Numerical Methods HW #05

1. x=1과 h=0.001, 0.01, 0.1 에 대하여 $f(x) = e^{-x}\cos x$의 1계 도함수를 다음 방법에 의해서 구하고, 표 5.1 에 있는 오차정도를 이용하여 오차를 계산하라.

단, 오차정도는 $O(h^2)$로 하라.

(a) 전진차분

<Code>

```c
#include<stdio.h>
#include<math.h>
double func(double x)
{
	return exp(-x)*cos(x);
}
int main()
{
	double h,x;
	printf("구간 간격 h를 입력하시오 :");
	scanf("%lf",&h);
	x=1;
	int i,j,o;
	printf("오차의 정도를 입력하시오 :");
	scanf("%d",&o);
	printf("[차분표]\n");
	double df[o+2][o+2]={0};
	for(i=0;i<=o+1;i++)
	{
		df[0][i]=func(x+h*i);
		printf("%lf       ",df[0][i]);
	}
	printf("\n");
	for(j=1;j<=o+1;j++)
	{
		for(i=0;i<=o+1-j;i++)
		{
			df[j][i]=df[j-1][i+1]-df[j-1][i];
			printf("%lf       ",df[j][i]);
		}
		printf("\n");
	}
	double differential=(df[1][0]-0.5*df[2][0])/h;
	printf("df = %lf\n",differential);
	double error=h*h*df[3][0]/3;
	printf("error = %lf",error);
}
```

<Result>

h=0.001

```
구간 간격 h를 입력하시오 :0.1
오차의 정도를 입력하시오 :2
[차분표]
0.198766        0.150989        0.109140        0.072902
-0.047777       -0.041849       -0.036238
0.005928        0.005611
-0.000317
df = -0.507411
error = -0.000106
```

h=0.01

```
구간 간격 h를 입력하시오 :0.01
오차의 정도를 입력하시오 :2
[차분표]
0.198766        0.193714        0.188723        0.183794
-0.005052       -0.004991       -0.004929
0.000062        0.000061
-0.000000
df = -0.508318
error = -0.000000
```

h=0.1

```
구간 간격 h를 입력하시오 :0.1
오차의 정도를 입력하시오 :2
[차분표]
0.198766        0.150989        0.109140        0.072902
-0.047777       -0.041849       -0.036238
0.005928        0.005611
-0.000317
df = -0.507411
error = -0.000001
```

(b) 후진차분

```c
 #include<stdio.h>
#include<math.h>
double func(double x)
{
	return exp(-x)*cos(x);
}
int main()
{
	double h,x;
	printf("구간 간격 h를 입력하시오 :");
	scanf("%lf",&h);
	x=1;
	int i,j,o;
	printf("오차의 정도를 입력하시오 :");
	scanf("%d",&o);
	printf("[차분표]\n");
	double df[o+2][o+2]={0};
	for(i=0;i<=o+1;i++)
	{
		printf("%d                  ",i);
	}
	printf("\n");
	for(i=0;i<=o+1;i++)
	{
		df[0][i]=func(x-h*i);
		printf("%lf        ",df[0][i]);
	}
	printf("\n");
	for(j=1;j<=o+1;j++)
	{
		for(i=0;i<=o+1-j;i++)
		{
			df[j][i]=df[j-1][i]-df[j-1][i+1];
			printf("%lf       ",df[j][i]);
		}
		printf("\n");
	}
	double differential=(df[1][0]+df[2][0]/2)/h;
	printf("f' = %lf\n",differential);
	double error=df[3][0]/3;
	printf("error = %lf",error);
}
```

h=0.001

```
구간 간격 h를 입력하시오 :0.001
오차의 정도를 입력하시오 :2
[차분표]
0               1               2               3
0.198766        0.199275        0.199784        0.200294
-0.000509       -0.000509       -0.000510
0.000001        0.000001
-0.000000
f' = -0.508326
error = -0.000000
```

h=0.01

```
구간 간격 h를 입력하시오 :0.01
오차의 정도를 입력하시오 :2
[차분표]
0               1               2               3
0.198766        0.203880        0.209057        0.214295
-0.005114       -0.005176       -0.005239
0.000062        0.000062
-0.000000
f' = -0.508319
error = -0.000000
```

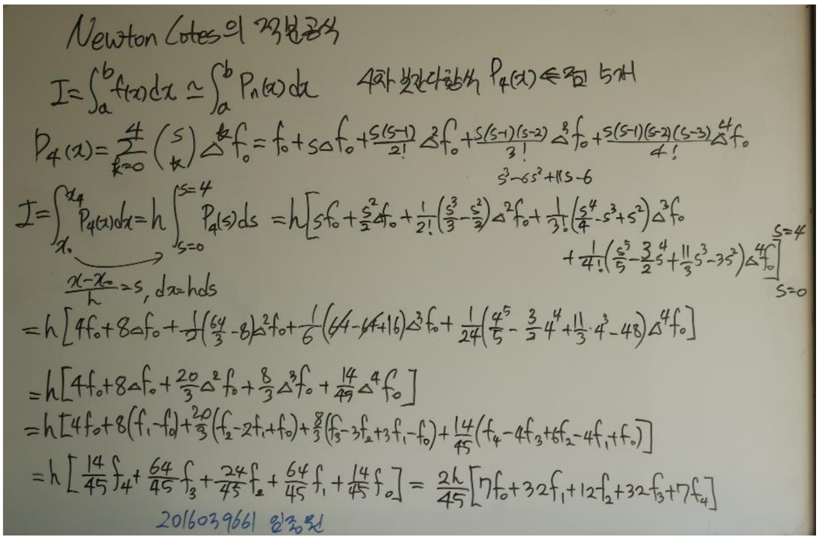
h=0.1

```
구간 간격 h를 입력하시오 :0.1
오차의 정도를 입력하시오 :2
[차분표]
0               1               2               3
0.198766        0.252728        0.313051        0.379809
-0.053962       -0.060323       -0.066759
0.006361        0.006436
-0.000075
f' = -0.507811
error = -0.000025
```

2. f(x) 를 4차 다항식으로 근사 시켰을 때 Newton-Cotes 적분공식 (5.36)을 유도하고 이 공식의 오차를 구하라. 유도한 공식을 이용하여 다음 적분을 계산하라.

$$\int_0^{\frac{\pi}{2}} e^x \sin x \, dx$$

(5.36)유도


Newton Cotes의 적분공식

$I = \int_a^b f(x)dx \simeq \int_a^b P_n(x)dx$ 4차 보간다항식 $P_4(x)$ ← 점 5개

$P_4(x) = \sum_{k=0}^{4} \binom{s}{k} \Delta^k f_0 = f_0 + s\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_0 + \frac{s(s-1)(s-2)(s-3)}{4!}\Delta^4 f_0$

$s^3-6s^2+11s-6$

$I = \int_{x_0}^{x_4} P_4(x)dx = h\int_{s=0}^{s=4} P_4(s)ds = h\left[sf_0 + \frac{s^2}{2}\Delta f_0 + \frac{1}{2!}\left(\frac{s^3}{3}-\frac{s^2}{2}\right)\Delta^2 f_0 + \frac{1}{3!}\left(\frac{s^4}{4}-s^3+s^2\right)\Delta^3 f_0 + \frac{1}{4!}\left(\frac{s^5}{5}-\frac{3}{2}s^4+\frac{11}{3}s^3-3s^2\right)\Delta^4 f_0\right]_{s=0}^{s=4}$

$\frac{x-x_0}{h} = s, \; dx = h\,ds$

$= h\left[4f_0 + 8\Delta f_0 + \frac{1}{2}\left(\frac{64}{3}-8\right)\Delta^2 f_0 + \frac{1}{6}(64-64+16)\Delta^3 f_0 + \frac{1}{24}\left(\frac{4^5}{5} - \frac{3}{2}4^4 + \frac{11}{3}4^3 - 48\right)\Delta^4 f_0\right]$

$= h\left[4f_0 + 8\Delta f_0 + \frac{20}{3}\Delta^2 f_0 + \frac{8}{3}\Delta^3 f_0 + \frac{14}{45}\Delta^4 f_0\right]$

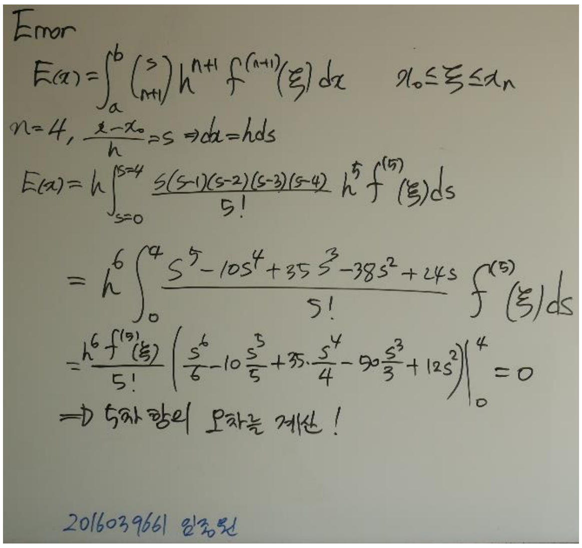
$= h\left[4f_0 + 8(f_1-f_0) + \frac{20}{3}(f_2-2f_1+f_0) + \frac{8}{3}(f_3-3f_2+3f_1-f_0) + \frac{14}{45}(f_4-4f_3+6f_2-4f_1+f_0)\right]$

$= h\left[\frac{14}{45}f_4 + \frac{64}{45}f_3 + \frac{24}{45}f_2 + \frac{64}{45}f_1 + \frac{14}{45}f_0\right] = \frac{2h}{45}\left[7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4\right]$

2016039661 김종현

$$\mathrm{I} = \frac{2\mathrm{h}}{45}[7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4]$$

오차항 계산


Error

$E(x) = \int_a^b \binom{s}{n+1} h^{n+1} f^{(n+1)}(\xi)\,dx \quad x_0 \le \xi \le x_n$

$n=4, \; \frac{x-x_0}{h} = s \Rightarrow dx = h\,ds$

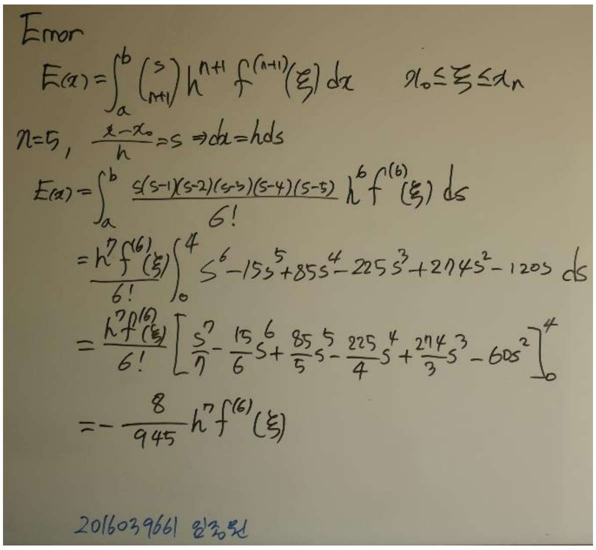
$E(x) = h\int_{s=0}^{s=4} \frac{s(s-1)(s-2)(s-3)(s-4)}{5!} h^5 f^{(5)}(\xi)\,ds$

$= h^6 \int_0^4 \frac{s^5 - 10s^4 + 35s^3 - 50s^2 + 24s}{5!} f^{(5)}(\xi)\,ds$

$= \frac{h^6 f^{(5)}(\xi)}{5!}\left(\frac{s^6}{6} - 10\frac{s^5}{5} + 35\cdot\frac{s^4}{4} - 50\frac{s^3}{3} + 12s^2\right)\Big|_0^4 = 0$

⇒ 5차 항의 오차는 계산 !

2016039661 김종현


Error

$E(x) = \int_a^b \binom{s}{n+1} h^{n+1} f^{(n+1)}(\xi)\,dx \quad x_0 \le \xi \le x_n$

$n=5, \; \frac{x-x_0}{h} = s \Rightarrow dx = h\,ds$

$E(x) = \int_a^b \frac{s(s-1)(s-2)(s-3)(s-4)(s-5)}{6!} h^6 f^{(6)}(\xi)\,ds$

$= \frac{h^7 f^{(6)}(\xi)}{6!}\int_0^4 s^6 - 15s^5 + 85s^4 - 225s^3 + 274s^2 - 120s\,ds$

$= \frac{h^7 f^{(6)}(\xi)}{6!}\left[\frac{s^7}{7} - \frac{15}{6}s^6 + \frac{85}{5}s^5 - \frac{225}{4}s^4 + \frac{274}{3}s^3 - 60s^2\right]_0^4$

$= -\frac{8}{945}h^7 f^{(6)}(\xi)$

2016039661 김종현

$$\mathrm{E(x)} = -\frac{8}{945}h^7 f^{(6)}(\xi)$$

<Code>

```
#include<stdio.h>
#include<math.h>
double func(double x)
{
        return exp(x)*sin(x);
}
int main()
{
        double a=0, b=M_PI/2;
        double h=(b-a)/4;
        double x[5]={0};
        int i;
        for(i=0;i<=4;i++)
        {
                x[i]=a+i*h;
        }
        double f[5]={0};
        for(i=0;i<=4;i++)
        {
                f[i]=func(x[i]);
                printf("%lf\n",f[i]);
        }
        double I=2*h/45*(7*(f[0]+f[4])+32*(f[1]+f[3])+12*f[2]);
        printf("%lf",I);
}
```

<Result>

```
f_0=0.000000
f_1=0.566744
f_2=1.550883
f_3=3.000934
f_4=4.810477
I=2.905094
```

참 값 = 2.905239 상대오차 = 0.0049%

3. Simpson의 1/3 법칙을 사용한 적응적분법을 이용하여 다음 적분을 구하라. 단, 허용 절대오차는 0.001 이내에 있도록 하라.

$$\int_0^2 e^{-10x^2}dx$$

Simpson 1/3 법칙을 사용 $\mathrm{I} = \sum_i \mathrm{I_i} = \sum_i \left[\frac{\mathrm{h}}{3}\left(f_{2i} + 4f_{2i+1} + f_{2(i+1)}\right)\right]$

1) h=1

구간(0,2)를 하나의 구간으로 본다. (점 3개, 2차 보간다항식)

이 때의 적분 값 $\mathrm{I_1}(0,2)$ =0.333393867

2) h=0.5

$$\mathrm{I_2}(0,1) = 0.221397566\,,\ \ \mathrm{I_2}(1,2) = 7.56677E - 06(0,2)$$

(0,2)구간 절대오차 = 0.111988734>0.001

3) h=0.25

$$\mathrm{I_3}(0,0.5) = 0.268594226,\ \ \mathrm{I_3}(0.5,1) = 0.008046388$$
$$,\mathrm{I_3}(1,1.5) = 3.83792E - 06,\ \ \mathrm{I_3}(1.5,2) = 1.41158\mathrm{E} - 11$$

(0,1)구간 절대오차 = 0.055243048>0.001

(1,2)구간 절대오차 = 3.72883E-06<0.001(허용 절대오차)

(0,1)구간에서 허용 절대오차를 만족하지 않으므로 (0,1)구간에서 위와 같은 계산을 계속한다.

4) h=0.125

$$I_4(0,0.25) = 0.206526781,\ \ I_4(0.25,0.5) = 0.066566191$$
$$,I_4(0.5,0.75) = 0.006923114,\ \ I_4(0.75,1) = 0.000231011$$

(0,0.5)구간 절대오차 = 0.004498746>0.001

(0.5,1)구간 절대오차 = 0.000892262<0.001(허용 절대오차)

(0,0.5)구간에서 계속 계산한다.

5) h=0.0625

$$I_5(0,0.125) = 0.118793911,\ \ I_5(0.125,0.25) = 0.087603297$$
$$,I_5(0.25,0.375) = 0.047640329,\ \ I_5(0.375,0.5) = 0.019105535$$

(0,0.25)구간 절대오차 = 0.000129573<0.001(허용 절대오차)

(0.25,0.5)구간 절대오차 = 0.000179672<0.001(허용 절대오차)

따라서 적분 값은 $I_3(1,2) + I_4(0.5,1) + I_5(0,0.5) = \ 0.280301035$

4. 한 놀이공원에서 정원 조경을 위한 분수대를 만들고자 한다. 분수대는 타원형으로 설계하고 가장 긴 쪽의 길이를 8m, 가장 짧은 쪽의 길이를 6m 가 되도록 한다. 이 분수대를 만드는 데 필요한 땅의 넓이를 수치적으로 계산하라. 단, 복합 Simpson의 1/3 법칙을 사용하라.

<Code>

```c
#include<stdio.h>
#include<math.h>
double func(double x)
{
	return 3*pow((1-pow(x,2)/16),0.5);
}
int main()
{
	int i,j;
	double a=0, b=4;
	double I[10]={0};
	double x[100][100]={0};
	double f[100][100]={0};
	for(i=1;i<=100;i++)
	{
		double h=(b-a)/pow(2,i);
		for(j=0;j<=pow(2,i);j++)
		{
			x[i-1][j]=a+h*j;
			f[i-1][j]=func(x[i-1][j]);
		}
		for(j=0;j<=pow(2,i-1)-1;j++)
		{
			I[i-1]=I[i-1]+h/3*(f[i-1][2*j]+4*f[i-1][2*j+1]+f[i-1][2*(j+1)]);
		}
		printf("%d %lf\n",i,I[i-1]);
		if(i!=1)
		{
			if(fabs(I[i-1]-I[i-2])/fabs(I[i-1])<0.0005) break;
		}
	}
	printf("Area of Ellipse=%lf",4*I[i-1]);
}
```

<Result> =>

```
1 8.928203
2 9.250785
3 9.363568
4 9.403193
5 9.417157
6 9.422085
7 9.423826
Area of Ellipse=37.695305
```

5. 어떤 저항체에 흐르는 전류는 $i(t) = (2-t)^2 + (2-t)\sin t^{1.5}$이고, 저항은 다음 식과 같은 전류의 함수이다.

R=5+i

t=0 에서 20초 사이의 평균 전압을 구하라.

Simpson 1/3법칙을 이용하여 풀었음.

<Code>

```c
#include<stdio.h>
#include<math.h>
double func(double t)
{
	return pow(2-t,2)+(2-t)*sin(pow(t,1.5));
}
int main()
{
	double a=0,b=20;
	int i,j,k;
	double Integral[30]={0};
	for(i=1;i<=30;i++)
	{
		int n=pow(2,i)+1;
		double h=(b-a)/pow(2,i);
		double t[n]={0};
		double I[n]={0};
		double R[n]={0};
		double V[n]={0};
		for(j=0;j<=pow(2,i);j++)
		{
			t[j]=a+h*j;
			I[j]=func(t[j]);
			R[j]=I[j]+5;
			V[j]=I[j]*R[j];
		}
		for(k=0;k<=pow(2,i-1)-1;k++)
		{
			Integral[i-1]=Integral[i-1]+h/3*(V[2*k]+4*V[2*k+1]+V[2*(k+1)]);
		}
		printf("%d %lf\n",i,Integral[i-1]);
		if(i!=1)
		{
			if(fabs((Integral[i-1]-Integral[i-2])/Integral[i-1])<0.0005) break;
		}
	}
	printf("Average Voltage = %lf",Integral[i-1]/20);
}
```

<Result>

```
1 373501.232826
2 341548.560885
3 389113.746217
4 382645.564239
5 390384.928469
6 388891.345819
7 388758.324439
Average Voltage = 19437.916222
```