NTNU

TMA4215 Numerical Mathematics — Fall 2016

Problem Set 7

1 Solve the linear system of equations $A\boldsymbol{x} = \boldsymbol{b}$, where

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 0 \\ 4 & 20 & 8 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 3 \\ 20 \end{bmatrix},$$

by naïve Gauss-elimination and Gauss-elimination with partial pivoting.

Solution. We display the result of each naïve GE-step on A. It is possible to place the multipliers in the free slots, and they are underlined for emphasis. Moreover, the current pivot elements are framed.

$$\begin{bmatrix} \boxed{1} & 2 & 1 \\ 3 & 4 & 0 \\ 4 & 20 & 8 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 & 1 \\ \underline{3} & \boxed{-2} & -3 \\ \underline{4} & 12 & 4 \end{bmatrix} \sim \begin{bmatrix} 1 & 2 & 1 \\ \underline{3} & -2 & -3 \\ \underline{4} & \underline{-6} & -14 \end{bmatrix}.$$

Hence, the LU decomposition of A has

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 4 & -6 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 & 2 & 1 \\ 0 & -2 & -3 \\ 0 & 0 & -14 \end{bmatrix}$$

Solving $A\boldsymbol{x} = \boldsymbol{b}$ is equivalent to first solving $L\boldsymbol{y} = \boldsymbol{b}$ and then $U\boldsymbol{x} = \boldsymbol{y}$. Using the techniques of forward and backward substitution yields $\boldsymbol{x} = (1, 0, 2)$. Alternatively, we could have worked with the augmented matrix $[A \mid \boldsymbol{b}]$ directly.

Partial pivoting is very similar to standard GE, except that the pivot in each column is the element with maximum modulus. In addition, we keep track of the row-permutations in order for the final matrix—which is U—to be upper triangular; the permutations vectors are shown below the matrices.

$$\begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 0 \\ \boxed{4} & 20 & 8 \end{bmatrix} \sim \begin{bmatrix} \underline{1/4} & -3 & -1 \\ \underline{3/4} & \boxed{-11} & -6 \\ 4 & 20 & 8 \end{bmatrix} \sim \begin{bmatrix} \underline{1/4} & \underline{3/11} & 7/11 \\ \underline{3/4} & -11 & -6 \\ 4 & 20 & 8 \end{bmatrix}$$

$(1, 2, 3)$ (no perm.) $\qquad$ $(3, 2, 1)$ $\qquad$ $(1, 2, 3)$ (no perm.)

Thus the LU factorization $PA = LU$ has

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 3/4 & 1 & 0 \\ 1/4 & 3/11 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 4 & 20 & 8 \\ 0 & -11 & -6 \\ 0 & 0 & 7/11 \end{bmatrix},$$

and we solve $A\boldsymbol{x} = \boldsymbol{b}$ by forward and backward subtitution as $L\boldsymbol{y} = P^{-1}\boldsymbol{b}$ and $U\boldsymbol{x} = \boldsymbol{y}$.

2 This task demonstrates how a badly conditioned problem can be improved by simple means—in this case, at least—and should be carried out in MATLAB.

The net domestic production of crude oil in Norway from 1986 to 2010 measured in standard cubic metres ($\mathrm{S\,m^3}$) is provided in Table 1.

Table 1: Norwegian oil production in 1986–2010. Source: Statistics Norway.

Year	Oil production $\left(10^6\ \mathrm{S\,m^3}\right)$
1986	48.771
1990	94.542
1994	146.282
1998	168.744
2002	173.649
2006	136.577
2010	104.354

a) Find the interpolation polynomial of degree 6 for the points in the table with help of the MATLAB-command `polyfit`. Notice that it complains about the polynomial being badly conditioned.

Solution. Let t denote the temporal variable. Then the polynomial is approximately

$$4.26 \times 10^{-5} t^6 - 0.511 t^5 + 2550 t^4 - 6.80 \times 10^6 t^3 + 1.02 \times 10^{10} t^2 - 8.15 \times 10^{12} t + 2.71 \times 10^{15}.$$

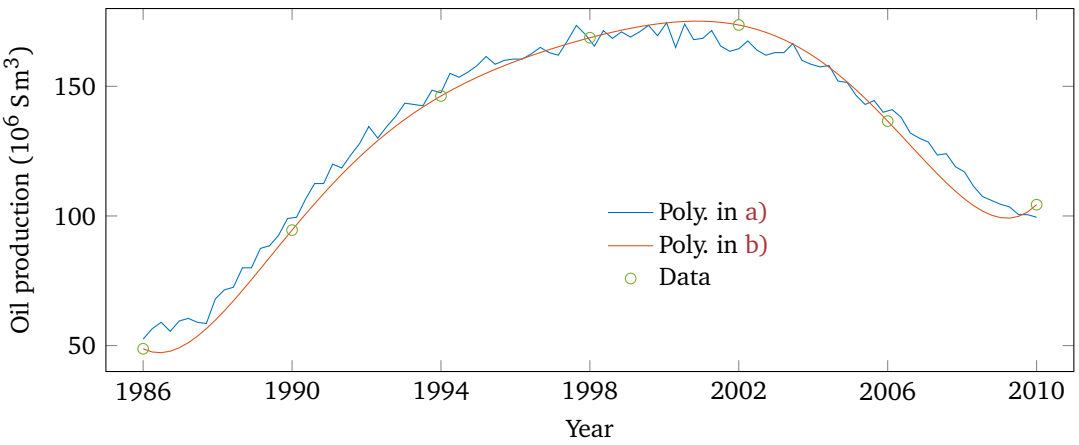

Figure 1: Oil data with fitted polynomials.

b) Next, repeat the exercise, but change the time axis by counting the years from 1986. Do you still get problems?

Solution. This time the fitted polynomial approximately becomes

$$6.46 \times 10^{-5} t^6 - 4.55 \times 10^{-3} t^5 + 0.123 t^4 - 1.58 t^3 + 9.33 t^2 - 7.27 t + 48.8,$$

and the previous problems are gone; see Figure 1 for a comparison.

c) In order to find the coefficients of the polynomial p_n interpolating a collection of points $\{(t_i, y_i)\}_{i=0}^n$, MATLAB solves the linear system

$$a_n t_i^n + a_{n-1} t_i^{n-1} + \cdots + a_1 t_i + a_0 = y_i, \qquad i = 0, \ldots, n$$

with respect to a_i's. Set up the corresponding coefficient-matrices for the cases in a) and b) and check the condition numbers of both. What do you observe?

Hint: Use the command `vander`.

Solution. In matrix-form the linear system equals

$$\begin{bmatrix} t_0^n & t_0^{n-1} & \ldots & t_0 & 1 \\ t_1^n & t_1^{n-1} & \ldots & t_1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ t_n^n & t_n^{n-1} & \ldots & t_n & 1 \end{bmatrix} \begin{bmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix},$$

where the coefficient-matrix is known as a Vandermonde matrix. Thus, with help of the commands `vander` and `cond`, the condition numbers—in the 2-norm—of the coefficient-matrices in a) and b) approx. equal 7.36×10^{34} and 7.06×10^{8}, respectively. Vandermonde matrices are notoriously ill-conditioned and our two cases show no exceptions. Case a), however, is much worse than b), which is in agreement with Figure 1.

3 Given the iteration scheme

$$\begin{aligned} 4x_{k+1} &= -x_k - y_k + z_k + 2, \\ 6y_{k+1} &= 2x_k + y_k - z_k - 1, \\ -4z_{k+1} &= -x_k + y_k - z_k + 4, \end{aligned}$$

prove that $\boldsymbol{x}^{(k)} = (x_k, y_k, z_k)$ converges to a limit $\boldsymbol{x}$ for all starting values $\boldsymbol{x}^{(0)}$ as $k \to \infty$. What is the limit $\boldsymbol{x}$?

Solution. It is helpful to first rewrite the iteration scheme on the form

$$P\boldsymbol{x}^{(k+1)} = N\boldsymbol{x}^{(k)} + \boldsymbol{b},$$

with

$$P = \operatorname{diag}(4, 6, -4), \qquad N = \begin{bmatrix} -1 & -1 & 1 \\ 2 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{b} = \begin{bmatrix} 2 \\ -1 \\ 4 \end{bmatrix}.$$

We can also write this as $\boldsymbol{x}^{(k+1)} = B\boldsymbol{x}^{(k)} + P^{-1}\boldsymbol{b}$, where $B = P^{-1}N$, and the scheme converges for any starting value if and only if the spectral radius $\rho(B) < 1$. Due to the fundamental relationship $\rho(B) < \|B\|$, it suffices to show that $\|B\| < 1$ in some consistent matrix norm $\|\cdot\|$. Since $\|B\|_\infty = 3/4$, for example, the scheme is convergent.

As regards the limit, we must have $P\boldsymbol{x} = N\boldsymbol{x} + \boldsymbol{b}$, or in other words, $\boldsymbol{x}$ solves the linear system $A\boldsymbol{x} = \boldsymbol{b}$, with $A = P - N$. Gaussian elimination yields $\boldsymbol{x} = \frac{1}{9}(1, 1, -12)$. Another solution strategy is simply to iterate until convergence.

4 Solve the following two systems of equations by Gauss–Seidel iterations:

a) $$\begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & -1 \\ 3 & 1 & -5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ -1 \end{bmatrix}.$$

b) $$\begin{bmatrix} 3 & 1 & 1 \\ 3 & 1 & -5 \\ 1 & 3 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ 3 \end{bmatrix}.$$

Use $(0.1, 0.1, 0.1)$ as the starting point. First do a few iterations by hand and then apply the attached MATLAB-program `gs.m`. Comment on the results. Do they comply with theory?

Solution. Recall that the Gauss–Seidel scheme for a system $A\boldsymbol{x} = \boldsymbol{b}$ takes the form

$$\boldsymbol{x}^{(k+1)} = B\boldsymbol{x}^{(k)} + L_*^{-1}\boldsymbol{b},$$

where $B = -L_*^{-1}U$ is the iteration matrix and L_* and U are the lower and strict upper triangular parts of A, respectively—so that $A = L_* + U$. We find that

a)
$$\boldsymbol{x}^{(1)} = \begin{bmatrix} 1.60 \\ 0.500 \\ 1.26 \end{bmatrix}, \qquad \boldsymbol{x}^{(2)} = \begin{bmatrix} 1.08 \\ 1.06 \\ 1.06 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{x}^{(3)} = \begin{bmatrix} 0.960\,00 \\ 1.0333 \\ 0.982\,67 \end{bmatrix}.$$

The iterations seem to converge, which is reasonable since A is strictly diagonally dominant. Notice also the same inference can be drawn from $\rho(B) < \|B\|_\infty = 2/3 < 1$.

b)
$$\boldsymbol{x}^{(1)} = \begin{bmatrix} 1.6 \\ -5.3 \\ -17.3 \end{bmatrix}, \qquad \boldsymbol{x}^{(2)} = \begin{bmatrix} 9.2 \\ -115.1 \\ -339.1 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{x}^{(3)} = \begin{bmatrix} 153.067 \\ -2155.700 \\ -6317.033 \end{bmatrix}.$$

This time the iterations diverge because $\rho(B) \geq \|B\|_\infty = 61/3 > 1$.

5 The system

$$\begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 6 \\ -2 \\ 6 \end{bmatrix}$$

is to be solved by an SOR method as follows.

Comment: the matrix can be built quickly in MATLAB with `toeplitz([4 -1 0 -1 0 0])` and setting two elements to 0.

a) Find an optimal ω and the corresponding $\rho(B_\omega)$. If you prefer, use the attached MATLAB-function `rhoSOR.m`, and plot $\rho(B_\omega)$ as a function of ω.

Solution. Remember first that the SOR method with parameter ω for the system $A\boldsymbol{x} = \boldsymbol{b}$ equals

$$\boldsymbol{x}^{(k+1)} = B_\omega \boldsymbol{x}^{(k)} + \boldsymbol{f}_\omega,$$

where $B_\omega = (D + \omega L)^{-1}\big[(1-\omega)D + \omega U\big]$ and $\boldsymbol{f}_\omega = \omega(D + \omega L)^{-1}\boldsymbol{b}$. Here we have decomposed A into its diagonal component D, and strict lower and upper triangular components L and U, respectively. Note that some authors use the negative of L and U; this just leads to a corresponding change of signs in B_ω and $\boldsymbol{f}_\omega$.

It is reasonable to expect that the speed of convergence increases as $\rho(B_\omega)$ gets smaller. Since A is symmetric and positive definite (why?), the SOR method converges if and only if $\omega \in (0, 2)$—see Property 4.3 in the book by Quarteroni *et al.* As such, ω_{opt} can be chosen as a minimizer of $\rho(B_\omega)$ with $\omega \in (0, 2)$, or symbolically,

$$\omega_{\text{opt}} \in \arg\min_{\omega \in (0,2)} \rho(B_\omega).$$

With help of `rhoSOR.m` we may now write a script as follows.

```
% Problem data.
A = toeplitz([4 -1 0 -1 0 0]); A(3, 4) = 0; A(4, 3) = 0;
b = [0, 5, 0, 6, -2, 6]';

% Compute and plot spectral radius for each value of omega.
omega = 0:0.001:2;
rho = zeros(1, length(omega));
for i = 1:length(omega);
    rho(i) = rhoSOR(A, omega(i));
end
plot(omega, rho);

% Find the spectral radius and the corresponding omega.
[rho_opt, index_opt] = min(rho);
omega_opt = omega(index_opt);
```

Figure 2 displays $\rho(B_\omega)$ as a function of ω, and we find that $\omega_{\text{opt}} \approx 1.113$, with $\rho(B_{\omega_{\text{opt}}}) \approx 0.113$.

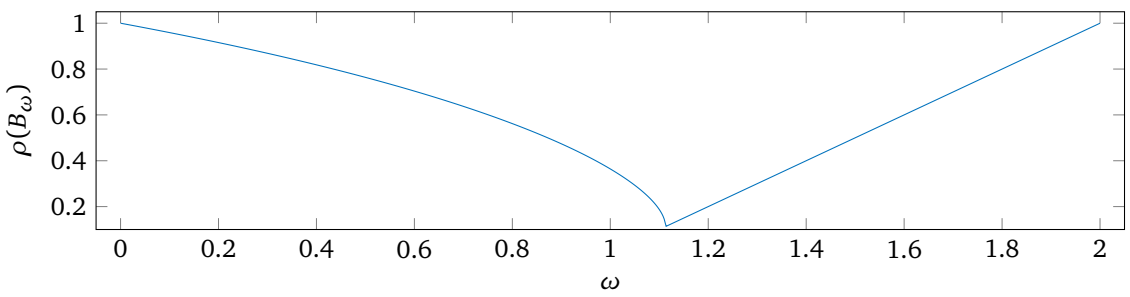

Figure 2: Spectral radius of B_ω as a function of ω.

b) Do 10 iterations with the optimal ω; you can for example put $\boldsymbol{x}^{(0)} = \boldsymbol{0}$. For each iteration, print the error $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}\|_2$.

Hint: rewrite the routine `gs.m` to do SOR iterations.

Solution. The difference between `gs.m` and the code for SOR is that we replace

```
x(i) = t / A(i, i);
```

with

```
x(i) = (1 - omega) * x(i) + omega * t / A(i, i);
```

and include ω as an input parameter. Using ω_{opt} from a) and $\boldsymbol{x}^{(0)} = \boldsymbol{0}$ yields the results in Table 2. Observe that the rate of convergence is approximately equal to $\rho(B_{\omega_{\text{opt}}}) \approx 0.113$.

Table 2: Errors in the SOR method using $\omega = \omega_{\text{opt}}$.

k	$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}\|_2$	$\frac{\|\boldsymbol{x}^{(k)} - \boldsymbol{x}\|_2}{\|\boldsymbol{x}^{(k-1)} - \boldsymbol{x}\|_2}$
1	1.5401	0.397 66
2	4.4071×10^{-1}	0.286 15
3	1.5087×10^{-1}	0.342 34
4	1.4298×10^{-2}	0.094 766
5	2.6222×10^{-3}	0.183 40
6	3.6698×10^{-4}	0.139 95
7	4.2805×10^{-5}	0.116 64
8	5.9554×10^{-6}	0.139 13
9	7.7858×10^{-7}	0.130 74
10	8.4566×10^{-8}	0.108 62

c) Repeat b) using other values of ω, *e.g.* 1.0 and 1.3. How does this affect the rate of convergence observed in b)? Is this as expected? Find a value of ω for which $\rho(B_\omega) = 1$, and perform iterations with values of ω around this value. How do the results comply with theory?

Solution. With $\omega = 1.0$ (reducing SOR to Gauss–Seidel) and $\omega = 1.3$, we obtain the list of errors in Table 3. Noticably, the errors are not as good as in b). Observed convergence rate for $\omega = 1.0$ is very close to $\rho(B_1) \approx 0.364$, while the rate for $\omega = 1.3$ is roughly of the same size as $\rho(B_{1.3}) = 0.3$. It is expected that the spectral radius is a measure for the speed of convergence.

Graphically from Figure 2, we see that $\rho(B_\omega) = 1$ when $\omega = 0$ or 2. As seen in Table 3, the SOR method fails to converge for these values of ω. This agrees with Property 4.3 in the book by Quarteroni *et al.*

Table 3: Errors in the SOR method for various values of ω.

k	$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}\|_2$			
	$\omega = 1$	$\omega = 1.3$	$\omega = 0.03$	$\omega = 1.95$
1	1.7567	1.2988	3.8111	2.7923
2	7.1870×10^{-1}	3.4412×10^{-1}	3.7511	2.6679
3	3.0347×10^{-1}	1.7701×10^{-1}	3.6928	2.7647
4	1.1309×10^{-1}	9.1103×10^{-2}	3.6363	2.2049
5	4.1352×10^{-2}	2.7128×10^{-2}	3.5813	2.1055
6	1.5073×10^{-2}	3.0010×10^{-3}	3.5278	2.3842
7	5.4914×10^{-3}	1.1024×10^{-3}	3.4758	2.6278
8	2.0004×10^{-3}	2.3147×10^{-4}	3.4251	2.0705
9	7.2872×10^{-4}	1.0237×10^{-4}	3.3757	1.9220
10	2.6546×10^{-4}	4.7021×10^{-5}	3.3275	2.0927