Lecturer: Barbara Re **TAs:** Davide Torlo, Stefano Colombo

Homework 4, Deadline: 23:00, Nov 11, 2019

Numerical Methods in Informatics (L + E)

Submission info:

- for **Exercise 1**: submit two files named `relaxmeth.m` and `relaxtest.m` as Exercise sheet 7,
- for **Exercise 2**: submit three files named `simp_comp.m`, `trap_comp.m`, and `test_quadr.m` as Exercise sheet 8.
- no submissions for **Exercise 3**.

Exercise 1 *Points 5*

For the solution of the system $A\mathbf{x} = \mathbf{b}$, with $A \in \mathbb{R}^{n\times n}$, consider the following *relaxation method*: given $\mathbf{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}]^{\mathrm{T}}$, for $k = 0, 1, \dots$ compute

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \quad i = 1, \dots, n$$

with $0 < \omega < 2$ a real parameter.

(a) Write a function called `relaxmeth.m` that implements this iterative method according to the following input and output specifications. Stop the iterations when the relative increment $(\|\boldsymbol{\delta}^{(k)}\|/\|\mathbf{b}\|)$ is less than a given tolerance ϵ.

INPUT:	`A` square matrix	*OUTPUT:*	`x` solution
	`b` vector of known values		`iter` number of iterations
	`x0` initial guess		
	`nmax` maximum number of iterations		
	`tol` value ϵ for the stopping criteria		
	`omega` relaxation parameter ω		

(b) Write a script called `relaxtest.m` which uses the relaxation method implemented before to solve the following linear system

$$\begin{aligned} 5x_1 - 12x_2 + 3x_3 &= -1 \\ -3x_1 + 9x_2 + 1x_3 &= 2 \\ 2x_1 - 1x_2 - 7x_3 &= 3 \end{aligned}$$

The script should

- build the iteration matrix B and compute the spectral radius for the set of values for $\omega \in [0.01 : 0.01 : 2]$.
- plot a graph of the spectral radius of the iteration matrix vs. the parameter ω.
- find the optimal relaxation parameter, ω^*, within the given set and add its value as a red circular mark to the previous plot. Save the plot as `relaxplot.fig`.
- compute the solution of the given linear system using the relaxation method with ω^*, for the initial gueass $\mathbf{x}^{(0)} = \mathbf{b}$ and $\epsilon = 0.0001$.
- **Additional Point (not to be submitted):** Approximately how much faster would the relaxation method with ω^* converge compared to Jacobi?

Hint: Look at the MATLAB commands `[M,I] = min(X)`, `[M,I] = max(X)`.

Exercise 2 *Points 5*

Approximate the integral of a given function $I(f) = \int_0^1 f(x)\mathrm{d}x$ using composite quadrature formulas: divide the integration interval $[0,1]$ into M equispaced subintervals of length H and in each subinterval I_k use

- for the Simpson formula: $I^c_{s\,k}(f) = \frac{H}{6}\left[f(x_{k-1}) + 4f(\bar{x}_k) + f(x_k)\right]$ with $\bar{x}_k = (x_k + x_{k-1})/2$;
- for the trapezoidal formula: $I^c_{t\,k}(f) = \frac{H}{2}\left[f(x_{k-1}) + f(x_k)\right]$.

(a) Implement the composite Simpson's rule in a function called `simp_comp` according to the following indications:

INPUT:	`f` function handle
	`n` number of sub-intervals
OUTPUT:	`I` approximated integral

(b) Implement the composite trapezoidal rule in a function called `trap_comp` according to the same indications as `simp_comp`.

(c) Test your quadrature functions with $f(x) = \pi x \sin(\pi x^2)$. Write the script `test_quadr.m` that

- evaluates analytically $I_{\mathrm{ex}}(f) = \int_0^1 \pi x \sin(\pi x^2)\mathrm{d}x$,
- computes the error for both composite quadrature formulas $e_n(f) = |I_{\mathrm{ex}}(f) - I^c_s f|$ and $e_n(f) = |I_{\mathrm{ex}}(f) - I^c_t f|$ for $n = 2^i$, $1 \leq i \leq 8$
- plots errors in an appropriate scale, along with expect theoretical convergence rates
- saves the plot as `plot_quadr.fig`.

Additional Exercise (not to be submitted)

Use a 5-point Gauss quadrature rule $I_{\mathrm{appr}} = \sum_{j=0}^{n} \alpha_j f(y_j)$ to compute

$$\int_{-3}^{3} \exp(x)\mathrm{d}x$$

(a) Compute analytically the expression of the Legendre polynomial $L_5(x)$

(b) Compute its roots in the interval $x \in [-1,1]$, which are the quadrature nodes $\{\bar{y}_j\}$
Hint: you can plot the polynomial to identify different initial guesses to be used in an appropriate iterative method.

(c) Compute the quadrature weights $\{\bar{\alpha}_j\}$, by integrating the Lagrangian polynomials $\phi_j(x)$ or using the expression for the *Gauss-Legendre formula* (see slides).

(d) Compute the quadrature (remind the interval transformation) and compare to the exact value.