

APiE Assignments – Introduction

Each of the Assignments is about equivalent to 0.5 EC, except for ODE (which is very much overlapping with PiE so that it is worth less if you have done PiE already, and should be seen as a warm-up exercise), and FEM and SPH, which are both worth 1 EC. Voluntary and optional excersises (inside assignments and as solitary blocks) can be used to improve the grade or earn additional EC-points.

List of modular Assignments:

ODE

MD1D and MD2D for solids

FEM for solids (2x)

MD for Fluids

Random Numbers

SPH Methods (2x)

Finite Volume (FV) Methods

Debugging (Voluntary)

Guideline for APiE

0) The Exercises Report should be submitted in pdf! (Word causes problems regularly due to different versions and operating systems) - so you dont have to send the WORD.

please send in the matlab code(s) and other things as one archived (zipped, tgz, rar, ...) directory.

FORMAT (grading 10 possible points/grade per sheet)

1) Feel free to copy questions, however, that is not necessary - its matter of taste. Actually it would be nicer if the answer(s) (in words) would reflect the question(s). Thus: answers should have some words (sentences) - not only formulas. (required for grade 8 or higher)

2) Figures should have (grade 9 or higher) axis labels (both), a legend if there are more lines/symbols, and a caption - i.e. the short text (again words) below the figure that describes it briefly. Assume you see the figure for the first time - what do you want to know about it?

3) For each figure (in a good text - grade 10), you have the following issues: a) what is plotted (and why)? b) what do we see (observation)? c) what does it mean? ... d) if you have multiple lines on top of each other - think about how you can improve the plot to make them visible ...

CODING

4) Use indent-s within loops and if-s etc. and Add comments - sensible/important ones a) about the block/module b) for the non-trivial commands c) additional explanations, traps, warnings, etc. (required for grade 7 or higher)

5) Avoid identical pieces of code (duplications) ... If you decided to work with matlab: Use vectors/matrices/structures, whenever possible, to do so. ... You can use functions to do so too. (required if asked for - and required anyway for grade 9 or higher)

6) Read the warnings Matlab gives you a) Avoid growing vectors inside loops! declare outside! b) group declarations and definitions such that one can easily change things - dont repeat (change) parameters-changes - those are hard to locate. (required for grade 9 or higher)

... Think about that someone else might want to understand the code.

APiE Exercise – ordinary differential equations (ODE)

Exercise 1 (Ex.1–5 account for 2 points each; total 10 pts.)

The motion of a harmonic oscillator is described by the differential equation:

$$m \frac{d^2 x(t)}{dt^2} = m\ddot{x} = -kx(t) \quad (1)$$

where m is the mass, k is the spring-constant, $x(t)$ is the position and d^2x/dt^2 is the second time-derivative of $x(t)$.

- Get *analytically* the solution of Eq. (3) for initial conditions $x_0 = x(0)$ and $v_0 = \dot{x}(0) = v(0)$ and plot it as function of time for $x_0 = 0$ m and $v_0 = 0.1$ m/s. What is the period T and the frequency ω ? For the plot use $m = 1$ kg and $k = 0.1$ N/m.
- Now compute *analytically* the trajectory $x(t)$ and then plot the kinetic, the potential, and the total energy of the harmonic oscillator as function of time (together).
- Modify mass m and/or spring-constant k and describe how T changes.

Exercise 2.

- Solve Eq. (3) numerically using the Euler algorithm (as introduced in the course). Plot solution and energies together with the analytical solution of the previous Exercise 1. Compare them for different time-steps. Which time-step do you find satisfactory?
- Solve Eq. (3) numerically using the Euler-Cromer algorithm as introduced in the course. Plot solution and energies together with the analytical solution of the previous exercises 1. and 2.a) (use different colors). Compare them for different time-steps. Which time-step do you find satisfactory for Euler-Cromer? in comparison to Euler?

Exercise 3.

In order to solve this differential equation numerically, you can use the so-called Verlet integration algorithm. It was derived in class, but we also show the steps here - repeat them for yourself!

- Write down the Taylor-series for $x(t + \Delta t)$ and $x(t - \Delta t)$ up to second order in time (terms higher than Δt^2 are ignored).
- Add $x(t - \Delta t)$ to $x(t + \Delta t)$ and rewrite the equation such that you have $x(t + \Delta t)$ on the left-hand side of the equation.
- This scheme can now be used to compute the trajectory of the mass (with given $x(t)$, $x(t - \Delta t)$, and acceleration $d^2x/dt^2 = f(t)/m$, with force $f(t) = -kx(t)$). For the old position use the estimate: $x(t - \Delta t) \approx x(t) - v(t)\Delta t$.

4. Plot the old time-x-series and the new results, using the same amplitude, phase-angle = 0, $m = 1$, $k = 0.1$, $\Delta t = 0.1$, and $t_{\max} = 200$, $x(0) = 0$ and $v(0) = 0.1$.
5. How big is the error of the calculation? How do you quantify it? Or, with other words, which time-step would you use to have a reasonably small error here? (define reasonable)

Plot also the total energy of the numeric harmonic oscillator as function of time. Plot separately kinetic and potential energy. Describe the differences to the analytical solution.

Exercise 4

Add a damping force $f_d = -\gamma v(t)$ to the spring-force and plot the new results for a γ such that the motion is damped completely at $t = 200$. Which γ -value did you use?

Solve the new corresponding differential equation *analytically* first - and then compare it to one of your numerical solution algorithms. How did the period change when you added the damping force? Modify γ and describe how T changes with γ . Implement the damping force in your Verlet numerical solution from exercise 3. Discuss the problems that occur.

Exercise 5

Solve the differential equation (3) with a built-in Matlab function of your choice and compare to the previous examples/solutions – and answer the questions also for this integrator.

Which (power) class of error e have the different methods? (Definition: The error increases with power α for time-step h such that $e \propto h^\alpha$.)

Exercise 6 (Voluntary +3 points)

Generalize the spring-mass system by adding a driving force $f_d \sin(\omega_d t)$ to the right hand side of equation (3). Plot a "Poincare-cut" for three different ω_d (smaller, similar and larger than ω) and f_d (very small, small, and large) and describe your observations.

Derive the differential equation for a pendulum with a mass-less rigid rod that can swing around a fixed point with a mass m at the other end. Show the same plots as for the driven harmonic oscillator and describe the differences.

Exercise 7 (Voluntary +2 points)

Generalize the spring-mass system to a linear chain of N masses as will be detailed in the Lecture on Debugging and Efficient Programming.

APiE Exercise – Molecular Dynamics (MD) for Solids 1D

Exercise 1

Goal is to set up a linear 1D (one-dimensional) chain. First, generalize your linear spring ODE-program to 2 “particles” or atoms connected by one spring (see APiE-script).

Recommendation: Preparation for future programming in 2D:

Using the linear spring model, implement in your solver the interaction force using the normal vector $\hat{n} = (\vec{x}_i - \vec{x}_j)/|\vec{x}_i - \vec{x}_j|$, and the departure from the equilibrium position length $\delta = |\vec{x}_i - \vec{x}_j| - x_e$. Take care that the sign is correct.

Implement the force calculation in a function that receives the two particles and returns the force (scalar in 1D, vector in 2D). Then establish for each particle a loop over all particles it has a spring-connection with (this will be relevant below for the linear chain and later for 2D) and sum up all forces acting on a particle. For each particle pair (i,j) , the forces acting on i by j and reverse are related by $f_{i \leftarrow j} = -f_{j \leftarrow i}$.

Note: Make sure that you program modular. Separate variable definition, input, output, force-calculation and integration clearly as different modules – or functions.

Display the motion of the pair of particles for some time and also display the total energy and the kinetic and potential fractions.

Exercise 2

Generalize the program to N particles and implement:

(a) a linear chain with 11 particles, see Fig. 1

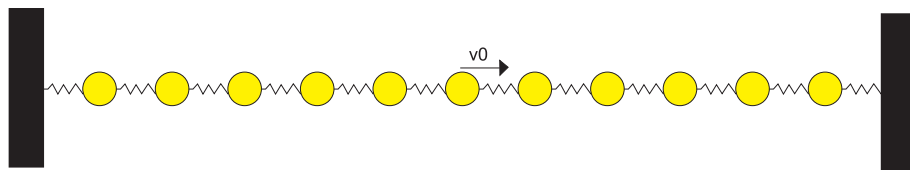


Figure 1: Linear Chain

The first and the last particle are connected to a fixed wall. The distance between the particles is x_e and this is equal to the equilibrium length of the springs. The central particle gets an initial velocity v_0 , all the other particles have initial velocity equal to zero.

Display the motion of the particles (in a graph).

(b) Write a function for the force-calculation using the method from above, such that the force calculation appears only once per particle-pair in the program. For this implement a loop over particle pairs.

Exercise 3

Visualize the movement of the particles in a movie. Let the color of each particle give a measure of the velocity of the particle.

Exercise 4 (Voluntary now – 2D is subject of the exercise MDSolids2D)

Generalize the pair of particles to a 2D (voluntary – 2 extra points are given for generalization to 3D) a square-system with $N=10 \times 10$ particles, see Fig. 2. *Advice: Implement the number of particles as a variable!*

Assign to the top particle on the right side an initial vertical and horizontal velocity v_0 , as shown in Fig. 2. All the other particles have initial velocity equal to zero, while their pair-wise separations are all equal to x_e . Set different spring stiffnesses for the vertical, horizontal, and diagonal springs.

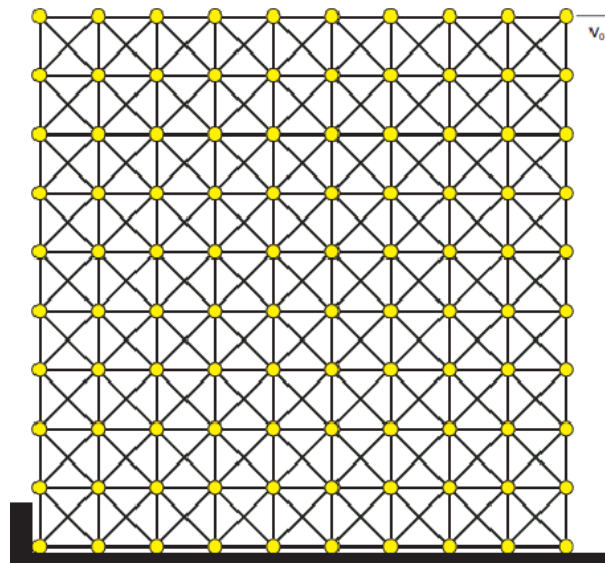


Figure 2: Square Lattice, where the bottom row is restricted to horizontal motion only $y = const.$ and the left-most particle is also fixed in horizontal direction $x = const.$.

Hints:

Make sure that you split the force-calculation and the integration (Verlet or Runge-Kutta) loops. Make sure that you set forces to zero at each new time.

Here an example algorithm (but better also see the script):

Loop 1: integration loop over time ($t_i = t_{i-1} + dt$)

- reset ALL forces ($f_x[\cdot] = 0$)
- **Loop 2:** over all particles ($i < N_{max}$)
 - **Loop 3:** over all contact partners ($j < i$)
 - * distance $dist = \sqrt{(x[i]-x[j])^2 + (y[i]-y[j])^2}$
 - * normal $n_x = (x[i]-x[j])/dist$
 - * overlap $delta = rad[i] + rad[j] - (x[i]-x[j])*n_x$
 - * contact: if $delta > 0$
 - relative velocity ($v_{rel} = -(v_x[i]-v_x[j])*n_x$)
 - interaction force ($f_x[i] = (k*delta + v*v_{rel})*n_x$)
 - partner interaction ($f_x[j] = -(k*delta + v*v_{rel})*n_x$)
 - **end Loop 3**
- temporary store position ($x_{tmp} = x[i]$)
- integrate ($x[i] = 2*x_{tmp} - s_x[i] + f_x[i]/m[i]*dt*dt$)
- save old position ($s_x[i] = x_{tmp}$)
- **end Loop 2**
- increase time ($t = t + dt$)

end Loop 1

end program

APiE Exercise - MDSolids2D

Consider the 2D spring network as shown in APiE Exercise MDSolids1D and below in Figure 3. (Each spring with masses is equivalent to a truss, as introduced later in the finite element part of the APiE-course.)

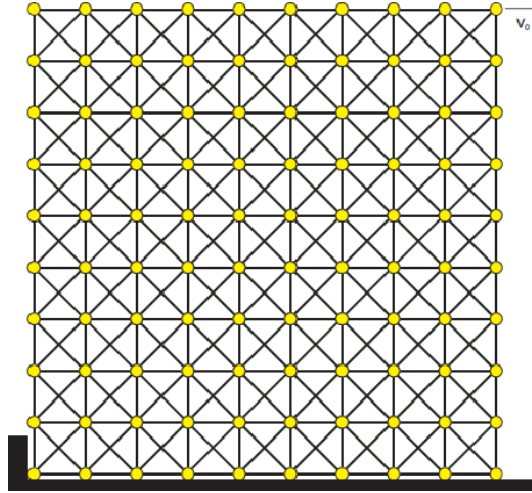


Figure 3: Square Lattice with nearest neighbor bonds and with additional diagonal bonds. The lowermost row of springs is connected to the wall, which can be represented by particles that do not move vertically - they are free to move horizontally. Only the lowermost left particle is fixed (constrained) in both directions!

We will now simulate a truss-network, i.e., a mass-spring network, which represents a 2D-solid, by a Molecular Dynamics (MD) simulation. Particles/atoms all have equal mass, here, and are connected by linear springs, which can be different (vertical/horizontal vs. diagonal). *This system will also be solved in the FEM part of the APiE course.*

- Choose the mass of the particles such that the mass of the total system is equal to the total mass of the truss-elements (springs) in the (square+diagonal-network) FEM simulation.

The parameters used in the FEM exercise are given here: Each truss/spring has a cross-sectional area $A = 0.0031 \text{ m}^2$, density $\rho = 7800 \text{ kg/m}^3$, and length 1 m (horizontal and vertical) resp. $\sqrt{2} \text{ m}$ (diagonal). Then the total mass is

$$[(2 \cdot 9 \cdot 10) \cdot L_0 + 2 \cdot 9^2 \cdot \sqrt{2}L_0] \cdot A \cdot \rho = 9892.1 \text{ kg}$$

Thus, each particle weighs $m = 98.921 \text{ kg}$. Verify this calculation.

- Choose the spring constant k such that the spring retains its Young's modulus of $E = \frac{F/A}{\Delta L/L} = 210 \text{ GPa}$. Note: The diagonal springs have different constants than the vertical and horizontal springs.

The Young's modulus satisfies $E = \frac{FL}{A\Delta L}$. A linear spring satisfies $F = k\Delta L$. Thus, $k = \frac{EA}{L} = 6.5100 \cdot 10^8 \text{ N/m}$ for horizontal and vertical springs, $k = 4.6033 \cdot 10^8 \text{ N/m}$ for diagonal springs. Verify this calculation.

- In some cases, the MD result has to be relaxed, i.e., the kinetic energy has to be dissipated. Then, choose a uniform damping coefficients $\gamma = 65020$ kg/s such that the horizontal/vertical springs have a restitution coefficient of $\epsilon = 0.5$, where the restitution coefficient is defined as the velocity after a half-period of the damped harmonic oscillator, divided by the initial velocity. Give the analytical solution for a mass-spring system from the ODE exercise, and for a two particle system from the MDSolids1D exercise.

The oscillation frequency of the springs is now $\omega = \sqrt{2k/m - (\nu/m)^2}$, the collision time is $t_c = \pi/\omega$. You should use a time step of $\Delta t = t_c/50$.

Exercise 1 (Implementation 5 pts. + animation 2 pts.)

When the structures are implemented in the 2D MD code, give an initial velocity to the upper right particle and animate the motion in a movie. Explain the differences/similarities in the reaction of the two structures.

Exercise 2 (Moduli and sound-speed measurement 3 pts.)

This question is equal to the FEM exercise, only here we use MD. After you have done both MD and FEM, comment on the agreement/disagreement between the two methods.

Using MD simulations, obtain the results for the Young's modulus E_{struct} , the shear modulus G_{struct} , the Poisson ratio ν_{struct} and sound speed V_s of the bulk structure – and compare them to the finite element simulation results (when they are available later). For more details see the script.

(a) The Young's modulus is defined as the ratio of longitudinal stress and strain, $E_{struct} = \frac{F_y/d/L_x}{\Delta L_y/L_y}$, where F_y is the total vertical force on all top-row particles (nodes), and d is the depth of the system (for trusses (see FEM exercise) the thickness has to be explicitly computed from the given data, i.e. from the area $A = \pi d^2/4$). The vertical longitudinal strain is the ratio of change of length to initial vertical length. Apply 10 kN vertical load in the positive (up) direction to all nodes (particles) in the uppermost row of the structures.

(b) From the same simulation, the Poisson-ratio can be measured as the ratio of (changes of) horizontal to vertical lengths $\nu_{struct} = \frac{\Delta L_x}{\Delta L_y}$.

(c) The shear modulus is defined as the ratio of shear-stress and -strain, $G_{struct} = \frac{F_x/d/L_x}{\Delta L_x/L_y}$, where F_x is the total horizontal force on all top-row particles (nodes). The shear-strain is the ratio of change of horizontal position to the vertical height. For this, apply 10 kN horizontal load in the positive (right) direction to all nodes (particles) in the uppermost row of the structures.

(d) Estimate the speed of sound (roughly) in the structure by observing the time t_s it takes for the applied force at the top (or the applied initial velocity from Ex.1) to become visible at the bottom. The speed can then be computed as $V_s = L/t_s$.

For the tests, plot the kinetic to potential energy ratio as a function of time. How long does it take until the kinetic energy is dissipated? Which criterion did you use? How does it change for different damping coefficients?

How does this change if you add so-called “global” background-damping, i.e., a damping force that is $\vec{f}_i = -\gamma\vec{v}_i$ negative proportional to the velocity of each particle relative to the background.

After each relaxation, check that the displacement at the left and right is “homogeneous”, i.e., the left and right nodes are aligned on a line. Discuss your observations.

Exercise 3 (Voluntary extra: 4 pts.)

Make the packing anisotropic. Either cut one of the diagonal springs, or make, e.g., the vertical springs stiffer. *Note that for anisotropic systems, one has to measure the Young- and Shear-moduli in all directions, since they are not equal anymore.*

Perform the same operations and calculations as above. Report and describe the differences.

Hint: look again at the hint of MDSolids1D

APIE Excercise – 1D FEM code for bar

(Note: You are NOT required to hand in solutions to this exercise.)

Goal:

Understand the various parts of the 1D Matlab code demonstrated in the lecture and carry out the following tasks:

1. Determine the analytical solution of the PDE below and compute the error between analytical solution and the Finite Element solution, plot the curve of this error w.r.t number of element in FEM.

$$\frac{d}{dx} \left(E(x) \frac{du}{dx} \right) = f, \quad x \in [0, 1]$$

and $u = 0$ at $x = 0$ and $f = x$ and assume $E(x) = E_0 = 1$.

2. Modify the 1D Matlab code provided during the lecture to accomodate a spatially varying Young's modulus of the bar such that,

$$E(x) = E_0(2 - x)$$

Where E_0 is a constant value.

3. Check the Eigen values/shapes of the global stiffness matrix, How does it differ from the case with constant E_0 ?
4. Now modify the code provided such that Young's modulus is spatially constant but depends on the displacement 'u' itself, i.e. system is nonlinear,

$$E(u) = E_0(1 - du/dx)$$

(Hint: To solve Non-linear problems you need iterative methods like Newton-Raphson)

Success!

APiE Excercise – FEM for 2D Trusses

Consider a 2D truss as shown in Fig.1. The horizontal and vertical truss elements have an initial length $L_0 = 1\text{m}$ and the diagonal elements have an initial length of $\sqrt{2}L_0$. All members have $E = 210\text{GPa}$, $A = 0.0031\text{m}^2$ and $\rho = 7800\text{kg/m}^3$. Constrain all the bottom nodes in vertical direction only (free along X) and constrain all the left side nodes in horizontal direction (free along Y),

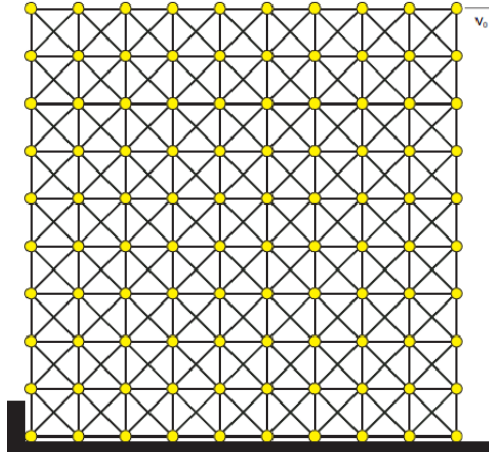


Figure 4: 2D truss model system (see text for details)

1. Apply a load 100 KN to the top right node +Y direction and use a finite element method to compute the displacement of this node. Compute the lowest 6 eigenpairs (eigenvalues and eigenvectors (also called mode shapes) of the stiffness matrix, check Matlab eig function for this). Plot the mode shapes and discuss your plots.
2. Considering the system as a solid structure, compute the Young's Modulus (E_{struc}), Shear Modulus (G_{struc}) and Poisson ratio (ν_{struc}) of the bulk structure using the following formula's ,
 - (a) Apply a 10 KN vertical load (+Y direction) on all the top nodes of the structure measure the elongation (ΔL).

$$E_{struc} = \frac{\text{Longitudinal Stress}}{\text{Longitudinal Strain}} = \frac{\text{TotalLoad}/(dW)}{\Delta L/L}$$

Where, $TotalLoad$ is sum of all the loads on the top nodes, and d is the cross-sectional diameter of the truss members (Compute using $A = \pi d^2/4$).

- (b) Also measure the change in width (ΔW) of the structure and compute Poisson ratio as,

$$\nu = \frac{\text{Change in transverse length}}{\text{Change in longitudinal length}} = \frac{\Delta W}{\Delta L}$$

- (c) To measure shear modulus first remove all the constraints on side nodes and fix *all* the bottom nodes in Y-direction, also fix just *one* corner(bottom) node in both X and Y directions. Now, apply a shear load (+X direction) of 10 KN on all the top nodes and compute G_{struc} as follows,

$$G_{struc} = \frac{\text{Shear Stress}}{\text{Shear Strain}} = \frac{\text{Total Load}/(dW)}{\Delta W/L}$$

here, ΔW is the displacement of the top left node.

3. {Optional: For extra points only} Solve the dynamic/transient case of this problem by the finite element method. To do this formulate mass-matrix of the structure and solve $[M_{global}]\{\ddot{U}\} + [K_{global}]\{U\} = \{f\}$. Assign to the top node on the right side an initial horizontal velocity $V_0 = 0.02m/sec$ Hint: This implies $\{f\}$ is zero, only initial condition drives the system. Visualize the motion of the truss in a movie. Use Newmark scheme and lumped mass matrix approach. Compute the lowest 6 eigenpairs (Solve the generalized eigenvalue problem). Plot the mode shapes and discuss your plots.
- (a) Estimate the speed of sound (roughly) in the structure by observing the time t_s it takes for the top-displacement to become visible at the bottom. The speed can then be computed as $V_s = L/t_s$.

APiE Excercise – FEM for 2D Trusses (Hint)

This is a quick overview of the steps for solving the Exercise FEM for 2D Trusses. As usual, there are different ways to implement the code - some more and some less efficient. For extremely large networks (1000 x 1000 or more) efficient solution techniques also matter and involve using sparse matrices, preconditioners and iterative solvers such as conjugate gradient etc. (see script/lecture for more details).

General Instructions:

- A mass-spring network is given, which is similar to a network of bars in compression/tension only i.e. a truss. Each node in the truss can move in both x,y directions (except nodes that are fixed at the bottom).
- As shown in the script (and lecture) you can write the FEM stiffness matrix for each bar/spring in the network. Given case is simple since there are only two orientations (0, 90) in the network. We STRONGLY encourage you to write a -function- which returns an element stiffness matrix (and force vector) and not compute matrix for each member individually. Thus, in Matlab you can have a function as follows,

```
function [Ke, Fe]= esf_2d(E,theta)
—compute element stiffness matrix—
—compute element force vector—
return Ke, Fe;
end
```

You may want to code it for arbitrary orientation (θ) and stiffness E or keep matters simple for now.

- Call the above function from the main program and assemble the element matrices in a global (big) matrix and apply boundary conditions and solve.

Algorithm

Step 1: Create a global numbering for each node in the network, store node locations (x, y) in a matrix.

Step 2: Create a connectivity array for all the elements in the network for e.g,

```
Elem(1,1) = 1;
```

```
Elem(1,2) = 7; %Element no. 1 is connected to global node number 1 and 7%
```

etc. ...

Step 3: Initialize element stiffness matrix ($[K_e]$) and element force vector ($\{F_e\}$) to zero.

Step 4: Initialize a global stiffness matrix ($[K]$) and global force vector ($\{F\}$). Determine the size based on total number of nodes $\times 2$.

Step 5: Loop over all the elements and compute element stiffness matrix ($[K_e, F_e]=esf_2d(200,90)$ etc.) and add $[K_e]$ and $\{F_e\}$ to $[K]$ and $\{F\}$ based on connectivity array.

Step 6: Apply boundary conditions i.e. remove rows and columns corresponding to the nodes which are fixed, this will reduce the size of your global stiffness matrix.

Step 7: Solve, $\{U\} = [K] \setminus \{F\}$.

Success!

APiE Exercise – Random Numbers

Exercise 1 – RNG (2pts.)

Implement a small Random Number Generator (RNG) yourself.

$$x_{i+1} = (c * x_i)_{modp} \quad (2)$$

with start “Seed” $x_0 \neq 0$.

Advice: Define a function `rng_init(seed)` that initializes the RNG and a second function `rng_irand()` that returns the integer random number, and another function `rng_rand()` that returns the random real number between 0 and 1.

While doing this, check for the smallest and the largest (integer) number the RNG returns and also check the period for:

a) $p = 2^{15} - 1$ and $c = 171$, and 65539

b) $p = 2^{31} - 1$ and $c = 171$, and 65539

What are the values?

From now on, in all following exercises, use your simple RNG and a matlab built-in. For some exercises also use a “bad” RNG in order to see and understand what is going wrong. Comment for all exercises on the results/performance of either.

Exercise 2 – Tests (3pts.)

Choose a seed number.

Perform the square test with your RNG – and with the built-in MATLAB random number generator (which one did you choose?). Perform the following tests for your RNG and the MATLAB RNG.

Compute the average of the random numbers for $M = 1, 10, \dots, 10^7$.

Compute the standard-deviaton of the random numbers for $M = 1, 10, \dots, 10^7$ and scale them by the mean.

(Voluntary) Compute the χ^2 average of the random numbers for $M = 1, 10, \dots, 10^7$, e.g. using the matlab function, and evaluate the shape and quality of the RNG.

Use the MATLAB Fourier-transform in order to analyse a series of 1024 random numbers. Explain, discuss the result.

Exercise 3 – Random Walk (3pts.)

Implement a simple 1-dimensional random walk starting at $x_0 = 0$, with $\Delta t = 1$ and $\Delta x = 1$ and compute $\langle x \rangle$ and $\langle x^2 \rangle$. Plot both as function of time for the two RNGs used for 2^{20} (time) steps.

What is the diffusion coefficient of this simple random walk expressed in Δt and Δx ?

Plot the probability distribution of positions after 2^{20} steps and compare it (fit it) with a Gaussian function. Discuss the relation between the width of the Gaussian and the distribution.

(Voluntary 2pts.) Write down (find) the differential equation that is solved by this Gaussian function. Solve the differential equation with another numerical method and with the random walk and compare the models, parameters, performance.

Exercise 4 – Integration (2pts.)

Compute π using your RNG and the MATLAB RNG. (From integration of a quarter circle, by choosing random points and counting them if they are inside the circle-section.)

How many random numbers do you need to get π with 2,3,4,5,6 digits accurate?

Do the same using a three-dimensional integral of a sphere.

Exercise 5 – Fractal (Voluntary 4pts.)

Implement the Random Midpoint Displacement Fractal, based on a square-lattice and iterate until level 8. Display the result with the MATLAB surface function and choose a nice color-scale.

APiE Exercise – MDFluids

Exercise 1 (2 pt)

The motion of the damped harmonic oscillator is described by the differential equation:

$$m \frac{d^2 x(t)}{dt^2} = m\ddot{x} = -kx(t) - \gamma \frac{d}{dt}x \quad (3)$$

where m is the mass, k is the spring-constant, $x(t)$ is the position and d^2x/dt^2 is the second time-derivative of $x(t)$.

Get analytically the solution of Eq. (3) for initial conditions $x_0 = x(0)$ and $v_0 = \dot{x} = v(0)$ and plot it as function of time.

The equation is dimensionless with the parameters $k = 0.1$, $\gamma = 10^{-4}$, $m = 1$, and initial conditions: $x_0 = 0$, $v_0 = 0.1$. What is the period T and the frequency ω ? The unit of length is $\sigma_0 = 0.1$ nm, the unit of density is $\rho = 2000$ kg/m³ and the unit of velocity is $v_0 = 0.001$ ms. What is the period T and the frequency ω in these dimensions?

That was the simple question.

Now make a table with 5 columns:

First column: name of quantity ... all of them ...

Second column: quantity as used in the simulation (dimensionless)

Third column: quantity with dimensions

Fourth column: quantity with dimensions (use SI-units!)

Fifth column: different new units - see next Ex.2.

Plot the kinetic and potential and total energy of the harmonic oscillator as function of time (together) in dimensionless units. Now translate the dimensionless units to SI-units and plot the same quantities again with units.

Exercise 2. (1 pt.)

Make the equation dimensionless using different units and report the results in column 5 of the table in SI-units. Now, the unit of time is $t_0 = 2.161 \times 10^{-12}$ s, the unit of length is $\sigma_0 = 0.34$ nm. (This corresponds to Argon.) Choose the unit of energy that the initial kinetic energy corresponds to $T_0 = E_0/k_B = 273$ Kelvin.

Insert the results in the table above and plot the corresponding energies in the new units as function of time.

As you will see the numbers are very ugly. This would change if you would simulate argon atoms and non-dimensionalize those.

Exercise 3. (2pt.)

Use your old MD program with the square-system connected by springs. Run it to see if it still produces the same results.

Change the initial conditions that all particles are at rest and only particle (1, 1) (at bottom-left) gets velocity $[1, 1]$, moving inside the system. Run again.

(i) Change the interaction force (linear spring) such that there is $f = 0$ when the particles are further away than r_c and $\vec{f}_{ij} = k(r_{ij} - r_{eq})\vec{n}_{ij}$ otherwise, with r_c the cutoff length, r_{eq} the equilibrium length, k the spring stiffness, and \vec{n} the unit-direction vector $\vec{n}_{ij} = (\vec{r}_j - \vec{r}_i)/|\vec{r}_j - \vec{r}_i|$. Use $r_c = r_{eq} = l_0$, where l_0 is the spacing of the lattice.

(ii) Change the boundary conditions such that the particles at the outside are NOT moving. Only the particles inside are moving.

(iii) Run the simulation again - and enjoy.

Plot the total energy of the system as function of time. Plot also kinetic and potential energy. Describe and explain what is happening.

Exercise 4 (2pt.)

Change the interaction force to the Lennard-Jones force with $r_c = l_0$ and $\epsilon = 1$ such that the cut-off happens in the minimum of the potential. (This leads to a purely repulsive LJ-potential and -force). Plot the total energy of the system as function of time. Plot also kinetic and potential energy. Describe and explain what is happening. Compare to the previous simulations above. Set the energy unit such that they become comparable in energy.

Exercise 5 (1pt. + 3pt.)

ALTERNATIVE (collaboration is encouraged! 2pt. extra):

(i) Keep the system as is and jump to the next task.

(ii) Implement flat walls outside your system - l_0 away from the outermost particles.

(iii) Implement periodic boundaries.

Then: Compute the pressure in the system according to the force on the walls - not for (iii) - or using the Virial theorem.

Compute (and plot) the pressure for different densities and/or different temperatures.

Exercise 6 (1pt. + 3pt.)

ALTERNATIVE (collaboration is encouraged! 2pt. extra):

(i) Compute the velocity distribution function - first in each direction, $f(v_\alpha)$, with $\alpha = x, y, z$ and then the $f(|v|)$. Plot the functions and fit them with either Gaussian or Maxwell distribution. Compare the fit-factors with temperature - are the results consistent?

(ii) Compute the pair correlation function - of particles away from the wall. For this the system has to have a certain minimal size. Compare the pair-correlation functions for different densities/temperatures.

(iii) Compute the $\langle r^2 \rangle$ and from that the diffusion coefficient.

APiE Exercise – SPH

Exercise: Sod Shock (1D) using SPH (20 pt. total)

The Sod Shock is a common test case for CFD codes. It involves the simulation of a 1D Riemann problem for an ideal gas. To the left of a discontinuity at $x = 0$ the parameters of the gas are $(v_l, \rho_l, P_l) = (0.0, 1.0, 1.0)$. To the right the parameters are $(v_r, \rho_r, P_r) = (0.0, 0.125, 0.1)$.

Use SPH to calculate the evolution of a Sod Shock over the spatial domain $-0.5 < x < 0.5$. Write a report on your solution at $t = 0.2s$. Include plots of the density and the thermal energy and discuss the solution near the front of the rarefaction wave (see label 'A' in Figure 1), the contact discontinuity (label 'B') and the shock (label 'C'). As a guide, Figure 1 shows the analytical solution for the density at $t = 0.2s$. Compare your solution against this plot.

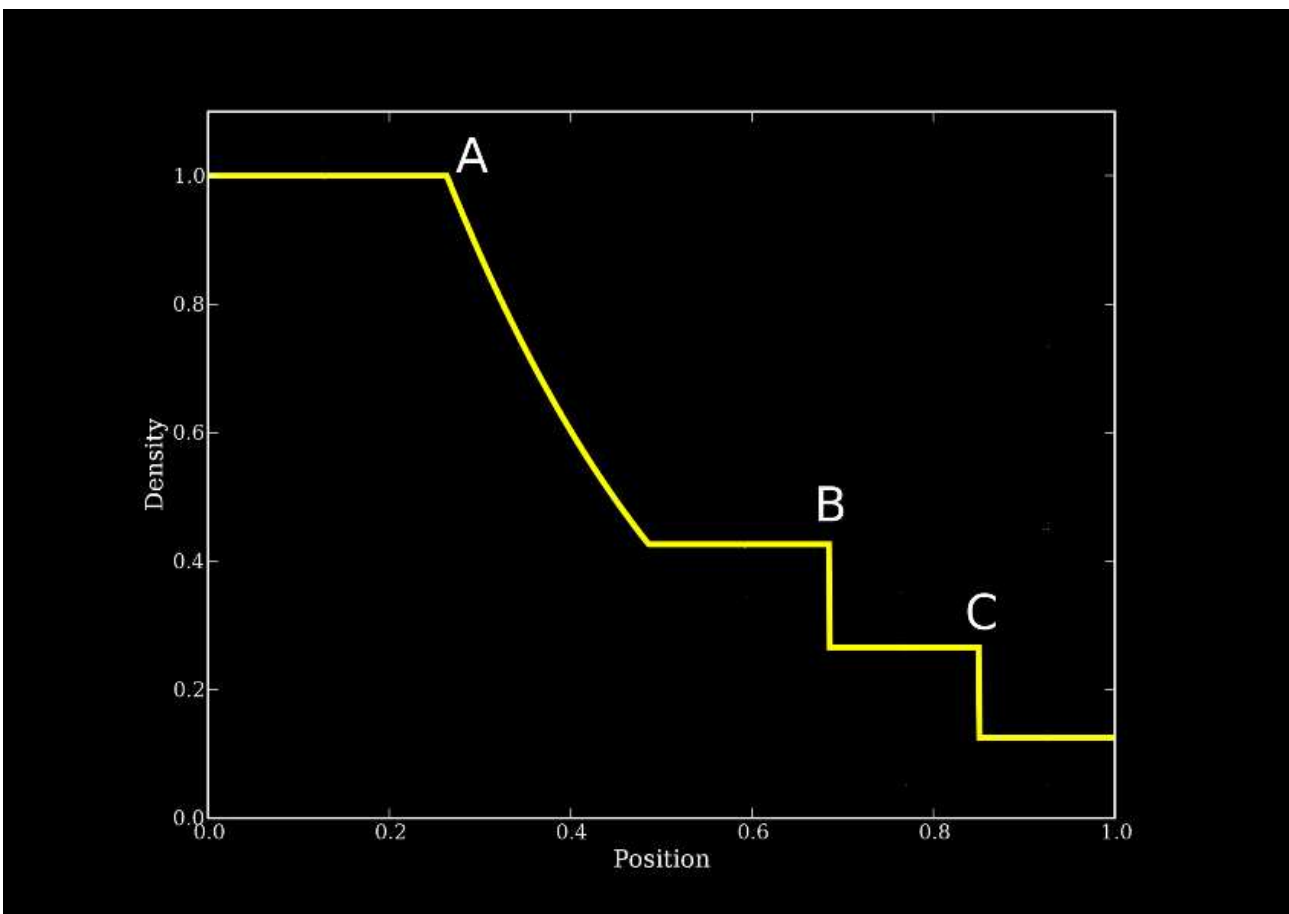


Figure 5: Plot of density at $t = 0.2s$ for the given Sod Shock parameters

Use the following equation of state for an ideal gas in terms of the density ρ and thermal energy u

$$P = (\gamma - 1)u\rho, \quad (4)$$

with an adiabatic index of $\gamma = 1.4$. This parameter choice is due to the test case specifications. For a more detailed explanation of the Sod Shock parameters and the equation of state used, please see the paper by Sod [G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comp. Phys. 27(1):131, 1978].

Implement the following SPH equations for density, acceleration and rate of change of thermal energy for particle a :

$$\rho_a = \sum_b m_b W_{ab} \quad (5)$$

$$\frac{dv_a}{dt} = - \sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} + \Phi_{ab} \right) \nabla_a W_{ab} \quad (6)$$

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a^2} \sum_b m_b v_{ab} \cdot \nabla_a W_{ab} \quad (7)$$

where $v_{ab} = v_a - v_b$.

Use the following form for the artificial viscosity in order to stabilise the shock

$$\Phi_{ab} = - \frac{K v_{sig}(v_{ab} \cdot x_{ab})}{\bar{\rho}_{ab} |x_{ab}|} \quad (8)$$

with $K = 1.0$ and signal velocity $v_{sig} = 0.5(c_a + c_b - 4v_{ab} \cdot \hat{x}_{ab})$. Only apply the viscosity for particle pairs that are approaching each other ($v_{ab} \cdot x_{ab} < 0$).

Integrate the particle variables using the following timestep

$$\Delta t = 0.3 \min \left(\frac{h_a}{v_{sig}} \right) \quad (9)$$

where the function 'min' calculates the minimum value over all the SPH particles.

Use the following kernel function

$$W(r, h) = \frac{2}{3h} \begin{cases} 1 - \frac{3}{2}(r/h)^2 + \frac{3}{4}(r/h)^3 & \text{for } 0 \leq r/h < 1 \\ \frac{1}{4}[2 - (r/h)]^3 & \text{for } 1 \leq r/h < 2 \\ 0 & \text{for } r/h \geq 2 \end{cases} \quad (10)$$

and use a symmetric kernel between particle pairs (e.g. $W_{ab} = W(x_a - x_b, 0.5(h_a + h_b))$).

Use a smoothing length that varies with the density so that

$$h_a = 1.3 \frac{m_a}{\rho_a} \quad (11)$$

It might be easier to modify one of your old MD codes to an SPH code rather than write it from scratch. Compute the SPH-hydrodynamic acceleration using Eq. (6) and use it instead of f_i/m_i in the MD program for integration. Integrate the thermal energy u in the same manner as the particle velocity using Eq. (7). At the beginning of each acceleration computation make sure to update the density using Eq. (5) and the pressure P using the given equation of state in Eq. (4).

Hint: In order to model the initial density discontinuity, use different particle spacings to the left and right of $x = 0$. Aim to use approximately 450 particles, but feel free to experiment.

Hint: Do not evolve the variables for the first and last 4 particles in the line. These particles will model the boundary condition.

Hint: For radially symmetric kernels the following applies:

$$\nabla_a W_{ab} = \frac{\mathbf{r}_{ab}}{|\mathbf{r}_{ab}|} \frac{\partial W_{ab}}{\partial \mathbf{r}_a}. \quad (12)$$

APiE Exercise - Finite Volume Methods (Scalar Eqn's)

Part (i) is compulsory and is required to pass the assignment. The other parts are all optional and are only required for the higher grades. It should be noted that part (ii) – (iv) are independent i.e. it is possible to complete (i) and (iii) or (i) and (iv) only etc...

(i) Write a simple finite volume code to solve the one dimensional linear advection equation (without limiters)

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho}{\partial x} = 0.$$

Implement inflow on the left and outflow conditions on the right. Choose a numerical flux. Consider both

(a) square wave initial conditions

$$\rho(x) = \begin{cases} 0 & 0 \leq x \leq 1 \\ 1 & 1 < x \leq 3 \\ 0 & 3 < x \leq 10 \end{cases}$$

and (b) triangular wave initial conditions

$$\rho(x) = \begin{cases} x & 0 \leq x \leq 1 \\ (2 - x) & 1 < x \leq 2 \\ 0 & 2 < x \leq 10 \end{cases}$$

In the report please show the time evolution of the solution and explain how you implemented the boundary conditions.

(ii) Optional: Repeat question(i) for the one dimensional inviscid Burgers equation

$$\frac{\partial \rho}{\partial t} + \frac{1}{2} \frac{\partial \rho^2}{\partial x} = 0.$$

(iii) Optional: Investigate the effect of adding a limiter, try minmod, superbee and Woodward. Comment on which gives the best results for each problem.

(iv) **Voluntary +1 points:** For the problem in (i) find the exact solution. For different grid resolutions consider the difference between the exact solution and your numerical solution. Hence comment on the error in your numerical method as a function of the grid size.

APiE Exercise - Finite Volume (System of Equations)

Exercise 1

Part (i) and (ii) is compulsory and is required to pass the assignment. The other parts are all optional and are only required for the higher grades.

Extend your finite volume code to solve the isothermal Euler equations, which are of the form

$$\frac{\partial \vec{\omega}}{\partial t} + \frac{\partial \vec{f}(\vec{\omega})}{\partial x} = 0,$$

with

$$\vec{\omega} = \begin{pmatrix} \rho \\ \rho u \end{pmatrix} \text{ and } \vec{f} = \begin{pmatrix} \rho u \\ \rho u^2 + a^2 \rho \end{pmatrix}$$

(i) Initially solve the equations in a unit domain with inflow on the left and outflow on right side, with uniform density $\rho = 1$ and velocity $u = 1$. Prescribe as inflow $\rho = 0.5$ and $u = 1$. Plot the evolution of the solution until it reaches steady state.

(ii) Place a partition wall at $x = 1$ and give the fluid a uniform density and zero velocity to the left of this wall, place no material to the right of the wall (you may need to place a small density to the right). Start your code at $t = 0$ with the wall already removed. Plot the evolution of the density profile with time. You should see a sharp front (shock) propagating into the empty space.

(iii) Optional: Change to solid walls at $x = 0$ and $x = 2$.

(iv) Optional: Investigate the effect of implementation a limited for case (ii).

(v) Optional: Add a Hancock predictor step, i.e. change the scheme into a Modified Total Variation Diminishing Scheme (MTVDLF). Comment on the effect of adding this step. You may want to also use the problem from the previous examples sheet, where you could write down the exact time-dependent solution (Note: The wave speeds for the Euler equation are $u \pm |a|$).

Exercise 2 (Voluntary +3 points)

ADVANCED : Use dimensional splitting to extend your code to two dimensions. Consider a square domain with all solid walls and the initial conditions

$$\rho(x, y) = \begin{cases} 0.1 & 0.5 > (x^2 + y^2) \\ 1 & 0.5 \leq (x^2 + y^2) \end{cases}$$

and $u = 0$ and $v = 0$ everywhere.

This problem is a circular explosion trapped in a square room.