Methods for linear system Direct and Iterative

M.Sc. I.M. Manuel F. Mejía De Alba

Maestría en Modelado y Simulación
Universidad Central y Universidad Jorge Tadeo Lozano

Contens

- Introduction problems
- Basic systems
  - Diagonal, Triangular superior and inferior matrix
- Direct methods:
  - Gauss method and its variations
  - Special factorizations for direct methods
  - Matlab commands
- Iteratives methods:
  - Jacobi, Gauss Seidel and Successive Over Relaxation
  - Gradient methods
  - Matlab commands
- Homework

Introduction problem

It's a race!
You can run 0,2 km every minute. The Horse can run 0,5 km every minute. But it takes 6 minutes to saddle the horse. How far can you get before the horse catches you?

Can you show the problem like the system

$$\begin{pmatrix} 1 & 0{,}2 \\ 1 & 0{,}5 \end{pmatrix} * \begin{pmatrix} t \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ -3 \end{pmatrix}$$

The question here is: **What method has the best performance doing this job?**

Basic systems

- **Diagonal system**

$$x_i = \frac{b_i}{a_{i,i}}$$

- **Triangular inferior**

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} * x_j}{a_{i,i}}$$

- **Operations between rows**

$$\text{fila}i \leftarrow \text{fila}i - (a_{ik}/a_{kk}) * \text{fila}k$$

Gauss methods

The fundamental idea below of Gauss methods is generate a triangular equivalent system through simple algebraic operations between rows and solve it using the correct technique.

The error in this method is only by rounded. So we have to avoid the division by small numbers, for do that we have to change rows or columns:

- LU Factorization: If in the process was not necessary to change rows we can decompose the matrix A in two matrix: Lower triangular L and Upper triangular U. And split the problems: $Ly = b$ and $Ux = y$.
- Gauss using partial pivoting: Is the most eficient, you only have to find the better divisor in elements below the diagonal and in this column. Is the most eficient !!!
- LU=PA Factorization: Equal to LU but with partial pivoting, you have to solve: $z = Pb$, $Ly = z$ and $Ux = y$.
- Gauss using total pivoting: Find the best divisor in the submatrix $k \leq i, j \leq n$. Maybe won on precision but to a highly cost.
- Gauss Jordan: Find the best divisor in all matrix and make zeros to both sides of the diagonal, turn the matrix A in a identity matrix.

Matlab commands

- The easy way to solve linear system on Matlab is with the commands:

  ```
  x = A\b
  ```

  or

  ```
  x = linsolve(A,b)
  ```

  With `x = linsolve(A,b,opts)` is posible pass information about of matrix wich improve the solution.

- LU Factorization:

  ```
  [L,U,P] = lu(A)
  ```

- Cholesky factorization:

  ```
  U = chol(A)
  ```

  It works for positive definite matrix, this matrix are symmetric and the all eigenvalues are real and higher of zero.

  You can found all the information about how to solve linear system in the Matlab, using the help.

Jacobi, Gauss Seidel and Successive Over Relaxation

The main idea is evaluate each unknowns one by one, it is necessary an initial approximation.

The methods are iterative and needs a stop criteria, the most comun are:

$$|x^k - x^{k-1}| < tol \qquad \text{or} \qquad |b - A * x^k| < tol$$

- Jacobi: Evaluate all unknowns with the last values and then update the vector

$$x_i^{k+1} = x_i^k + \frac{b_i - A_i * x^k}{A_{i,i}}$$

- Gauss Seidel: Update the vector as soon as each unknown is evaluated

$$x_i^{k+1} = x_i^{k+1} + \frac{b_i - A_i * x^{k+1}}{A_{i,i}}$$

- SOR: Like GS but with a weight parameter

$$x_i^{k+1} = x_i^{k+1} + \omega \frac{b_i - A_i * x^{k+1}}{A_{i,i}}$$

Iterative methods in a matrix form

The general form the iterative methods could be write:

$$x^{k+1} = Mx^k + p$$

Each matrix could be express

$$A = D + L + U$$

- Jacobi:
$$M = -D^{-1}(L + U) \qquad p = D^{-1}b$$
- Gauss Seidel:
$$M = -(D + L)^{-1}(U) \qquad p = (D + L)^{-1}b$$
- SOR:
$$M = (D + \omega L)^{-1}((1 - \omega)D - \omega U) \qquad p = \omega(D + \omega L)^{-1}b$$

And this methods converge when the spectral radius $\rho(M) < 1$, $\rho(M)$ is the maximum absolute eigenvalue of the matrix M.

Matlab has the commands: `diag`, `triu`, `tril`, `max`, `abs`, `eig`.

Gradient based methods

- Under construction...

Homework

- Under construction...