

第二次上机作业

Anonymity 3220100000*

* College of Control Science and Engineering, Robotics Engineering

July 8, 2024

1 问题描述

编写非线性方程求根的不动点迭代算法程序：

从不同的初始点开始，分别采用以下三种迭代方式求解方程 $f(x) = x^2 - 3 = 0$ 的正根，记录迭代过程并说明原因。

(1) $\varphi(x) = x^2 - 3 + x$

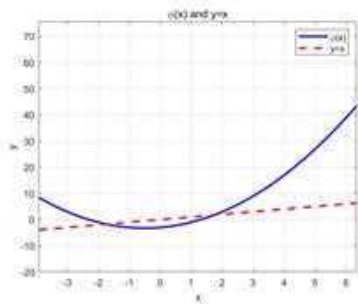
(2) $\varphi(x) = x - \frac{x^2-3}{4}$

(3) $\varphi(x) = \frac{1}{2} \left(x + \frac{3}{x} \right)$

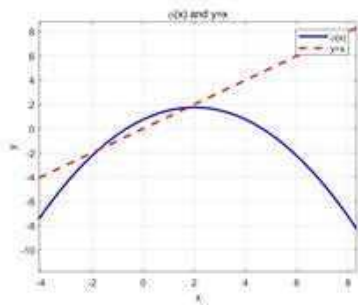
2 思路分析

首先 matlab 绘制三种方法的对应函数图像示意图：

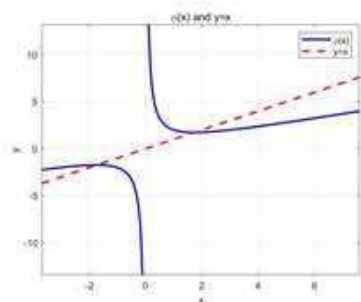
$\varphi(x) = x^2 - 3 + x$ 与 $f(x) = x$



$\varphi(x) = x - \frac{x^2-3}{4}$ 与 $f(x) = x$



$$\varphi(x) = \frac{1}{2} \left(x + \frac{3}{x} \right) \text{ 与 } f(x) = x$$



根据不动点定理：

假设 (i) $g, g' \in C[a, b]$;

(ii) 当 $x \in [a, b]$, 有 $g(x) \in [a, b]$;

(iii) $p_0 \in (a, b)$;

(iv) K 是一个正常数;

如果对于所有 $x \in [a, b]$, 有 $|g'(x)| \leq K < 1$, 则迭代 $p_n = g(p_{n-1})$ 将收敛到唯一的不动点 $P \in [a, b]$ 。在这种情况下, P 称为吸引不动点。

如果对于所有 $x \in [a, b]$, 有 $|g'(x)| > 1$, 则迭代 $p_n = g(p_{n-1})$ 将不会收敛到 P 。在这种情况下, P 称为排斥不动点, 而且迭代显示出局部发散性。

因此对于问题, 采取不同的初始值会得到不同结果。同时, 由于方程存在负根, 因此可能最终得到的是负根而非要求所求的正根。

总而言之, 初始值的选择对于能否得到答案至关重要。

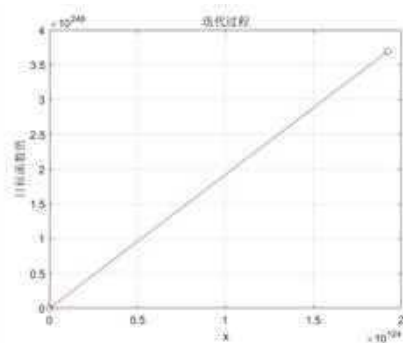
3 运行结果与分析

使用不动点迭代法, 初始值分别选择 1,2,3,4,5,6, 得到的三个函数的运算结果分别如下图:

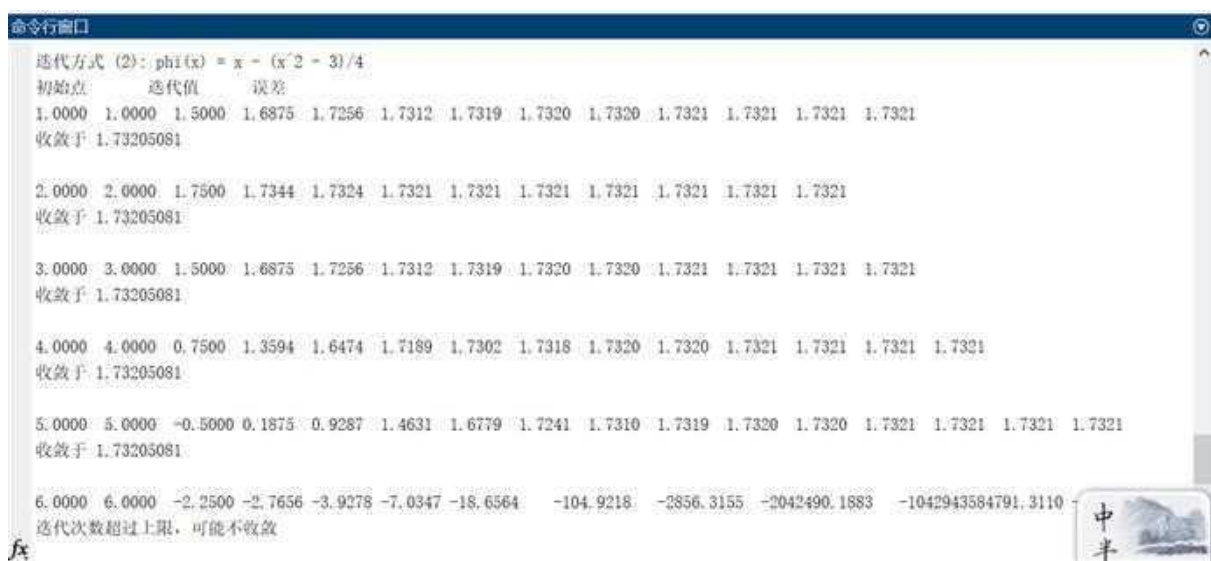
$$\varphi(x) = x^2 - 3 + x$$



迭代不收敛示意图：



$$\varphi(x) = x - \frac{x^2 - 3}{4}$$



方程 2 在初始值去 1-6 时在迭代 10 次时的示意图 (6 不收敛):

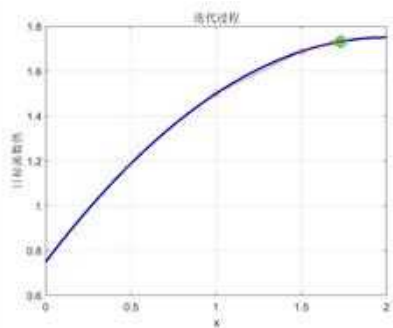


Figure 1: 初始值为 1

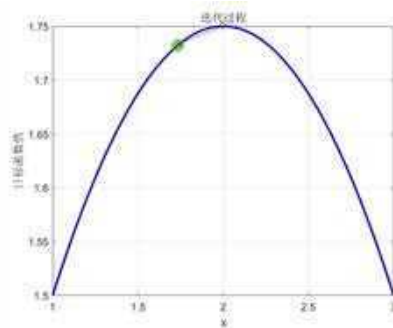


Figure 2: 初始值为 2

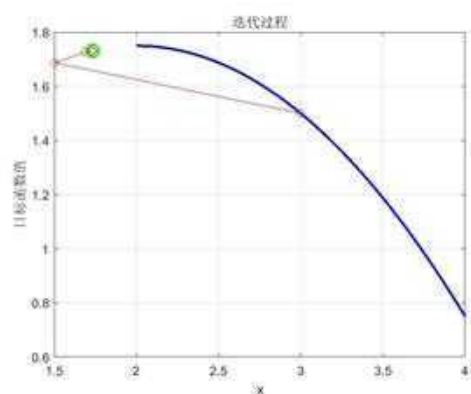


Figure 3: 初始值为 3

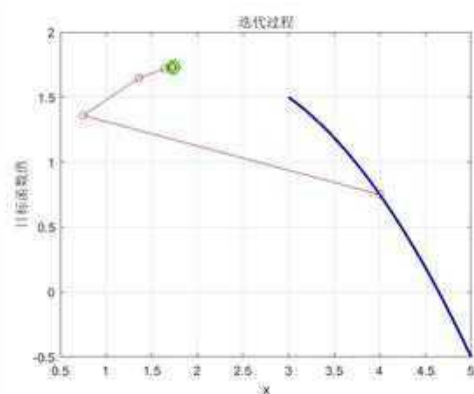


Figure 4: 初始值为 4

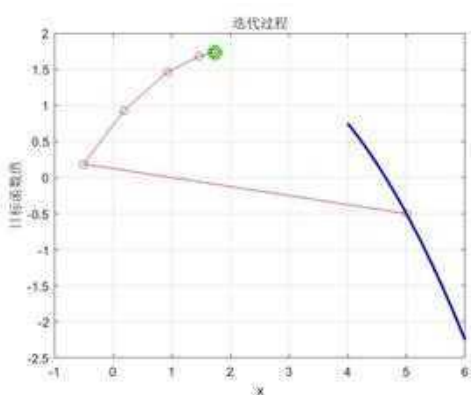


Figure 5: 初始值为 5

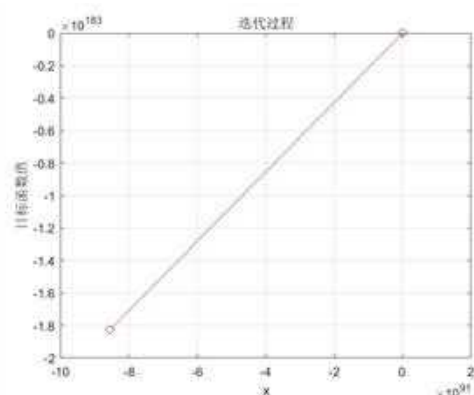
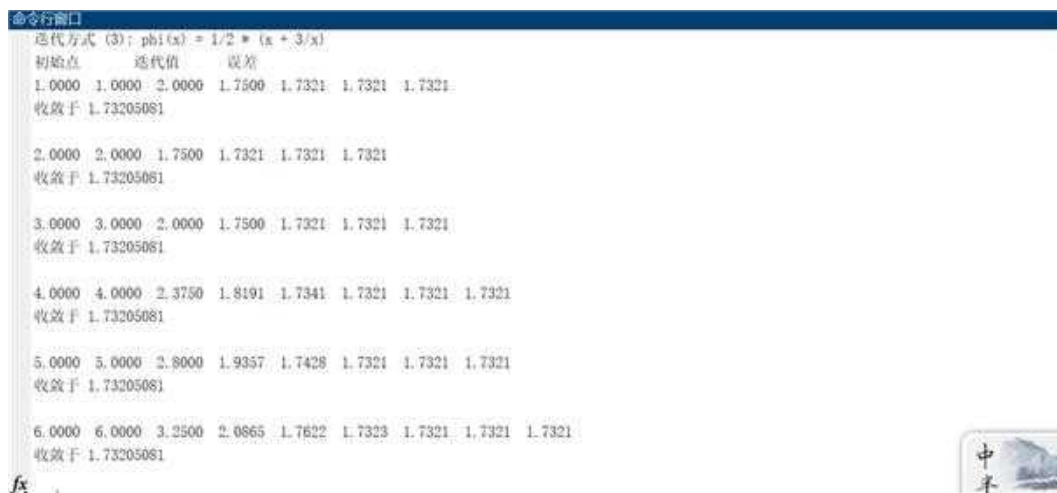


Figure 6: 初始值为 6

$$\varphi(x) = \frac{1}{2} \left(x + \frac{3}{x} \right)$$



方程三无论初始值如何选取，始终收敛，在 x=5 和 x=6 时迭代示意图如下：


```

1 % 定义函数
2 phi = @(x) 1/2 * (x + 3./x); % 使用 ./ 表示逐元素的除法
3 y = @(x) x;
4
5 % 生成 x 的值
6 x_values = linspace(-5, 10, 1000000); % 选择合适的范围和点数
7
8 % 计算函数值
9 phi_values = phi(x_values);
10 y_values = y(x_values);
11
12 % 绘制图形
13 figure;
14 plot(x_values, phi_values, 'b-', 'LineWidth', 2, 'DisplayName', '\phi(
    x)');
15 hold on;
16 plot(x_values, y_values, 'r--', 'LineWidth', 2, 'DisplayName', 'y=x');
17
18 % 添加标签和图例
19 xlabel('x');
20 ylabel('y');
21 title('\phi(x) and y=x');
22 legend('show');
23
24 % 设置纵坐标范围为-10到10
25 ylim([-10, 10]);
26
27 % 显示网格
28 grid on;
29
30 % 显示图形
31 hold off;

```

5.2 不动点求法函数代码

```

1 function [root, iter] = iterate(phi, x0, max_iter, epsilon)
2     iter = 0;
3     root = x0;
4
5     fprintf('%0.4f\t', root);

```

```

6
7 % Initialize vectors for visualization
8 x_vals = zeros(1, max_iter + 1);
9 y_vals = zeros(1, max_iter + 1);
10 x_vals(1) = root;
11 y_vals(1) = phi(root);
12
13 % Plot the target function
14 x_range = linspace(min(x0) - 1, max(x0) + 1, 100);
15 y_range = arrayfun(phi, x_range); % Use arrayfun for element-wise
    operation
16
17 figure;
18 plot(x_range, y_range, '-b', 'LineWidth', 2);
19 hold on;
20
21 while iter < max_iter
22     x1 = phi(root);
23     fprintf('%0.4f\t', x1);
24
25     % Update visualization vectors
26     x_vals(iter + 2) = x1;
27     y_vals(iter + 2) = phi(x1);
28
29     % Plot the iteration points for this iteration
30     plot(x_vals(iter + 1:iter + 2), y_vals(iter + 1:iter + 2), '-o
        ', 'Color', [1, 0, 0]);
31
32     if abs(x1 - root) < epsilon
33         fprintf('\n');
34         fprintf('收敛于 %0.8f\n', x1);
35
36     % Highlight the converged point
37     plot(x1, phi(x1), 'Marker', 'o', 'MarkerSize', 10, '
        MarkerEdgeColor', 'g', 'LineWidth', 2);
38     title('迭代过程');
39     xlabel('x');
40     ylabel('目标函数值');
41     grid on;
42     hold off;
43     return;

```

```

44         end
45
46         root = x1;
47         iter = iter + 1;
48     end
49
50     fprintf('\n');
51     fprintf('迭代次数超过上限，可能不收敛\n');
52
53     % Plot the iteration points if the maximum iterations are reached
54     plot(x_vals, y_vals, '-o', 'Color', [1, 0, 0]);
55     title('迭代过程');
56     xlabel('x');
57     ylabel('目标函数值');
58     grid on;
59     hold off;
60 end

```

5.3 不同初始值求根代码

```

1  % 定义函数
2  phi = @(x) 1/2 * (x + 3./x); % 使用 ./ 表示逐元素的除法
3  y = @(x) x;
4
5  % 生成 x 的值
6  x_values = linspace(-5, 10, 1000000); % 选择合适的范围和点数
7
8  % 计算函数值
9  phi_values = phi(x_values);
10 y_values = y(x_values);
11
12 % 绘制图形
13 figure;
14 plot(x_values, phi_values, 'b-', 'LineWidth', 2, 'DisplayName', '\phi(x)');
15 hold on;
16 plot(x_values, y_values, 'r--', 'LineWidth', 2, 'DisplayName', 'y=x');
17
18 % 添加标签和图例
19 xlabel('x');
20 ylabel('y');

```



```

21 title('\phi(x) and y=x');
22 legend('show');
23
24 % 设置纵坐标范围为-10到10
25 ylim([-10, 10]);
26
27 % 显示网格
28 grid on;
29
30 % 显示图形
31 hold off;

```

5.4 寻找敛散变化点

```

1 % 初始值范围
2 x_range_start = 5.73;
3 x_range_end = 5.74;
4
5 % 步长
6 step_size = 0.0001;
7
8 % 迭代次数上限
9 max_iter = 100;
10
11 % 误差容限
12 epsilon = 1e-8;
13
14
15 % 第二种迭代方式:  $\phi(x) = x - (x^2 - 3)/4$ 
16 fprintf('\n迭代方式 (2):  $\phi(x) = x - (x^2 - 3)/4$ \n');
17 fprintf('初始值范围 (%.2f 到 %.2f)\n', x_range_start, x_range_end);
18 fprintf('初始值\t\t 是否收敛\n');
19
20 for x0 = x_range_start:step_size:x_range_end
21     fprintf('%.4f\t', x0);
22     [root, iter] = iterate(@(x) x - (x^2 - 3)/4, x0, max_iter, epsilon);
23
24     if iter >= max_iter
25         fprintf('不收敛\n');
26     else

```

```
27         fprintf('收敛于 %.8f\n', root);
28     end
29 end
```