

Wykład V

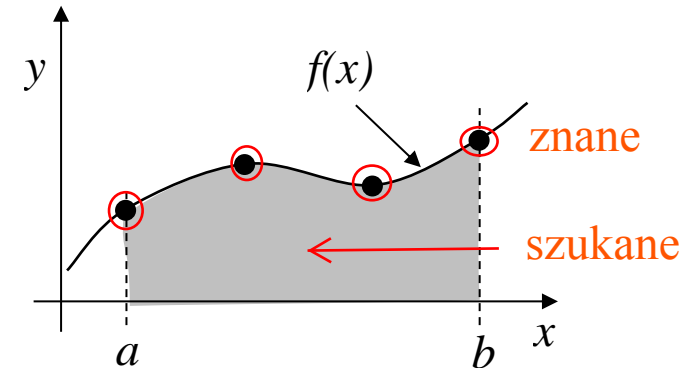
Różniczkowanie i całkowanie numeryczne

- Numeryczne przybliżenia pochodnej
- Filtry różniczkujące w DSP i przetwarzaniu obrazów
- Kwadratury Newtona-Cotesa
- Całkowanie adaptacyjne
- Kwadratury Gaussa
- Całki wielokrotne

Zadanie całkowania numerycznego

Na podstawie znajomości wartości y_i funkcji $y = f(x)$ w punktach x_i (narzuconych lub wybieranych) wyznaczyć wartość I całki oznaczonej (kwadratury) z funkcji w

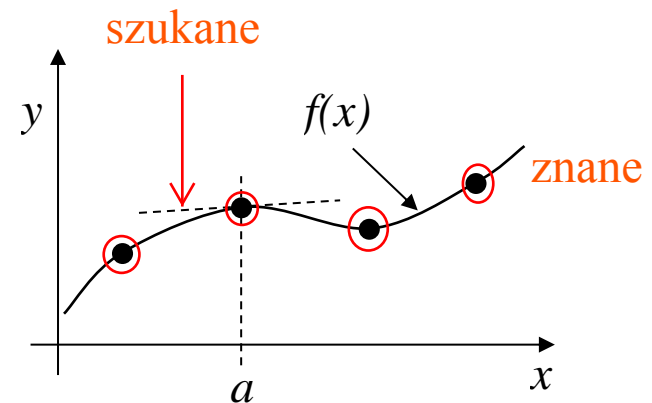
przedziale $\langle a, b \rangle$:
$$I = \int_a^b f(x) dx$$



Zadanie różniczkowania numerycznego

Na podstawie znajomości wartości y_i funkcji $y = f(x)$ w punktach x_i (narzuconych lub wybieranych) wyznaczyć wartość D pochodnej funkcji w punkcie a , czyli nachylenie stycznej do funkcji w punkcie a :

$$D = \left. \frac{df(x)}{dx} \right|_{x=a}$$



Zastosowania całek i różniczek numerycznych

Wyznaczanie wielkości definiowanych całką

- Splot
- Wartość średnia, skuteczna
- Energia sygnału
- Pola powierzchni, objętości
- Prawdopodobieństwo na podstawie funkcji gęstości prawdopodobieństwa
- Prędkość i położenie wyznaczone na podstawie sygnału przyspieszenia

Wartość przybliżona: odpowiadające im operacje sumacyjne na sygnałach dyskretnych
W obliczeniach małej precyzji wystarczają proste metody prostokątów i trapezów.

Wyznaczanie wielkości definiowanych różniczką:

- Czułość przyrządu, nachylenie charakterystyki przetwarzania, wrażliwość układu
- Nachylenie (gradient) w metodach poszukiwania minimum funkcji
- Prędkość i przyspieszenie wyznaczone na podstawie sygnału położenia

Wyznaczanie przybliżone tych wartości np. przez lokalną linearyzację.

W obliczeniach małej precyzji wystarczają proste metody dwupunktowych ilorazów różnicowych.

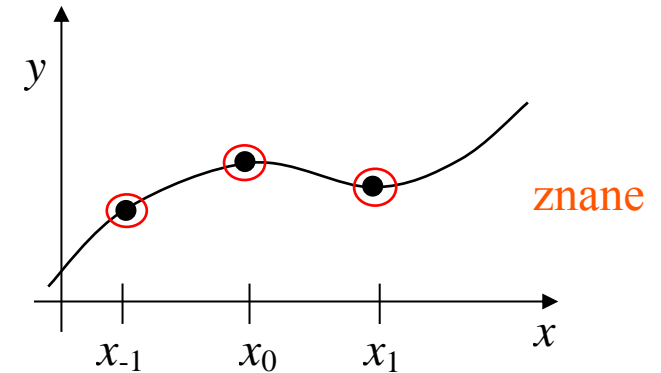
Prosty przypadek na początek

Jakie dobrać wartości współczynników ogólnego wyrażenia:

$$A = c_{-1}y_{-1} + c_0y_0 + c_1y_1$$

żeby wartość A przybliżała:

- całkę w przedziale $\langle x_{-1}, x_1 \rangle$,
- pochodną w punkcie x_0 ,
- drugą pochodną w punkcie x_0



Założmy punkty równoodległe z odstępem h ($x_{-1} = x_0 - h$, $x_1 = x_0 + h$).

Wielkość przybliżana	Wartości c_{-1}, c_0, c_1	Operacja numeryczna	Dokładna wartość dla wielomianu stopnia
Całka (mnożnik h)	1, 1, 0	Suma prostokątów	Zerowego
	0.5, 1, 0.5	Suma trapezów	Pierwszego
	1/3, 4/3, 1/3	Wzór Simpsona	Trzeciego
Pochodna ($1/h$)	0, -1, 1	Iloraz różnicowy wprzód	Pierwszego
	-1, 1, 0	Iloraz różnicowy wstecz	Pierwszego
	(1/(2h)) -1, 0, 1	Iloraz różnicowy centralny	Drugiego
Druga pochodna ($1/h^2$)	1, -2, 1	Iloraz ilorazów wprzód/wstecz	Trzeciego

Analogia do średnich ruchomych, filtrów różniczkujących, dolnoprzepustowych, itd.

Metody uzyskiwania przybliżeń różnicowych pochodnych

Dla węzłów nierównoodległych

- różniczkowanie wielomianu interpolacyjnego w postaci:
 - klasycznej
 - Lagrange'a
 - Newtona

Dla węzłów równoodległych

- różniczkowanie wielomianu interpolacyjnego dla węzłów równoodległych
- wykorzystanie rozwinięcia w szereg Taylora

Generowanie ilorazów różnicowych na podstawie wielomianu interpolacyjnego w postaci Newtona

Przybliżając funkcję $f(x)$ wielomianem interpolacyjnym $P(x)$ na N punktach x_1, \dots, x_N

$$P(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + \dots + a_{N-1}(x - x_1)(x - x_2) \dots (x - x_{N-1})$$

możemy przybliżyć pochodną funkcji poprzez **pochodną wielomianu interpolacyjnego**.

$$P'(x) = a_1 + a_2[(x - x_1) + (x - x_2)] + a_3[(x - x_1)(x - x_2) + (x - x_1)(x - x_3) + (x - x_2)(x - x_3)] + \dots + a_{N-1} \sum_{k=1}^{N-1} \prod_{j=1, j \neq k}^{N-1} (x - x_j)$$

Podstawiając za x węzeł, którego pochodnej poszukujemy (zeruje się większość składników), uzyskujemy formułę różnicową (w przód, wstecz, lub centralną), przy czym węzły interpolacji nie muszą być równoodległe.

Przykład: Przybliżenie pochodnej funkcji wielomianem interpolującym (Newtona) na 3 punktach

$$P(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2), \quad a_0 = y_1, \quad a_1 = \frac{y_2 - y_1}{x_2 - x_1}, \quad a_2 = \frac{y_3 - y_2}{(x_3 - x_2)(x_3 - x_1)} - \frac{y_2 - y_1}{(x_2 - x_1)(x_3 - x_1)}$$

Pochodna w dowolnym punkcie przedziału interpolacji: $f'(x) \approx P'(x) = a_1 + a_2[(x - x_1) + (x - x_2)]$

Dla węzłów równoodległych ($x_2 = x_1 + h$, $x_3 = x_1 + 2h$) wartości pochodnej w węzłach:

$$P'(x_1) = \frac{1}{2h}(-3y_1 + 4y_2 - y_3) \quad P'(x_2) = \frac{1}{2h}(y_3 - y_1) \quad P'(x_3) = \frac{1}{2h}(y_1 - 4y_2 + 3y_3)$$

Wyprowadzanie wzorów i szacowanie dokładności z użyciem szeregu Taylora

(Taylor series expansion) więcej - zob. Mathews J.H., Numerical Methods..., str.333

Na podstawie rozwinięcia Taylora funkcji:

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2!}f''(x)h^2 + \dots$$

możemy szacować dokładność przybliżeń numerycznych w funkcji rzędu pochodnej i kroku h .

Przykład:

rozwinięcie Taylora ograniczone do pierwszego nie skracającego się czynnika:

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2!}f''(x)h^2 + \frac{1}{3!}f'''(x)h^3$$

$$f(x-h) = f(x) - f'(x)h + \frac{1}{2!}f''(x)h^2 - \frac{1}{3!}f'''(x)h^3$$

skąd: $f(x+h) - f(x-h) = 2f'(x)h + \frac{f'''(c_1) + f'''(c_2)}{3!}h^3$

Oszacowanie pochodnej ilorazem różnicowym centralnym (c jest pewnym punktem w przedziale):

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

Wnioski: wzór jest dokładny dla wielomianów do 2 stopnia, błąd jest proporcjonalny do h^2 .

Zadanie: z powyższych równań wyprowadź wzór na oszacowanie drugiej pochodnej.

Wrażliwość przybliżeń różnicowych na błędy reprezentacji i zakłócenia w danych

Stwierdzenie, że błąd jest zależny z określoną potęgą od kroku h sugeruje, że **zmniejszanie kroku wyrażenia różnicowego może tylko poprawić jakość wyniku** (dokładność pochodnej). W tej analizie nie uwzględniono jednak błędu reprezentacji zmiennoprzecinkowej wartości funkcji. Dodatkowo silny wpływ na błędy wyznaczanej pochodnej mogą mieć zakłócenia pomiaru wartości funkcji (np. szumy addytywne w próbkowaniu sygnału). Przeanalizujemy własności dwóch podstawowych wyrażeń różnicowych pod kątem zaburzenia danych.

$$f'(x_0) = \frac{y_1 - y_{-1}}{2h} - \frac{f^{(3)}(c)}{6} h^2$$

$$f''(x_0) = \frac{y_1 - 2y_0 + y_{-1}}{h^2} - \frac{f^{(4)}(c)}{12} h^2$$

Przyjmując **zaburzenie danych** (szum lub błąd reprezentacji) na poziomie ε i **ograniczenie odpowiedniej pochodnej przez liczbę M** , uzyskujemy łączny błąd wyznaczenia pochodnych:

$$\delta' = \frac{\varepsilon}{h} + \frac{Mh^2}{6}$$

$$\delta'' = \frac{4\varepsilon}{h^2} + \frac{Mh^2}{12}$$

Najmniejszy błąd uzyskamy dla optymalnej wartości kroku h (*step-size dilemma*):

$$h'_{opt} = \left(\frac{3\varepsilon}{M} \right)^{1/3} \quad 6? \downarrow$$

$$h''_{opt} = \left(\frac{48\varepsilon}{M} \right)^{1/4}$$

Wnioski:

wyższy rząd pochodnej \rightarrow większy krok \rightarrow mniejsze wzmocnienie błędu reprezentacji

duże zakłócenia w danych \rightarrow lepiej korzystać z wielomianu aproksymującego lub filtrować

Przykład: Różniczkowanie sygnału prędkości silnika synchronicznego w czasie rozruchu

Sygnał prędkości silnika synchronicznego w czasie rozruchu asynchronicznego został zarejestrowany na wyjściu tachoprądnicy i jest zakłócony przez szum na szczotkach, interferencje sieciowe i błąd przetwarzania A/C (częstotliwość próbkowania 50Hz) – zob. rys.1.

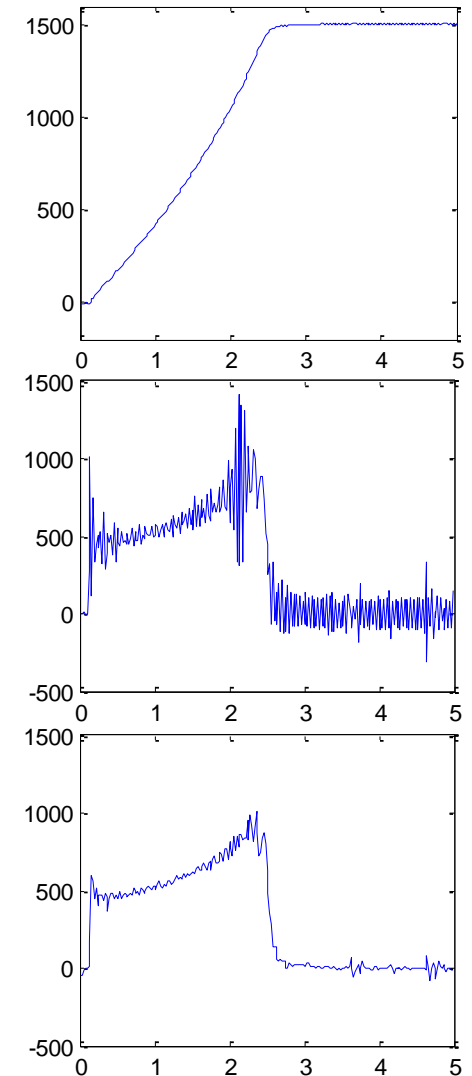
Naszym celem jest przetworzenie programowe tego sygnału dla uzyskania wielkości proporcjonalnej do momentu napędowego wg zależności

$$M_n = J \frac{d\omega}{dt}. \text{ Bezpośrednie przybliżenie pochodnej ilorazem różnicowym}$$

(`diff(wo)./diff(t)`) nie kończy się dobrze – iloraz różnicowy (tak samo jak szerokopasmowa pochodna) jest bardzo wrażliwy na zakłócenia (rys.2). Błąd, który popełniamy to niepotrzebne mocne wzmacnianie szumu w paśmie zakłóceń powyżej pasma sygnału.

Pomysłem na rozwiązanie problemu jest wstępne ograniczenie zakłóceń przez **filtrację dolnopasmową** (np. `[b]=fir1(3,5/25); wof=filter(b,[1],wo);`) i **później zastosowanie ilorazu różnicowego**. Wynik jest lepszy (rys.3).

Krok dalej – połączyć operację filtrowania i różniczkowania .



Różniczkowanie numeryczne a filtry różniczkujące w DSP

Jakie są własności czasowe i częstotliwościowe numerycznych przybliżeń pochodnej w porównaniu z filtrami różniczkującymi FIR używanymi w DSP ?

Zbadajmy schemat/FIR 5-cio punktowy:

$$f'_{O(h^4)}(x) \approx \frac{y_{-2} - 8y_{-1} + 8y_1 - y_2}{12h}$$

(zapis $O(h^4)$ – oznacza dokładność rzędu h^4)

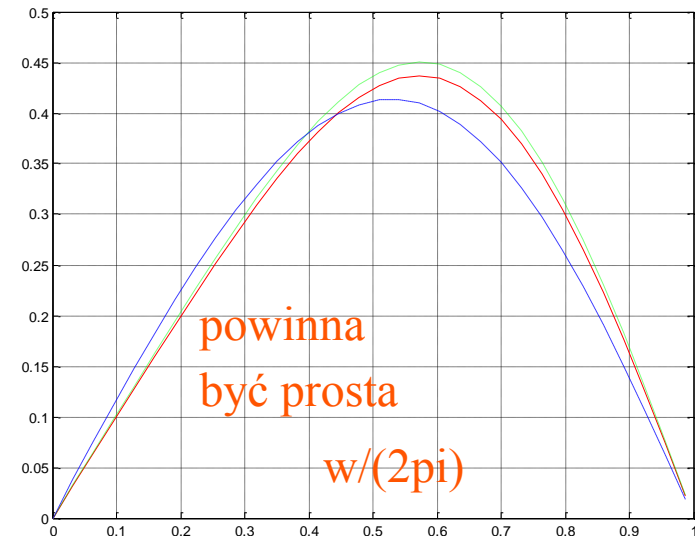
Test właściwości częstotliwościowych:

- schematu różnicowego
- filtru różniczkującego FIR z aproksymacją LS minimum sumy kwadratów odchyłek (*firls*)
- filtru różniczkującego FIR z aproksymacją MIN-MAX Czebyszewa (*remez*)

```

1) w=0:pi/100:pi;
   d=1/12*[-1 8 0 -8 1]
   d =    -0.0833    0.6667    0    -0.6667    0.0833
2) d1=firls(4,[0 0.5 0.7 1],[0 0.5*pi 0 0],'differentiator')
   d1 =    -0.0890    0.6862    0    -0.6862    0.0890
3) d2=remez(4,[0 0.5 0.7 1],[0 0.5*pi 0 0],'differentiator')
   d2 =    -0.0295    0.6474    0    -0.6474    0.0295
plot(w/pi,abs(freqz(d,1,w))/pi,'r',w/pi,abs(freqz(d1,1,w))/pi,'g',w/pi,abs(freqz(d2,1,w))/pi,'b')

```



Różniczkowanie w wielu wymiarach

Przejsie z wyprowadzonymi wzorami do wielu wymiarów nie jest skomplikowane w przypadku **siatki prostokątnej**. Wtedy różniczkowanie po poszczególnych zmiennych przekłada się na **stosowanie reguł różniczkowania numerycznego wzdłuż kierunków siatki**. Przypadkiem trudniejszym jest siatka **nierównomierna** (np. obszar dzielony na **trójkąty**), ale tym się nie będziemy zajmować.

Przykład: wykrywanie krawędzi w obrazach

Częstym zadaniem w przetwarzaniu obrazów jest wyodrębnianie poszczególnych obiektów w scenie. Operacje różniczkowania numerycznego są wykorzystywane do detekcji krawędzi tych obiektów. Proces różniczkowania można przedstawić jako sumowanie wartości pikseli skalowanych nakładaną maską (wagami). Można ten proces również interpretować używając znanego nam już pojęcia splotu i odpowiedzi impulsowej.

Przykładowe maski różniczkujące mają postać:

Wzdłuż x, I rząd;

0	0	0
-1	0	1
0	0	0

wzdłuż y, II rząd;

0	1	0
0	-2	0
0	1	0

Laplasjan;

0	1	0
1	-4	1
0	1	0



Kwadratury Newtona-Cotesa (a interpolacja wielomianowa)**CAŁKOWANIE**

Tak jak w przypadku przybliżeń pochodnych, tak i w przypadku całek oznaczonych narzucającym się rozwiązaniem jest interpolacja funkcji na N węzłach i przybliżenie całki z funkcji całką z wielomianu interpolacyjnego. Tak są definiowane kwadratury Newtona-Cotesa kolejnych rzędów. Przez analogię do ilorazów różnicowych, możemy również oszacować dokładność kwadratur w zależności od pochodnych funkcji i wielkości kroku całkowania, co podajemy poniżej.

Metoda (stopień wielomianu interpolacyjnego) dla stałego kroku h :

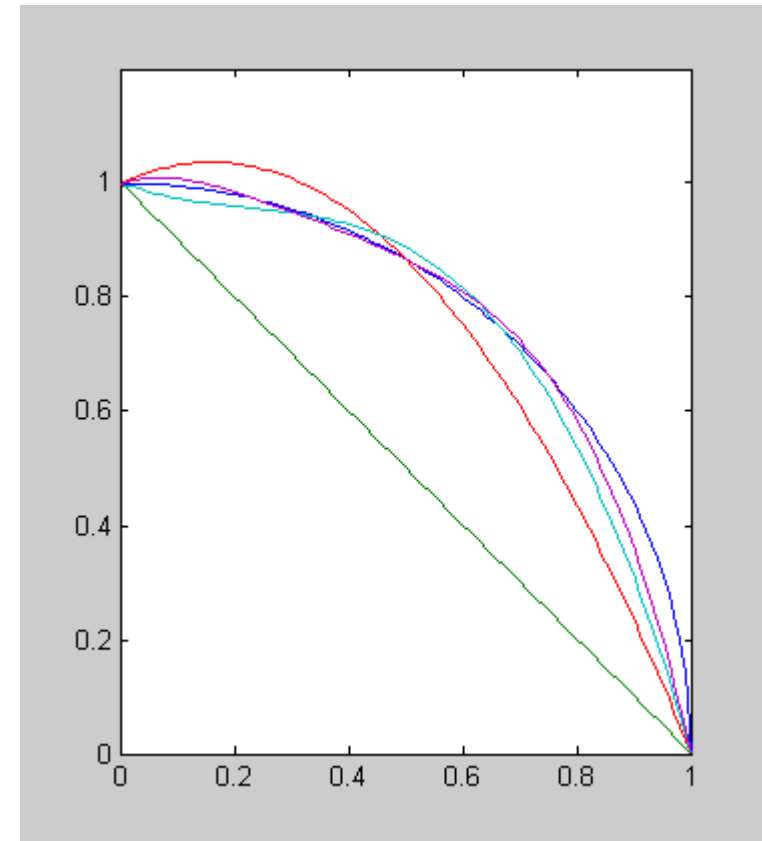
trapezów (I):	$\int_{x_1}^{x_2} f(x) dx = \frac{h}{2} (y_1 + y_2) - \frac{f^{(2)}(c)}{12} h^3$
Simpsona (II):	$\int_{x_1}^{x_2} f(x) dx = \frac{h}{3} (y_1 + 4y_2 + y_3) - \frac{f^{(4)}(c)}{90} h^5$
Simpsona 3/8 (III):	$\int_{x_1}^{x_2} f(x) dx = \frac{3h}{8} (y_1 + 3y_2 + 3y_3 + y_4) - \frac{3f^{(4)}(c)}{80} h^5$
Boole'a (IV):	$\int_{x_1}^{x_2} f(x) dx = \frac{2h}{45} (7y_1 + 32y_2 + 12y_3 + 32y_4 + 7y_5) - \frac{8f^{(6)}(c)}{945} h^7$

Zilustrować rysunkiem na tablicy.

Przykład: kwadratura koła

Pole koła o jednostkowym promieniu ma wartość π . Policzmy tę wartość metodami Newtona-Cotesa kolejnych rzędów na ćwiartce koła.

```
Methods={
'trapezyl',
'simpson2',
'simpson3',
'bool4'
};
Ns=2:5;
W={
[1 1]/2,
[1 4 1]/3,
[1 3 3 1]*3/8,
[7 32 12 32 7]*2/45
};
for i=1:length(Ns);
    method=Methods{i}
    N=Ns(i);
    h=1/(N-1);
    x=0:h:1;
    y=sqrt(1-x.^2);
    kk=sum(W{i}.*y)*h*4
end
```



Dlaczego dostajemy tak niedokładne wyniki ? Jak możemy zwiększyć dokładność ?

Złożone kwadratury Newtona-Cotesa

Na podstawie kwadratury *bazowej* w podstawowym przedziale interpolacji (między dwoma punktami dla wzoru trapezów z interpolacją pierwszego stopnia, między pięcioma dla wzoru Boole'a z wielomianem interpolacyjnym czwartego stopnia) możemy wyznaczyć kwadraturę w dowolnym przedziale przez **sumowanie kwadratur w przedziałach podstawowych**. Wynikają z tego zwięźle **wzory na całkę w przedziale N punktów**. Zwróć uwagę na inne oszacowanie błędu.

Trapezy (*trapz*):

Inne postacie:

$$\int_{x_1}^{x_N} f(x) dx = \frac{h}{2} \sum_{k=2}^N (y_{k-1} + y_k) - \frac{f^{(2)}(c)}{12} (x_N - x_1) h^2$$

$$T(f, h) = \frac{h}{2} (y_1 + 2y_2 + \dots + 2y_{N-1} + y_N),$$

$$T(f, h) = \frac{h}{2} (y_1 + y_N) + h \sum_{k=2}^{N-1} y_k$$

Simpson ($N = 2M + 1$):

Inne postacie:

$$\int_{x_1}^{x_N} f(x) dx = \frac{h}{3} \sum_{k=1}^M (y_{2k-1} + 4y_{2k} + y_{2k+1}) - \frac{f^{(4)}(c)}{180} (x_N - x_1) h^4$$

$$S(f, h) = \frac{h}{3} (y_1 + 4y_2 + 2y_3 + 4y_4 + \dots + 2y_{2M-1} + 4y_{2M} + y_{2M+1})$$

$$S(f, h) = \frac{h}{3} (y_1 + y_N) + \frac{2h}{3} \sum_{k=1}^{M-1} y_{2k+1} + \frac{4h}{3} \sum_{k=1}^M y_{2k}$$

Nie omawiamy rekursywnych zależności pomiędzy kwadraturami używanych do zwiększania dokładności wyniku całkowania i kwadratur Romberga (zob. np. Mathews).

Adaptacyjny dobór kroku metody

Zasada dobierania długości kroku h dla uzyskania żądanej dokładności opiera się na oszacowaniu błędu względem wartości dokładnej na podstawie różnicy między kwadraturą o długości kroku $2h$ i kwadraturą o kroku h . Jako przykład przedstawimy taką implementację dla metody Simpsona.

Dla każdego pięciopunktowego podprzedziału całkowania o węzłach x_1, \dots, x_5 zachodzi zależność:

Kwadratura Simpsona na węzłach x_1, x_3, x_5 : $S(f_{1,3,5}, 2h) = \frac{2h}{3}(y_1 + 4y_3 + y_5) - \frac{16f^{(4)}(c_1)}{90}h^5$

Suma dwóch kwadratur na węzłach x_1, x_2, x_3 i x_3, x_4, x_5 : $S(f_{1,2,3,4,5}, h) = S(f_{1,2,3}, h) + S(f_{3,4,5}, h)$

Gdzie: $S(f_{1,2,3}, h) = \frac{h}{3}(y_1 + 4y_2 + y_3) - \frac{f^{(4)}(c_2)}{90}h^5$,

$S(f_{3,4,5}, h) = \frac{h}{3}(y_3 + 4y_4 + y_5) - \frac{f^{(4)}(c_3)}{90}h^5$

Zakładając równość pochodnych w oszacowaniu błędu uzyskujemy po przekształceniach wynik:

$$\varepsilon = \left| \int_{x_1}^{x_5} f(x) dx - S(f_{1,2,3,4,5}, h) \right| \approx \frac{1}{15} |S(f_{1,2,3,4,5}, h) - S(f_{1,3,5}, 2h)| \quad \text{różnica}$$

Porównując oszacowanie bieżącego błędu z maksymalnym dopuszczalnym błędem możemy zdecydować o rekurencyjnym wywołaniu całkowania dla podprzedziałów (zagęszczanie węzłów).

Proces adaptacji można śledzić w Matlabie z ustawionym parametrem *trace* dla np.funkcji *quad*.

Funkcja `quad` jest implementacją adaptacyjnej metody Simpsona.

```
function [Q,fcnt,warn] = quadstep
(f,a,b,fa,fc,fb,tol,trace,fcnt,hmin,varargin)
%QUADSTEP Recursive core routine for function QUAD.
maxfcnt = 10000;
% Evaluate integrand twice in interior of subinterval [a,b].
h = b - a;
c = (a + b)/2;
if abs(h) < hmin | c == a | c == b
    % Minimum step size reached; singularity possible.
    Q = h*fc;
    warn = 1;
    return
end
x = [(a + c)/2, (c + b)/2];
y = feval(f, x, varargin{:});
fcnt = fcnt + 2;
if fcnt > maxfcnt
    % Maximum function count exceeded; singularity likely.
    Q = h*fc;
    warn = 2;
    return
end
fd = y(1);
fe = y(2);
% Three point Simpson's rule.
Q1 = (h/6)*(fa + 4*fc + fb);
% Five point double Simpson's rule.
Q2 = (h/12)*(fa + 4*fd + 2*fc + 4*fe + fb);
% One step of Romberg extrapolation.
Q = Q2 + (Q2 - Q1)/15;
if ~isfinite(Q)
    % Infinite or Not-a-Number function value encountered.
    warn = 3;
    return
end
if trace
    disp(sprintf('%8.0f %16.10f %18.8e %16.10f',fcnt,a,h,Q))
end
% Check accuracy of integral over this subinterval.
if abs(Q2 - Q) <= tol
    warn = 0;
    return
% Subdivide into two subintervals.
else
    [Qac,fcnt,warnac] =
quadstep(f,a,c,fa,fd,fc,tol,trace,fcnt,hmin,varargin{:});
    [Qcb,fcnt,warncb] =
quadstep(f,c,b,fc,fe,fb,tol,trace,fcnt,hmin,varargin{:});
    Q = Qac + Qcb;
    warn = max(warnac,warncb);
end
```


Kwadratury Gaussa-Legendre'a (z narzuconymi węzłami)

Jak już wiemy, prosty algorytm trapezów daje dokładne wyniki dla wielomianów do 1 stopnia. Pomysłem na lepszą kwadraturę jest przesunięcie węzłów do takiego położenia, żeby całka była dokładna również dla wielomianów wyższego stopnia. Ilustruje to poniższy rysunek dla przedziału całkowania $\langle -1, 1 \rangle$.

Założmy, że zmodyfikowana w ten sposób metoda trapezów ma być dokładna dla wielomianów do stopnia 3. Chcemy tak dobrać węzły x_1, x_2 i skojarzone z nimi wagi w_1, w_2 , żeby równość

$$\int_{-1}^1 f(x) dx = w_1 f(x_1) + w_2 f(x_2)$$

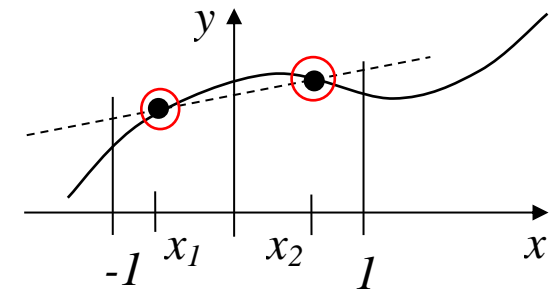
była spełniona dla funkcji $f(x) = 1, x, x^2, x^3$. Daje to układ równań

nieliniowych:

1)
$$\begin{cases} w_1 + w_2 = 2 \\ w_1 x_1 + w_2 x_2 = 0 \\ w_1 x_1^2 + w_2 x_2^2 = \frac{2}{3} \\ w_1 x_1^3 + w_2 x_2^3 = 0 \end{cases}$$
 którego rozwiązaniem są:

2)

$$\begin{cases} w_1 = w_2 = 1 \\ -x_1 = x_2 = \frac{1}{\sqrt{3}} \end{cases}$$



Ostateczne wyrażenie na dwupunktową całkę Gaussa-Legendre'a ma postać

(z oszacowaniem błędu, którego nie wyprowadzamy):
$$\int_{-1}^1 f(x) dx = f\left(\frac{-1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) + \frac{f^{(4)}(c)}{135}$$

Przykładem kwadratury Gaussa wyższego rzędu niech będzie algorytm trzypunktowy:

$$\int_{-1}^1 f(x) dx = \frac{1}{9} \left[5f\left(-\sqrt{3/5}\right) + 8f(0) + 5f\left(\sqrt{3/5}\right) \right] + \frac{f''(0)}{15750}$$

W praktyce położenia węzłów (które są zerami wielomianów Legendre'a) dla kwadratur wyższego rzędu są pamiętane w procedurach całkowania w postaci przeliczonej z określoną dokładnością.

Problemem do rozwiązania jest zastosowanie kwadratur wyprowadzonych dla standardowego przedziału $<-1, 1>$ w dowolnym przedziale $<a, b>$. Możemy to osiągnąć stosując podstawienie zmiennych:

$$t = \frac{a+b}{2} + \frac{b-a}{2}x, \quad dt = \frac{b-a}{2}dx$$

Wtedy całka i N -punktowa kwadratura Gaussa z węzłami/wagami $x_{N,k}, w_{N,k}$ dla przedziału $<-1, 1>$:

$$\int_a^b f(t) dt = \int_{-1}^1 f\left(\frac{a+b}{2} + \frac{b-a}{2}x\right) \frac{b-a}{2} dx \quad G_N(f, a, b) = \frac{b-a}{2} \sum_{k=1}^N w_{N,k} f\left(\frac{a+b}{2} + \frac{b-a}{2}x_{N,k}\right)$$

Funkcja Matlabu `quadl` jest implementacją adaptacyjnej metody Gaussa-Legendre'a z 4/7 węzłami.

Przykład: prawdopodobieństwo wg rozkładu normalnego

Dla zilustrowania dokładności obliczeń całki poszczególnymi metodami porównajmy wartości całki z funkcji gęstości prawdopodobieństwa rozkładu normalnego $N(0,1)$, $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ w przedziale $\pm 3\sigma$, co wg aproksymacji zaimplementowanej w Matlabie jest równe

$\text{erf}(3/\sqrt{2}) = 0.99730020393674$. Funkcja do całkowania została zadana przez

```
f=inline('1/sqrt(2*pi)*exp(-x.^2/2)');
```

Metoda	Implementacja	Wartość całki	Błąd
Trapezy, krok 0.6, 11 węzłów, 10 podprz.	<code>x=-3:0.6:3, w=ones(size(x)), w(2:2:end-1)=2; c=0.6/2*w*f(x)';</code>	0.99653089305	-7.69e-4
Simpson 1/3 , krok 0.6, 11 węzłów, 5 podprz.	<code>x=-3:0.6:3, e=length(x), w=ones(1,e); w(3:2:e-2)=2, w(2:2:e-1)=4; c=0.6/3*w*f(x)';</code>	0.99719530908	-1.05e-4
Gauss, 3 węzły	<code>x=[-sqrt(3/5) 0 sqrt(3/5)], w=1/9*[5 8 5]; c=3*w*f(3*x)';</code>	1.15321648276	0.15
Gauss, po 3 węzły w 3 podprzedziałach	<code>x=[-sqrt(3/5) 0 sqrt(3/5)], w=1/9*[5 8 5]; c=w*f(x-2)'+w*f(x)'+w*f(x+2)';</code>	0.99763817396	3.38e-4
<i>quad</i> , tolerancja 1e-6, 57 wywołań funkcji	<code>[c,i]=quad(f,-3,3);</code>	0.99729991863	-2.9e-7
<i>quadl</i> , tolerancja 1e-6, 48 wywołań funkcji	<code>[c,i]=quadl(f,-3,3);</code>	0.99730020452	5.9e-10

Całkowanie wielokrotne (w wielu wymiarach)

Rozszerzenie algorytmów całkowania do przypadku wielowymiarowego jest nieskomplikowaną operacją. Ponieważ całkowanie po poszczególnych zmiennych można wykonać sekwencyjnie, tzn.:

$$2) \quad \int_c^d \int_a^b f(x, y) dx dy = \int_c^d F(y) dy, \quad F(y) = \int_a^b f(x, y = \text{const}) dx \quad 1)$$

więc realizacja numeryczna będzie polegać na dyskretyzacji całkowanej funkcji w wybranych lub narzuconych węzłach wzdłuż zmiennej y , a następnie dla każdego dyskretnego węzła wzdłuż zmiennej x . Zauważmy, że skala problemu rośnie geometrycznie z ilością zmiennych.

Przykład: Policz prawdopodobieństwo wystąpienia w dwóch niezależnych pomiarach błędu pomiarowego (modelowanego jednym rozkładem normalnym $N(0, \sigma^2)$) większego od σ ale mniejszego od 3σ (uwzględniając znak błędu).

Rozwiązaniem jest całka z dwuwymiarowej funkcji gęstości prawdopodobieństwa zdarzeń niezależnych

$$\int_{\sigma}^{3\sigma} \int_{\sigma}^{3\sigma} p(x, y) dx dy, \text{ gdzie } p(x, y) = (2\pi\sigma^2)^{-1} e^{-\frac{1}{2} \left(\frac{x^2 + y^2}{\sigma^2} \right)}.$$

Numerycznie (trapezy):

```
[X, Y]=meshgrid(1:0.2:3,1:0.2:3);
p=1/(2*pi)*exp(-1/2*(X.^2+Y.^2));
surf(X,Y,p); % zobacz co całkujesz
w=0.2/2*[1 2*ones(1,9) 1];
c=w*p*w'
```

ans = 0.0250

Podsumowanie, kluczowe elementy tego wykładu:

- algorytmy różniczkowania i całkowania numerycznego **bazują na interpolacji wielomianowej** pomiędzy węzłami
- w tym przypadku łatwość wyprowadzania formuł numerycznych wynika z prostych reguł analitycznego różniczkowania i całkowania wielomianów
- **różniczkowanie jest wrażliwe na zaburzenie danych** (błędy reprezentacji, zakłócenia, szумы), praktyczne podejście łączy różniczkowanie i filtrację dolnopasmową (odszumianie)
- różniczkowanie jest stosowane do detekcji zmian (np. krawędzi w obróbce obrazów)
- **całkowanie numeryczne polega na wyznaczaniu pola (kwadratury) pod wielomianem interpolującym**
- możliwy jest **adaptacyjny dobór kroku całkowania** do zmienności funkcji całkowanej dla uzyskania zadanej dokładności poprzez monitorowanie zmian wyniku przy zmianie kroku
- wymuszenie **położenia węzłów interpolacji** zwiększa dokładność wyniku (kwadratury Gaussa-Legendre'a)