Math 307: Problems for section 1.2

Many problems in this homework make use of a few MATLAB/Octave `.m` files that are provided on the website. In order to use them, make sure that the files are in the same directory that you are running MATLAB/Octave from (to see which directory this is, type `pwd` in MATLAB/Octave).

1. **Compute the determinant of a 4×4 Vandermonde matrix. Bonus: show that the general formula for the determinant of a Vandermonde matrix is correct.**

   Here is the general calculation, although you only need to hand in the 4×4 version of this. The main point is that you can write the determinant for the $n \times n$ matrix as an expression involving the determinant of the $(n-1) \times (n-1)$ matrix. If you start with the 4×4 case, you can then substitute the expression we already computed for the 3×3.

   The general case goes by induction. We know the formula is true for $n = 1$ (also $n = 2, 3$). The inductive step is to show that the formula for n follows from the formula for $n-1$. Let $d(n; x_1, x_2, \ldots, x_n)$ be the determinant of the $n \times n$ Vandermonde matrix with variables $x_1, x_2, \ldots, x_n$. We begin with the same steps as the 3×3 example done in lectures. First we subtract x_n times the second column from the first column, then x_n times the third column from the second column, and so on. This doesn't change the determinant

$$
\begin{aligned}
d(n; x_1, x_2, \ldots, x_n) &= \det \left(\begin{bmatrix} x_1^{n-1} & x_1^{n-2} & \cdots & x_1^2 & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \cdots & x_2^2 & x_2 & 1 \\ x_3^{n-1} & x_3^{n-2} & \cdots & x_3^2 & x_3 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_n^{n-1} & x_n^{n-2} & \cdots & x_n^2 & x_n & 1 \end{bmatrix} \right) \\
&= \det \left(\begin{bmatrix} x_1^{n-1} - x_1^{n-2}x_n & x_1^{n-2} - x_1^{n-3}x_n & \cdots & x_1^2 - x_1x_n & x_1 - x_n & 1 \\ x_2^{n-1} - x_2^{n-2}x_n & x_2^{n-2} - x_2^{n-3}x_n & \cdots & x_2^2 - x_2x_n & x_2 - x_n & 1 \\ x_3^{n-1} - x_3^{n-2}x_n & x_3^{n-2} - x_3^{n-3}x_n & \cdots & x_3^2 - x_3x_n & x_3 - x_n & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_n^{n-1} - x_n^{n-2}x_n & x_n^{n-2} - x_n^{n-3}x_n & \cdots & x_n^2 - x_nx_n & x_n - x_n & 1 \end{bmatrix} \right) \\
&= \det \left(\begin{bmatrix} (x_1 - x_n)x_1^{n-2} & (x_1 - x_n)x_1^{n-3} & \cdots & (x_1 - x_n)x_1 & x_1 - x_n & 1 \\ (x_2 - x_n)x_2^{n-2} & (x_2 - x_n)x_2^{n-3} & \cdots & (x_2 - x_n)x_2 & x_2 - x_n & 1 \\ (x_3 - x_n)x_3^{n-2} & (x_3 - x_n)x_3^{n-3} & \cdots & (x_3 - x_n)x_3 & x_3 - x_n & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \right)
\end{aligned}
$$

   Now expand along the bottom row and remove the common factors.

$$
\begin{aligned}
d(n; x_1, x_2, \ldots, x_n) &= (x_1 - x_n)(x_2 - x_n) \cdots (x_{n-1} - x_n) d(n-1, x_1, x_2, \ldots, x_{n-1}) \\
&= (-1)^{n-1}(x_n - x_1)(x_n - x_2) \cdots (x_n - x_{n-1}) d(n-1, x_1, x_2, \ldots, x_{n-1})
\end{aligned}
$$

   We may assume that the result for $n-1$ is known. That is

$$
d(n-1, x_1, x_2, \ldots, x_{n-1}) = (-1)^{(n-1)(n-2)/2} \prod_{n \geq i > j \geq 2} (x_i - x_j)
$$

Thus we conclude

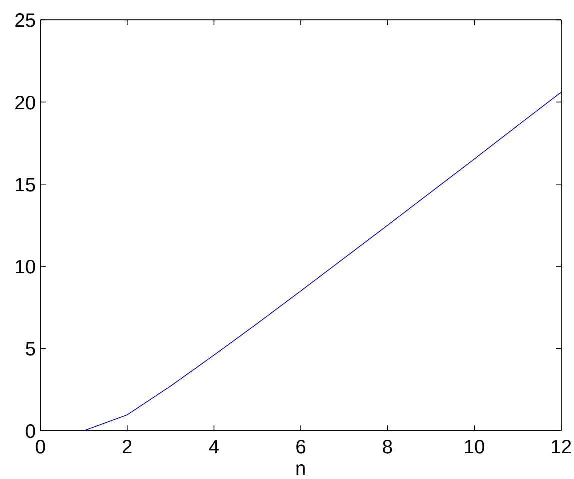
$$
\begin{aligned}
d(n; x_1, x_2, \ldots, x_n) &= (-1)^{n-1}(x_n - x_1)(x_n - x_2)\cdots(x_n - x_{n-1})(-1)^{(n-1)(n-2)/2} \prod_{n \geq i > j \geq 2} (x_i - x_j) \\
&= (-1)^{(n-1)+(n-1)(n-2)/2} \prod_{n \geq i > j \geq 1} (x_i - x_j)
\end{aligned}
$$

and the result follows from the calculation $(n-1) + (n-1)(n-2)/2 = (2n - 2 + (n-1)(n-2))/2 = (2n - 2 + n^2 - 3n + 2)/2 = n(n-1)/2$.

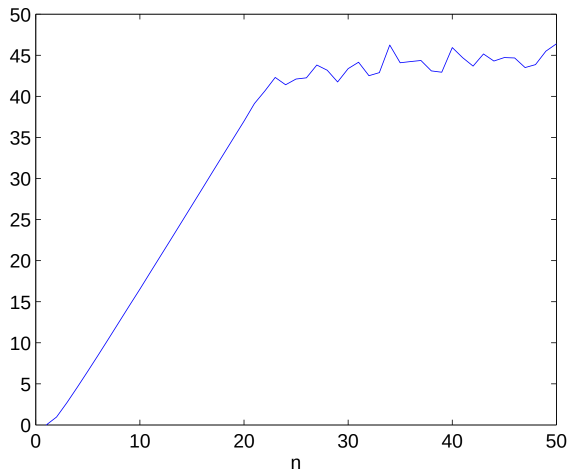
2. **Let V_n be the Vandermonde matrix for n equally spaced points between 0 and 1. Do you think the condition number of V_n is increasing exponentially in n? To make an informed guess, use MATLAB/Octave to make a plot of $\log(\mathrm{cond}(V_n))$ against n. You will need to use relatively small values of n (say $n < 20$ or so) to get a reasonable looking plot. What do you think is happening when you use larger values of n?**

   The point behind this plot is that if $\mathrm{cond}(V_n)$ is exponentially increasing, that is $\mathrm{cond}(V_n) \sim Ce^{\alpha n}$ then taking logs we get $\log(\mathrm{cond}(V_n)) \sim \log(C) + \alpha n$, that is, a linear function of n with slope α. So if we plot $\log(\mathrm{cond}(V_n))$ and it looks linear, then this is evidence of exponential growth for $\mathrm{cond}(V_n)$.

   The quantity $\ln(\mathrm{cond}(V_n))$ can be computed as `log(cond(vander(linspace(1,0,n))))`. If you compute this for a collection of n's and plot the resulting points against n you get a graph that looks like .... well it depends on what n's you choose. My first try (for relatively small values of n) looks nicely linear.



   However if n gets large the graph starts to look rough:



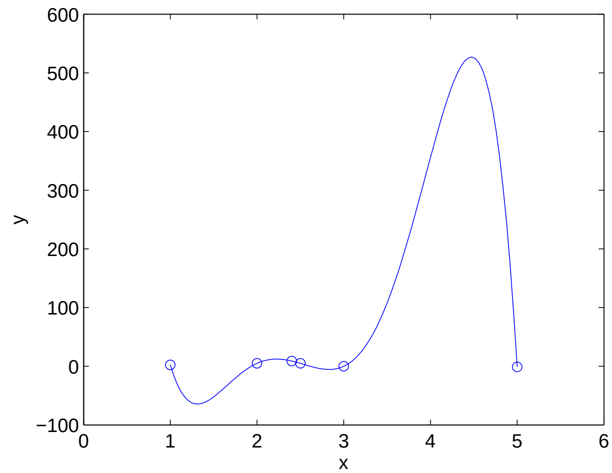
This is almost certainly due to truncation error when the computer is dealing with extremely large numbers. So I would still count this as evidence that the condition number is increasing exponentially, based on the first part of the graph (although maybe not completely convincing!).

3. **Use MATLAB/Octave to plot the Lagrange interpolating function through the points** $(1, 2.3)$**,** $(2, 5)$**,** $(2.4, 9)$**,** $(2.5, 5)$**,** $(3, 0)$ **and** $(5, -1)$**.**

   The MATLAB/Octave commands to plot this are:

```
> X = [1 2 2.4 2.5 3 5]
X =

   1.0000   2.0000   2.4000   2.5000   3.0000   5.0000

> Y = [2.3 5 9 5 0 -1]
Y =

   2.30000   5.00000   9.00000   5.00000   0.00000  -1.00000

> V = vander(X);
> a = V\Y';
> XL = linspace(1,5,100);
> YL = polyval(a,XL);
> plot(X,Y,'bo')
> hold on
> plot(XL,YL,'b-')
> axis([0 6 -100 600])
```

   The resultant plots looks like this:



The purpose of this problem was to show that even with only a few points, if those points are badly distributed the Lagrange interpolating polynomial can give a terrible fit.

4. **Derive the matrix equation to solve in order to find the cubic spline passing through the three points $(0,1)$, $(0.5,2)$ and $(1,4)$. Plot the resulting spline (you may use the file `plotspline.m`).**

   Take the cubic polynomials in the two intervals to be

$$p_1(x) = a_1x^3 + b_1x^2 + c_1x + 1 \qquad 0 \le x \le 1/2$$
$$p_2(x) = a_2(x-1/2)^3 + b_2(x-1/2)^2 + c_2(x-1/2) + 2 \qquad 1/2 \le x \le 1$$

At $x = x_2 = 1/2$ we impose $p_1 = p_2 = 2$ (the latter of which is already satisfied), $p_1' = p_2'$ and $p_1'' = p_2''$. This gives the three equations

$$a_1(1/2)^3 + b_1(1/2)^2 + c_1(1/2) + 1 = 2, \qquad ((1))$$
$$3a_1(1/2)^2 + 2b_1(1/2) + c_1 = c_2, \qquad ((2))$$
$$6a_1(1/2) + 2b_1 = 2b_2. \qquad ((3))$$

At $x = x_1 = 0$ impose $p_1 = 1$ (already satisfied), $p_1'' = 0$. This gives

$$2b_1 = 0. \qquad ((5))$$

At $x = x_3 = 1$ impose $p_2 = 4$, $p_2'' = 0$. This gives

$$a_2(1/2)^3 + b_2(1/2)^2 + c_2(1/2) + 2 = 4, \qquad ((4))$$
$$6a_2(1/2) + 2b_2 = 0. \qquad ((6))$$

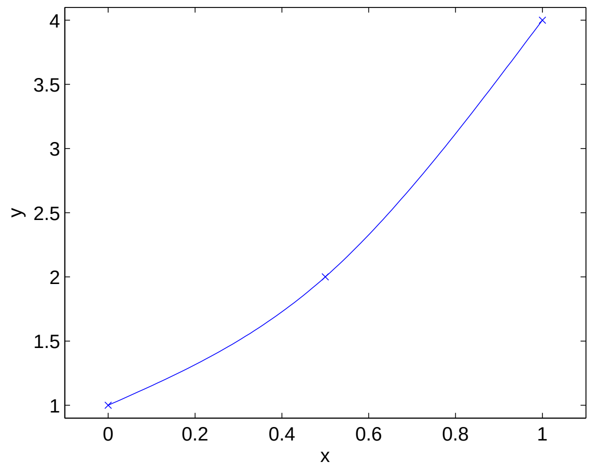
Multiplying equation (2) by 1/2 and equations (3), (5) and (6) by $(1/2)^2$. Writing this in matrix form find

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & -1 \\ 6 & 2 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 2 & 0 \end{bmatrix} \begin{bmatrix} a_1(1/2)^3 \\ b_1(1/2)^2 \\ c_1(1/2) \\ a_2(1/2)^3 \\ b_2(1/2)^2 \\ c_2(1/2) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}.$$

We use `plotspline.m` to create the plot

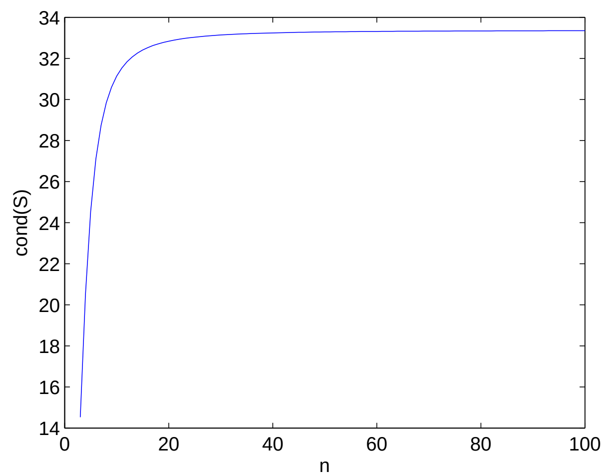
```
> X = [0 1/2 1];
> Y = [1 2 4];
> plotspline(X,Y)
> axis([-0.1 1.1 0.9 4.1])
```

producing the plot



5. **What happens to the condition number of the matrix S used in cubic spline interpolation as the size n becomes large (you may use the file `splinemat.m`)?**

   In contrast to the Vandermonde matrix, the condition number of the spline matrix approaches a constant. Here is a plot



6. **A parabolic runout spline is the interpolating function you get by changing the condition $f''(x_1) = f''(x_n) = 0$ to the condition that $p_1(x)$ and $p_{n-1}(x)$ should be quadratic polynomials (that is, $a_1 = a_{n-1} = 0$). Modify the file `splinemat.m` so that it computes the matrix relevant to this modified problem. Call the modified file `splinematpr.m`. (Hand in a description of your changes, or a print-out of the modified file.) Use your new file to graph the parabolic**

runout spline for the points $(1, 1)$, $(2, 1)$, $(3, 2)$, $(4, 4)$ and $(5, 3)$. (The easiest way to do this is to change `splinemat` to `splinematpr` inside the file `plotspline.m` and call the modified file `plotsplinepr.m`. Use this new file to plot the modified spline.) Hand in a plot of both the parabolic runout spline and the cubic spline on the same graph.

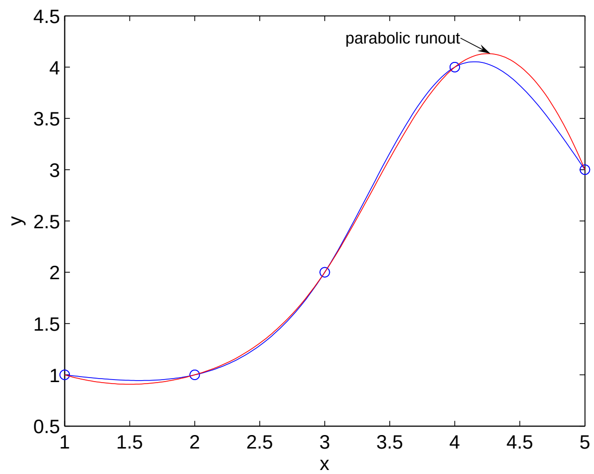
In `splinemat.m` change the lines

```
T=[0 0 0;0 2 0; 0 0 0];
V=[1 1 1;0 0 0;6 2 0];
```

to

```
T=[0 0 0;1 0 0; 0 0 0];
V=[1 1 1;0 0 0;1 0 0];
```

Here are the two types of spline on the same graph.



7. **Consider the problem of interpolating four points (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4) with a function $f(x)$ that is given by a quadratic polynomial in each interval x_i, x_{i+1}, (i.e., $p_i(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i$) and whose first derivative $f'(x)$ is continuous across the points x_i. Write down the system of equations for this problem. Is there a unique solution to this problem?**

   The equations are

$$p_1(x_1) = y_1, \quad p_2(x_2) = y_2, \quad p_3(x_3) = y_3$$
$$p_1(x_2) = y_2, \quad p_2(x_3) = y_3, \quad p_3(x_4) = y_4$$
$$p_1'(x_2) = p_2'(x_2)$$
$$p_2'(x_3) = p_3'(x_3)$$

These can be written

$$c_1 = y_1 \tag{1}$$
$$c_2 = y_2 \tag{2}$$
$$c_3 = y_3 \tag{3}$$
$$a_1(x_2 - x_1)^2 + b_1(x_2 - x_1) + c_1 = y_2 \tag{4}$$
$$a_2(x_3 - x_2)^2 + b_2(x_3 - x_2) + c_2 = y_3 \tag{5}$$
$$a_3(x_4 - x_3)^2 + b_3(x_4 - x_3) + c_3 = y_4 \tag{6}$$
$$2a_1(x_2 - x_1) + b_1 = b_2 \tag{7}$$
$$2a_2(x_3 - x_2) + b_2 = b_3 \tag{8}$$

This is system of 8 equations in 9 unknowns and therefore does not have a unique solution.