

Całkowanie i różniczkowanie numeryczne

Całkowanie numeryczne

Całkowanie numeryczne polega w gruncie rzeczy na zastąpieniu funkcji podcałkowej przez jakiś wielomian interpolacyjny, a następnie scałkowaniu tego wielomianu. Jak łatwo przewidzieć, całkowanie wielomianów wysokich rzędów nie będzie zbyt wygodne, korzystnie jest zatem ograniczyć się do wielomianów niższych rzędów – do ich utworzenia wystarczy wykorzystywać tylko kilka punktów – w każdym przedziale całkujemy wówczas wielomian tego samego rzędu, ale o innych współczynnikach. Jak pamiętamy, w sąsiednich przedziałach funkcja interpolująca często odbiega od wartości funkcji interpolowanej w przeciwnych kierunkach – wynika stąd, że wartość całki może być obciążona mniejszym błędem względnym niż użyty do jej wyznaczenia wielomian interpolacyjny.

Rozpatrywać będziemy tylko sytuację gdy znamy postać funkcji podcałkowej – tzn. jesteśmy w stanie obliczyć jej wartości w dowolnym punkcie.

Punkty, które wykorzystamy do obliczenia wartości całki mogą być rozmieszczone równomiernie (w stałych od siebie odległościach) lub nierównomiernie. Metody całkowania numerycznego wykorzystujące punkty rozmieszczone równomiernie nazywane są kwadraturami Newtona-Cotesa, zaś z punktami nierównomiernymi – kwadraturami Gaussa.

Wzory interpolacyjne Newtona

W przypadku punktów równoodległych szczególnie wygodne są wzory interpolacyjne Newtona umożliwiające znalezienie wielomianu interpolacyjnego na podstawie różnic wartości funkcji w punktach sąsiednich. Poniższy wzór wykorzystuje różnice przednie.

$$y(x) = y_0 + \alpha \Delta y_0 + \frac{\alpha(\alpha+1)}{2!} \Delta^2 y_0 + \dots + \frac{\alpha(\alpha+1)\dots(\alpha+n-1)}{n!} \Delta^n y_0 + R_n$$

$$R_n = \frac{\alpha(\alpha+1)\dots(\alpha+n)}{(n+1)!} f^{(n+1)}(\xi)$$

gdzie $\alpha = (x - x_0)/h$, a R_n jest resztą. Symbol Δ^i oznacza różnicę rzędu i :

$$\begin{aligned} \Delta y_i &= y_{i+1} - y_i \\ \Delta^2 y_i &= \Delta y_{i+1} - \Delta y_i \\ \Delta^3 y_i &= \Delta^2 y_{i+1} - \Delta^2 y_i \\ &\vdots \\ \Delta^n y_i &= \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i \end{aligned}$$

Kwadratury z punktami równoodległymi (kwadratury Newtona-Cotesa)

W poniższych wzorach przez y_i będziemy oznaczać wartości funkcji $f(x)$ (funkcji całkowanej) w punktach x_i , które są rozmieszczone w stałych odległościach wynoszących h .

Wzór trapezów

$$\int_a^b f(x) dx \approx h \left(\frac{y_0 + y_n}{2} + y_1 + y_2 + \dots + y_{n-1} \right)$$

Reszta (błąd obcięcia) jest postaci:

$$R_t = -\frac{nh^3}{12} f''(\xi) = -\frac{(b-a)^3}{12n^2} f''(\xi) = -\frac{(b-a)h^2}{12} f''(\xi)$$

gdzie ξ znajduje się w przedziale całkowania (a, b) . Jak widać podwojenie gęstości rozmieszczenia punktów o znanych wartościach funkcji całkowanej spowoduje w przybliżeniu czterokrotne zmniejszenie błędu (dla innych punktów znanych inna będzie też wartość $f''(\xi)$).

Wzór trapezów daje dokładne wartości całki tylko dla funkcji liniowych (już druga pochodna tych funkcji jest tożsamościowo równa zero).

W zasadzie należałoby rozpocząć od „wzoru prostokątów” – w którym wartość całki otrzymujemy sumując wartości funkcji w węzłach i mnożąc je przez długość „kroku”:

$$\int_a^b f(x) dx \approx h \left(y_0 + y_1 + y_2 + \dots + y_{n-1} \right) \quad \text{lub} \quad h \left(y_1 + y_2 + y_3 + \dots + y_n \right)$$

jednak wzór ten, jako wyrażenie mniej dokładny od wzoru trapezów a zarazem nie dający żadnej oszczędności obliczeniowej, praktycznie nie jest stosowany.

Wzór Simpsona (wzór parabol)

$$\int_a^b f(x) dx \approx \frac{h}{3} \left(y_0 + y_{2m} + 2 \left(y_2 + y_4 + \dots + y_{2m-2} \right) + 4 \left(y_1 + y_3 + \dots + y_{2m-1} \right) \right)$$

Reszta ma postać:

$$R_S \approx -\frac{mh^5}{90} f^{(4)}(\xi) \approx -\frac{(b-a)h^4}{180} f^{(4)}(\xi)$$

gdy

$$h = \frac{b-a}{2m}$$

Warto zauważyć, że w przypadku wzoru Simpsona liczba przedziałów n musi być parzysta, (tj. liczba punktów musi dać się przedstawić w postaci $2m+1$).

Wzór Newtona (wzór „trzech ósmych”)

$$\int_a^b f(x) dx \approx \frac{3h}{8} \left(y_0 + y_{3m} + 2 \left(y_3 + y_6 + \dots + y_{3m-3} \right) + 3 \left(y_1 + y_2 + y_4 + y_5 + \dots + y_{3m-2} + y_{3m-1} \right) \right)$$

Reszta ma postać:

$$R_N \approx -\frac{3mh^5}{80} f^{(4)}(\xi) \approx -\frac{(b-a)h^4}{80} f^{(4)}(\xi)$$

gdy

$$h = \frac{b-a}{3m}$$

Warto pamiętać, że w tej metodzie liczba przedziałów musi być podzielna przez trzy (tj. liczba punktów musi dać się przedstawić w postaci $3m+1$).

Znane są również wzory wyższych rzędów. Ogólnie wzory tego typu przedstawiane są w postaci:

$$\int_a^b f(x) dx \approx (b-a) \sum_{n=0}^N \frac{I_{n,N}}{m_n} f(x_n) \approx Kh^{p+1} f^{(p)}(\xi)$$

gdzie współczynniki kwadratury $I_{n,N}$ i m_n oraz współczynniki K i p potrzebne do oszacowania błędu zestawiono w tabelce:

N	$I_{n,N}$	m_n	p	K	nazwa
1	1, 1	2	2	1 / 12	trapezy
2	1, 4, 1	6	4	1 / 90	Simpson
3	1, 3, 3, 1	8	4	3 / 80	Newton

4	7, 32, 12, 32, 7	90	6	8 / 945	Milne
5	19, 75, 50, 50, 75, 19	288	6	275 / 12096	Bode
6	41, 216, 27, 272, 27, 216, 41	840	8	9 / 1400	Weddle

Udokładnianie wartości całek (metoda Richardsona)

Założmy, że J_n i E_n są odpowiednio wartościami całki i błędami uzyskanymi dla $n+1$ punktów. Założmy też, że obliczenia przeprowadzono metodą trapezów dla dwóch ilości punktów: n_1 i n_2 ; prawdziwą wartość całki oznaczmy przez J .

$$J = J_{n_1} + E_{n_1} = J_{n_2} + E_{n_2}$$

$$\frac{E_{n_2}}{E_{n_1}} = \frac{\frac{(b-a)^3}{12n_2^2} f''(\xi_2)}{\frac{(b-a)^3}{12n_1^2} f''(\xi_1)}$$

Zakładając równość $f''(\xi_1)$ i $f''(\xi_2)$ otrzymamy:

$$E_{n_2} = \left(\frac{n_1}{n_2}\right)^2 E_{n_1}$$

co po podstawieniu da wartość dokładną (a przynajmniej dokładniejszą niż J_1 i J_2):

$$J = J_{n_1} + \frac{J_{n_2} - J_{n_1}}{1 - \left(\frac{n_1}{n_2}\right)^2}$$

Dla $n_2 = 2n_1$ otrzymamy:

$$J = \frac{4}{3}J_{n_2} - \frac{1}{3}J_{n_1}$$

Analogicznie można wyprowadzić wzór udokładniony dla metod Simpsona i Newtona (wzory są identyczne bo w obu metodach błąd jest proporcjonalny do czwartej potęgi długości przedziału).

$$J = \frac{16}{15}J_{n_2} - \frac{1}{15}J_{n_1}$$

Dla przykładu scałkujemy funkcję e^{-x^2} w przedziale (0,1). Rozwiązaniem dokładnym jest:

$$\int_0^1 \frac{(1)^k}{(2k+1)k!} (b^{2k+1} - a^{2k+1})$$

Do wyznaczenia wyniku dokładnego wykorzystano 170 wyrazów. Całkowanie numeryczne przeprowadzimy dla 72 i 36 przedziałów (taki wybór umożliwi przeprowadzanie obliczeń wszystkimi omówionymi metodami). Wyniki zebrano w poniższej tabeli.

Wartość dokładna: 0.7468241328124270		
	wartość	błąd względny
<i>Trapezy:</i>		
72:	0.746812305337	-1.58e-05
36:	0.746776821997	-6.33e-05
udokładnienie:	0.746824133117	4.07e-10

Simpson:			
72:	0.746824133117	4.07e-10	
36:	0.746824137679	6.52e-09	
udokładnienie:	0.746824132812	6.02e-14	
Newton:			
72:	0.746824133497	9.16e-10	
36:	0.746824143760	1.47e-08	
udokładnienie:	0.746824132812	2.72e-13	

Kwadratury z punktami nierównoodległymi

Ogólnie rzecz biorąc ideą kwadratur tego rodzaju jest zastąpienie całki oznaczonej przez sumę wartości funkcji podcałkowej w wybranych punktach (należących do przedziału całkowania) pomnożonych przez pewne współczynniki (zależne od metody).

Gauss-Legendre

Wzory dla kwadratury Gaussa-Legendre'a mają następującą postać:

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

gdzie n jest liczbą przedziałów, x_i są wartościami argumentu, dla których wyznaczone zostaną wartości funkcji podcałkowej zaś w_i są współczynnikami wagowymi przypisanymi do poszczególnych argumentów x_i . Ich wartości zebrane są w poniższej tabeli:

Argumenty x_i	Współczynniki wagowe w_i
Wzór dla dwóch punktów	
$\pm 0.57735\ 02692\ 41483$	1.00000 00000 00000
Wzór dla trzech punktów	
0.00000 00000 00000	0.88888 88888 88889
$\pm 0.77459\ 66692\ 41483$	0.55555 55555 55556
Wzór dla czterech punktów	
$\pm 0.33998\ 10435\ 84856$	0.65214 51548 62546
$\pm 0.86113\ 63115\ 94053$	0.34785 48451 37454
Wzór dla pięciu punktów	
0.00000 00000 00000	0.56888 88888 88889
$\pm 0.53846\ 93101\ 05683$	0.47862 86704 99366
$\pm 0.90617\ 98459\ 38664$	0.23692 68850 56189
Wzór dla sześciu punktów	

$\pm 0.23861\ 91860\ 83197$	$0.46791\ 39345\ 72691$
$\pm 0.66120\ 93864\ 66265$	$0.36076\ 15730\ 48139$
$\pm 0.93246\ 95142\ 03152$	$0.17132\ 44923\ 79170$
Wzór dla dziesięciu punktów	
$\pm 0.14887\ 43389\ 81631$	$0.29552\ 42247\ 14753$
$\pm 0.43339\ 53941\ 29247$	$0.26926\ 67193\ 09996$
$\pm 0.67940\ 95682\ 99024$	$0.21908\ 63625\ 15982$
$\pm 0.86506\ 33666\ 88985$	$0.14945\ 13491\ 50581$
$\pm 0.97390\ 65285\ 17172$	$0.06667\ 13443\ 08688$
Wzór dla piętnastu punktów	
$0.00000\ 00000\ 00000$	$0.20257\ 82419\ 25561$
$\pm 0.20119\ 40939\ 97435$	$0.19843\ 14853\ 27111$
$\pm 0.39415\ 13470\ 77563$	$0.18616\ 10001\ 15562$
$\pm 0.57097\ 21726\ 08539$	$0.16626\ 92058\ 16994$
$\pm 0.72441\ 77313\ 60170$	$0.13957\ 06779\ 26154$
$\pm 0.84820\ 65834\ 10427$	$0.10715\ 92204\ 67172$
$\pm 0.93727\ 33924\ 00706$	$0.07036\ 60474\ 88108$
$\pm 0.98799\ 25180\ 20485$	$0.03075\ 32419\ 96117$

Zazwyczaj podaje się je właśnie w postaci dla przedziału całkowania $(-1,1)$, ale dla dowolnego przedziału (a,b) można wprowadzić podstawienie:

$$z = \frac{2x - (a+b)}{b-a}$$

które sprowadzi go do $(-1,1)$.

Można też przekształcić wzór do postaci, która czasami może być wygodniejsza:

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{z(b-a)+b+a}{2}\right) dz = \frac{b-a}{2} \sum_{i=0}^n w_i f\left(\frac{z_i(b-a)+b+a}{2}\right)$$

Wartości z_i są zgodne z powyższą tabelą.

Dla przykładu wykonamy analogiczne całkowanie jak w poprzednim przykładzie.

Kwadratury Gaussa-Legendre'a	
funkcja: e^{x^2}	
wartość dokładna: 0.7468241	
wartość	błąd
2 pkt	
0.7465947	-3.07e-04
3 pkt	
0.7468146	-1.30e-05
4 pkt	
0.7468245	4.00e-07

5 pkt		
0.7468241		-8.05e-09
6 pkt		
0.7468241		1.02e-10

Gauss-Czebyszew

Analogicznie jak w przypadku kwadratur Gaussa-Lagrange'a można skonstruować inne kwadratury wykorzystujące aproksymację funkcji wielomianami ortogonalnymi. Dla całek postaci:

$$\int_{-1}^1 \frac{1}{\sqrt{1-z^2}} F(z) dz \approx \sum_{i=0}^n w_i F(z_i)$$

(należy zwrócić uwagę na pierwiastek w mianowniku – nie do każdej funkcji musi się to nadawać) mamy kwadratury Gaussa-Czebyszewa – ich węzły i współczynniki wynoszą:

$$x_i = \cos \frac{(2i+1)\pi}{2N+2} \quad ; \quad w_i = \frac{\pi}{N+1}$$

Gauss-Laguerre i Gauss-Hermite

Metoda Gaussa-Laguerre'a służy do obliczania całek postaci:

$$\int_0^4 f(x) e^{-x} dx$$

zaś Gaussa-Hermite'a:

$$\int_{-\infty}^{\infty} f(x) e^{-x^2} dx$$

Węzły i współczynniki zbieram dla obu metod w jednej tabelce, wynoszą one:

N	Gauss-Laguerre			Gauss-Hermite		
	i	x_i	w_i	i	x_i	w_i
1	0	0,58578 64376	0,85355 39906	0;1	$\pm 0,707106 781187$	0,886226 925453
	1	3,41421 35624	0,14644 66094			
2	0	0,41577 45568	0,71109 30099	0;2 1	$\pm 1,224744 87139$ 0	0,295408 975151 1,181635 9006
	1	2,29428 03603	0,27851 77336			
	2	6,28994 50829	0,01038 92565			
3	0	0,32254 76896	0,60315 41043	0;3 1;2	$\pm 1,650680 12389$ $\pm 0,524647 623275$	0,081312 8354472 0,804914 090006
	1	1,74576 11012	0,35741 86924			
	2	4,53662 02969	0,03888 79085			
	3	9,39507 09123	0,00053 92947			

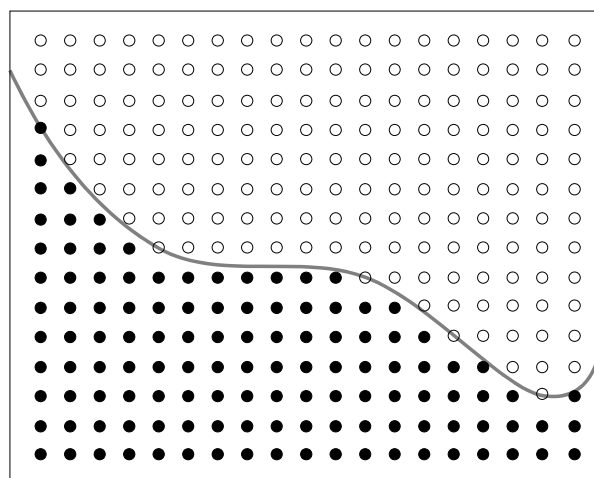
4	0	0,26356 03197	0,52175 56106	0;4	$\pm 2,020182\ 87046$	0,019953 242059
	1	1,41340 30591	0,39866 68110	1;3	$\pm 0,958572\ 464614$	0,393619 323152
	2	3,59642 57710	0,07594 24497	2	0	0,945308 720483
	3	7,08581 00059	0,00361 17587			
	4	12,64080 08443	0,00002 33700			
5	0	0,22284 66042	0,45896 46740			
	1	1,18893 21017	0,41700 08308			
	2	2,99273 63261	0,11337 33821			
	3	5,77514 35691	0,01039 91975			
	4	9,83746 74184	0,00026 10172			
9	5	15,98287 39806	0,00000 08985			
	0	0,13779 34705	0,30844 11158			
	1	0,72945 45495	0,40111 99292			
	2	1,80834 29017	0,21806 82876			
	3	3,40143 36979	0,06208 74561			
	4	5,55249 61401	0,00950 15170			
	5	8,33015 27468	0,00075 30084			
	6	11,84378 58379	0,00002 82592			
	7	16,27925 78313	0,00000 04249			
	8	21,99658 58119	0,00000 00018			
	9	29,92069 70123	1e-12			

Całkowanie metodą Monte-Carlo

Wszystkie powyżej opisane metody całkowania numerycznego dotyczyły całkowania funkcji jednej zmiennej. Całkowanie funkcji większej liczby zmiennych można zazwyczaj przeprowadzić „rozkładając” całkę wielu zmiennych na wiele całek jednej zmiennej obliczanych kolejno, ale powoduje to ogromny wzrost czasu obliczeń – proporcjonalny do n^k gdzie n jest liczbą punktów, a k liczbą zmiennych. Ponieważ jednak w celu uzyskania wystarczającej dokładności n musi być dość duże – okazuje się, że metoda taka jest raczej nie do przyjęcia. Alternatywną metodą całkowania jest zastosowanie metody Monte-Carlo. Wyjaśnię ją tu na przykładzie przypadku funkcji jednej zmiennej, a następnie uogólnię dla większej liczby zmiennych.

(Uwaga: Na poniższych rysunkach narysowane punkty **nie są** losowe – po prostu tak było mi łatwiej rysować ;-).

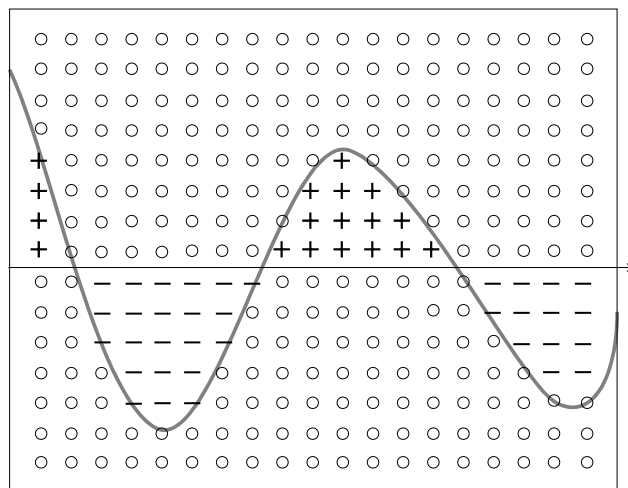
Najpierw zajmiemy się prostszym przypadkiem – gdy całkowana funkcja przyjmuje tylko wartości dodatnie (ilustruje to rysunek obok). Jeśli tylko znamy postać funkcji podcałkowej (a to jest przy całkowaniu numerycznym oczywiste) możemy stwierdzić w jakim zakresie mieszczą się wartości tej funkcji (oczywiście jej argument zmienia się tylko w zakresie x odpowiadającym granicom całkowania a i b) – będzie to zakres od f_{\min} do f_{\max}). Następnie na płaszczyźnie xy wybieramy prostokąt ograniczony na osi x granicami całkowania (a i b), a na osi y dwiema wartościami – większą lub równą f_{\max} i mniejszą lub równą f_{\min} . Następnie przeprowadzamy dużą liczbę losowań par liczb z zakresów (a,b) i (f_{\min},f_{\max}) . Zakładając, że pierwsza liczba (ta z zakresu (a,b)) jest wartością x możemy obliczyć odpowiadającą jej wartość funkcji podcałkowej i sprawdzić czy jest ona większa czy mniejsza od



drugiej z wylosowanych liczb (tej z zakresu (f_{\min}, f_{\max})). Umożliwia to zaliczenie wylosowanego punktu do obszaru znajdującego się pod wykresem funkcji albo nad nim. Zliczamy liczbę wszystkich wylosowanych punktów (N) oraz liczbę punktów trafiających pod wykres funkcji (N_+). Jeśli tylko wylosujemy dostatecznie wiele punktów możemy liczyć na to, że wartość całki będzie się mieścić do całkowitego pola powierzchni wybranego prostokąta tak jak liczba trafień do całkowitej liczby losowań:

$$\frac{\int_a^b f(x) dx}{(b-a) \cdot (f_{\max} - f_{\min})} \approx \frac{N_+}{N}$$

Jednak w ogólnym przypadku funkcja podcałkowa może przyjmować wartości zarówno dodatnie jak i ujemne (co zilustrowano obok). W takim przypadku warunek jaki będziemy sprawdzać będzie nieco bardziej skomplikowany. W przypadku wylosowania wartości dodatniej „licznik trafień” będziemy zwiększać jeśli wartość ta jest mniejsza od wartości funkcji podcałkowej, natomiast w przypadku wylosowania wartości ujemnej równocześnie większej od wartości funkcji podcałkowej licznik ten będziemy zmniejszać (co będzie odpowiadać „ujemnej” powierzchni po osi x) – równie dobrze możemy prowadzić dwa liczniki – jeden dla wartości dających dodatni wkład do całki i drugi dla wartości dających wkład ujemny.



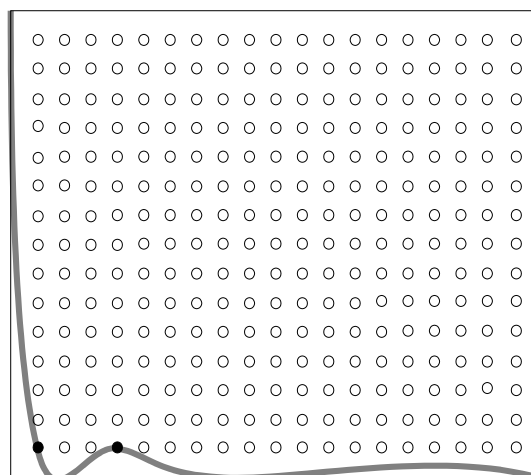
$$\frac{\int_a^b f(x) dx}{(b-a) \cdot (f_{\max} - f_{\min})} \approx \frac{N_+ - N_-}{N}$$

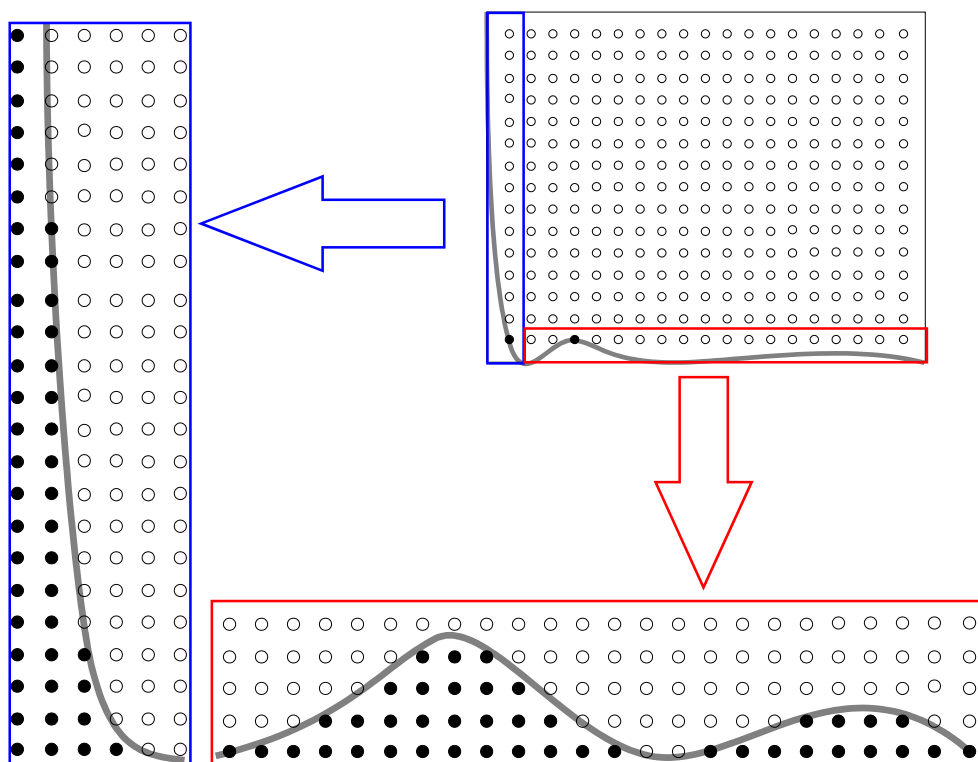
Jeśli wylosowane punkty równomiernie pokrywają cały obszar to każdemu z nich można przypisać pewną powierzchnię:

$$\delta S = \frac{(f_{\max} - f_{\min})(b - a)}{N}$$

gdzie N jest liczbą wylosowanych punktów. S ma oczywiście wymiar szukanej całki.

Widzimy oczywiście, że otrzymana wartość całki będzie obarczona błędem statystycznym zależnym od liczby losowań, co więcej, kolejne obliczenia tej samej całki wcale nie muszą dawać takiego samego wyniku. Łatwo też zauważyć, że dokładność oszacowania maksimum i minimum wartości funkcji, oraz gładkość samej funkcji podcałkowej będą mieć wpływ na dokładność obliczonej wartości całki. Rysunek obok ilustruje jedną z możliwości popełnienia ogromnego błędu podczas całkowania tą metodą – mamy tu do czynienia z sytuacją gdy większość wylosowanych punktów nie daje żadnego wkładu do wartości całki. Gdybyśmy spotkali się z koniecznością scałkowania takiej funkcji należałoby raczej rozłożyć ją na dwie całki – takie, aby podczas całkowania każdej części zarówno punkty dające wkład do wartości całki jak i te go nie dające miały znaczący udział w całkowitej liczbie punktów (przykładowy podział przedstawiono na rysunku poniżej).





Zaletą metody Monte Carlo jest łatwość jej zaprogramowania oraz (co chyba najważniejsze) fakt, że można ją bardzo łatwo zaadaptować do przypadku wielowymiarowego – zamiast par liczb trzeba będzie losować zestawy $(n+1)$ liczb (liczba zmiennych plus jeden). Zakresy wartości dobieramy odpowiednio do granic całkowania (dla pierwszych n liczb) oraz do oszacowanej minimalnej i maksymalnej wartości funkcji podcałkowej (dla liczby $n+1$ -ej). Dalsze postępowanie jest analogiczne jak dla funkcji jednej zmiennej – pierwsze n liczb interpretujemy jako wartości zmiennych, a $n+1$ -szą porównujemy z obliczoną wartością funkcji. Stosunek liczby trafień do całkowitej liczby losowań będzie taki sam jak stosunek wartości całki do objętości (w n wymiarach) wybranego obszaru.

W stosunku do całkowania funkcji jednej zmiennej istnieje jednak jedna dodatkowa komplikacja – podczas całkowania funkcji jednej zmiennej pierwsza z wylosowanej pary liczb losowych, interpretowana jako wartość argumentu funkcji zawsze trafiała w zakres całkowania. Tymczasem podczas całkowania funkcji np. dwóch zmiennych takiej gwarancji już nie ma. Załóżmy, że chcemy scałkować funkcję $f(x,y)$ po obszarze będącym kołem $r^2 \leq x^2 + y^2$. Mamy wówczas do wyboru dwie możliwości: albo losować będziemy od razu dwie liczby, każdą z zakresu $\langle -r, +r \rangle$, godząc się z faktem, że nie każda z tak wylosowanych par będzie należeć do koła będącego obszarem całkowania – wówczas algorytm nasz będzie zawierać „puste” losowania – pary nie należące do obszaru całkowania będziemy odrzucać (nie będą one powodować zmiany żadnego z liczników). Druga możliwość polega na wylosowaniu tylko jednej z liczb, zinterpretowaniu jej jako np. x i wyliczeniu na jej podstawie zakresu losowania drugiej liczby (interpretowanej oczywiście już jako y), która teraz już z całą pewnością będzie do obszaru całkowania należeć. Jak widać w metodzie drugiej nie występują „puste” losowania, jednak sam proces losowania musi trwać dłużej. Jak widać trudno powiedzieć, która z metod będzie szybsza. Pierwsza z powyżej przedstawionych metod jest prostsza i działa zawsze jeśli tylko potrafimy zapisać test sprawdzający czy punkt o danych współrzędnych należy do obszaru całkowania czy nie. Oczywiście w przypadku całkowania funkcji o większej liczbie zmiennych nasze działanie będzie analogiczne.

Można rozważać jeszcze jedną metodę przyspieszenia obliczeń (niestety raczej nieznaczniego) – w przypadkach gdy prawdopodobieństwo nietrafienia w obszar całkowania jest znaczne można losowanie ostatniej liczby losowej (tej interpretowanej jako wartość funkcji) odłożyć aż do podjęcia decyzji o trafieniu w obszar całkowania.

Alternatywną metodą całkowania Monte Carlo może być przypisanie każdej ze zmiennych określonej szerokości przedziału – wynikającej z zakresu całkowania i liczby losowanych punktów, a następnie obliczanie wartości funkcji w

punkcie o wylosowanych współrzędnych i ich sumowanie. Po pomnożeniu obliczonej sumy wartości przez wartość „elementarnej objętości” otrzymujemy wartość całki. Metoda ta jest niewątpliwie prostsza w implementacji. Poniżej przedstawiam przykład całkowania funkcji jednej zmiennej (x_L oznacza wylosowaną, za każdym razem inną, wartość argumentu całkowanej funkcji).

$$\int_a^b f(x) dx \approx \sum_{j=1}^N f(x_{Lj}) \frac{(b-a)}{N}$$

W przypadku konieczności scałkowania funkcji n zmiennych (tu z kolei x_L oznacza wylosowany zespół argumentów całkowanej funkcji):

$$\int_{V_n} f(x_1 \dots x_n) dV_n \approx \sum_{a_1}^{b_1} \dots \sum_{a_n}^{b_n} f(x_1 \dots x_n) \prod_{i=1}^n \Delta x_i \approx \sum_{j=1}^N f(x_{Lj}) \prod_{i=1}^n \frac{(b_i - a_i)}{N}$$

Generatory liczb losowych

Osobnym problemem, z którym ściśle wiąże się dokładność obliczonych metodą Monte Carlo całek jest jakość używanego generatora liczb losowych. Jak wcześniej powiedziano dokładność obliczonej wartości całki wzrasta wraz ze wzrostem ilości wygenerowanych liczb losowych. Jednak stwierdzenie to jest prawdziwe tylko przy założeniu, że generowane liczby losowe nie powtarzają się. Tymczasem istnieje możliwość, że komputerowy generator liczb losowych w rzeczywistości generuje tylko ciąg liczb (tzw. ciąg liczb pseudolosowych) o określonej długości po czym zaczyna generować od początku te same wartości. W takim przypadku, jeżeli okres generatora wynosi np. 10000, wyniki otrzymane dla 10000, 20000 i 30000 będą identyczne, a dodatkowy czas zużyty na obliczenia będzie stracony.

Jakość generatora liczb losowych ocenia się na podstawie następujących kryteriów:

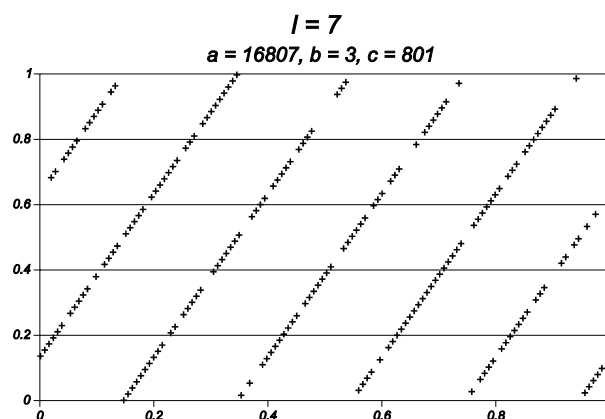
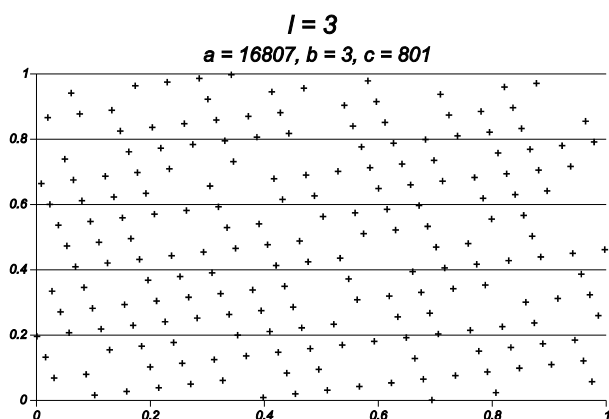
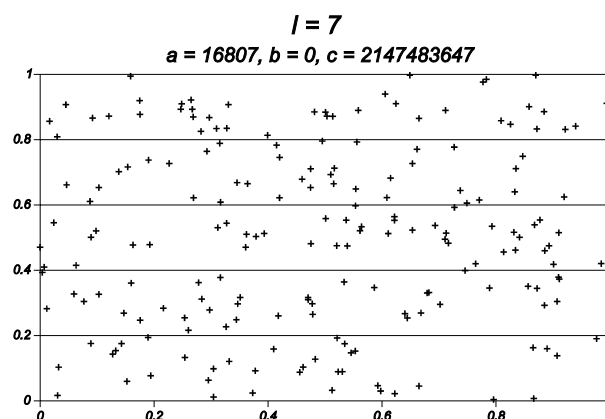
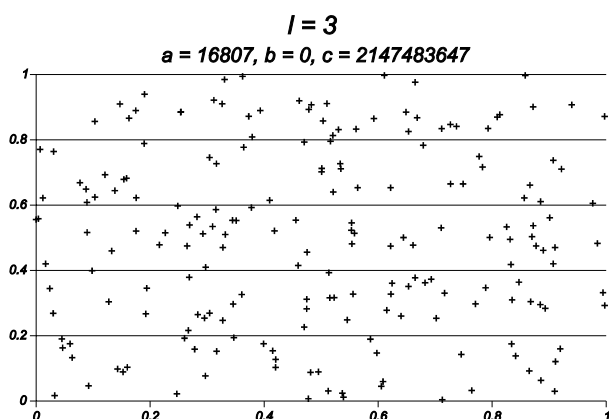
1. *Długość okresu.* Powinna być zbliżona do zakresu liczb całkowitych danego komputera.
2. *Losowość liczb.* Pomiędzy liczbami nie powinny występować korelacje. Jednym ze sposobów sprawdzenia tego jest wykreślenie punktów (x_i, x_{i+1}) . Jeśli mamy do czynienia z dobrym generatorem dla dowolnych wartości l rozkład punktów na płaszczyźnie nie będzie wykazywać niejednorodności (w postaci pasm, linii czy regularnych wzorów).
3. *Szybkość działania.* To kryterium jest oczywiste jako, że w metodzie Monte Carlo niemal z definicji konieczne jest przeprowadzanie bardzo wielu losowań.

Najprostszym jednorodnym generatorem liczb losowych jest następujący ciąg liczb:

$$x_{i+1} = (ax_i + b) \bmod c$$

gdzie a , b i c są odpowiednio wybranymi liczbami całkowitymi (od ich wyboru zależy będzie jakość generatora). Jedną z możliwości jest: $a = 7^7 = 16807$, $b = 0$, $c = 2^{31} - 1 = 2.147.483.647$ (ta ostatnia wartość jest właściwą dla komputerów o słowie 32-bitowym – zapewnia okres zbliżony do zakresu liczb całkowitych, który w takim przypadku wynosi właśnie $2^{31} - 1$). Warto pamiętać, że przedstawiony tu sposób generacji liczb losowych jest najprostszy, a niekoniecznie najlepszy do wszelkich zastosowań.

Dla ilustracji drugiego z wymienionych powyżej kryteriów oceny generatorów poniżej przedstawiono przykładowe wykresy otrzymane dla $l = 3$ i $l = 7$ dla dwóch generatorów: pierwszy z nich wykorzystywał parametry a , b i c podane powyżej jako dobre, w drugim zaś przyjęto wartości $a = 16807$, $b = 3$ i $c = 801$. Oba wygenerowały po 200 liczb.



Różniczkowanie numeryczne

Do znalezienia wyrażeń przybliżających pochodne możemy wykorzystać poznane wcześniej wzory interpolacyjne. Weźmy dla przykładu pierwszy wzór interpolacyjny Newtona:

$$y(x) \approx y_0 + q \Delta y_0 + \frac{q(q+1)}{2!} \Delta^2 y_0 + \frac{q(q+1)(q+2)}{3!} \Delta^3 y_0 + \dots + \frac{q(q+1)\dots(q+n-1)}{n!} \Delta^n y_0$$

wzór ten możemy zróżniczkować ze względu na x (pamiętając, że tylko q jest w nim zależne od x):

$$y'(x) \approx \frac{1}{h} \Delta y_0 + \frac{1}{h} \frac{(2q+1)}{2!} \Delta^2 y_0 + \frac{1}{h} \frac{(3q^2+6q+2)}{3!} \Delta^3 y_0 + \frac{1}{h} \frac{(4q^3+18q^2+22q+6)}{4!} \Delta^4 y_0 + \dots$$

Otrzymane w ten sposób wzory szczególnie upraszczają się jeśli chcemy poznać wartość pochodnej w którymś z punktów węzłowych (możemy wówczas przyjąć ten właśnie węzeł za x_0 , we wzorze podstawimy więc $q=0$):

$$y'(x) \approx \frac{1}{h} \left(\Delta y_0 + \frac{1}{2} \Delta^2 y_0 + \frac{1}{3} \Delta^3 y_0 + \frac{1}{4} \Delta^4 y_0 + \dots \right)$$

Różniczkując wzór na pierwszą pochodną otrzymamy wzór na drugą pochodną:

$$y''(x) \approx \frac{1}{h^2} \Delta^2 y_0 + \frac{(q+1)}{h^2} \Delta^3 y_0 + \frac{(6q^2+18q+11)}{12h^2} \Delta^4 y_0 + \dots$$

Ten wzór, analogicznie jak poprzedni uprości się dla przypadku obliczeń w punkcie węzłowym:

$$y^{(n)}(x) = \frac{1}{h^2} \left(\Delta^2 y_0 + \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 + \dots \right)$$

Oczywiście tak postąpić możemy z dowolnym wielomianowym wzorem interpolacyjnym. Możliwe jest oczywiście wyprowadzenie w ten sposób wzorów na pochodne dowolnych rzędów, jednak dokładność uzyskiwanych wyników pogarsza się raptownie ze wzrostem rzędu pochodnej – w praktyce nie należy obliczać numerycznie pochodnych rzędów wyższych niż drugi.

Możemy też wyprowadzić „własne” wzory na pochodne – na podstawie definicji pochodnej możemy zapisać:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad \vee \quad f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

jest to tzw. różnica przednia (ponieważ do obliczenia pochodnej wykorzystuje oprócz wartości funkcji w punkcie, który badamy, również wartość następną).

Warto zastanowić się jaką uzyskujemy w ten sposób dokładność. W tym celu zapiszemy najpierw wzór na rozwinięcie funkcji f w szereg Taylora:

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2!}f''(x_0) + \dots + \frac{(x - x_0)^n}{n!}f^{(n)}(x_0) + \dots$$

Dla punktu znajdującego się o Δx w prawo od punktu o współrzędnej x możemy więc zapisać:

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!}f''(x) + \dots + \frac{\Delta x^n}{n!}f^{(n)}(x) + \dots$$

$$\vee \quad f(x + \Delta x) = f(x) + \Delta x f'(x) + O(\Delta x^2)$$

gdzie $O(\Delta x^2)$ oznacza wielkość małą rzędu Δx^2 . Po przekształceniu otrzymamy:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + O(\Delta x)$$

czyli zapisaną już wcześniej różnicę przednią. Jak widać popełniamy w ten sposób błąd rzędu Δx .

Analogicznie moglibyśmy zapisać:

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!}f''(x) + \dots + \frac{\Delta x^n}{n!}f^{(n)}(x) + \dots$$

$$\vee \quad f(x + \Delta x) = f(x) + \Delta x f'(x) + O(\Delta x^2)$$

co po przekształceniu daje:

$$f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x} + O(\Delta x)$$

jest to tak zwana różnica wsteczna. Pod względem popełnianego błędu wzór ten nie różni się od różnicy przedniej.

Możemy jednak do obliczeń wziąć dwa punkty:

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!}f''(x) + O(\Delta x^3)$$

$$f(x - \Delta x) = f(x) - \Delta x f'(x) + \frac{\Delta x^2}{2!}f''(x) + O(\Delta x^3)$$

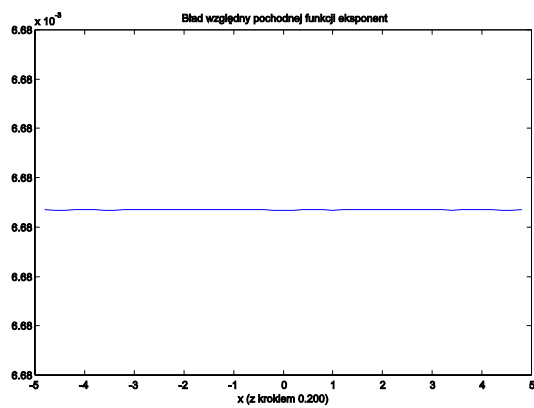
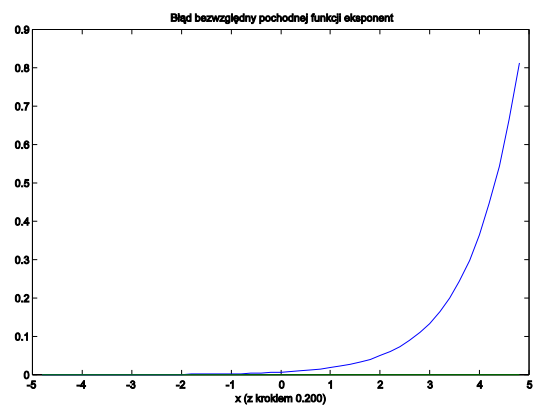
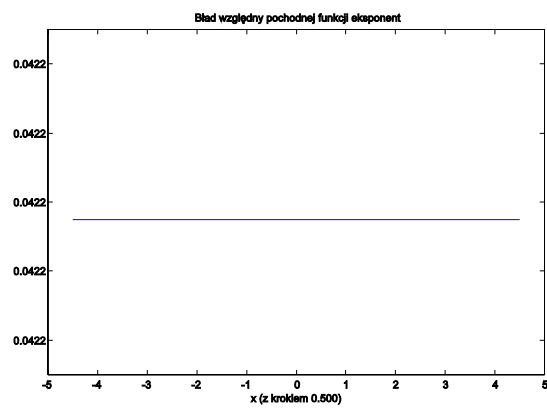
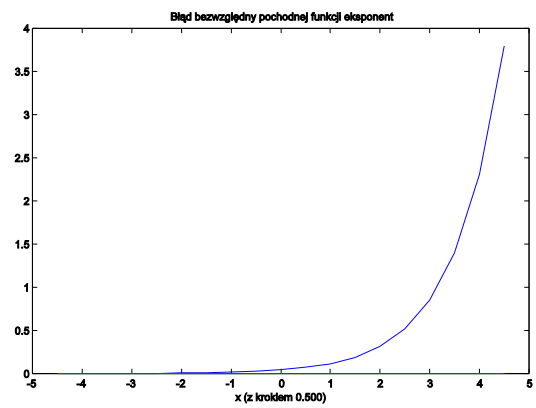
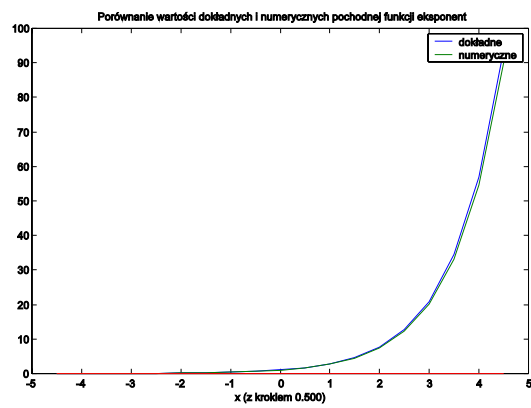
$$\vee \quad f(x + \Delta x) - f(x - \Delta x) = 2\Delta x f'(x) + O(\Delta x^3)$$

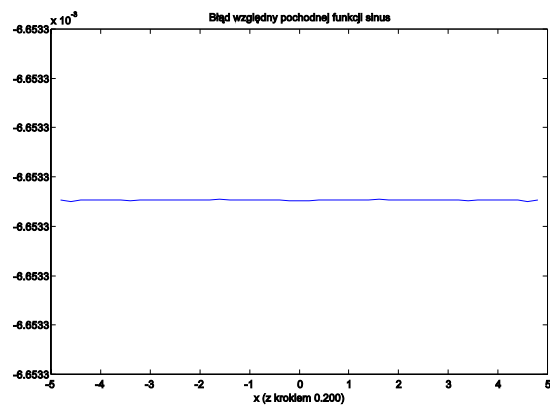
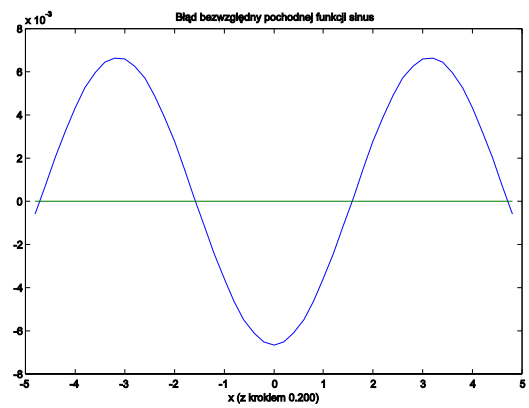
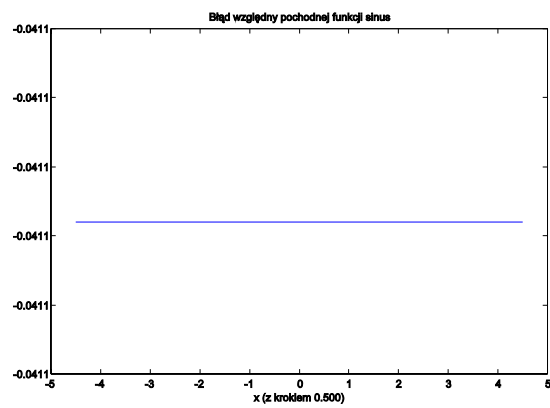
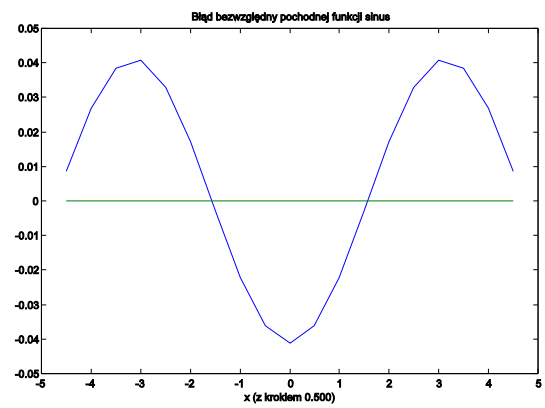
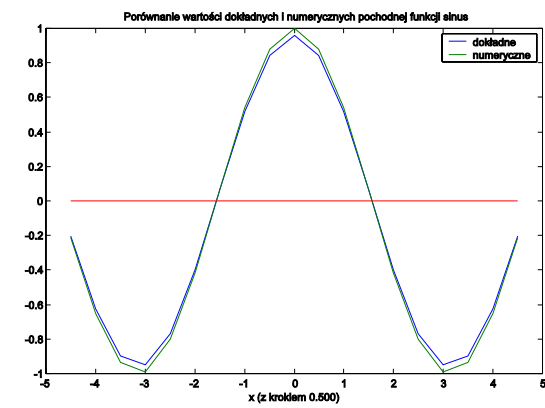
czyli:

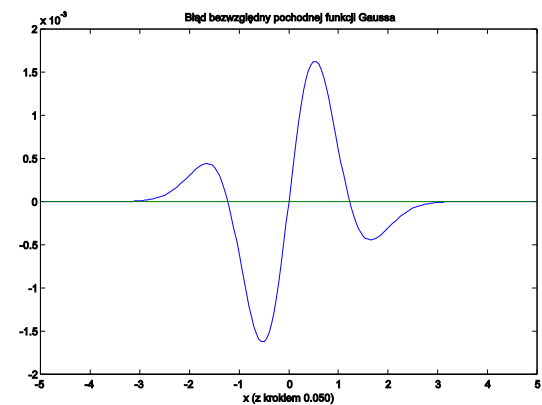
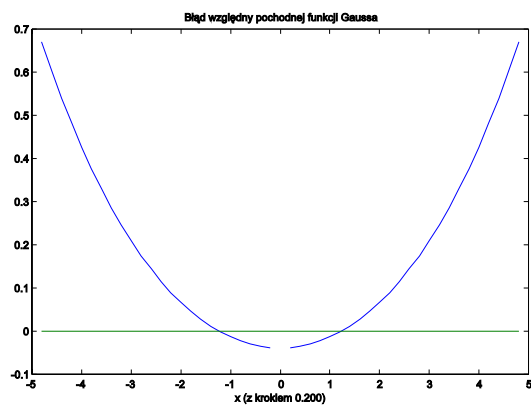
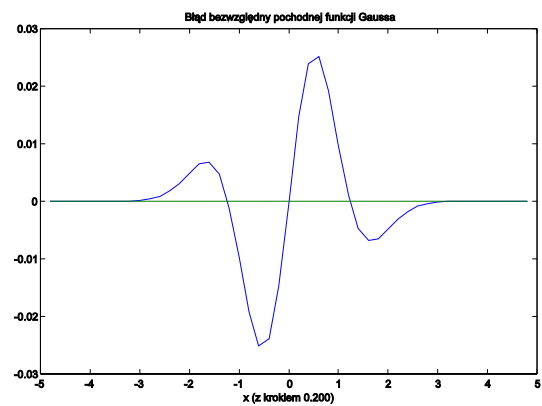
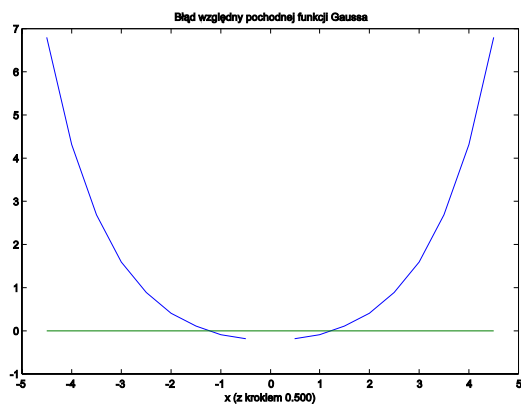
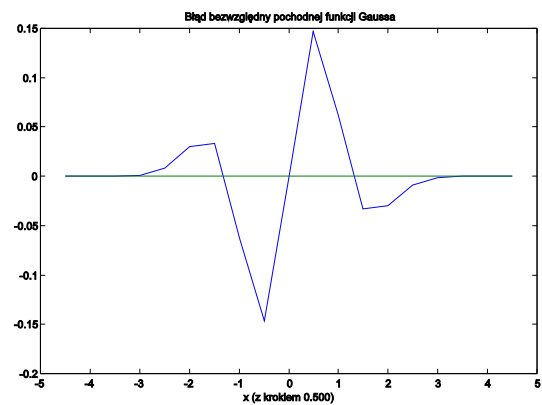
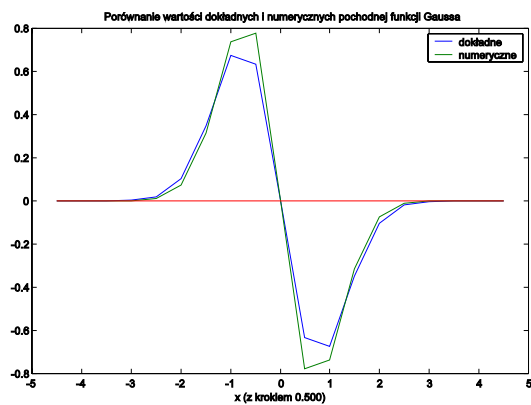
$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + O(\Delta x^2)$$

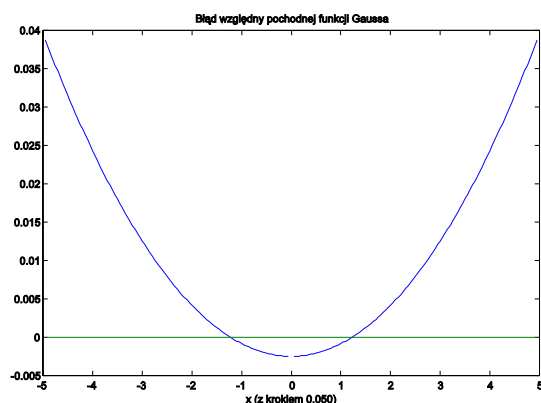
Jest to tzw. różnica centralna, jak widać popełniany błąd jest o rząd wielkości mniejszy niż w przypadku różnic przednich lub wstecznych.

Dla ilustracji rozważymy numeryczne wyznaczanie pochodnych trzech funkcji: e^x , $\sin(x)$ i krzywej „dzwonowej” Gaussa. We wszystkich przypadkach obliczenia dotyczyć będą przedziału $(-5, 5)$, a kroki wynosić będą 0.5, 0.2 i 0.05.









Jak łatwo zauważyć podczas wyprowadzania wzoru na różnicę centralną wzięliśmy pod uwagę dwa punkty, dzięki czemu wykorzystując dwa równania na rozwinięcie Taylora mogliśmy wyrugować składniki zawierające drugą pochodną. Dało to w efekcie większą dokładność nowego wzoru. W taki sam sposób możemy uzyskać wzory o dowolnej dokładności – konieczne będzie jedynie wykorzystywanie coraz większej liczby punktów sąsiednich. Jeśli wykorzystamy jeszcze dwa punkty otrzymamy wzór:

$$f'(x) = \frac{f(x+2\Delta x) - 8f(x+\Delta x) + 8f(x-\Delta x) - f(x-2\Delta x)}{12\Delta x} + O(\Delta x^4)$$

Podobnie wyprowadzamy wzory na pochodne wyższych rzędów, oczywiście konieczne będzie wykorzystywanie większej liczby punktów – jako że konieczne będzie wyrugowanie wszystkich składników zawierających niepożądane pochodne niższego rzędu niż pochodna szukana. I tak dla drugiej pochodnej (podobnie jak w przypadku różnicy centralnej dla pierwszej pochodnej):

$$\begin{aligned} f(x+\Delta x) &= f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + O(\Delta x^3) \\ f(x-\Delta x) &= f(x) - \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + O(\Delta x^3) \\ \text{Y} \quad f(x+\Delta x) - f(x-\Delta x) &= 2f'(x) + \frac{2\Delta x^3}{3!} f'''(x) + O(\Delta x^5) \end{aligned}$$

czyli:

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O(\Delta x^2)$$

Albo z wykorzystaniem pięciu punktów:

$$f'(x) = \frac{-f(x+2\Delta x) + 16f(x+\Delta x) - 30f(x) + 16f(x-\Delta x) - f(x-2\Delta x)}{12\Delta x^2} + O(\Delta x^4)$$

Należy zwrócić uwagę, że umiejętność wyprowadzania wzorów różnicowych na pochodne różnych rzędów (najczęściej jednak pierwszego i drugiego) o różnych dokładnościach i „opierających się” na różnych punktach jest niezbędna podczas rozwiązywania równań różniczkowych cząstkowych (a czasem i zwyczajnych). Myślę tu o warunkach brzegowych drugiego rodzaju – tzn. warunkach określających wartość pochodnej (najczęściej pierwszej) na brzegu (a nie wartość funkcji, jak to ma miejsce w przypadku warunków brzegowych pierwszego rodzaju).

Jako przykład rozpatrzmy jednowymiarowe zagadnienie przepływu ciepła. Rozkład temperatury wewnątrz ciała opisany jest znanym równaniem, które po zapisaniu w postaci różnicowej umożliwia obliczenie temperatury w dowolnym punkcie:

$$\Delta T = \frac{\kappa T}{\Delta x^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2}$$

Jednak nie dotyczy to punktów brzegowych, które nie mają sąsiadów z jednej strony. Zamiast temperatury na brzegu ciała (co odpowiadałoby warunkowi brzegowemu pierwszego rzędu) znana jest temperatura otoczenia i współczynnik

przekazywania ciepła. Strumień ciepła na brzegu i w objętości ciała opisują równania:

$$Q = k \Delta T \quad ; \quad Q = \alpha \frac{MT}{\Delta x}$$

Oczywiście oba te strumienie muszą być sobie równe i ten właśnie fakt można wykorzystać do zapisania warunku brzegowego w postaci różnicowej. Konieczne jest jednak zapisanie takiego wzoru różnicowego na pierwszą pochodną, który dałoby się zastosować w punkcie brzegowym – zatem mogą w nim występować tylko punkty znajdujące się po jednej stronie punktu, w którym obliczamy pochodną. Wzór ten wyprowadzimy następująco:

Zapiszemy rozwinięcia Taylora dla dwóch sąsiednich punktów:

$$T_1 = T_0 + \frac{MT}{\Delta x} \Delta x + \frac{1}{2!} \frac{MT}{\Delta x^2} \Delta x^2 + O(\Delta x^3)$$

$$T_2 = T_0 + \frac{MT}{\Delta x} (2\Delta x) + \frac{1}{2!} \frac{MT}{\Delta x^2} (2\Delta x)^2 + O(\Delta x^3)$$

Z równań tych wyruguję drugą pochodną:

$$4T_1 - T_2 = 3T_0 + 2 \frac{MT}{\Delta x} \Delta x^2 + O(\Delta x^3)$$

$$\frac{MT}{\Delta x} = \frac{4T_1 - T_2 - 3T_0}{2\Delta x} + O(\Delta x^2)$$

W ten sposób otrzymaliśmy równanie różnicowe o dokładności $O(\Delta x^2)$. Może się jednak zdarzyć, że potrzebować będziemy większej dokładności:

$$T_1 = T_0 + \frac{MT}{\Delta x} \Delta x + \frac{1}{2!} \frac{MT}{\Delta x^2} \Delta x^2 + \frac{1}{3!} \frac{MT}{\Delta x^3} \Delta x^3 + O(\Delta x^4)$$

$$T_2 = T_0 + \frac{MT}{\Delta x} (2\Delta x) + \frac{1}{2!} \frac{MT}{\Delta x^2} (2\Delta x)^2 + \frac{1}{3!} \frac{MT}{\Delta x^3} (2\Delta x)^3 + O(\Delta x^4)$$

$$T_3 = T_0 + \frac{MT}{\Delta x} (3\Delta x) + \frac{1}{2!} \frac{MT}{\Delta x^2} (3\Delta x)^2 + \frac{1}{3!} \frac{MT}{\Delta x^3} (3\Delta x)^3 + O(\Delta x^4)$$

Tym razem oczywiście wyrugować trzeba będzie drugą i trzecią pochodną (z tego powodu musieliśmy wziąć trzy równania), aby to zrobić pierwsze z równań mnożymy przez 9, a drugie przez $-9/2$:

$$9T_1 - \frac{9}{2}T_2 + T_3 = \frac{11}{2}T_0 + \frac{MT}{\Delta x} \Delta x + O(\Delta x^4)$$

$$\frac{MT}{\Delta x} = \frac{11T_0 - 9T_1 + 2T_2 - T_3}{11\Delta x} + O(\Delta x^3)$$

To wyrażenie ma już dokładność rzędu Δx^3 .

W ten sposób możemy wyprowadzić wzory różnicowe o dowolnej właściwie dokładności i oparte na dowolnych punktach.