Universität Duisburg-Essen
Computational Mechanics
Campus Essen

Dr. S. Vanis

Fall Term 2010/11

13. January 2011

Introduction to Numerical Methods
Tutorial 12

Exercise 1:
The task to find a fixed point $x^* \in \mathbb{R}^2$ with $g(x^*) = x^*$ for

$$g : \mathbb{R}^2 \to \mathbb{R}^2 \quad , \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \begin{pmatrix} \frac{1}{2}x_1 + \frac{1}{4}x_2 + 1 \\ \frac{2}{3}x_1 + \frac{1}{6}x_2 - 3 \end{pmatrix}$$

should be considered step-by-step in this exercise.

(i) Show that g is contractive on $\mathbb{R}^2$ with respect to either the "column-sum-norm", i.e., $\|A\|_1 := \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}|$, or the "row-sum-norm", i.e., $\|A\|_\infty := \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ji}|$. This means, you have to show that it is contractive for one of these two norms and not contractive for the other one.

(ii) Compute how many fixed point iteration steps you need to guarantee an accuracy of your solution of 10^{-2} when starting with $x^{(0)} = (0,0)^T$, i.e., for which $k \in \mathbb{N}$ will $\|x^{(k)} - x^*\|_\infty \leq 10^{-2}$ if x^* denotes the unknown solution.

(iii) Compute 5 steps of the fixed point iteration with initial guess $x^{(0)} = (0,0)^T$. How good is your result? I.e., compare $x^{(5)}$ to the analytic solution.

Exercise 2:
Let $g : \mathbb{R}_+ \to \mathbb{R}, x \mapsto x + \ln(x)$ be given, with $\mathbb{R}_+ := \{x \in \mathbb{R} : x > 0\}$.

(i) Show that $g(x) = 0$ admits a solution. You should show this analytically, i.e., a graphical solution is not sufficient.

(ii) Determine an approximate solution using a graphic.

(iii) Which of the following three methods would solve the equation and which one would you expect to solve it with least steps?

(1) $x^{(k+1)} = \ln(x^{(k)})$

(2) $x^{(k+1)} = e^{-x^{(k)}}$

(3) $x^{(k+1)} = \dfrac{x^{(k)} + e^{-x^{(k)}}}{2}$

(iv) Make 5 fixed point iteration steps for each method in (ii) which can solve the equation. Use $x^{(0)} = 0.5$ and compute $|x^{(j)} - x^{(j-1)}|$ in every step. What do you discover? Use 4 digits of accurarcy for your computation.

Exercise 3:
Let the nonlinear equation system

$$F(x) = F(x_1, x_2) = \begin{pmatrix} x_1^3 + x_2^3 - 4 \\ x_1^3 - x_2^3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

be given.

(i) Show that the Newton iteration to solve $F(x) = 0$ converges for every initial guess $x^{(0)} \in [1,2] \times [1,2]$.
Hint: Use the $\|\cdot\|_1$-norm and a theorem from the lecture.

(ii) Compute $x^{(1)}$ and $x^{(2)}$ using Newton iteration with the initial guess $x^{(0)} = (1,1)^T$.

(*)-Exercise 4: (6 points)
Show that linear equation systems, i.e., problems of the form find $x \in \mathbb{R}^n$ such that $Ax = b$ with $A \in \mathbb{R}^{n \times n}$ independent of x and $b \in \mathbb{R}^n$ independent of x, are solved within one step by the Newton iteration, i.e., $x^{(1)}$ is the solution of the system for every initial guess $x^{(0)}$.

Programming Exercise 4: (delivery date: 27.01.2011, 8 points)
Write a program which computes the solution of the system given in Exercise 3 up to an accuracy of 10^{-7}. Remember that you should not use the inverse in your programs.
Make a monotonicity test in every step with $L = 0.5$, i.e., test if the difference between the iterates converges monotone to zero by checking if the quotient between the current difference and the difference of the last step is smaller than L.
You have two options what to do when the program is aborted because of missing monotonicity

1. the program may ask the user for a new initial guess and restart
2. the program determines a new initial guess near to the old one and restarts.

Try different initial guesses for your program to test it and the monotonicity check. Give a documentation of the computing for at least three different initial guesses where one should be $(x_1^{(0)}, x_2^{(0)}) = (-0.3, 1.3)$.

The Newton method can algorithmically be described as follows.

```
x^(0) = (x_1^(0), x_2^(0))          % initial guess - input parameter
k = 1;                              % iteration count
Δx^(k) = (J_f(x^(k)))^(-1) · f(x^(k));

while(norm(Δx^(k)) > 10^(-7))

    x^(k+1) = x^(k) - (J_f(x^(k)))^(-1) · f(x^(k));

    Δx^(k+1) = (J_f(x^(k+1)))^(-1) · f(x^(k+1));
    Δx^(k) = (J_f(x^(k)))^(-1) · f(x^(k));

    Δ = norm(Δx^(k+1)) / norm(Δx^(k));

    if(Δ > L)
        disp('no monotonicity ');
        break;     % the program is aborted by leaving the while-loop
    else
        k = k + 1;
    end
end

if(norm(Δx^(k)) <= 10^(-7))
    disp(['Solution at' num2str(x^(k)) '.']);
end
```

Delivery: 20. January 2011