**Algorithm 2: Newton method**

|  |
| --- |
| Method introduction: |
| For the Newton–Cotes rules to be accurate, the step size h needs to be small, which means that the interval of integration [a,b] must be small itself, which is not true most of the time. For this reason, one usually performs numerical integration by splitting [a,b] into smaller subintervals, applying a Newton–Cotes rule on each subinterval, and adding up the results. This is called a composite rule. |
| Algorithm Design |
| step 1：Divide the interval [a,b] into n halves  step 2：the lower-order newton-cotes formula is first used on each child interval [xk,xk+1]  step 3：the integral approximation IK is obtained, then k = 0, 1, .., n-1 the approximate value of IK summation as exact integral I. |
| Matlab code |
| function yy = mymulNewtonCotes(ft,a,b,m,n)  % 复化Newton-Cotes数值积分公式，即在每个子区间上使用Newton-Cotes公式，然后求和,  % 参考的输入形式为mymulNewtonCotes(ft,0,1,10,2)  % 参数说明:  % ft——被积函数，此题中ft=@(t)t.\*exp(t^2/2)  % a——积分下限  % b——积分上限  % m——将区间[a,b]等分的子区间数量  % n——采用的Newton-Cotes公式的阶数，必须满足n<8，否则积分没法保证稳定性  % (1)n=1时为复化梯形公式  % (2)n=2时为复化辛普森公式  xx = linspace(a,b,m+1);  for l = 1:m  s(l) = myNewtonCotes(ft,xx(l),xx(l+1),n);  end  yy = sum(s);  function [y,Ck,Ak] = myNewtonCotes(ft,a,b,n)  % 牛顿-科特斯数值积分公式  % Ck——科特斯系数  % Ak——求积系数  % y——牛顿-科特斯数值积分结果  xk = linspace(a,b,n+1);  for j = 1:n+1  ff(j) = ft(xk(j));  end  % 计算科特斯系数  for i=1:n+1  k=i-1;  Ck(i)=(-1)^(n-k)/factorial(k)/factorial(n-k)/n\*quadl(@(t)intfun(t,n,k),0,n);  end  % 计算求积系数  Ak=(b-a)\*Ck;  % 求和算积分  y=Ak\*ff';  function f=intfun(t,n,k)  % 科特斯系数中的积分表达式  f=1;  for i=[0:k-1,k+1:n]  f=f.\*(t-i);  end |
| Examples and Result |
| Use composite rule rule *C*n=(n=2,3,4,5,6,7,8,9) to calculate , compare with exact solution …    Remarks |
| 此处写该方法程序设计的一些注意事项，也可以空白 |
|  |