CA205 Assignment 3: Review Notes

Greg Gamble

email: gregg@maths.uwa.edu.au

WWW: http://maths.uwa.edu.au/~gregg/CA205/

November 6, 1997

1 General Comments

The structure of reports continued to improve. The theme of this assignment was somewhat different to the previous two: rather than write programs, one had to determine how given programs worked, modify them for new (but similar) purposes and compare programs written for the same purpose for efficiency.

2 Marking

The marking guide did not specify marks for Part 2. However, later parts of the assignment required that the program be modified. It was important for the student to explain the modifications required. Thus, in particular, 4 marks of the 9 possible for Part 5, were allocated for understanding the program.

3 Understanding the program

To come to grips with the program one needed to understand the mathematics. Firstly, we have the following d.e.s:

$$\begin{aligned} z'(x) &= w(x) \\ w'(x) &= \frac{v(x)}{EI(x)} \\ v'(x) &= u(x) \\ u'(x) &= q(x), \end{aligned}$$

where $q(x) =$ applied load, $u(x) =$ shear force, $v(x) =$ bending moment, $w(x) =$ slope, $z(x) =$ deflection.

Essentially, this is a fourth order system modelled as four first order d.e.s; and rewriting using matrices we obtain

$$\underset{\sim}{\boldsymbol{y}}' = \boldsymbol{B}\underset{\sim}{\boldsymbol{y}} + \underset{\sim}{\boldsymbol{c}}, \tag{3.1}$$

where

$$\underset{\sim}{\boldsymbol{y}} = \begin{pmatrix} z(x) \\ w(x) \\ v(x) \\ u(x) \end{pmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1/EI(x) & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \underset{\sim}{\boldsymbol{c}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ q(x) \end{pmatrix}.$$

Modelling (3.1) by finite differences, writing h for the step-change in x, we obtain at the k^{th} step:

$$\frac{\underset{\sim}{\boldsymbol{y}}_k - \underset{\sim}{\boldsymbol{y}}_{k-1}}{h} \approx \boldsymbol{B}_{k-\frac{1}{2}} \underset{\sim}{\boldsymbol{y}}_{k-\frac{1}{2}} + \underset{\sim}{\boldsymbol{c}}_{k-\frac{1}{2}}, \tag{3.2}$$

where the meaning of '$k - \frac{1}{2}$' is evaluate any expression $\phi(x)$ by taking the average of the values at x_k and x_{k-1}. In particular,

$$\underset{\sim}{\boldsymbol{y}}_{k-\frac{1}{2}} = \tfrac{1}{2}(\underset{\sim}{\boldsymbol{y}}_k + \underset{\sim}{\boldsymbol{y}}_{k-1}).$$

Substituting the latter equation in (3.2) and rearranging we get:

$$(-\boldsymbol{I} - \tfrac{h}{2}\boldsymbol{B}_{k-\frac{1}{2}})\underset{\sim}{\boldsymbol{y}}_{k-1} + (\boldsymbol{I} - \tfrac{h}{2}\boldsymbol{B}_{k-\frac{1}{2}})\underset{\sim}{\boldsymbol{y}}_k = h\underset{\sim}{\boldsymbol{c}}_{k-\frac{1}{2}},$$

where $\boldsymbol{I}$ is the 4×4 identity matrix; which can be rewritten in matrix block-form as so:

$$\left[-\boldsymbol{I} - \tfrac{h}{2}\boldsymbol{B}_{k-\frac{1}{2}} \;\middle|\; \boldsymbol{I} - \tfrac{h}{2}\boldsymbol{B}_{k-\frac{1}{2}} \right] \begin{pmatrix} \underset{\sim}{\boldsymbol{y}}_{k-1} \\ \underset{\sim}{\boldsymbol{y}}_k \end{pmatrix} = h\underset{\sim}{\boldsymbol{c}}_{k-\frac{1}{2}}.$$

For a pin-jointed beam, the boundary conditions at $x = 0$, are $z_1 = 0, v_1 = 0$ which in matrix form is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underset{\sim}{\boldsymbol{y}}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Similarly, the boundary conditions at $x = L$, are $z_{M+1} = 0, v_{M+1} = 0$ which in matrix form is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underset{\sim}{\boldsymbol{y}}_{M+1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

When all of these equations in the $\underset{\sim}{\boldsymbol{y}}_k$ are put together in one block matrix one obtains $\boldsymbol{A}\underset{\sim}{\boldsymbol{Y}} = \underset{\sim}{\boldsymbol{R}}$, where

$$\boldsymbol{A} = \left[\begin{array}{c|c|c|c|c} \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} & & & & \\ \hline -\boldsymbol{I} - \frac{h}{2}\boldsymbol{B}_{\frac{1}{2}} & \boldsymbol{I} - \frac{h}{2}\boldsymbol{B}_{\frac{1}{2}} & & & \\ \hline & -\boldsymbol{I} - \frac{h}{2}\boldsymbol{B}_{\frac{3}{2}} & \boldsymbol{I} - \frac{h}{2}\boldsymbol{B}_{\frac{3}{2}} & & \\ & & \ddots & & \\ & & & -\boldsymbol{I} - \frac{h}{2}\boldsymbol{B}_{M+\frac{1}{2}} & \boldsymbol{I} - \frac{h}{2}\boldsymbol{B}_{M+\frac{1}{2}} \\ \hline & & & & \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} \end{array}\right],$$

$$\underset{\sim}{\boldsymbol{Y}} = \begin{pmatrix} \underset{\sim}{\boldsymbol{y}}_1 \\ \underset{\sim}{\boldsymbol{y}}_2 \\ \vdots \\ \underset{\sim}{\boldsymbol{y}}_M \\ \underset{\sim}{\boldsymbol{y}}_{M+1} \end{pmatrix}, \quad \underset{\sim}{\boldsymbol{R}} = \begin{pmatrix} \underset{\sim}{\boldsymbol{0}} \\ h\underset{\sim}{\boldsymbol{c}}_{\frac{1}{2}} \\ h\underset{\sim}{\boldsymbol{c}}_{\frac{3}{2}} \\ \vdots \\ h\underset{\sim}{\boldsymbol{c}}_{M+\frac{1}{2}} \\ \underset{\sim}{\boldsymbol{0}} \end{pmatrix}, \quad \underset{\sim}{\boldsymbol{0}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The bulk of the programs `beam.m` and `beamsp.m` is spent building the matrix $\boldsymbol{A}$. Notice how close to Row-Echelon form the matrix $\boldsymbol{A}$ is, so it is not so surprising that using sparse matrices and MATLAB's special matrix-solve syntax to determine the vector $\underset{\sim}{\boldsymbol{Y}}$ (as in the program `beamsp.m`) makes the program both much faster and much more efficient.

Note it is a good idea to change the programs `beam.m` and `beamsp.m` so that they become functions, e.g. for `beamsp.m` one could add (at the beginning) the line

```
function [] = beamsp(M)
```

and remove the line

```
M=20;
```

Then one can use the MATLAB functions `tic` and `toc` to time the programs for various values of M. In order to compare `beam.m` and `beamsp.m` one should really also comment out the lines

```
figure(1);
clf;
spy(A);
```

After doing this on a Silicon Graphics Indy running MATLAB 4.2(c), the following was obtained:

```
>> tic; beam(100); toc;
elapsed_time =
   11.8197
>> tic; beamsp(100); toc;
elapsed_time =
    1.4687
>> tic; beam(200); toc;
elapsed_time =
   90.6801
>> tic; beamsp(200); toc;
elapsed_time =
    4.0851
```

By considerations such as these one can get some empirical evidence of how the programs depend on M … do the algorithms have complexity $\mathcal{O}(M)$ or $\mathcal{O}(M^2)$ or something else?

In modifying the program for fixed ends it was important to explain that the boundary conditions at $x = 0$ and $x = L$ become $z_1 = 0, w_1 = 0$ and $z_{M+1} = 0, w_{M+1} = 0$, respectively, which in matrix form is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \underset{\sim}{\boldsymbol{y}}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

and

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \underset{\sim}{\boldsymbol{y}}_{M+1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

respectively, with consequent changes to the first and last blocks of the matrix $\boldsymbol{A}$.

Similarly, for the cantilever beam, at the free end, $x = 0$, the boundary conditions are $v_1 = 0, u_1 = 0$, i.e.

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underset{\sim}{\boldsymbol{y}}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

and at the fixed end, $x = L$, as above, the boundary conditions are: $z_{M+1} = 0, w_{M+1} = 0$, i.e.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \underset{\sim}{\boldsymbol{y}}_{M+1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$