

Computing for Mathematical Physics

2022/23

Homework5

Mark for homework5: 26/32 but unfortunately over 10 days too late.
(to be competed by your marker)

Feedback from marker:

Q1: 11/14.

Q2: 8/9.

Q3: 5/5.

Q4: 2/4 (note, the marking scheme is deliberately strict in Q4).

Good answers, losing marks to just relatively minor issues, which you could catch with a little more time/care/attention to detail. See my comments in purple under your answers throughout the script below for more information on these. The main message, looking ahead to the last 2-3 weeks of the course + exams, is that you can do this.

(to be competed by your marker)

Give your answers in the code cells marked (* Enter your solution here *)

Double click the vertical braces on the RHS of each of the question headings to open and view them.

■ 1. Plotting multiple functions on the same graph, labeled, and legended. [14 marks]

- a) Define the wavefunction $\psi[n_, x_]$ in your code according to

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}} \frac{1}{\pi^{1/4}} e^{-\frac{1}{2}x^2} H_n(x),$$

wherein $H_n(x)$ are the Hermite polynomials, implemented in *Mathematica* as `HermiteH[n, x]`.

- b) Plot $\psi_n(x)$ for $n = 1, 2, 3$, over the range $-5 \leq x \leq 5$.

- Fix the y-axis range to $-0.7 \leq y \leq 0.7$.
- Use a different colour for each of the three lines plotted.
- Label the axes.
- Include a title “Wavefunctions $\psi_n(x)$ ”
- Include a legend indicating clearly what each line represents.
- Place the legend in the top right of the graph.
- Use `LabelStyle` to set the Times font family, in Black, with font size 18, for all textual elements.
- Use grid lines.

[14 marks]

3/6/23 10:59:25 In[1]:=

$$\psi[n_, x_] := \frac{1}{\sqrt{2^n n!}} \frac{1}{\pi^{0.25}} e^{-0.5 x^2} \text{HermiteH}[n, x]$$

(*Create function for phi*)

3/6/23 10:59:26 In[2]:=

n = {1, 2, 3}

3/6/23 10:59:26 Out[2]=

{1, 2, 3}

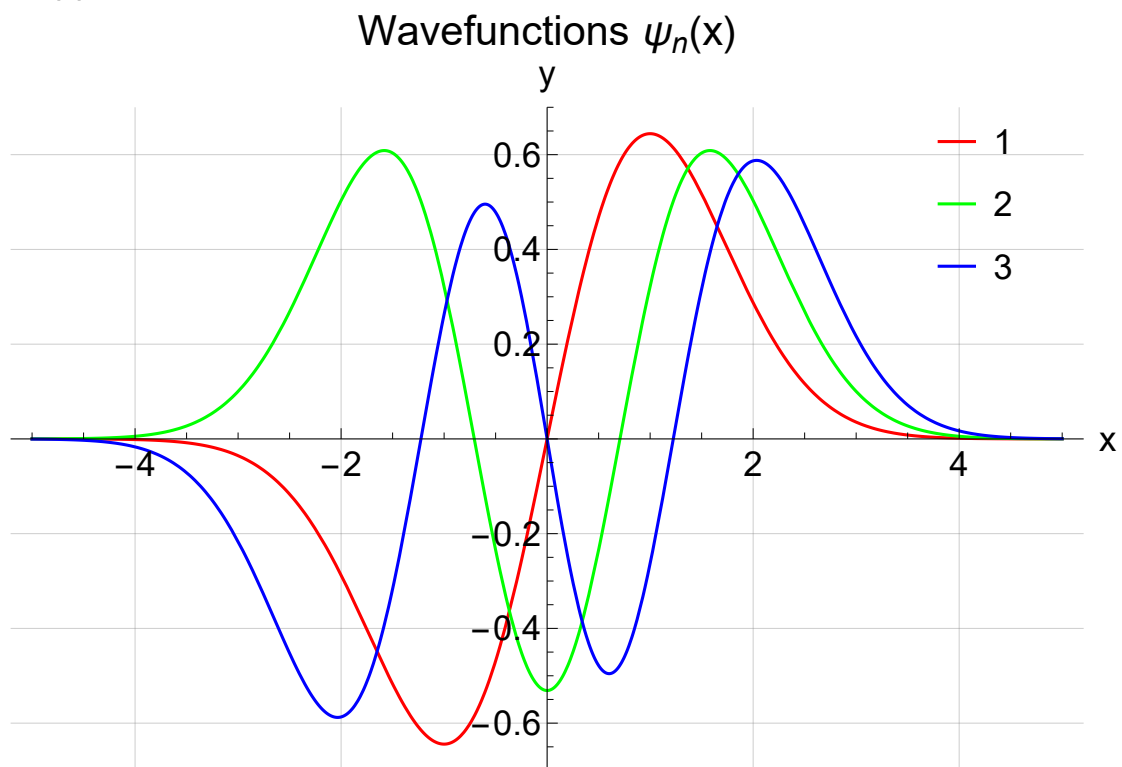
3/6/23 11:29:41 In[21]:=

```

Plot[Evaluate[Thread[ψ[n, x]]], (*use the function and
  thread so that all graphs for all values of n can be created*)
{x, -5, 5}, (*Define Range for x values*)
PlotLabel → "Wavefunctions ψn(x)", (*Label Graph*)
PlotRange → {-0.7, 0.7}, (*set limit for y axis*)
PlotLegends → Placed[{"1", "2", "3"}, {0.9, 0.85}],
(*Create legend and place in the top right of the graph*)
AxesLabel → {"x", "y"}, (*Label Axis*)
PlotStyle → Table[{Hue[i]}, {i, 0, 1,  $\frac{1}{\text{Length}[n]}$ }],
(*Give each line a different colour*)
LabelStyle → {Black, Times, FontSize → 18}, (*Give style for each line*)
GridLines → Automatic, (*Add gridlines*)
ImageSize → Large (*make image large but that hasnt worked*)
]
? Placed[{"1", "2", "3"}, {1, 1}]

```

3/6/23 11:29:41 Out[21]=



3/6/23 11:29:41 Out[22]=

```
Information[Placed[{{1, 2, 3}, {1, 1}}, LongForm → False]
```

3/6/23 10:59:26 In[5]:=

- Question 1: 11/14
- Good effort.

- Lost 1 mark for not getting the font right: you just need replace Times by FontFamily->"Times" in LabelStyle.
- Lost 1 mark for not labelling y-axis as " $\psi_n(x)$ ".
- Lost 1 mark for the legend contents being unclear/abstract ("1", "2", "3").
- **Please**, see also my answer below for an alternative perspective, with slightly more standard ways of doing things, e.g. providing multiple functions to plot inside a single plot command.

```

(* Marker/Coordinator's code here *)

(* Part a *)
(* Defining  $\psi_n(x)$  as written in the question *)

$$\psi[n_, x_] := \frac{1}{\sqrt{2^n n!}} \frac{1}{\pi^{1/4}} \text{Exp}\left[-\frac{1}{2} x^2\right] \times \text{HermiteH}[n, x]$$


(* Part b *)
Plot[
  (* We are directed to plot  $\psi_n(x)$  for  $n = 1, 2, 3$  *)
  {
     $\psi[1, x]$ ,
     $\psi[2, x]$ ,
     $\psi[3, x]$ 
  },

  (* Question asks us to plot the x range  $-5 \leq x \leq 5$  *)
  {x, -5, 5},

  (* Fix the y-axis range to  $-0.7 \leq y \leq 0.7$  *)
  PlotRange  $\rightarrow$  {-0.7, 0.7},

  (* Use a different colour for each of the three lines plotted *)
  PlotStyle  $\rightarrow$  {Red, Green, Blue},

  (* Label the axes *)
  AxesLabel  $\rightarrow$  {"x", " $\psi_n(x)$ "},

  (* Include the title "Wavefunctions  $\psi_n(x)$ " *)
  PlotLabel  $\rightarrow$  "Wavefunctions  $\psi_n(x)$ \n",

  (* Include a legend indicating clearly what each line *)
  (* represents placing the legend in the top right of *)
  (* the graph *)
  PlotLegends  $\rightarrow$  Placed[LineLegend["Expressions"], {1.0, 0.85}],

  (* Use LabelStyle to Set the overall font style *)
  LabelStyle  $\rightarrow$  {FontFamily  $\rightarrow$  "Times", FontColor  $\rightarrow$  Black, FontSize  $\rightarrow$  18},

  (* Finally we introduce the grid lines *)
  GridLines  $\rightarrow$  Automatic
]

```

■ 2. Primitive graphics animation. [9 marks]

- `Point[{x,y}]` is a Graphics primitive which represents a point at the position (x,y).

`PointSize[...]` is a function which sets the size of the point.

Write a piece of code which will represent, as an animation, the motion of a ball oscillating in a harmonic potential well, according to the following steps.

a) Plot $y = x^2$ over the range $-2 \leq x \leq 2$, with the y axis range limited to $-0.2 \leq y \leq 4.2$, to represent the potential well. Give the plot a frame and gridlines. Call this plot `potentialWell`.

b) Write a function `ballPosition[t_]`, returning the coordinates, $\{x,y\}$, of the ball at time t , as it oscillates in the potential well. The x coordinate evolves in time according to $x = 2 \cos(2t)$, with the corresponding y coordinate given by the shape of the well in which the ball oscillates.

c) Use `Show`, inside `Animate`, to combine the `potentialWell` plot together with a `Graphics` object made from a `Point`, whose position argument is given by `ballPosition[t]`. Consider which range of values for t will give you a complete cycle in your animation, and select an appropriate time-step.

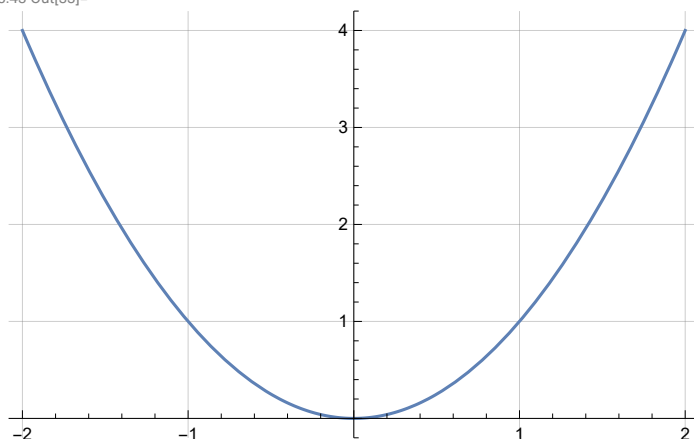
[Hint: you may need to check `Evaluation→Dynamic Updating Enabled` is selected in the drop-down menus.]

[9 marks]

3/6/23 11:45:43 In[33]:=

```
potentialWell =
  Plot[x^2, {x, -2, 2}, PlotRange → {-0.2, 4.2}, GridLines → Automatic]
(*for some reason, adding a frame stops the animation from working*)
```

3/6/23 11:45:43 Out[33]=



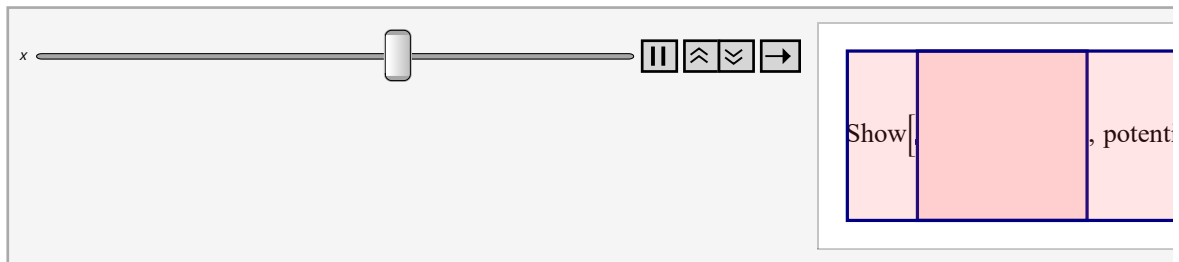
3/6/23 11:45:43 In[34]:=

```
ballPosition[t_] := {2 Cos[2 t], (2 Cos[2 t])^2}
(*parametric equation for the ball*)
```

3/6/23 11:45:43 In[35]:=

```
Animate[Show[Graphics[Point[ballPosition[x]]], potentialWell],
  {x, 0,  $\pi$ , 0.01}] (*show both the well and the ball inside of it*)
```

3/6/23 11:45:43 Out[35]:=



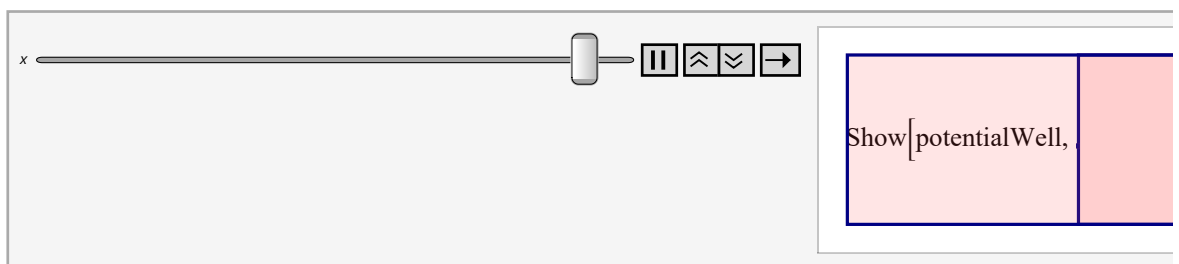
3/6/23 10:59:26 In[9]:=

- Question 2: 8/9
- Good work.
- One minor problem here is that you should have taken the hint to use `PointSize[...]` at the start of the question to make the moving ball bigger.
- Also if you had interchanged the order of the well and the ball inside `Show[...]` the gridlines and axes of the well would have become visible, as in the edited code I supply in the next input cell:

3/6/23 11:45:46 In[36]:=

```
(* Coordinator's edited version of your code here. *)
Animate[Show[potentialWell,
  Graphics[{PointSize[0.03], Point[ballPosition[x]]}], {x, 0,  $\pi$ , 0.01}]
(*show both the well and the ball inside of it*)
```

3/6/23 11:45:46 Out[36]:=



■ 3. Approximate root-finding using `Plot`. [5 marks]

- Use `Plot` to determine the value of x , between 0 and 10, to 3 significant figures, which solves the equation $\text{Log}[1/x] == \text{Tanh}[x]$ as follows. Make a `Plot` showing both $\text{Log}[1/x]$ and $\text{Tanh}[x]$, and then *manually* refine the x range (as given in the second argument of `Plot`), *repeatedly*, so as to zoom in on the point where the two lines intersect.

You need only display the final plot in the series of plots leading to your final answer. Your answer should be written in a text cell to 3 significant figures.

[5 marks]

Turns out I read the question wrong, I could've made my life much easier using just the

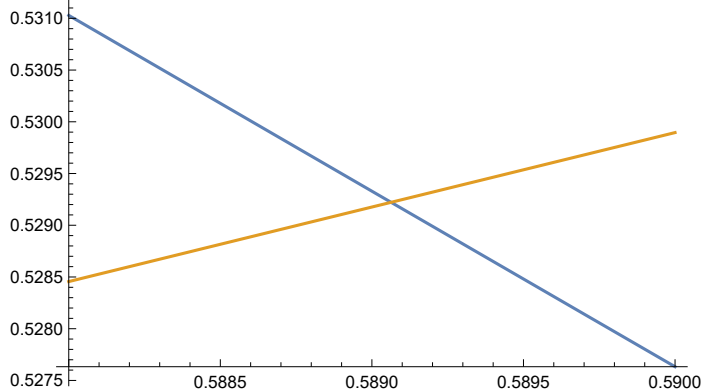
plot function

3/6/23 11:46:04 In[37]:=

```
Plot[{Log[1 / x], Tanh[x]}, {x, 0.588, 0.590}]
```

(*Adjust the range until intersection can be seen at a very precise reading*)

3/6/23 11:46:04 Out[37]:=



The answer is 0.589 to three significant figures.

3/6/23 10:59:26 In[11]:=

• Question 3: 5/5

• Correct.

■ 4. Primitive graphics animation revisited. [4 marks]

■ This question generalises question 2.

■ Define a potential $V[x_]$:= $\frac{34}{31} - x^2 + x^3 + x^4$.

■ a) Plot $V[x]$ over the range $-2.5 \leq x \leq 2.5$, limiting the y-axis range shown to $0 \leq y \leq 6$.

Give the plot a frame and gridlines.

Call this plot `q4Well`.

■ b) The equation of motion for a particle moving along the x-axis under the influence of this potential is given by

$$\frac{d^2}{dt^2} x[t] = -\frac{d}{dx} V[x[t]]$$

where $x[t]$ is the particle's position at time t .

Use `NDSolve` to solve the equation of motion for $x[t]$ subject to the boundary conditions

$$x[0] = -2$$

$$x'[0] = 0$$

for t in the range $0 < t < 20$.

■ c) Write a function `particle[t_]`, returning $\{x[t], V[x[t]]\}$, with $x[t]$ given by the solution to part b).

■ d) Use `Show`, inside `Animate`, to combine the `q4Well` plot together with a `Graphics` object made from a `Point`, whose position argument is given by

`particle[t]`. Adjust the animation rate until it is smooth using the `AnimationRate` option.

[Hint: depending on your approach, it may help to define `particle[t_]` with `Set`, i.e. `=`, rather than `SetDelayed`, i.e. `:=`.]

[4 marks : only *fully functioning, correct*, solutions for this question are eligible for marks.]

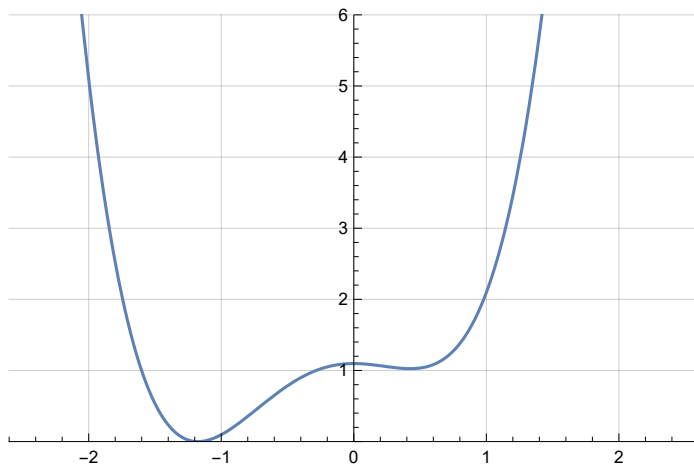
3/6/23 11:46:17 In[38]:=

```
V[x_] := 34 / 31 - x^2 + x^3 + x^4
```

3/6/23 11:46:17 In[39]:=

```
q4Well = Plot[V[x], {x, -2.5, 2.5}, PlotRange -> {0, 6}, GridLines -> Automatic]
(*Once again, the framed option stops my animation from working*)
```

3/6/23 11:46:17 Out[39]:=





3/6/23 11:46:17 In[40]:=

```
NDSolve[{x''[t] == -V'[x[t]], x[0] == -2, x'[0] == 0}, x[t], {t, 0, 20}]
```

3/6/23 11:46:17 Out[40]:=

```
{ {x(t) -> InterpolatingFunction[ Domain: ( 0. 20. ) Output: scalar ](t)} }
```

3/6/23 11:46:17 In[41]:=

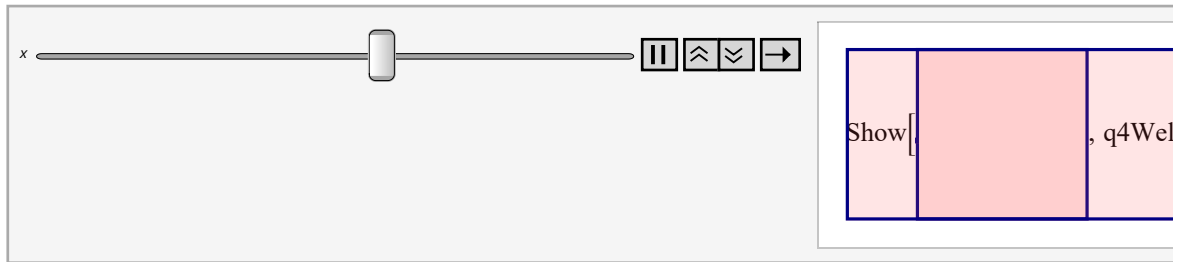
```
particle[t_] := {InterpolatingFunction[ Domain: {{0., 20.}} Output: scalar ] [t],
V[InterpolatingFunction[ Domain: {{0., 20.}} Output: scalar ] [t]]}
```

(*using set or setDelayed makes no difference, at least for my simulation*)

3/6/23 11:46:17 In[42]:=

```
Animate[Show[Graphics[Point[particle[x]]], q4Well], {x, 0,  $\pi$ , 0.01}]
```

3/6/23 11:46:17 Out[42]=



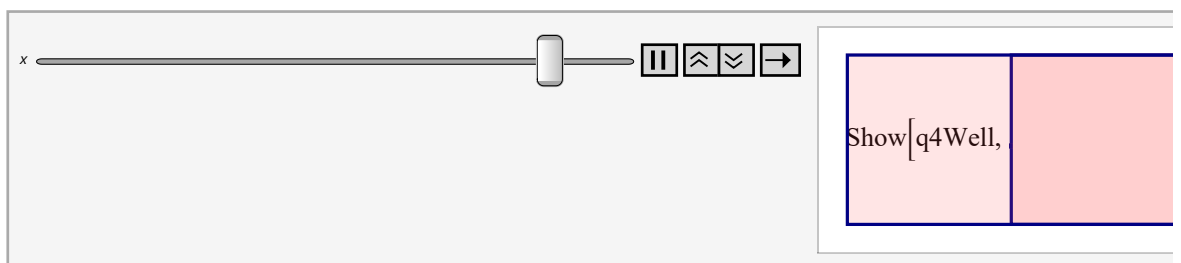
- Question 4: 2/4
- Same issue as in Q2 : the moving point/ball is obviously too small/too hard to see.
- Also if you had interchanged the order of the well and the ball inside Show[...] the gridlines and axes of the well would have become visible, as in the edited code I supply in the next input cell:

3/6/23 11:47:13 In[43]:=

(* Coordinator's edited version of your code: *)

```
Animate[Show[q4Well, Graphics[{PointSize[0.03], Point[particle[x]]}], {x, 0,  $\pi$ , 0.01}]
```

3/6/23 11:47:13 Out[43]=



■ Total marks available: 32

- Solutions are due by 1200 noon on Thursday February 23rd [here](#): allow time for uploading on moodle.
- A 10% mark deduction will be made (3 marks) if the template isn't used.
- Name your solution notebook file in the format **WK5_HMWK_<Initials>_<Family Name>.nb**, e.g. **WK5_HMWK_K_Hamilton.nb**
- Make a *backup copy* of your solutions.
- Delete all cell evaluation output by selecting **Cell → Delete All Output** from the drop-down menus at the top of the screen, then save and upload *that* file to Moodle.
- The first thing your marker will do when they receive your notebook is to evaluate all of it, to regenerate the output, by clicking **Evaluation → Evaluate Notebook** from the drop-down menus at the top of the screen. *It is your responsibility to check that carrying out this process with a fresh kernel will produce the output you intend it to, before you upload your work.*

K. Hamilton

J. Bhamrah, J. Underwood, A. Harker

Last revision 13:33 07 Feb 2023