B0° 21B5↩

Computational Methods in Ordinary Differential Equations

Dr Jon Shiach, Department of Computing and Mathematics, Manchester Metropolitan University

Indirect Methods for Solving Systems of Linear Equations

Learning outcomes

On successful completion of this page readers will be able to:

- Understand the concept of an hen used to solve a system of linear equations.
- Apply the
  Use the o determine the accuracy of the current estimate of the solution to the linear system.
  Determine whether an indirect method is or a particular linear system and anaylse the theoretical rate of convergence for indirect methods.

Indirect Methods

Indirect methods for solving systems of linear equations use an iterative approach to repeatedly update estimates of the exact solution to the linear system. They are called *indirect* methods since multiple applications of the method is required to calculate a solution unlike *direct* methods such as Gaussian elimiation and LU decomposition which require a single application to calculate the solution. However, direct methods are inefficient for large systems of equations for which we tend to using indirect methods instead.

An indirect method for solving a system of linear equations of the form $Ax = b$ is

$$x^{k+1} = Tx^k + c$$

where x^k is the current estimate of x, x^{k+1} is the improved estimate of x, T is an **iteration matrix** and c is some vector. This equation is iterated updating the values of the estimates such that $x^{(k)} \to x$ as $k \to \infty$. Note that unlike direct methods which will calculate the exact solution, indirect only calculate an estimate (albeit very close) of the exact solution.

The Jacobi method

The Jacobi method is the simplest indirect method. Splitting the coefficient matrix A into the of elements from the lower triangular, diagonal and upper triangaular parts of A to form matrices L, D and U such that $A = D + L + U$, e.g.,

The residual

The Jacobi method is applied by iterating equation (2) until the solution $x^{(k+1)}$ is accurate enough for our needs. Since we do not know what the exact solution is we can quantity the accuracy of an estimate by using the **residual** which is defined as

$$r = b - Ax^{(k)},$$

so that as $x^{(k)} \to x$, $r \to 0$. The convergence criteria used is

$$|r| < tol,$$

where tol is some small number. The smaller tol is, the closer $x^{(k)}$ is to the exact solution but this will require more iterations. In practice a compromise is made between the accuracy required and the computational resources avaialble. Typical values of tol are around 10^{-4} or maybe even 10^{-6}.

Example 1

Calculate the first iteration of the Jacobi method to solve the following system of linear equations and calculate the norm of the residual.

$$\begin{aligned} 4x_1 + 3x_2 &= -2, \\ 3x_1 + 4x_2 - x_3 &= -8, \\ -x_2 + 4x_3 &= 14. \end{aligned}$$

The Jacobi iterations are

$$\begin{aligned} x_1^{(k+1)} &= \tfrac{1}{4}(-2 - 3x_2^{(k)}), \\ x_2^{(k+1)} &= \tfrac{1}{4}(-8 - 3x_1^{(k)} + x_3^{(k)}), \\ x_3^{(k+1)} &= \tfrac{1}{4}(14 + x_2^{(k)}). \end{aligned}$$

Using starting values of $x^{(0)} = (0, 0, 0)^T$ the first iteration is

$$x_1^{(1)} = \frac{1}{4}(-2 - 3(0)) = -0.5,$$
$$x_2^{(1)} = \frac{1}{4}(-8 - 3(0) + 0) = -2,$$
$$x_3^{(1)} = \frac{1}{4}(14 + 0) = 3.5.$$

Calculate the residual

$$r^{(1)} = b - Ax^{(1)} = \begin{pmatrix} -2 \\ -8 \\ 14 \end{pmatrix} - \begin{pmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} -0.5 \\ -2 \\ 3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ -2 \end{pmatrix},$$

and the norm of the residual is $|r^{(1)}| = \sqrt{6^2 + 5^2 + (-2)^2} = 8.06226..$

Example 2

Write a MATLAB program to solve the system of linear equations from sing the Jacobi method ceasing iterations when $|r| < 10^{-4}$.

The function called elow is defined in the ection at the bottom of this page solves a linear system of equations defined by the arrays `A` and `b`. Iterations cease when $|r| < tol$ or `maxiter` is exceeded.

```
function x = jacobi(A, b, maxiter, tol)

% Calculates the solution to the system of linear equations Ax = b using
% the Jacobi method

% Initialise solution array
N = length(b);
x = zeros(maxiter + 1, N);

% Iteration loop
for k = 1 : maxiter

    % Calculate Jacobi method
    for i = 1 : N

        % Calculate sum
        for j = 1 : N
            if i ~= j
```