

## Practical Work #2

The aim of this session is to apply numerical methods to find approximate solutions to problems  $f(\alpha) = 0$  where  $f$  is a function and  $\alpha$  a real number and  $Ax = b$  where  $A$  is a matrix,  $x$  and  $b$  are vectors. The numerical methods we consider are sequences that converges towards the solution of those problems.

### 1 Root of a function

Let  $f$  be the function

$$f(x) = x^2 e^{-x} - 1.$$

There exists a unique  $\alpha \in \mathbb{R}$  such that  $f(\alpha) = 0$ . Moreover, it is straightforward that  $\alpha \in (-1, 0)$ . We assess in this section several methods to provide a “good” approximation of  $\alpha$ .

#### 1.1 Bisection

The bisection method relies on the intermediate value theorem which states that as  $f$  is continuous, if  $f(x) < 0$  and  $f(y) > 0$ , then  $\alpha$  lies between  $x$  and  $y$ . The sequence  $a_n$  which is expected to converge towards  $\alpha$  is thus constructed like this:

---

#### Algorithm 1 Bisection method

---

Data:  $a, b$  such that  $f(a) < 0$  and  $f(b) > 0$

$a_1 \leftarrow a, b_1 \leftarrow b, c \leftarrow \frac{a+b}{2}$

$n \leftarrow 1$

**while**  $f(c) \neq 0$  **do**

**if**  $f(a_n)f(c) < 0$  **then**

$a_{n+1} \leftarrow a_n$

$b_{n+1} \leftarrow c$

**else**

$a_{n+1} \leftarrow c$

$b_{n+1} \leftarrow b_n$

**end if**

$c \leftarrow \frac{a_{n+1} + b_{n+1}}{2}$

$n \leftarrow n + 1$

**end while**

---

The key point is the stopping criterion. We may never obtain  $f(a_n) = 0$  due to round off errors. Here are some possibilities depending on a user-tuned threshold  $\epsilon > 0$ :

- Relative error: **while**  $|a_{n+1} - a_n| > \epsilon$
- Absolute error: **while**  $|a_n - \alpha| > \epsilon$  (requires the exact solution  $\alpha \dots$ )
- Residual: **while**  $|f(a_n)| > \epsilon$

In any case, there must be a cautious criterion including a maximal number of iterations for the case where the method may not converge and thus to prevent infinite loops.

### 1.2 Newton's method

This method reads

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)}.$$

For the method to be well-posed,  $f'$  must not vanish in the neighbourhood of  $\alpha$ . As a consequence, the first term  $a_1$  which is chosen by the user must be close enough to the very solution of the equation.

### 1.3 Secant method

The Newton's method can be modified in order to avoid evaluating the derivative function. It results in the secant method

$$a_{n+1} = a_n - f(a_n) \frac{a_n - a_{n-1}}{f(a_n) - f(a_{n-1})}.$$

The restriction is the same as the original method, *i.e.* that  $a_1$  and  $a_2$  must be different and close to  $\alpha$ .

### 1.4 Directions

1. Implement the resolution of  $f(x) = 0$  by means of the three methods.
2. Set the accuracy threshold  $\varepsilon$  as well as the maximal number of iterations. Take relevant values for  $a_1$ ,  $b_1$  (and potentially  $a_2$ ). Compare the number of iterations for each method to reach the required accuracy.
3. Compute for each method the increment  $e_n = |a_{n+1} - a_n|$ . The method is said to be of order  $p$  if

$$e_{n+1} \leq C e_n^p \quad \text{or equivalently} \quad \ln(e_{n+1}) \leq \ln(C) + p \ln(e_n)$$

for some constant  $C > 0$ . Determine the order of each method by plotting  $\ln(e_{n+1})$  as a function of  $\ln(e_n)$ .

## 2 Iterative methods for linear systems

Let  $A \in \mathcal{M}_n(\mathbb{R})$  be an invertible matrix. We introduce the following notations

$$D = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ -a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n,1} & \cdots & -a_{n,n-1} & 0 \end{pmatrix},$$

such that  $A = D - L - U$ .

### 2.1 Settings

To solve the linear system  $Ax = b$  for a given  $b \in \mathbb{R}^n$ , we investigate two iterative strategies of the form

$$x^{k+1} = M^{-1}(Nx^k + b)$$

with

- for the *Jacobi method*:  $M = D$ ,  $N = L + U$ ;
- for the *SOR method*:  $M = \frac{1}{\omega}D - L$ ,  $N = \frac{1-\omega}{\omega}D + U$ .

To initialize the method we have to choose a first guess  $x^1$ .

## 2.2 Directions

We take

$$A = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}.$$

Let us set for instance  $n = 5$ ,  $\bar{x} = (1, 1, 1, 1, 1)^T$  and  $x^0 = (0, 0, 0, 0, 0)^T$ .

1. Compute  $b = A\bar{x}$ .
2. Implement the two previous iterative methods with some relevant stopping criteria (see 1.1).
3. Is the SOR method convergent for any value of  $\omega$ ?
4. In the case of convergence, plot the errors  $e_j^k = \|x_j^k - \bar{x}\|$  and  $e_{SOR}^k = \|x_{SOR}^k - \bar{x}\|$  as a function of  $k$ . We consider here the  $L^2$  norm :  $\|x\| = \sqrt{\sum_i x_i^2}$ .