数值积分

- 求积公式的代数精度
- Newton-Cotes求积公式
- 复化梯形求积公式
- 复化Simpson公式
- 复化Cotes公式

求积公式的代数精度

要使机械求积公式：

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k)$$

具有m次代数精度，则需要使求积公式对$f(x) = 1, x^1, x^2, \ldots, x^m$都准确成立。

其中：

$$I_n = \sum_{k=0}^{n} A_k f(x_k)$$

$$A_k = \int_a^b l_k(x)dx = \int_a^b \prod_{i=0\&i\neq k}^{n} \frac{(x - x_i)}{x_k - x_i}dx$$

属于插值型求积公式.

对于求积公式：

$$I_n = \sum_{k=0}^{n} A_k f(x_k)$$

至少有n次代数精度，则该求积公式属于插值型求积公式，同时有：

$$\sum_{k=0}^{n} A_k = b - a$$

例题：

试确定一个至少具有2次代数精度的公式

$$\int_0^4 f(x)dx \approx Af(0) + Bf(1) + Cf(3)$$

至少具有2次代数精度，则对$f(x) = 1, x, x^2$，求积公式均准确成立。因此可列出如下方程组：

$$\begin{cases} A + B + C = 4 \\ B + 3C = 8 \\ B + 9C = \frac{64}{3} \end{cases}$$

解得：$A = \frac{4}{9}, B = \frac{4}{3}, C = \frac{20}{9}$

Newton-Cotes求积公式

Newton-Cotes求积公式如下：

$$I_n = (b-a)\sum_{k=0}^{n} C_k^{(n)} f(x_k)$$

其中$c_k^{(n)}$称为Cotes系数。系数值如下：

n	$C_k^{(n)}$						
1	1/2	1/2					
2	1/6	4/6	1/6				
3	1/8	3/8	3/8	1/8			
4	7/90	32/90	12/90	32/90	7/90		
5	19/288	75/288	50/288	50/288	75/288	19/288	
6	41/840	216/840	27/840	272/840	27/840	216/840	41/840

例题：

求解：$\int_1^2 \ln x dx$

取$n = 1$,则$I_1 = (2-1) * [\frac{1}{2}f(1) + \frac{1}{2}f(2)]$,其中$f(x) = \ln x$

解得：$I_1 = 0.3467$

> $\ln 1 = 0; \ln 2 = 0.69314718$

复化梯形求积公式

$$\int_a^b f(x)dx \approx \frac{h}{2}[f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)] = T_n$$

其中，$h = \frac{b-a}{n}, x_k = a + kh(k = 0, 1, \ldots, n)$

例题：

求解积分$\int_0^1 \frac{x}{4+x^2}dx, n = 8$

首先，计算h, x_k：

$$h = (b-a)/n = \frac{1}{8}, x_k = a + kh = \frac{k}{8}$$

再列出n=8时的复化梯形公式：

$$T_8 = \frac{h}{2}[f(a) + 2\sum_{k=1}^{7} f(x_k) + f(b)]$$

展开：

$$T_8 = \frac{1}{16}f(a) + \frac{1}{16}f(b) + \frac{1}{8}f(\frac{1}{8}) + \frac{1}{8}f(\frac{2}{8}) + \ldots + \frac{1}{8}f(\frac{7}{8})$$

解得：$T_8 = 0.111402$

```
#include<iostream>
#include<cmath>

using namespace std;

double myFunction(double x){
    double result = x / (4 + pow(x,2));
    return result;
}

double getXk(double k){
    double xk = k / 8.0;
   return xk;
}

int main(){
    double a = 0,b=1;
```

```
    double h = (b-a) / 8.0;
    double ans = myFunction(a) + myFunction(b);
    double sum = 0;
    for(int k=1;k<8;k++){
        sum += myFunction(getXk(k));
    }
    ans += (sum * 2);
    ans *= h / 2;
    cout << ans << endl;
    return 0;
}
```

复化Simpson公式

$$\int_a^b f(x)dx \approx \frac{h}{6}[f(a) + 4\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}}) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)] = S_n$$

其中,$h = \frac{b-a}{n}, x_k = a + kh(k = 0, \dots, n)$

例题:

还是以上题为例，首先，先计算出h, x_k:

$$h = \frac{1}{8}; x_k = \frac{k}{8}$$

然后列出$n = 8$时的复化Simpson公式:

$$S_8 = \frac{h}{6}[f(a) + 4\sum_{k=0}^{7} f(x_{k+\frac{1}{2}}) + 2\sum_{k=1}^{7} f(x_k) + f(b)]$$

最后解得$S_8 = 0.111572$

```
#include<iostream>
#include<cmath>

using namespace std;

double myFunction(double x){
    double result = x / (4 + pow(x,2));
    return result;
}

double getXk(double k){
    double xk = k / 8.0;
   return xk;
}

int main(){
```

```
    double a = 0,b=1;
    double ans = (myFunction(a) + myFunction(b));
    double sum = 0;
    for(int k=0;k<8;k++){
        double xk = getXk(k + 0.5);
        sum += myFunction(xk);
    }
    sum *= 4;
    ans += sum;
    sum = 0;
    for(int k=1;k<8;k++){
        sum += myFunction(getXk(k));
    }
    sum *= 2;
    ans += sum;

    ans *= (1.0) / 48;
    cout << ans << endl;

    return 0;
}
```

复化Cotes公式

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n-1} \frac{h}{90}[7f(x_k) + 32f(x_{k+\frac{1}{4}}) + 12f(x_{k+\frac{1}{2}}) + 32f(x_{k+\frac{3}{4}}) + 7f(x_{k+1})] = C_n$$

其中,$h = \frac{b-a}{n}, x_k = a + kh(k = 0, \ldots, n)$

例题

还是以上题为例，先计算h, x_k

$$h = \frac{(b-a)}{n} = \frac{1}{8}, x_k = a + kh = \frac{k}{8}$$

列出$n = 8$时的Cotes公式

$$C_8 = \sum_{k=0}^{7} \frac{h}{90}[7f(x_k) + 32f(x_{k+\frac{1}{4}}) + 12f(x_{k+\frac{1}{2}}) + 32f(x_{k+\frac{3}{4}}) + 7f(x_{k+1})]$$

最终解得$T_8 = 0.111572$

```
#include<iostream>
#include<cmath>

using namespace std;
```

```
double myFunction(double x){
    double result = x / (4 + pow(x,2));
    return result;
}

double getXk(double k){
    double xk = k / 8.0;
   return xk;
}

int main(){
    double a = 0,b=1;
    double h = (b-a) / 8.0;
    double ans = 0;
    for(int k = 0;k < 8; k++){
        ans += (h/90) * (7 * myFunction(getXk(k)) + 32 * myFunction(getXk(
    }
    cout << ans << endl;
   return 0;
}
```