# Fortran Programming Exercises for Materials Science Students

## General Instructions

All exercises are designed to be solvable in roughly one hour each, assuming basic knowledge of Fortran (variables, loops, arrays, simple file I/O).

Use modern Fortran (Fortran 90 or later) and `implicit none` in all programs. Double precision arithmetic is recommended (e.g. `real(8)`).

## Exercise 1: Density of Crystalline Solids

In this exercise, you will compute the mass density of cubic crystals (SC, BCC, FCC).

For a cubic crystal, the density is

$$\rho = \frac{nM}{N_\mathrm{A} a^3},$$

where

- $\rho$ – density in $\mathrm{g\,cm^{-3}}$,
- $n$ – number of atoms per unit cell:
  - SC: $n = 1$,
  - BCC: $n = 2$,
  - FCC: $n = 4$,
- $M$ – atomic mass in $\mathrm{g\,mol^{-1}}$,
- $N_\mathrm{A}$ – Avogadro's number, $N_\mathrm{A} \approx 6.022 \times 10^{23}\ \mathrm{mol^{-1}}$,
- $a$ – lattice parameter in cm.

### Tasks

1. Write a Fortran program that:
   (a) Asks the user to input
   - the crystal structure as a string: `"SC"`, `"BCC"` or `"FCC"`,
   - the atomic mass $M$ in $\mathrm{g\,mol^{-1}}$,
   - the lattice parameter $a$ in Å.

   (b) Converts the lattice parameter from Å to cm.

   (c) Chooses the correct number of atoms per unit cell $n$ based on the crystal structure.

   (d) Computes the density $\rho$ in $\mathrm{g\,cm^{-3}}$.

   (e) Prints the input data and the computed density in a clearly formatted way.
2. Include basic error handling for an invalid structure string (e.g. if the user inputs something other than SC, BCC, FCC, print an error message).

## Exercise 2: Stress–Strain Curve and Young's Modulus

You are given a file containing data from a tensile test. The file, called `stress_strain.dat`, contains two columns:

- column 1: strain $\varepsilon$ (dimensionless),
- column 2: stress $\sigma$ in MPa.

Young's modulus $E$ can be estimated from the slope of the initial linear elastic region of the stress–strain curve.

### Tasks

1. Write a Fortran program that:
   (a) Reads the data from `stress_strain.dat` into two arrays, `eps(:)` and `sigma(:)`.
   (b) Asks the user to input a maximum strain value $\varepsilon_{\text{max}}^{\text{lin}}$ defining the end of the linear region (for example, $\varepsilon_{\text{max}}^{\text{lin}} = 0.01$).
   (c) Selects only those data points with $\varepsilon \leq \varepsilon_{\text{max}}^{\text{lin}}$.
   (d) Computes Young's modulus $E$ using a simple linear regression (least squares) of $\sigma$ versus $\varepsilon$. For data points $(\varepsilon_i, \sigma_i)$,
   $$E = \frac{\sum_i(\varepsilon_i - \bar{\varepsilon})(\sigma_i - \bar{\sigma})}{\sum_i(\varepsilon_i - \bar{\varepsilon})^2},$$
   where $\bar{\varepsilon}$ and $\bar{\sigma}$ are the mean values of strain and stress in the selected region.
   (e) Prints the estimated Young's modulus $E$ in MPa and in GPa.
2. Optionally, write the selected linear-region data to a new file, `linear_region.dat`.

## Exercise 3: 1D Heat Diffusion in a Rod (Finite Differences)

We consider transient heat conduction in one dimension,

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2},$$

where $T$ is temperature and $\alpha$ is the thermal diffusivity.

We discretize the equation using an explicit finite difference scheme,

$$T_i^{n+1} = T_i^n + \frac{\alpha \Delta t}{\Delta x^2}\left(T_{i+1}^n - 2T_i^n + T_{i-1}^n\right).$$

### Physical setup

- Rod length: $L$ (in meters).
- Number of interior grid points: $N_x$.
- Thermal diffusivity: $\alpha$ (in m$^2$/s).
- Total simulation time: $t_{\text{final}}$ (in seconds).
- Initial temperature: $T(x, 0) = 300$ K everywhere.
- Boundary conditions:
  - Left end ($x = 0$): $T = 400$ K for all times.
  - Right end ($x = L$): $T = 300$ K for all times.

### Tasks

1. Write a Fortran program that:

    (a) Declares and allocates arrays to store the temperature field $T(x)$ along the rod.

    (b) Sets up the spatial grid with spacing $\Delta x = L/(N_x + 1)$.

    (c) Uses a time step $\Delta t$ such that the stability condition

    $$\frac{\alpha \Delta t}{\Delta x^2} \leq 0.5$$

    is satisfied.

    (d) Initializes the temperature field with the given initial and boundary conditions.

    (e) Advances the solution in time using the explicit scheme until $t_{\text{final}}$.

    (f) Writes the final temperature profile $T(x)$ to a file `T_profile.dat` (e.g. two columns: $x$ and $T$).

2. Optionally, also write out intermediate temperature profiles at a few selected times for later plotting.

## Exercise 4: Mean Squared Displacement from a Random Walk

Random walks can model atomic diffusion. We consider a one-dimensional random walk where, at each time step, a particle moves either left or right by a fixed step size $a$ with equal probability.

### Tasks

1. Write a Fortran program that:

    (a) Asks the user for:

    - the number of walkers $N_{\text{walk}}$,
    - the number of steps $N_{\text{steps}}$,
    - the step size $a$.

    (b) For each walker, starts at position $x = 0$ and performs a random walk with $N_{\text{steps}}$ steps.

    (c) Uses Fortran's intrinsic random number generator to decide for each step whether to move $+a$ or $-a$ (with probability 1/2 each).

    (d) Stores the final position $x_k$ of each walker.

    (e) Computes the mean squared displacement (MSD),

    $$\text{MSD} = \frac{1}{N_{\text{walk}}} \sum_{k=1}^{N_{\text{walk}}} x_k^2.$$

    (f) Prints the MSD and compares it qualitatively with the expected linear dependence on $N_{\text{steps}}$.

2. Optionally, write the final positions of all walkers to a file `positions.dat`.

# Exercise 5: Simple Equation of State – Pressure vs. Volume

We consider a simple equation of state relating the pressure $P$ to the volume $V$:

$$P(V) = B_0 \left( \frac{V_0}{V} - 1 \right),$$

where:

- $V_0$ – equilibrium volume,
- $B_0$ – bulk modulus,
- $V$ – volume.

### Tasks

1. Write a Fortran function

   ```
   pressure(V, V0, B0)
   ```

   that returns $P(V)$ given $V$, $V_0$ and $B_0$.

2. In the main program:

   (a) Ask the user to input $V_0$ and $B_0$.

   (b) Generate a set of volumes $V$ between $0.8V_0$ and $1.2V_0$ (e.g. 21 points).

   (c) For each volume $V$, compute the corresponding pressure $P(V)$ using the function.

   (d) Write $V$ and $P(V)$ to a file `eos.dat` in two columns.

3. Optionally, extend the program to implement a more realistic equation of state (e.g. Murnaghan or Birch–Murnaghan), selectable via a simple menu.