Technische Universität München
Institut für Informatik
Dr. habil. Miriam Mehl

WT 2011/12

Introduction to Scientific Computing

Final Exam, February 7th 2012

Name: ....
Matr.Nr.: ....
Programm: ....

General Instructions

Material: You may only use one hand-written sheet of paper (size A4, on both pages). All other materials including electronic devices of any kind are forbidden. You may use this exercise sheet and the exam paper that was handed out to solve the exercises (for notes and sketches, you can obtain additional exam sheets). Do not use a pencil or red or green ink.

General hint: Often, exercises b), c), etc. can be solved without the results from the previous exercise a): if you are stuck with exercise a, then do not immediately skip exercises b, c, etc.

Maximum score: The maximum score is 35 points. 15 points are required to pass the exam.

Working time: 90 minutes.

1 Population Modelling (3+3+1 pts)

We consider a given model for the population p and q of two species

$$\dot{p}(t) = \frac{3}{2} - p(t) + \frac{1}{2}q(t), \tag{1}$$

$$\dot{q}(t) = \frac{1}{2}p(t) - q(t). \tag{2}$$

a) Compute the critical points of this model.

b) Give the type of each critical point. Justify your answer!

c) Give the name of the model from **a)**.

2 Numerics for ODEs (3 pts)

We solve an ordinary differential equation

$$\begin{aligned} \dot{p}(t) &= f(t, p(t)) \text{ in }]0;T[, && (3) \\ p(0) &= p_0 && (4) \end{aligned}$$

numerically using the second order Runge-Kutta scheme

$$p^{(n+1)} = p^{(n)} + \frac{\tau}{2}\left(f\left(t^{(n)}, p^{(n)}\right) + f\left(t^{(n+1)}, p^{(n)} + \tau f\left(t^{(n)}, p^{(n)}\right)\right)\right), \qquad (5)$$

$n = 0, 2, 3, \ldots,$ where $t^{(n)} = n \cdot \tau$ and $p^{(n)}$ denotes our approximation for $p\left(t^{(n)}\right)$.

Determine the order of consistency of the given second order Runge-Kutta method.

3 Numerics for ODEs (6 pts)

We solve the linear differential equation

$$\begin{aligned} \dot{p}(t) &= \frac{1}{2}p(t) \text{ in }]0;10[, && (6) \\ p(0) &= 1 && (7) \end{aligned}$$

with the parareal method and explicit Euler time stepping.

Perform one parareal iteration with fine grid time step $\tau = 1$ and coarse time step $\tilde{\tau} = 5$ with initial guess $p(0) = p(5) = 1$. Give the solution vector after this iteration.

4 PDE Numerics (2+6+6+1 pts)

We solve the two-dimensional Poisson equation

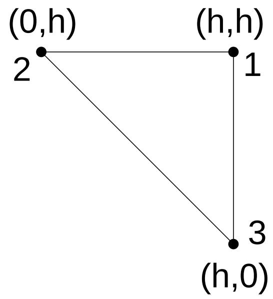
$$
\begin{aligned}
u_{xx}(x,y,t) + u_{yy}(x,y,t) &= f(x,y,t) \in \Omega, && (8)\\
u(x,y,t) &= 0 \text{ at } \partial\Omega && (9)
\end{aligned}
$$

on a domain Ω.

a) Give the weak form of (8).

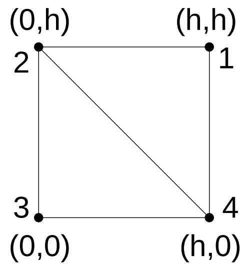
b) We solve (8) using finite elements on a triangular grid. Compute the lement stiffness matrix for the triangle displayed below with the nodal basis functions

$$\phi_1(x,y) = \frac{1}{h}x + \frac{1}{h}y - 1, \quad (10)$$
$$\phi_2(x,y) = -\frac{1}{h}x + 1, \quad (11)$$
$$\phi_3(x,y) = -\frac{1}{h}y + 1, \quad (12)$$



where ϕ_i is the basis function associated to node i of the triangle. Test and ansatz space are the same in our example (both given by $\phi_i, i = 1, 2, 3$ on the triangle).

c) Use the results from **b)** to assemble the matrix for the following grid consisting of two elements:



Hint: Consider only the given two elements without boundary conditions. Be carefull with the numbering of nodes. The resulting two-element matrix should use the numbering given in the sketch above.

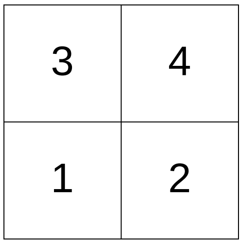
d) Give a reason why you might want to assemble element stiffness matrices instead of a discretization stencil.

5 Computational Grids (4 pts)

Assume you have the description of a tree-structured Cartesian grid given by the bit-code

$$11001000000110000000\,0. \tag{13}$$

In addition, you know that the grid is a two-dimensional octree grid (i.e. regular refinement of a square cell into four equal subsquares), the ordering of cells follows the so-called Morton order (equal on all levels with rotations or mirroring),



1 stands for a refined cell, 0 for a unrefined cell, and the traversal is done in a depth-frst order (descend to children before going on to the neighbour on the same level).

Sketch the adaptive Cartesian grid corresponding to the given bit-code.