Technische Universität München
Institut für Informatik
Dr. Miriam Mehl

ST 2011

Scientific Computing II

Final Exam, August 6, 2011

Note:

- **The exam is to be solved within 90 minutes, and the answers are to be written in German or English.**
- **Please try to answer all parts of the questions precisely and briefly (mind the number of credits assigned to a question).**
- **A handwritten sheet of paper (size A4, front and back page) may be used during the exam as mnemonic, as well as a dictionary (paper book!). No other material is allowed.**

Iterative Solvers (22 pts)

a) Parallel Gauss-Seidel (5 pts)

We want to solve the Poisson equation

$$\begin{aligned} \Delta u &= f \text{ in }]0;1[^2, \\ u &= 0 \text{ at } \partial]0;1[^2. \end{aligned}$$

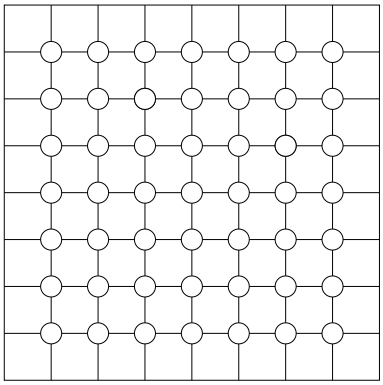
discretising the Laplace operator Δ using the stencil

$$\frac{1}{h^2}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Develop a parallel Gauss-Seidel variant analogue to the red-black algorithm, fulfilling the following requirements:

1) The method uses classes (colours) of grid points that can be processed independently, i.e. updating values at grid points of a class does not change the residual at other grid points of the same class.
2) One iteration of the method consists of a sweeps over all grids points of a class. i.e. one grid traversal per class.
3) The method uses a minimal number of classes/colours.

i) Mark the used classes of vertices with different colours in the Cartesian grid below. **($\approx$ 2 pts)**



ii) Write a pseudo-code of one resulting Gauss-Seidel iteration with your colouring. Avoid the usage of an explicitly stored system matrix. **(≈ 3 pts)**

b) Steepest Descent and Conjugate Gradients (11 pts)

i) We solve a linear system $Ax = b, A \in \mathbb{R}^{N\times N}, x, b \in \mathbb{R}^N$ with the steepest descent or conjugate gradient method. After how many iterations do we achieve the exact solution? **(≈ 2 pts)**

ii) Write a pseudo code for a variant of the conjugate gradient method solving the system $Ax = b$ with a symmetric **negative** definite matrix A. One iteration is sufficient. **(≈ 4 pts)**

iii) Assume, we would try to solve the system

$$\underbrace{\begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 10^1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 10^{N-1} & 0 \\ 0 & \cdots & \cdots & 0 & 10^N \end{pmatrix}}_{=: A} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N-1} \\ b_N \end{pmatrix}$$

using the conjugate gradient method. What is the convergence rate of the conjugate gradient method for this example? **(≈ 2 pts)**

iv) Would a Jacobi preconditioner improve the convergence of the conjugate gradient method for the example from **iii)**? Shortly justify your answer! **(≈ 3 pts)**

c) Multigrid Methods (4 pts)

We solve a system $Ax = b, A \in \mathbb{R}^{N\times N}, x, b \in \mathbb{R}^N$ using a multigrid method and look for a good smoother. Hereby, we can choose from a suite methods for which we know that the iteration matrix M with

$$e^{k+1} = Me^k,$$

where e^k is the error after iteration k, has eigenvectors

$$q_m = (\sin(m\pi hj))_{j=1,\ldots,N} \in \mathbb{R}, m = 1, \ldots, N, h = \frac{1}{N+1}$$

with eigenvalues

1) method 1: $\lambda_m = \frac{1}{2}(1 + \cos(m\pi h))$,
2) method 2: $\lambda_m = \cos(m\pi h)$,
3) method 3: $\lambda_m = \frac{1}{4}(3 + \cos(m\pi h))$.

Which of these methods is a good smoother in case of standard coarsening, that is a doubling of the mesh width h from the fine grid to the next coarser grid level? **($\approx$ 4 pts)**

d) Parallel Iterative Solvers (2 pts)

We want to solve a large two-dimensional Poisson equation

$$\begin{aligned} \Delta u &= f \text{ in }]0;1[^2, \\ u &= 0 \text{ at } \partial]0;1[^2. \end{aligned}$$

discretising the Laplace operator Δ using the 5-point stencil

$$\frac{1}{h^2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

with a multigrid or conjugate gradient solver on a distributed memory architecture. For our current problem size, multigrid and conjugate gradients are equally fast. Which method would you choose if we want to

i) further reduce the computing time by using more processors for a constant problem size, **($\approx$ 1 pts)**

ii) solve larger problems, that is reduce the mesh width h (and increase the number of processors proportional to the problem size). **($\approx$ 1 pts)**

Give short reasons for your answers.

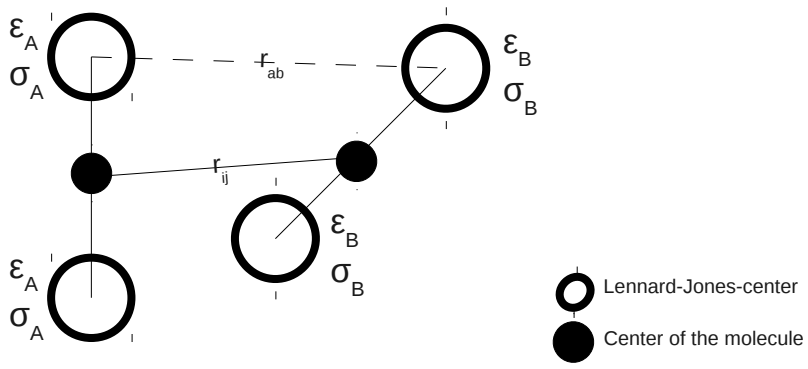
Molecular Dynamics (25 pts)

a) Molecular Modelling and Potentials (9 pts)

i) One simple model used in molecular dynamics is the Hardsphere Model. Sketch the underlying potential graphically and describe the basic algorithm (3 steps) of a simulation using it! **(≈ 2 pts)**

ii) A more realistic model is given by the Lennard-Jones-12-6 potential. Sketch the potential graphically (label the axes, mark prominent points indicating the potential parameters!). Which two physical interactions does it model? **(≈ 3 pts)**

iii) We want to simulate a mixture of two different fluids A and B. The molecules of both fluids are modelled as two-centered Lennard-Jones molecules, where both centers of a molecule have the same model parameters ϵ_A, σ_A and ϵ_B, σ_B, see the sketch:



Describe the calculation of the potential between the two molecules in detail! (Use formulas!) **(≈ 4 pts)**

b) Time Integration Methods (8 pts)

i) Shortly describe two important properties of time integration schemes in molecular dynamics! **(≈ 2 pts)**

ii) Compare the explicit Euler Scheme with a higher-order multistep method. Discuss advantages and / or disadvantages. **(≈ 3 pts)**

iii) The explicit Euler method is given by

$$r(t + \Delta t) = r(t) + \Delta t \cdot v(t) + O(\Delta t^2)$$

$$v(t + \Delta t) = v(t) + \Delta t \cdot a(t) + O(\Delta t^2)$$

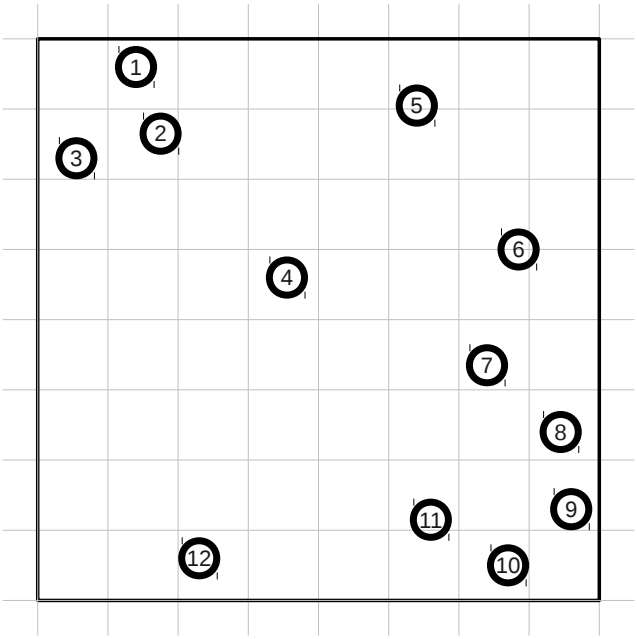
In the lecture it was claimed that this method is not time reversible. Show why this is the case for the position equation! **(≈ 3 pts)**

c) Methods for Long-range Potentials (8 pts)

In simulations using long-range potentials, each molecule influences each other molecule, so the force evaluation has computational complexity of $O(N^2)$. One method to calculate the long-range potential more efficiently is the Barnes Hut method.

i) What is the basic idea of the Barnes-Hut method? Shortly describe the algorithm! What role does the θ-criterion play? **($\approx$ 4 pts)**

ii) Construct and sketch the Barnes-Hut tree for the scenario below. Label the particles in the tree with their corresponding numbers from the figure! **($\approx$ 2 pts)**



iii) Do you know a better algorithm than the Barnes-Hut method? What is the basic idea of the improvement? Which complexity for the calculation of the potential do you achieve with that algorithm? **($\approx$ 2 pts)**