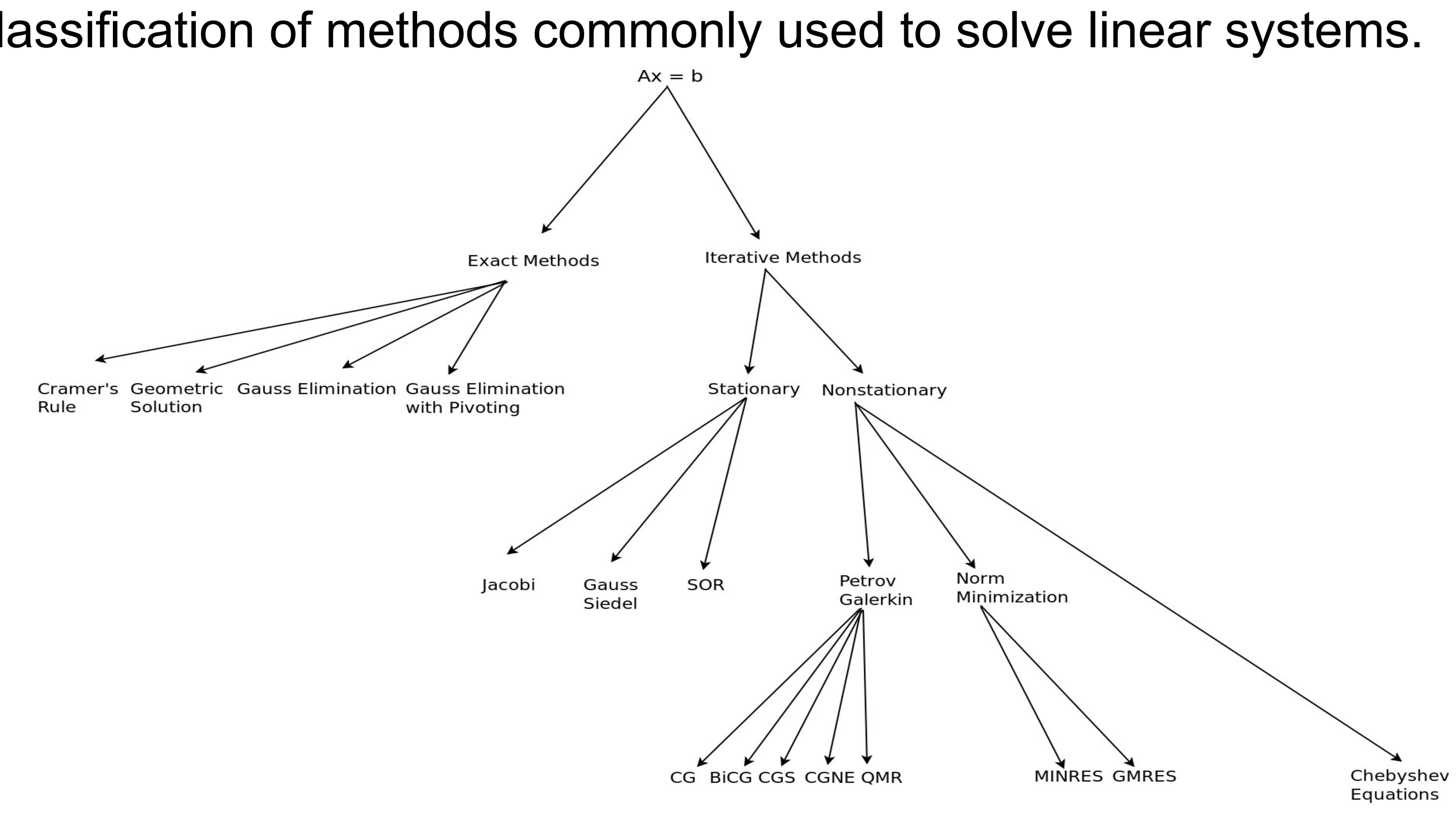
Numerical Survey of Krylov Subspace Methods and Parallel Computing for Solving Non-Hermitian Linear Systems

Nitish Keskar and Murugesan Venkatapathi

Methods to Solve Linear Systems

Often in scientific computation, there is a need to solve linear systems which possess no special forms such as sparsity or symmetry. In such cases, the traditional methods of Gaussian Elimination fail to converge in an practically feasible time-frame. To solve such problems Krylov Subspace Methods (KSM) are used. The figure below shows the classification of methods commonly used to solve linear systems.



We will primarily deal with QMR and CGNE owing to its ability to solve non-hermitian matrices. Their convergence has always been studied empirically . On the same lines, we have attempted to conduct numerical experiments to ascertain the suitability of these algorithms given the structure and size of the input matrix.

An Overview of Krylov Subspace Methods

The idea for finding solution for dense system of the form Ax = b using KSM is to create a Krylov Subspace by multiplying a matrix A with a vector v.

For Instance : $K_{r,n} = x_0 + \text{span}\ (r , Ar , A^2r, A^3r \ldots A^{(n-1)}\ r)$

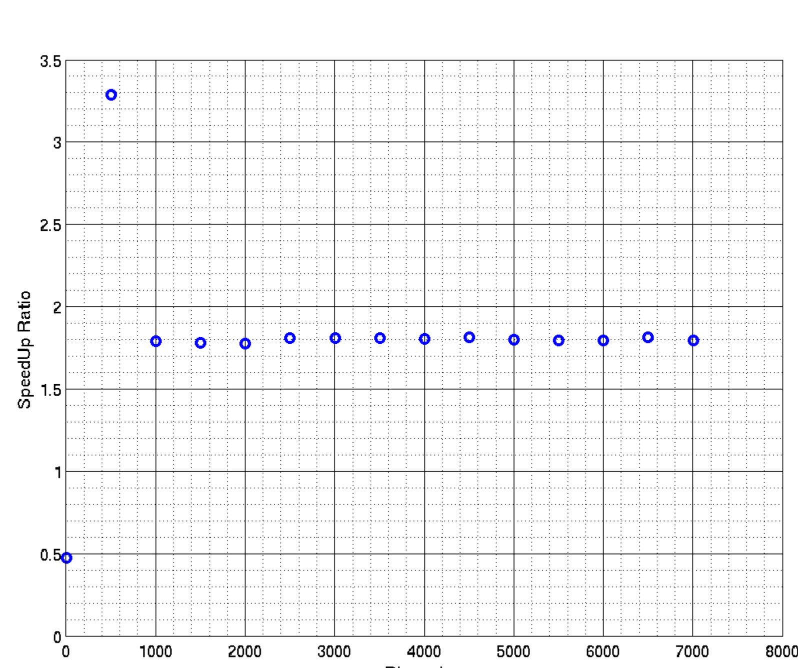
Is the Krylov Subspace of degree n created by r on A.

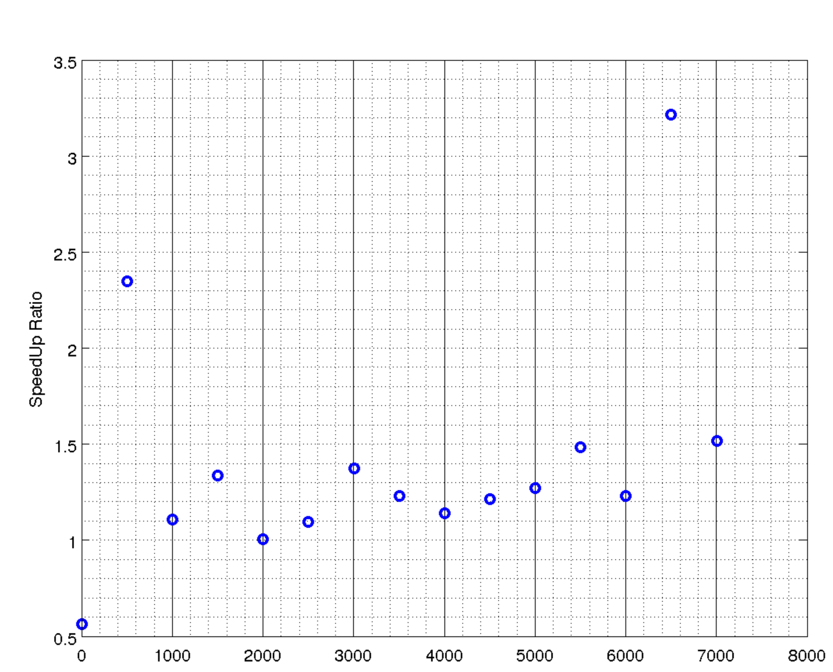
The goal of KSM is to successively find the best approximation of x in the newly created subspace formed by the matrix-vector multiplication. This condition of successive approximation can be converted to an iterative method through either the Petrov-Galerkin condition or the norm minimization condition as categorized above.

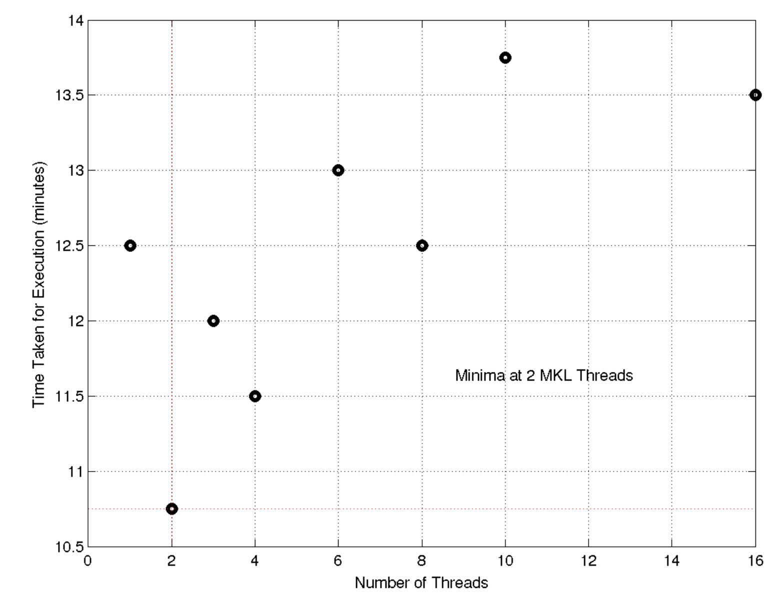
Parallelization

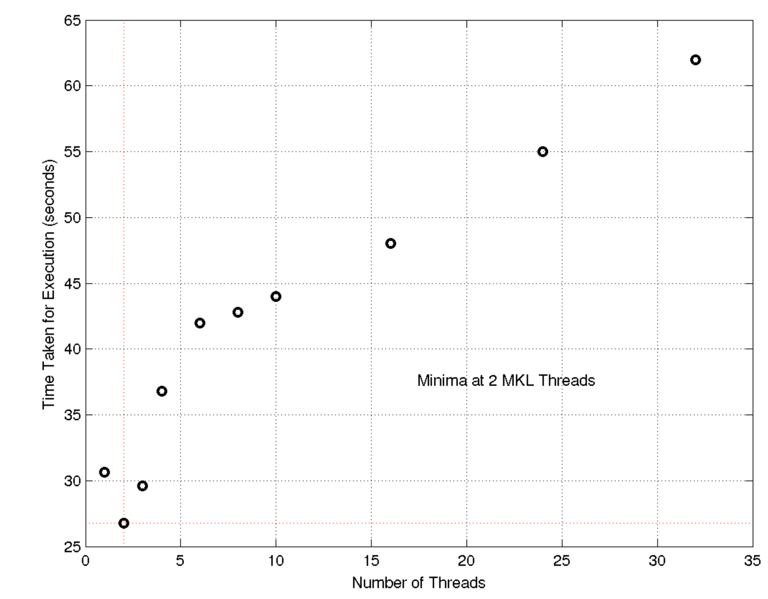
In the serial version of the code one iteration depends upon the values of the variables of the previous iteration, loop based parallelization is not possible . However, the linear algebra fundamental operations such as matrix-vector multiplication, dot product and 2-norms can be parallelized. The table below shows the number of (expensive) operations per iteration.

Algorithm	Dot Products	Matrix-Vector Multiplications
Steepest Descent	2	1
CG	2	1
BiCG	4	2
CGNE	2	2
QMR	5	2







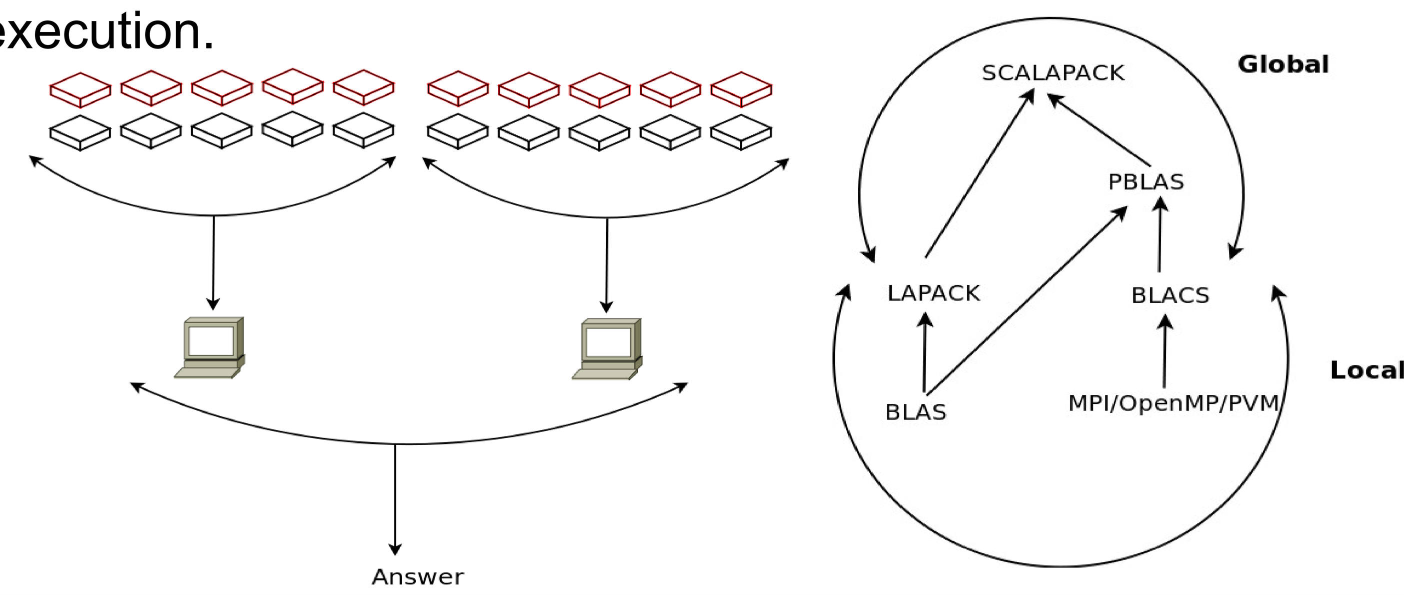


Parallelization Technique

Figures show the speed-up and scalability graphs of CGNE & QMR.

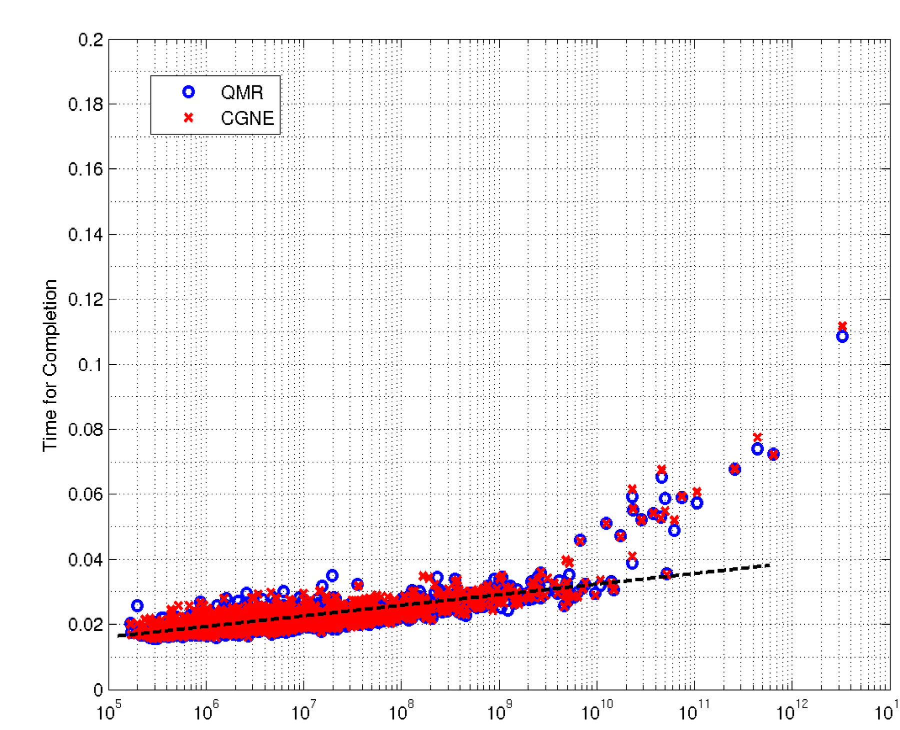
The scalability graphs are in accordance with Amdahl's law. The optimal choice of threads (2) is used to generate the scalability graphs.

Matrix-Vector, Dot Product and 2-Norm is parallelized using BLAS (Intel MKL) subroutines. Manual parallelization is used to further reduce the time of execution.



Analysis of Convergence

CGNE and QMR are 2 commonly used algorithms to solve Ax=b where A need not be positive definite. Although both methods are based on Petrov-Galerkin condition of Krylov Subspace, they differ in the operations they perform on A.



The graph above shows the behaviour of the algorithms with condition number of random matrices (N=75). The results show that there is no “clear winner” and both algorithms perform within the same time frame. It also shows an approximate linear relation between time and condition number.

The graph below on the left shows the behaviour of CGNE and QMR with condition number of random ill conditioned matrix (N=50). CGNE and QMR follow suit in the time needed for execution. Failure of one usually guarantees the failure of the other. The graph on the right shows the behaviour of CGNE and QMR with dimension. It establishes the threshold for QMR to converge faster than CGNE. However, the results are system specific since the analysis is for elapsed time which is a system dependent parameter.