MATH 210 Assignment 5

Approximating Solutions of Definite Integrals and ODEs with SciPy

INSTRUCTIONS

- Create a new Jupyter notebook, set the kernel to Python 3, present your solutions in the notebook and clearly label the solutions
- There are 4 questions and each is worth 5 points for 20 total points
- Submit the `.ipynb` file to Connect by **11pm Friday March 4**
- You may work on these problems with others but you must write your solutions on your own

QUESTIONS

1. Let $f(x)$ be an integrable function over an interval $[a,b]$ and consider the definite integral

$$I = \int_a^b f(x)\,dx \ .$$

Recall, we can approximate the integral I by choosing a partition $a = x_0 < x_1 < \cdots < x_n = b$ of the interval $[a,b]$ and computing the Riemann sum

$$I = \int_a^b f(x)\,dx \approx \sum_{i=1}^{n} f(x_i^*)(x_i - x_{i-1})$$

where, for each i, x_i^* is a point in the interval $[x_{i-1}, x_i]$. If $x_i^* = x_{i-1}$, then we say the Riemann sum uses *left endpoints*; if $x_i^* = x_i$, then we say the Riemann sum uses *right endpoints*; and if $x_i^* = (x_i + x_{i-1})/2$, then we say the Riemann sum uses *midpoints*.

Define a function called `riemann_sum` with parameters `y`, `x` and `method` (in that order and with default value `method='left'`) where `y` and `x` are arrays and `method` is a string, and which returns the corresponding Riemann sum using left endpoints or right endpoints as determined by `method`. (In other words, if `method='right'` then `riemann_sum` uses right endpoints, and if `method='left'` then `riemann_sum` uses left endpoints. The arrays `x` and `y` define values of a function $f(x)$ over an interval $[a,b]$ where the first entry of `x` is a and the last is b, and the Riemann sum approximates $\int_a^b f(x)\,dx$.

2. Use the function `scipy.integrate.quad` to approximate the left side of the integral equation

$$\int_0^1 \frac{x \arcsin x}{1+x^2}\,dx = \frac{\pi}{2} \ln\left(\frac{2\sqrt{2}}{1+\sqrt{2}}\right)$$

and show that the approximation is within the error estimate output by `scipy.integrate.quad` as compared to the right side. In other words, compute the approximation of the integral using `quad` and then compute the (absolute value of the) difference between the approximation and the right side of the equation and show that the difference is less than the error estimate (which is the second value in the tuple returned by `quad`).

3. Define a function called `fresnel` with parameters `alpha`, `b` and `trig_fun` (in that order and with default values `alpha=1`, `b=5` and `trig_fun='sin'`) which does the following:

   ◇ if `trig_fun='sin'`, plot the Fresnel sine function

   $$S_\alpha(t) = \int_0^t \sin(\alpha x^2)\, dx$$

   for $t \in [0, \texttt{b}]$ and where $\alpha = \texttt{alpha}$

   ◇ if `trig_fun='cos'`, plot the Fresnel cosine function

   $$C_\alpha(t) = \int_0^t \cos(\alpha x^2)\, dx$$

   for $t \in [0, \texttt{b}]$ and where $\alpha = \texttt{alpha}$

   For example:

   ◇ `fresnel()` plots $S_1(t)$ for $t \in [0, 5]$

   ◇ `fresnel(2,3)` plots $S_2(t)$ for $t \in [0, 3]$

   ◇ `fresnel(3,trig_fun='cos')` plots the function $C_3(t)$ for $t \in [0, 5]$

   ◇ `fresnel(1,2,'cos')` plots the function $C_1(t)$ for $t \in [0, 2]$

   (Hint: $S_\alpha(t)$ is the solution of the differential equation $y' = \sin(\alpha t^2)$ with $y(0) = 0$ and similarly for $C_\alpha(t)$.)

4. Define a function called `trig_ode` with parameters `alpha`, `beta`, `y0` and `tf` (in that order and with default values `alpha=1`, `beta=1`, `y0=0` and `tf=10`), and which plots the solution of the equation

   $$y' = \sin(\alpha y) + \cos(\beta t)\ ,\ y(0) = \texttt{y0}$$

   over the interval $[0, t_f]$ where $t_f = \texttt{tf}$, and where $\alpha = \texttt{alpha}$ and $\beta = \texttt{beta}$.