Fundamentals of Simulation Methods

1st Exam – WS 20/21 – 02.03.2021

Name: Elias Olofsson, Matrikel-Nr.: 3674528

1. Short Questions. Question 1, Page 1 of 3.

a)

1.) Correct. By adaptively changing the size of the time step, errors in "steep" regions can be avoided by correspondingly making Δt smaller here.

2.) Incorrect. The Verlet scheme is a 2nd order integrator, but due to its symplectic properties, total energy and momentum are conserved if a Hamiltonian system is being simulated.

3.) Incorrect. Actually, the leapfrog is mathematically identical to the Verlet scheme, due there is no difference in the time step size required.

4.) Correct. This is an (bad) option to fix the temperature of the simulation. However, this also leads to residual distortions in the velocity distribution, which then does not accurately reflect an NVT-ensamble. An better alternative would be to use the Noose-Hoover thermostat.

b)

1.) Correct. Red-black ordering is a method to split the gridpoints into two disjoint sets for an iterative solver as the Jacobi-solver, which can enable better parallellization.

2.) Incorrect. Red-black ordering does not speed up convergence, it only enables parallellization of a iterative solver like Jacobi.

c)

1.) Incorrect. The tree method for gravity uses a truncated multipole expansion of distant groups of particles to estimate their influence on a target particle. I.e. it is an approximative calculation of the forces.

2.) Correct. The normal tree method can violate Newton's 3rd law since the multipole expansion is only done on one end. However, this can be remedied with "Fast" tree methods, where multipole expansions are done of both the target and source groups.

d)

1.) Correct. Since SPH is particle-based, if all particles remain in the system and no particle loses mass, the total mass is by construction conserved exactly.

2.) Incorrect. Typically one has to make Δt small enough such that thermal fluctuations of the particles does not traverse the typical kernel width h.

3.) Incorrect. In fact, SPH has excellent conservation properties of linear and angular momentum, energy, mass and entropy. Eulerian codes on the other hand have trouble conserving angular momentum.

4.) Correct. Since entropy is strictly conserved by definition in normal SPH, we need to introduce necessary dissipation in form of artificial viscosity, to treat shocks where is always <u>not</u> conserved.

2.

If we expand the exact solution:

$$y_{ex}(x+\Delta x) = y(x) + \dot{y}(x)\Delta x + \frac{1}{2}\ddot{y}(x)\Delta x^2 + O(\Delta x^3)$$
$$= y(x) + f(x)\Delta x + \frac{1}{2}\dot{f}(x)\Delta x^2 + O(\Delta x^3)$$

where we used the ODE $\dot{y} = f(y(x))$ in the last step.

Expanding

$$y_{n+1} = y_n + \Delta x\, f(y_n + A\Delta x\, f(y_n + B\Delta x f(y_n)) + C\Delta x f(y_n)$$
$$\approx y_n + \Delta x\left[f(y_n) + \dot{f}(y_n) A\Delta x f(y_n + B\Delta x f(y_n))\right]$$
$$+ C\Delta x f(y_n)$$
$$\approx y_n + \Delta x f(y_n) + \Delta x \dot{f}(y_n)$$

.....

$$y_{ex}(x+\Delta x) - y_{n+1} = \ldots + O(\Delta x^3)$$

(Ran out of time, could not complete task.)

3.

a)

```
# initialize
s = zeros(N, N)

for i in (0, N^2-1) do :
    u = rand(0, 1)          # uniform
    if u < 0.5 :
        s[i] = 1
    else :
        s[i] = -1
```

b)

```
for N_states do :

    s_old = s
    i = select lattice site (0, N^2-1) at random.

    if s[i] == 1 :
        s[i] = -1
    else :
        s[i] = 1

    E_new = 0
    for all nearest neighbour pairs (i, j)
        E_new = J s[i] s[j]

    r = min(1, exp[-β(E_new - E)])

    u = rand(0, 1)

    if u ≤ r :
        output(s)

    else :
        output(s_old
        s = s_old
```

c) Assuming symmetric proposal probability:

$$p_{eq}(E)\, W(E \to E_{new}) = \exp[-\beta E] \exp[-\beta [E_{new} - E]]$$
$$= \exp[-\beta E_{new}]$$

$$p_{eq}(E_{new})\, W(E_{new} \to E) = \exp(-\beta E_{new}) \exp[-\beta (E - E_{new})]$$
$$= \exp[-\beta E]$$

If $E \neq E_{new}$, which is the case here by construction we do not fulfill detailed balance.

d) Yes, since only the nearest neighbours are considered in the sum, they naturally form two disjoint sets which can be calculated independently.

4.

a) First, the differential form of the PDE is integrated over the volume of a discrete cell. Then, the cell averages of the state-vector are identified, these will then be represented by scalar values at the cell centers.

Then the Flux-functions, describing the flow through the cell interface, are identified.

b) A Riemann problem is an initial value problem for a hyperbolic system of two piecewise constant phases characterized by state vectors $\vec{u}_L$, $\vec{u}_R$ at $t=0$ for the left and right states. We then evolve the system in time $t>0$.

c) In finite volume schemes, we solve an Riemann problem at each cell-interface to determine the flux of the state vector between neighbouring cells.

d) By studying the edges of the diagrams, we can determine the initial conditions to

$$\vec{u}_L = \begin{bmatrix} \rho_L \\ p_L \\ v_L \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{u}_R = \begin{bmatrix} \rho_R \\ p_R \\ v_R \end{bmatrix} = \begin{bmatrix} 1 \\ 10 \\ 0 \end{bmatrix}$$

e) This scheme is only first order due to the constant reconstruction from the cell centres to the cell interfaces.

A second order scheme can be obtained by piece-wise linear reconstructions, i.e. we also have an estimate of the gradient in each cell. We adopt the MUSCL-Hancock scheme:

for each interface $i+\frac{1}{2}$ do:

Get gradient estimates (dq_i, dq_{i+1}) by finite differencing.

$$q_L = q_i + \left[\frac{\Delta x}{2}\mathbb{1} - \frac{\Delta t}{2}\mathbb{J}(q_i)\right] \cdot dq_i$$

$$q_R = q_{i+1} + \left[-\frac{\Delta x}{2}\mathbb{1} - \frac{\Delta t}{2}\mathbb{J}(q_{i+1})\right] \cdot dq_{i+1}$$

$$F_{i+\frac{1}{2}} = F_{HLL}(q_L, q_R)$$

end for

Here, $\mathbb{J}$ is a (analytically derived) jacobian for the current system being simulated.

We also extrapolate half a timestep $\frac{\Delta t}{2}$ forward for improved stability.