EPFL

Numerical Analysis and Computational Mathematics

Fall Semester 2024 – CSE Section

Prof. Laura Grigori

Assistant: Israa Fakih

Session 8 – November 6, 2024

Numerical integration & Linear systems: direct methods

Exercise I (MATLAB)

Consider a function $f : [a, b] \to \mathbb{R}$, with $f \in C^0([a, b])$, whose integral is $I(f) = \int_a^b f(x)\,dx$. The approximation of $I(f)$ by means of a simple interpolatory formula reads:

$$I_{approx}(f) = \sum_{j=0}^{n} \alpha_j\, f(y_j),$$

where α_j are the quadrature weights and y_j the quadrature nodes, with $j = 0, \ldots, n$. The type of polynomial approximation of the function $f(x)$ in $[a, b]$ determines the specific quadrature formula. We observe that such formulas are typically defined on the reference interval $[\overline{a}, \overline{b}] = [-1, 1]$ where the quadrature nodes $\overline{y}_j$ and the weights $\overline{\alpha}_j$ are assigned. Then, the quadrature nodes and weights corresponding to the generic interval $[a, b]$ are obtained as:

$$y_j = \frac{a+b}{2} + \frac{b-a}{2}\overline{y}_j, \qquad \alpha_j = \frac{b-a}{2}\overline{\alpha}_j, \qquad \text{for } j = 0, \ldots, n.$$

The (simple) *Gauss-Legendre quadrature formulas* constitute a family of interpolatory formulas, each one specified by n, where $n + 1$ is the number of quadrature nodes and weights. The Gauss-Legendre quadrature formula for $n \geq 0$ has degree of exactness equal to $2n + 1$. The quadrature nodes and weights for some of the Gauss-Legendre quadrature formulas on the reference interval $[\overline{a}, \overline{b}] = [-1, 1]$ are:

n	$\{\overline{y}_j\}$	$\{\overline{\alpha}_j\}$
0	$\{0\}$	$\{2\}$
1	$\left\{-\frac{1}{\sqrt{3}}, +\frac{1}{\sqrt{3}}\right\}$	$\{1, 1\}$
2	$\left\{-\frac{\sqrt{15}}{5}, 0, +\frac{\sqrt{15}}{5}\right\}$	$\left\{\frac{5}{9}, \frac{8}{9}, \frac{5}{9}\right\}$

a) Write the MATLAB function `gauss_legendre_simple_quadrature.m` that implements the approximation of $I(f)$ by means of the simple Gauss-Legendre quadrature formula for $n = 0, 1, 2$. Use the template `gauss_legendre_simple_quadrature_template.m`.

```
function [ Ih ] = gauss_legendre_simple_quadrature( fun, a, b, n )
% GAUSS_LEGENDRE_SIMPLE_QUADRATURE approximate the integral of a function in
% the interval [a,b] by means of the simple Gauss-Legendre quadrature formula
%  [ Ih ] = gauss_legendre_simple_quadrature( fun, a, b, n )
%  Inputs: fun = function handle,
%          a,b = extrema of the interval [a,b]
%          n + 1 = number of quadrature nodes and weights
%  Output: Ih = approximate value of the integral
%
...
return
```

b) Consider the function $f(x) = \sin(\frac{7}{2}x) + e^x - 1$ with $a = 0$ and $b = 1$, for which $I(f) = 2/7\,(1-\cos(7/2)) + e - 2$. Use the MATLAB function implemented at point a) to approximate the integral $I(f)$ by means of the simple Gauss-Legendre formulas, for $n = 0, 1, 2$.

c) We set $f(x) = x^d$, $a = 0$, and $b = 1$, with $d \in \mathbb{N}$, so that $I(f) = 1/(d+1)$. By using the MATLAB function implemented at point a), verify the degrees of exactness of the Gauss-Legendre formulas for $n = 0, 1, 2$, by approximating the integral $I(f)$ for different values of d. Then, compare the results with the approximated values of the integrals obtained by means of the simple midpoint, trapezoidal, and Simpson quadrature formulas, using the MATLAB functions from Series 7.

Exercise II (Theoretical)

Prove that, on the reference interval $I = [-1, 1]$, the Gauss-Legendre formula for $n = 1$ has degree of exactness $r = 3$. i.e. $I_{GL,1}(f) = I(f)$ for all $f \in \mathbb{P}_3$.

Exercise III (MATLAB)

Consider the linear system $A\mathbf{x} = \mathbf{b}$ with $A \in \mathbb{R}^{n\times n}$, $\mathbf{x}$, $\mathbf{b} \in \mathbb{R}^n$, with $n \geq 1$. We are interested in computing the solution vector $\mathbf{x}$ by means of the LU factorization method.

a) Write the MATLAB functions {`forward,backward`}`_substitutions.m`, which implement the forward and backward substitutions algorithms, respectively. The forward substitution algorithm should solve the linear system $L\mathbf{y} = \mathbf{b}$, with $L \in \mathbb{R}^{n\times n}$ a lower triangular matrix and $\mathbf{y} \in \mathbb{R}^n$. The backward substitution algorithm should solve the linear system $U\mathbf{x} = \mathbf{y}$, with $U \in \mathbb{R}^{n\times n}$ an upper triangular matrix. Use the templates {`forward,backward`}`_substitutions_template.m`:

```
function [ y ] = forward_substitutions( L, b )
% FORWARD_SUBSTITUTIONS solve the linear system L y = b by means of the
% forward subsitutions algorithm; L must be a lower triangular matrix
%  [ y ] = forward_substitutions( L, b )
%  Inputs: L = lower triangular matrix (square matrix)
%          b = vector (right hand side of the linear system)
%  Output: y = solution vector (column vector)
%
...
return
```

```
function [ x ] = backward_substitutions( U, y )
% BACKWARD_SUBSTITUTIONS solve the linear system U x = y by means of the
% backward subsitutions algorithm; U must be an upper triangular matrix
%  [ x ] = backward_substitutions( U, y )
%  Inputs: U = upper triangular matrix (square matrix)
%          y = vector (right hand side of the linear system)
%  Output: x = solution vector (column vector)
%
...
return
```

b) For $n = 3$, define the linear system $A\mathbf{x} = \mathbf{b}$ by setting

$$A = \begin{bmatrix} 4 & -2 & -1 \\ -1 & 3 & -1 \\ -1 & -3 & 5 \end{bmatrix},$$

and by assigning a priori the exact solution

$$\mathbf{x}_{ex} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} : \qquad \mathbf{b} = A\mathbf{x}_{ex}.$$

Solve the linear system by means of the LU factorization method using the MATLAB functions implemented at point a). In order to obtain the LU factorization of the matrix A, use the MATLAB function `lu` with the syntax `[L,U,P]=lu(A)` (see `help lu`). Note that the MATLAB function `lu` returns by default the LU factorization with pivoting even when not strictly required. Compare the result $\mathbf{x}$ and the exact solution $\mathbf{x}_{ex}$ by computing the error norm $\|\mathbf{x} - \mathbf{x}_{ex}\|_2$. Verify whether the pivoting technique has been applied by displaying the permutation matrix P.

c) Consider the following matrix $A \in \mathbb{R}^{n \times n}$ and vectors $\mathbf{x}_{ex}$, $\mathbf{b} \in \mathbb{R}^n$:

$$A = \begin{bmatrix} 4 & -1 & 0 & & & & \cdots & 0 & 1 \\ -2 & 4 & -1 & 0 & & & \cdots & 0 & 0 \\ -1 & -2 & 4 & -1 & 0 & & \cdots & & 0 \\ 0 & -1 & -2 & 4 & -1 & 0 & \cdots & & 0 \\ \vdots & & & \ddots & \ddots & \ddots & \ddots & & \vdots \\ 0 & & \cdots & & 0 & -1 & -2 & 4 & -1 & 0 \\ 0 & 0 & \cdots & & & 0 & -1 & -2 & 4 & -1 \\ -1 & 0 & \cdots & & & & 0 & -1 & -2 & 4 \end{bmatrix}, \quad \mathbf{x}_{ex} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{b} = A\mathbf{x}_{ex}.$$

Set $n = 20$ and use the MATLAB function `diag` to define the matrix A. Then, repeat point b). In particular, visualize matrices A, L, and U by means of the MATLAB function `spy`.

d) Consider the matrix A introduced at point c), with $n = 1000$. Define the matrix A in MATLAB in the *full* format (as done at point c)) and in *sparse* format. In the latter case you can use the MATLAB function `sparse`. Compare the memory required to store in MATLAB the matrix A in the full and sparse formats by using the command `whos`.

e) With the notation of point b), consider the linear system $A\mathbf{x} = \mathbf{b}$, with:

$$A = \begin{bmatrix} 4 & -2 & -1 \\ -2 & 7 & -4 \\ -1 & -4 & 6 \end{bmatrix}, \qquad \mathbf{x}_{ex} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \qquad \mathbf{b} = A\mathbf{x}_{ex}.$$

The matrix A is symmetric and positive definite. Solve the linear system by means of the *Cholesky* factorization method using the MATLAB functions implemented at point a). In order to obtain the *Cholesky* factorization of the matrix A, use the MATLAB function `chol`. Compare the result $\mathbf{x}$ and the exact solution $\mathbf{x}_{ex}$ by computing the error norm $\|\mathbf{x} - \mathbf{x}_{ex}\|_2$.

Exercise IV (Theoretical)

Consider the following matrix $A \in \mathbb{R}^{3\times 3}$, which depends on the parameter $\alpha \in \mathbb{R}$:

$$A = \begin{bmatrix} 1 & \alpha & -1 \\ \alpha & \frac{35}{3} & 1 \\ -1 & \alpha & 2 \end{bmatrix}.$$

a) Calculate the values of the parameter $\alpha \in \mathbb{R}$ for which the matrix A is invertible.

b) Without using pivoting, calculate the Gauss factorization of the matrix A (when non-singular) for a generic value of the parameter $\alpha \in \mathbb{R}$. In particular, identify the values of the parameter $\alpha \in \mathbb{R}$ for which the Gauss factorization (without pivoting) of the matrix A (when non-singular) exists and is unique.

c) Use the pivoting technique to calculate the Gauss factorization LU of the matrix A in the case $\alpha = \sqrt{\frac{35}{3}}$.

d) For $\alpha = 1$, the matrix A is symmetric and positive definite. Calculate the corresponding Cholesky factor of the matrix A, i.e. the upper triangular matrix R (with positive elements on the diagonal), such that $A = R^T R$.