PH307: Introduction to Numerical Analysis

Tutorial Sheet 10

Note: This tutorial sheet deals with iterative methods of solving linear equations. You have to write the computer program corresponding to the last problem, and get it graded from your TA.

1. Consider the equation systems

   (a)

$$\begin{pmatrix} 1 & 0 & -0.25 & -0.25 \\ 0 & 1 & -0.25 & -0.25 \\ -0.25 & -0.25 & 1 & 0 \\ -0.25 & -0.25 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}.$$

   (b) $$\begin{pmatrix} 7 & 1 & -1 & 2 \\ 1 & 8 & 0 & -2 \\ -1 & 0 & 4 & -1 \\ 2 & -2 & -1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 \\ -5 \\ 4 \\ -3 \end{pmatrix}$$

   Using starting vectors as $(0, 0, 0, 0)$, and $(1, 1, 1, 1)$, perform a few iterations of (a) Jacobi method, and (b) Gauss-Seidl method using a calculator. Which method leads to faster convergence?

2. Write a computer program which will solve a system of linear equations using:

   (a) Point-Jacobi method

   (b) Gauss-Seidl method.

   The method to be used for solving the equations should be decided by the user, as also the starting vector X_0. Note that for the Gauss-Seidl method there is no need for you to invert any matrices. As mentioned in the lectures, same result can be achieved by the "back-substitution" process in each iteration. Those writing Fortran 90 codes may want to use powerful array features of the language such as the use of intrinsic function MATMUL to multiply a vector by a matrix. Additionally, if X

and Y are declared as arrays of same dimensions, their addition can be performed with the simple command $Z = X + Y$, instead of the long one

```
do i=1,n
Z(i)=X(i)+Y(i)
end do
```

You should consider the results to be converged when the norm $|D_{i+1}|$ of the difference vector $D_{i+1} = X_{i+1} - X_i$ is less than some convergence parameter ϵ (X_i is the solution vector at the end of the i-th iteration). Note that in Fortran 90, $|D_{i+1}|^2$ can be easily computed using the DOT_PRODUCT intrinsic function. You can use the linear equations of the previous problem to try out your code.