MA386 Project 2 - A *New* Quadrature Method

We'll use special functions called *spherical basis functions* (SBFs) to help integrate periodic functions. You may use `MATLAB` or `Mathematica` to complete this project.

In Section 4.3 we approximated definite integrals of the form $\int_a^b f(t)\,dt$ by first interpolating $f(t)$ with a polynomial and then integrating the interpolant. In this project you will do something similar, but we'll be using a different type of interpolant generated by a *spherical basis function* given by

$$\phi(t) = e^{-(1-\cos(t))^2}.$$

In the example below, you will see how to add *shifts* of this function together in a way that interpolates a given function.

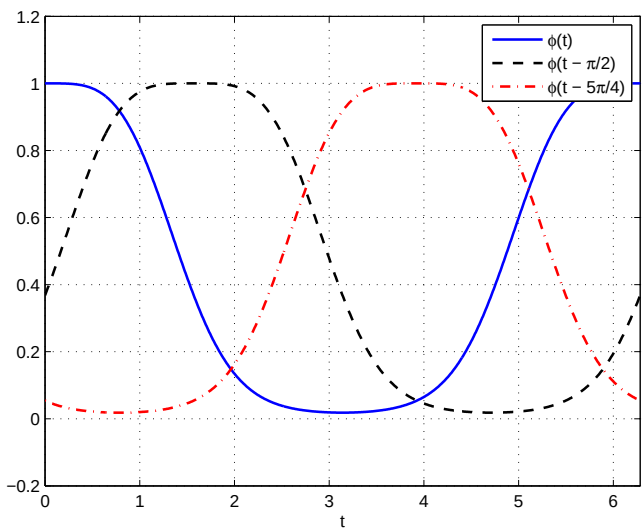

Figure 1: A graph of the spherical basis function $\phi(t)$ and a few of its shifts.

1. **Example: Finding an interpolant.** Consider the function $\phi(t) = e^{-(1-\cos(t))^2}$ and the two *nodes* (or *sample points*) $t_1 = 0$ and $t_2 = 4$. Each node is associated with a *basis* function, which we get by shifting $\phi(t)$:

$$\begin{aligned} \phi_1(t) &= \phi(t - t_1) = e^{-(1-\cos(t-0))^2} \\ \phi_2(t) &= \phi(t - t_2) = e^{-(1-\cos(t-4))^2} \end{aligned}$$

One can use these to interpolate functions at t_1 and t_2.

For example, here is how we can interpolate the function $f(t) = \cos(t) + (\sin(t))^2 + 0.5$. We look for weight c_1 and c_2 so that $c_1\phi_1(t) + c_2\phi_2(t)$ interpolates $f(t)$ at $t = t_1$ and $t = t_2$. In other words, we need to solve the following equations:

$$\begin{aligned} c_1\phi_1(t_1) + c_2\phi_2(t_1) = f(t_1) \quad &\Rightarrow \quad c_1 \cdot 1 + c_2 \cdot 0.064924 = 1.5 \\ c_1\phi_1(t_2) + c_2\phi_2(t_2) = f(t_2) \quad &\Rightarrow \quad c_1 \cdot 0.064924 + c_2 \cdot 1 = 0.419106 \end{aligned}$$

Solving this system gives $c_1 = 1.47902$ and $c_2 = 0.323082$. With our weights solved for, we define the interpolant to $f(t)$ at the sample points $T = t_1, t_2$ to be

$$s_{f,T}(t) = 1.47902\phi_1(t) + 0.323082\phi_2(t).$$

Below is a plot of $f(t)$ and $s_{f,T}(t)$ on the same graph. Note that the functions agree at t_1 and t_2, as expected.

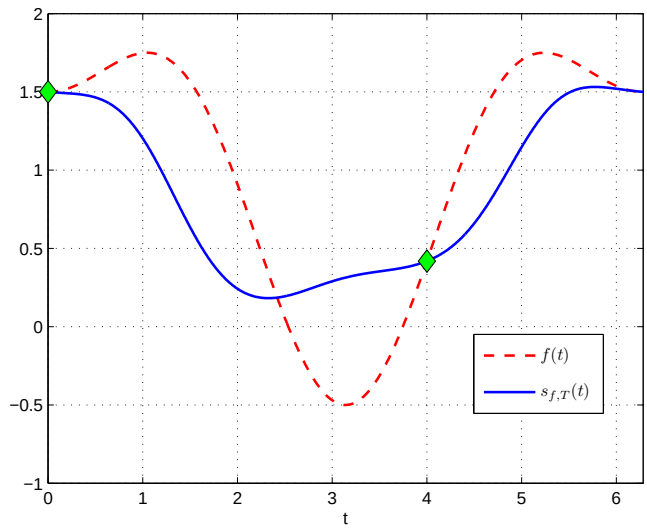

Figure 2: A graph of the target function $f(t)$ and its SBF interpolant $s_{f,T}(t)$.

2. **Example: Approximating $\int_0^{2\pi} f(t)\,dt$.** Consider the example above. We will approximate the integral by integrating the interpolant we found.

$$\begin{aligned}\int_0^{2\pi} f(t)\,dt &\approx \int_0^{2\pi} s_{f,T}(t)\,dt = \int_0^{2\pi} c_1\phi_1(t) + c_2\phi_2(t)\,dt \\ &= 1.47902\int_0^{2\pi}\phi_1(t)\,dt + 0.323082\int_0^{2\pi}\phi_2(t)\,dt \\ &= 1.47902(2.90513) + 0.323082(2.90513) = 5.23534.\end{aligned}$$

The real value for $\int_0^{2\pi} f(t)\,dt$ is about 2π. If had used more data points, perhaps we would have a better approximation.

3. The function $\phi(t)$ has the amazing property that no matter how many points you have, an interpolant built from shifts of $\phi(t)$ *always* exists. Given n points $T = \{t_1, t_2, \ldots, t_n\}$ and a target function $f(t)$, we look for an interpolant of the form

$$s_{f,T}(t) = c_1\phi_1(t) + c_2\phi_2(t) + \cdots + c_n\phi_n(t),$$

where $\phi_j(t)$ is given by shifting $\phi(t)$:

$$\phi_j(t) = \phi(t - t_j) = e^{-(1-\cos(t-t_j))^2}.$$

In order to find the weights, use the interpolation conditions to get n equations, i.e.,

$$s_{f,T}(t_j) = c_1\phi_1(t_j) + c_2\phi_2(t_j) + \cdots + c_n\phi_n(t_j) = f(t_j),$$

where $j = 1, 2, \ldots, n$. Solving these equations will give us the c_j's.

To approximate $\int_0^{2\pi} f(t)\,dt$, we just integrate $s_{f,T}(t)$.

4. Let's see how well this method works. Let $f(t) = \cos(t) + (\sin(t))^2 + 0.5$. Consider the following node sets:

   (a) $T = \left\{0, \frac{2\pi}{3}, \frac{4\pi}{3}\right\}$.

   (b) $T = \left\{0, \frac{2\pi}{5}, \frac{4\pi}{5}, \frac{6\pi}{5}, \frac{8\pi}{5}\right\}$.

   (c) $T = \left\{0, \frac{2\pi}{7}, \frac{4\pi}{7}, \frac{6\pi}{7}, \frac{8\pi}{7}, \frac{10\pi}{7}, \frac{12\pi}{7}\right\}$.

   For each of these sets,

   - Find the interpolant $s_{f,T}(t)$.
   - Graph $f(t)$ and $s_{f,T}(t)$ on the same graph.
   - Calculate $\int_0^{2\pi} s_{f,T}(t)\,dt$.
   - Calculate the error $\left|\int_0^{2\pi} s_{f,T}(t)\,dt - \int_0^{2\pi} f(t)\,dt\right|$.

5. Now do the same thing, but with the target function $g(t) = |\cos(t)|$. In other words, for each of the node sets above,

   - Find the interpolant $s_{g,T}(t)$.
   - Graph $g(t)$ and $s_{g,T}(t)$ on the same graph.
   - Calculate $\int_0^{2\pi} s_{g,T}(t)\,dt$.
   - Calculate the error $\left|\int_0^{2\pi} s_{g,T}(t)\,dt - \int_0^{2\pi} g(t)\,dt\right|$.

6. Discuss the outcome of your experiments.[1] For each of the target functions, what can you say about the approximation rate of the method, i.e. how does the error seem to depend on the space between the nodes? Include "log log" plots and "big $\mathcal{O}$" notation in your analysis. Given the choice between Newton-Cotes quadrature, the composite Simpson's method, or this integration method, which would you choose and why?

7. Finding the coefficients of the inerpolant amounts to solving a system of the form $A\mathbf{c} = \mathbf{d}$, where $\mathbf{c}$ is a vector containing the coefficients, $\mathbf{d}$ is the vector of data obtained from our target function, and A is a matrix whose (i, j)th entry is given by

   $$A_{i,j} = \phi(t_i - t_j).$$

   The matrix A is one of the "special" matrices from Section 6.6. Which one is it, and what methods could you use from Section 6.6 to make solving $A\mathbf{c} = \mathbf{d}$ easier?

[1] Bonus: Get crazy and use this method to integrate a discontinuous function. What happens as you add more and more nodes? Can you explain what is happening?

8. Include all of the above in a typed, professionally written report; by-hand work will not be accepted. Include any computer code you use in an appendix. Put any ideas you find into your own words, and be sure to document your sources. If you have *any* questions at all, let me know immediately. Good luck!