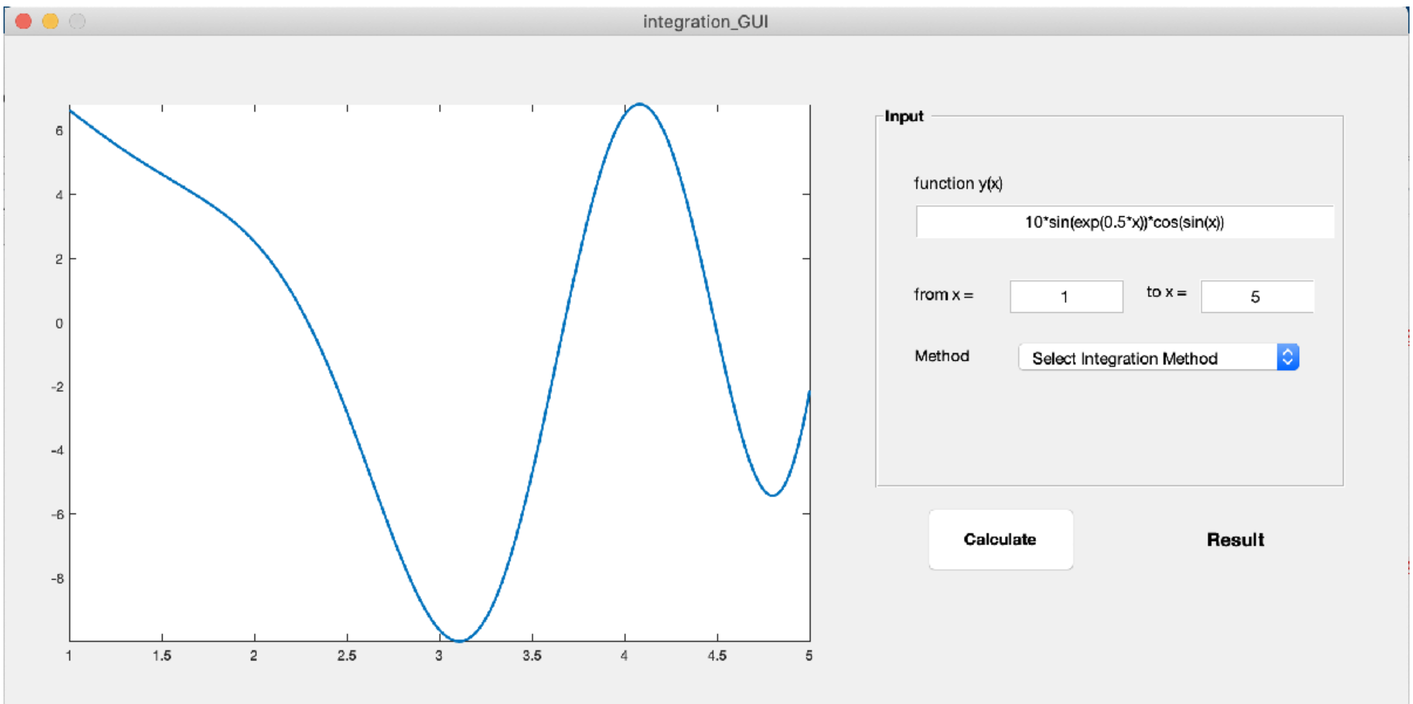
ACCOMPLISH: Numerical Integration

Dr. Zheng Chen, Mathematics Department, University of Massachusetts Dartmouth

Goal:

The goal of this module is to finish building a graphical user interface (GUI) MATLAB to approximate integrations. Here is how the GUI looks:



The user will input the formula of the integrand function, the range of integration, and choose a numerical integration method to approximate the integration. The framework of this GUI is provided to students in the file "**integration_GUI.m**". The full version is provided to the advisor, and the students will work on the student version with blank parts of codes.

The student's mission is to study different numerical integration methods and finish the subroutines/functions for each methods. The blank parts are under the routine "***function Calculate_pushbutton_Callback(hObject, eventdata, handles)***". More details will be shown later under each method.

```
% --- Executes on button press in Calculate_pushbutton.
function Calculate_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Calculate_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

x1 = str2double(get(handles.fromx,'String'));
x2 = str2double(get(handles.tox,'String'));
```

There are many numerical integration methods. In this GUI, the following methods are included:

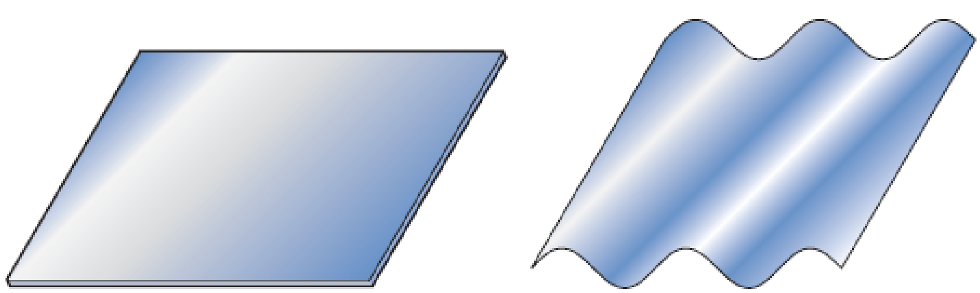
1. **Midpoint Rule**
2. **Trapezoidal Rule**
3. **Simpon's Rule**
4. **Simpson's 3/8 Rule**
5. **Composite Numerical Integration**
6. **Monte Carlo Integration**

The GUI has potential to be expanded to include other methods as well.

Table of Contents

1. Motivation of numerical integration

Here is a sheet of corrugated roofing, which is constructed by pressing a flat sheet of aluminum into one whose cross section has the form of a sine wave:



If such a corruated sheet 4ft long is needed, the height of each wave is 1inch from the center line, and each wave has a period of approximately 2π in.

Question: How long is the original flat sheet?

$$L = \int_0^{48} \sqrt{1 + ((\sin(x))')^2}\, dx = \int_0^{48} \sqrt{1 + (\cos(x))^2}\, dx.$$

Question: How to approximate this integration numerically (i.e. using computer to compute)?

2. Some Basic Numerical Integration Methods

Elements of Numerical Integration

Goal Evaluating the definite integral of a function that has no explicit antiderivative or whose antiderivative is not easy to obtain.

e.g. $\int_0^1 \cos(x^2)dx, \qquad \int_0^2 e^{-x^2}dx.$

Numerical Integration - numerical quadrature

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i)$$

where $\{x_i\}$ are called **nodes**, and $\{a_i\}$ are called **weights**.

The integral is approximated by a weighted sum of the selected function's point values. Different choices of the nodes and weights yield different numerical integration methods.

2.1 Midpoint Rule

The midpoint rule is to approximate the region under the graph of the function $y = f(x)$ as a rectangle that has the hight of the point value on the middle point.

$$\int_a^b f(x)dx \simeq h * f\left(\frac{a+b}{2}\right),$$

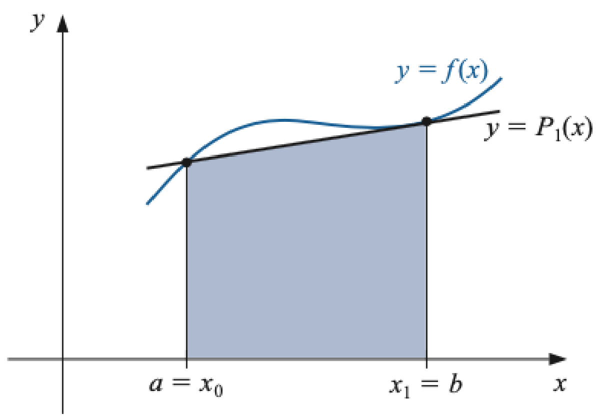
where $h = b - a$ is the length of interval of integration.

What to do?

```
%% Midpoint Rule
if method == 3

    % Code here to calculate the approximation of integration by Midpoint
    % Rule and assign the result to variable "integral"

    set(handles.integration_result,'String',integral);
end
```

2.2 Trapezoidal Rule

The trapezoidal rule works by approximating the region under the graph of the function $y = f(x)$ as a trapezoid and calculating its area.



$$\int_a^b f(x)dx \simeq \frac{h}{2}[f(a) + f(b)],$$

where $h = b - a$.

What to do?

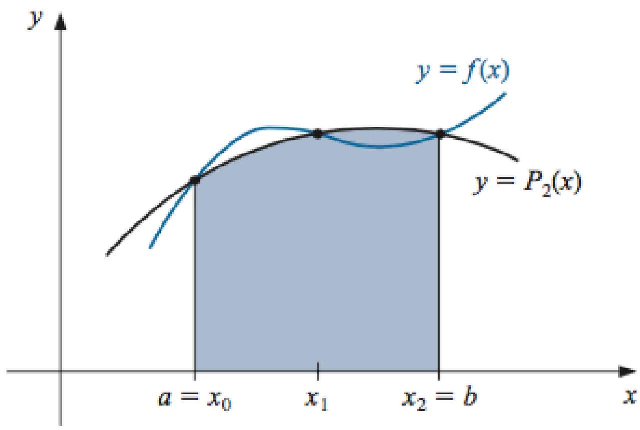
```
%% Trapezoidal Rule
if method == 4

    % Code here to calculate the approximation of integration by Trapezoidal
    % Rule and assign the result to variable "integral"

    set(handles.integration_result,'String',integral);
end
```

2.3 Simpson’s Rule

$$\int_a^b f(x)dx \simeq \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)],$$

where $h = x_1 - x_0 = x_2 - x_1$.

It is exact when applied to any polynomial of degree three or less.



What to do?

```
%% Simpsons
if method == 5

    % Code here to calculate the approximation of integration by Simpsons
    % Rule and assign the result to variable "integral"

    set(handles.integration_result,'String',integral);
end
```

2.4 Simpson's 3/8 Rule

$$\int_a^b f(x)dx \simeq \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)],$$

where $h = x_1 - x_0 = x_2 - x_1 = x_3 - x_2$, and $x_0 = a, x_3 = b$.

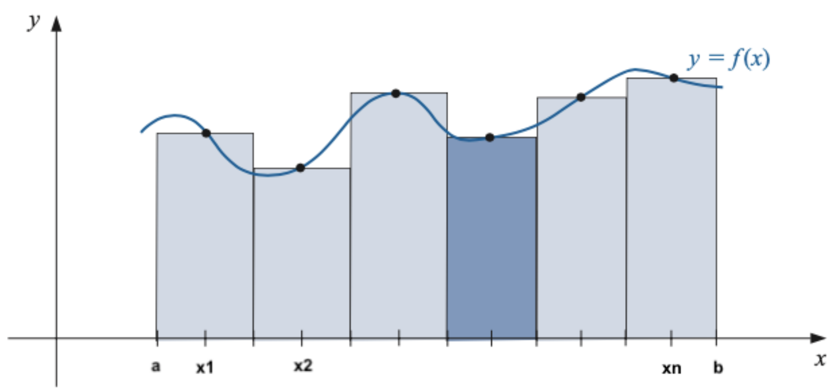
What to do?

```
%% Simpsons 3/8
if method == 6

    % Code here to calculate the approximation of integration by Simpsons
    % 3/8 Rule and assign the result to variable "integral"

    set(handles.integration_result,'String',integral);
end
```

3 Composite Numerical Integration

Idea a piecewise approach: divide the integration interval into small subintervals, and apply the above simple formulas on subintervals.

3.1 Composite Midpoint rule



The Composite Midpoint rule applies Midpoint rule for each subinterval and sum them up.

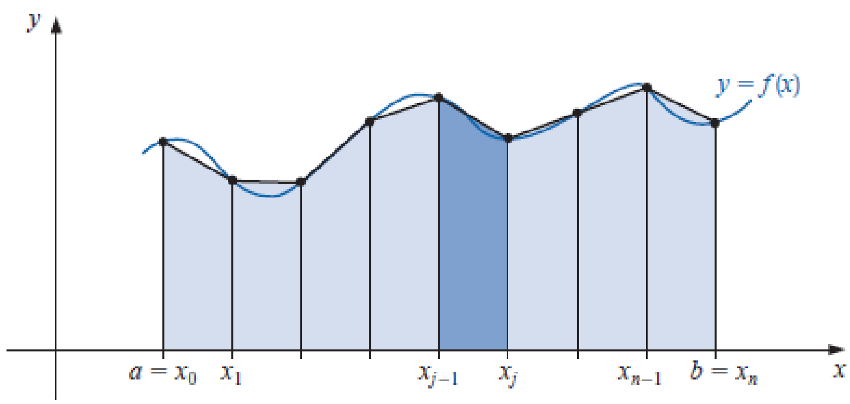
$$\int_a^b f(x)dx \simeq h \sum_{j=1}^{n} f(x_j),$$

where $x_j = a + (j + 1/2)h$ are midpoints for subintervales and $h = \dfrac{b-a}{n}$ is the length of the subintervals.

What to do?

```
%% Composite Midpoint Rule
if method == 7
    N = str2double(get(handles.edit_trap,'String')); % number of subintervals

    % Code here to calculate the approximation of integration by Composite
    % Midpoint Rule and assign the result to variable "integral"

    set(handles.integration_result,'String',integral);
end
```

3.2 Composite Trapezoidal Rule



The Trapezoidal rule is applied for each subinterval and then gets summed up.

$$\int_a^b f(x)dx \simeq \frac{h}{2}[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)],$$

where $h = \frac{b-a}{n}$ is the length of subintervals.

The doubled weight for interior points x_j $(1 \leq j \leq n-1)$ comes from the double counting of these point value from both left and right subintervals $[x_{j-1}, x_j]$ and $[x_j, x_{j+1}]$.

What to do?

```
%% Composite Trapezoidal Rule
if method == 8
   N = str2double(get(handles.edit_trap,'String')); % number of subintervals

    % Code here to calculate the approximation of integration by Composite
    % Trapezoidal Rule and assign the result to variable "integral"

   set(handles.integration_result,'String',integral);
end
```

4. Monte Carlo integration

In mathematics, **Monte Carlo integration** is a technique for numerical integration using random numbers. It is a particular Monte Carlo method that numerically computes a definite integral. While other algorithms usually evaluate the integrand at a regular grid, Monte Carlo randomly chooses points at which the integrand is evaluated. This method is particularly useful for higher-dimensional integrals.

There are many ways to choose points randomly. For more details, we refer to [https://cs.dartmouth.edu/wjarosz/publications/dissertation/appendixA.pdf.](https://cs.dartmouth.edu/wjarosz/publications/dissertation/appendixA.pdf)

The GUI frame provided has this option and the students are not required to code this part up. Just have fun!