Norwegian University of Science and Technology
Department of Mathematical Sciences

TMA4205 Numerical Linear Algebra Fall 2015

Semester project – part 1: Matrix splitting methods

1 Consider the following model 1D boundary value problem (BVP) for the advection-diffusion differential equation:

$$\begin{aligned} -U_{xx} + aU_x &= f, \quad \text{in } \Omega = (0,1) \\ U(0) = 1, &\quad U(1) = -1, \end{aligned} \tag{1}$$

where $a = a(x)$ and $f = f(x)$ are given functions. We subdivide Ω into n identical subintervals of length $h = 1/n$ with end-points $x_j = jh$, $j = 0, \ldots, n$. We will seek a numerical approximation u_j of the solution $U(x_j)$ to (1) as a solution to the linear algebraic system, obtained from (1) by replacing the differential operators with their finite-difference approximations. The diffusive term will be approximated using a 3-point central scheme

$$U_{xx}(x_j) \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2}. \tag{2}$$

For the advective term we consider three possible alternatives:

$$U_x(x_j) \approx \frac{u_{j+1} - u_{j-1}}{2h}, \qquad U_x(x_j) \approx \frac{u_j - u_{j-1}}{h}, \qquad \text{or} \qquad U_x(x_j) \approx \frac{u_{j+1} - u_j}{h}. \tag{3}$$

These choices correspond to central, backward, and forward finite differences. Substituting (2) and one of the choices in (3) into (1) evaluated (collocated) at $x = x_j$, we arrive at a sparse linear system of equations $Au = b$, where $u = (u_1, \ldots, u_{n-1})^{\mathrm{T}}$ is the vector of unknowns; note that $u_0 = U(0)$ and $u_n = U(1)$ are given for this BVP. In our case A and b are of the form

$$A = \begin{bmatrix} \alpha_1 & \delta_1 & & & \\ \gamma_2 & \alpha_2 & \delta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{n-2} & \alpha_{n-2} & \delta_{n-2} \\ & & & \gamma_{n-1} & \alpha_{n-1} \end{bmatrix}, \qquad b = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-2} \\ \beta_{n-1} \end{bmatrix},$$

Let β_j be expressed as $\beta_j = f(x_j) + \tau_j$, where τ_j accounts for the boundary contributions.

a) Put $a(x_j) = a_j$. Write down the expressions for α_j, δ_j, γ_j and τ_j for each of the choices in (3).

b) For every $j = 1, \ldots, n-1$ select the discretization of the advection operator in (3) in such a way that the resulting matrix A is guaranteed to be irreducibly row diagonally dominant. (According to Theorem 4.9 in [S] such a selection guarantees the convergence of Jacobi and Gauss–Seidel methods.) *Note*: we can vary the discretization of the advection term from one collocation point x_j to another depending on a_j and h! *Hint:* If we replace all non-zero entries in the matrix by one and view the resulting matrix as the adjacency matrix of a directed graph, the original matrix is irreducible if and only if the directed graph is strongly connected.

From now on we will utilize the forward finite difference approximation of the advection operator and assume $a = -2$, so $\alpha = \alpha_j$, $\delta = \delta_j$, and $\gamma = \gamma_j$ are independent of j.

c) Give an explicit formula for the eigenvalues of A.

Hint: Use the note "Eigenvalues of tridiagonal Toeplitz matrices", which can be found on the home page. No derivations are required.

We will now study the behaviour of the simple matrix-splitting methods for our problem. Let $A = D - E - F$, where D, $-E$, and $-F$ are respectively the diagonal, strict lower, and strict upper parts of A (see section 4.1 in [S]).

d) Consider the Jacobi iteration $u^{(k+1)} = G_J u^{(k)} + D^{-1} b$, where $G_J = D^{-1}(D - A)$. Starting from the result in **c)**, find the eigenvalues of the iteration matrix G_J. What is the spectral radius of G_J? What does Gershgorin's theorem say about the eigenvalues of G?

e) How would you expect the error $e^{(k)} = u - u^{(k)}$ between the kth Jacobi iterate $u^{(k)}$ and the solution u to $Au = b$ to behave as a function of k and n? In other words, if you double n, what must you do with k in order to get close to the same error $e^{(k)}$?

Hint: (i) G_J has $n-1$ distinct eigenvalues and therefore it is diagonalizable; (ii) you may find Taylor series expansions with respect to a small parameter $h = n^{-1}$ useful when providing estimates.

f) Consider again the problem (1) with exact solution given by $U(x) = \cos(\pi x)$. What is the corresponding right-hand side f? Let $n = 20$ and use Jacobi iteration to solve the corresponding discrete system with this choice of f. Define u_* to be the vector with entries $U(x_i)$, $i = 1, \ldots, n-1$, i.e. the continuous solution evaluated at the interior grid points. Define also $e_*^{(k)} = u_* - u^{(k)}$ and plot $\log(\|e_*^{(k)}\|_\infty)$ as a function of k. Iterate until the error $e_*^{(k)}$ no longer changes. Next, increase n to 40, and repeat the solution process. Finally, do it with $n = 80$. Compare the convergence behaviour for all three cases (e.g. in one single plot). Are the results as expected? Can you explain your observations?

Hint: $u_* - u^{(k)} = (u_* - u) + (u - u^{(k)})$

g) In the notation of **f)**, put $n = 40$ and compare the behavior of Jacobi and forward and backward Gauss–Seidel methods:

$$
\begin{aligned}
u_J^{(k+1)} &= D^{-1}(E+F)u_J^{(k)} + D^{-1}b, \\
u_{fGS}^{(k+1)} &= (D-E)^{-1}Fu_{fGS}^{(k)} + (D-E)^{-1}b, \quad \text{and} \\
u_{bGS}^{(k+1)} &= (D-F)^{-1}Eu_{bGS}^{(k)} + (D-F)^{-1}b,
\end{aligned}
$$

by for example plotting $\log(\|e_*^{(k)}\|_\infty)$ for all the methods as a function of k on the same graph.

Could you provide a "physical" explanation as to why one version of Gauss–Seidel outperforms the other on this problem?

h) Let I_O and I_E denote respectively the odd and even indices in $1, 2, \ldots, n-1$. Let us reorder the unknowns u as $[u_{I_O}, u_{I_E}]$, and similarly reorder the equations as $[b_{I_O}, b_{I_E}]$. Show that after the reordering the matrix A will have the following block-structure:

$$
\begin{bmatrix} D_O & A_{OE} \\ A_{EO} & D_E \end{bmatrix},
$$

where D_O and D_E are diagonal matrices.

Identify the blocks D_O, D_E, A_{EO}, A_{OE}, and implement a Gauss-Seidel iteration (forward or backward) for the reordered system. Note that one can utilize vectorization in Matlab to "simultaneously" (in parallel on a multi-core system) update the unknowns in each block u_{I_O} and u_{I_E}. Thus, every Gauss-Seidel iteration should contain two vectorized updates of block-unknowns.

Vizualize the behaviour of this method as you have previously done in **g)**. Compare the performance of this method with the ones you have tested in **g)**.