An introduction to sparse linear system solvers

We have seen that we can efficiently represent large sparse matrices with suitable data structures. Moreover, we can efficiently evaluate matrix vector products if the sparse matrix is given as CSR format.

What is missing is a way to efficiently solve linear system with this data structure. We could attempt to use standard LU Decomposition (Gaussian Elimination). But the computational complexity is $O(n^3)$, making this method infeasible for very large sparse system. The issue is that standard LU decomposition does not take into account that most elements of a matrix are zero. There are different ways to overcome this issue.

- **Sparse direct solvers**. Sparse direct solvers are essentially variants of LU decomposition, but tuned for taken into account that most of the matrix consist of zero elements. Sparse direct solvers are highly efficient for PDE problems in two dimensions and still very good for many three dimensional problems. However, there performance deteriorates on matrices arising from complex three dimensional meshes.
- **Iterative Methods**. The most widely used iterative solvers are based on so-called Krylov subspace iterations. The idea is that the matrix is only known through its actions on vectors, that is we are allowed to use matrix-vector products only. A sequence of matrix-vector products is then used to build up a low-dimensional model of the matrix that can be solved efficiently and well approximates the solution of the original large linear system. Iterative methods are widely used in applications and can give almost optimal complexity in the number of unknowns. However, the problem is that the performance of iterative methods depends very much on certain properties of the matrix that reflect the underlying physical problem and so-called preconditioning techniques often need to be used to accelerate iterative solvers. These preconditioners can themselves be complex to develop for specific applications and are a topic of much research.
- **Multigrid Methods**. Multigrid methods follow a different idea. Here, starting from our discretisation we move to coarser ans coarser discretisation levels to refine the solution. Eventually, we are on a very coarse level on which the solution is trivial to accomplish. From there we refine again. Multigrid can be used as solver on its own or as preconditioner for iterative methods.

Software for sparse solvers

In the following we want to give a very incomplete overview of some frequently used software packages for the solution of sparse linear systems of equations.

Sparse direct solver packages

- UMFPACK (Part of Suitesparse) is a widely used sparse direct solver. It is built into Matlab and also available in Python through scikit-umfpack. It is very efficient and constantly being developed.
- Pardiso is available either directly under a closed source license or as part of the Intel MKL, with the caveat that the Intel MKL version is old and significantly slower than the directly available version.
- SuperLU is the standard sparse solver that is also built into Scipy. Scipy only offers the serial version of the library, which is sufficient for smaller to medium problems.
- Mumps is a massively parallel sparse direct solver. It is often used on parallel clusters.
- Amesos2 is part of Trilinos, a large collection of libraries for the parallel solution of partial differential equations. Amesos2 provides its own sparse direct solver, as well as interfaces to many other sparse direct solvers.
- Eigen is a templated C++ linear algebra library for dense and sparse operations. It provides its own sparse direct solver and also interfaces to many external solvers

Sparse iterative solvers

- Scipy has a good selection of sparse iterative solvers built in. For medium sized matrix problems it is a very good choice.
- Petsc is a parallel sparse solver library with a range of built-in iterative solvers.
- Belos is part of Trilinos and provides a number of parallel iterative solvers.
- Eigen not only has sparse direct but also several iterative solvers built in.

Multigrid solvers

- AmgX is an algebraic multigrid library for Nvidia GPUs.
- PyAMG is a Python based algebraic multigrid package.
- ML is the multigrid solver as part of the Trilinos package.

Most of the above packages are written in C/C++. But many of them have Python bindings. In the following sessions we will discuss sparse direct solvers, iterative solvers, and multigrid in more detail, and then give examples using some of the above software packages.

By Timo Betcke

© Copyright 2020.