techniques for the solution of sparse structurally symmetric linear systems. First, various supernodal factorization algorithms are considered: column- and block-oriented forms. Column-oriented supernodal factorization uses supernodes to calculate one single column (and row) of the factors. On the other hand, block supernodal factorization uses supernodes to compute an entire supernode of the factors. The latter is a technique to further reduce indirect memory references.

A second major part of the thesis considers the parallel solution of sparse linear systems in a distributed-memory environment. The reason why a distributed-memory environment approach was chosen is simply the fact that the vast majority of users does not have access to multicomputers but workstation clusters. The thesis describes a parallel supernodal sparse linear solver implemented as a Master-Slave model using the PVM message passing library for communication. Initially, the master decomposes the coefficient matrix into blocks which are then distributed onto the slave processors. The slave processors solve the linear system in parallel using supernodal factorization on local data. Eventually, the slave processors send their partial results back to the master process which assembles the final result into the solution vector.

Finally, the thesis considers a hybrid solution approach, namely the combination of direct supernodal and iterative techniques for the solution of sparse linear systems. During numerical simulation usually many linear systems have to solved. If iterative methods do not work, direct methods have to be used which require a factorization for each linear system. The hybrid approach tries to avoid as many computationally expensive factorizations as possible by using an iterative method instead. To increase the likelihood of success for the iterative algorithm, the last factorization is used as a preconditioner. This approach has shown to save up to 90% of the factorizations.