Universität Duisburg-Essen
Computational Mechanics
Fakultät für Mathematik

Dipl.–Math. Andreas Fischle
Dipl.–Math. Patrick Radtke

Summer Term 2012
05/24/2012

Advanced Numerical Methods. Exercise 6.

Exercise 1: (optional, 8 Bonus points)
Let all functions in this exercise have positive values for $x > 0$. Prove the following relations for the asymptotic behavior as $x \to \infty$:

1. Let $c > 0$ be a constant. Then, $O(cg) = O(g)$, i.e.,

$$f \in O(g) \Longleftrightarrow c \cdot f \in O(g)$$

2.
$$f_1 \in O(g_1) \wedge f_2 \in O(g_2) \Longrightarrow f_1 \cdot f_2 \in O(g_1 \cdot g_2)$$

3.
$$f_1 \in O(g_1) \wedge f_2 \in O(g_2) \Longrightarrow f_1 + f_2 \in O(g_1 + g_2)$$

4.
$$\forall k \in \mathbb{N} : O(x^k) \subset O(x^{k+1}) \subset O(e^x)$$

Exercise 2: (8 Points)

1. Compute the multivariate Taylor polynomials of order $k = 0, 1, 2, 3, 4$ of the function

$$f(x, y) = e^{-(x^2+y^2)}$$

   choosing $(x_0, y_0)^T = (0, 0)^T$.
   Hint: Partial derivatives taken w.r.t. to the same variables, but in different order, are identical. E.g., $\partial_{xy} f(x, y) = \partial_{yx} f(x, y)$, if f is sufficiently often differentiable.

2. Use MATLAB to plot each of the previously computed Taylor polynomials on the rectangle $[0, 2] \times [-2, 2] \subset \mathbb{R}^2$ together with the original function f (superimposed). The figures should show that the Taylor polynomials approximate f in the vicinity of $(x_0, y_0)^T$. It suffices to turn in a printout of the figures.
   Hint: Use `surf` or `ezsurf`.

Exercise 3 (Programming, 2 weeks time): (12 Points)
Rewrite your simulation of a particle in a force field such that:

- it uses the MATLAB ode solver commands, e.g., `ode23`, `ode45`, `ode113`. Your implementation should allow you to *pass a specific matlab solver as an argument* to your simulation.
- the position of the particle is plotted after each time step. This can be achieved by choosing the `odephas2` output function for the value of the property 'OutputFcn' and by setting the property 'OutputSel' appropriately. To set these properties, the command `odeset` is used.

Investigate which of the solvers `ode23`, `ode45` and `ode113` performs best for your simulation and argue on which basis you decided this. You might have to play with the tolerances, as described in e.g. `doc ode23`. You can measure the runtime with the commands `tic` and `toc`. A fair comparison should ensure that the stepsizes are comparable.

Due Date for theory: 05/31/2012 . 12:30h
Due Date for programming: 06/06/2012 . 12:30h