Metody obliczeniowe w nauce i technice

SPRAWOZDANIE

Ćwiczenie 5
CAŁKOWANIE NUMERYCZNE

Łukasz Gajewski

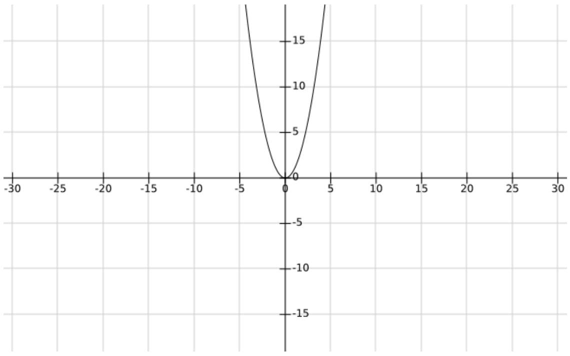
I. Cel ćwiczenia

- Napisać wlasna metode calkowania (np. metode prostokątow lub trapezow.)
- Wybrac dwie funkcje dla ktorych latwo wyliczyc wartosc całki analitycznie.
  - gladka typu wielomian (np x^2)
  - ciekawsza, zawierajaca osobliwosc (np sqrt(x))
- Narysowac wykresy tych funkcji.
- Dla w/w wybranych dwoch przykladow porównac metody całkowania w zadanym przedziale zawierajacym te osobliwosc (np. dla podanych przykladow [0,1]). Porownac uzyskane wyniki oraz **ilosc potrzebnych obliczen funkcji podcalkowej, aby uzyskać okresloną dokladność (blad)** dla nastepujacych metod:
  - wlasnej
  - czterech metod z GSL: nieadaptacyjnej (gsl_integration_qng), adaptacyjnej (gsl_integration_qag), adaptacyjnej z osobliwosciami (gsl_integration_qags) oraz adaptacyjnej ze znanymi osobliwosciami (gsl_integration_qagp)
  - metodami quad oraz quadgk dostepnymi w octave . Tutorial do octave.
- Metody dla funkcji oscylacyjnych z GSL. Porownac wyniki oraz ilosc wywolan funkcji podcałkowej (dla zadanego błędu) dla całki funkcji a* cos(a*x) w przedziale [0..9/2*PI]. Zasady:
  - Nalezy porównać funkcje z GSL: nieadaptacyjną (gsl_integration_qng), adaptacyjną (gsl_integration_qag) oraz dla funkcji oscylacyjnych (gsl_integration_qawo).
  - Nalezy zacząć od a=1, policzyc całkę analitycznie, a nastepnie sprawdzić otrzymany wynik dla wszystkich sposobów, aby zweryfikować poprawność uzycia funkcji.
  - zwiekszyc "a" i obserwowac zmiane potrzebnych ilosci wywołań funkcji (przedziałów). Narysowac wykres ilości potrzebnych przedziałów w zaleznosci od a (np dla wybranych a z przedzialu [1..100].)
- Na wybranym przykładzie pokazac przyklad wykorzystania metod całkowania do nieskonczonosci z GSL. Narysowac wykres funkcji, dla ktorej obliczana jest calka.
- Wyniki zawrzeć w sprawozdaniu.

II. Metoda trapezów – implementacja

```
double integral_trapeze(gsl_function fun, double a, double b, double exact_val, int* neval_r) {
    double (*foo)(double, void*);
    foo = fun.function;
    void* params = fun.params;
    int neval = 0;
    double result;
    while(fabs(exact_val-result) > EPS_ABS || fabs(exact_val-result)/exact_val > EPS_REL) {
        result = 0;
        neval++;
        double h = (b-a)/neval;
        double tmp_a = a;
        int intervals;
        for(intervals=0; intervals<neval; intervals++) {
            double h1 = foo(tmp_a, params);
            double h2 = foo(tmp_a+h, params);
            double area =  + (h1+h2)*(tmp_a+h - tmp_a)/2; //(a+b)*h/2
            result += area;
            tmp_a += h;
        }
    }
    *neval_r = neval;
    return result;
}
```

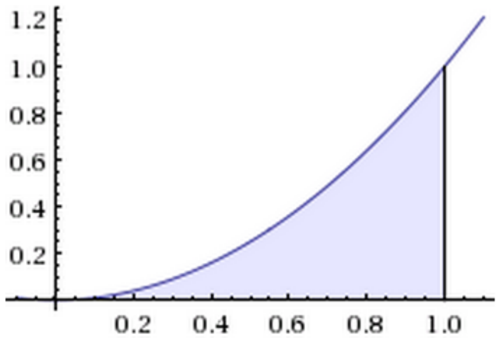
III. Wybrane funkcje proste – gładka, osobliwa



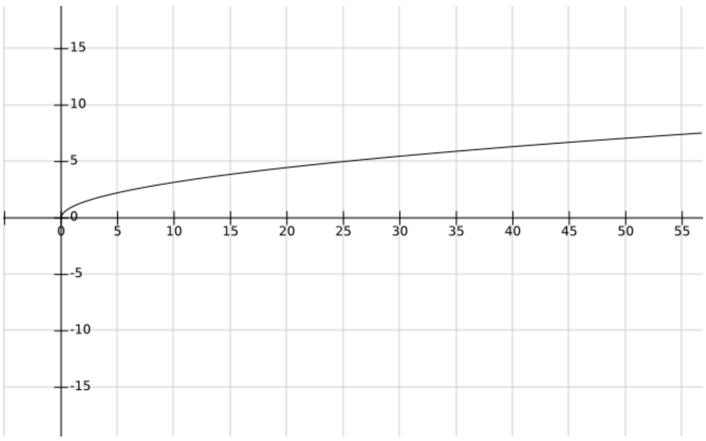
Definite integral:

$$\int_0^1 x^2\, dx = \frac{1}{3} \approx 0.33333$$

Visual representation of the integral:



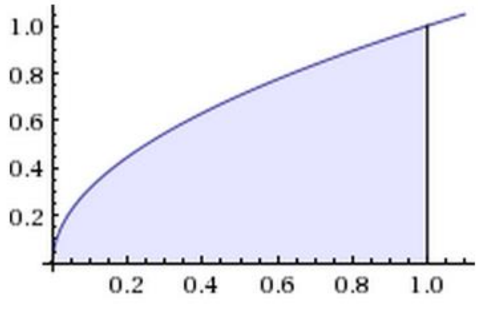
$$f(x) = x^2$$



Definite integral:

$$\int_0^1 \sqrt{x}\, dx = \frac{2}{3} \approx 0.66667$$

Visual representation of the integral:



$$f(x) = \sqrt{x}$$

IV. Porównanie metod całkowania dla wyżej wymienionych funkcji prostych

```
[Trapezoidal Rule]         x2    neval: 130, result: 0.33334
[gsl_integration_qng]      x2    neval:  21, result: 0.33333
[gsl_integration_qag]      x2    neval:   1, result: 0.33333
[gsl_integration_qags]     x2    neval:   1, result: 0.33333
[gsl_integration_qagp]     x2    neval:   1, result: 0.33333

[Trapezoidal Rule]      sqrt_x   neval: 753, result: 0.66666
[gsl_integration_qng]   sqrt_x   neval:  43, result: 0.66667
[gsl_integration_qag]   sqrt_x   neval:   5, result: 0.66667
[gsl_integration_qags]  sqrt_x   neval:   3, result: 0.66667
[gsl_integration_qagp]  sqrt_x   neval:   3, result: 0.66667
```

gdzie: $EPS_{ABS} = 10^{-5}$, $EPS_{REL} = 10^{-3}$

$\boldsymbol{neval}$ − *ilosc obliczen funkcji podcalkowej do uzyskania odpowiedniego błędu*

$\boldsymbol{result}$ − *wynik otrzymany przez wywolanie konkretnej metody*

V. Całkowanie funkcji za pomocą narzędzia OCTAVE

```
octave:19> [I1, ierr1, nfun1, err1] = quad(f1, 0, 1)
I1 =  0.33333
ierr1 = 0
nfun1 =  21
err1 =    3.7007e-15
octave:20> [I2, ierr2, nfun2, err2] = quad(f2, 0, 1)
I2 =  0.66667
ierr2 = 0
nfun2 =  231
err2 =    7.4015e-16
octave:21> quadgk(f1,0,1)
ans =  0.33333
octave:22> quadgk(f2,0,1)
ans =  0.66667
```

gdzie: $f_1(x) = x^2$, $f_2(x) = \sqrt{x}$

$I_{1,2}$ − *wartosc calki w przedziale* $[a, b]$

$ierr_{1,2}$ − *kod bledu* $(0 - sukces)$

$nfun_{1,2}$ − *liczba ewaluacji*

$err_{1,2}$ − *oszacowanie bledu w rozwiazaniu*

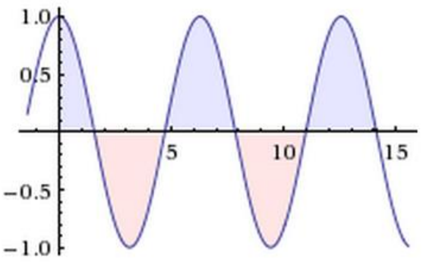
$ans_{1,2}$ − *wartosc calki w przedziale* $[a, b]$

VI. Funkcja oscylacyjna – porównanie wyników za pomocą metod z biblioteki GSL

Definite integral:

$$\int_0^{\frac{9\pi}{2}} \cos(x)\, dx = 1$$

Visual representation of the integral:



$$f(x) = a\cos(ax)$$

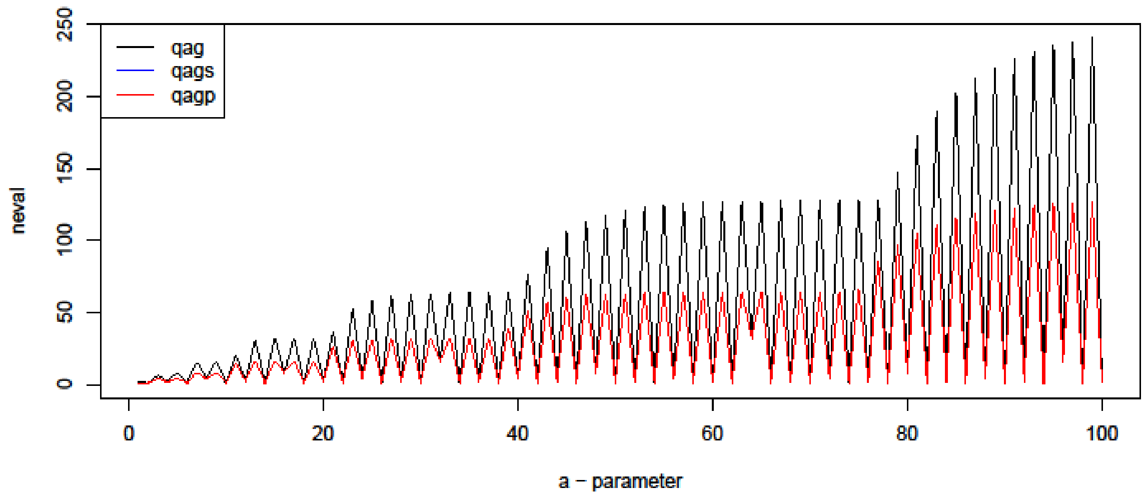
$$gdzie\ a = 1$$

Poniżej znajdują się wyniki dla a=1. Pozostałe wartości dla a z przedziału [1..100] umieszczone zostały w pliku tekstowym ***f_osc.txt***

```
[gsl_integration_qag]   acos_ax,1       neval:   2, result: 1.00000
[gsl_integration_qags]  acos_ax,1       neval:   1, result: 1.00000
[gsl_integration_qagp]  acos_ax,1       neval:   1, result: 1.00000
[gsl_integration_qag]   acos_ax,1       neval:   2, result: 1.00000
[gsl_integration_qags]  acos_ax,1       neval:   1, result: 1.00000
[gsl_integration_qagp]  acos_ax,1       neval:   1, result: 1.00000
[gsl_integration_qawo]  acos_ax,1       neval:   1, result: 1.00000
```

Wykres przedstawiający zależność między liczbą a z przedziału [1..100], a ilością potrzebnych przedziałów (iteracji wywołań)

Bardziej czytelna wersja do wglądu w ***osc_neval.pdf***

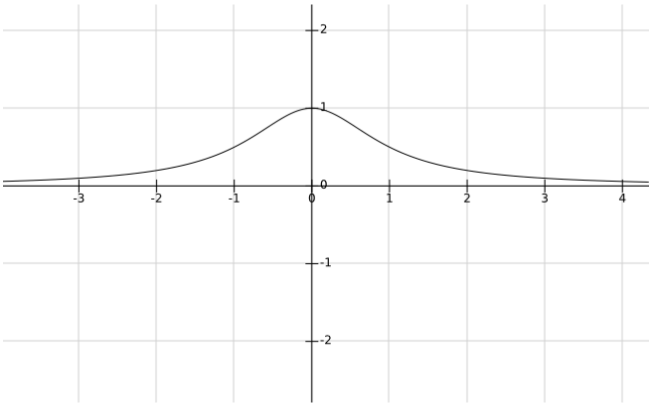


VII. Przykład wykorzystania metody całkowania do nieskończoności

- **implementacja**

```c
// gsl_integration_qagiu
gsl_function fun5;
fun5.function = &tan_derivative;
fun5.params = NULL;
gsl_integration_workspace* workspace = gsl_integration_workspace_alloc(512);
gsl_integration_qagiu(&fun5, 0.004, EPS_ABS, EPS_REL, 512, workspace, &result, &error);
printf("[gsl_integration_qagiu] 1/(x^2+1)\tneval: %3zu, result: %3.5lf\n", workspace->size, result);
gsl_integration_workspace_free(workspace);
```

- **własności funkcji – wykres, wartość całki**



Definite integral:

$$\int_0^{\infty} \frac{1}{x^2+1}\, dx = \frac{\pi}{2} \approx 1.5708$$

Indefinite integral:

$$\int \frac{1}{x^2+1}\, dx = \tan^{-1}(x) + \text{constant}$$

```
[gsl_integration_qagiu] 1/(x^2+1)        neval:   1, result: 1.56680
```

Łukasz Gajewski