Elemental numerical methods, 2h00

Exercise 1 (Cholesky decomposition)

Let L be a lower triangular $n \times n$ matrix of the form

$$L = \begin{bmatrix} d_1 & & & 0 \\ b_1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & b_{n-1} & d_n \end{bmatrix}.$$

1. Compute the product LL^T when $n = 4$.
2. Let A be a tridiagonal $n \times n$ matrix, which is symmetric positive definite

$$A = \begin{bmatrix} \delta_1 & \beta_1 & & 0 \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ 0 & & \beta_{n-1} & \delta_n \end{bmatrix}.$$

Write an algorithm in order to compute L such that

$$A = LL^T.$$

Exercise 2 (Newton's method)

We consider a smooth function $f : \mathbb{R} \to \mathbb{R}$ with only one point r in $\mathbb{R}$ such that $f(r) = 0$. The Newton's method for finding r consists in choosing an initial guess $x^{(0)}$ and then computing the sequence $x^{(p)} \to r$ defined by

$$x^{(p+1)} = x^{(p)} - \frac{f(x^{(p)})}{f'(x^{(p)})}, \quad p = 1, 2, 3, \cdots$$

The function f and its derivative f' are given in two fortran functions `f(x)` and `df(x)`. Write a fortran subroutine `newton(x0,r,eps)` that computes r with a precision of $\varepsilon =$`eps`, starting from $x^{(0)} =$`x0`.

Exercise 3 (Second order implicit scheme)

We consider the following scheme for solving the differential equation $x'(t) = g(x(t))$, $x_n \simeq x(n\Delta t)$,

$$x_n = x_{n-1} + \frac{\Delta t}{2} \left(g(x_{n-1}) + g(x_n) \right), \quad n = 1, 2, 3, \cdots, \quad x_0 = x(0).$$

1. Prove that this scheme is of order 2.
2. What are the advantages and drawbacks of this scheme ?
3. With the Newton's method of Exercise 2, construct **for each fixed** n a sequence $(x_n^{(p)})_{p \in \mathbb{N}}$, depending on p, such that $x_n^{(p)} \underset{p \to \infty}{\to} x_n$. Describe the algorithm. For a given n, how do you choose the initial guess $x_n^{(0)}$ in the Newton's method ?
4. The function g and g' are given in two fortran functions `g(x)` and `dg(x)`. Write a fortran program for computing the numerical values $(x_n)_{n \in \mathbb{N}}$ with a precision $\varepsilon =$`eps`. Your program has to use the subroutine `newton(x0,r,eps)` of Exercise 2. Write the fortran functions `f(x)` and `df(x)`.

Exercise 4

We consider the square $\Omega = [a, b] \times [c, d]$. Write a simple 9-point numerical integration rule, based on the Simpson's method, for computing

$$I = \int_\Omega f = \int_{x=a}^{b} \int_{y=c}^{d} f(x, y) dx dy.$$