Linear equation systems: iteration methods

Marcin Chrząszcz, Danny van Dyk
mchrzasz@cern.ch,
danny.van.dyk@gmail.com

University of Zurich UZH

Numerical Methods,
17 October, 2016

Linear eq. system

⇛ This and the next lecture will focus on a well known problem. Solve the following equation system:

$$A \cdot x = b,$$

⇛ $A = a_{ij} \in \mathbb{R}^{n \times n}$ and $\det(A) \neq 0$
⇛ $b = b_i \in \mathbb{R}^n$.
⇛ The problem: Find the x vector.

LU method with main element selection

$\Rrightarrow$ The algorithm for the LU matrix decomposition from previous lecture doesn't have the main element selection.

$\Rrightarrow$ This might be problematic. For example (backboard example)

Chelosky decomposition

$\Rrightarrow$ If $A \in \mathbf{R}^{N \times N}$ is a symmetric ($A^T = A$) and positively defined matrix, then there exits a special case of LU factorization such that:

$$A = CC^T$$

where C is a traingular matrixwith diagonal elements greater then zero.
$\Rrightarrow$ Finding the Chelosky decomposition is two times faster the finding the LU decomposition.
$\Rrightarrow$ The Chelosky decomposition has the same algorithm then the LU decomposition.

LDL factorization

$\Rightarrow$ If matrix can be Chelosky decomposed it can also be decomposed to:

$$A = LDL^T$$

where L is botton triangular matrix such that $\forall_i : l_{ii} = 1$ and D is diagonal matrix with positive elements.

$\Rightarrow$ The advantage of the LDL decomposition compared to Cehlosky decomposition is the fact that we don't need to square root in the calculations.

Iterative methods

⇛ The exact methods are the ones that require more computations to get the solutions.
⇛ Because of this they are not really suited to solve big linear systems.
⇛ The iteration methods are simple and the main goal of them is to transform:

$$Ax = b$$

to:

$$x = Mx + c$$

where A,M are matrices of $n \times n$ size.
b and c are vectors of the size n.
⇛ Once we get the second system (btw. remember MC lectures?) we can use iterative methods to solve them.

Expansion

$\Rrightarrow$ If $\overline{x}$ is the solution of the $Ax = b$ system then also:

$$\overline{x} = M\overline{x} + c$$

$\Rrightarrow$ We construct the a sequence that approximates the $\overline{x}$ in the following way:

$$x^{(k+1)} = Mx^{(k)}, \quad k = 0, 1, ...$$

$\Rrightarrow$ The necessary and sufficient requirement for the convergence of the set is:

$$\rho(M) < 1$$

Jakobi method

⇛ The simplest method is the Jakobi method.
⇛ We start from decomposition of A matrix:

$$A = L + U + D$$

where

- $L = (a_{ij})_{i>j}$ - triangular lower matrix.
- $D = (a_{ik})_{i=j}$ - diagonal matrix.
- $U = (a_{ij})_{i<j}$ - triangular upper matrix.

⇛ Now the new system:

$$(L + D + U)x = b$$

⇛ Translating them:

$$Dx = -(L + U)x + b$$

Jakobi method

⇛ Now the D matrix can be easily reverted (it is diagonal!):

$$x = -D^{-1}(L+U)x + D^{-1}b$$

⇛ So the iteration would have the form:

$$x^{(k+1)} = -D^{-1}(L+U)x^{(k)} + D^{-1}b$$

⇛ The matrix:

$$M_J = -D^{-1}(L+U)$$

is called the Jakobi matrix.

Jakobi method

⇛ The algorithm:

- Construct the matrix A
- Assume the precision of your calculations ϵ.
- Decomposite the A matrix into L, D, U.
- Calculate the Jocobi matrix M_J.
- Check the convergence of the method:
    - Calculate the $\rho(M_J)$.
    - Check the convergence conditions.
    - If both are ok then continue, else stop and go home ;)
- Choose the starting point $x^{(0)}$
- Calculate the $k+1$ point.
- Calculate the difference in each step:

$$\|x^{(k+1)} - x^{(k)}\|$$

- If above is smaller then ϵ the stop and assume $(k+1)$ is the solution.

Gauss-Seidle method

- A different iterative method is so-called Gauss-Seidle method.
- We start from the previous: $(L + D + U)x = b$.
- Write the eq. in form:

$$(L + D)x == Ux + b$$

- The $(L + D)$ matrix is triangular and can easily be inverted.
- From this one gets:

$$x = -(L + D)^{-1}Ux + (L + D)^{-1}b$$

- So the iteration equation will take form:

$$x_i^{(k+1)} = -\frac{1}{a_{ii}} \left(\sum_{j<i} a_{ij} x_j^{(k+1)} + \sum_{i>i} a_{ij} x_j^{(k)} \right) + \frac{b_i}{a_{ii}}$$

- The matrix:

$$M_{GS} = -(L + D)^{-1}U$$

is so-called Gauss-Seidl matrix.

Convergence of Gauss-Seidle and Jacobi methods

Reminder:

The necessary and sufficient condition for the iteration to be convergence is that:

$$\rho(M) < 1$$

where M is the matrix of a given method.

$\Rrightarrow$ Now calculating the $\rho(M)$ might be already a computationally expensive...
$\Rrightarrow$ A very useful way of getting the $\rho(M_J)$ is:

$$r_J = \frac{\|x_{k+1} - x_k\|}{\|x_k - x_{k-1}\|} \approx |\rho(M_J) - 1|$$

$\Rrightarrow$ Another useful equations:

$$\rho(M_{GS}) = \rho(M_J)^2$$

Convergence of Gauss-Seidle and Jacobi methods

Theorem:

If matrix (a_{ij}) fulfils one of the conditions:
$\Rightarrow$ $|a_{ii} > \sum_{j\neq i} |a_{ij}| \;\; i = 1, ..., n$, so-called strong row criteria.
$\Rightarrow$ $|a_{jj} > \sum_{i\neq j} |a_{ij}| \;\; j = 1, ..., n$, so-called strong column criteria.

Practical note:

$\Rightarrow$ One needs to note that calculating the $\rho(M)$ might be a as time consuming calculation as the solution is self...
$\Rightarrow$ If that is the case usually one neglects this steps and computes the solution with extra care at each step checking the convergence.

SOR method

⇛ The Successive Over-Relaxation method is an extension of the Gauss-Seidl methods.
⇛ The modification is that we can reuse the previous step to get a more precise approximation of the solution.
⇛ The "hack" happens when we calculate the $x_{GS}^{(k+1)}$ using the Gauss-Seidl method. Instead assuming that the next step is the $x_{GS}^{(k+1)}$ we "relax" the condition using liner combination:

$$x^{(k+1)} = \omega x_{GS}^{(k+1)} + (1-\omega)x^{(k)}$$

⇛ From the above once can see that if $\omega = 1$ then we end up with standard Gauss-Seidl method.
⇛ The iteration method equation has the form:

$$x^{(k+1)} = (D+\omega L)^{-1}((1-\omega)D - \omega U)x^{(k)} + (D+\omega L)^{-1}\omega b$$

SOR method, convergence

$\Rrightarrow$ The main difficulty in the SOR method is the fact we need to choose the ω parameter.

$\Rrightarrow$ One can easily prove that if the method is converging the $\omega \in (0,2)$.

$\Rrightarrow$ This is necessary condition but it's not enough!

$\Rrightarrow$ If the A is symmetric and positively defined then the above condition is also sufficient!

Conclusions

$\Rrightarrow$ We learned more exact method of solving the linear system.
$\Rrightarrow$ For big matrix we need to use iterative method which are not exact.
$\Rrightarrow$ My personal experience: Did not have big enough matrices to really need to apply the iterative methods but this is field dependent!

Backup