Approximations of Differential Systems

EXERCISE 1. *Newton and fixed point method for scalar nonlinear problem.*

We want to solve scalar nonlinear problem. For example, we want to find root(s) of nonlinear equations such that

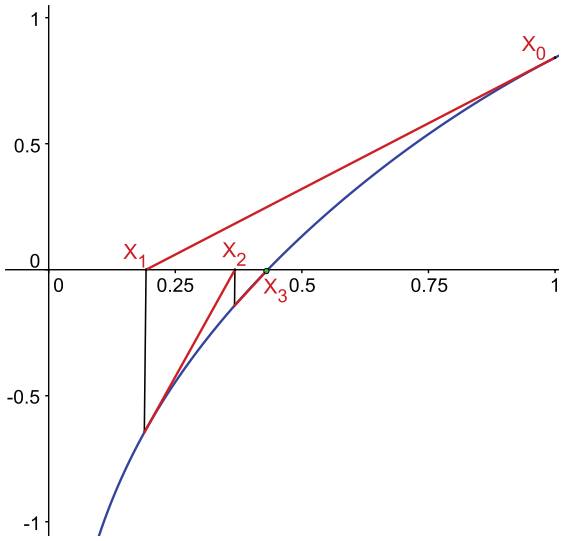
$$f(x) = \tanh(x)\cos(x^2) + x + 2 = 0.$$

The two following methods are well known

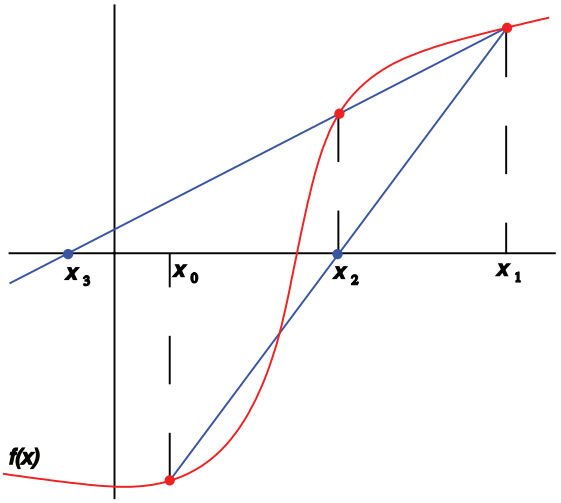
Newton method $y_{k+1} = y_k - \dfrac{f(y_k)}{f'(y_k)}$

Secant method $y_{k+1} = y_k - f(y_k)\dfrac{y_k - y_{k-1}}{f(y_k) - f(y_{k-1})}$

which can be illustrated by the two figures Implement these two methods in Matlab and Fortran 90 (or



(a) Newton method



(b) Secant method

in C Language).

EXERCISE 2. *Forward Euler, Backward Euler and Crank Nicolson methods in Matlab*

In this exercise, we want to build functions to approximate the ordinary differential equations

$$\begin{cases} y'(t) = f(y(t),t), & t \in (a,b], \\ y(a) = y_a, \end{cases}$$

where $f : \mathbb{R}^2 \to \mathbb{R}$.
For this purpose, implement three Matlab functions, respectively for Forward Euler, Backward Euler and Crank-Nicolson methods, following the prototype described below.

```
function [t,u]= feuler(odefun ,tspan ,y0 ,Nh ,varargin )
% FEULER Solves differential equations using the forward
% Euler method.
% [T,Y]= FEULER(ODEFUN ,TSPAN ,Y0 ,NH) with TSPAN=[T0 ,TF]
```

```
% integrates the system of differential equations
% y'=f(t,y) from time T0 to TF with initial condition
% Y0 using the forward Euler method on an equispaced
% grid of NH intervals .
% Function ODEFUN(T,Y) must return a vector , whose
% elements hold the evaluation of f(t,y), of the
% same dimension of Y.
% Each row in the solution array Y corresponds to a
% time returned in the column vector T.
% [T,Y] = FEULER(ODEFUN ,TSPAN ,Y0 ,NH ,P1 ,P2 ,...) passes
% the additional parameters P1 ,P2 ,... to the function
% ODEFUN as ODEFUN(T,Y,P1 ,P2 ...).
```

In order to use your functions, you can follow the next procedure which aims to solve

$$\begin{cases} y'(t) = cos(2y(t)), & t \in (0,1], \\ y(0) = 0. \end{cases} \tag{1}$$

```
tspan=[0 ,1]; y0 =0; f=inline('cos (2*y)','t','y');
Nh =2;
[t,ufe ]= feuler(f,tspan ,y0 ,Nh);
```

For Backward Euler or Crank Nicolson method, you could have to build a Newton function to solve a nonlinear problem.

You are invited to test your functions on the ODE given by Eq. (1). The exact solution is

$$y(t) = \frac{1}{2} \arcsin \frac{e^{4t}-1}{e^{4t}+1}.$$

We want to evaluate the (numerical) order of the different methods. To do so, we can take advantage of the exact solution. Indeed, let us denote by $(y^n_{\Delta t})_{n\in[0,N]}$ the sequence generated by a numerical scheme with Δt as time step and N time steps. We know that if the scheme is or order p, we have

$$|y^N_{\Delta t} - y_{ex}(N\Delta t)| = C\Delta t^p.$$

1. Use this relation and the same for time step $\Delta t/2$ to give an expression of p.
2. Generate different values of p for the different numerical schemes
3. Learn to use the Matlab functions `ode23` and `ode45`. Perform the same experiments.

EXERCISE 3. *Forward Euler, Backward Euler and Crank Nicolson methods in Fortran*

Make the same exercise by implementing your functions in Fortran 90 (or in C). You also have to build equivalent functions of `ode23` and `ode45`. Use text files to exchange your data with Matlab in order to plot them.

EXERCISE 4. *Newton and quasi Newton methods in higher dimensions*

We want to solve the nonlinear equation $F(X)=0$ where $F:\mathbb{R}^d\to\mathbb{R}^d$ and $X\in\mathbb{R}^d$. The Newton method takes the following form

$$Y_{k+1}=Y_k-[\nabla F(Y_k)]^{-1}F(Y_k)$$

where $\nabla F(Y_k)$ denotes the Jacobian matrix

$$\nabla F(Y_k)=\begin{pmatrix}\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_d}\\ \frac{\partial f_2}{\partial x_1} & \ddots & & \vdots\\ \vdots & & \ddots & \vdots\\ \frac{\partial f_d}{\partial x_1} & \cdots & \cdots & \frac{\partial f_d}{\partial x_d}\end{pmatrix}$$

The quasi-Newton methods are based on the secant approximation. The idea is to replace the evaluation of $\nabla F(Y_k)$ by a matrix A_k which has to satisfy

$$A_k(Y_k-Y_{k-1})=F(Y_k)-F(Y_{k-1}).$$

There exists infinity possible matrices A_k.

The **Broyden** method assumes $A_kZ=A_{k-1}Z$ if $(Y_k-Y_{k-1})'Z=0$. The algorithm becomes
- initialization Y_0 and A_0 (finite differences)
- iteration

$$\begin{aligned}Y_{k+1}&=Y_k-A_k^{-1}F(Y_k)\\ U_{k+1}&=F(Y_{k+1})-F(Y_k)\\ S_{k+1}&=Y_{k+1}-Y_k\\ A_{k+1}&=A_k+\frac{(U_{k+1}-A_kS_{k+1})(S_{k+1})'}{\|S_{k+1}\|^2}.\end{aligned}$$

1. Implement the two methods
2. Test your methods on the example

$$\begin{cases}3x-\cos(yz)-\frac{1}{2}=0,\\ x^2-81(y+0.1)^2+\sin(z)+1.06=0\\ e^{-xy}+20z+\frac{10\pi-3}{3}=0\end{cases}$$

If one uses the starting point $(x_0,y_0,z_0)=(0.1,0.1,-0.1)$, the schemes should converge to $(0.5,0,-\pi/6)$.

EXERCISE 5. *Numerical methods for ODE systems*

With the help of solutions of Exercise 4, reproduce exercises 2 and 3 for ODE systems.

EXERCISE 6. *Study of the spherical pendulum*

The motion of a point $x(t)=(x_1(t),x_2(t),x_3(t))^T$ with mass m subject to the gravity force $F=(0,0,-gm)^T$ (with $g=9.8m/s^2$) and constrained to move on the spherical surface of equation $\phi(x)=x_1^2+x_2^2+x_3^2-1=0$ is described by the following system of ordinary differential equations

$$x''=\frac{1}{m}\big(F-\frac{mx'^THx'+\nabla\phi^TF}{|\nabla\phi|^2}\nabla\phi\big)\text{ for }t>0. \tag{2}$$

We denote by x' the first derivative with respect to t, with x'' the second derivative, with $\nabla\phi$ the spatial gradient of ϕ, equal to $2x$, with H the Hessian matrix of ϕ whose components are $H_{ij}=\partial^2\phi/\partial x_i\partial x_j$

for $i, j = 1,2,3$. In our case H is a diagonal matrix with coefficients all equal to 2. System (2) must be provided with the initial conditions $x(0) = x_0$ and $x'(0) = v_0$.

To numerically solve (2) let us transform it into a system of differential equations of order 1 in the new variable y, a vector with 6 components. Having set $y_i = x_i$ and $y_{i+3} = x_i'$ with $i = 1,2,3$, and

$$\lambda = \frac{m(y_4, y_5, y_6)^T H(y_4, y_5, y_6) + \nabla\phi^T F}{|\nabla\phi|^2},$$

we obtain, for $i = 1,2,3$,

$$\begin{aligned} y_i' &= y_{3+i}, \\ y_{3+i}' &= \tfrac{1}{m}\big(F_i - \lambda \tfrac{\partial\phi}{\partial y_i}\big). \end{aligned}$$

Study these equations for $t \in [0,25]$ and $y(t=0) = (0,1,0,.8,0,1.2)^T$ with different discretization scheme.

EXERCISE 7. *Stiff problems*

Study the equation

$$\begin{cases} y'(t) = \lambda(y(t) - g(t)) + g'(t), & t > 0, \\ y(0) = y_0 \end{cases}$$

where g is a regular function and $\lambda \ll 0$, whose solution is

$$y(t) = (y0 - g(0))e^{\lambda t} + g(t), t \geq 0.$$

With $g(t) = t$, $\lambda = -100$, solve the problem over the interval $(0,100)$ with differente discretization scheme.