



Doctoral Thesis

## Solution of large unsymmetric systems of linear equations

**Author(s):**

Pommerell, Claude

**Publication Date:**

1992

**Permanent Link:**

<https://doi.org/10.3929/ethz-a-000669614> →

**Rights / License:**

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Diss. ETH No. 9838

# **Solution of Large Unsymmetric Systems of Linear Equations**

A dissertation submitted to the  
SWISS FEDERAL INSTITUTE OF TECHNOLOGY  
ZURICH

for the degree of  
Doctor of Technical Sciences

presented by  
CLAUDE POMMERELL  
Dipl. Informatik-Ing. ETH  
born 30 June 1964  
citizen of Luxembourg

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

1992



CatE

# Abstract

The numerical solution of large systems of linear equations lies at the heart of many scientific computing efforts. Sparse systems with several hundreds of thousands of unknowns have to be solved today. Ill-conditioned systems with unsymmetric matrices whose irregular sparsity structures reflect irregularly refined discretization grids for partial differential equations are particularly difficult to solve. Dozens or hundreds of such systems occur in a single semiconductor device simulation, and the linear solver is the main time-consuming operation in this application.

Direct sparse solvers, based on classical triangular factorizations, cannot be used for very large matrices. Their storage requirements increase superlinearly with the problem size, and the growth factor increases with the dimensionality of the discretization grid.

Iterative linear solvers are a viable alternative, even if none of the current approaches suits all the requirements alone. All successful iterative methods approximate the solution by stationary points in sequences of Krylov subspaces. GMRES uses the minimum number of matrix-vector products to minimize the residual in a given Krylov subspace, but its storage requirements increase linearly with the iteration number. Truncated or restarted variants are much less effective. BiCG and variants thereof do not have such a minimization property, but are often more successful, and have constant and low storage requirements. In our experience, Bi-CGSTAB is currently the fastest and most robust method in this class. BiCG and its variants have a bad reputation because of breakdown and cancellation problems, but the rare real occurrences of these effects can be fixed through restarting.

The convergence of iterative method is strongly improved, and often even enabled by preconditioning. Approximate factorizations constitute a family of effective preconditioners. Incomplete factorizations without fill are fast preconditioners capable of solving most linear systems. Particularly ill-conditioned systems can only be solved with more powerful preconditioners, such as a new efficient approximate factorization based on numerical dropping. Nested iterative solvers are another new alternative.

Because of the huge resource requirements, iterative solvers have to run on the most powerful supercomputers available. The efficient vectorization, parallelization, and balanced distribution of operations like regular and transposed matrix-vector multiplication with irregular sparsity structures, the solution of sparse triangular systems in incomplete factorization preconditioners, and the set-up of approximate factorizations, require sophisticated data structures. The techniques to exploit the different high-performance features of vector and parallel computers with shared or distributed memory are based on graph theory concepts, such as mapping, matching, and coloring. Flexibility, portability, and automatic adaptation can be achieved without efficiency penalty in an object-oriented software design for an iterative solver package.

The actual performance of iterative solvers on supercomputers depends mostly on aspects of the memory system, like indirect addressing, memory bandwidth, interconnection network latency, and storage capacity, and varies with the characteristics of the linear system. The fastest method on one platform may not be the most efficient on another computer.

# Zusammenfassung

Die numerische Lösung großer linearer Gleichungssysteme gehört zu den Kernproblemen vieler Projekte des wissenschaftlichen Rechnens. Heutzutage müssen schwachbesetzte Systeme mit mehreren hunderttausend Unbekannten gelöst werden. Besonders schwierig gestaltet sich das Lösen von schlecht konditionierten Systemen mit unsymmetrischen Matrizen, deren unregelmäßige Struktur die ungleichmäßig verfeinerten Diskretisierungsgitter für partielle Differentialgleichungen widerspiegelt. Dutzende oder Hunderte solcher Gleichungssysteme treten in einer einzigen Simulation von Halbleiterstrukturen auf, und die Lösung linearer Systeme ist die zeitaufwendigste Tätigkeit in dieser Anwendung.

Direkte Lösungsverfahren für schwachbesetzte Gleichungssysteme, aufbauend auf der klassischen Zerlegung in Dreiecksmatrizen, können für sehr große Matrizen nicht eingesetzt werden. Ihr Speicherbedarf wächst stärker als linear mit der Problemgröße, und der Wachstumsfaktor nimmt mit der Dimensionalität des Diskretisierungsgitters zu.

Iterative Lösungsverfahren stellen eine gangbare Alternative dar, auch wenn keiner der gegenwärtigen Ansätze allein alle Bedürfnisse erfüllt. Alle erfolgreichen iterativen Methoden nähern sich der Lösung über stationäre Punkte in Folgen von Krylov-Unterräumen. GMRES braucht die minimale Anzahl Matrixvektorprodukte, um das Residuum in einem bestimmten Krylov-Raum zu minimieren, der Speicherbedarf steigt aber linear mit der Anzahl Iterationen an. Bei Abschneiden oder Neustarts verlieren GMRES-artige Methoden schnell an Wirkung. Varianten von BiCG haben keine solche Minimierungseigenschaften, sind aber oft erfolgreicher und haben konstanten, niedrigen Speicherbedarf. Innerhalb dieser Klasse ist nach unserer Erfahrung Bi-CGSTAB das derzeit schnellste und robusteste Verfahren. BiCG und Varianten haben einen schlechten Ruf aufgrund der Möglichkeit frühzeitigen

Abbrechens oder Auslöschungsfehlern, aber in der Praxis kann das seltene Vorkommen solcher Erscheinungen durch Neustarts behoben werden.

Die Konvergenz von iterativen Verfahren wird durch Vorkonditionierung stark verbessert, und manchmal sogar erst ermöglicht. Unvollständige Zerlegungen bilden eine Familie von wirksamen Vorkonditionierern. Schnelle Vorkonditionierer erhält man durch unvollständige Zerlegungen ohne Füllen ursprünglicher Nulleinträge. Außergewöhnlich schlecht konditionierte Gleichungssysteme können nur mit leistungsfähigeren Vorkonditionierern gelöst werden, zum Beispiel durch eine neue angenäherte Zerlegung mit numerisch begründeter Vernachlässigung von Matrixeinträgen. Verschachtelte iterative Lösungsverfahren bieten eine weitere neue Alternative.

Aufgrund des hohen Bedarfs an Rechen- und Speicheraufwand müssen iterative Löser auf den leistungsfähigsten zur Verfügung stehenden Superrechnern laufen können. Die effiziente Vektorisierung, Parallelisierung und ausgeglichene Verteilung von Operationen wie die reguläre und die transponierte Matrixvektormultiplikation mit unregelmäßigen Strukturen, die Lösung von schwachbesetzten Dreieckssystemen in auf unvollständiger Zerlegung aufbauenden Vorkonditionierern und der Aufbau von angenäherten Zerlegungen verlangt den Einsatz komplizierter Datenstrukturen. Die Algorithmen, welche die Ausnutzung der Hochleistungskomponenten von Vektor- oder Parallelrechner mit gemeinsamen oder verteiltem Speicher ermöglichen, beruhen auf Konzepten der Graphentheorie, wie Abbildung, Zuordnung und Färbung. Flexibilität, Portierbarkeit und automatische Anpassung können ohne Effizienzeinbußen mit Hilfe einer objektorientierten Programmentwicklung in einem Paket von iterativen Lösern erreicht werden.

Die tatsächliche Leistung iterativer Lösungsverfahren auf Superrechnern steht hauptsächlich im Zusammenhang mit den Eigenschaften des Speichers, wie indirekte Adressierung, Bandbreite, Netzwerklatenz und Kapazität, und ändert sich mit der Struktur der linearen Gleichungssysteme. Das schnellste Verfahren auf dem einen Rechner kann unter Umständen auf einer anderen Maschine nicht mehr das effizienteste sein.