

# HW2: Fortran exercises

A. If you are unfamiliar with Fortran, work your way through the tutorials on the Canvas > All Pages > Lecture notes webpage.

B. Build and test Fortran programs that do the following:

## 1. trap.f90

Integrates a simple function  $f(x)$  between two limits using the Trapezoidal rule (see next page):

Your code will need to:

- Have a Fortran function routine that you can hardwire the desired  $f(x)$  into
- Have a main routine that:
  - ✓ Requests the upper and lower limits of integration from the user
  - ✓ Requests the number of intervals  $N$  in the quadrature required from the user
  - ✓ Perform the integration using a DO loop
  - ✓ Output the result

- Test your program on  $\int_0^2 x^2 dx$  and  $\int_0^\pi \sin(x) dx$

Q1: What are the right answers? Q2: How many intervals do you have to use to get a decent answer?

## 2. ones.f90

Dynamically creates a square array of a size input by the user at runtime

Assigns certain of the values in the array to be 1's and the rest 0's (by some method = random, input by user, etc)

Calculates another array that contains a 1 at any location if exactly 3 of the 8 surrounding neighbouring locations in the original array contain 1's.

# The trapezoidal rule

$$\int_a^b f(x) dx \approx \sum_{k=1}^N \frac{f(x_{k-1}) + f(x_k)}{2} \Delta x_k$$

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$$

$$\Delta x_k = x_k - x_{k-1}$$

Uniform grid version:

$$\int_a^b f(x) dx \approx \frac{\Delta x}{2} \sum_{k=1}^N (f(x_{k-1}) + f(x_k))$$

$$\Delta x_k = \Delta x = \frac{b - a}{N}$$

# HW2: Fortran exercises

Example of how to update one point (the red point):

Original matrix

	1	0	1	
	1	0	0	
	0	0	0	

New matrix containing updated values

		1		

At cell(i,j)

If (sum\_of\_neighbors = 3) then updated\_value = 1 else 0

# HW submission

Submit to Canvas a **tar file** that contains:

1. Your source code
2. A text file that contains the commands necessary to compile and run your code
3. Sample output of your code tests
  - the answers to the integrals for various numbers of intervals  $N$ , showing convergence towards the answer (answering Q1 and Q2)
  - an array of assigned 1,0's and the subsequent calculated array determined by the neighbours

Note: To create a tar file called “**your\_username.tar**” containing **<file list>** files, you can use the Linux command: `tar cvf your_username.tar <file list>`