对于函数f(x) = sin(x) ，试用复化梯形公式、复化Simpson公式及复化Cotes公式计算积分，并比较其误差。

首先，在MATLAB输入以下程序定义函数：

**[plain]** [view plain](http://blog.csdn.net/ednah/article/details/53233438" \o "view plain" \t "http://blog.csdn.net/ednah/article/details/_blank) [copy](http://blog.csdn.net/ednah/article/details/53233438" \o "copy" \t "http://blog.csdn.net/ednah/article/details/_blank)

1. function y=f(x)
2. y=sin(x);

1.复化梯形求积公式的代码

**[plain]** [view plain](http://blog.csdn.net/ednah/article/details/53233438" \o "view plain" \t "http://blog.csdn.net/ednah/article/details/_blank) [copy](http://blog.csdn.net/ednah/article/details/53233438" \o "copy" \t "http://blog.csdn.net/ednah/article/details/_blank)

1. function Tn=Tn(a,b,n)
2. format long
3. h=(b-a)/n;
4. sum=0;
5. for k=1:n-1
6. sum=sum+f(a+k.\*h);
7. end
8. Tn=(f(a)+2\*sum+f(b))\*h/2;
9. end

2.复化Simpson公式的代码

**[plain]** [view plain](http://blog.csdn.net/ednah/article/details/53233438" \o "view plain" \t "http://blog.csdn.net/ednah/article/details/_blank) [copy](http://blog.csdn.net/ednah/article/details/53233438" \o "copy" \t "http://blog.csdn.net/ednah/article/details/_blank)

1. function Sn = Sn(a,b,n)
2. format long
3. h = (b-a)/n;
4. sum1 = 0;
5. sum2 = 0;
6. for i = 0:n-1
7. sum1 = sum1 + f(a+(i+1/2).\*h);
8. end
9. for j = 1:n-1
10. sum2 = sum2 + f(a+j.\*h);
11. end
12. Sn = h/6\*(f(a)+4\*sum1+2\*sum2+f(b));

3.复化Cotes公式的代码

**[plain]** [view plain](http://blog.csdn.net/ednah/article/details/53233438" \o "view plain" \t "http://blog.csdn.net/ednah/article/details/_blank) [copy](http://blog.csdn.net/ednah/article/details/53233438" \o "copy" \t "http://blog.csdn.net/ednah/article/details/_blank)

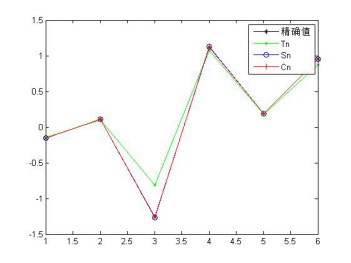
1. function Cn = Cn(a,b,n)
2. format long
3. h = (b-a)/n;
4. sum1 = 0;
5. sum2 = 0;
6. for i = 0:n-1
7. sum1 = sum1 + 32\*f(a+(i+1/4).\*h)+12\*f(a+(i+1/2).\*h)+32\*f(a+(i+3/4).\*h);
8. end
9. for j = 1:n-1
10. sum2 = sum2 + 14\*f(a+j.\*h);
11. end
12. Cn = h/90\*(7\*f(a)+sum1+sum2+7\*f(b));

4.不同区间和分割次数下的精确值和近似值的比较代码

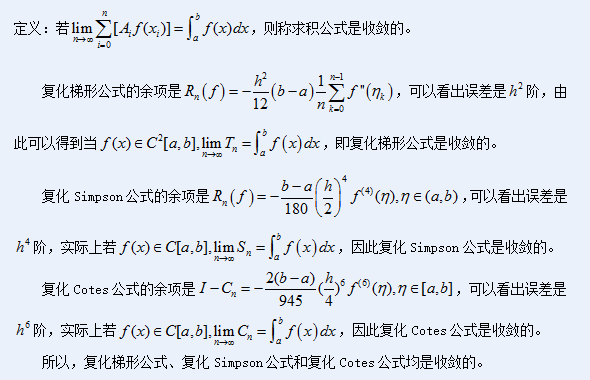
**[plain]** [view plain](http://blog.csdn.net/ednah/article/details/53233438" \o "view plain" \t "http://blog.csdn.net/ednah/article/details/_blank) [copy](http://blog.csdn.net/ednah/article/details/53233438" \o "copy" \t "http://blog.csdn.net/ednah/article/details/_blank)

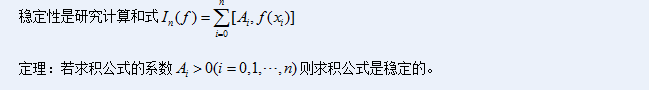
1. y=[-0.150920964949986,0.106044306328975,-1.260000791649389,1.122733727670800,0.185427920752058,0.955618503172607];
2. Tn=[-0.144707438957434 ,0.103357516942886,-0.809037206720437,1.074587530720150,0.176653381231435 ,0.874739367319743];
3. Sn=[-0.150933735388946,0.106047691917755 ,-1.267931587831384,1.122836754412495,0.185448626084829 ,0.956086587141625];
4. Cn=[ -0.150920958200157,0.106044290463326,-1.259953005777527,1.122733636186552,0.185427890841067,0.955744078179222];
5. plot(y,'-k\*');
6. hold on;
7. plot(Tn,'-g.');
8. hold on;
9. plot(Sn,'-bo');
10. hold on;
11. plot(Cn,'-r+');
12. legend('精确值','Tn','Sn','Cn');

5.图像



6.结果分析

1 误差分析  
在上面的数值计算中，复化Cotes公式计算所得的误差最小，其次是复化Simpson公式，误差最大的是复化梯形公式。  
  
2 精确值比较  
它们的精度有很大的差别：与精确值比较，复化梯形公式的结果只有一位有效数字，复化Simpson公式的结果有四位有效数字，而复化Cotes公式的结果却有七位有效数字。  
由此可知，复化Cotes公式的代数精度最高，复化Simpson其次，复化梯形公式的代数精度最低。  
  
3 收敛性分析  
  
  
4 稳定性分析

  
  
可知三个求积公式的求积系数都为正，所以复化梯形公式、复化Simpson公式和复化Cotes公式均是稳定的。