# PH502: Scientific Programming Concepts

Irish Centre for High End Computing (ICHEC)

September 23, 2020

# Overview

- In this lecture we are going to talk about more advanced data types.
- We know about the basic variable types e.g. int, float, double ...
- More complex types can be constructed from the basic ones.
- These are called structs in C and types in FORTRAN.
- As an example we are going to categorise principal minors of a matrix by their determinant.

# Principal Minor

$$\begin{pmatrix} 0 & 1 & 2 & 3 & \cdots & n \\ 1 & 2 & 3 & 4 & \cdots & n+1 \\ 2 & 3 & 4 & 5 & \cdots & n+2 \\ 3 & 4 & 5 & 6 & \cdots & n+3 \\ \vdots & & & & & \vdots \end{pmatrix}$$

# Structs and Types

- Supposing you wanted to classify the 4x4 principal minors of a matrix.
- For each class for a particular order you want to store:
  1. Number (integer)
  2. Determinant (float)
  3. Example matrix (array[4][4])
- Representing the above data structure can be achieved using structs.

```
struct pmclass {
  int num;
  float det;
  float exam[4][4];
};
```

```
type pmclass
  integer (kind=4) :: num
  real (kind=4) :: det
  real (kind=4) :: exam(4,4)
end type
```

- Structs/Types allow explicit associations between variables which can make the program easier to interpret. It can reduce the number of different variables needed or reduce memory usage. (Consider the data structure of different orders).

# Using Structs/Types

## C

- Below are examples of using elements of a struct or type.

```
int i,j,k,Nclasses;
struct pmclass pmc[Nclasses];

for (i=0; i<Nclasses; i++) {
   pmc[i].num = 0;
   pmc[i].det = 0.0;
   for (j=0; j<4; j++) {
     for (k=0; k<4; k++) {
       pmc[i].exam[j][k] = 0.0;
     }
   }
}
```

# Using Structs/Types

## FORTRAN

```
integer (kind=4) :: i,j,k
type (pmclass) :: pmc(Nclasses)

forall (i=1:Nclasses)
   pmc(i)%num = 0
   pmc(i)%det = 0.0
   pmc(i)%exam = 0.0
end forall
```