

一、一般节点的重心第二形式与切比雪夫零点重心第二形式

通过学习我们知道采用一般节点的重心第二形式在计算的复杂度上并没有显著的优势，但是采用了切比雪夫的零点作为差值的节点之后计算效率得到了大幅度提升，因此本文将比较两种形式的拟合情况，误差大小以及一些补充的计算方法，其中代码均有自己的思考与相关注释

1.1 相关代码

两种重心形式的代码如下所示：

一般节点的重心第二形式

```
1 function v = bary_normal(x, f, u)
2 %插值多项式一般的第二重心形式
3 %x为插值节点，y为差值函数，u为输入的节点
4 n = length(x);
5 y = f(x);
6 w = ones(n, 1); %开始构造重心权
7 for i = 1:n
8     for j = 1:n
9         if j ~= i
10             w(i) = w(i) / (x(i) - x(j));
11         end
12     end
13 end
14 numer = zeros(size(u)); %构造与u行列相同的0矩阵
15 denom = zeros(size(u));
16 exact = zeros(size(u));
17 for j = 1:n
18     xdiff = u - x(j);
19     temp = w(j) ./ xdiff;
20     numer = numer + temp * y(j);
21     denom = denom + temp;
22     %这一步是防止去的节点和cheb的节点重合
23     exact(xdiff == 0) = j; %在矩阵exact中将所有xdiff=0的元素赋值为j，
        %xdiff == 0表示取出xdiff为0的位置
24     %j不能改！找y(exact(jj))时有用
25 end
26 v = numer ./ denom;
27 jj = find(exact); %返回非0元素的下标，对于矩阵按照列来进行排序
28 v(jj) = y(exact(jj)); %若有u取的是cheb节点的值，则这个值=f(x_j)
```

以切比雪夫零点作为节点的第二重心形式

```
1 function v = bary_cheb_zero(f, n, u)
2 %f为逼近的函数，n为插值多项式的次数，u为输入的节点
3 %在画图的时候，bary_interp本身就是一个函数，和sin(x)之类的效果一样，直接用即可
```

```

4     x = cos((2.*(0:n)' + 1) * pi / (2 * n + 2));    %切比雪夫的零点, 注意要用列向量的形式
5     y = f(x);
6     w = sin((2.*(0:n)' + 1) * pi / (2 * n + 2)) .* (-1).^((0:n)');    %重心权
7     numer = zeros(size(u));    %构造与u行列相同的0矩阵
8     denom = zeros(size(u));
9     exact = zeros(size(u));
10    for j = 1:n+1
11        xdiff = u - x(j);
12        temp = w(j) ./ xdiff;
13        numer = numer + temp * y(j);
14        denom = denom + temp;
15        %这一步是防止去的节点和cheb的节点重合
16        exact(xdiff == 0) = j;    %在矩阵exact中将所有xdiff=0的元素赋值为j,
        %xdiff == 0表示取出xdiff为0的位置
17        %j不能改! 找y(exact(jj))时有用
18    end
19    v = numer ./ denom;
20    jj = find(exact);    %返回非0元素的下标, 对于矩阵按照列来进行排序
21    v(jj) = y(exact(jj));    %若有u取的是cheb节点的值, 则这个值=f(x_j)

```

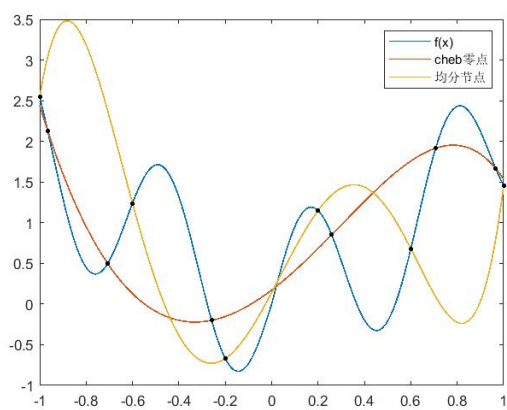
1.2 实例分析

我们以 $f(x) = |x| + x^2 + \sin(10x)$ 为例, 在一般形式的重心公式中取均分节点
其中节点的显示表达式分别为:

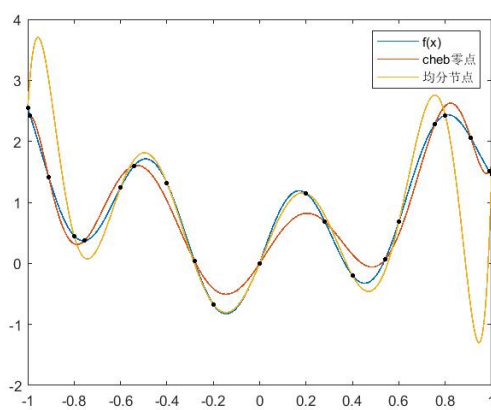
- (1) 均分节点 $x_k = -1 + kh$, $h = 2/n$, $k = 0, 1, \dots, n$
- (2) 第一类切比雪夫零点 $x_k = \cos\left(\frac{2k+1}{2n+2}\pi\right)$, $k = 0, 1, \dots, n$

1.2.1 函数拟合曲线图

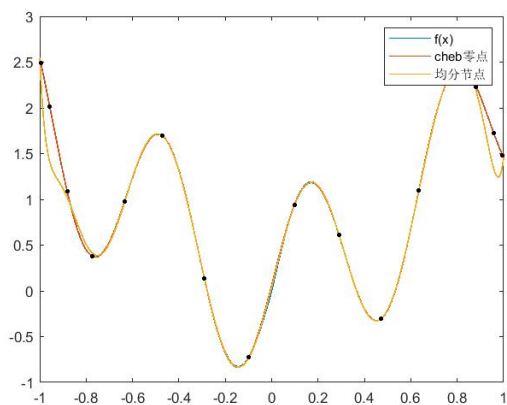
本文分别采用了5个, 10个, 15个, 20个节点进行差值拟合, 两类差值拟合曲线与原函数在-1至1上的图像如下所示



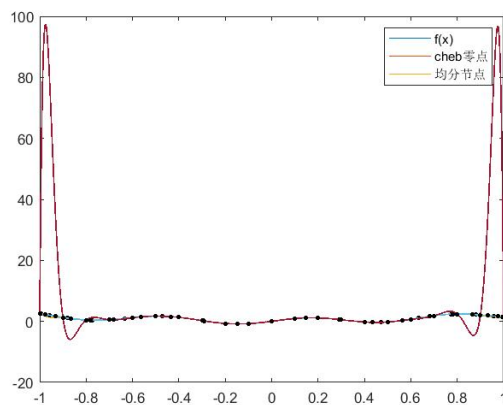
(a) $n = 5$



(b) $n = 10$



(c) $n = 15$



(d) $n = 20$

图 1: 函数拟合曲线图

从图中可以看出，在 $n=10$ 和 $n=15$ 情形下节点为cheb零点的重心第二公式要明显优于均分节点。而在 $n=20$ 的情形下，均分节点的重心第二公式发生了“龙格现象”，出现严重的偏差。

1.2.2 误差分析

两种插值方法的误差分析如下所示：

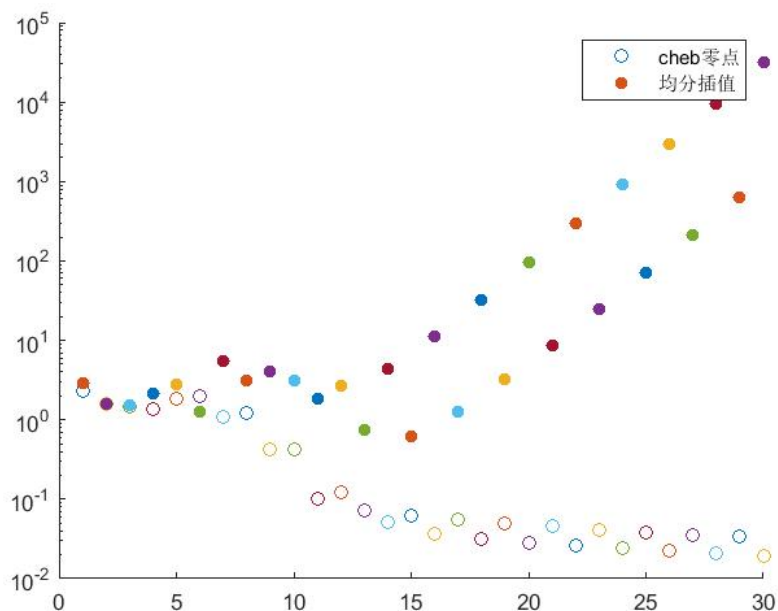


图 2: 函数误差分析图

可以看出在 n 大于15后，均分节点的误差由于龙格现象不断扩大，而cheb零点的误差不断缩小

1.3 补充分析

除了采用cheb零点的重心形式差值是一种比较好的插值方法外，Hermite插值也是一种优秀的插值方式下面将给出Hermite插值的相关代码

```

1 function v = hermite_interp(n, y, dy, u)
2 %我们采用cheb零点作为差值的节点，因为cheb零点具有较好的性质，y为差值函数，dy为差值函数的导
   数，u为输入的节点
3 %注意u要输入列向量！
4 x = cos((2.:(n-1)+1)*pi/(2*n+2));
5 y = y(x);
6 dy = dy(x);
7 m = length(u);
8 U = zeros(m, n); V = zeros(m, n); %hermit基函数
9 L = ones(m, n); %Lagrange基函数
10 d = zeros(n, 1); %Lagrange基导数
11 for j = 1:n
12     for k = 1:n
13         if k~=j
14             L(:, j) = L(:, j).*(u - x(k))./(x(j) - x(k));
15             d(j) = d(j) + 1./(x(j) - x(k));
16         end
17     end
18 end
19 for j = 1:n
20     U(:, j) = 1 - 2*(u - x(j))*d(j);

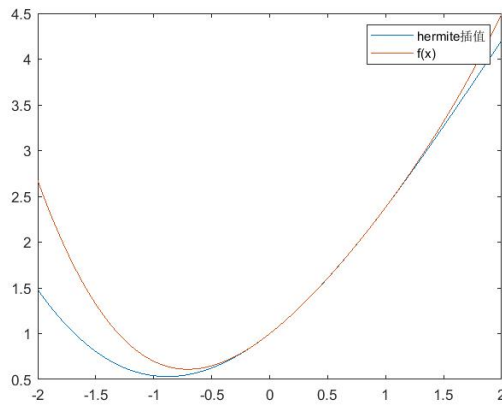
```

```

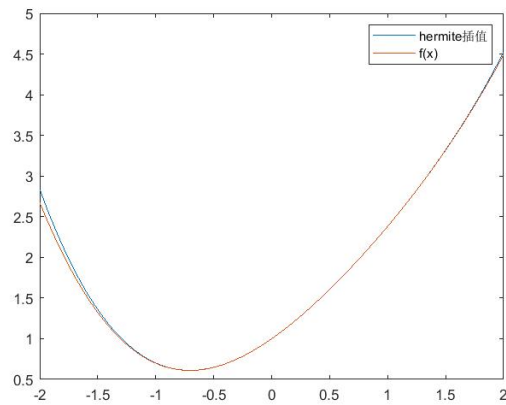
21     V(:,j) = u - x(j);
22     end
23     U = U.*(L.^2);
24     V = V.*(L.^2);
25     v = U*y + V*dy;

```

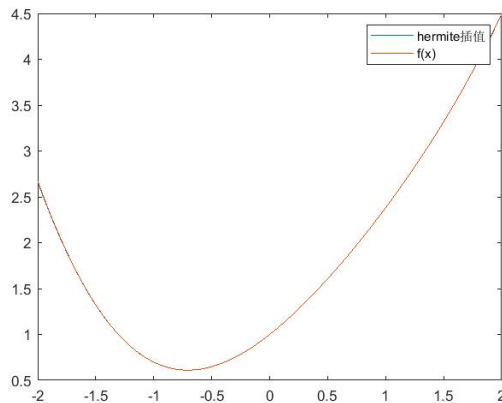
其中由于cheb极值点具有比较好的性质，因此我们采用cheb极值点作为插值节点
下面我们对 $f(x) = \sin(x) + \cos(x) + x^2$ 进行插值其插值图像如下所示



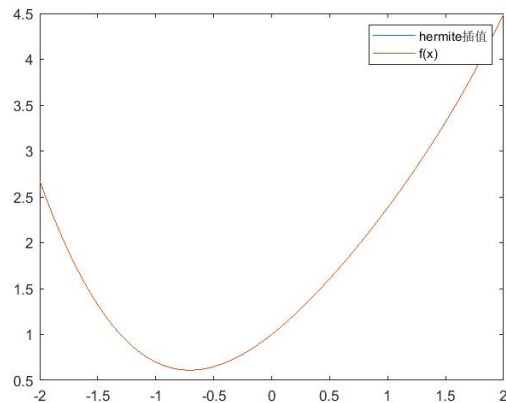
(a) $n = 2$



(b) $n = 3$



(c) $n = 4$



(d) $n = 5$

图 3: 函数拟合曲线图

可以看出在 $n=5$ 时已经基本没有误差，此时最大误差为0.0004432，因此Hermite插值是一个十分优秀的插值方法

本文还演示了Newton插值的拟合情况，相关代码如下所示：

```

1     function v = Newton_interp(x, f, u)
2     n = length(x);
3     m = length(u);
4     D = zeros(n, n);
5     v = zeros(size(m));
6     %构造差商
7     D(:,1) = f(x);
8     for k = 2:n

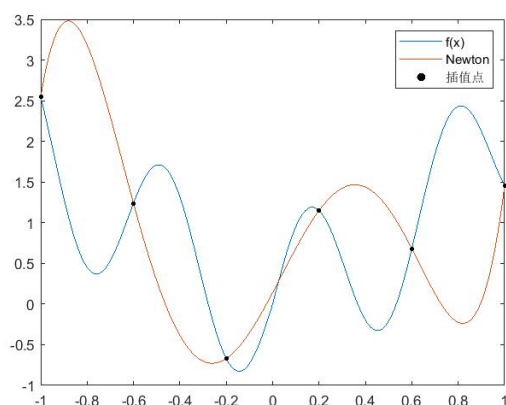
```

```

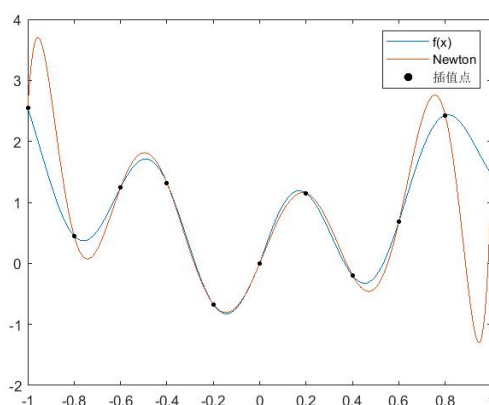
9         for j = k:n
10             D(j,k) = (D(j,k-1) - D(j-1,k-1))./(x(j) - x(j-k+1));
11         end
12     end
13     a = diag(D); %差商就是矩阵的对角线上的值
14
15     %开始通过秦久绍算法计算
16     v = repmat(a(n), m, 1); %首先构造最里面的值
17     for k = n-1:-1:1
18         v = a(k) + v.*(u - x(k));
19     end

```

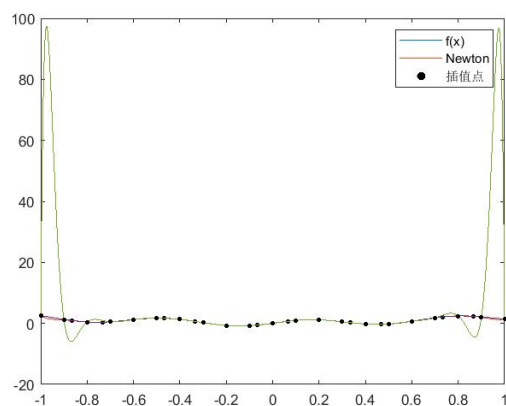
拟合图像如下所示，节点与函数的选取与前文相同



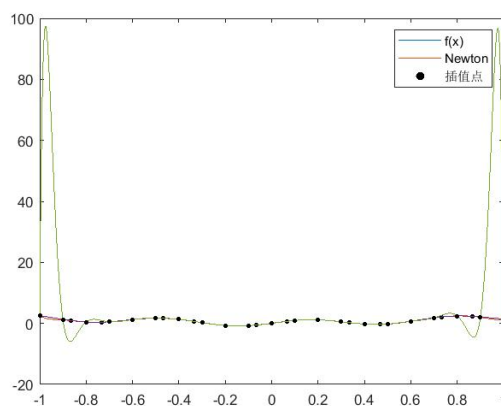
(a) $n = 5$



(b) $n = 10$



(c) $n = 15$



(d) $n = 20$

图 4: Newton拟合曲线图

可以看出Newton形式的插值虽然在节点改变时不用全部计算新的节点，但是仍然由于插值节点的选择问题在 n 比较大的时候出现了龙格现象。

二、勒让德零点与极值点的重心第二形式插值拟合

本文使用了勒让德零点与极值点作为插值节点，采用重心第二形式进行插值拟合，具体代码如下：

勒让德零点:

```
1 function v = lengend(n, f, u)
2 %计算勒让德多项式的零点和权
3 beta = .5./sqrt(1 - (2*(1:n)).^(-2));
4 T = diag(beta, 1) + diag(beta, -1);
5 [V, D] = eig(T);
6 x = diag(D);
7 wL = (2*V(1,:).^2)';
8 y = f(x);
9 %计算勒让德多项式的重心权
10 w = (-1).^((0:n)')*.sqrt((1-x.*x).*wL);
11 numer = zeros(size(u)); %构造与u行列相同的0矩阵
12 denom = zeros(size(u));
13 exact = zeros(size(u));
14 for j = 1:n+1
15     xdiff = u - x(j);
16     temp = w(j)./xdiff;
17     numer = numer + temp*y(j);
18     denom = denom + temp;
19     %这一步是防止去的节点和lengend的节点重合
20     exact(xdiff == 0) = j; %在矩阵exact中将所有xdiff=0的元素赋值为j,
    %xdiff == 0表示取出xdiff为0的位置
21     %j不能改! 找y(exact(jj))时有用
22 end
23 v = numer./denom;
24 jj = find(exact); %返回非0元素的下标, 对于矩阵按照列来进行排序
25 v(jj) = y(exact(jj));
```

勒让德极值点:

```
1 function v = lengend_poles(n, f, u)
2 %计算勒让德多项式的极值点和权
3 beta = .5./sqrt(1 - (2*(1:n-1)).^(-2));
4 beta(n) = sqrt(n./(2*n-1));
5 T = diag(beta, 1) + diag(beta, -1);
6 [V, D] = eig(T);
7 x = diag(D);
8 wL = (2*V(1,:).^2)';
9 y = f(x);
10 %计算勒让德极值点的重心权
11 w = (-1).^((0:n)')*.sqrt(wL);
12 numer = zeros(size(u)); %构造与u行列相同的0矩阵
13 denom = zeros(size(u));
14 exact = zeros(size(u));
15 for j = 1:n+1
16     xdiff = u - x(j);
17     temp = w(j)./xdiff;
```

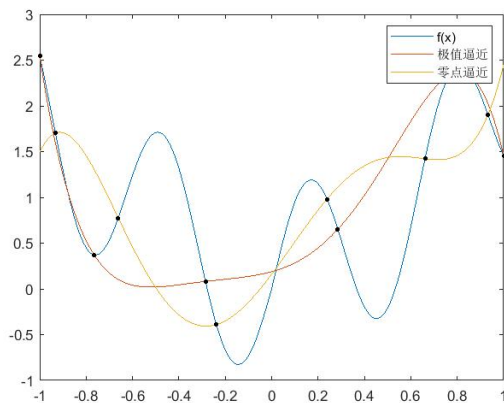
```

18     numer = numer + temp*y(j);
19     denom = denom + temp;
20     %这一步是防止去的节点和legend的节点重合
21     exact(xdiff ==0) = j;          %在矩阵exact中将所有xdiff=0的元素赋值为j,
        xdiff == 0表示取出xdiff为0的位置          %j不能改! 找y(exact(jj))时有用
22
23     end
24     v = numer./denom;
25     jj = find(exact);          %返回非0元素的下标, 对于矩阵按照列来进行排序
26     v(jj) = y(exact(jj));

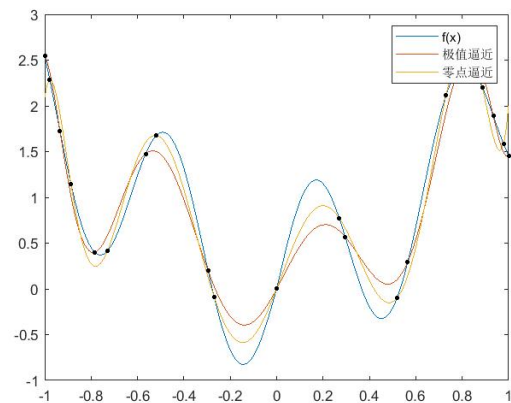
```

2.1 函数拟合曲线图

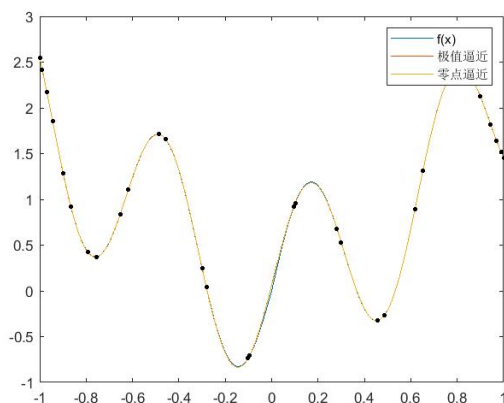
我们以 $f(x) = |x| + x^2 + \sin(10x)$ 为例, 本文分别采用了5个, 10个, 15个, 20个节点进行差值拟合, 两类差值拟合曲线与原函数在-1至1上的图像如下所示:



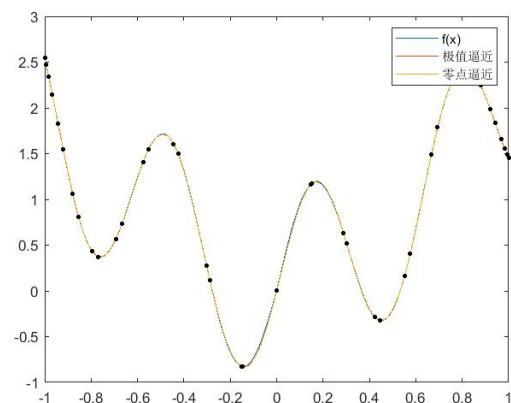
(a) $n = 5$



(b) $n = 10$



(c) $n = 15$



(d) $n = 20$

图 5: 函数拟合曲线图

2.2 误差分析

两种插值多项式的误差分析如下图:

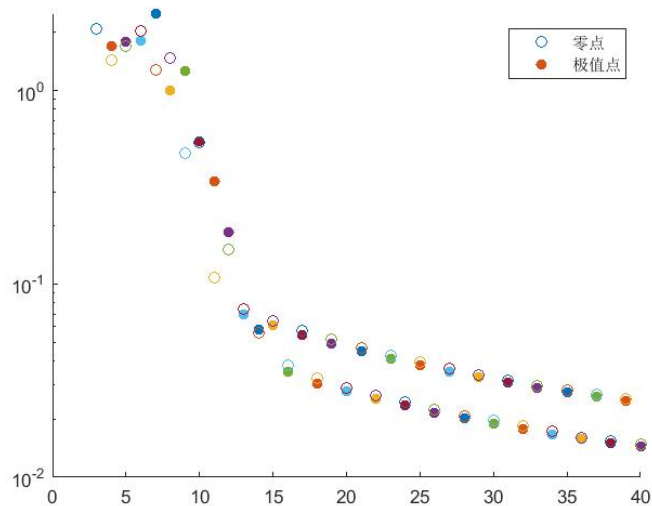


图 6: 函数误差分析图

可以看出两种节点的插值逼近误差基本相同

2.3 吉布斯现象

补充一个有趣的现象，在函数为间断函数时，插值逼近的误差在间断点会震荡，并且不论 n 为多少，误差始终存在

下面以-1和1的方波函数为例：

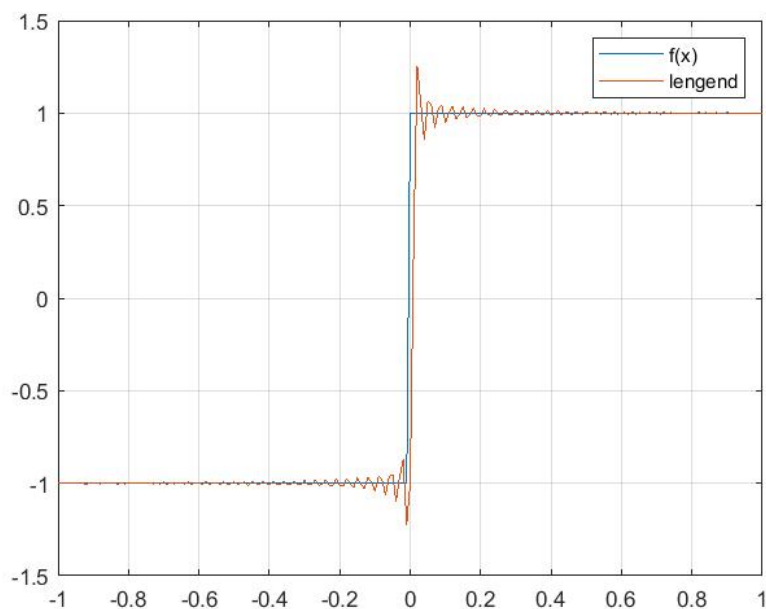


图 7: 吉布斯现象图($n = 100$)

三、切比雪夫逼近

本文主要研究了第一类切比雪夫逼近的形式，并且分析其收敛速度与误差情况。

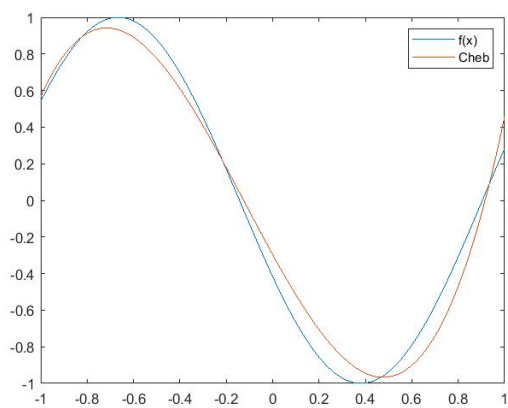
第一类切比雪夫逼近的代码如下所示：其中在计算定积分时使用了高斯勒让德积分计算公式

```
1 function v = Cheb_interp(f, n, u)
2 %f为要逼近的函数，n为切比雪夫多项式的次数，u为逼近的节点
3 syms t;
4 T = t.*ones(1,n+1);
5 T(1) = 1;
6 T(2) = t;
7 c = zeros(1, n+1);
8 %f首先计算前两个c值
9 c(1) = gauss_L(matlabFunction(f(t)*T(1)/sqrt(1-t^2)),2000)/pi; %f这里使用了
    高斯积分
10 c(2)=2*gauss_L(matlabFunction(f(t)*T(2)/sqrt(1-t^2)),2000)/pi;
11 v = c(1) + c(2)*t;
12 %f计算剩余的
13 for i = 3: n+1
14     T(i) = 2*t*T(i-1)-T(i-2);
15     c(i) = 2*gauss_L(matlabFunction(f(t)*T(i)/sqrt(1-t^2)),2000)/pi;
16     v = v + c(i)*T(i);
17     v = vpa(v,6); %f控制精度
18     if i == n+1
19         v = subs(v,'t',u); %f带入节点
    的值
20     end
21 end
22 end
```

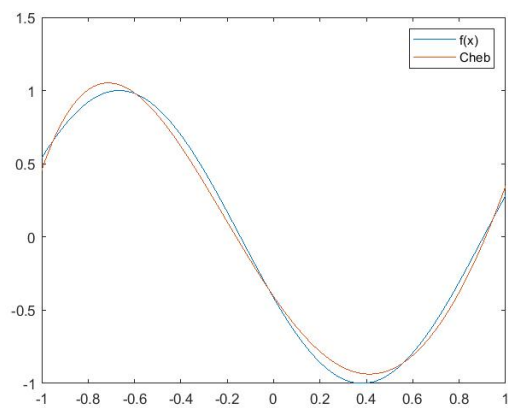
高斯勒让德积分计算公式的计算代码

```
1 function I = gauss_L(f,n)
2 beta = .5./sqrt(1 - (2*(1:n)).^(-2));
3 T = diag(beta, 1) + diag(beta, -1);
4 [V, D] = eig(T);
5 x = diag(D);
6 w = 2*V(1,:).^2;
7 I = w*f(x);
8 end
```

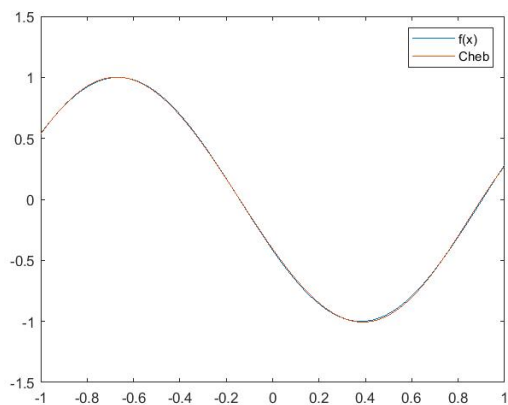
下面我们将对函数 $f(x) = \cos(3x + 2)$ 使用切比雪夫进行逼近，并计算多项式不同次数时的误差拟合图像如下所示



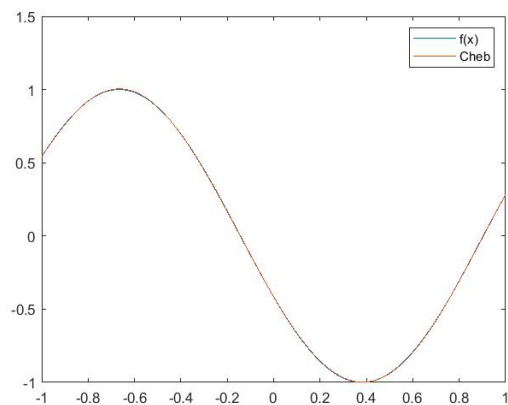
(a) $n = 3$



(b) $n = 4$



(c) $n = 5$



(d) $n = 6$

图 8: Cheb拟合曲线图

可以看出在 $n = 6$ 时拟合曲线与函数曲线基本重合, 说明cheb多项式拟合效果很好
其误差如下图所示

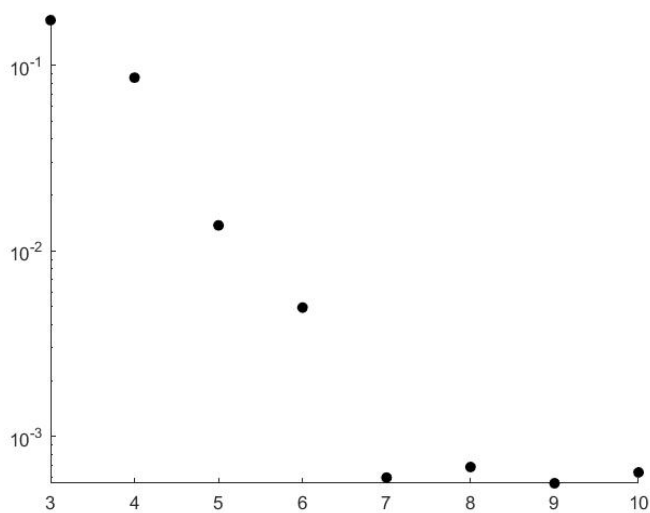


图 9: 函数误差分析图

可以看出随着多项式的次数增加, 误差在不断减小, 但由于没有对函数进行变形, 而采用了高斯勒让德积分计算公式直接计算, 所以在 $n > 7$ 后误差不再下降

四、数值积分

本文使用了切比雪夫第一形式与第二形式的数值积分公式，因为这两个方法的代码在书中没有给出，并且计算了相应的误差

其代码如下所示:

```
1 function v = gauss_cheb1 (f, n)
2 x = cos((2.*(0:n)' + 1) * pi / (2 * n + 2));
3 y = f(x);
4 v = pi / (n + 1) .* sum(y);
```

```
1 function v = gauss_cheb2 (f, n)
2 x = cos((0:n)' + 1) * pi / (n + 2);
3 w = pi ./ (n + 2) .* (sin((0:n)' + 1) * pi / (n + 2)).^2;
4 v = f(x)' * w;
```

取 $f(x) = 1/(1 + 4^x)$ ，计算相应的误差,并且做出相应的误差图形

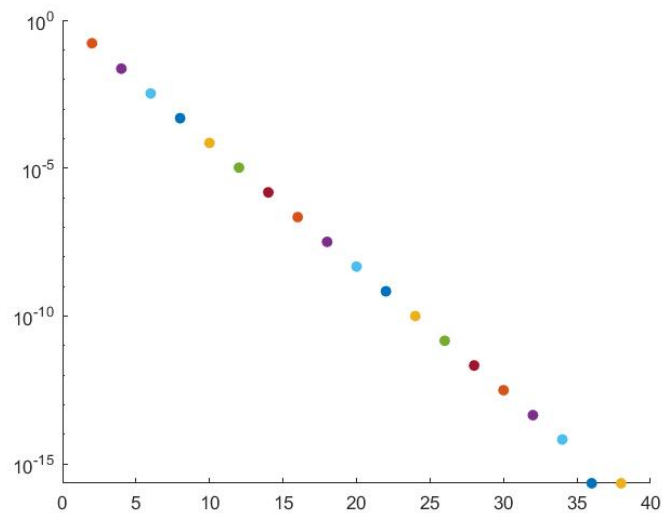


图 10: 切比雪夫第一形式误差分析图

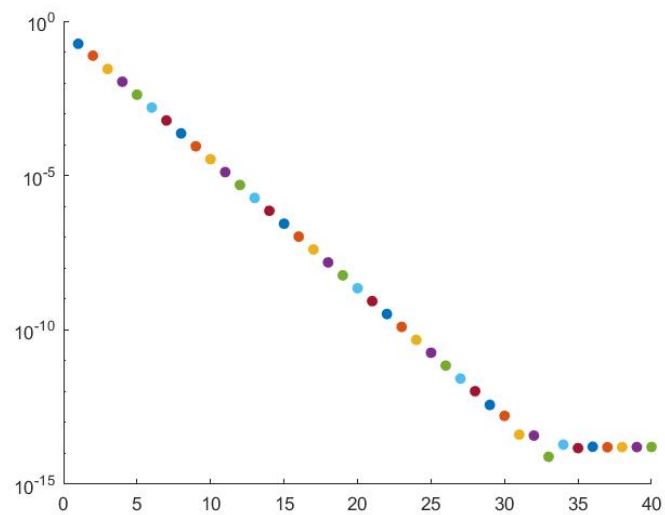


图 11: 切比雪夫第二形式误差分析图