CORRECTIONS

CH-250, part I for 2015/16

Exercise 1. (15 points)
We have the system

$$A = \begin{pmatrix} 1 & -5 \\ 7 & -1 \end{pmatrix} \qquad b = \begin{pmatrix} -4 \\ 6 \end{pmatrix}$$

We apply the Jacobi method

$$x_1^{(k)} = -\frac{a_{12}}{a_{11}} x_2^{(k-1)} + \frac{b_1}{a_{11}} = 5x_2^{(k-1)} - 4$$

$$x_2^{(k)} = -\frac{a_{21}}{a_{22}} x_1^{(k-1)} + \frac{b_2}{a_{22}} = 7x_1^{(k-1)} - 6$$

We take the given starting point (x1, x2)=(0, 0) and perform 3 iterations:

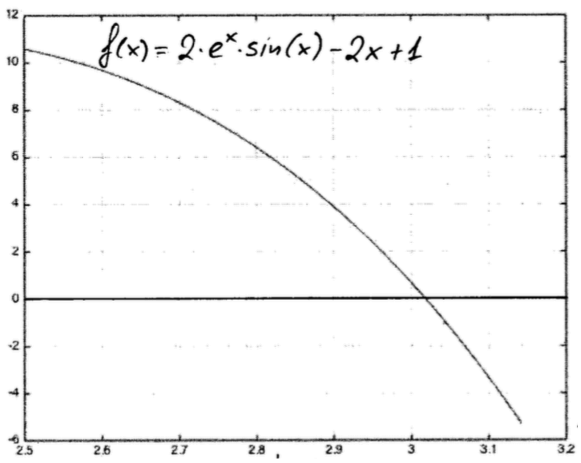
$$x_1^{(1)} = 5 \cdot 0 - 4 = -4$$
$$x_2^{(1)} = 7 \cdot 0 - 6 = -6$$

$$x_1^{(2)} = 5 \cdot (-6) - 4 = -34$$
$$x_2^{(2)} = 7 \cdot (-4) - 6 = -34$$

$$x_1^{(3)} = 5 \cdot (-34) - 4 = -174$$
$$x_2^{(3)} = 7 \cdot (-34) - 6 = -244$$

The solution does not converge. If the matrix A was diagonally dominant, the method would converge. It is not the case here.

Exercise 2. (20 points)



a) Regula Falsi (10 points)

1st Iteration ($a_1 = 2.5$, $b_1 = \pi$):

$$c_1 = b_1 - \frac{f(b_1)(b_1 - a_1)}{f(b_1) - f(a_1)} = 2.92793597$$
$$f(a_1) \cdot f(c_1) = 32.48 > 0, \textit{ so } a_2 = c_1$$

<u>2nd Iteration (a_2 = 2.92793597, b_2 = π):</u>

$$c_2 = b_2 - \frac{f(b_2)(b_2 - a_2)}{f(b_2) - f(a_2)} = 3.00645312$$
$$f(a_2) \cdot f(c_2) = 1.331 > 0, \textit{ so } a_3 = c_2$$

<u>3rd Iteration (a_3 = 3.00645312, b_3 = π):</u>

$$c_3 = b_3 - \frac{f(b_3)(b_3 - a_3)}{f(b_3) - f(a_3)} = 3.01671883$$

<u>Accuracy of solution:</u>

$$\left| x^* - c_3 \right| = 0.00144538$$

b) Newton Raphson (10 points)

<u>Derivative:</u>

$$f'(x) = 2 \cdot e^x (sin(x) + cos(x)) - 2$$

<u>Starting point:</u>

$$x_0 = 2.5$$

<u>1st Iteration:</u>

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 4.02517042$$

<u>2nd Iteration:</u>

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 3.43863015$$

<u>Accuracy of solution:</u>

$$\left| x^* - x_2 \right| = 0.42046594$$

Exercise 3. (25 points)

a) Adams-Bashforth-Moulton (15 points)

1) Since the Adams-Bashforth-Moulton method is not a self-starting method, we need to calculate 3 initial points:

$$\frac{dx}{dt} = -2x(t)$$
$$h = \frac{1}{3}$$

$$x(t_0) = 3$$
$$x(t_1) = x(t_0) + h * f(x(t_0), t_0) = 3 - \frac{1}{3} * 6 = 1$$
$$x(t_2) = x(t_1) + h * f(x(t_1), t_1) = 1 - \frac{1}{3} * 2 = \frac{1}{3}$$

$$x(t_3) = x(t_2) + h * f(x(t_2), t_2) = \frac{1}{3} - \frac{1}{3} * \frac{2}{3} = \frac{1}{9}$$

2) Application of the method

$$h = \frac{1}{3}$$

$$x^p(t_4) = \frac{1}{9} + \frac{1}{72} * \left(55 * \left(-\frac{2}{9}\right) + 59 * \left(\frac{2}{3}\right) - 37 * 2 + 9 * 6\right) = 0.2$$

$$x(t_4) = \frac{1}{9} + \frac{1}{72} * \left(9 * (-0.4) + 19 * \left(-\frac{2}{9}\right) - 5 * \left(-\frac{2}{3}\right) - 2\right) = 0.02$$

$$x^p(t_5) = 0.02 + \frac{1}{72} * \left(55 * (-0.04) + 59 * \left(\frac{2}{9}\right) - 37 * \frac{2}{3} + 18\right) = 0.07$$

$$x(t_5) = 0.02 + \frac{1}{72} * \left(9 * (-0.16) - 19 * 0.04 + 5 * \frac{2}{9} - \frac{2}{3}\right) = 0$$

b) (5 points)

The Adam-Bashforth-Moulton method can be used to solve stiff systems. This is because the Adams-Moulton corrector is an implicit method.

c) (5 points)

The Euler backward method is an implicit method. This is because in order to estimate the point $x(t_{k+1})$ the algorithm require the value of $f(x(t_{k+1}),t_{k+1})$.

Exercise 4. (15 points)

a) Central differences (6 points)

At t=1s, we can use h=2s, therefore we get

$$a(t_0) = \frac{v\left(t_0 + \frac{h}{2}\right) - v\left(t_0 + \frac{h}{2}\right)}{h} \text{ ----> } a(1) = \frac{v(2) - v(0)}{2} = 12.4\, m/s^2$$

At t=4s, we have to take h=4s, therefore we get

$$a(4) = \frac{v(6) - v(2)}{4} = 5.7\, m/s^2$$

The accuracy is better at t=1s as we have used the smaller differentiation step.

b) Integration (9 points)

Since the data are given in 5 equidistant points (i.e. 4 intervals) we can use Simpson 1/3. Therefore, we have

$$I = \frac{h}{3}[f(t_0) + 4f(t_1) + 2f(t_2) + 4f(t_3) + f(t_4)] = 280.333\, m$$

Exercise 5. (25 points)

a) (3 points)

The missing code at * is:

```
errF=norm(A*x-b);
```

b) (5 points)

The mistake in the code: for negative values of *Vin* elements the size of the vector *Vin* is reduced in the loop causing the problems with accessing all the elements of *Vin*. One of the possible solutions is to store the positive elements in a new vector. Another is to access the elements of *Vin* in the decreasing order of indices.

c) (8 points)

The missing code:

```
while (diff_x>convThresh)&&(iterationN<max_it)
      x_new=x_old-fun(x_old)./dfun(x_old);
      ...
      iterationN=iteration+1;
end
```

d) (3 points)

```
N=10;
r=0.5;
V=r.^(0:N); %another option would be V=power(r,0:N);
```

e) (3 points)

The function computes the derivative of a function *fH* at x_0 using the central differences method;

f) (3 points)

```
x0=2;
h=1e-6;
fH=@(x) (sin(x)*exp(2*x));
[F,dF]=myFunction(fH,x0,h);
```