HPC Doctoral Taught Centre: Autumn Academy

Preliminary Exercises by J.H. Davenport — `J.H.Davenport@bath.ac.uk`

17 July 2012

1 Matrices

Much of High-Performance Computing deals with *regular* mathematical structures, of which the most obvious kinds are vectors and matrices. Unfortunately there are not in everyone's mathematical background:

- the simplest (and cheapest) reference text that we have found that covers this seems to be: A Level Mathematics for Edexcel: Further Pure FP1 (ISBN 9780435519230);
- the Wikipedia page on `Matrix (mathematics)` and its linked pages are not bad (except that it uses [. . .] where I am using (. . .): I hope this doesn't confuse, but both notations are in use).

2 Exercises

All of you should be familiar with programming in *some* language. These exercises are to be carried out in whatever language you feel most comfortable with: we will show C/Fortran equivalents at the Academy itself. For those of you whose programming language of choice is MatLab, please use MatLab `for` or `while` statements, rather than the built-in MatLab features, and the same applies to other languages with built-in matrix manipulation.

I say "Write a function to" — the precise method will depend on your langauge: it might be a method, function, procedure or subroutine.

1. Write a function to add two $m \times n$ matrices together, i.e. input the matrices of $a_{i,j}$ and $b_{i,j}$, and output the matrix of $c_{i,j}$:

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{pmatrix} + \begin{pmatrix} b_{1,1} & \dots & b_{1,n} \\ \vdots & \ddots & \vdots \\ b_{m,1} & \dots & b_{m,n} \end{pmatrix} = \begin{pmatrix} c_{1,1} & \dots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{m,1} & \dots & c_{m,n} \end{pmatrix},$$

where $c_{i,j} = a_{i,j} + b_{i,j}$.

2. Write a function to multiply an $m \times n$ matrix by an n-vector, i.e. input the matrix of $a_{i,j}$ and the vector of c_j and output the vector of v_j:

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{pmatrix} \times \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_m \end{pmatrix},$$

where $d_i = \sum_{j=1}^{n} a_{i,j} c_j$.

3. Write a function to multiply an $m \times n$ matrix by an $n \times p$ matrix, i.e. input the matrices of $a_{i,j}$ and $b_{i,j}$, and output the matrix of $c_{i,j}$:

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{pmatrix} \times \begin{pmatrix} b_{1,1} & \dots & b_{1,p} \\ \vdots & \ddots & \vdots \\ b_{n,1} & \dots & b_{n,p} \end{pmatrix} = \begin{pmatrix} c_{1,1} & \dots & c_{1,p} \\ \vdots & \ddots & \vdots \\ c_{m,1} & \dots & c_{m,p} \end{pmatrix},$$

where $c_{i,k} = \sum_{j=1}^{n} a_{i,j} b_{j,k}$.

Note that, depending on the system you are using, you may or may not find it easier to re-use exercise 2 here.

4, **harder** Write a function to solve a set of linear equations, i.e. input the matrix of $a_{i,j}$ and the vector of c_j and output the vector of x_j:

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{pmatrix} \times \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix},$$

where $d_i = \sum_{j=1}^{n} a_{i,j} c_j$. This is a process known as *Gaussian elimination*, and actually has many subtleties when translated into a numerical process, some of which will be discussed during the Academy.