实验报告

第1题

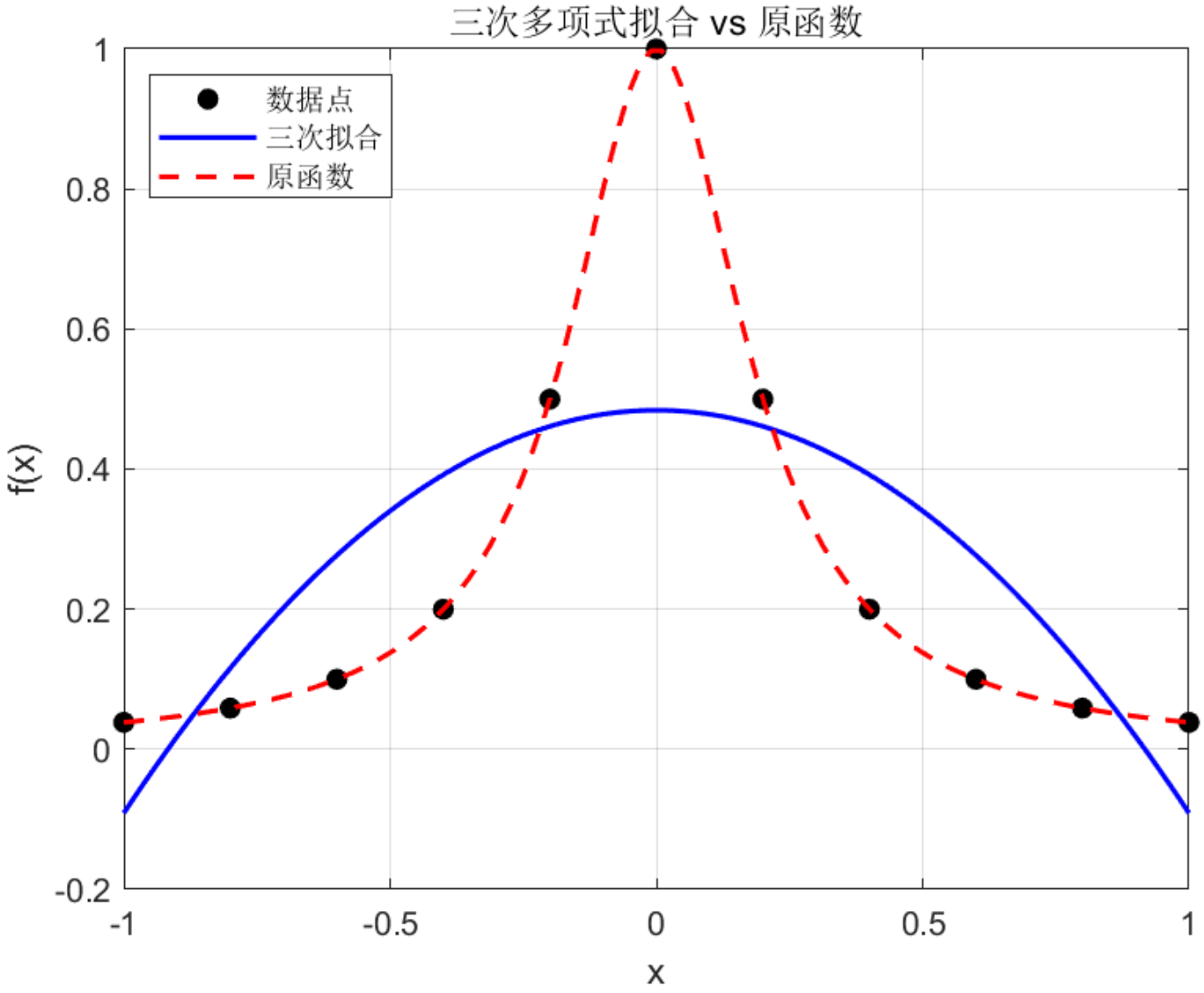
代码

```
% 生成数据点
i = 0:10;
xi = -1 + 0.2 * i;
yi = 1 ./ (1 + 25 * xi.^2);

% 三次多项式拟合
p = polyfit(xi, yi, 3);

% 生成密集点用于绘图
x_plot = linspace(-1, 1, 100);
y_fit = polyval(p, x_plot);
y_true = 1 ./ (1 + 25 * x_plot.^2);

% 绘制图像
figure;
plot(xi, yi, 'ko', 'MarkerFaceColor', 'k', 'DisplayName', '数据点');
hold on;
plot(x_plot, y_fit, 'b-', 'LineWidth', 1.5, 'DisplayName', '三次拟合');
plot(x_plot, y_true, 'r--', 'LineWidth', 1.5, 'DisplayName', '原函数');
xlabel('x');
ylabel('f(x)');
title('三次多项式拟合 vs 原函数');
legend('Location', 'northwest');
grid on;
hold off;

% 打印拟合方程
fprintf('三次拟合方程: y = %.4fx^3 + %.4fx^2 + %.4fx + %.4f\n', p(1), p(2), p(3),
p(4));
```

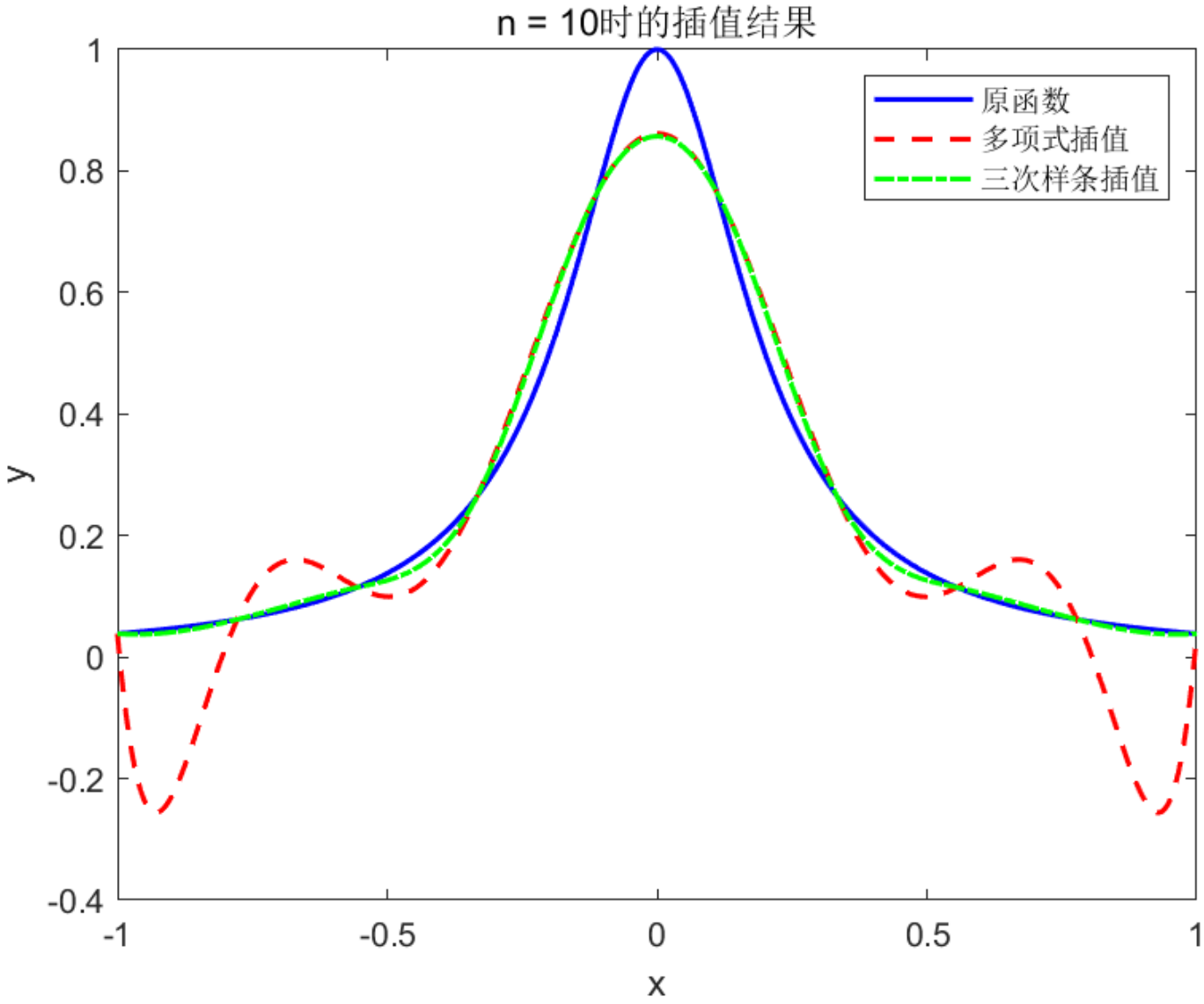
思路

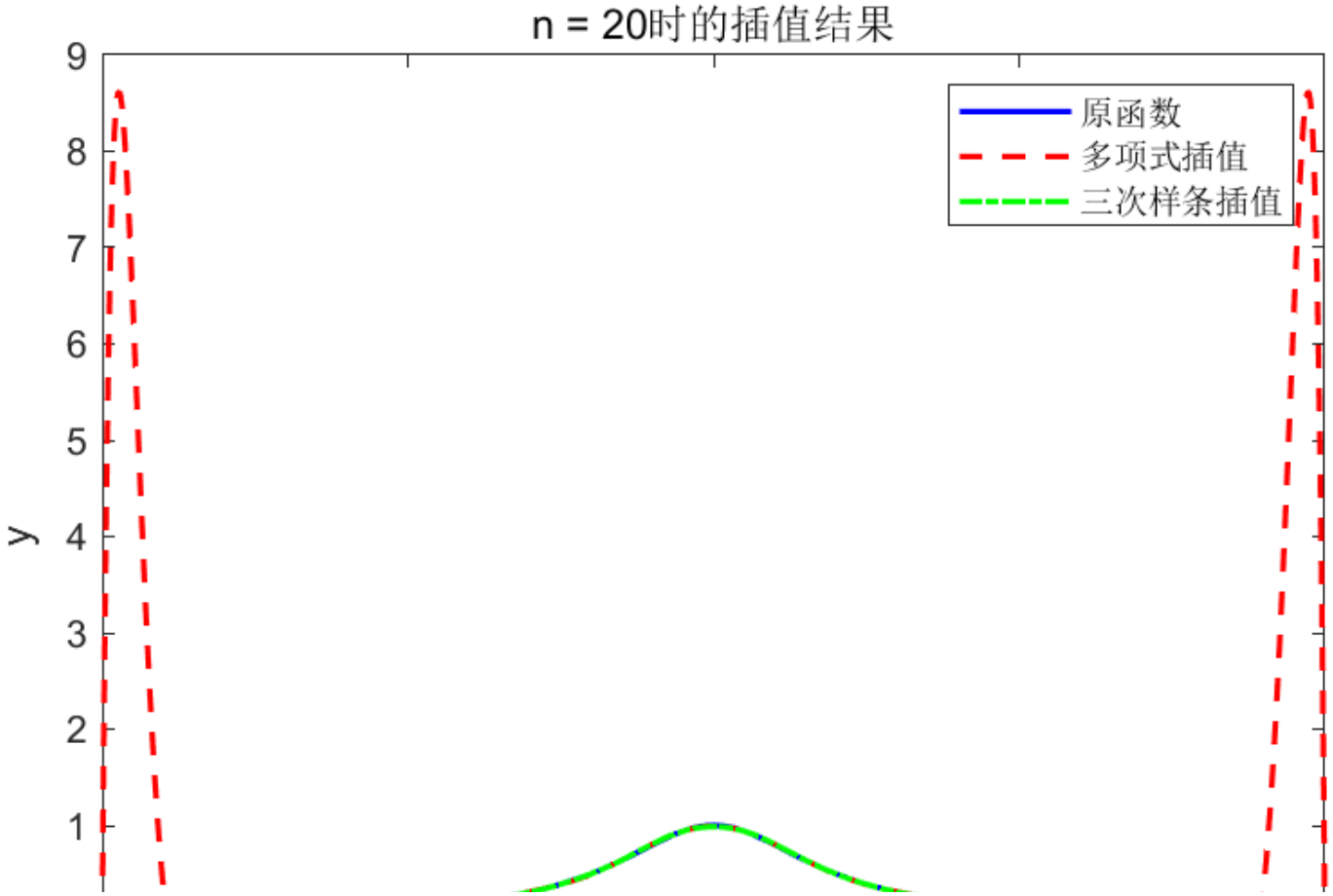
1. 数据点生成：在区间 [-1, 1] 上以间隔 0.2 生成 11 个数据点xi，并根据龙格函数计算对应函数值yi。
2. 三次多项式拟合：用polyfit函数对数据点(xi, yi)进行三次多项式拟合，得到系数向量p。
3. 数据准备绘图：在 [-1, 1] 上生成 100 个密集点x_plot，用polyval根据拟合系数p计算拟合函数值y_fit，同时计算原函数在x_plot上的值y_true

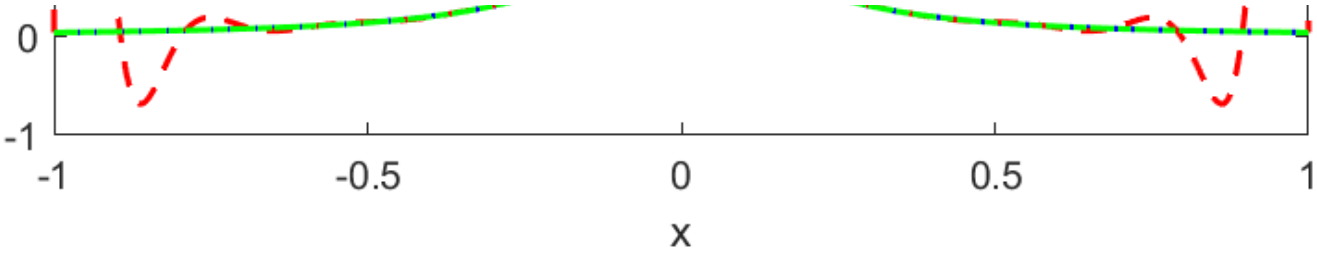
图像



与上一章第二题比较


n = 10时的插值结果
原函数
多项式插值
三次样条插值
y
x
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-1
-0.5
0
0.5
1


n = 20时的插值结果
原函数
多项式插值
三次样条插值
y
9
8
7
6
5
4
3
2
1



可见三次曲线拟合效果不是很好

三次拟合方程

y = 0.0000x^3 + -0.5752x^2 + -0.0000x + 0.4841

第2题

代码

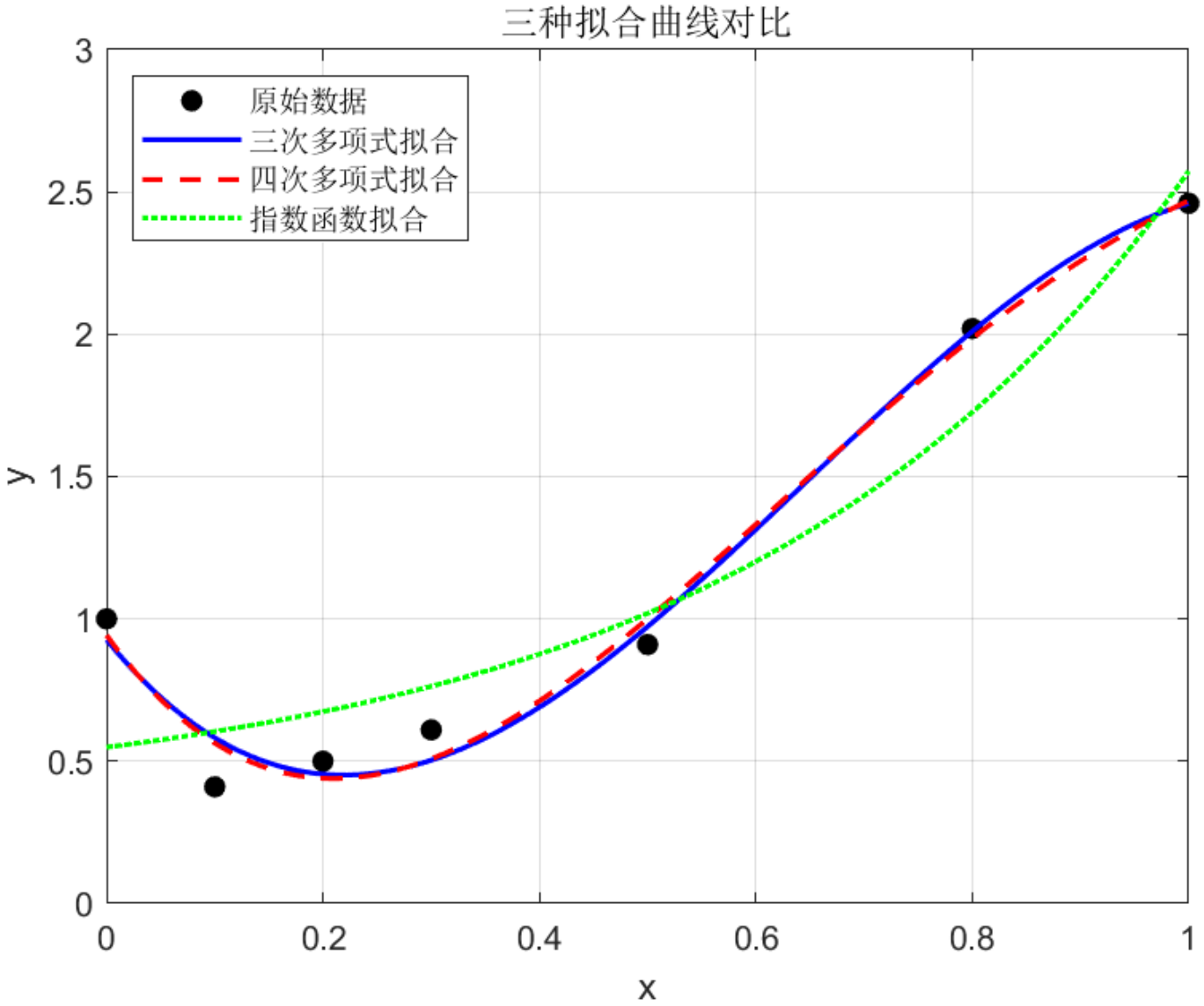
```
% 输入数据
x = [0.0, 0.1, 0.2, 0.3, 0.5, 0.8, 1.0];
y = [1.0, 0.41, 0.50, 0.61, 0.91, 2.02, 2.46];

% 三次多项式拟合
p3 = polyfit(x, y, 3);

% 四次多项式拟合
p4 = polyfit(x, y, 4);

% 自定义函数拟合：假设使用指数函数 y = a*exp(b*x) + c
% 定义指数模型和初始猜测值
ft = fittype('a*exp(b*x) + c', 'independent', 'x');
opts = fitoptions('Method', 'NonlinearLeastSquares');
opts.StartPoint = [1, 1, 0]; % 初始猜测值
[fit_result, ~] = fit(x', y', ft, opts);
a = fit_result.a;
b = fit_result.b;
c = fit_result.c;

% 生成密集点用于绘图
x_plot = linspace(0, 1, 100);
y_p3 = polyval(p3, x_plot);
y_p4 = polyval(p4, x_plot);
y_custom = a*exp(b*x_plot) + c;

% 绘制图像
figure;
plot(x, y, 'ko', 'MarkerFaceColor', 'k', 'DisplayName', '原始数据');
hold on;
plot(x_plot, y_p3, 'b-', 'LineWidth', 1.5, 'DisplayName', '三次多项式拟合');
plot(x_plot, y_p4, 'r--', 'LineWidth', 1.5, 'DisplayName', '四次多项式拟合');
plot(x_plot, y_custom, 'g:', 'LineWidth', 1.5, 'DisplayName', '指数函数拟合');
```

```
xlabel('x');
ylabel('y');
title('三种拟合曲线对比');
legend('Location', 'northwest');
grid on;
hold off;

% 打印方程
fprintf('三次多项式方程: y = %.4fx^3 + %.4fx^2 + %.4fx + %.4f\n', p3(1), p3(2),
p3(3), p3(4));
fprintf('四次多项式方程: y = %.4fx^4 + %.4fx^3 + %.4fx^2 + %.4fx + %.4f\n', p4(1),
p4(2), p4(3), p4(4), p4(5));
fprintf('指数函数方程: y = %.4f*exp(%.4fx) + %.4f\n', a, b, c);
```

思路

1. 多项式拟合：分别使用polyfit函数对数据进行三次和四次多项式拟合，得到相应的系数向量p3和p4。
2. 自定义函数拟合：定义指数函数模型y = a*exp(b*x) + c，设置拟合方法为非线性最小二乘法并给出初始猜测值，通过fit函数进行拟合，得到参数a、b、c。
3. 数据准备绘图：在区间 [0, 1] 上生成 100 个密集点x_plot，使用polyval函数分别计算三次、四次多项式在这些点上的值y_p3、y_p4，同时计算指数函数在这些点上的值y_custom

图像



拟合方程

三次多项式方程：y = -6.6221x^3 + 12.8147x^2 + -4.6591x + 0.9266 四次多项式方程：y = 2.8853x^4 + -12.3348x^3 + 16.2747x^2 + -5.2987x + 0.9427 指数函数方程：y = 0.2041*exp(2.3901x) + 0.3451

第3题

代码

```
% 定义参数
N = 33;         % 采样点数（满足 Nyquist 条件）
M = 16;         % 三角多项式最高次数

% 生成采样点
x = -pi + (0:N-1)*(2*pi)/N;
fx = x.^2 .* cos(x);

% 执行FFT
```

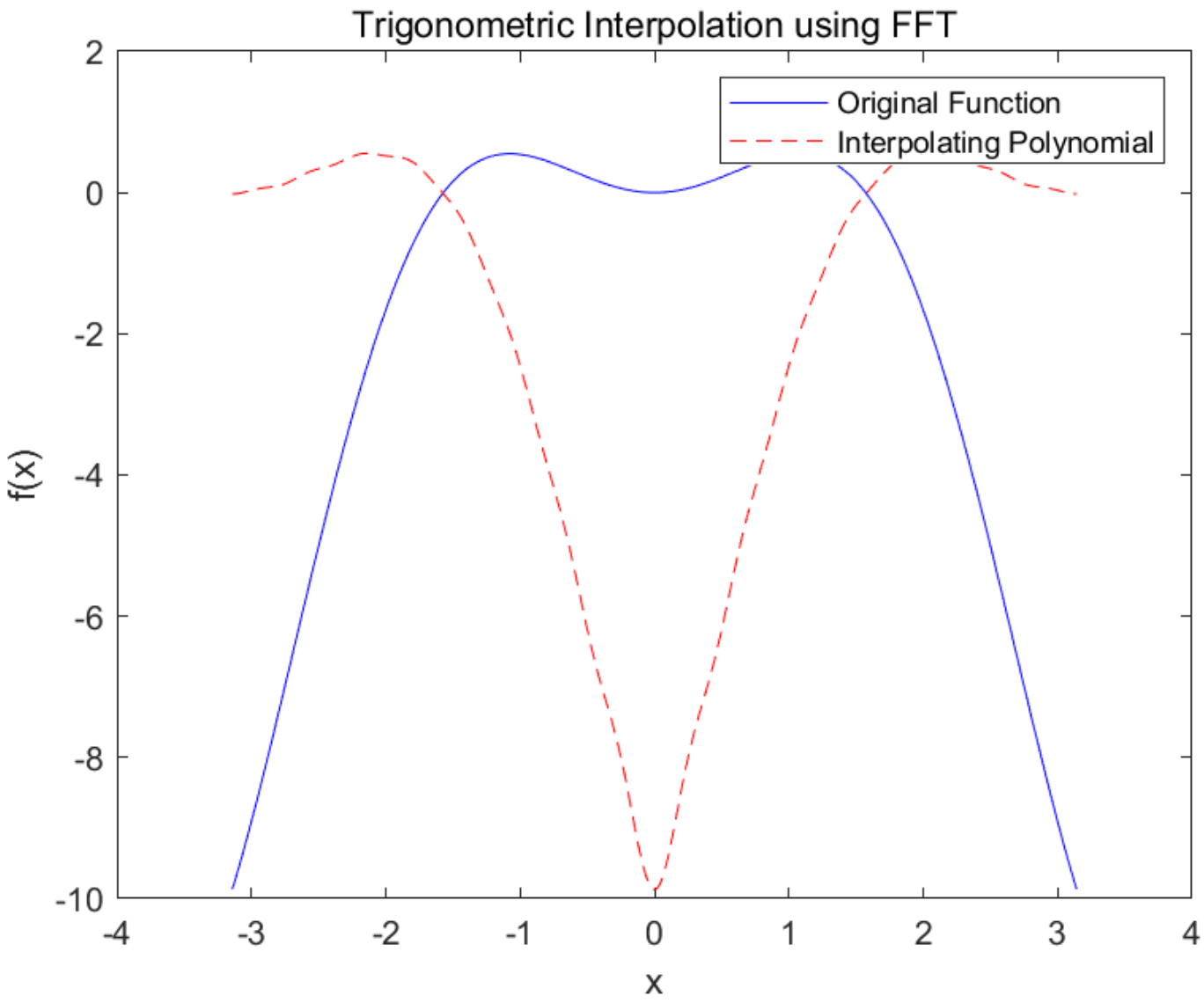
```
F = fft(fx);

% 提取系数
a0 = real(F(1)) / N;
a = 2 * real(F(2:M+1)) / N;
b = -2 * imag(F(2:M+1)) / N;

% 构建插值多项式
syms t;
P = a0 + sum(a .* cos((1:M)*t) + b .* sin((1:M)*t));
P = vpa(P, 6);  % 保留6位小数

% 验证插值误差
P_func = matlabFunction(P);
error = max(abs(fx - P_func(x)));
fprintf('最大插值误差: %.4e\n', error);

% 输出插值多项式
poly_str = char(P);
fprintf('插值多项式为: %s\n', poly_str);

% 可视化
xx = linspace(-pi, pi, 1000);
fx_exact = xx.^2 .* cos(xx);
P_exact = P_func(xx);

figure;
plot(xx, fx_exact, 'b', 'DisplayName', 'Original Function');
hold on;
plot(xx, P_exact, 'r--', 'DisplayName', 'Interpolating Polynomial');
legend;
xlabel('x');
ylabel('f(x)');
title('Trigonometric Interpolation using FFT');

figure;
plot(xx, abs(fx_exact - P_exact), 'g', 'DisplayName', 'Interpolation Error');
legend;
xlabel('x');
ylabel('Error');
title('Interpolation Error');
```

思路

1. 参数定义：设定采样点数 N 为 33，因为是16次三角插值多项式。
2. 生成采样点：在区间 [-π, π] 上生成 N 个等间距的采样点 x，并计算函数 f(x) = x.^2.* cos(x) 在这些点上的函数值 fx。
3. 执行快速傅里叶变换（FFT）：对 fx 执行 FFT 运算，得到变换结果 F。
4. 提取系数：从 FFT 结果 F 中提取出三角插值多项式的系数，包括常数项系数 a0，以及余弦项系数 a 和正弦项系数 b。

5. 构建插值多项式：利用符号变量 t 构建三角插值多项式 P，并使用 vpa 函数将多项式的系数保留 6 位小数。
6. 输出插值多项式：将符号形式的插值多项式 P 转换为字符串形式并输出。

图像



插值多项式

插值多项式为: 8.8742e-17*sin(8.0*t) - 0.31469*cos(4.0*t) - 0.0791708*cos(8.0*t) - 0.0366678*cos(16.0*t) - 0.0653805*cos(9.0*t) - 3.34823e-16*sin(2.0*t) - 3.21554e-16*sin(4.0*t) - 2.23442*cos(2.0*t) + 4.58449e-17*sin(16.0*t) - 0.193235*cos(5.0*t) - 0.0559035*cos(10.0*t) - 1.31334e-17*sin(9.0*t) - 0.0492237*cos(11.0*t) - 1.00503e-16*sin(5.0*t) + 6.01062e-16*sin(10.0*t) - 0.637307*cos(3.0*t) - 0.133761*cos(6.0*t) - 0.0444653*cos(12.0*t) + 1.86469e-16*sin(11.0*t) - 0.0410991*cos(13.0*t) - 5.064e-16*sin(3.0*t) - 4.58449e-17*sin(6.0*t) + 1.44814e-16*sin(12.0*t) - 0.100076*cos(7.0*t) - 0.038798*cos(14.0*t) - 3.59805e-16*sin(13.0*t) - 0.0373599*cos(15.0*t) + 1.10439e-16*sin(7.0*t) + 4.49253e-18*sin(14.0*t) + 5.27211e-17*sin(15.0*t) - 3.802*cos(t) - *6.66367e-17*sin(t) - 2.00605