Foundations of Data Science № 2

Xianjin Wu, U202215258, Huazhong University of Science and Technology 12/18/2023

Problem 1

Consider the financial function defined over the interval $x \in [1, 5]$by

$$F(x) = 2x^2 \sin(x) + 0.8x^3.$$

0.1 Question a

Compare various numerical differentiation methods for calculating the rate of change of the financial function (i.e., the gradient) with step sizes $\Delta x = 0.02, \Delta x = 0.2, and \Delta x = 0.5$. Utilize forward difference, backward difference, and central difference. Plot the curves of $F(x)$ and the financial function gradients obtained using these methods in a figure, and include it in your report (Note: Ensure proper labeling of the x- and y-axes and include a legend in your plot).

0.2 Result a

$$\text{Forward Difference:}(f, x, \Delta x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$
$$\text{Backward Difference:}(f, x, \Delta x) = \frac{f(x) - f(x - \Delta x)}{\Delta x}$$
$$\text{Central Difference:}(f, x, \Delta x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2 \cdot \Delta x}$$

The code is as follows

Listing 1: MATLAB Code

```matlab
% Define the financial function
F = @(x) 2 * x.^2 .* sin(x) + 0.8 * x.^3;

% Define numerical differentiation methods
forward_difference = @(f, x, delta_x) (f(x + delta_x) - f(x)) / delta_x;
backward_difference = @(f, x, delta_x) (f(x) - f(x - delta_x)) / delta_x;
central_difference = @(f, x, delta_x) (f(x + delta_x) - f(x - delta_x)) / ↩
    (2 * delta_x);

% Define the interval [1, 5]
x_values = linspace(1, 5, 100);

% Define different step sizes
delta_x_values = [0.02, 0.2, 0.5];
```

```matlab

% Plot for each delta_x
for i = 1:length(delta_x_values)
    delta_x = delta_x_values(i);

    % Calculate gradients
    gradient_forward = forward_difference(F, x_values, delta_x);
    gradient_backward = backward_difference(F, x_values, delta_x);
    gradient_central = central_difference(F, x_values, delta_x);

    % Plot financial function
    figure;
    plot(x_values, F(x_values), 'LineWidth', 2, 'DisplayName', 'F(x)');
    hold on;

    % Plot gradients
    plot(x_values, gradient_forward, '--', 'DisplayName', ['Forward, \Deltax=', num2str(delta_x)]);
    plot(x_values, gradient_backward, '--', 'DisplayName', ['Backward, \Deltax=', num2str(delta_x)]);
    plot(x_values, gradient_central, '--', 'DisplayName', ['Central, \Deltax=', num2str(delta_x)]);

    xlabel('x');
    ylabel('Function Value / Gradient');
    title(['Financial Function and Numerical Gradients (\Deltax=', num2str(delta_x), ')']);
    legend();
    grid on;

    hold off;
end
```

0.3 Question b

The derivative of the financial function gradient is referred to as the financial flux. Derive the mathematical formula in your report for computing the financial flux from $F(x)$(Hint: Start with the Taylor series expansion, paying extra attention to the computation at the first and last points). Implement your algorithm in MATLAB to compute the financial flux.

0.4 Result b

To derive the mathematical formula, I start from the first-order Taylor series expansion,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + O(h^3)$$

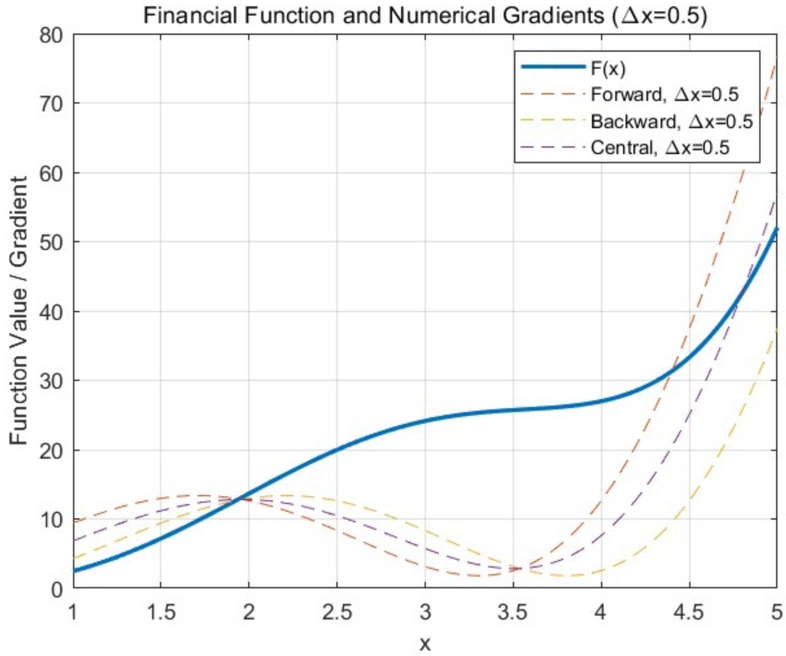

Figure 1: $\Delta x = 0.5$, the image of three different difference methods.

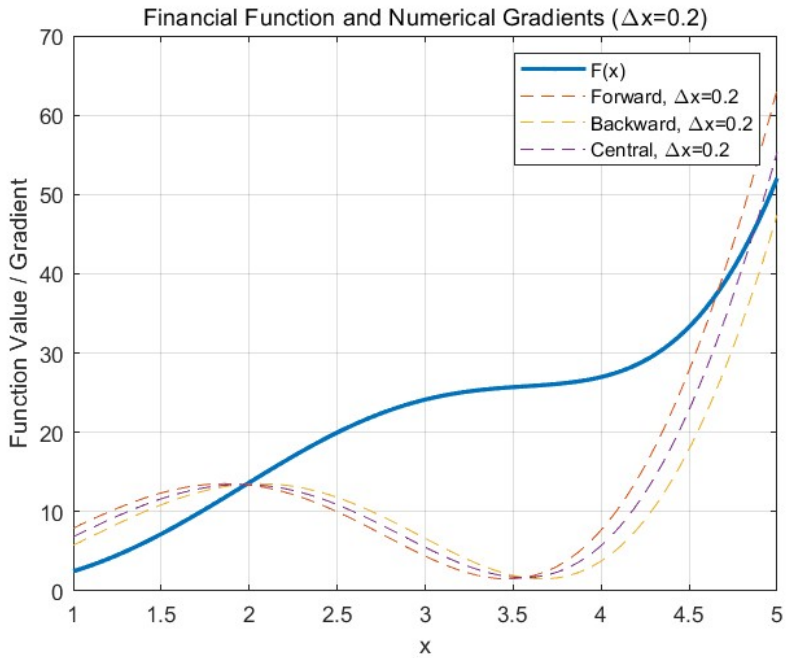

Figure 2: $\Delta x = 0.2$, the image of three different difference methods.

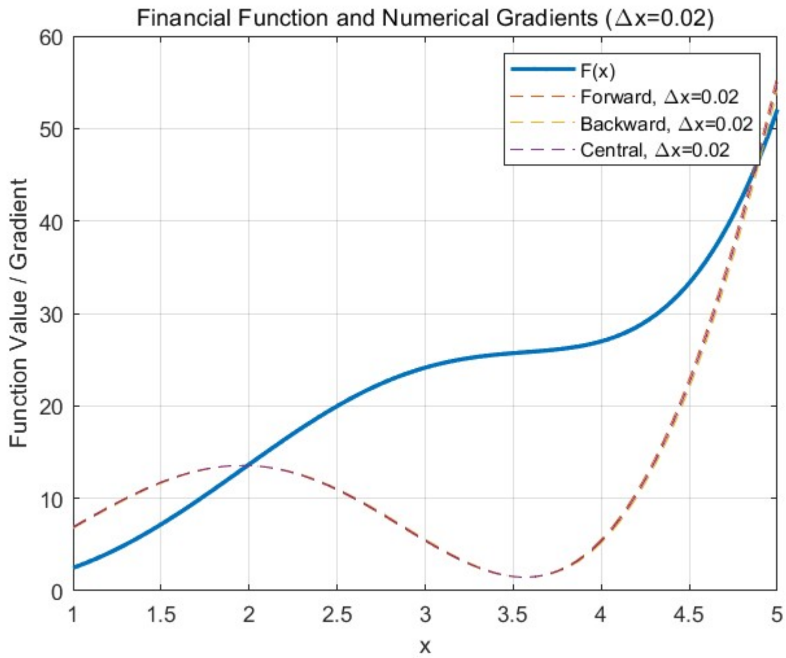

Figure 3: $\Delta x = 0.02$, the image of three different difference methods.

Subtracting $f(x-h)$from both sides:

$$f(x+h) - f(x-h) = 2hf'(x) + h^2 f''(x) + O(h^3)$$

Dividing both sides by $2h$ to get the central difference for the first derivative:

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h}{2} f''(x) + O(h^2)$$

Performing central difference again for the second derivative:

$$\frac{f'(x+h) - f'(x-h)}{2h} = f''(x) + O(h)$$

Substituting x with x_0 for the second derivative:

$$\frac{f''(x_0+h) - f''(x_0-h)}{2h} = f''(x_0) + O(h)$$

For the start and end points, I use a one-sided difference to compute the financial flux. The code is as follows

Listing 2: MATLAB Code

```
% Define the financial function
F = @(x) 2 * x.^2 .* sin(x) + 0.8 * x.^3;
Gradient = @(x) 4 * x .* sin(x) + 2 * x.^2 .* cos(x) + 2.4 * x.^2;

% Define numerical differentiation methods with consideration for ↩
    endpoints
one_sided_difference_start = @(f, x, delta_x) (f(x + delta_x) - f(x)) / ↩
    delta_x;
```

```
one_sided_difference_end = @(f, x, delta_x) (f(x) - f(x - delta_x)) / delta_x;

% Define numerical differentiation method (central difference)
central_difference = @(f, x, delta_x) (f(x + delta_x) - f(x - delta_x)) / (2 * delta_x);

% Define the interval [1, 5] and step size
x_values = linspace(1, 5, 100);
delta_x = 0.01; % Choose a small delta_x for accuracy

% Compute financial flux using one-sided difference at the start
financial_flux_start = one_sided_difference_start(@(x) Gradient(x), x_values(1), delta_x);

% Compute financial flux using one-sided difference at the end
financial_flux_end = one_sided_difference_end(@(x) Gradient(x), x_values(end), delta_x);

% Compute financial flux using central difference for the interior points
financial_flux_interior = central_difference(@(x) Gradient(x), x_values(2:end-1), delta_x);

% Concatenate all parts of financial flux
financial_flux = [financial_flux_start, financial_flux_interior, financial_flux_end];

% Plot financial function and financial flux
figure;
yyaxis left;
plot(x_values, Gradient(x_values), 'LineWidth', 2, 'DisplayName', 'Gradient(x)');
ylabel('Flux Value');

yyaxis right;
plot(x_values, financial_flux, '--', 'LineWidth', 2, 'DisplayName', 'f''(x)');
ylabel('Financial Flux');

xlabel('x');
title('Gradient Function and Financial Flux');
legend();
grid on;
```

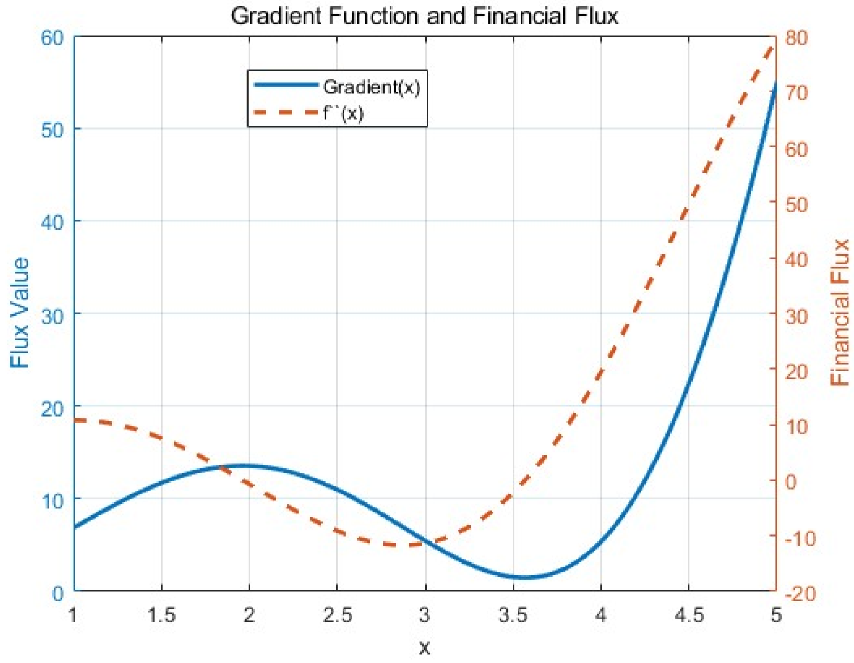

Figure 4: Gradient and financial flux

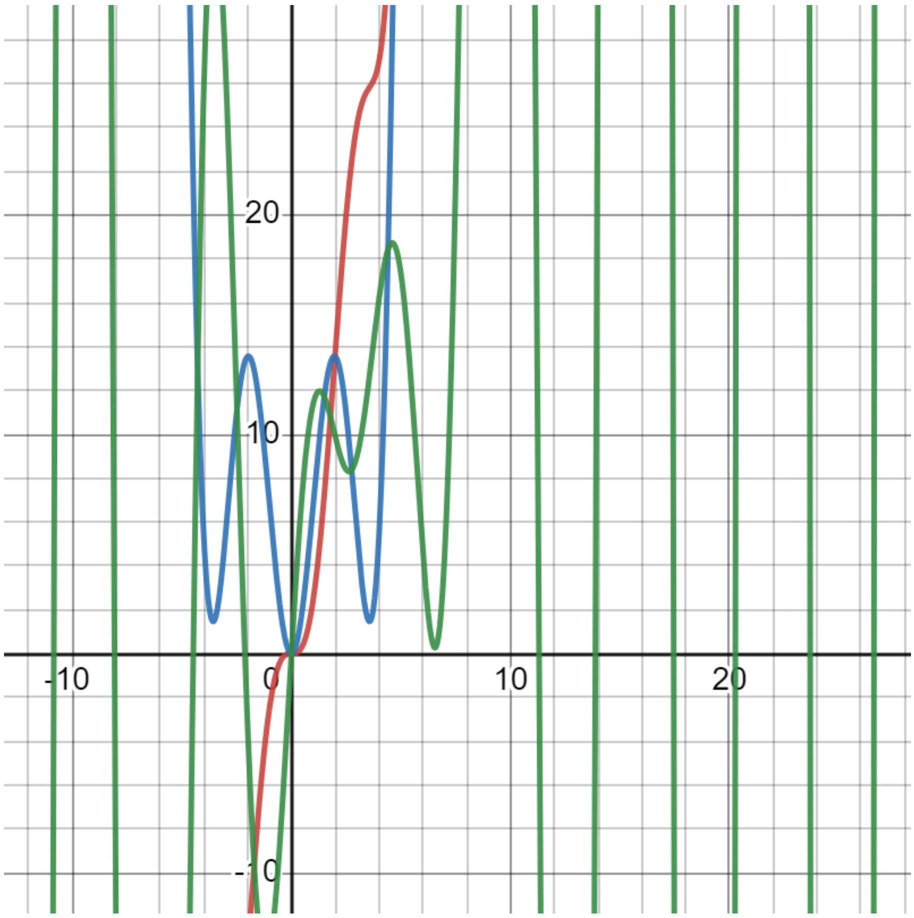

Figure 5: desmos-graph, Green: financial flux, Blue: Gradient, Red:$F(x)$