Numerical Analysis of Ordinary Differential Equations
Programming Exam

Summer semester 2016

Universität Heidelberg - IWR
Prof. Dr. Guido Kanschat, Dr. Dörte Beigel,
Stefan Meggendorfer, Philipp Siehr

Due: 26.07.2016 12 noon
by email to
`kanschat@uni-heidelberg.de`

Multiple shooting

Consider the Thomas-Fermi boundary value problem

$$\begin{aligned} u'(t) &= \begin{pmatrix} t u_2(t) \\ 4u_1(t)^{3/2} \end{pmatrix}, \quad t \in [0,5] \\ u_1(0) &= 1, \\ u_1(5) &= 0. \end{aligned}$$

Solve this BVP with a multiple shooting approach.

For the numerical approximation of the initial value problems use one of the integration methods you learned in the lecture. This method must be implemented by yourself, but you can use your work from previous homework assignments.

Wherever you need a nonlinear solver, you must implement the method yourself. You are allowed to use solvers for linear systems of equations from existing libraries or Matlab.

The program must be able to handle a variable number of shooting intervals and must be tested with 10 and 20 intervals. Find suitable initial values for the subintervals.

Plot the solutions and present results in a reasonable way.

Exam rules

1. You can prepare the assignment by yourself or in a team of two students.
2. Only programs which compile and run are considered. There is no partial credit for a program not producing results.
3. Program must be handed in electronically by July 26th, 12 noon.
4. There will be short oral tests with the authors of each program in order to verify authorship and help determining a grade. You will have to demonstrate the running program, possibly with a new BVP, and explain its details. These exams will be scheduled Tue July 26th to Thu July 28th.

5. Please submit at the oral test a signed declaration: “I/We have prepared the assignment myself/ourselves and I/we have only used the sources declared in comments to the program”.
6. In the hopefully unlikely event that you do not pass with your program, it counts as one exam attempt. You can choose between one of the two written exams for your second attempt.

Grading criteria

The following criteria will be used to determine a grade for the program. None of these is necessary, and additional achievements will be counted as well.

- well readable, modular program structure
- meaningful tests for correctness, for instance convergence order
- sophistication of methods, for instance
  - implicit time integrator
  - embedded time integrator with adaptive step size
  - Newton method with globalization
  - efficient implementation of Newton’s method
  - automatic, internal or external numerical differentiation

Warning: the sophistication criteria improve your grade **if the program works**. Therefore, rather have a running program first and make it more sophisticated later.