Porównanie dokładności złożonych kwadratur Newtona-Cotesa

Agnieszka Wrzos

10 czerwca 2019

1 Wstęp

W projekcie zostaną przeze mnie omówione metody przybliżonego numerycznego obliczania całek. Porównane zostaną trzy złożone kwadratury Newtona-Cotesa: prostokątów, trapezów oraz Simpsona. Zastanowimy się czy całkowanie numeryczne jest w stanie zastąpić klasyczne metody analityczne. W jakich przypadkach warto jest użyć programu *Octave* zamiast tradycyjnego liczenia na kartce.

2 Skrypty w programie *Octave*

Poniżej przedstawione zostaną skrypty użyte do zbadania funkcji przy pomocy trzech złożonych kwadratur Newtona-Cotesa. Obliczeia numeryczne zostały wykonane przy użyciu programu *Octave*.

Kwadraturę złożoną Newtona-Cotesa realizuje się w praktyce w następujący sposób:

- przedział całkowania jest dzielony na pewną liczbę podprzedziałów
- w każdym podprzedziale stosuje się kwadraturę Newtona-Cotesa niskiego rzędu (najczęściej z wielomianem interpolującym stopnia pierwszego lub drugiego) i sumuje otrzymane wartości przybliżone całek.

2.1 Skrypt programu pomocniczego

Pomocniczo został utworzony plik funkcyjny, w którym należy zadeklarować funkcję. Następnie w odpowiedni sposób należy odwołać się do niej w następnych skryptach. Plik po wpisaniu przykładowej funkcji e^{-x^2} .

```
function y=f(x);
y=exp(-x.^2);
endfunction
```

2.2 Skrypt głównego programu

Niżej pokazany skrypt jest pomocny przy wyliczaniu funkcji omówionych w tym projekcie. Poprzez wpisanie odpowiednich cyfr wybieramy z menu interesującą nas funkcję i metodę. Jest także opcja obliczenia całki z dowolnego wielomianu do piątego stopnia po podaniu współczynników. Następnie podajemy przedział i liczbę podprzedziałów (tutaj jest dowolność, czy badamy daną funkcję, na przedziałach i o tej samej liczbie podprzedziałów co w dalej omówionych przykładach).

```
%program główny
format long
%Wprowadzanie danych z konsoli:
printf('Program liczacy calki w oparciu o metody numeryczne. \n');
```

```
printf('Wybierz funkcje podcalkowa poprzez wpisanie odpowiedniej cyfry: \n');
printf('1 - f(x)=7sin(x^2) \n');
printf('2 - f(x)=ln((1/16)x+1) \n');
printf('3 - f(x)=x^5+0.5 \n');
printf('4 - f(x)=(x^2)/sqrt(1-x^2) \n');
printf('5 - f(x)=a*x^5+b*x^4+c*x^3+d*x^2+e*x+f \n');
n=input('Numer funkcji podcalkowej: ');
%sprawdzenie jaka podana została cyfra i przypisanie odpowiedniej
%funkcji do programu
if(n==1)
g=@(x) 7.*sin(x.^2);
elseif(n==2)
g=@(x) log((1./16).*x+1);
elseif(n==3)
g=@(x) x.^5+0.5;
elseif(n==4)
g=@(x)(x.^2)./sqrt(1-x.^2)
elseif(n==5)
%wprowadzenie danych z konsoli w przypadku wybrania wielomianu
printf('Podaj wspolczynniki wielomianu \n');
printf('Wielomian a1*x.^5+a2*x.^4+a3*x.^3+a4*x.^2+a5*x+a6 \n');
a1=input('a= ');
a2=input('b= ');
a3=input('c= ');
a4=input('d= ');
a5=input('e= ');
a6=input('f= ');
g=@(x) a1*x.^5+a2*x.^4+a3*x.^3+a4*x.^2+a5*x+a6;
%komunikat w przypadku podania zlej cyfry
else
printf('Podano zla cyfre \n')
endif
%wybor metody przy uzyciu konsoli
printf('Wybierz metode poprzez wpisanie odpowiedniej cyfry: \n');
printf('1 - Zlozona Metoda Prostokatow \n');
printf('2 - Zlozona Metoda Trapezow \n');
printf('3 - Zlozona Metoda Simpsona \n');
m=input('Numer metody: ');
%sprawdzenie wpisanej cyfry i uruchomienie odpowiedniej metody
if(m==1)
%Wprowadzanie danych z konsoli:
printf('Podaj: \n');
a=input('Poczatek przedzialu calkowania: ');
b=input('Koniec przedzialu calkowania: ');
N=input('Podaj liczbe podprzedzialow: ');
%Dlugosc podprzedzialow h:
```

```
h=(b-a)./N;
%obliczanie sumy uzytej we wzorze:
sumac=0;
for i=1:N
i2=i-0.5;
wart=(g(a+i2.*h));
sumac=sumac+wart;
endfor
%wzor na zlozona kwadrature prostokatow:
calka=h.*sumac;
%zwrocenie wartosci calki w konsoli:
printf('Wartosc calki: %d \n',calka)
elseif(m==2)
%Wprowadzanie danych z konsoli:
printf('Podaj: \n');
a=input('Poczatek przedzialu calkowania: ');
b=input('Koniec przedzialu calkowania: ');
N=input('Podaj liczbe podprzedzialow: ');
%Dlugosc podprzedzialow h:
h=(b-a)./N;
%obliczanie sumy uzytej we wzorze:
sumac=0;
for i=1:(N-1)
a2=a+i.*h;
wart=(g(a2));
sumac=sumac+wart;
endfor
%wzor na zlozona kwadrature prostokatow:
calka=h.*((g(a)+g(b))./2+sumac);
%zwrocenie wartosci calki w konsoli:
printf('Wartosc calki: %d \n',calka)
elseif(m==3)
%Wprowadzanie danych z konsoli:
printf('Podaj: \n');
a=input('Poczatek przedzialu calkowania: ');
b=input('Koniec przedzialu calkowania: ');
N=input('Podaj liczbe podprzedzialow: ');
%Dlugosc podprzedzialow h:
h=(b-a)./N;
%obliczanie pierwszej sumy uzytej we wzorze:
sumac1=0;
for i=1:(N-1)
a2=a+i.*h;
wart=(g(a2));
sumac1=sumac1+wart;
endfor
```

```
%obliczanie drugiej sumy uzytej we wzorze:
sumac2=0;
for i=0:(N-1)
a2=a+i.*h+h./2;
wart=(g(a2));
sumac2=sumac2+wart;
endfor
%wzor na zlozona kwadrature prostokatow:
calka=h./3.*((g(a)+g(b))./2+sumac1+2.*sumac2);
%zwrocenie wartosci calki w konsoli:
printf('Wartosc calki: %d \n',calka)
%komunikat w przypadku podania zlej cyfry
else
printf('Podano zla cyfre \n')
endif
```

2.3 Skrypt - Złożona Metoda Prostokątów

Algorytm prostokątów, jest najprostszym z omawianych. Interpolacja przeprowadzana wielomianem stopnia 0 (funkcją stałą). Jako wartość do obliczeń brana wartość funkcji w środku przedziału interpolowanego (stąd alternatywna nazwa: metoda punku środkowego Naśladuje bezpośrednio konstrukcję su całkowitych Riemanna. Funkcja całkowana jest aproksymowana funkcją stałą w kolejnych równych podprzedziałach, na jakie podzielono przedział całkowania $[a, b]$. Wzór na wartość przybliżoną całki oznaczonej Riemanna $I(f)$ na danym przedziale $[a, b]$ o N podprzedziałach jest następujący:

$$I(f) \approx h \sum_{i=1}^{N-1} f(x_i)$$

```
format long
%Wprowadzanie danych z konsoli:
printf('Podaj: \n');
a=input('Poczatek przedzialu calkowania: ');
b=input('Koniec przedzialu calkowania: ');
N=input('Podaj liczbe podprzedzialow: ');
%Dlugosc podprzedzialow h:
h=(b-a)./N;
%obliczanie sumy uzytej we wzorze:
sumac=0;
for i=1:N
i2=i-0.5;
wart=(f(a+i2.*h));
sumac=sumac+wart;
endfor
%wzor na zlozona kwadrature prostokatow:
```

```
calka=h.*sumac;
%zwrocenie wartosci calki w konsoli:
printf('Wartosc calki: %d \n',calka)
```

2.4 Skrypt - Złożona Metoda Trapezów

Zgodnie z algorytmem należy podzielić przedział całkowania na liczbę N (wyrażającą liczbę podprzedziałów), o tej samej długości $h = \frac{a+b}{N}$. W każdym z podprzedziałów funkcja całkowana jest aproksymowana przez interpolację funkcją liniową. Wtedy wzór na wartość przybliżoną całki oznaczonej Riemanna przyjmuje postać:

$$I(f) \approx h\frac{(f(a) + f(b))}{2} + \sum_{i=1}^{N-1} f(a + ih)$$

```
format long
%Wprowadzanie danych z konsoli:
printf('Podaj: \n');
a=input('Poczatek przedzialu calkowania: ');
b=input('Koniec przedzialu calkowania: ');
N=input('Podaj liczbe podprzedzialow: ');
%Dlugosc podprzedzialow h:
h=(b-a)./N;
%obliczanie sumy uzytej we wzorze:
sumac=0;
for i=1:(N-1)
a2=a+i.*h;
wart=(f(a2));
sumac=sumac+wart;
endfor
%wzor na zlozona kwadrature prostokatow:
calka=h.*((f(a)+f(b))./2+sumac);
%zwrocenie wartosci calki w konsoli:
printf('Wartosc calki: %d \n',calka)
```

2.5 Skrypt - Metoda Simpsona

Stosując interpolację funkcją kwadratową otrzymujemy metodę Simpsona (parabol). W algorytmie Simpsona przedział całkowania $[a, b]$ jest dzielony na pewną liczbę N równych podprzedziałów o długości $h = \frac{b-a}{N}$, przy czym N jest z założenia liczbą parzystą.

$$I(f) \approx \frac{h}{3}(\frac{(f(a) + f(b))}{2} + \sum_{i=1}^{N-1} f(a + ih) + 2\sum_{i=0}^{N-1} f(a + ih + \frac{h}{2}))$$

```
format long
%Wprowadzanie danych z konsoli:
printf('Podaj: \n');
a=input('Poczatek przedzialu calkowania: ');
b=input('Koniec przedzialu calkowania: ');
N=input('Podaj liczbe podprzedzialow: ');
%Dlugosc podprzedzialow h:
h=(b-a)./N;
%obliczanie pierwszej sumy uzytej we wzorze:
sumac1=0;
for i=1:(N-1)
a2=a+i.*h;
wart=(f(a2));
sumac1=sumac1+wart;
endfor
%obliczanie drugiej sumy uzytej we wzorze:
sumac2=0;
for i=0:(N-1)
a2=a+i.*h+h./2;
wart=(f(a2));
sumac2=sumac2+wart;
endfor
%wzor na zlozona kwadrature prostokatow:
calka=h./3.*((f(a)+f(b))./2+sumac1+2.*sumac2);
%zwrocenie wartosci calki w konsoli:
printf('Wartosc calki: %d \n',calka)
```

2.6 Program do rysowania wykresów funkcji

Prosty program odwołujący się jak w poprzednich przypadkach do programu pomocniczego. Wystarczy wpisać odpowiedni przedział $[a, b]$, w tym przypadku, dla przykładu został wpisany $[-1, 2]$

```
a=-1;
b=2;
x=linspace(a,b,100);
plot(x,f(x),'m-');
grid on
```

3 Wyniki testów numerycznych

Zbadane zostaną cztery funkcje:

- $y = 7sin(x^2)$ na przedziale $[-10, 10]$
- $y = log(1/16x + 1)$ na przedziale $[0, 500]$

- $y = x^5 + 0.5$ na przedziale $[-6, 6]$
- $y = \frac{x^2}{\sqrt{1-x^2}}$ na przedziale $[-1, 1]$

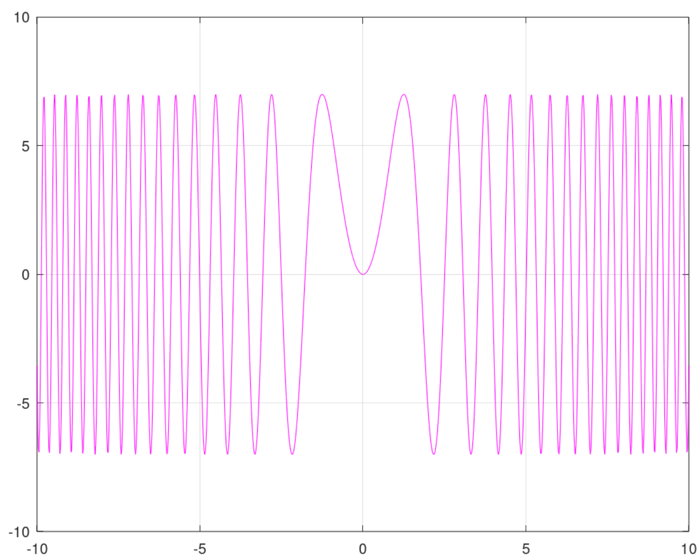
W celu łatwiejszej interpretacji dane funkcje zostaną najpierw przedstawione na wykresie (na odpowiednich przedziałach). Następnie zaprezentowane zostaną wyniki obliczeń numerycznych i krótkie wnioski wynikające ze zbadania danej funkcji podcałkowej.

3.1 Przykład I

Za funkcję podcałkową przyjmujemy $y = 7sin(x^2)$. Obliczona zostanie na przedziale $[-10, 10]$.

3.1.1 Wykres funkcji

Poniżej przedstawiony jest wykres funkcji $y = 7sin(x^2)$ na przedziale $[-10, 10]$.



3.1.2 Wyniki obliczeń numerycznych

W poniższej tabeli zostały zaprezentowane wyniki.

N (l. podprzedziałów)	Metoda prostokątów	Metoda Trapezów	Metoda Simpsona
1	0	-70.8912	-23.6304
2	-18.5292	-35.4456	-24.168
6	25.5253	-37.3634	4.56241
12	12.4892	-5.91905	6.35312
24	34.6274	3.28508	24.18
48	-0.0207733	18.9562	6.3049
100	7.45734	9.31719	8.07729
500	8.167355	8.20393	8.17131
1000	8.16735	8.17946	8.17139
1000000	8.17139	8.17139	8.17139

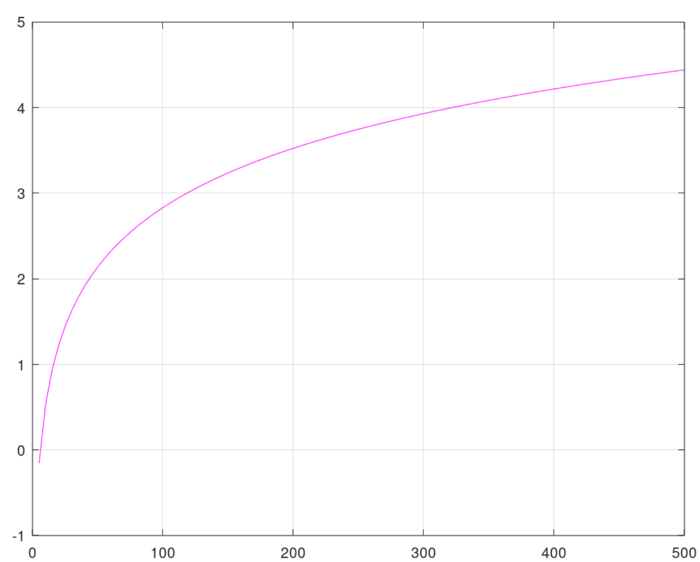
3.1.3 Wnioski

Ze względu na złożoność funkcji $y = 7sin(x^2)$ widać problemy, przy mniejszej liczbie węzłów. Utrudnieniami są też okresowość funkcji i jej zmienna monotoniczność. Z wszystkich trzech sposobów najmniej radzi sobie tutaj metoda prostokątów, zaś zdecydowanie najlepsza okazuje się metoda Simpsona.

3.2 Przykład II

Tym razem obliczymy całkę z funkcji $y = ln(\frac{x}{16} + 1)$ na przedziale $[0, 500]$.

3.2.1 Wykres funkcji



3.2.2 Wyniki obliczeń numerycznych

N (l. podprzedziałów)	Metoda prostokątów	Metoda Trapezów	Metoda Simpsona
1	1405.45	868.38	1226.43
2	1343.07	1136.92	1274.35
6	1303.39	1265.79	1290.86
12	1295.89	1284.59	1292.12
24	1293.35	1290.24	1292.31
48	1292.6	1291.79	1292.33
100	1292.4	1292.21	1292.34
1000	1292.34	1292.33	1292.34

3.2.3 Wnioski

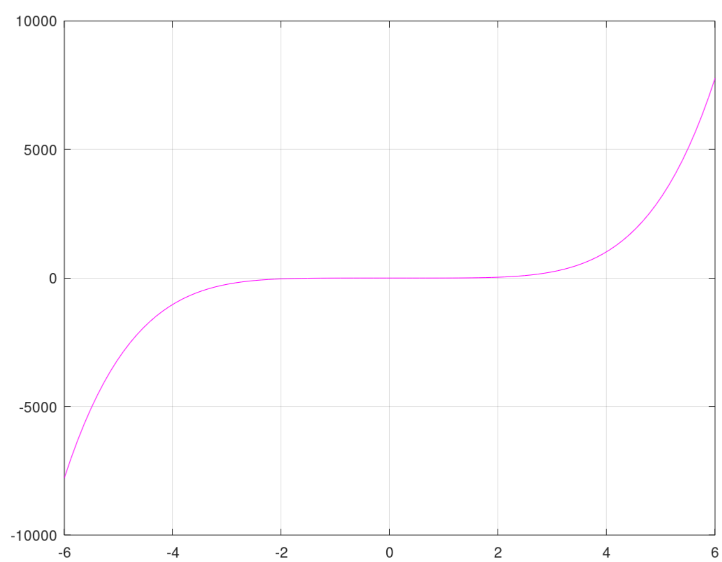
Podczas obliczania całki z funkcji $y = log(\frac{x}{16} + 1)$ wszystkie metody poradziły sobie bardzo dobrze. Metoda Simpsona wypadła nieco lepiej, ale różnica nie w liczbie podprzedziałów nie jest aż tak duża.

3.3 Przykład III

Kolejna wybrana funkcja podcałkowa $y = x^5 + 0.5$ zostanie zbadana na przedziale $[-6, 6]$.

3.3.1 Wykres funkcji

Poniżej przedstawiony jest wykres funkcji $y = x^5 + 0.5$ na przedziale $[-6, 6]$.



3.3.2 Wyniki obliczeń numerycznych

N (l. podprzedziałów)	Metoda prostokątów	Metoda Trapezów	Metoda Simpsona
1	6	6	6
2	6	6	6
6	6	6	6
12	6	6	6
24	6	6	6
48	6	6	6
100	6	6	6
1000	6	6	6
100000	6	6	6
1000000	6	6	6

3.3.3 Wnioski

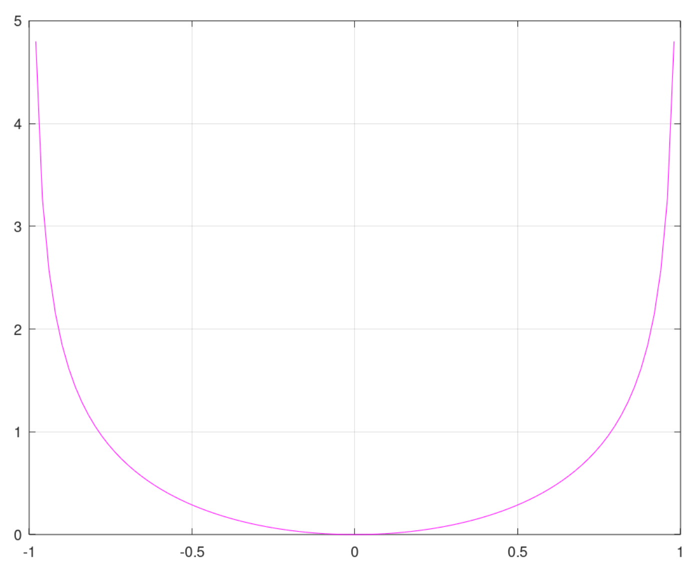
Wszystkie metody poradziły sobie tak samo dobrze, nawet przy bardzo małej liczbie podprzedziałów.

3.4 Przykład IV

Ostatnia wybrana funkcja podcałkowa $y = \frac{x^2}{\sqrt{1-x^2}}$ zostanie zbadana na przedziale $[-1, 1]$.

3.4.1 Wykres funkcji

Poniżej przedstawiony jest wykres funkcji $y = \frac{x^2}{\sqrt{1-x^2}}$ na przedziale $[-1, 1]$.



3.4.2 Wyniki obliczeń numerycznych

N (l. podprzedziałów)	Metoda prostokątów	Metoda Trapezów	Metoda Simpsona
1	0	Inf*	Inf**
2	0.57735	Inf*	Inf**
6	1.04876	Inf*	Inf**
12	1.21146	Inf*	Inf**
24	1.32025	Inf*	Inf**
48	1.3949	Inf*	Inf**
100	1.44939	Inf*	Inf**
1000	1.53253	Inf*	Inf**
100000	1.56697	Inf*	Inf**
1000000	1.56959	Inf*	Inf**

*Komunikat:

```
warning: division by zero
warning: called from
f at line 2 column 2
ktrapezow at line 17 column 6
warning: division by zero
warning: called from
f at line 2 column 2
ktrapezow at line 17 column 6
Wartosc calki: Inf
```

**Komunikat:

```
warning: division by zero
warning: called from
f at line 2 column 2
ksimpsona at line 24 column 6
warning: division by zero
warning: called from
f at line 2 column 2
ksimpsona at line 24 column 6
Wartosc calki: Inf
```

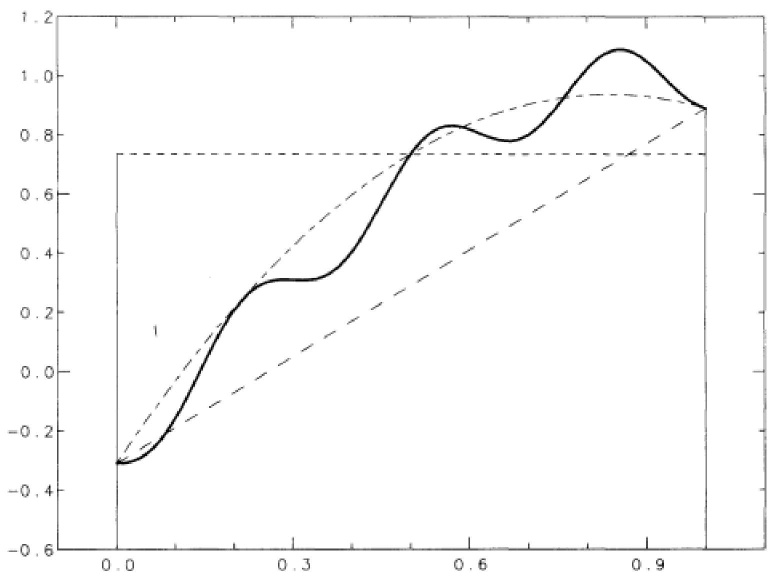
3.4.3 Wnioski

Wzory w metodach trapezów i Simpsona nie pozwalają na wyliczenie wartości całki. Poradziła sobie jedynie najprostsza z metod - kwadratura prostokątów. Wnioskujemy, że należy ostrożnie dobierać wzory funkcji podcałkowych do odpowiednich wzorów. Należy odpowiednio szacować, co może stać się po wyliczeniu wartości funkcji podcałkowej w krańcach przedziału całkowania.

4 Wnioski

Całkowanie numeryczne polega na przybliżeniu całki w oparciu o wartości funkcji w pewnych punktach. Dobór tych punktów zależy od metody. Punkty należy wybrać dopiero po ustaleniu, która z metod sprawdzi się najlepiej. Z całkowania numerycznego korzystamy między innymi w przypadkach, gdy nie potrafimy obliczyć potrzebnej całki oznaczonej w sposób analityczny albo nie znamy wzoru na funkcję podcałkową, a tylko jej wartości w pewnych punktach. Ponadto musimy wiedzieć czy dana całka istnieje. Techniki użyte do obliczeń bazują na interpolacji. Zamiast obliczać całkę analitycznie można zastąpić ją sumą całek na pewnych podprzedziałach. Przy użyciu prostych lub parabol obliczenia stają się znacznie prostsze, a błąd wraz ze zwiększaniem liczby węzłów - zmniejsza się. Poniżej przedstawione zostało porównanie trzech metod na przykładowej funkcji

Poniższy rysunek ukazuje w jaki sposób działają omówione metody. Przez odpowiednio proste lub parabole, przybliżamy dany wykres funkcji w celu ułatwienia obliczeń.



Rysunek 1: Metoda prostokątów, trapezów i Simpsona.
Źródło: D.Kahaner, C.Moler, S. Nash: *Numerical Methods and Software*

Formuły trapezów oraz Simpsona są formułami zamkniętymi, tzn. wymagają, aby wartości funkcji f (a) i f (b) były określone. Nie mogą być zatem stosowane do całek posiadających nieusuwalne osobliwosci na końcach przedziałów. Ze względu na to, że obrane metody są metodami podstawowymi, wszystkie przedziały są równej długości, co upraszcza liczenie. Najdokładniejszą metodą w większości przypadków okaże się metoda Simpsona, chociaż nie radzi sobie z funkcjami "spłaszczonymi". W przypadku złożonej kwadratury prostokątów błąd wynikający z wykorzystywania przybliżonych wartości funkcji całkowanej będzie największy, gdy funkcja ma bardzo zmienną monotoniczność, a węzły są przy tym nieodpowiednio dobrane. Metoda trapezów jest zbliżona do metody prostokątów. W większości przypadków wykazuje większą dokładność.

5 Podsumowanie

Obliczanie niektórych całek bez korzystania z metod przybliżonych byłoby zbyt trudne do wykonania. Często jedyną możliwością jest aproksymowanie numeryczne. Jednak nie tylko wtedy mają miejsca przybliżenia. Pojawienie się liczby (e, pi itd) w obliczeniach pisemnych wymusza na nas przybliżone podanie wartości. Zawsze należy zastanowić się, która metoda będzie najkorzystniejsza. Należy ostrożnie podchodzić do wyników i starać się oszacować ich poprawność.

Literatura

[1] A. Szatkowski, J. Cichosz: *Metody Numeryczne. Podstawy Teoretyczne*

[2] M. Kosiorowska, T. Stanisz: *Metody Numeryczne*

[3] J. Klamka, Z. Ogonowski, M. Jamicki, M. Stasik: *Metody Numeryczne*

[4] A. Poliński: *Metody Numeryczne i Algorytmy*

[5] R. Klempka, B. Świątek, A. Garbacz-Klempka: *Programowanie, Algorytmy Numeryczne i Modelowanie w Matlabie*

[6] W. Regel: *Obliczenia Symboliczne i Numeryczne w Programie Matlab*

[7] L. H. Hodges: *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*

[8] D. Kahaner, C. Moler, S. Nash: *Numerical Methods and Software*