1st Exam

Fundamentals of Simulation Methods

WS 2020/21

Lecturers: Prof. Dr. Frauke Gräter, Prof. Dr. Friedrich Röpke

02. März 2021

1	2	3	4	Total

Name: ______________________ *Matrikel-Nr.:* ____________

1) Short questions [24pt]

State whether the following statements are right or wrong, and discuss in 1-2 sentences why this is the case. Note: There might be any number of correct statements.

a) Time integration in MD simulations:

1) Adaptive time stepping can be used to reduce the integration error of an Euler integration scheme.
2) The Verlet integration conserves momentum exactly and energy up to third order.
3) For the same accuracy, the Verlet scheme requires a two-fold smaller time step compared to the leapfrog integrator.
4) For a constant temperature simulation (e.g. NVT ensemble), particle positions are rescaled at each integration step.

b) Red black ordering:

1) Can be used within multigrid methods.
2) Shows better convergence than Jacobi iteration.

c) Tree method for modeling gravity

1) With the tree method, all gravitational forces in the system are computed exactly.
2) A problem of the tree method is that it breaks Newton's third law: the forces between two particles are not necessarily symmetric.

d) Smoothed particle hydrodynamics (SPH):

1) SPH conserves the total mass of the system exactly.
2) In SPH arbitrarily long time steps are allowed.
3) Smoothed particle hydrodynamics has problems with conserving the angular momentum of the system.

4) In order to treat hydrodynamical shocks in SPH, it is necessary to introduce an artificial viscosity.

2) Order of integration scheme [24pt]

We want to solve the following differential equation numerically:

$$\frac{dy(x)}{dx} = f(y(x)) \tag{1}$$

To achieve this we are given the following explicit updating scheme:

$$y_{n+1} = y_n + \Delta x f(y_n + A\Delta x f(y_n + B\Delta x f(y_n)) + C\Delta x f(y_n)) \tag{2}$$

Proof that there is no set of parameters $\{A, B, C\}$ that locally ensures this scheme to be accurate up to and including third order.

3) Metropolis-Hasting's [24pt]

Consider a 2D Ising Model, with N^2 spins, and $N = 10$.

$$E = -J \sum_{i,\langle i \rangle} s_i s_{\langle i \rangle}. \tag{3}$$

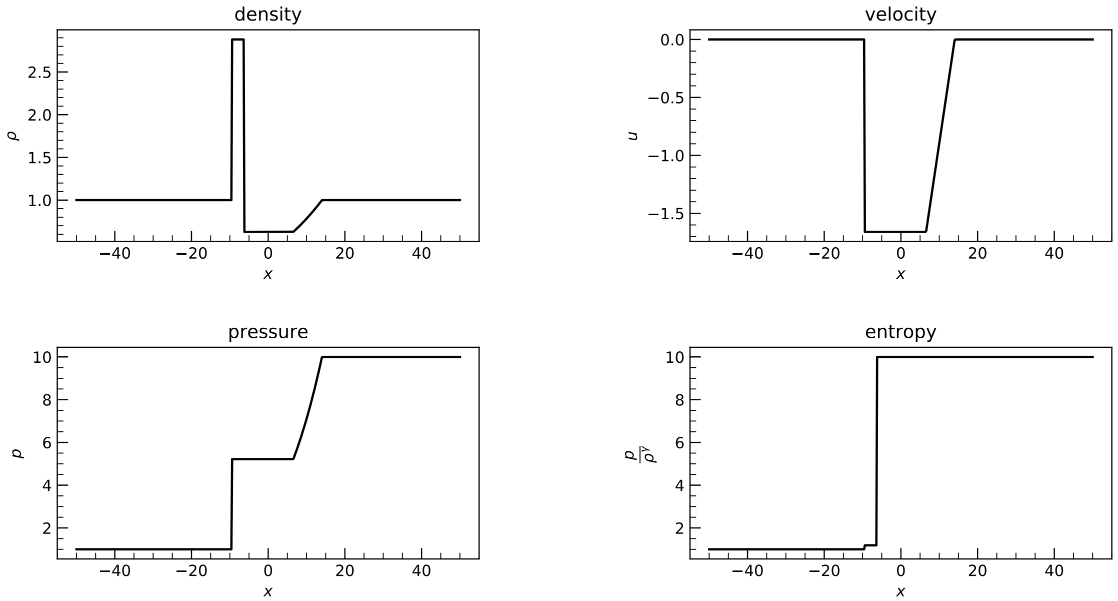
(a) Devise a piece of code that initializes the system with spins $s_{\mathbf{x}} = \pm 1$ randomly assigned to the lattice sites with equal probability. Write pseudo-code. Assume that you have a random number generator at hand that yields reals uniformly distributed between 0 and 1. Only nearest neighbors are considered, this means the sum$(i, \langle i \rangle)$ is only over pairs of i and nearest lattice sites $\langle i \rangle$; only their spin interaction is counted. A magnetic field is absent. $\beta = 1/T$ measures the temperature.

(b) Devise a second piece of code that uses the Metropolis algorithm to sample the spin states. Select a single lattice site, change the spin of this site, calculate the change in energy, and then accept or reject the spin flip. Again, write pseudo-code.

(c) Does this procedure fulfill detailed balance? Why?

(d) Would there be an advantage of using red-black ordering for selecting the lattice site in b), and if so, why?

4) Finite volume methods [24pt]



a) Describe the discretization approach in a finite volume method to solve the equations of fluid dynamics (in words or perhaps with the help of a sketch).

b) How is a Riemann problem defined? What wave patterns can arise from a Riemann problem for the case of the Euler equations of fluid dynamics?

c) Why is it useful in constructing finite volume schemes and how is it applied there?

d) The plots above illustrate the solution to a Riemann problem after some evolution time. What was the initial condition of this problem, i.e. (ρ, P, v) at $t = 0$?

e) In one of the homework problems, a finite volume scheme with an approximate Riemann solver (HLL) was discussed. In pseudocode it reads:

Algorithm 1 Finite Volume 1D code

```
Create grid
Set initial conditions
t=0
while $t < t_{max}$ do
    Get $\Delta t$ according to CFL condition
    Apply boundary conditions
    for each interface $i + 1/2$ do
        Get $\mathbf{q_L}, \mathbf{q_R}$ states at interfaces using constant reconstruction
        $\mathbf{F}_{i+1/2} = \mathbf{F}_{HLL}(\mathbf{q_L}, \mathbf{q_R})$
    end for
    for each cell $i$ do
        Get residuals: $\mathbf{R}_i = \frac{\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}}{\Delta x} \Delta t$
        Update (RK1): $\mathbf{q}_i = \mathbf{q}_i - \mathbf{R}_i$
    end for
    t = t + $\Delta t$
end while
```

In this form, the spatial discretization is only first order accurate. Why? How can the spatial discretization be improved to second order accuracy? Write a subroutine in pseudocode that does the trick and indicate where in the above code this subroutine has to be called.