UNIVERSITY OF LEICESTER

Computational Project

Student name: *Junbiao Li*

Course: *MA2252 Introduction to Computing*
Due date: *December 29, 2022*

DECLARATION
All sentences or passages quoted in this Project Report from other people's work have been specifically acknowledged by clear and specific cross referencing to author, work and page(s), or website link. I understand that failure to do so amounts to plagiarism and will be considered grounds for failure in this module and the degree as a whole.
Name: Junbiao Li
Signed: Junbiao Li
Date: Dec. 3rd, 2022

1. Topic 1

Answer. To get the second-order accurate forward, backward and centered finite difference schemes for finding the derivative.[3][2] We first write down the Taylor series expansions of $f(x+h)$, $f(x-h)$, $f(x+2h)$, $f(x-2h)$, $f(x+3h)$ and $f(x-3h)$

$$\begin{aligned}
f(x+h) &= f(x) + h\frac{f'(x)}{1!} + h^2\frac{f''(x)}{2!} \\
f(x-h) &= f(x) - h\frac{f'(x)}{1!} + h^2\frac{f''(x)}{2!} \\
f(x+2h) &= f(x) + 2h\frac{f'(x)}{1!} + 4h^2\frac{f''(x)}{2!} \\
f(x-2h) &= f(x) - 2h\frac{f'(x)}{1!} + 4h^2\frac{f''(x)}{2!} \\
f(x+3h) &= f(x) + 3h\frac{f'(x)}{1!} + 9h^2\frac{f''(x)}{2!} \\
f(x-3h) &= f(x) - 3h\frac{f'(x)}{1!} + 9h^2\frac{f''(x)}{2!}
\end{aligned} \tag{1.1}$$

For the second-order accurate forward difference, we want to find out $f'(x) = \frac{1}{h}(a_0 f(x) + a_1 f(x+h) + a_2 f(x+2h) + \mathcal{O}(h^3))$ where a_0, a_1, a_2 are all constant. Hence, we have

$$\begin{aligned}
\frac{f'(x)}{1!} = f'(x) = \frac{1}{h}(a_0 f(x) + \\
& a_1 f(x) + a_1 f'(x)h + a_1 f''(x)h^2 \\
& a_2 f(x) + 2a_2 f'(x)h + 4a_2 f''(x)h^2 + \mathcal{O}(h^3))
\end{aligned} \tag{1.2}$$

Which is equivalent to solving a linear system

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 1 & 4 & 0 \end{array}\right) \tag{1.3}$$

Hence, $a_0 = -\frac{3}{2}$, $a_1 = 2$ and $a_3 = -\frac{1}{2}$, and

$$f'(x) = \frac{-\frac{3}{2}f(x) + 2f(x+h) - \frac{1}{2}f(x+2h)}{h} + \mathcal{O}(h^2) \tag{1.4}$$

With the same process, We can calculate second-order accurate forward, backward and centered finite difference schemes for the first derivative and second derivative:

1. Second order accurate centered difference approximations:

$$\begin{aligned} f'(x) &:(f(x+h) - f(x-h))/(2h) \\ f''(x) &:(f(x+h) - 2f(x) + f(x-h))/h^2 \end{aligned} \tag{1.5}$$

2. Second order accurate forward difference approximations:

$$\begin{aligned} f'(x) &:(-3f(x) + 4f(x+h) - f(x+2h))/(2h) \\ f''(x) &:(2f(x) - 5f(x+h) + 4f(x+2h) - f(x+3h))/h^2 \end{aligned} \tag{1.6}$$

3. Second order accurate backward difference approximations:

$$\begin{aligned} f'(x) &:(3f(x) - 4f(x-h) + f(x-2h))/(2h) \\ f''(x) &:(2f(x) - 5f(x-h) + 4f(x-2h) - f(x-3h))/h^2 \end{aligned} \tag{1.7}$$

Implement. I chose $x^4 + \sin(x)$ as test function, with $0 \leq x \leq 2$ and $h = 0.1$

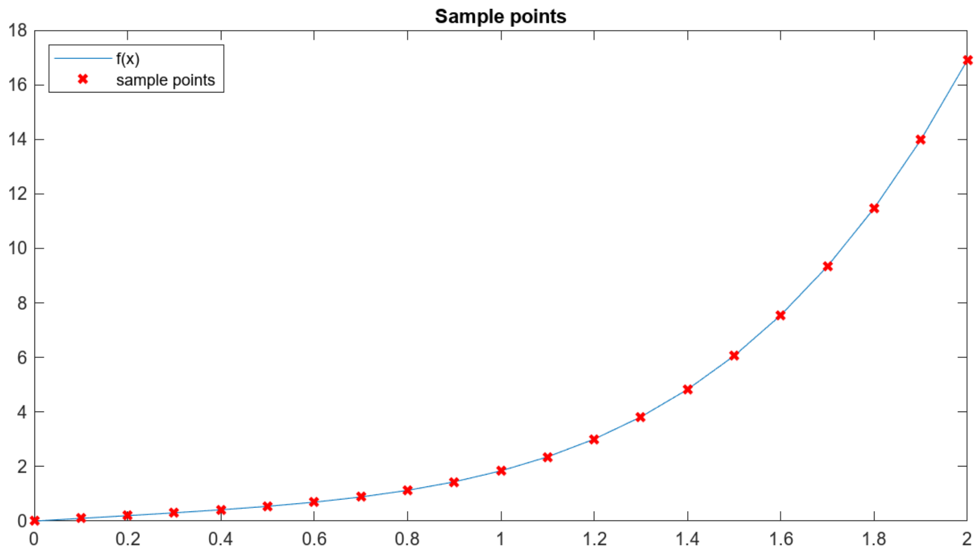

Figure 1.1: Sample Points of f(x)

After applying second-order accurate difference approximations, I draw $f'(x)$ and $f''(x)$ to check the difference between a real derivative and finite difference schemes.

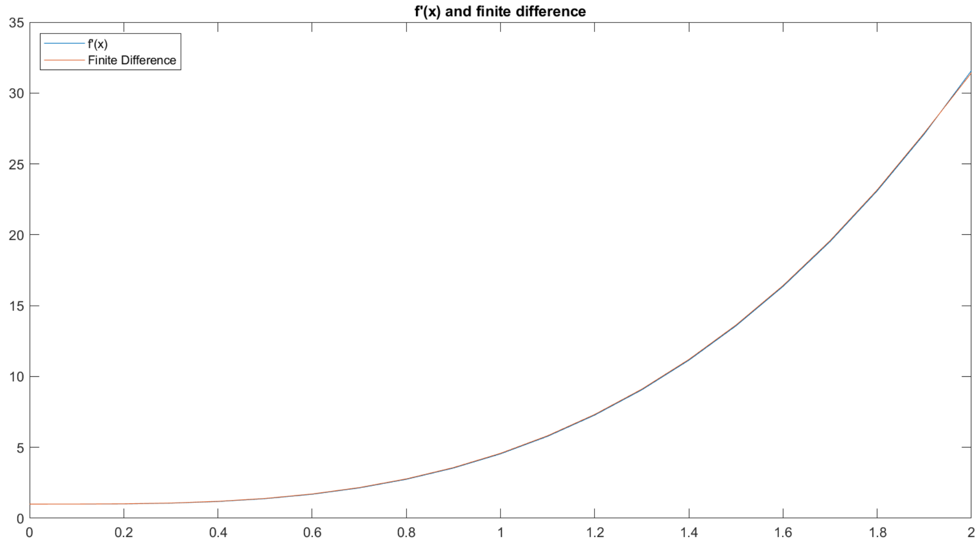

Figure 1.2: Compare real first derivative and finite difference

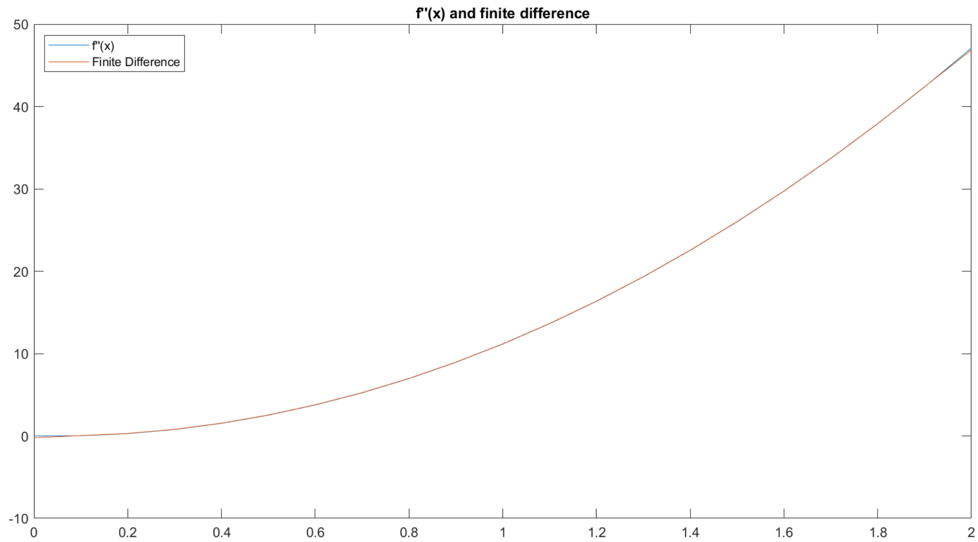

Figure 1.3: Compare real second derivative and finite difference

Figure.1.2 and Figure.1.3 told us that the second-order accurate difference approximations are close to the real derivative, even their actual difference. And I compared the deviations of the difference approximations with the actual difference. It can be found that both the first-order derivatives Figure.1.4 and the second-order 1.5 derivatives are better than the forward and backward differential approximations, which is why we need to use the central differential approximation as much as possible and only use the forward and backward differential approximations at the endpoints. You can find more implementation detail in the code.

2. Topic 2

Answer. To evaluate double integrals which are performed on the rectangle $R = \{(x, y) \in \mathbb{R}^2 : a \leq x \leq b \text{ and } c \leq y \leq d\}$, We first consider the uniform-grid in each dimension. Let grid space of x and y be the h, which means we have $x_1, x_2, \ldots, x_{n_x}$ and $y_1, y_2, \ldots, y_{n_y}$.[1] [2] For integrals in one dimension, we could start with something

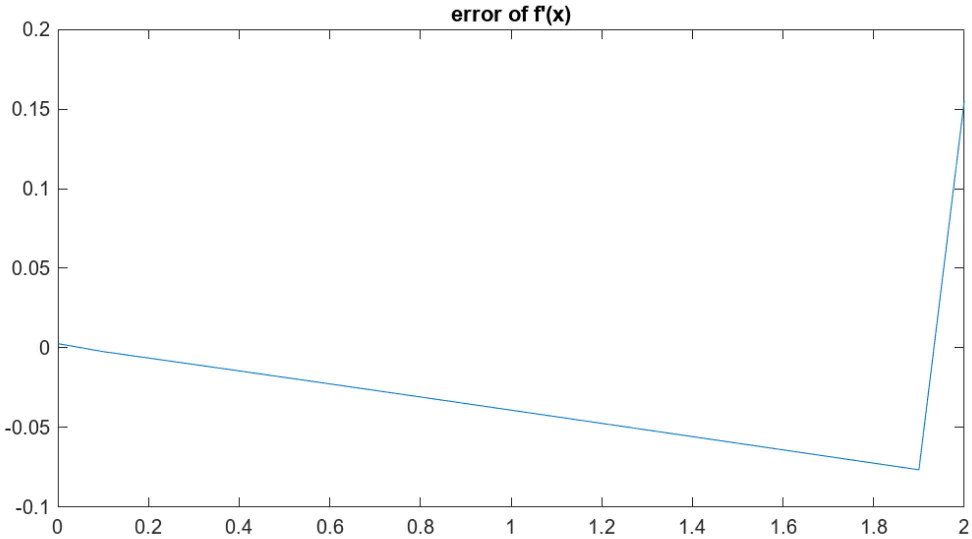

Figure 1.4: errors of f′(x)

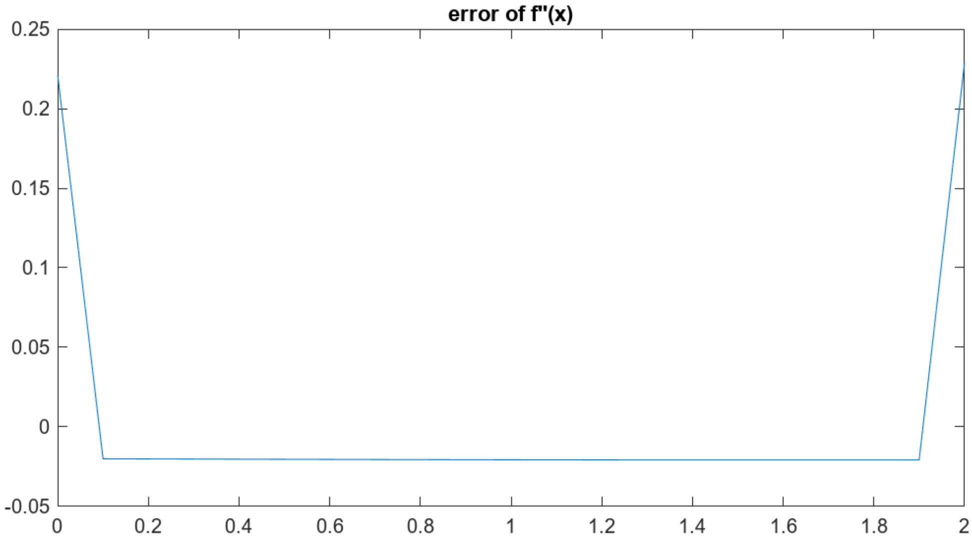

Figure 1.5: errors of f″(x)

simple like the trapezoidal rule

$$\int_a^b f(x)dx \approx \frac{h}{2}[f(a) + f(b) + 2\sum_{i=2}^{n-1} f(x_i)] \tag{2.1}$$

We can now extend this definition to the two-dimensional case, i.e

$$A_{2D} = \int_d^c \int_b^a f(x,y)\,dxdy = \int_d^c g(y)\,dy \tag{2.2}$$

where,

$$g(y) = \int_a^b f(x,y)\,dx \tag{2.3}$$

Task 1 Now apply the trapezoidal rule to calculate the integral to x with fixed y.

$$\begin{aligned} g(y) &= \int_a^b f(x,y)\,dx \\ &= \sum_{i=1}^{n_x-1} \frac{1}{2}[f(x_i)+f(x_{i+1})]h \\ &= \frac{h}{2}[f(x_0,y)+f(x_{n_x},y)+2\sum_{i=2}^{n_x-1} f(x_i,y)] \\ &= \frac{h}{2}[f(a,y)+f(b,y)+2\sum_{i=2}^{n_x-1} f(x_i,y)] \end{aligned} \tag{2.4}$$

Finally we get $g(y) = \frac{h}{2}[f(a,y)+f(b,y)+2\sum_{i=2}^{n_x-1} f(x_i,y)]$

Task 2 After integrate x, we now integrate y,

$$\begin{aligned} I &= \int_c^d g(y)\,dy \\ &= \sum_{i=1}^{n_y-1} \frac{1}{2}[g(y_i)+g(y_{i+1})]h \\ &= \frac{h}{2}[g(c)+g(d)+2\sum_{j=2}^{n_y-1} g(y_j)] \end{aligned} \tag{2.5}$$

Substituting eq.(2.4) into eq.(2.5) we have

$$\begin{aligned} I = \frac{h}{2}\{&\frac{h}{2}[f(a,c)+f(b,c)+2\sum_{i=2}^{n_x-1} f(x_i,c)] \\ &+\frac{h}{2}[f(a,d)+f(b,d)+2\sum_{i=2}^{n_x-1} f(x_i,d)] \\ &+\frac{h}{2}2\sum_{j=2}^{n_y-1}[f(a,y_j)+f(b,y_j)+2\sum_{i=2}^{n_x-1} f(x_i,y_j)]\} \end{aligned} \tag{2.6}$$

Now we can simplify eq.(2.6) as much as possible

$$\begin{aligned} I = (\frac{h}{2})^2[&f(a,c)+f(b,c)+f(a,d)+f(b,d) \\ &+2\sum_{i=2}^{n_x-1}[f(x_i,c)+f(x_i,d)] \\ &+2\sum_{j=2}^{n_y-1}[f(a,y_j)+f(b,y_j)] \\ &+4\sum_{j=2}^{n_y-1}\sum_{i=2}^{n_x-1} f(x_i,y_j)] \end{aligned} \tag{2.7}$$

Implement. Here is the implemented code with Matlab. I used a for-loop to implement the above algorithm. *my_integral*2 function takes 5 parameters:

- **f**: the function
- **a, b**: the start and end points of x,
- **c, d**: the start and end points of y,
- **h**: the length of the partition (for comparing computational accuracy)

For the four summation symbols in the formula, I use four for-loops to traverse all the elements and sum them and then get the integral result.

```
function I=my_integral2(f, a, b, c, d, h)
  x=a:h:b;
  y=c:h:d;

  n_x=length(x);
  n_y=length(y);
  I=f(a, c)+f(b,c)+f(a,d)+f(b,d);


  for i=2:n_x-1
    I = I + 2*(f(x(i),c)+f(x(i),d));
  end


  for j=2:n_y-1
    I = I + 2*(f(a, y(j))+f(b, y(j)));
  end


  for i=2:n_x-1
    for j=2:n_y-1
      I = I + 4*f(x(i), y(j));
    end
  end

  I = h^2/4*I;
end
```

After doing some experiments, I found that the finer the segmentation and the larger the rectangle, the higher the accuracy can be obtained. You can see more detailed data in the following two tables (Table 2.1) (Table 2.2). Here I use the formula $\frac{|my\_integral2-real|}{|real|}$ to calculate the error.

Table 2.1: Accuracy of difference h with function $y^4 + \sin(x)$

	a=0;b=1;c=2;d=4;h=0.1	a=0;b=1;c=2;d=4;h=0.5
my_integral2	199.505289	203.962661
real	199.319395	199.319395
error	**0.0933%**	**2.3296%**

Table 2.2: Accuracy of difference R with function $y^4 + \sin(x)$

	a=0;b=1;c=2;d=4;h=0.1	a=0;b=1;c=0;d=10;h=0.1
my_integral2	199.505289	20007.926445
real	199.319395	20004.596977
error	**0.0933%**	**0.0166%**

References

[1] David Keffer. Numerical techniques for the evaluation of multi-dimensional integral equations. http://utkstair.org/clausius/docs/che505/pdf/IE_eval_N-Dints.pdf, September 1999.

[2] Abdon Atangana Kolade M. Owolabi. *Numerical Methods for Fractional Differentiation*. Springer Singapore, 2019.

[3] Alice C. Yew. Numerical differentiation: finite differences. https://www.dam.brown.edu/people/alcyew/handouts/numdiff.pdf, Spring 2011.