Assignment 3 — 2.086 Fall 2015

Due: *Tuesday, 13 October, at 5 PM.*

Upload your solution to Stellar as a zip file "`YOURNAME_ASSIGNMENT_3.zip`" which includes the script for each question in the proper format *as well as* any MATLAB functions (of your own creation) your scripts may call. Remember to verify your upload before the assignment due time by downloading your solution folder yourself and re-running `grade_o_matic`. Full instructions for assignment preparation are given in the document "Preparing assignments for `grade_o_matic`" available on the 2.086 website.
In this assignment, the student mode of `grade_o_matic` will not provide partial credit. You may still use it to search for syntax errors (which you can also do by running your script manually). For problem 2, it is very important to insert your answers in the **precise** format given in the template. For problems 1 and 5, please submit your answer in pdf format; you may type your answer in your favorite software, but scans of handwritten material will not be accepted. Problems 1 and 5 are not graded by `grade_o_matic` because we want to see detailed explanations of how your arrived at your answer. Having the right answer at the end without a clear (and correct) explanation will only yield partial credit.

Questions

1. (20 points). The figure shows a double oscillator made of 2 Wilberforce springs. Wilberforce springs are special flexural elements that couple extension and rotation and as a result, tend to twist when extended/contracted and tend to extend/contract when twisted. More specifically, the force required to hold a Wilberforce spring at extension x and rotation (with respect to a reference angle) θ is given by

$$f = kx + \epsilon\theta$$

while at the same time, the torque required to hold the spring in this state is

$$\tau = \delta\theta + \epsilon x$$

The equilibrium equations for the system shown in the figure can be written as follows

$$\begin{aligned} 2kx_1 - kx_2 + 2\epsilon\theta_1 - \epsilon\theta_2 &= 0 \\ -kx_1 + kx_2 - \epsilon\theta_1 + \epsilon\theta_2 &= F \\ 2\epsilon x_1 - \epsilon x_2 + 2\delta\theta_1 - \delta\theta_2 &= 0 \\ -\epsilon x_1 + \epsilon x_2 - \delta\theta_1 + \delta\theta_2 &= T \end{aligned} \tag{1}$$

where x_1, x_2 denote the locations of masses 1 and 2 in the vertical direction, respectively, θ_1 and θ_2 denote the rotation of masses 1 and 2 from a fixed reference direction, F is the force exerted on mass 2 and T is the torque exerted on mass 2, as shown in the figure. All coordinates are measured from the equilibrium position defined by $F = 0$ and $T = 0$.

The system of equations (1) is linear and can be written in the form

$$Au = b, \tag{2}$$

where $u = (x_1, x_2, \theta_1, \theta_2)^{\mathrm{T}}$, $b = (0, F, 0, T)^{\mathrm{T}}$ and A is a matrix.

Please answer the questions below

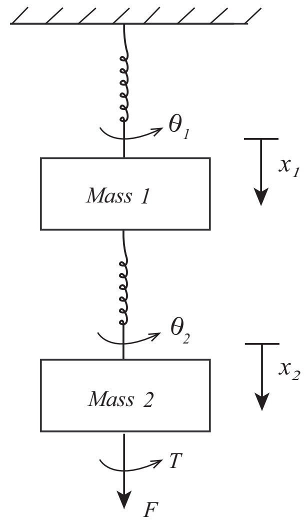

Figure 1: A double oscillator made of 2 Wilberforce springs that is the subject of Question 1.

(i) (10 points) What are the entries of matrix A?

(ii) (10 points) Through experimentation, we find that when $F = 1N$ and $T = 0Nm$, $x_1 = 1mm$; when $T = 1Nm$ and $F = 0N$, $x_1 = 3mm$. Find x_1 when $F = 2N$ **and** $T = 1Nm$.

There is no Matlab script for this question. Submit your answers to this question in a pdf file named A3Q1.pdf; place this file in the zip file containing your matlab scripts and functions as outlined above.

2. (20 points) You are given a matrix A of size 2×3,

$$A = \begin{pmatrix} a & -b & c \\ d & e & f \end{pmatrix} ,$$

in MATLAB `A = [a, -b, c; d, e, f]`, and a second matrix B of size 3×2,

$$B = \begin{pmatrix} g & h \\ i & -j \\ k & l \end{pmatrix} ,$$

in MATLAB `B = [g, h; i, -j; k, l]`. Here the size of a matrix, $m \times n$, refers to the number of rows (m) and number of columns (n) in the matrix. For the questions i-iv below:

(i) (5 points) The product $C = AB$ (in MATLAB $C = A * B$)

(ii) (5 points) The product $C = BA$ (in MATLAB $C = B * A$)

(iii) (5 points) The product $C = B^{\mathrm{T}} A^{\mathrm{T}}$.

(iv) (5 points) The sum $C = A + B$

select the right answer from the options (a) - (k):

(a) $C = \begin{pmatrix} ag+dh & -bg+eh & cg+fh \\ ai-dj & -bi-ej & ci-fj \\ ak+dl & -bk+el & ck+fl \end{pmatrix}$

(b) $C = \begin{pmatrix} ag+dh & ai-dj & ak+dl \\ -bg+eh & -bi-ej & -bk+el \\ cg+fh & ci-fj & ck+fl \end{pmatrix}$

(c) $C = \begin{pmatrix} ag & -bh \\ di & -cj \end{pmatrix}$

(d) $C = \begin{pmatrix} ag & di \\ -bh & -cj \end{pmatrix}$

(e) $C = \begin{pmatrix} ag-bi+ck & ah+bj+cl \\ dg+ei+fk & dh-ej+fl \end{pmatrix}$

(f) $C = \begin{pmatrix} ag-bi+ck & dg+ei+fk \\ ah+bj+cl & dh-ej+fl \end{pmatrix}$

(g) $C = \begin{pmatrix} a+g & -b+i & c+k \\ d+h & e-j & f+l \end{pmatrix}$

(h) $C = \begin{pmatrix} a+g & d+h \\ -b+i & e-j \\ c+k & f+l \end{pmatrix}$

(k) can not be performed

where "can not be performed" means that the operation is not allowed by the rules of matrix algebra.

The template `A3Q2_Template.m` contains the multiple-choice format required by `grade_o_matic`.

3. (*20 points*) The following script will likely crash your computer due to memory limitations; please do *not* run it. Instead, work out the questions that follow by hand.

```
% begin script

clear
% note we "clear" the workspace

A = zeros(1000000,1000000);
for i = 1:1000000
    A(i,1) = 1.0;
end
```

```
A(426,12) = 4.0;
A(12,426) = 3.0;
A(426,426) = -1.0;
A(999999,1000000) = 5.0;

w = 3*ones(1000000,1); % note w is a column vector of all three's

v = A*w;

M = max(v);

% end script
```

where we recall that `max` is the MATLAB built-in function which returns the maximum of a vector. Please perform the described operations mentally and assign the results to the output variables indicated:

- (*i*) (5 points) Assign what would be the value of `v(12)` to the output variable `v_12_out` *Hint*: Consider the "row interpretation" of the matrix-vector product.
- (*ii*) (5 points) Assign what would be the value of `v(426)` to the output variable `v_426_out`
- (*iii*) (5 points) Assign what would be the value of `v(1000000)` to the output variable `v_1000000_out`
- (*iv*) (5 points) Assign what would be the value of `M` to the output variable `M_out`

Hint: One could modify the above script so that it will run and use it to verify their results.

The template `A3Q3_Template.m` contains the output variables required by `grade_o_matic`. There are no input variables.

4. (20 points–each question carries 2 points except from question (*v*) which carries zero points) Write a script which performs the following operations (in sequence):

- (*i*) creates a new row vector `x` with 21 elements in ascending order from -1 to 1 by steps of 0.1. Vector `x` is an output of your script, so make sure to not modify it below.
- (*ii*) creates a new row vector `y` with 41 elements in ascending order from -0.5 to 1.5 by steps of 0.05. Vector `y` is an output of your script, so make sure to not modify it below.
- (*iii*) creates two new arrays `X` and `Y` of size 41×21 where each row of `X` is a copy of the vector `x` and each column of `Y` has the same elements as the vector `y`. Arrays `X` and `Y` are outputs of your script, so make sure to not modify them below.
- (*iv*) create a new array `A` that is the sum of the square of the corresponding entries of arrays `X` and `Y`—specifically, `A(i,j)=X(i,j)*X(i,j)+Y(i,j)*Y(i,j)`. Array `A` is an output of your script, so make sure to not modify it below.
- (*v*) creates a new array `F` of size 41×21 with elements `F(i,j)=1/A(i,j)`.
- (*vi*) creates a new array `F1` = `F`, then finds the mean of the elements of `F1` that are not infinity and assigns that value to the elements where `isinf(F1)==1`. Array `F1` is an output of your script, so make sure to not modify it below.
- (*vii*) creates a new array `F2` of size 41×42 that contains two copies of `F1` side by side (i.e. with elements `F2(i,j)=F1(i,j)` and `F2(i,j+21)=F1(i,j)` for i∈ $\{1, 2, \ldots, 41\}$ and

j$\in \{1, 2, \ldots, 21\}$). Array `F2` is an output of your script, so make sure to not modify it below.

(*viii*) creates a new array `F3` = `F2` and adds a new 42nd row to the end of `F3` to make it an array of size 42×42; every element in that row should be set to 100. Array `F3` is an output of your script, so make sure to not modify it below.

(*ix*) creates an array `F4` = `F3` and assigns all the elements in the first row *and* first column *and* last column of `F4` to 100. Array `F4` is an output of your script, so make sure to not modify it below.

(*x*) creates an array `F5` = `F4` and assigns a value of 200 to the following elements `F5(21,21)`, `F5(29,12)`, `F5(29,31)`, and `F5(30,k)` for k$\in \{13, 14, \ldots, 30\}$. Array `F5` is an output of your script, so make sure to not modify it below.

(*xi*) creates a scalar `bigsum` which is the sum of all the elements (1764 in total) of the array `F5`. Scalar `bigsum` is an output of your script, so make sure to not modify it below.

Bonus: (no points) Arrays can naturally be used to represent images. Try the command `imagesc(F5)` to see a visual representation of the array you created. **Please remove this command from your final submission.**

The script takes no input. The template is provided in `A3Q4_Template`.

5. (20 points). We consider the system $Au = f$ given by

(*i*) (10 points)

$$\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}}_{u} = \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}}_{f}.$$

For $f = (1 \quad 1 \quad 1)^{\mathrm{T}}$, find the solution vector $u = (u_1 \quad u_2)^{\mathrm{T}}$. If no solution exists, explain why. If an infinity of solutions exists, describe them in the most general form.

(*ii*) (10 points)

$$\underbrace{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 2 & 2 & 4 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}}_{u} = \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}}_{f}.$$

For $f = (1 \quad 1 \quad 4)^{\mathrm{T}}$, find the solution vector $u = (u_1 \quad u_2 \quad u_3)^{\mathrm{T}}$. If no solution exists, explain why. If an infinity of solutions exists, describe them in the most general form.

There is no Matlab script for this question. Submit your answers to this question in a pdf file named A3Q5.pdf; place this file in the zip file containing your matlab scripts and functions as outlined above.