Homework Assignment #6

Math Methods

Reading Quiz Due: Friday, October 15th, 11:00am

Homework Due: Tuesday, October 19th, midnight

Instructions:

Reading: Please read Chapter 3. The reading quiz will be due Wednesday before class. Much of the chapter deals with concepts in matrices that I believe you already understand. If this is not the case, be sure to let me know by posting to the discussion board and also providing a detailed answer to the quiz question about what material I should review.

Problems: Below is a list of questions and problems from the texbook due by the time and date above. It is not sufficient to simply obtain the correct answer. You must also explain your calculation, and each step so that it is clear that you understand the material.

Homework should be written legibly, on standard size paper. Do not write your homework up on scrap paper. If your work is illegible, it will be given a zero.

---

1. Submit your answer to problem (4) from the previous homework set. (This is the long homework problem dealing with $F_{\mu\nu}$.)

2. Consider the function
$$f(x) = 4x^3 - 32x^2 + 66x - 18$$
   (a) Solve $f(x) = 0$ for x, obtaining both symbolic and numeric answers.
   (b) Solve $f'(x) = 0$ for x, obtaining both symbolic and numeric answers.
   (c) Plot the function and verify that your roots and extrema are correct.

3. Plot the Taylor series expansion of the sine function over the interval $0 < x < 2\pi$ and determine how many terms you need in order to get reasonable accuracy. (*Hint:* You can use the **Table** command to generate a list of the terms and the **Total** command to sum up the list.)

4. Consider the 10×10 matrix $\mathcal{A}_{ij} = \sin(ij)$ and vector 10×1 column vector $b_i = i$. Solve the numerical problem
$$\mathcal{A}\,\vec{x} = \vec{b}$$
   for the unknown vector $\vec{x}$ using the command **LinearSolve**. (*Hint:* If this takes a while to calculate on your computer, then you have gotten caught in one of the traps I warned you about in lecture.)

5. **Euler method breakdown:** Consider a simple harmonic oscillator governed by
$$\ddot{x} = -x$$
   with $x(0) = 1$, and $\dot{x}(0) = 0$. (We have set the spring constant and the mass to one). Define $v \equiv \dot{x}$.
   (a) Consider the curve $(x(t), v(t))$. What should it look like for the exact solution?

(b) Solve this with a standard Euler approach:

$$\begin{aligned} v_{i+1} &= v_i - x_i \Delta t \\ x_{i+1} &= x_i + v_i \Delta t \end{aligned}$$

(Convince yourself that this is a reasonable discretization scheme and is consistent with the equation of motion.) Plot the orbit (x_i, v_i) for $0 \leq t \leq 8\pi$ for some reasonable choice of step size Δt. What type of orbit do you get?

(c) Alter the algorithm so that you instead use:

$$\begin{aligned} v_{i+1} &= v_i - x_i \Delta t \\ x_{i+1} &= x_i + v_{i+1} \Delta t \end{aligned}$$

Thus the new position depends upon the *new* velocity. Plot the orbits again. Does this give a better result? Note that this is partially an *implicit scheme*, in that we use some information at time $i+1$ to calculate the solution at time $i+1$.

(d) The energy E_i at time i is given by $(x_i^2 + v_i^2)/2$. Calculate analytically the energy at time $i+1$ in the algorithm of part (b), in terms of x_i and v_i. What do you get for $(E_{i+1} - E_i)/\Delta t \sim \dot{E}$? Is this consistent with your graph? Do the same for the algorithm of part (c). Why is it better?

6. Assume you wish to solve for the eigenstates of an electron in an infinite 1D quantum well with an electric field, $\mathcal{E}$ applied across the well. Your Hamiltonian is given by:

$$-\frac{\hbar^2}{2m}\frac{\partial^2 \psi}{\partial x^2} - e\mathcal{E}x\,\psi = E\,\psi$$

where $0 \leq x \leq L$. You are interested in finding out how the energy of the first two eigenstates changes as a function of the strength of the electric field.

(a) What is the unit in which you will measure distance?

(b) What is the unit with which you will measure the energy? What is its physical meaning?

(c) What is the dimensionless parameter that is the "control knob" that corresponds to increasing the field? What is its meaning?

(d) At what value of this parameter would you expect to see deviations from the infinite square well solutions?

(e) Solve the problem numerically. Plot the energy of the lowest ten eigenstates for a "small" value of the applied field. How do they vary with n, the number of the eigenstate?

(f) Solve the problem numerically. Plot the energy of the lowest ten eigenstates for a "large" value of the applied field. How do they vary with n, the number of the eigenstate?

(g) Plot the energy of the lowest two eigenstates as a function of the applied field sweeping from small to large values of the applied field. Is your estimate above for the critical values of the field correct?

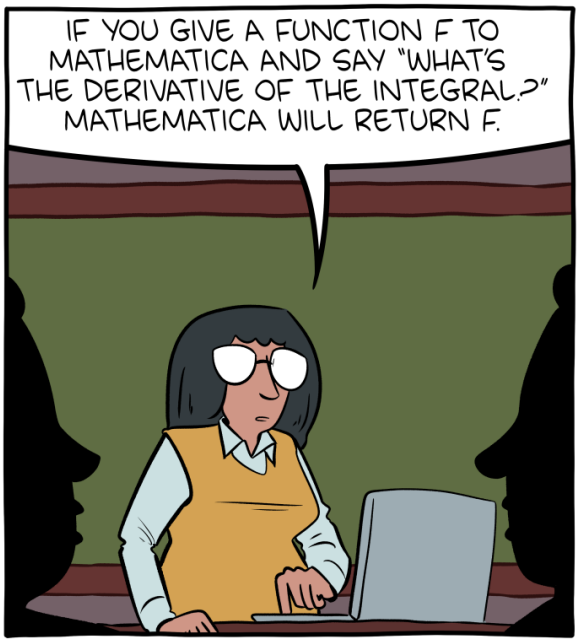


Figure 1: Cartoon by Z. Weinersmith, *"Saturday Morning Breakfast Cereal"*.

HW 2/1

a) From Maxwell's equations:

$$\vec{\nabla} \times \vec{B} = \frac{4\pi}{c}\vec{J} + \frac{1}{c}\frac{\partial \vec{E}}{\partial t}$$

Taking the divergence:

$$\vec{\nabla} \cdot (\vec{\nabla} \times \vec{B}) = \frac{4\pi}{c}\vec{\nabla}\cdot\vec{J} + \frac{1}{c}\frac{\partial}{\partial t}(\vec{\nabla}\cdot\vec{E})$$

where we have interchanged $\vec{\nabla}$ & $\frac{\partial}{\partial t}$. The LHS is zero.

Substituting: $\vec{\nabla}\cdot\vec{E} = 4\pi\rho$:

$$\frac{4\pi}{c}\vec{\nabla}\cdot\vec{J} + \frac{1}{c}\frac{\partial}{\partial t}(4\pi\rho) = 0$$

$$\Rightarrow \vec{\nabla}\cdot\vec{J} + \frac{\partial \rho}{\partial t} = 0.$$

b) Normally, in the static case, $\frac{\partial \vec{B}}{\partial t} = 0$, then $\vec{\nabla}\times\vec{E} = 0$ implies we may choose $\vec{E} = -\vec{\nabla}\varphi$, so that $\vec{\nabla}\times\vec{E} = \vec{\nabla}\times(\vec{\nabla}\varphi) = 0$ is automatically satisfied.

Now, we are not only interested in the static case. Let $\vec{E} = \vec{\nabla}\varphi + \vec{\Omega}$. Then

$$\vec{\nabla}\times\vec{E} = \vec{\nabla}\times(\vec{\nabla}\varphi) + \vec{\nabla}\times\vec{\Omega} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}$$

$\vec{\nabla}\times(\vec{\nabla}\varphi) = 0$

This can be satisfied if we choose $\vec{\Omega} = -\frac{1}{c}\frac{\partial \vec{A}}{\partial t}$

$$\vec{\nabla}\times\vec{\Omega} = \vec{\nabla}\times\left(-\frac{1}{c}\frac{\partial \vec{A}}{\partial t}\right) = -\frac{1}{c}\frac{\partial}{\partial t}(\vec{\nabla}\times\vec{A}) = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}$$

Thus: $\vec{E} = -\vec{\nabla}\varphi - \frac{1}{c}\frac{\partial}{\partial t}\vec{A}$

c) If $F_{ij} = \epsilon_{ijk} B_k$, then

$$\tfrac{1}{2}\epsilon_{ijk} F_{ij} = \tfrac{1}{2}\epsilon_{ijk}\epsilon_{ijl} B_l$$

We know $\epsilon_{ijk}\epsilon_{klm} = \delta_{il}\delta_{jm} - \delta_{jl}\delta_{im}$

$$= -\epsilon_{kji}\epsilon_{klm} = \epsilon_{kij}\epsilon_{klm}$$

If we multiply by δ_{il}

$$\epsilon_{kij}\epsilon_{klm}\delta_{il} = \epsilon_{kij}\epsilon_{kim}$$

The indices differ from $\epsilon_{ijk}\epsilon_{ijl}$, but the pattern is the same. Simplifying:

$$\epsilon_{kij}\epsilon_{kim} = \delta_{il}(\delta_{il}\delta_{jm} - \delta_{jl}\delta_{im})$$

$$= \delta_{il}\delta_{il}\delta_{jm} - \delta_{jl}\delta_{im}$$

$$= 3\delta_{jm} - \delta_{jm} = 2\delta_{jm}$$

Or, if we rename indices. ($k \to i$, $i \to j$, $j \to k$, $m \to l$)

$$\epsilon_{ijk}\epsilon_{ijl} = 2\delta_{kl}.$$

So $\tfrac{1}{2}\epsilon_{ijk}\epsilon_{ijl} B_l = \tfrac{1}{2} 2\delta_{kl} B_l = B_k$

or $\tfrac{1}{2}\epsilon_{ijk} F_{ij} = B_k$

d) $F_{ij} = \epsilon_{ijk} B_k$ (as above) But

$\vec{B} = \vec{\nabla} \times \vec{A}$ or $B_k = \epsilon_{\ell mk} \partial_\ell A_m$

$\Rightarrow F_{ij} = \epsilon_{ijk} \epsilon_{\ell mk} \partial_\ell A_m$

$= -\epsilon_{ijk} \epsilon_{km\ell} \partial_\ell A_m$

$= +\epsilon_{ijk} \epsilon_{k\ell m} \partial_\ell A_m$

$= (\delta_{i\ell} \delta_{jm} - \delta_{j\ell} \delta_{im}) \partial_\ell A_m$

$= \partial_i A_j - \partial_j A_i$

e) $F_{4j} = \partial_4 A_j - \partial_j A_4 = -(\partial_j A_4 - \partial_4 A_j) = -F_{j4}$

Thus these components are also antisymmetric.

Furthermore:

$F_{4j} = \partial_4 A_j - \partial_j A_4$

$= \frac{1}{ic} \frac{\partial}{\partial t} A_j - \partial_j (i\varphi)$

$= -i \left(\vec{\nabla} \varphi + \frac{1}{c} \frac{\partial}{\partial t} \vec{A} \right) = E_j$

or $\vec{E} = -\vec{\nabla} \varphi - \frac{1}{c} \frac{\partial}{\partial t} \vec{A}$

f) $F_{ij} = \partial_i A_j - \partial_j A_i$

then $\partial_i F_{jk} + \partial_j F_{ki} + \partial_k F_{ij}$

$= \partial_i (\partial_j A_k - \partial_k A_j) + \partial_j (\partial_k A_i - \partial_i A_k)$

$+ \partial_k (\partial_i A_j - \partial_j A_i)$

Since $\partial_i \partial_j A_k = \partial_j \partial_i A_k$, this sums to zero.

g) Let $j=k$. (It doesn't matter which case we choose.) Then (1) becomes:

$$\partial_i F_{jj} + \partial_j F_{ji} + \partial_j F_{ij}$$

$$= \partial_j (-F_{ij}) + \partial_j F_{ij} = 0.$$

If $i=j=k$, then all F's are zero. Note that all these cases are true by definition, & are not related to Maxwell's eqns.

h) Let $i=4$. $j,k = (1,2,3)$

$$\partial_4 \partial F_{jk} + \partial_j F_{k4} + \partial_k F_{4j} =$$

$$\frac{1}{ic} \frac{\partial}{\partial t} \epsilon_{jkl} B_l + \partial_j (-i E_k) + \partial_k (i E_j)$$

$$= i \left\{ -\epsilon_{jkl} \frac{1}{c} \frac{\partial}{\partial t} B_l + \partial_k E_j - \partial_j E_k \right\}$$

$$= -i \left(\vec{\nabla} \times \vec{E} + \frac{1}{c} \frac{\partial}{\partial t} \vec{B} \right)_l$$

i) $F_{ij} = \epsilon_{ijk} B_k$, for $ijk = (1,2,3)$

Note if we specify ij, then we know k.

$$\partial_i F_{jk} + \partial_j F_{ki} + \partial_k F_{ij}$$

$$\partial_i \epsilon_{jkl} B_l + \partial_j \epsilon_{kim} B_m + \partial_k \epsilon_{ijn} B_n$$

($l=i$, $m=j$, $n=k$)

$$= \partial_i \epsilon_{jki} B_i + \partial_j \epsilon_{kij} B_j + \partial_k \epsilon_{ijk} B_k$$

$$= -\partial_i \epsilon_{ikj} B_i - \partial_j \epsilon_{ikj} B_j + \partial_k \epsilon_{ijk} B_k$$

$$= \partial_i \epsilon_{ijk} B_i + \partial_j \epsilon_{ijk} B_j + \partial_k \epsilon_{ijk} B_k$$

$$= \epsilon_{ijk} (\partial_i B_i + \partial_j B_j + \partial_k B_k) = 0$$

$$\Rightarrow \vec{\nabla} \cdot \vec{B} = 0$$

J) $\mathcal{E}_q(2)$ is:

$$\partial_k F_{\ell k} = \frac{4\pi}{c} J_\ell$$

For $\ell = 4$

$$\partial_k F_{4k} = \frac{4\pi}{c} J_4$$

Using our above expression for $F_{4k} = iE_k$, $k = 1, 2, 3$

$$i \partial_k E_k \Big|_{k=1,2,3} + \underbrace{\partial_4 F_{44}}_{0} = \frac{4\pi}{c} i c\rho$$

or $$\vec{\nabla} \cdot \vec{E} = 4\pi\rho.$$

For $\ell = 1$, 2 or 3.

$$\partial_k F_{\ell k} = \partial_k \epsilon_{\ell km} B_m \Big|_{k=1,2,3} + \partial_4 F_{\ell 4}$$

$$= \epsilon_{kml} \partial_k B_m \Big|_{k=1,2,3} + \frac{1}{ic} \frac{\partial}{\partial t} (-iE_\ell) = \frac{4\pi}{c} J_\ell$$

∂_2 $\quad (\vec{\nabla} \times \vec{B})_\ell - \left(\frac{1}{c}\frac{\partial}{\partial t}\vec{E}\right)_\ell = \left(\frac{4\pi}{c}\vec{J}\right)_\ell$

$\Rightarrow \quad \vec{\nabla} \times \vec{B} = \frac{1}{c}\frac{\partial}{\partial t}\vec{E} + \frac{4\pi}{c}\vec{J}$

k) Let ℓ be 1, 2 or 3. $\quad J = (\rho v_x, \rho v_y, \rho v_z, ic\rho)$

$$f_\ell = \frac{1}{c} F_{\ell k} J_k$$

$$= \frac{1}{c}\left\{ \epsilon_{\ell k j} B_j \Big|_{\substack{j,k \\ \neq 4}} \rho v_k + -iE_\ell c\rho \right\}$$

$$= \left(\rho \vec{E} + \rho \frac{\vec{v} \times B}{c}\right)_\ell .$$

For $\ell = 4$

$$f_4 = \frac{1}{c} F_{4k} J_k = \frac{1}{c}(iE_k)\cdot \rho \cdot v_k = \frac{i}{c}(\vec{v}\cdot\vec{E})$$

= work done / unit time. = <u>power</u>.

l) $J' = (0, 0, -c\rho \sinh\alpha, ic\rho\cosh\alpha)$

For $v \ll c$,

$$\cosh\alpha = \frac{1}{\sqrt{1 - v^2/c^2}} \approx \frac{1}{1 - \frac{1}{2}\frac{v^2}{c^2}} \sim 1 + \frac{1}{2}\frac{v^2}{c^2}.$$

$\cosh^2\alpha - \sinh^2\alpha = 1$

$\sinh\alpha \sim \frac{v}{c}$ for $v \ll c$.

$J_z = -\rho v$

R = J_4 increases due to Lorentz contraction.

m) $F'_{23} = L_{2i} L_{3j} F_{ij}$

$= L_{22} L_{34} F_{24} = i \sinh\alpha (-iE_y)$

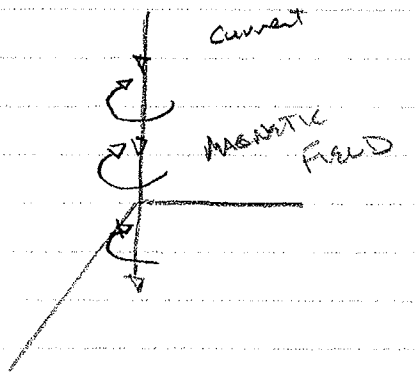
$= + \sinh\alpha E_y = B_x'$

$F'_{31} = L_{3i} L_{1j} F_{ij}$

$= L_{34} L_{11} F_{41} = i \sinh\alpha (iE_x) = B_y'$

$\vec{B} = \lambda \sinh\alpha \ln r (\sin\theta \hat{x} - \cos\theta \hat{y})$

$= -\lambda \sinh\alpha \ln r \hat{\theta}$



The field is that about a wire with a current flowing down wire!

HW6 Solutions

Problem 2

(a) Zeroes

Solve the equation:

$$4x^3 - 32x^2 + 66x - 18 = 0$$

obtaining both analytic and numerical answers. Plot the function to verify that the roots are correct.

In[315]:=
```
eqn = 4 x^3 - 32 x^2 + 66 x - 18 == 0
```

Out[315]=
$$-18 + 66\,x - 32\,x^2 + 4\,x^3 == 0$$

In[316]:=
```
Solve[eqn, x]
```

Out[316]=
$$\left\{\{x \to 3\}, \left\{x \to \frac{1}{2}\left(5 - \sqrt{19}\right)\right\}, \left\{x \to \frac{1}{2}\left(5 + \sqrt{19}\right)\right\}\right\}$$

In[317]:=
```
soln1 = NSolve[eqn, x]
```

Out[317]=
$$\{\{x \to 0.320551\}, \{x \to 3.\}, \{x \to 4.67945\}\}$$

(b) Extrema

Now we look at the extrema:

In[422]:=
```
deqn = D[eqn, x]
```

Out[422]=
$$66 - 64\,x + 12\,x^2 == 0$$

In[423]:=
```
Solve[deqn, x]
```

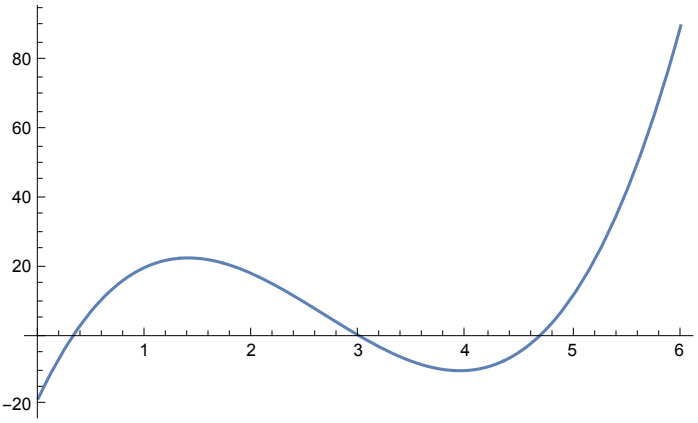
Out[423]=
$$\left\{\left\{x \to \frac{1}{6}\left(16 - \sqrt{58}\right)\right\}, \left\{x \to \frac{1}{6}\left(16 + \sqrt{58}\right)\right\}\right\}$$

In[425]:=
```
soln2 = NSolve[deqn, x]
```

Out[425]=
$$\{\{x \to 1.39737\}, \{x \to 3.93596\}\}$$

In[321]:=
```
Plot[4 x^3 - 32 x^2 + 66 x - 18 , {x, 0, 6}]
```

Out[321]=



If we want to get fancy, we can mark the positions of the roots. First, we need to extract the answers which were given in the form of replacement rules. We want them as a list:

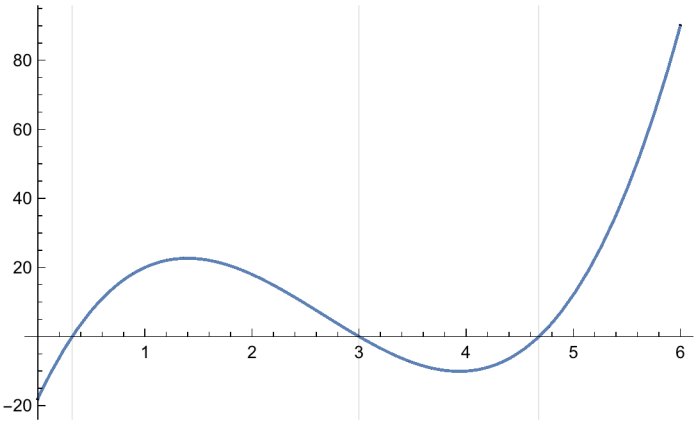
In[429]:=
```
xvals1 = x /. soln1
```

Out[429]=
```
{0.320551, 3., 4.67945}
```

Now we can pass this list to the option **GridLines**.

In[433]:=
```
Plot[4 x^3 - 32 x^2 + 66 x - 18 , {x, 0, 6}, GridLines → {xvals1, None}]
```

Out[433]=



Note that they hit the zeroes, as promised. We can add the extrema to the plot as well. There are lots of ways to do this. Here's one: First get the x-values as a list:

In[428]:=
```
xvals2 = x /. soln2
```

Out[428]=
```
{1.39737, 3.93596}
```

Add them to our list of x-grid lines

In[430]:=
```
xvals = Join[xvals1, xvals2]
```

Out[430]=
```
{0.320551, 3., 4.67945, 1.39737, 3.93596}
```

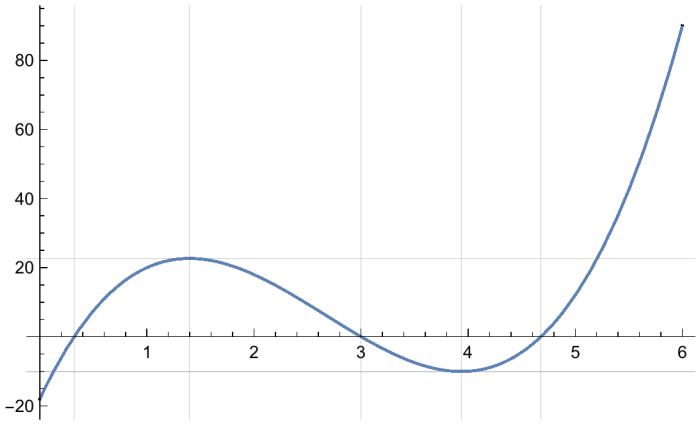
To get fancy, add the y-values of the extrema to our gridlines:

In[426]:=
```
yvals = (4 x^3 - 32 x^2 + 66 x - 18) /. soln2
```

Out[426]=
```
{22.6561, -10.0635}
```

In[448]:=
```
Plot[4 x^3 - 32 x^2 + 66 x - 18 , {x, 0, 6}, GridLines → {xvals, yvals}]
```

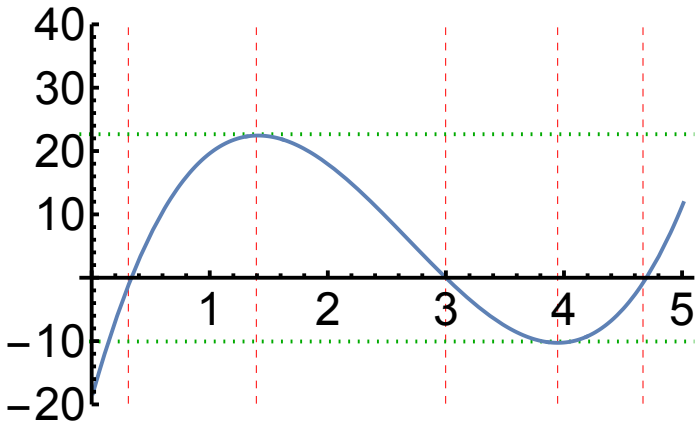
Out[448]=



Too simple. Let's add color! Fonts! Line styles!

In[461]:=
```
Plot[4 x^3 - 32 x^2 + 66 x - 18 , {x, 0, 5},
 GridLines → {xvals, yvals}, PlotStyle → Thick, AxesStyle → Thick,
 GridLinesStyle → {Directive[Red, Dashed], Directive[Darker[Green], Dotted, Thick]},
 BaseStyle -> {FontWeight -> "Plain", FontSize → 24}, PlotRange → {-20, 40}]
```

Out[461]=



Problem 3

Plot the Taylor series expansion of Sin[x] in the interval $0< x < 2\pi$. Determine the total number of terms you need to get reasonable accuracy.

Below we make a list of the expansions. The innermost **Table** command generates each term in the expansion up to some order; the **Total** command sums up the list; the outer **Table** command repeats the process, each time increasing the number of terms in the innermost **Table** command.

In[323]:=
```
sineExpansions = Table[Total[Table[(-1)^n (x)^(2 n+1)/(2 n+1)!, {n, 0, nmax}]], {nmax, 0, 9}];
```

Here they are in a nice format:

```
TableForm[sineExpansions]
```

$$x$$

$$x - \frac{x^3}{6}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362\,880}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362\,880} - \frac{x^{11}}{39\,916\,800}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362\,880} - \frac{x^{11}}{39\,916\,800} + \frac{x^{13}}{6\,227\,020\,800}$$

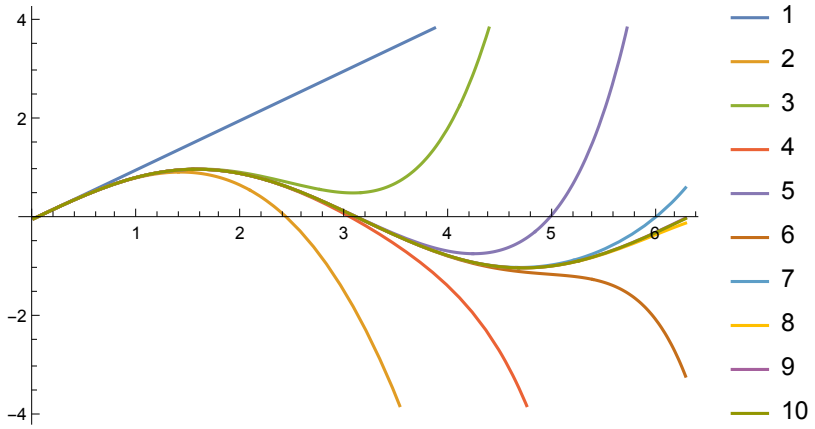
$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362\,880} - \frac{x^{11}}{39\,916\,800} + \frac{x^{13}}{6\,227\,020\,800} - \frac{x^{15}}{1\,307\,674\,368\,000}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362\,880} - \frac{x^{11}}{39\,916\,800} + \frac{x^{13}}{6\,227\,020\,800} - \frac{x^{15}}{1\,307\,674\,368\,000} + \frac{x^{17}}{355\,687\,428\,096\,000}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362\,880} - \frac{x^{11}}{39\,916\,800} + \frac{x^{13}}{6\,227\,020\,800} - \frac{x^{15}}{1\,307\,674\,368\,000} + \frac{x^{17}}{355\,687\,428\,096\,000} - \frac{x^{19}}{121\,645\,100\,408\,832\,000}$$

Here we plot them. It's clear that we need at least 8 terms in the series, which still gives us a noticeable error at the end.

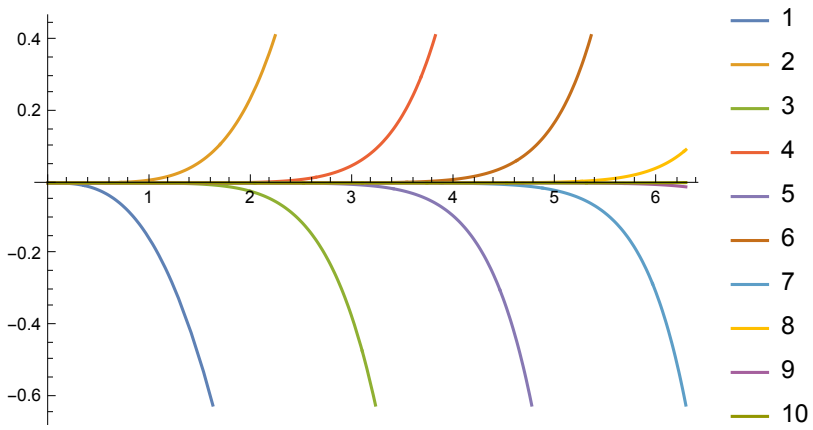
```
Plot[sineExpansions, {x, 0, 2 π}, PlotLegends → Table[n + 1, {n, 0, 9}]]
```



The scale makes it hard to judge. Let's subtract off the sine function and look at the error.

```
errors = Table[Sin[x] - Total[Table[(-1)^n (x)^(2 n+1)/(2 n+1)!, {n, 0, nmax}]], {nmax, 0, 9}];
```

```
Plot[errors, {x, 0, 2 π}, PlotLegends → Table[n + 1, {n, 0, 9}]]
```



From the above curve we see that with 8 terms we have about 10% error at the end of the interal. It's better to go with 9 terms.

Problem 4

Solve the problem

$$\mathcal{A}\ x = b$$

where $\mathcal{A}_{i,j} = \sin(i * j)$ and $b_i = i$ where i, j run from 1 to 10.

```
amat = Table[Sin[i * j], {i, 1, 10}, {j, 1, 10}];
b = Table[i, {i, 1, 10}];
```

```
LinearSolve[amat, b]
```

```
$Aborted
```

Oops! This would give a messy symbolic answer. We want a numerical answer. So simply make sure we numbers and not symbols in our problem:

```
numAmat = N[amat];
```

```
bsoln = LinearSolve[numAmat, b]
```

```
{2.83492, -5.65071, -16.77, -9.82246,
 2.24527, -5.75988, -2.63877, 2.96037, 25.6627, 23.0544}
```

We can verify the answer:

```
amat.bsoln // MatrixForm
```

$$\begin{pmatrix} 1. \\ 2. \\ 3. \\ 4. \\ 5. \\ 6. \\ 7. \\ 8. \\ 9. \\ 10. \end{pmatrix}$$

Problem 5

We consider a simple harmonic oscillator. The equation of motion is:

$$\frac{\partial^2 x}{\partial t^2} = -x$$

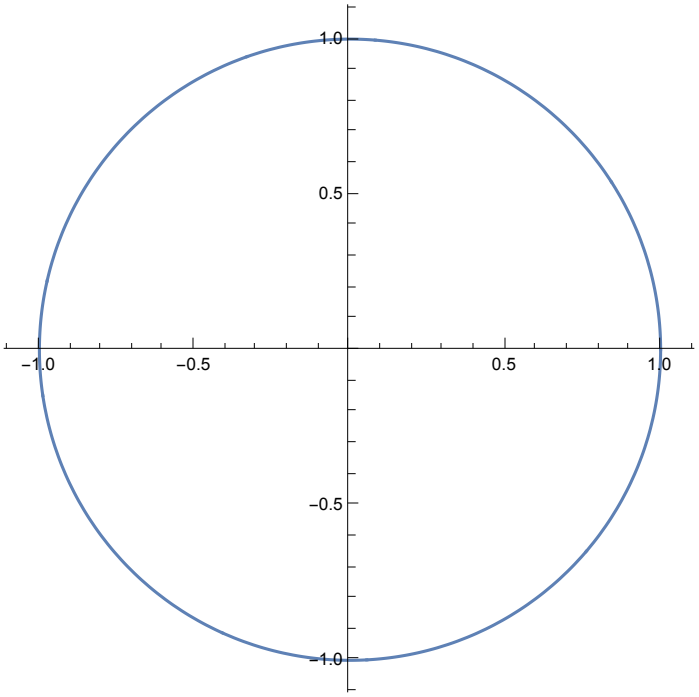
with the initial conditions x(0)=1 and $\dot{x}(0)=0$.

(a) Exact solution

By inspection x(t)=cos(t). A phase space plot is a circle:

In[380]:=
```
cplot = ParametricPlot[{Cos[t], -Sin[t]}, {t, 0, 2 π}]
```

Out[380]=



(b) Standard Euler

Here we have an explicit scheme:

$$v_{i+1} = v_i - x_i \Delta t$$

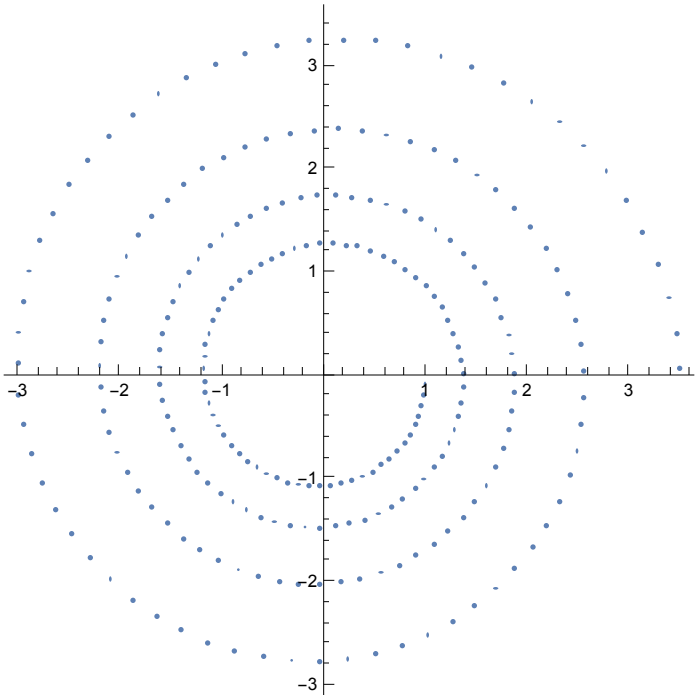
$$x_{i+1} = x_i + v_i \Delta t$$

We do this by a simple table.

In[360]:=
```
state = {1, 0};
dt = .1;
trajectory = Table[x = state[[1]] + state[[2]] * dt;
    v = state[[2]] - state[[1]] * dt;
    state = {x, v};
    state, {t, 0, 8 π, dt}];
```

In[363]:=
```
ListPlot[trajectory, AspectRatio → 1]
```

Out[363]=

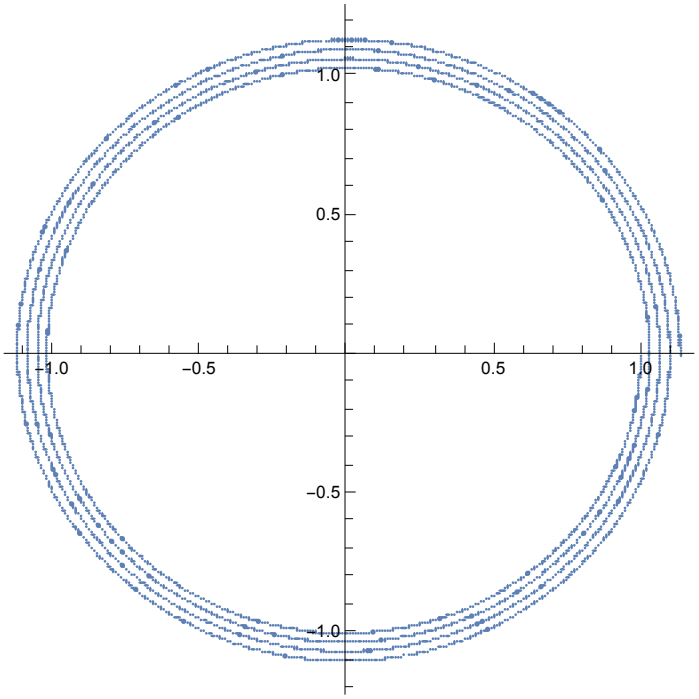


Yikes! Let's try a smaller time step:

In[364]:=
```
state = {1, 0};
dt = .01;
trajectory = Table[x = state[[1]] + state[[2]] * dt;
    v = state[[2]] - state[[1]] * dt;
    state = {x, v};
    state, {t, 0, 8 π, dt}];
```

In[367]:=
```
ListPlot[trajectory, AspectRatio → 1]
```

Out[367]=



It's better, but not stable.

(c) Implicit scheme:

Here we have an implicit scheme:

$v_{i+1} = v_i - x_i \Delta t$

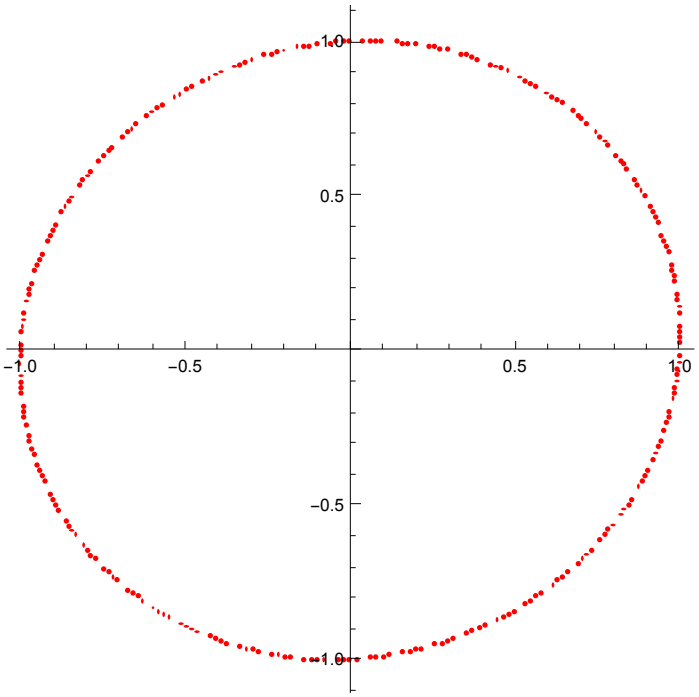
$x_{i+1} = x_i + v_{i+1} \Delta t$

We do this by a simple table. Note that we use the new value of v to update x.

In[368]:=
```
state = {1, 0};
dt = .1;
trajectory = Table[v = state[[2]] - state[[1]] * dt;
    x = state[[1]] + v * dt;
    state = {x, v};
    state, {t, 0, 8 π, dt}];
```

In[382]:=
```
tplot = ListPlot[trajectory, AspectRatio → 1, PlotStyle → Red]
```

Out[382]=



It's stable at the larger time step.

(d) Examining the energy

In the explicit scheme:

In[373]:=
```
Clear[v, x]
```

In[359]:=
```
Etot = Simplify[ (vi - xi Δt)^2 +  (xi + vi Δt)^2]
```

Out[359]= $\left(1+\Delta t^2\right)\left(v_i^2+x_i^2\right)$

This shows that the energy increases by a factor of $(1+\Delta t^2)$ every time step. So decreasing the time step helps, but it will still be unstable. In the implicit scheme:

$$v_{i+1} = v_i - x_i\,\Delta t$$
$$x_{i+1} = x_i + v_{i+1}\,\Delta t = x_i + (v_i - x_i\,\Delta t)\,\Delta t$$

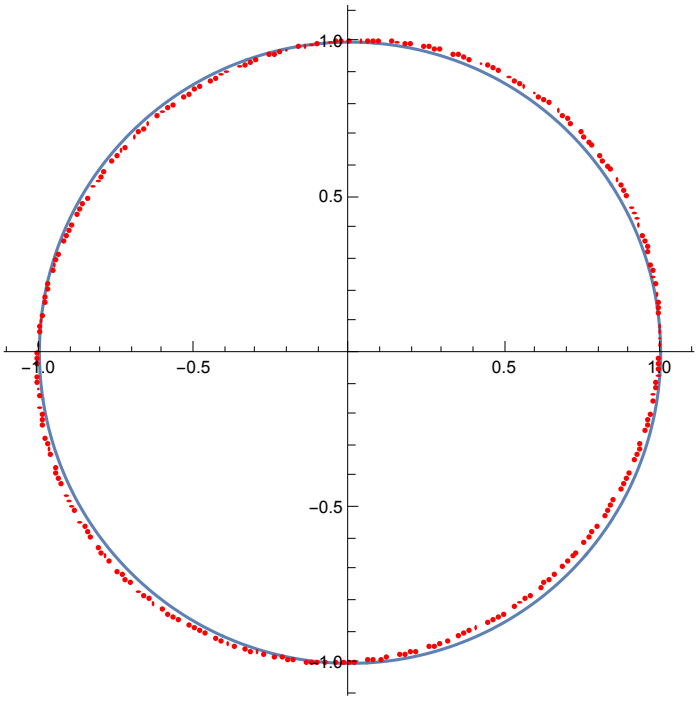
In[378]:=
```
Etot = Simplify[Expand[ (vi - xi Δt)^2 +  (xi + (vi - xi Δt) Δt)^2]]
```

Out[378]= $\left(1+\Delta t^2\right) v_i^2 - 2\,\Delta t^3\, v_i\, x_i + \left(1-\Delta t^2+\Delta t^4\right) x_i^2$

What a mess! Why does this work? Well we see that the iteration systematically overestimates the kinetic energy at the next update and underestimates the potential. In this scaling it is by roughly the same amount. If we plot the exact solution and this implicit solution we see:

In[383]:=
```
Show[cplot, tplot]
```

Out[383]=



They are not the same! Our new solution is stable, but not entirely accurate. Do not confuse stability with accuracy!

Bonus material:

A better implicit scheme is:

$$x_{i+1} = x_i + (v_i + v_{i+1}) \frac{\Delta t}{2}$$

$$v_{i+1} = v_i - (x_i + x_{i+1}) \frac{\Delta t}{2}$$

(Can you see why this might be a good thing to try?)

In order to use these we must solve the equations for the new {x,v} in terms of the old ones:

In[404]:=
```
Clear [dt, v1, v2, x1, x2]
```

In[406]:=
```
ans = Part[
  Simplify[Solve[{x2 == x1 + (v1 + v2) dt / 2, v2 == v1 - (x1 + x2) dt / 2}, {x2, v2}]], 1]
```

Out[406]=
$$\left\{x2 \to \frac{4\,dt\,v1 + 4\,x1 - dt^2\,x1}{4 + dt^2},\ v2 \to -\frac{\left(-4 + dt^2\right) v1 + 4\,dt\,x1}{4 + dt^2}\right\}$$

This gives us the updated x and v in terms of the old. If we check energy conservation, it's exact.

In[407]:=
```
Etot = (x2^2 + v2^2) /. ans // Simplify
```

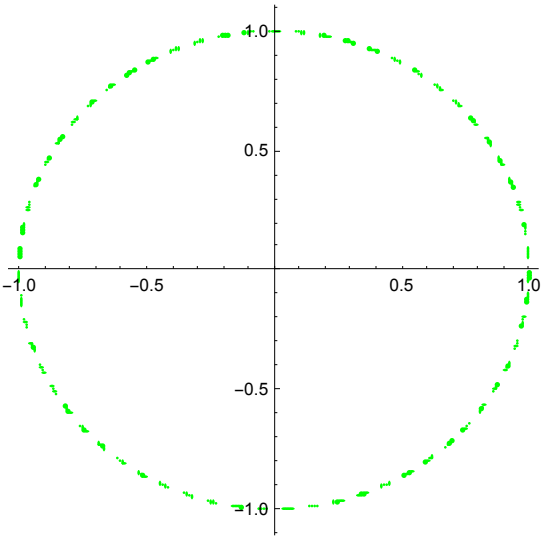
Out[407]=
$$v1^2 + x1^2$$

Let's implement the scheme:

In[416]:=
```
state = {1, 0};
dt = .1;
trajectory = Table[x1 = state[[1]];
   v1 = state[[2]];
   v2 = -((-4 + dt^2) v1 + 4 dt x1)/(4 + dt^2);
   x2 = (4 dt v1 + 4 x1 - dt^2 x1)/(4 + dt^2);
   state = {x2, v2};
   state, {t, 0, 8 π, dt}];
```

In[420]:=
```
iplot = ListPlot[trajectory, AspectRatio → 1, PlotStyle → Green]
```

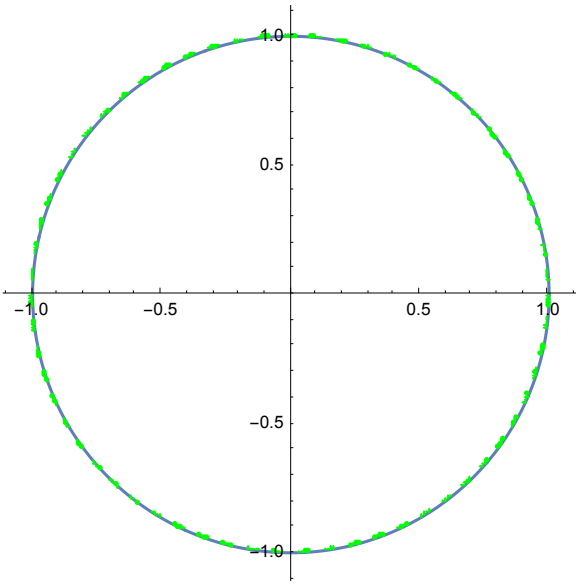
Out[420]=



And compare it to the exact solution:

In[421]:=
```
Show[cplot, iplot]
```

Out[421]=



HW6 Solutions

Problem 6

Part (a)

We will measure distances (or x) in units of L, the well width. Other choices prevent us from taking the $\mathcal{E} \to 0$ limit.

Part (b)

The energy unit is $E_0 = \frac{\hbar^2}{2\,m\,L^2}$. This is the confinement energy associated with particle-in-a-box states. The energies for PIB states will vary as $n^2\,\pi^2$ in these units.

Part (c)

The dimensionless parameter is $\mathscr{E} = \frac{e\mathcal{E}L}{E_0}$. It represents the maximum work done (or change in potential energy) of an electron moved across the well, measured in units of E_0.

Part (d)

We expect to see deviations in the nth state when $\mathscr{E} \sim n^2\,\pi^2$

Part (e)

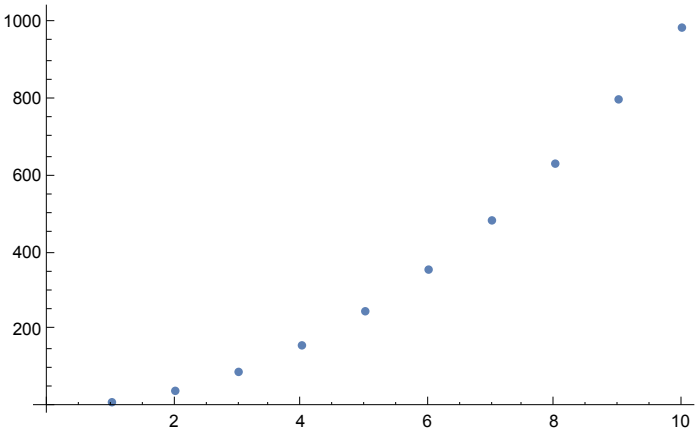
The only change in our routine is the potential energy. Here's the matrix:

```
npts = 200;
dx = 1. / (npts + 1);
hmat = Table[0, {i, 1, npts}, {j, 1, npts}];
Do[hmat[[i, i]] = eps (i dx) + 2 / dx^2, {i, 1, npts}]
Do[hmat[[i, i + 1]] = -1 / dx^2, {i, 1, npts - 1}]
Do[hmat[[i + 1, i]] = -1 / dx^2, {i, 1, npts - 1}]
hmatVal = hmat /. {eps → 4};
```

```
evals = Eigenvalues[hmatVal];
```

Now we plot the lowest few energies:

```
ListPlot[Take[Sort[evals], 10]]
```



Here are their values. For small $\mathscr{E}$ they are close to $n^2\,\pi^2$, but of course, they are not equal to it. At higher values of n the energy becomes closer to $n^2\,\pi^2$

```
Take[Sort[evals], 10] / π^2
```

```
{1.20084, 4.20285, 9.20131, 16.1976,
 25.19, 36.1764, 49.1539, 64.1194, 81.0692, 99.9993}
```

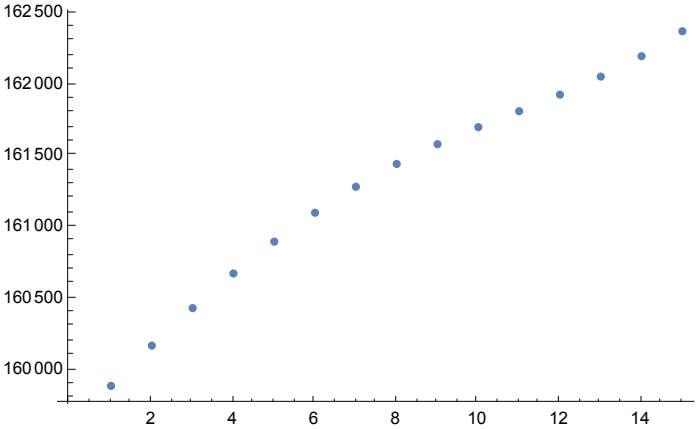
Part (f)

```
hmatVal = hmat /. {eps → 1000};
```

```
evals = Eigenvalues[hmatVal];
```

Now we plot the lowest few energies:

```
ListPlot[Sort[Take[evals, 15]]]
```



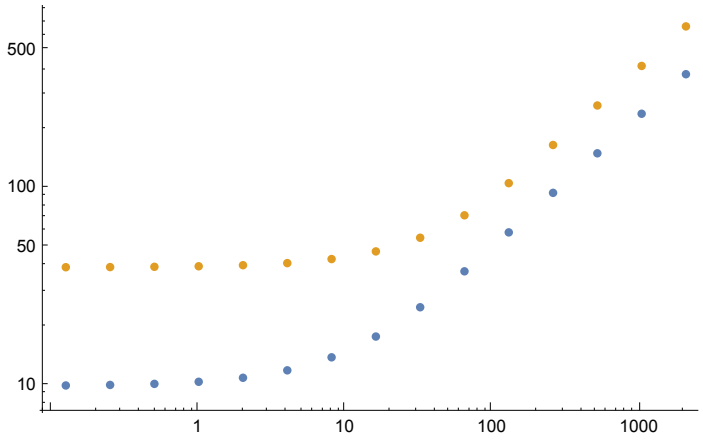
We expect the crossover between states that feel the potential and those that don't to be when $\mathscr{E} \sim n^2\,\pi^2$. If $\pi^2 \sim 10$, then we should see a crossover around the 10th state. This is bourne out by the numerics.

Part (g)

Now let's see how the groundstate and first excited state vary with the perturbation. We expect to see the groundstate change when $\mathscr{E} \sim \pi^2$, and the first excited state change when $\mathscr{E} \sim 2^2\,\pi^2$.

```
maxn = 15;
allEvals = Table[hmatVal = (hmat /. eps → 2^(n-4));
    Eigenvalues[hmatVal], {n, 0, maxn}];

list1 = Table[{2^(n-4), allEvals[[n + 1, npts]]}, {n, 1, maxn}];
list2 = Table[{2^(n-4), allEvals[[n + 1, npts - 1]]}, {n, 1, maxn}];
ListLogLogPlot[{list1, list2}]
```



This looks pretty good!