

Metody Obliczeniowe w Nauce i Technice laboratorium

zestaw 4: kwadratury

1: Obliczyć całkę $\int_0^1 e^{-x^2} \sqrt{x} dx$ stosując kwadratury Newtona-Cotesa.

Do obliczenia całki oznaczonej na przedziale $[a,b]$ można wykorzystać np. wzór złożony trapezów, który jest przypadkiem kwadratury złożonej Newtona – Cotesa.

Przedział całkowania $[a,b]$ dzielony jest na m części o długości $h=(b-a)/m$.

Całkę wyznaczamy jako:

$$\int_a^b f(x) \approx \sum_{k=0}^{m-1} \frac{1}{2} h (f_k + f_{k+1}), \quad \text{gdzie } f_i = f(a + ih).$$

W programie wykorzystałem funkcję **trapzd** z biblioteki *Numerical Recipes*, wyznaczającą całkę z zadanej funkcji dla zadanego przedziału metodą trapezów.

Treść programu:

```
#include <stdio.h>
#include <math.h>

extern float trapzd(float (*func)(float), float a, float b, int n);

float func(float x)          // całkowana funkcja
{
    return exp(-x*x)*sqrt(x);
}

void main(void)
{
    float a = 0, b = 1;      // granice przedziału
    int n;

    for (n=1; n<=20; n++)
    {
        printf("n=%d I=%f\n", n, trapzd(&func, a, b, n));
    }
}
```

Wynik działania programu:

```
n=1 I=0.183940
n=2 I=0.367318
n=3 I=0.424447
n=4 I=0.443479
n=5 I=0.449964
n=6 I=0.452199
n=7 I=0.452975
n=8 I=0.453246
n=9 I=0.453340
n=10 I=0.453374
n=11 I=0.453385
n=12 I=0.453390
n=13 I=0.453391
n=14 I=0.453392
n=15 I=0.453391
n=16 I=0.453392
n=17 I=0.453393
n=18 I=0.453394
n=19 I=0.453400
n=20 I=0.453416
```

Błąd takiej kwadratury określony jest wzorem:

$$E(f) = -\frac{(b-a)^3}{12m^2} f^{(2)}(x)$$

Czyli np. dla podziału na 10 przedziałów, błąd całkowania naszej funkcji wynosi

$$-\frac{1}{1200} e^{-x^2} \left(4x^{\frac{5}{2}} - 4x^{\frac{1}{2}} - \frac{1}{4} x^{-\frac{3}{2}} \right).$$

2a: Obliczyć całkę $\int_0^1 e^{-x^2} \sqrt{x} dx$ stosując kwadraturę Gaussa-Laguerre'a

Kwadratury Gaussa – Laguerre'a stosowane są do obliczania całek dla przedziału jednostronnie nieskończonego zgodnie z wzorem:

$$\int_0^{\infty} e^{-x} f(x) dx \approx \sum_{k=0}^N A_k f(x_k),$$

gdzie współczynniki A_k określone są wzorem:

$$A_k = \frac{((N+1)!)^2}{L_{N+1}(x_k) L_{N+2}(x_k)},$$

zaś x_k są zerami wielomianu Laguerre'a $L_{N+1}(x)$.

Wielomiany Laguerre'a określone są wzorem:

$$L_n = (-1)^n e^x \frac{d^n}{dx^n} (x^n e^{-x})$$

Całkę oznaczoną dla przedziału $[a, b]$ można wyznaczyć jako:

$$\int_a^b f(x) dx = \int_a^{\infty} f(x) dx - \int_b^{\infty} f(x) dx, \quad \int_a^{\infty} f(x) dx = \int_0^{\infty} f(x-a) dx$$

Treść programu:

```
#include <math.h>
#include <stdio.h>

/*
 * Funkcja obliczająca całkę oznaczoną na przedziale [a,b] metoda
 * Gaussa-Laguerre'a
 */
double gaulag(float (*func)(float), float a, float b, int n)
{
    float x[4][5] = { { 0.585786, 3.414214, 0.0, 0.0, 0.0 },
                    { 0.415775, 2.294280, 6.289945, 0.0, 0.0 },
                    { 0.322548, 1.745761, 4.536620, 9.395071, 0.0 },
                    { 0.263560, 1.413403, 3.596426, 7.085810, 12.640801 } };
    float A[4][5] = { { 0.853553, 0.146447, 0.0, 0.0, 0.0 },
                    { 0.711093, 0.278518, 0.010389, 0.0, 0.0 },
                    { 0.603154, 0.357419, 0.038888, 0.000539, 0.0 },
                    { 0.521756, 0.398667, 0.075942, 0.003612, 0.000032 } };

    float res = 0.0;
    int i;

    for (i=0; i<=n; i++)
        res += (A[n-1][i] * (*func)(x[n-1][i] + a)) / exp(-x[n-1][i]);
    for (i=0; i<=n; i++)
        res -= (A[n-1][i] * (*func)(x[n-1][i] + b)) / exp(-x[n-1][i]);
}
```

```

    return res;
}

/*
 * Calkowana funkcja
 */
float func(float x)
{
    return exp(-x*x) * sqrt(x);
}

/*
 * Czesc glowna programu
 */
void main(void)
{
    float a = 0.0, b = 1.0;           // granice przedzialu
    int n;

    for(n=1; n<=4; n++)
        printf("n=%2d I=%f\n", n, gaulag(&func, a, b, n));
}

```

Wynik działania programu:

```

n=1 I=0.676569
n=2 I=0.433368
n=3 I=0.386299
n=4 I=0.427315

```

Błąd kwadratury Gaussa – Laguerre’a wyrażany jest wzorem:

$$E(f) = \frac{((N+1)!)^2}{(2N+2)!} f^{(2N+2)}(h), \quad h \in (0, \infty).$$

W naszym przypadku $f(x) = e^{x-x^2} \sqrt{x}$, więc błąd wynosi:

Dla N=1:

$$\frac{1}{96} (256x^8 - 512x^7 - 640x^6 + 1024x^5 + 112x^4 - 64x^3 - 24x^2 + 24x - 15) \frac{e^{x-x^2}}{x^{\frac{7}{2}}}$$

Dla N=2:

$$\frac{1}{1280} \left(4096x^{12} - 12288x^{11} - 21504x^{10} + 66560x^9 + 15360x^8 - 62208x^7 + 1984x^6 + 4032x^5 \right) \frac{e^{x-x^2}}{x^{\frac{11}{2}}}$$

itd...

2b: Obliczyć całkę $\int_{-\infty}^{\infty} e^{-x^2} x^2 dx$ stosując kwadraturę Gaussa-Hermite’a.

Kwadratury Gaussa – Hermite’a stosowane są do obliczania całek dla przedziału nieskończonego zgodnie z wzorem:

$$\int_0^{\infty} e^{-x^2} f(x) dx \approx \sum_{k=0}^N A_k f(x_k),$$

gdzie współczynniki A_k określone są wzorem:

$$A_k = \frac{2^{N+2} (N+1)!}{H_{N+1}(x_k) H_{N+2}(x_k)},$$

x_k są zerami wielomianu Hermite'a $H_{N+1}(x)$, zaś same wielomiany Hermite'a określone są wzorem:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}.$$

W programie posłużyłem się funkcją **gauher** z biblioteki *Numerical Recipes*, wyznaczającą współczynniki oraz węzły kwadratury Gaussa – Hermite'a.

Treść programu:

```
#include "nr.h"
#include "nrutil.h"
#include <math.h>

extern void gauher(float x[], float w[], int n);

/*
 * Calkowana funkcja (bez e^(-x^2))
 */
float func(float x)
{
    return x*x;
}

#define MAX_N 15

/*
 * Czesc glowna programu
 */
void main(void)
{
    int n;
    int i;
    float *x, *w, xx;

    x = vector(1, MAX_N);
    w = vector(1, MAX_N);

    for(n=3; n<=MAX_N; n++)
    {
        gauher(x, w, n);
        xx = 0.0;
        for(i=1; i<=n; i++)
            xx += (w[i]*func(x[i]));
        printf("n=%d: I=%f\n", n, xx);
    }
}
```

Wynik działania programu:

```
n=3: I=0.886227
n=4: I=0.886227
```

Błąd kwadratury Gaussa – Hermite'a określony jest wzorem:

$$E(f) = \frac{(N+1)! \sqrt{p}}{2^{N+1} (2N+2)!} f^{(2N+2)}(h), \quad h \in (-\infty, \infty)$$

W naszym przypadku $f(x)=x^2$. Dla $N=1$ $f^{(2N+1)}(x)=f^{(4)}(x)=0$, a więc błąd wynosi 0.

W rzeczywistości wartość tej całki wynosi 0.886226 a błąd jest najprawdopodobniej wynikiem zaokrąglenia.