Lista zagadnień wymaganych na egzaminie z Metod Numerycznych dla 2 roku Informatyki Stosowanej + pozostałe kierunki, semestr letni 2023/2024

Piotrowski Dawid
Informatyka Stosowana
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

09.06.2024r.

Spis treści

Rozdział 1

Zagadnienia

1. **Błędy numeryczne**
   - klasyfikacja błędów: wejściowe, obcięcia i zaokrąglenia; lemat Wilkinsona i jego interpretacja
   - błędy zaokrągleń podczas sumowania N liczb i sposób ich minimalizacji
   - uwarunkowanie zadania numerycznego na przykładzie obliczania iloczynu skalarnego dwóch wektorów
2. **Metody rozwiązywania układów równań liniowych i nadokreślonych**
   - metoda eliminacji Gaussa z częściowym i pełnym wyborem elementu głównego
   - rozkład LU macierzy metodą Gaussa (wyprowadzenie)
   - ortogonalizacja Grama-Schmidta (dla bazy wektorowej)
   - rozwiązanie układu nadokreślonego przy użyciu rozkładu QR (ogólnie, bez wyznaczania elementów macierzy Q i R)
3. **Wyznaczanie wartości i wektorów własnych**
   - metoda potęgowa poszukiwania wartości i wektorów własnych, redukcja Hottelinga (wyprowadzenie)
   - wyznaczanie wartości i wektorów własnych macierzy trójdiagonalnej metodą bisekcji
   - rozwiązanie uogólnionego problemu własnego (przedstawienie kolejnych kroków postępowania)
4. **Metody iteracyjnego rozwiązywania układów równań liniowych**
   - metody: Jacobiego, Gaussa-Seidla (wyprowadzenie)
   - metoda sprzężonego gradientu (wyprowadzenie)
5. **Metody poszukiwania pierwiastków równania nieliniowego z jedną niewiadomą**
   - metody: bisekcji, siecznych, regula falsi, Newtona (wyprowadzenie wzorów iteracyjnych) oraz modyfikacje tych metod dla pierwiastków wielokrotnych (bez szczegółowej analizy rzędu metody)
6. **Interpolacja**
   - wyprowadzenie wzoru interpolacyjnego Lagrange'a, oszacowanie błędu wzoru interpolacyjnego
7. **Aproksymacja**
   - definicje norm stosowanych w aproksymacji
   - ogólna metoda aproksymacji średniokwadratowej (wyprowadzenie), zapis w postaci macierzowej
   - aproksymacja średniokwadratowa w bazie jednomianów (wyprowadzenie), zapis w postaci macierzowej
8. **Całkowanie numeryczne - kwadratury Newtona-Cotesa**
   - wyprowadzenie ogólnego wzoru na współczynniki kwadratury Newtona-Cotesa
   - wzór trapezów z błędami (wyprowadzenie)

- wzór parabol z błędami (wyprowadzenie)

9. **Minimalizacja wartości funkcji**
   - metoda złotego podziału (wyprowadzenie)
   - metoda interpolacji kwadratowej Powella (wyprowadzenie)
   - metoda Newtona dla funkcji kwadratowej w $\mathbb{R}^n$
10. **Szybka transformacja Fouriera (FFT)**
   - przedstawienie metody Radix-2
   - przykład wykorzystania FFT do szybkiego mnożenia wielomianów
11. **Generatory liczb pseudolosowych**
   - definicja kongruencji
   - definicja generatora liniowego o rozkładzie równomiernym $U(0,1)$ i jego parametry statystyczne (wartość oczekiwana zmiennej losowej, odchylenie standardowe, funkcja autokorelacji oraz ich estymatory)
   - metoda odwracania dystrybuanty dla rozkładów: jednomianowego, eksponencjalnego i normalnego (metoda Boxa-Mullera) (wyprowadzenie)
   - testowanie generatorów: opis testu χ^2
12. **Całkowanie metodą Monte Carlo**
   - metoda podstawowa
   - sposób estymacji wartości oczekiwanej oraz odchylenia standardowego

Rozdział 2

Opracowanie

2.1 Błędy numeryczne

2.1.1 Klasyfikacja błędów

Najprostszy podział błędów numerycznych:

1. Błędy wejściowe
2. Błędy zaokrągleń
3. Błędy obcięcia

2.1.2 Błędy wejściowe

- Występują, gdy dane liczbowe wprowadzane do pamięci komputera odbiegają od wartości dokładnych.
- W szczególności:
  - Gdy wprowadzane dane pomiarowe są obarczone błędami pomiarowymi (np. pomiar wielkości fizycznych takich jak oporu czy napięcia).
  - Gdy ze względu na skończoną długość słowa binarnego dochodzi do wstępnego zaokrąglenia liczb (ułamki dziesiętne lub zaokrąglanie liczb niewymiernych jak np.: e, π).

Przykład – zapis 8-bitowy

Liczba $x_{(10)} = 3.25$ ma reprezentację:

$$x_{(2)} = \underbrace{(0)1101}_{M}\underbrace{(0)10}_{W}$$

Ale dla liczby $x = 0.2$ pojawia się problem:

$$x_{(2)} = 0.0011(0011)...$$

po zaokrągleniu wyniku do najbliższej liczby:

$$x'_{(2)} = 0.001100$$

$$x'_{(10)} = 0.1875$$

co daje błąd bezwzględny równy 0.0125 i błąd względny na poziomie 6.25

2.1.3 Błędy obcięcia

Błędy obcięcia powstają podczas zmniejszania liczby działań, np.:

a) przy obliczaniu wartości szeregów (ucięcie szeregu),

b) wyznaczaniu granic (obliczanie wartości całki),

c) zastępowaniu pochodnej funkcji ilorazem różnicowym.

Przykład

Należy wyznaczyć wartość e^x, korzystamy z rozwinięcia:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}, \quad -\infty < x < \infty$$

Ale numerycznie lepiej zrobić to „nieco inaczej":

$$e^x = e^{E(x)} \cdot e^q$$

gdzie $0 \leq q < 1$, $E(x)$ jest częścią całkowitą liczby x, a q jest częścią ułamkową.

- Pierwszy wyraz jest potęgą, a drugi liczymy wg rozwinięcia.
- Do wyznaczenia pozostaje tylko błąd obcięcia, ponieważ szereg musimy gdzieś "uciąć".
- Szereg ucinamy na n-tym wyrazie (jakie przyjąć n?).

Reszta szeregu (uwzględniamy n wyrazów):

$$R_n(x) = \frac{e^{\theta q}}{(n+1)!} q^{n+1}, \quad 0 < \theta < 1$$

Wnioski:

- Licznik jest ograniczony $< e \approx 2.73$.
- Mianownik rośnie ze względu na silnię.
- Szereg jest szybkozbieżny.

$$R_n(x) = R_n(q) = \frac{q^{n+1}}{(n+1)!} \left(1 + \frac{q}{n+2} + \frac{q^2}{(n+2)(n+3)} + \dots\right)$$

$$< \frac{q^{n+1}}{(n+1)!} \left(1 + \frac{q}{n+2} + \left(\frac{q}{n+2}\right)^2 + \dots\right) \text{ (szereg geometryczny)}$$

$$< \frac{q^{n+1}}{(n+1)!} \frac{1}{1 - \frac{q}{n+2}} \approx \frac{q^{n+1}}{(n+1)!} \frac{n+2}{n+1}$$

$$< \frac{q^{n+1}}{(n+1)!} \frac{n+1}{n}$$

$$0 < R_n(q) < \frac{q^n}{n!} \cdot \frac{q}{n} = u_n \cdot \frac{q}{n}, \quad 0 < q < 1$$

Jak liczymy wartość szeregu? Sumując kolejne wyrazy, aż R_n stanie się akceptowalnie małe.

$$e^x = u_0 + u_1 + u_2 + \dots + R_n(x)$$

Stosujemy schemat rekurencyjny:

$$u_k = \frac{x}{k} u_{k-1}, \quad s_k = s_{k-1} + u_k, \quad k = 0, 1, 2, \dots, n$$

$$R_k = u_k \cdot \frac{x}{k} < \epsilon \quad \leftarrow \text{ Do momentu, aż ten warunek zostanie spełniony.}$$

Parametry startowe: $u_0 = 1$, $s_{-1} = 0$.

Przykład

Obliczmy wartość $\sqrt{e}$ z dokładnością 2.5×10^{-6}.

$$\begin{aligned} u_0 &= 1 \\ u_1 &= 0.5 \\ u_2 &= 0.125 \\ u_3 &= 0.0208333 \\ u_4 &= 0.0026042 \\ u_5 &= 0.0002604 \\ u_6 &= 0.0000217 \\ u_7 &= 0.0000016 \quad < \epsilon \end{aligned}$$

$$e^{1/2} \approx u_0 + u_1 + u_2 + \ldots + u_7 = 1.648721$$

2.1.4 Błędy zaokrągleń

Błędy zaokrągleń pojawiają się podczas wykonywania operacji arytmetycznych i wynikają z ograniczonej reprezentacji liczb zmiennopozycyjnych.

Wielkość błędów zaokrągleń zależy od:

a) dokładności reprezentacji,

b) sposobu zaokrąglania wyniku,

c) rodzaju przeprowadzanej operacji.

Lemat Wilkinsona

Błędy zaokrągleń powstające podczas wykonywania działań zmiennopozycyjnych są równoważne zastępczemu zaburzeniu liczb, na których wykonujemy działania.

Zaburzone liczby możemy zapisać w użytecznej postaci:

$$\mathrm{fl}(x) = x(1+\epsilon_x), \quad \mathrm{fl}(y) = y(1+\epsilon_y), \quad \epsilon_x, \epsilon_y \ll 1$$

Błędy względne zaokrągleń

Mnożenia

$$\frac{\mathrm{fl}(x) \cdot \mathrm{fl}(y) - xy}{xy} = (1+\epsilon_x)(1+\epsilon_y) - 1 = \epsilon_x + \epsilon_y + \epsilon_x\epsilon_y \approx \epsilon_x + \epsilon_y$$

Dzielenia

$$\frac{\mathrm{fl}(x)/\mathrm{fl}(y) - x/y}{x/y} = \frac{(1+\epsilon_x)}{(1+\epsilon_y)} - 1 = \frac{\epsilon_x - \epsilon_y}{1-\epsilon_y} \approx \epsilon_x - \epsilon_y$$

Dodawania i odejmowania

$$\frac{\mathrm{fl}(x) \pm \mathrm{fl}(y) - (x \pm y)}{x \pm y} = \frac{x\epsilon_x \pm y\epsilon_y}{x \pm y} = \frac{x}{x \pm y}\epsilon_x \pm \frac{y}{x \pm y}\epsilon_y$$

Zwłaszcza przy odejmowaniu możemy dostać duży błąd, gdy $x \approx y$ oraz $\epsilon_x \approx -\epsilon_y$, ze względu na kasowanie mianownika.

Wykonywanie kolejnych operacji na wynikach poprzednich operacji prowadzi do kumulacji błędów zaokrągleń. To pesymistyczny scenariusz, ale tego należy oczekiwać podczas obliczeń numerycznych.

Błędy można zmniejszyć:

1. Ustalając odpowiednio sposób i kolejność wykonywanych działań (np. algorytm Kahana dla iloczynu skalarnego).
2. Zwiększając precyzję obliczeń (nie zawsze można - naukowe i inżynierskie w zasadzie zawsze wykonujemy w podwójnej precyzji).
3. Stosując inny algorytm implementujący daną metodę:

- W zasadzie nie stosuje się - sortowanie jest czasochłonne.
- Używamy już silnej arytmetyki.
- Najbardziej użyteczny sposób - zmiana algorytmu.

Przykłady szacowania błędów zaokrągleń - sumowanie N liczb

Jedną z częściej wykonywanych operacji jest sumowanie liczb. Rozważmy sumę:

$$s = \sum_{i=1}^{n} x_i$$

Oznaczenie:

$$s_k = \text{fl}(s_{k-1} + x_k) = s'_{k-1} + x'_k$$

Zgodnie z lematem Wilkinsona:

$$s'_{k-1} = s_{k-1}(1 + \epsilon'_{k-1})$$
$$x'_k = x_k(1 + \epsilon'_k)$$

Gdzie:

$$|\epsilon'_{k-1}| \leq \epsilon$$
$$|c'| \leq \epsilon$$

Liczymy wartość sumy, sumując kolejne wkłady:

$$s = x_1(1 + \epsilon_1^x)(1 + \epsilon_2^s) \cdots (1 + \epsilon_{n-1}^s) \leftarrow \text{najbardziej zaburzony wyraz}$$

$$+x_2(1 + \epsilon_2^x)(1 + \epsilon_3^s) \cdots (1 + \epsilon_{n-1}^s) + \ldots$$
$$+x_{n-1}(1 + \epsilon_{n-1}^x)(1 + \epsilon_{n-1}^s) + x_n(1 + \epsilon_n^x) \leftarrow \text{najmniej zaburzony wyraz}$$

2.1.5 Uwarunkowanie zadania numerycznego na przykładzie obliczania iloczynu skalarnego dwóch wektorów

Uwarunkowanie zadania numerycznego odnosi się do wpływu błędów wejściowych na błędy wyjściowe. Przeanalizujmy to na przykładzie obliczania iloczynu skalarnego dwóch wektorów.

Rozważmy dwa wektory $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ oraz ich iloczyn skalarny:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i$$

Załóżmy, że istnieją błędy w wektorach wejściowych $\mathbf{a}$ i $\mathbf{b}$, takie że:

$$\mathbf{a}' = \mathbf{a} + \Delta\mathbf{a}$$
$$\mathbf{b}' = \mathbf{b} + \Delta\mathbf{b}$$

Iloczyn skalarny z zaburzonymi wektorami wynosi:

$$\mathbf{a}' \cdot \mathbf{b}' = \sum_{i=1}^{n} (a_i + \Delta a_i)(b_i + \Delta b_i)$$

Rozwijając wyrażenie, otrzymujemy:

$$\mathbf{a}' \cdot \mathbf{b}' = \sum_{i=1}^{n} a_i b_i + \sum_{i=1}^{n} a_i \Delta b_i + \sum_{i=1}^{n} b_i \Delta a_i + \sum_{i=1}^{n} \Delta a_i \Delta b_i$$

Przy założeniu, że błędy Δa_i i Δb_i są małe, wyrażenie $\sum_{i=1}^{n} \Delta a_i \Delta b_i$ można zaniedbać. Ostatecznie mamy:

$$\mathbf{a}' \cdot \mathbf{b}' \approx \mathbf{a} \cdot \mathbf{b} + \sum_{i=1}^{n} a_i \Delta b_i + \sum_{i=1}^{n} b_i \Delta a_i$$

Stąd błąd względny w iloczynie skalarnym można wyrazić jako:

$$\frac{|\mathbf{a}' \cdot \mathbf{b}' - \mathbf{a} \cdot \mathbf{b}|}{|\mathbf{a} \cdot \mathbf{b}|} \approx \frac{1}{|\mathbf{a} \cdot \mathbf{b}|} \left(\left| \sum_{i=1}^{n} a_i \Delta b_i \right| + \left| \sum_{i=1}^{n} b_i \Delta a_i \right| \right)$$

Uwarunkowanie obliczania iloczynu skalarnego jest więc związane z długościami wektorów oraz wartością ich iloczynu skalarnego. W praktyce, gdy wektory są ortogonalne (ich iloczyn skalarny jest bliski zeru), zadanie jest słabo uwarunkowane i bardziej podatne na błędy numeryczne.

2.2 Metody rozwiązywania układów równań liniowych i nadokreślonych

2.2.1 Metoda eliminacji Gaussa

Metoda eliminacji Gaussa składa się z dwóch etapów:

1. Przekształcamy macierz do postaci trójkątnej.
2. Rozwiązujemy układ z macierzą trójkątną.

Etap 1: eliminacja zmiennych

Układ pierwotny:

$$A^{(1)}\vec{x} = \vec{b}^{(1)}$$

$$\begin{cases} a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \ldots + a_{1n}^{(1)} x_n = b_1^{(1)} \\ a_{21}^{(1)} x_1 + a_{22}^{(1)} x_2 + \ldots + a_{2n}^{(1)} x_n = b_2^{(1)} \\ \vdots \\ a_{n1}^{(1)} x_1 + a_{n2}^{(1)} x_2 + \ldots + a_{nn}^{(1)} x_n = b_n^{(1)} \end{cases}$$

Odejmujemy od i-tego wiersza ($i = 2, 3, \ldots, n$) wiersz pierwszy pomnożony przez współczynnik:

$$l_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}$$

Po eliminacji zmiennej x_1 otrzymujemy:

$$A^{(2)}\vec{x} = \vec{b}^{(2)}$$

$$\begin{cases} a_{11}^{(2)} x_1 + a_{12}^{(2)} x_2 + \ldots + a_{1n}^{(2)} x_n = b_1^{(2)} \\ a_{22}^{(2)} x_2 + \ldots + a_{2n}^{(2)} x_n = b_2^{(2)} \\ \vdots \\ a_{n2}^{(2)} x_2 + \ldots + a_{nn}^{(2)} x_n = b_n^{(2)} \end{cases}$$

Powtarzamy operację, odejmując od i-tego wiersza ($i = 3, 4, \ldots, n$) wiersz drugi pomnożony przez współczynnik:

$$l_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}$$

Kontynuujemy eliminację zmiennych, aż uzyskamy trójkątny układ równań w postaci:

$$A^{(n)}\vec{x} = \vec{b}^{(n)}$$

$$\begin{cases} a_{11}^{(n)} x_1 + a_{12}^{(n)} x_2 + \ldots + a_{1n}^{(n)} x_n = b_1^{(n)} \\ a_{22}^{(n)} x_2 + \ldots + a_{2n}^{(n)} x_n = b_2^{(n)} \\ \vdots \\ a_{nn}^{(n)} x_n = b_n^{(n)} \end{cases}$$

Etap 2: postępowanie odwrotne

Rozwiązanie (kolejne składowe wektora $\vec{x}$) znajdujemy, stosując wzór iteracyjny dla macierzy trójkątnej.

Wyznaczenie rozwiązania metodą Gaussa (przekształcenie macierzy do postaci trójkątnej) wymaga wykonania:

- M operacji mnożenia i dzielenia:

$$M = \frac{1}{3}n^3 + n^2 - \frac{1}{3}n$$

- D operacji dodawania i odejmowania:

$$D = \frac{1}{3}n^3 + \frac{1}{2}n^2 - \frac{5}{6}n$$

Problem niestabilności metody

Metoda eliminacji w tej postaci jest niestabilna numerycznie z powodu problemu dzielenia przez 0 lub liczbę bliską zeru.

Rozwiązanie problemu niestabilności:

- Częściowy wybór elementów głównych
- Pełny wybór elementów głównych

Częściowy wybór elementów głównych

W k-tym kroku szukamy elementu $|a_{rk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$ i przestawiamy wiersze r oraz k.

Pełny wybór elementów głównych

W k-tym kroku szukamy elementu $|a_{rs}^{(k)}| = \max_{k \leq i,j \leq n} |a_{ij}^{(k)}|$ i przestawiamy wiersze k oraz r i kolumny k oraz s.

- Stosując wybór elementu głównego, rozwiązanie otrzymujemy zawsze.
- W trakcie wyboru elementu głównego należy zmienić także kolejność w $\vec{x}$ i $\vec{b}$.
- Modyfikacji tej można nie stosować dla:
  - Macierzy z dominującą przekątną:

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^{n} |a_{ij}| \quad (i = 1, \ldots, n)$$

  - Macierzy symetrycznej i jednocześnie dodatniookreślonej.

2.2.2 Rozkład LU metodą Gaussa

Metodę Gaussa można użyć do znalezienia takich macierzy L i U, które z macierzą A związane są relacją:

$$A = LU$$

Procedura wyznaczania elementów tych macierzy nosi nazwę rozkładu LU.

Sposób postępowania

1. Mnożenie wiersza pierwszego przez czynnik:

$$l_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}$$

i odjęcie go od i-tego wiersza ($i = 2, \ldots, n$) zastępujemy mnożeniem przez macierz:

$$L^{(1)} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -l_{21} & 1 & 0 & \cdots & 0 \\ -l_{31} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & 1 & \vdots \\ -l_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}_{n \times n}$$

co można zapisać macierzowo:

$$L^{(1)} A^{(1)} = A^{(2)}$$
$$L^{(1)} \vec{b}^{(1)} = \vec{b}^{(2)}$$

2. Eliminacja zmiennej z równań ($i = 3, 4, \ldots, n$) wygląda podobnie. Mnożymy wiersze zmodyfikowanego układu równań o indeksach $i = 3, 4, \ldots, n$ przez czynnik:

$$l_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}$$

i odejmujemy od nich wiersz drugi. Operację tę można przeprowadzić mnożąc układ równań obustronnie przez macierz:

$$L^{(2)} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & -l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & 1 & \vdots \\ 0 & -l_{n2} & 0 & \cdots & 1 \end{bmatrix}$$
$$L^{(2)} A^{(2)} = A^{(3)}$$
$$L^{(2)} \vec{b}^{(2)} = \vec{b}^{(3)}$$

Po wykonaniu $(n-1)$ takich operacji dostajemy:

$$L^{(n-1)} L^{(n-2)} \cdots L^{(1)} A^{(1)} = A^{(n)}$$

$$L^{(n-1)} L^{(n-2)} \cdots L^{(1)} \vec{b}^{(1)} = \vec{b}^{(n)}$$

Macierze $L^{(i)}$ są nieosobliwe - można znaleźć dla każdej macierz odwrotną.
Przemnażając obie strony powyższych równań przez $(L^{(n-1)})^{-1}(L^{(n-2)})^{-1} \cdots (L^{(1)})^{-1}$, otrzymamy:

$$A^{(1)} = (L^{(1)})^{-1}(L^{(2)})^{-1} \cdots (L^{(n-1)})^{-1} A^{(n)}$$

$$\vec{b}^{(1)} = (L^{(1)})^{-1}(L^{(2)})^{-1} \cdots (L^{(n-1)})^{-1} \vec{b}^{(n)}$$

Wprowadzamy oznaczenia:

$$L = (L^{(1)})^{-1}(L^{(2)})^{-1} \cdots (L^{(n-1)})^{-1}$$

$$U = A^{(n)} = (L^{(n-1)} L^{(n-2)} \cdots L^{(1)}) A^{(1)}$$

Otrzymujemy:

$$A = LU$$

Jak znaleźć macierze L i U

$$L^{(1)} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & 1 & \vdots \\ -l_{n1} & 0 & \cdots & 1 \end{bmatrix} \qquad (L^{(1)})^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & 1 & \vdots \\ l_{n1} & 0 & \cdots & 1 \end{bmatrix}$$

Macierz L jest macierzą dolną z jedynkami na diagonali:

$$L = (L^{(1)})^{-1}(L^{(2)})^{-1} \cdots (L^{(n-1)})^{-1}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & 1 & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix}$$

Macierz U jest macierzą górną z niezerowymi elementami na diagonali:

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & \cdots & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

Rozwiązanie układu równań

Dysponując macierzami L i U można rozwiązać układ równań:

$$A\vec{x} = \vec{b}$$

$$LU\vec{x} = \vec{b} \qquad L\underbrace{(U\vec{x})}_{\vec{y}} = \vec{b}$$

poprzez rozwiązanie dwóch układów równań:

$$L\vec{y} = \vec{b}$$

$$U\vec{x} = \vec{y}$$

Rozwiązanie każdego z równań wiąże się z nakładem obliczeń jak dla układu z macierzą trójkątną $\mathcal{O}(n^2)$.

Rozkład LU (eliminacja Gaussa) to nakład rzędu $\mathcal{O}\left(\frac{n^3}{2}\right)$.

Zalety stosowania rozkładu LU

- Duża wydajność dla dużej liczby równań. Rozkład LU opłaca się stosować w przypadku rozwiązywania wielu układów równań z tą samą macierzą współczynników układu A. Każdy układ równań różni się wtedy tylko wektorem wyrazów wolnych. Rozkład LU wykonuje się w takim przypadku tylko raz (ilość operacji $\mathcal{O}(n^3)$).
- Oszczędność zajmowanej pamięci. Elementy macierzy L i U mogą zostać zapisane w macierzy A.
- Jeśli macierz A jest symetryczna i dodatniookreślona, to nie trzeba dokonywać wyboru elementów podstawowych.

2.2.3 Ortogonalizacja Grama-Schmidta dla bazy wektorowej

Dane są dwa wektory liniowo niezależne i nieortogonalne: u_1 i u_2. Należy je przekształcić tak, aby były ortogonalne i unormowane.

$$\{u_1, u_2\} \to \{q_1, q_2\}$$

$$u_i^T u_j \neq 0$$
$$q_i^T q_j = \delta_{i,j}$$

Jako $\mathbf{q_1}$ przyjmujemy kierunek $\mathbf{u_1}$ a wektor normujemy

$$q_1 = \frac{u_1}{\|u_1\|}$$

Dla $\mathbf{u_2}$ odczytujemy

$$u_2 = u_2^{\perp} + u_2^{\parallel} \iff u_2 = u_2^{\perp} + r \cdot \mathbf{q_1} \quad q_1^T \cdot /$$

$$q_1^T u_2 = \underbrace{q_1^T u_2^{\perp}}_{=0} + \underbrace{r q_1^T q_1}_{=1} \implies r = q_1^T u_2$$

$$u_2^{\perp} = u_2 - q_1(q_1^T u_2)$$

Jako drugi wektor (unormowany) przyjmujemy:

$$q_2 = \frac{u_2^{\perp}}{\|u_2^{\perp}\|}$$

Uogólnienie dla k-elementowej bazy w n-wymiarowej przestrzeni

Dla wektorów $\{u_1, u_2, \ldots, u_k\}$ gdzie $u_i \in \mathbb{R}^n$, szukamy bazy wektorów ortonormalnych (ortogonalnych i unormowanych).

Jako pierwszy wektor wybieramy jak poprzednio:

$$q_1 = \frac{u_1}{\|u_1\|}$$

Kolejne wektory ortogonalizujemy do elementów już znalezionych:

$$\tilde{q}_j = u_j - \sum_{i=1}^{j-1} q_i(q_i^T u_j), \quad j = 2, 3, \ldots, k$$

i normalizujemy:

$$q_j = \frac{\tilde{q}_j}{\|\tilde{q}_j\|}$$

Wektory q możemy użyć jako np. kolumn macierzy Q:

$$Q = Q_{n \times k} = [q_1, q_2, \ldots, q_k]$$

$$Q^T Q = D = \operatorname{diag}(d_1, d_2, \ldots, d_k) = I$$

$$Q^T Q = I$$

2.2.4 Rozwiązanie układu nadokreślonego przy użyciu rozkładu QR

Układ równań jest nadokreślony, gdy liczba równań jest większa niż liczba niewiadomych, czyli macierz A jest prostokątna o wymiarach $m \times n$, gdzie $m > n$. W takim przypadku nie zawsze istnieje dokładne rozwiązanie. Zamiast tego szukamy najlepszego przybliżenia w sensie najmniejszych kwadratów.

Dla macierzy A o rozmiarach $m \times n$, w której kolumny są niezależne liniowo, istnieje jednoznaczny rozkład w postaci:

$$A = QR$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \left[\begin{array}{c|c|c|c} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} \end{array}\right] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

Q jest macierzą ortogonalną:

$$Q^T Q = D$$

D jest macierzą diagonalną o elementach nieujemnych:

$$D = \text{diag}(d_1, d_2, \ldots, d_n), \quad d_k > 0 \quad k = 1, \ldots, n$$

R jest macierzą trójkątną górną z elementami:

$$r_{kk} = 1 \quad k = 1, \ldots, n$$

Warunek minimalizacji normy wektora reszt w sensie średniokwadratowym przyjmuje postać:

$$Ax = b \implies A^T Ax = A^T b$$

$$R^T Q^T QRx = R^T Q^T b$$
$$R^T DRx = R^T Q^T b$$
$$DRx = Q^T b$$
$$Rx = D^{-1} Q^T b = y$$

Wiedząc, jak rozwiązać układ z macierzą trójkątną, możemy znaleźć rozwiązanie x.

2.3 Wyznaczanie wartości i wektorów własnych

Wektory własne

Wektor własny macierzy A to taki wektor $\vec{x}$, który po pomnożeniu przez A zmienia tylko swoją długość (skalowanie), a nie kierunek. Formalnie, wektor $\vec{x}$ jest wektorem własnym macierzy A wtedy i tylko wtedy, gdy istnieje skalar λ (wartość własna) spełniający równanie:

$$A\vec{x} = \lambda\vec{x}$$

W powyższym równaniu: - A to macierz kwadratowa $n \times n$, - $\vec{x}$ to wektor własny, - λ to odpowiadająca wartość własna.

Wartości własne

Wartość własna λ to skalar, który odpowiada wektorowi własnemu $\vec{x}$ w równaniu:

$$A\vec{x} = \lambda\vec{x}$$

Wartości własne można znaleźć, rozwiązując równanie charakterystyczne:

$$\det(A - \lambda I) = 0$$

gdzie det oznacza wyznacznik, a I to macierz jednostkowa o tych samych wymiarach co A. Rozwiązania tego równania to wartości własne λ macierzy A.

2.3.1 Metoda potęgowa poszukiwania pojedynczych wartości własnych i wektorów własnych

Załóżmy, że istnieje n liniowo niezależnych wektorów własnych macierzy A, stanowiących bazę przestrzeni liniowej:

$$\{x_1, x_2, x_3, \ldots, x_n\}$$

Wówczas dla dowolnego wektora v_0:

$$v_0 = \sum_{i=1}^{n} a_i x_i$$

Jeśli λ_i są wartościami własnymi macierzy A, to:

$$Av_0 = \sum_{i=1}^{n} a_i \lambda_i x_i$$

$$v_m = A^m v_0 = \sum_{i=1}^{n} a_i \lambda_i^m x_i$$

Zakładamy, że wartości własne tworzą ciąg:

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

Jeśli λ_1 jest dominującą wartością własną oraz wektor v_0 ma składową w kierunku x_1, to wówczas zachodzi:

$$\lim_{m\to\infty} \frac{A^m v_0}{\lambda_1^m} = a_1 x_1$$

Z czego można wysnuć wniosek, że wartość własną można obliczyć następująco:

$$\lambda_1 = \lim_{m\to\infty} \frac{y^T v_{m+1}}{y^T v_m}$$

Dla dowolnego wektora y nieortogonalnego do x_1. Zazwyczaj y ma 1 na pozycji elementu o największym module w v_m, a na pozostałych 0.

Zbieżność metody

$$v_m = \lambda_1^m \left[a_1 x_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1} \right)^m a_i x_i \right]$$

Zależy od $\left(\frac{\lambda_1}{\lambda_2}\right)^m$, ale również od współczynników a_i, czyli od wyboru v_0. Jeśli wartość własna o największym module jest zespolona, to ciąg nie jest zbieżny.

Wyznaczanie wektora własnego x_1

Ponieważ:

$$v_m \approx \lambda_1^m a_1 x_1$$

więc unormowany wektor własny będzie miał postać:

$$x_1 = \frac{v_m}{|v_m|}$$

Jeśli wartość własna jest pierwiastkiem wielokrotnym równania charakterystycznego, to metoda jest zbieżna, bo składnik z λ_1 dominuje:

$$v_m = A^m v_0 = \lambda_1^m \sum_{i=1}^{k} a_i x_i + \sum_{i=k+1}^{n} \lambda_i^m a_i x_i$$

Uwaga: problem pojawia się, gdy $\lambda_1 = -\lambda_j$, tj. identyczne moduły generują oscylacje (wtedy wybieramy ciąg wektorów v_x).

Redukcja Hotellinga (macierze symetryczne)

Za wektor v przyjmujemy lewy wektor własny przynależny do wartości własnej λ_2, ale na ogół nie znamy lewych wektorów. Metoda jest więc skuteczna tylko w przypadku macierzy symetrycznych, wtedy lewe wektory są identyczne z prawymi:

$$v = x_1$$

Iloczyn zewnętrzny/tensorowy (tworzymy operator macierzowy):

$$W_1 = A - \lambda_1 x_1 x_1^T$$

lub rekurencyjnie:

$$W_0 = A$$
$$W_i = W_{i-1} - \lambda_{i-1} x_{i-1} x_{i-1}^T, \quad i = 1, 2, \ldots, n-1$$

2.3.2 Wyznaczanie wartości własnych macierzy trójdiagonalnej metodą bisekcji

Po przekształceniu macierzy A do postaci:

$$J = \begin{bmatrix} \delta_1 & \overline{\gamma_2} & 0 & \cdots & 0 \\ \gamma_2 & \delta_2 & \overline{\gamma_3} & \cdots & 0 \\ 0 & \gamma_3 & \delta_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \overline{\gamma_n} \\ 0 & 0 & 0 & \gamma_n & \delta_n \end{bmatrix}$$

Ta macierz jest nieredukowalna, jeśli spełniony jest warunek:

$$\gamma_i \neq 0, \quad i = 2, \ldots, n$$

W przeciwnym razie można ją zapisać w postaci:

$$J = \begin{bmatrix} J_1 & & & 0 \\ & J_2 & & \\ & & \ddots & \\ 0 & & & J_k \end{bmatrix}$$

i rozwiązywać problem dla mniejszych macierzy $J_1, J_2, \ldots, J_k$, ponieważ widmo wartości J oraz ciągu J jest identyczne.

W metodzie bisekcji wykorzystujemy wielomian charakterystyczny macierzy J.

Sposób wyznaczania wartości wielomianu charakterystycznego J

1. Zakładamy dowolną wartość λ.
2. Obliczamy wartość $W(\lambda)$ rozwijając wyznacznik względem kolejnych kolumn macierzy:

$$\omega_i(\lambda) = \det(J_i - \lambda I)$$

co prowadzi do procedury rekurencyjnej:

$$\begin{aligned} &\omega_0(\lambda) = 1 \\ &\omega_1(\lambda) = \delta_1 - \lambda \\ &\quad \cdots \\ &\omega_i(\lambda) = (\delta_i - \lambda)\omega_{i-1}(\lambda) - |\gamma_i|^2 \omega_{i-2}(\lambda), \quad i = 2, 3, \ldots, n \\ &W(\lambda) = \omega_n(\lambda) \end{aligned}$$

Macierz jest hermitowska, więc $\delta_i, |\gamma_i|^2 \in \mathbb{R}$, i wszystkie wartości pośrednie też są rzeczywiste.

Wyznaczanie wartości własnej w metodzie bisekcji

Wybieramy dowolną liczbę i obliczamy wartość wielomianu charakterystycznego rekurencyjnie, zachowując informacje o ciągu:

$$\omega_0(\lambda), \omega_1(\lambda), \ldots, \omega_n(\lambda)$$

a następnie korzystamy z poniższych twierdzeń:

Twierdzenie 1. Jeżeli elementy $\gamma_2, \gamma_3, \ldots, \gamma_n$ (pozadiagonalne) są niezerowe, to wartości własne macierzy J są pojedyncze.

Twierdzenie 2. Jeżeli elementy $\gamma_2, \gamma_3, \ldots, \gamma_n$ (pozadiagonalne) są niezerowe, to ciąg wartości:

$$\omega_0(\lambda), \omega_1(\lambda), \ldots, \omega_n(\lambda)$$

spełnia poniższe warunki:

a) Jeżeli $\omega_i(\lambda) = 0$ dla pewnego $i < n$, to przyjmujemy:

$$\omega_{i-1}(\lambda) \cdot \omega_{i+1}(\lambda) < 0$$

b) Jeżeli $\omega_n(z) = \omega(z) \neq 0$, to liczba zmian znaków sąsiednich liczb w ciągu jest równa liczbie wartości własnych macierzy J mniejszych od λ.

c) Jeżeli $\omega_n(\lambda) = 0$, to λ jest wartością własną macierzy J, a ponadto jest tyle wartości własnych mniejszych niż λ, ile nastąpiło zmian znaków w ciągu.

Zalety metody bisekcji

- Metoda bisekcji jest bardzo dokładna. - Umożliwia obliczenie wartości własnej o określonym indeksie k.

Liczba iteracji potrzebna do wyznaczenia λ_k wynosi:

$$IT = \log_2 \left(\frac{\beta_0 - \alpha_0}{\rho} \right)$$

gdzie β_0 - przedział poszukiwań wartości własnej, ρ - dokładność wyznaczenia wartości własnej.

Wady metody bisekcji

- Uzyskiwanie dużych wartości ciągu $\omega_0(\lambda), \omega_1(\lambda), \ldots, \omega_n(\lambda)$, jeśli λ znacznie różni się od wartości własnych J.

Wektory własne w metodzie bisekcji

Znając k-tą wartość własną macierzy J, wektor własny x_k wyznaczamy według wzorów:

$$\begin{aligned} x_1 &= 1 \\ x_2 &= \frac{\lambda_k - \delta_1}{\gamma_2} \\ x_{i+1} &= \frac{(\lambda_k - \delta_i)x_i - \gamma_i x_{i-1}}{\gamma_{i+1}}, \quad i = 2, 3, \ldots, n-1 \end{aligned}$$

2.3.3 Uogólniony problem własny

Uogólniony problem własny definiujemy następująco:

$$A\vec{x} = \lambda B\vec{x}$$

Najprościej byłoby przekształcić powyższe równanie tak, aby przeprowadzić je do zwykłego problemu własnego:

$$B^{-1}A\vec{x} = C\vec{x} = \lambda\vec{x}$$

Problemem pozostaje jednak, jak znaleźć B^{-1}?

W przypadku, gdy B oraz A są macierzami symetrycznymi, możemy posłużyć się rozkładem LL^T (w ogólnym przypadku można skorzystać z rozkładu LU):

$$B = LL^T$$

$$BB^{-1} = I = LL^T(L^T)^{-1}L^{-1}$$
$$B^{-1} = (L^T)^{-1}L^{-1}$$

Wówczas, wykorzystując rozkład LL^T, można znaleźć macierz podobną do $B^{-1}A$:

$$L^T(B^{-1}A)(L^T)^{-1} = L^T(L^T)^{-1}L^{-1}A(L^T)^{-1} = L^{-1}A(L^T)^{-1} = G$$

Dzięki temu przekształceniu, macierz G jest symetryczna jak A i posiada identyczne widmo wartości własnych (ale inne wektory własne):

$$G\vec{y} = \lambda\vec{y}$$

Jak znaleźć G? Najpierw należy znaleźć macierz F:

$$F = A(L^T)^{-1}$$

Rozwiązując układ równań:

$$FL^T = A \implies LF^T = A^T = A$$

A następnie wyznaczamy G:

$$G = L^{-1}F$$

Rozwiązując układ równań:

$$LG = F$$

Rozkład LL^T wymaga wykonania $\frac{n^3}{6}$ mnożeń, a wyznaczenie macierzy G $\frac{2}{3}n^3$. Macierz G jest symetryczna, więc w celu wyznaczenia jej wartości i wektorów własnych korzystamy z metod przeznaczonych dla tej klasy macierzy.

Wektory własne macierzy A wyznaczamy przekształcając wektory macierzy G lub rozwiązując układ:

$$L^T\vec{x} = \vec{y}$$

2.4 Metody iteracyjnego rozwiązywania układów równań liniowych

2.4.1 Metoda Jacobiego

Rozważmy układ równań liniowych w postaci:

$$A\vec{x} = \vec{b}$$

gdzie:

$$\vec{x} = \left[x_1, x_2, \ldots, x_n\right]^T, \quad \vec{b} = \left[\beta_1, \beta_2, \ldots, \beta_n\right]^T$$

Dla dowolnie wybranego przybliżenia rozwiązania $\vec{x}_0$, chcemy tak przekształcać iteracyjnie wektor $\vec{x}^{(k)}$, aby doprowadzić do znikania składowych wektora reszt w k-tej iteracji:

$$(b - A\vec{x}^{(k)})_i = 0$$

co można zapisać jako:

$$\beta_i - \sum_{j=1}^{n} a_{ij}\xi_j^{(k)} = 0$$

Rozwiązując układ równań liniowych metodami iteracyjnymi, mamy:

$$a_{ii}\xi_i^{(k)} = \beta_i - \sum_{\substack{j \\ j\neq i}}^{n} a_{ij}\xi_j^{(k)}, \quad i = 1, 2, \ldots, n$$

Składowe wektora reszt znikają w kolejnych iteracjach, więc możemy zapisać:

$$\xi_i^{(k+1)} = \frac{1}{a_{ii}}\left(\beta_i - \sum_{\substack{j \\ j\neq i}}^{n} a_{ij}\xi_j^{(k)}\right)$$

oraz dla całego wektora:

$$\vec{x}^{(k+1)} = -D^{-1}(L+U)\vec{x}^{(k)} + D^{-1}\vec{b}$$

gdzie: - D jest diagonalną częścią macierzy A, - L jest dolną trójkątną częścią macierzy A (bez diagonali), - U jest górną trójkątną częścią macierzy A (bez diagonali).

W metodzie Jacobiego obliczamy kolejno wszystkie składowe nowego przybliżenia wektora rozwiązań:

$$\vec{x}^{(k+1)} = \left[x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_n^{(k+1)}\right]^T$$

2.4.2 Metoda Gaussa-Seidla

Metoda Gaussa-Seidla różni się od metody Jacobiego tym, że obliczone już składniki $\xi_i^{(k)}$ dla $i = 1, 2, \ldots, j$ wykorzystywane są w obliczeniach składników $i+1, i+2, \ldots, n$.

Równanie iteracyjne można zapisać jako:

$$\beta_i - \sum_{j=1}^{i-1} a_{ij}\xi_j^{(k+1)} - a_{ii}\xi_i^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}\xi_j^{(k)} = 0$$

Skąd wynika:

$$\xi_i^{(k+1)} = \frac{1}{a_{ii}} \left(-\sum_{j=1}^{i-1} a_{ij}\xi_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}\xi_j^{(k)} + \beta_i \right)$$

Możemy to również zapisać dla całego wektora:

$$b - L\vec{x}^{(k+1)} - D\vec{x}^{(k+1)} - U\vec{x}^{(k)} = 0$$

co daje:

$$\vec{x}^{(k+1)} = -D^{-1}L\vec{x}^{(k+1)} - D^{-1}U\vec{x}^{(k)} + D^{-1}\vec{b}$$

Rozwiązywanie układów równań liniowych (URL) metodami iteracyjnymi.

Metody Jacobiego i Gaussa-Seidla można zapisać ogólnie w postaci:

$$M\vec{x}^{(k+1)} = N\vec{x}^{(k)} + \vec{b} = (M - A)\vec{x}^{(k)} + \vec{b}$$

gdzie:

$$A = M - N$$

Dla metody Jacobiego:

$$M = D$$

Dla metody Gaussa-Seidla:

$$M = D + L$$

Uwaga: Aby użyć powyższego wzoru, należy "odwrócić"macierz M, tj. musimy mieć możliwość łatwego rozwiązania układu równań. Macierz diagonalna lub trójkątna daje taką możliwość.

2.4.3 Metoda sprzężonego gradientu

Założenia: - $\vec{x}_d$ jest rozwiązaniem dokładnym, - ciąg wektorów $\{\vec{y}_1, \vec{y}_2, \vec{y}_3, \ldots, \vec{y}_n\}$ stanowi bazę w n-wymiarowej przestrzeni euklidesowej.

Warunek ortogonalności bazy:

$$\vec{y}_i^T \vec{y}_j = \begin{cases} 0 & \text{gdy } i \neq j \\ \neq 0 & \text{gdy } i = j \end{cases}$$

Różnicę rozwiązania dokładnego i przybliżonego możemy zapisać w postaci kombinacji liniowej elementów bazy:

$$\Delta\vec{x} = \vec{x}_d - \vec{x}_i = \sum_{j=1}^{n} \alpha_j \vec{y}_j$$

Jeśli elementy bazy są ortogonalne, to można łatwo wyznaczyć współczynniki kombinacji liniowej:

$$\vec{y}_m^T(\vec{x}_d - \vec{x}_i) = \sum_{j=1}^{n} \alpha_j \vec{y}_m^T \vec{y}_j = \sum_{j=1}^{n} \alpha_j \delta_{m,j} \vec{y}_m^T \vec{y}_j = \alpha_m \vec{y}_m^T \vec{y}_m$$

$$\alpha_m = \frac{\vec{y}_m^T(\vec{x}_d - \vec{x}_i)}{\vec{y}_m^T \vec{y}_m}$$

Jednak powyższy wzór wymaga modyfikacji, ponieważ nie znamy wektora $\vec{x}_d$. Wiemy jednak, że:

$$A\vec{x}_d = \vec{b}$$

Policzmy ponownie współczynnik α, ale użyjmy nowej bazy A-ortogonalnej:

$$\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \ldots, \vec{v}_n\} \iff \underbrace{\vec{v}_i^T A \vec{v}_j = \delta_{i,j} \vec{v}_i^T A \vec{v}_i}_{Warunek \quad A-ortogonalności \quad bazy}$$

$$A(\vec{x}_d - \vec{x}_i) = \sum_{j=1}^{n} \alpha_j A \vec{v}_j$$

$$\vec{v}_m^T A(\vec{x}_d - \vec{x}_i) = \sum_{j=1}^{n} \alpha_j \vec{v}_m^T A \vec{v}_j = \sum_{j=1}^{n} \alpha_j \delta_{m,j} \vec{v}_m^T A \vec{v}_j = \alpha_m \vec{v}_m^T A \vec{v}_m$$

$$\alpha_j = \frac{\vec{v}_j^T A(\vec{x}_d - \vec{x}_i)}{\vec{v}_j^T A \vec{v}_j} = \frac{\vec{v}_j^T (\vec{b} - A\vec{x}_i)}{\vec{v}_j^T A \vec{v}_j} = \frac{\vec{v}_j^T \vec{r}_i}{\vec{v}_j^T A \vec{v}_j}$$

Żądamy więc, aby wektory bazy spełniały warunek A-ortogonalności (wektory A-sprzężone):

$$\vec{v}_j^T A \vec{v}_i = 0 \iff i \neq j$$

Dla macierzy dodatniookreślonej zachodzi warunek:

$$\vec{v}_i^T A \vec{v}_i \neq 0$$

Jak skonstruować bazę A-ortogonalną?
Jeśli dysponujemy zwykłą bazą wektorów $\{\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n\}$, to możemy ją poddać procesowi ortogonalizacji Grama-Schmidta:

$$\vec{v}_1 = \vec{u}_1$$

$$\vec{v}_{i+1} = \vec{u}_{i+1} - \sum_{k=1}^{i} \beta_{i+1,k} \vec{v}_k, \quad \beta_{i+1,k} = \frac{\vec{v}_k^T A \vec{u}_{i+1}}{\vec{v}_k^T A \vec{v}_k}$$

Jak utworzyć ciąg wektorów $\vec{u}_i$?
W metodzie sprzężonego gradientu (CG) bazę stanowią wektory reszt (kierunki gradientów), które dzięki A-ortogonalizacji są sprzężone.

W podstawowej metodzie CG w każdej iteracji należy wykonać dwa mnożenia macierz-wektor $A\vec{v}_i$ $A\vec{r}_{i+1}$. To te dwie operacje determinują nakład obliczeń. Algorytm metody CG można przedstawić w alternatywnej postaci, gdzie wymagamy tylko jednego mnożenia macierz-wektor:

Kolejne przybliżenia w podstawowej metodzie CG wyznaczamy zgodnie z poniższym schematem:

$$\vec{v}_1 = \vec{r}_1 = \vec{b} - A\vec{x}_1$$

$$\alpha_i = \frac{\vec{r}_i^T \vec{r}_i}{\vec{v}_i^T A \vec{v}_i}$$

$$\vec{x}_{i+1} = \vec{x}_i + \alpha_i \vec{v}_i$$

$$\vec{r}_{i+1} = \vec{r}_i - \alpha_i A \vec{v}_i$$

$$\beta_i = \frac{\vec{r}_{i+1}^T \vec{r}_{i+1}}{\vec{r}_i^T \vec{r}_i}$$

$$\vec{v}_{i+1} = \vec{r}_{i+1} - \beta_i \vec{v}_i$$

Dzięki A-ortogonalności w każdej iteracji wystarczy wyznaczyć tylko jeden współczynnik β (reszta współczynników znika).

Maksymalna liczba iteracji w metodzie CG wynosi $n + 1$, więc jest metodą skończoną. Zazwyczaj do uzyskania akceptowalnego rozwiązania wystarcza wykonanie znacznie mniejszej liczby iteracji. Rozważyliśmy tylko szczególny przypadek: macierz A jest symetryczna i dodatniookreślona.

Jeśli macierz jest niesymetryczna, konieczne jest użycie innej metody. Metody te wykorzystują podprzestrzeń Kryłowa (różni je sposób generowania ciągu wektorów $\vec{v}$).

2.5 Metody poszukiwania pierwiastków równania nieliniowego z jedną niewiadomą

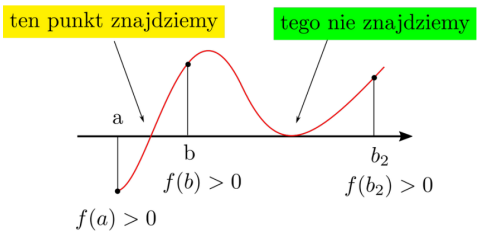
Metoda bisekcji

Rozwiązania szukamy w przedziale, w którym znajduje się miejsce zerowe funkcji, w tzw. przedziale izolacji pierwiastka (wewnątrz tego przedziału pierwsza pochodna funkcji nie zmienia znaku). Przedział wyznacza się, badając zmianę znaku funkcji.

Założenia:

- W przedziale (a, b) znajduje się dokładnie jeden pierwiastek. Gdy są dwa pierwiastki po zawężeniu przedziału, znak funkcji na obu krańcach może być identyczny, wtedy metoda zawodzi.
- Na końcach przedziału wartości funkcji mają różne znaki:

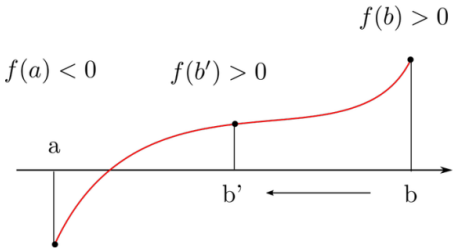
$$f(a) \cdot f(b) < 0$$



Rysunek 2.1: Metoda bisekcji

Algorytm metody bisekcji:

1. Dzielimy przedział izolacji na pół:
$$x_1 = \frac{b+a}{2}$$
2. Sprawdzamy, czy spełniony jest warunek $f(x_1) = 0$. Jeśli tak, to mamy rozwiązanie. Jeśli nie, to przechodzimy do kolejnego punktu.
3. Z dwóch przedziałów $[a, x_1]$ oraz $[x_1, b]$ wybieramy ten, w którym wartości funkcji na krańcach przedziałów mają różne znaki:
$$f(a) \cdot f(x_1) < 0 \quad \text{lub} \quad f(x_1) \cdot f(b) < 0$$



Powtarzamy kroki 1-3, co powoduje, że długości kolejnych przedziałów maleją:

$$|x_k - x_{k+1}| = \frac{1}{2^k}(b-a)$$

Zbieżność metody iteracyjnej: Ciąg przybliżeń jest zbieżny, gdy:

$$\lim_{k\to\infty} x_k = a \quad \text{i} \quad f(a) = 0$$

Zdefiniujmy błąd rozwiązania w k-tej iteracji:

$$\epsilon_k = a - x_k$$

W punkcie $x = a$ metoda jest rzędu p, jeśli istnieje liczba rzeczywista $p \geq 1$, dla której zachodzi:

$$\lim_{k\to\infty} \frac{|x_{k+1} - a|}{|x_k - a|^p} = \lim_{k\to\infty} \frac{|\epsilon_{k+1}|}{|\epsilon_k|^p} = C \neq 0$$

Liczbę C nazywamy stałą asymptotyczną błędu:

$$|\epsilon_{k+1}| = C|\epsilon_k|^p$$

Im wyższa wartość p, tym metoda jest wydajniejsza - błąd maleje szybciej.

2.5.1 Metoda siecznych

Metoda siecznych jest modyfikacją metody Regula Falsi. Prostą przeprowadza się przez dwa ostatnie przybliżenia x_k i x_{k-1} (metoda dwupunktowa). Kolejne przybliżenia w metodzie siecznych wyznacza się według relacji rekurencyjnej:

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$$

Zbieżność metody jest większa niż w metodzie Regula Falsi, rząd metody wynosi:

$$p = \frac{1 + \sqrt{5}}{2} \approx 1.618$$

Należy dodatkowo przyjąć, że $|f(x)|$ ma tworzyć ciąg wartości malejących. Jeśli w kolejnej iteracji $|f(x)|$ zaczyna rosnąć, należy przerwać obliczenia i ponownie wyznaczyć punkty startowe zawężając przedział izolacji.

2.5.2 Metoda Reguła Falsi

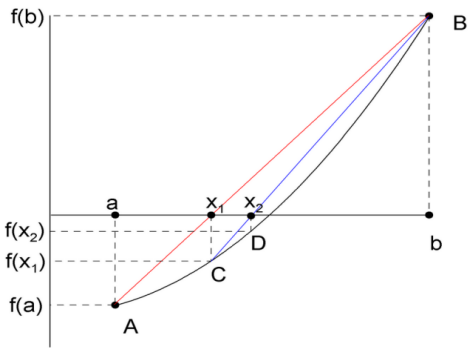
W metodzie tej wykorzystuje się założenie istnienia lokalnej liniowości funkcji (fałszywe, stąd nazwa). Zakładamy ponadto, że w przedziale $[a, b]$ funkcja ma tylko jeden pierwiastek:

$$f(a) \cdot f(b) < 0$$

Funkcja jest klasy C^2 (wielomian 1 stopnia), pierwsza i druga pochodna nie zmieniają znaku w przedziale $[a, b]$.

Rząd metody jak dla bisekcji:

$$p = 1$$



Rysunek 2.2: Idea metody Regula Falsi dla funkcji wypukłej

Algorytm metody Regula Falsi:

1. Przez punkty A i B prowadzimy prostą o równaniu:

$$\frac{f(b) - f(a)}{b - a}(x - a) + f(a)$$

Punkt x_1, w którym prosta przecina oś Ox, przyjmuje się za pierwsze przybliżenie szukanego pierwiastka równania:

$$x_1 = a - \frac{f(a)(b - a)}{f(b) - f(a)}$$

2. Sprawdzamy warunek, czy $f(x_1) = 0$. Jeśli tak, to przerywamy obliczenia.

3. Jeśli $f(x_1) \neq 0$, to sprawdzamy na końcach którego przedziału $[a, x_1]$ lub $[x_1, b]$ wartości funkcji mają różne znaki. Przez te punkty prowadzimy kolejną prostą i powtarzamy poprzednie kroki.

Uwaga: jeśli w przedziale $[a, b]$

- $f(x) > 0$ oraz $f(x) > 0$, to B jest punktem stacjonarnym (prawy brzeg ustalony).
- $f(x) > 0$ oraz $f(x) < 0$, to A jest punktem stacjonarnym.

Metoda generuje ciąg przybliżeń. Elementy ciągu wyznaczamy iteracyjnie:

$$x_{k+1} = x_k - \frac{f(x_k)(b-a)}{f(b) - f(a)}$$

gdzie $k = 1, 2, 3, \ldots$.

Uwagi:

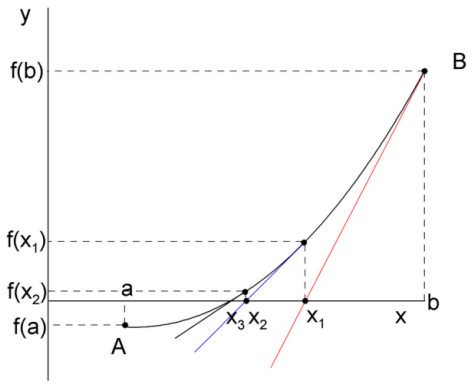
- Metoda Regula Falsi jest zbieżna dla dowolnej funkcji ciągłej w przedziale (a, b), jeśli wartość pierwszej pochodnej jest ograniczona i różna od zera w otoczeniu pierwiastka.
- Obliczenia przerywa się, jeśli dwa kolejne przybliżenia różnią się o mniej niż założone ϵ.
- Wadą jest wolna zbieżność ciągu przybliżeń; rząd metody $p = 1$.

2.5.3 Metoda Newtona-Raphsona (metoda stycznych)

Sposób postępowania:

1. Z końca przedziału $[a, b]$, w którym funkcja ma ten sam znak co druga pochodna, należy poprowadzić styczną do wykresu funkcji $y = f(x)$. (W ten sposób wykonujemy jedną iterację mniej, bo zbliżamy się do pierwiastka z jednej strony).
2. Styczna przecina oś OX w punkcie x_1, który stanowi pierwsze przybliżenie rozwiązania.
3. Sprawdzamy, czy $f(x_1) = 0$. Jeśli nie, to z tego punktu prowadzimy kolejną styczną.
4. Druga styczna przecina oś OX w punkcie x_2, który stanowi drugie przybliżenie.
5. Kroki 3-4 powtarzamy iteracyjnie, aż spełniony będzie warunek:

$$|x_{k+1} - x_k| \leq \epsilon$$



Rysunek 2.3: Ilustracja metody Newtona-Raphsona

Wzór iteracyjny w metodzie Newtona-Raphsona
Równanie stycznej poprowadzonej z punktu B:

$$y - f(b) = f'(b)(x - b)$$

Dla $y = 0$, otrzymujemy pierwsze przybliżenie:

$$x_1 = b - \frac{f(b)}{f'(b)}$$

Równanie stycznej w k-tym przybliżeniu:

$$y - f(x_k) = f'(x_k)(x - x_k)$$

Wzór iteracyjny na położenie k-tego przybliżenia pierwiastka równania nieliniowego w metodzie Newtona:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (k = 1, 2, \ldots)$$

Metoda Newtona jest metodą jednopunktową.

Szacowanie rzędu metody Newtona

Korzystamy z rozwinięcia Taylora w miejscu ostatniego przybliżenia x_k:

$$f(a) = f(x_k) + f'(x_k)(a - x_k) + \frac{f''(\zeta)}{2}(a - x_k)^2, \quad \zeta \in [a, x_k]$$

Wiemy, że $f(a) = 0$, więc po przekształceniu wzoru Taylora otrzymujemy:

$$0 = f(x_k) + f'(x_k)(a - x_k) + \frac{f''(\zeta)}{2}(a - x_k)^2$$

Rozwiązując względem a:

$$a = \underbrace{x_k - \frac{f(x_k)}{f'(x_k)}}_{x_{k+1}} \underbrace{- \frac{f''(\zeta)}{2f'(x_k)}(a - x_k)^2}_{\epsilon_{k+1}}$$

Szacowanie błędu:

$$\epsilon_{k+1} = a - x_{k+1} = -\frac{f''(\zeta)}{2f'(x_k)}\epsilon_k^2 \quad \Big/ \frac{1}{\epsilon_k^2}$$

$$\frac{\epsilon_{k+1}}{\epsilon_k^2} = -\frac{f''(\zeta)}{2f'(x_k)} \approx C \implies p = 2$$

Z tego wynika, że rząd metody Newtona-Raphsona wynosi $p = 2$:

$$\epsilon_{k+1} \approx C\epsilon_k^2, \quad \text{gdzie} \quad C = -\frac{f''(\zeta)}{2f'(x_k)}$$

2.5.4 Poszukiwanie pierwiastków wielokrotnych równania nieliniowego

Liczbę a nazywamy r-krotnym ($r \geq 2$) pierwiastkiem równania $f(x) = 0$ wtedy i tylko wtedy, gdy jest $(r-1)$-krotnym pierwiastkiem równania:

$$f'(x) = 0$$

Metody połowienia, Regula Falsi, siecznych nadają się do poszukiwania pierwiastków tylko o nieparzystej krotności. Rząd metody siecznych obniża się (wolniejsza zbieżność).

Metoda Newtona pozwala znaleźć pierwiastki o parzystej i nieparzystej krotności. Aby utrzymać rząd metody (przyśpieszyć zbieżność), stosuje się zmodyfikowane wzory iteracyjne.

Modyfikacje wzorów iteracyjnych

Znana krotność r pierwiastka równania - wówczas możemy wykorzystać tę informację w metodzie Newtona:

$$x_{k+1} = x_k - r\frac{f(x_k)}{f'(x_k)}, \quad r = 1, 2, \ldots$$

W praktyce bardzo rzadko znamy wartość r, przez co zastosowanie powyższego wzoru jest mocno ograniczone. Rząd metody pozostaje bez zmian $p = 2$.

Nieznana krotność pierwiastka - stosujemy podstawienie:

$$u(x) = \frac{f(x)}{f'(x)}$$

Dla funkcji pomocniczej $u(x)$ krotność pierwiastka wynosi $r = 1$.

Zmodyfikowany wzór iteracyjny w metodzie siecznych:

$$x_{k+1} = x_k - u(x_k)\frac{x_k - x_{k-1}}{u(x_k) - u(x_{k-1})}$$

Zmodyfikowany wzór iteracyjny w metodzie Newtona:

$$x_{k+1} = x_k - \frac{u(x_k)}{u'(x_k)}$$

$$u'(x_k) = 1 - \frac{f''(x_k)}{f'(x_k)}u(x_k)$$

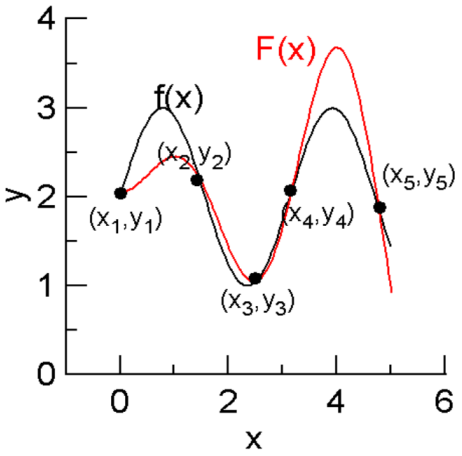
Liczenie drugiej pochodnej może być kłopotliwe.

2.6 Interpolacja

2.6.1 Wyprowadzenie wzoru interpolacyjnego Lagrange'a

Interpolacja polega na wyznaczeniu przybliżonych wartości funkcji w punktach nie będących węzłami oraz na oszacowaniu błędu przybliżonych wartości. W przedziale $[a, b]$ danych jest $n+1$ różnych punktów $x_0, x_1, x_2, \ldots, x_n$ (węzły interpolacji) oraz wartości funkcji $y = f(x)$ w tych punktach:

$$f(x_0) = y_0, \quad f(x_1) = y_1, \quad \ldots, \quad f(x_n) = y_n$$



Rysunek 2.4: Ilustracja interpolacji

Problem interpolacji sprowadza się do znalezienia funkcji interpolującej $F(x)$, która w węzłach przyjmuje wartości takie jak funkcja $y = f(x)$, czyli funkcja interpolowana (której postać funkcyjna może nie być nawet znana).

Do czego służy interpolacja?

- Dla stablicowanych wartości funkcji i określonych położeń węzłów szukamy przybliżenia funkcji pomiędzy węzłami.
- Zagęszczanie tablic.
- Efektywniejsze (szybsze) rozwiązywanie równań nieliniowych.
- Interpolacja wielomianowa pozwala lokalnie przybliżyć dowolną funkcję (np. wyrażającą się skomplikowaną formułą) wielomianem, co ułatwia analizę rozwiązań w modelach fizycznych, np. ułatwia całkowanie, numeryczne obliczanie wartości wyrażeń itp.
- Wykorzystuje się w całkowaniu numerycznym.
- W dwóch i trzech wymiarach do modelowania powierzchni.

Interpolację najczęściej przeprowadza się przy pomocy:

- Wielomianów algebraicznych (nieortogonalne lub ortogonalne).
- Wielomianów trygonometrycznych.
- Funkcji sklejanych.

Powyższe funkcje stanowią bazy funkcyjne - funkcja interpolująca jest kombinacją elementów bazowych.

Idea interpolacji wielomianowej

Twierdzenie: Istnieje dokładnie jeden wielomian interpolacyjny stopnia co najwyżej n ($n \geq 0$), który w punktach $x_0, x_1, \ldots, x_n$ przyjmuje wartości $y_0, y_1, \ldots, y_n$.

Dowód:

Niech $n+1$ węzłów rozmieszczonych jest w dowolny sposób w $[a, b]$. Szukamy wielomianu interpolacyjnego w postaci:

$$W_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

Podstawiając do $W_n(x)$ kolejno $x_0, x_1, \ldots, x_n$ dostajemy układ $n+1$ równań na współczynniki:

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \ldots + a_n x_0^n &= y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \ldots + a_n x_1^n &= y_1 \\ &\vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \ldots + a_n x_n^n &= y_n \end{aligned}$$

Macierz współczynników układu to macierz Vandermonde'a:

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix}$$

Wyznacznik tej macierzy jest wyznacznikiem Vandermonde'a:

$$D = \begin{vmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{vmatrix} = \prod_{0 \leq i < j \leq n} (x_j - x_i) \neq 0$$

Wniosek: Układ ma dokładnie jedno rozwiązanie

$$a_i = \frac{1}{D} \sum_{j=0}^{n} y_j D_{ij}$$

gdzie D_{ij} są wyznacznikami macierzy dopełnień algebraicznych. Wielomian interpolacyjny (Lagrange'a) opisuje się wzorem:

$$W_n(x) = \sum_{j=0}^{n} y_j \phi_j(x)$$

gdzie funkcje $\phi_j(x)$ są wielomianami co najwyżej stopnia n.

Interpolacja Lagrange'a

Korzystając z poprzedniego wyniku, podstawiamy:

$$a_i = \frac{1}{D} \sum_{j=0}^{n} y_j D_{ij}$$

Wówczas:

$$W_n(x) = \sum_{j=0}^{n} y_j \phi_j(x)$$

Funkcje $\phi_j(x)$ są wielomianami co najwyżej stopnia n. Dla dowolnego x, zachodzi zależność:

$$W_n(x_i) = \sum_{j=0}^{n} y_j \phi_j(x_i) = y_i$$

skąd wynika warunek:

$$\phi_j(x_i) = \begin{cases} 0 & \text{gdy } j \neq i \\ 1 & \text{gdy } j = i \end{cases}$$

Wniosek: aby określić funkcje $\phi_j(x)$, należy znaleźć taki wielomian, który zeruje się w węzłach

$$x_0, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n$$

oraz przyjmuje wartość 1 w węźle x_j. Szukaną funkcją mógłby być poniższy wielomian:

$$\phi_j(x) = \lambda(x - x_0)(x - x_1)\ldots(x - x_{j-1})(x - x_{j+1})\ldots(x - x_n)$$

który w x_j przyjmuje wartość 1:

$$1 = \lambda(x_j - x_0)(x_j - x_1)\ldots(x_j - x_{j-1})(x_j - x_{j+1})\ldots(x_j - x_n)$$

Otrzymaliśmy wielomian węzłowy Lagrange'a:

$$\phi_j(x) = \frac{(x - x_0)(x - x_1)\ldots(x - x_{j-1})(x - x_{j+1})\ldots(x - x_n)}{(x_j - x_0)(x_j - x_1)\ldots(x_j - x_{j-1})(x_j - x_{j+1})\ldots(x_j - x_n)}$$

Szukaną funkcją przyjmuje postać:

$$W_n(x) = y_0 \frac{(x - x_1)(x - x_2)\ldots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)}$$

$$+y_1 \frac{(x - x_0)(x - x_2)\ldots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\ldots(x_1 - x_n)} + \ldots + y_n \frac{(x - x_0)(x - x_1)\ldots(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)\ldots(x_n - x_{n-1})}$$

Lub krócej, oznaczając:

$$\omega_n(x) = (x - x_0)(x - x_1)\ldots(x - x_n)$$

wielomian interpolacyjny Lagrange'a ma postać:

$$W_n(x) = \sum_{j=0}^{n} y_j \frac{\omega_n(x)}{\omega_n'(x_j)(x - x_j)}$$

2.6.2 Oszacowanie błędu wzoru interpolacyjnego

Interesuje nas różnica pomiędzy wartościami funkcji interpolowanej i interpolującej w pewnym punkcie $x \in [x_0, x_n]$ nie będącym węzłem:

$$\epsilon(x) = f(x) - W_n(x)$$

Zakładamy, że funkcja $f(x)$ jest $n + 2$ krotnie różniczkowalna (n + 1 krotnie różniczkowalną funkcją jest wielomian W_n).

Wprowadzamy funkcję pomocniczą:

$$\epsilon(x) = K(x - x_0)\ldots(x - x_n)$$

Jeśli znajdziemy wartość K i zażądamy znikania funkcji pomocniczej w węzłach, to dodatkowy wyraz będzie opisywał błąd interpolacji:

$$\varphi(x) = f(x) - W_n(x) - K(x - x_0)(x - x_1)\ldots(x - x_n)$$

gdzie K jest stałą, która spełnia warunek interpolacji:

$$\varphi(x_0) = \varphi(x_1) = \ldots = \varphi(x_n) = 0$$

Wartość współczynnika K dobieramy tak, aby pierwiastkiem funkcji $\varphi(x)$ był punkt $\overline{x}$. Wówczas możemy zapisać warunek na stałą K:

$$K = \frac{f(\overline{x}) - W_n(\overline{x})}{(\overline{x} - x_0)(\overline{x} - x_1)\ldots(\overline{x} - x_n)} = \frac{f(\overline{x}) - W_n(\overline{x})}{\omega_n(\overline{x})}$$

Mianownik jest różny od zera, więc funkcja $\varphi(x)$ jest $n+2$ krotnie różniczkowalna. Pochodna funkcji $\varphi(x)$ ma co najmniej jedno miejsce zerowe w przedziale ograniczonym jej miejscami zerowymi (twierdzenie Rolle'a), więc ma ich co najmniej $n+1$. Każda kolejna pochodna ma o jedno miejsce zerowe mniej, istnieje zatem taki punkt ξ, że $\varphi^{(n+1)}(\xi) = 0$.

Podobnie dla wielomianu interpolującego:

$$W_n^{(n+1)}(x) = 0 \quad \text{oraz} \quad \omega_n^{(n+1)}(x) = (n+1)!$$

Następnie, $n+1$ pochodna funkcji pomocniczej ma postać:

$$\varphi^{(n+1)}(x) = f^{(n+1)}(x) - K(n+1)!$$

Podstawiamy $x = \xi$:

$$K = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

Wówczas oszacowanie błędu wzoru interpolacyjnego ma postać:

$$\epsilon(x) = f(x) - W_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_n(x)$$

Oznaczmy kres górny modułu $n+1$ pochodnej:

$$M_{n+1} = \sup_{x\in[a,b]} |f^{(n+1)}(x)|$$

$$|f(x) - W_n(x)| \leq \frac{M_{n+1}}{(n+1)!}|\omega_n(x)|$$

Wzór ten określa górną granicę błędu interpolacji Lagrange'a. Można go użyć do oszacowania błędu bezwzględnego wzoru interpolacyjnego pod warunkiem, że znamy maksymalną wartość $n+1$ pochodnej $f(x)$ w zadanym przedziale.

2.7 Aproksymacja

Aproksymacja funkcji

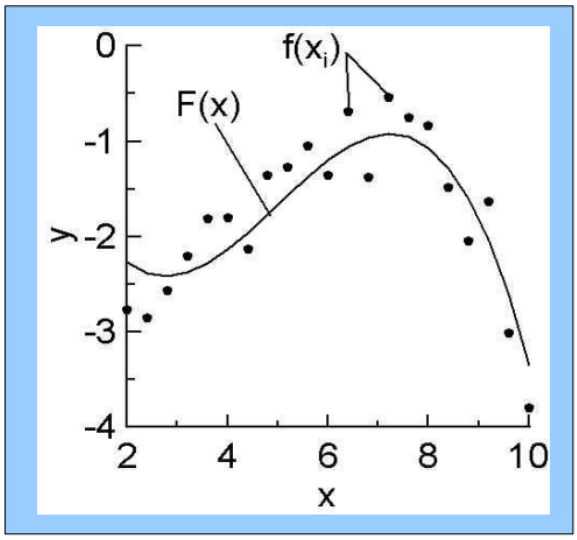
Aproksymacja liniowa funkcji $f(x)$ polega na wyznaczeniu współczynników $a_0, a_1, a_2, \ldots, a_m$ funkcji aproksymującej:

$$F(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \ldots + a_m\varphi_m(x)$$

gdzie $\varphi_i(x)$ są funkcjami bazowymi $(m+1)$-wymiarowej podprzestrzeni liniowej X_{m+1} $(X_{m+1} \subseteq X)$.

Żądamy, aby funkcja $F(x)$ spełniała warunek:

$$\|f(x) - F(x)\| = \min$$



Rysunek 2.5: Ilustracja aproksymacji

Wybór podprzestrzeni i bazy zależy od rodzaju problemu:

- Podprzestrzeń funkcji trygonometrycznych z bazą:

$$\{1, \sin(x), \cos(x), \sin(2x), \cos(2x), \ldots, \sin(kx), \cos(kx)\}$$

- Podprzestrzeń wielomianów stopnia m z bazą: $\{1, x, x^2, \ldots, x^m\}$
- Podprzestrzeń funkcji o własnościach ściśle związanych z własnościami rozważanego problemu, np.: $\{\exp(a_0 + a_1 x + a_2 x^2)\}$

2.7.1 Definicje norm stosowanych w aproksymacji

- **Norma Czebyszewa**

$$\|f(x) - F(x)\| = \sup_{a \le x \le b} |f(x) - F(x)|$$

- **Norma L_2**

$$\|f(x) - F(x)\| = \left(\int_a^b |f(x) - F(x)|^2 \, dx \right)^{\frac{1}{2}}$$

- **Norma L_2 z wagą**

$$\|f(x) - F(x)\| = \left(\int_a^b w(x) |f(x) - F(x)|^2 \, dx \right)^{\frac{1}{2}}$$

  gdzie $w(x)$ jest nieujemną ciągłą funkcją wagową.

  Jeśli funkcja $f(x)$ jest określona na dyskretnym zbiorze punktów, wówczas norma L_2 z wagą przyjmuje postać:

$$\|f(x) - F(x)\| = \left(\sum_{i=0}^{n} w(x_i) [f(x_i) - F(x_i)]^2 \right)^{\frac{1}{2}}$$

2.7.2 Aproksymacja średniokwadratowa

Dla funkcji ciągłej $f(x)$ określonej w przedziale $[a, b]$ poszukujemy minimum wartości całki:

$$\|F(x) - f(x)\| = \int_a^b w(x) [F(x) - f(x)]^2 \, dx$$

lub sumy, gdy funkcja jest określona na dyskretnym zbiorze $n + 1$ punktów (metoda najmniejszych kwadratów):

$$\|F(x) - f(x)\| = \sum_{i=0}^{n} w(x_i) [F(x_i) - f(x_i)]^2, \quad w(x_i) \ge 0,\ i = 0, 1, 2, \ldots, n$$

Metoda aproksymacji średniokwadratowej

Dysponując układem funkcji bazowych w podprzestrzeni X:

$$\varphi_i(x), \quad i = 0, 1, \ldots, m$$

szukamy wielomianu $F(x)$ będącego najlepszym przybliżeniem średniokwadratowym funkcji $f(x)$ na zbiorze $X = \{x_0, x_1, \ldots, x_n\}$:

$$F(x) = \sum_{i=0}^{m} a_i \varphi_i(x)$$

Dla $F(x)$ liczymy normę L_2:

$$H(a_0, a_1, \ldots, a_m) = \sum_{j=0}^{n} w(x_j) [f(x_j) - \sum_{i=0}^{m} a_i \varphi_i(x_j)]^2 = \sum_{j=0}^{n} w(x_j) R_j^2$$

gdzie R_j jest odchyleniem w punkcie x_j.

Szukamy minimum funkcji H (wielu zmiennych) ze względu na współczynniki $a_0, a_1, \ldots, a_m$:

$$\frac{\partial H}{\partial a_k} = 0, \quad k = 0, 1, \ldots, m$$

Warunek ten generuje $m+1$ równań liniowych z $m+1$ niewiadomymi:

$$\frac{\partial H}{\partial a_k} = -2\sum_{j=0}^{n} w(x_j)[f(x_j) - \sum_{i=0}^{m} a_i\varphi_i(x_j)]\varphi_k(x_j) = 0, \quad k = 0, 1, \ldots, m$$

Powyższy układ równań zwany jest układem normalnym. Ponieważ funkcje bazowe są liniowo niezależne, istnieje więc dokładnie jedno rozwiązanie minimalizujące wartość H. Układ równań można zapisać w postaci macierzowej (zakładamy $\varphi_0(x) = 1$):

$$D^T D\vec{a} = D^T\vec{f}$$

$$D = \begin{bmatrix} \varphi_0(x_0) & \ldots & \varphi_m(x_0) \\ \varphi_0(x_1) & \ldots & \varphi_m(x_1) \\ \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \ldots & \varphi_m(x_n) \end{bmatrix}$$

Uwaga:

- Macierz D może nie być kwadratowa, np. w tzw. regresji liniowej baza jest dwuelementowa $(1, x)$, a węzłów może być dowolna ilość.
- $D^T D$ jest macierzą kwadratową i symetryczną o rozmiarach $(m+1) \times (m+1)$.

$$\vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \quad \vec{f} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

2.7.3 Aproksymacja średniokwadratowa w bazie jednomianów

Jako bazę przyjmujemy ciąg jednomianów:

$$1, x, x^2, \ldots, x^m$$

Warunek minimum przyjmuje postać:

$$\frac{\partial H}{\partial a_k} = 0, \quad k = 0, 1, 2, \ldots, m$$

$$\sum_{j=0}^{n}[f(x_j) - \sum_{i=0}^{m} a_i x_j^i]x_j^k = 0, \quad k = 0, 1, 2, \ldots, m$$

Po zmianie kolejności sumowania:

$$\sum_{i=0}^{m} a_i \left(\sum_{j=0}^{n} x_j^{i+k}\right) = \sum_{j=0}^{n} f(x_j)x_j^k$$

I wprowadzeniu oznaczeń:

$$g_{ik} = \sum_{j=0}^{n} x_j^{i+k} \quad \text{oraz} \quad \rho_k = \sum_{j=0}^{n} f(x_j)x_j^k$$

Otrzymujemy układ normalny:

$$\sum_{i=0}^{m} a_i g_{ik} = \rho_k \implies G^T\vec{a} = \vec{\rho}$$

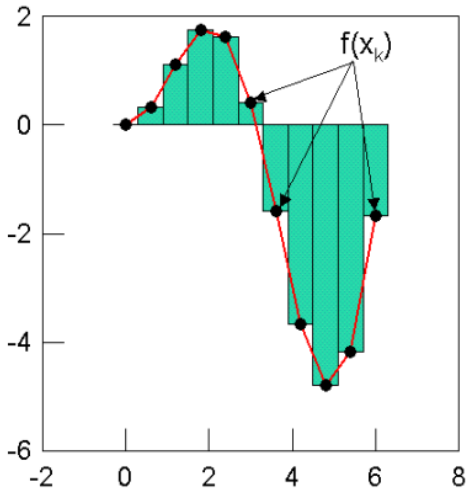
Uwagi:

- Jeżeli $m = n$, wówczas funkcja aproksymująca pokrywa się z wielomianem interpolującym.
- Stopień wielomianu aproksymującego powinien być znacznie mniejszy od liczby węzłów x_i, aby "wygładzić"ewentualne błędy pomiarowe.
- Dla $m \geq 6$ macierz układu staje się źle uwarunkowana (pojedyncza precyzja), najprostszym remedium jest zastosowanie silniejszej arytmetyki (podwójna precyzja).

2.8 Całkowanie numeryczne

Całkowanie numeryczne oznacza zastosowanie metod numerycznych w celu wyznaczenia przybliżonej wartości całki oznaczonej

$$C = \int_a^b f(x)\,dx$$



Skoro funkcję podcałkową możemy interpolować, to wielomian interpolacyjny można wykorzystać do całkowania.

Dla danego ciągu wartości funkcji podcałkowej $f(x_0), f(x_1), \ldots, f(x_N)$ definiujemy wielomian interpolacyjny Lagrange'a:

$$\varphi(x) = L_N(x) = \sum_{k=0}^{N} f(x_k)\Phi_k(x)$$

$$\Phi_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{N} \frac{x - x_j}{x_k - x_j}$$

Podstawiamy wielomian interpolacyjny w miejsce funkcji podcałkowej:

$$\int_a^b f(x)\,dx \approx \int_a^b L_N(x)\,dx = \sum_{k=0}^{N} A_k f(x_k) \iff A_k = \int_a^b \Phi_k(x)\,dx$$

Powyższe wzory definiują tzw. kwadraturę, A_k są współczynnikami kwadratur.

Jeśli spełniony jest warunek

$$|f(x) - L_N(x)| < \epsilon, \quad x \in [a, b]$$

wówczas zachodzi:

$$\left| \int_a^b f(x)\,dx - \sum_{k=0}^{N} A_k f(x_k) \right| = \left| \int_a^b (f(x) - L_N(x))\,dx \right| \leq \epsilon(b - a)$$

Dokładność wyznaczonej wartości całki jest ograniczona dokładnością przybliżenia funkcji podcałkowej wielomianem (lub inną funkcją).

Jeśli funkcja podcałkowa posiada osobliwości (np. jest nieograniczona, lub przedział całkowania jest nieskończony), wówczas powyższy schemat całkowania ulega modyfikacji.

Funkcję podcałkową zastępujemy iloczynem funkcji wagowej $p(x)$ i nowej gładkiej funkcji:

$$F(x) = p(x)f(x)$$

Funkcja wagowa $p(x)$ zawiera wszystkie osobliwości funkcji $F(x)$ lub jej dobór wynika z zastosowanych wielomianów ortogonalnych:

$$\int_a^b F(x)\,dx = \int_a^b p(x)f(x)\,dx \approx \int_a^b p(x)\varphi(x)\,dx = \sum_{k=0}^{N} A_k' f(x_k)$$

$$A_k^{'} = \int_a^b p(x)\Phi(x)$$

Postać funkcji wagowej określa typ kwadratury. Chcemy wyznaczyć wartość całki:

$$I(f) = \int_a^b p(x)f(x)\,dx$$

stosując wzór:

$$S(f) = \sum_{k=0}^{N} A_k f(x_k), \quad x \in [a,b]$$

Powyższy wzór nosi nazwę kwadratury, a punkty $x_1, x_2, \ldots, x_N$ węzłami kwadratury.
Błąd przybliżenia całki kwadraturą (błąd metody):

$$E(f) = I(f) - S(f)$$

Kryterium dokładności kwadratury można przyjąć zgodność $I(W) = S(W)$.
Gdy W jest wielomianem, wówczas mówimy, że dana kwadratura jest rzędu r ($r \geq 1$), jeśli

$$I(W) = S(W)$$

dla wszystkich wielomianów stopnia mniejszego niż r.

Rząd kwadratury w dużym stopniu decyduje o dokładności całkowania numerycznego (drugi czynnik to liczba użytych węzłów).

2.8.1 Kwadratury Newtona-Cotesa

Rozważamy przypadek z węzłami równoodległymi $x = a + ih$, $i = 0, 1, 2, \ldots, N$. Jeśli końce przedziału są również węzłami, wówczas kwadratury noszą nazwę kwadratur zamkniętych.

Przybliżamy funkcję podcałkową wielomianem Lagrange'a stopnia co najwyżej N:

$$f(x) \approx L_N(x) = \sum_{k=0}^{N} f(x_k)\Phi_k(x)$$

$$\Phi_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{N} \frac{x - x_j}{x_k - x_j}$$

Błąd przybliżenia (interpolacji):

$$R_{N+1}(x) = f(x) - L_N(x) = \frac{1}{(N+1)!}\omega_{N+1}(x)f^{(N+1)}(\xi), \quad \xi \in (a,b)$$

Szukamy współczynników A kwadratury. Musimy wykonać całkowanie wielomianu, ułatwimy sobie zadanie, jeśli wprowadzimy nową zmienną t:

$$x = a + ht \implies \frac{x - x_j}{x_k - x_j} = \frac{a + ht - a - hj}{a + hk - a - hj} = \frac{t-j}{k-j} \implies \Phi_k(t) = \prod_{\substack{j=0 \\ j\neq k}}^{N} \frac{t-j}{k-j}$$

Przyjmujemy oznaczenia:

$$h = \frac{b-a}{N}, \quad f_k = f(a + kh)$$

$$\int_a^b f(x)\,dx = \int_a^b L_N(x)\,dx = \sum_{k=0}^{N} f_k \int_a^b \Phi_k(x)\,dx = \sum_{k=0}^{N} f_k h \int_0^N \Phi_k(t)\,dt = \sum_{k=0}^{N} f_k A_k$$

Skąd otrzymujemy:

$$S(f) = \sum_{k=0}^{N} A_k f_k$$

Współczynniki kwadratury Newtona-Cotesa:

$$A_k = h\frac{(-1)^{N-k}}{k!(N-k)!}\int_0^N \frac{t(t-1)\dots(t-N)}{t-k}\,dt$$

Wyprowadzenie wzoru określającego współczynniki kwadratury:

$$\Phi_k(t) = \prod_{\substack{j=0\\ j\neq k}}^{N}\frac{t-j}{k-j} = \frac{(t-0)(t-1)\dots(t-(k-1))(t-(k+1))\dots(t-N)}{(k-0)(k-1)\dots(k-(k-1))(k-(k+1))\dots(k-N)}$$

$$= \frac{(t-0)(t-1)\dots(t-(k-1))(t-(k+1))\dots(t-N)}{(1\cdot 2\cdot \dots\cdot k)\cdot(-1)\cdot(-2)\cdot\dots\cdot(-(N-k))}$$

$$= \frac{(t-0)(t-1)\dots(t-(k-1))(t-(k+1))\dots(t-N)}{k!(-1)^{N-k}(N-k)!}$$

$$= \frac{(-1)^{N-k}}{k!(N-k)!}\frac{t(t-1)\dots(t-N)}{t-k}$$

$$A_k = h\int_0^N \Phi_k(t)\,dt = h\frac{(-1)^{N-k}}{k!(N-k)!}\int_0^N \frac{t(t-1)\dots(t-N)}{t-k}\,dt$$

Własności kwadratur NC

- Gdy N jest nieparzyste, wówczas kwadratura jest rzędu $(N+1)$ (dokładna dla wielomianów stopnia N), dla parzystego N rząd kwadratury wynosi $(N+2)$.
- Jeżeli funkcja podcałkowa jest r-krotnie różniczkowalna, wówczas błąd metody można przedstawić w postaci:

$$E(f) = C_r f^{(r)}(\xi), \quad \xi \in [a,b]$$

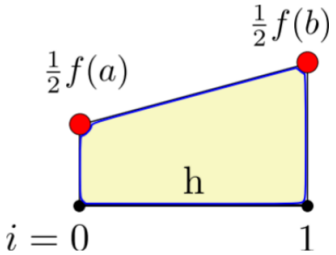
współczynnik C_r nie zależy od f.

-Dla dużych N oszacowanie błędu jest trudne ze względu na pochodne wysokich rzędów lub ze względu na numeryczne kasowanie się współczynników A_k.
-Współczynniki A_k zależą od N. W szczególności zachodzi:

$$\lim_{k\to\infty}|A_k| = \infty$$

dlatego metoda kwadratur Newtona-Cotesa nie jest zbieżna w klasie funkcji ciągłych.
-W praktyce przedział całkowania dzieli się na m podprzedziałów, w każdym podprzedziale określa się N ($N = 1, 2, 3$) i przeprowadza całkowanie, taka procedura prowadzi do uzyskania kwadratur złożonych.

2.8.2 Wzór trapezów (N=1)



$$h = b - a$$

$$A_k = h\cdot\frac{(-1)^{N-k}}{k!(N-k)!}\int_0^N t(t-1)\dots(t-N)(t-k)\,dt$$

Dla $N = 1$:

$$\left.\begin{array}{l} A_0 = -h \cdot \int_0^1 (t-1)\,dt = \frac{1}{2}h \\ A_1 = h \cdot \int_0^1 t\,dt = \frac{1}{2}h \end{array}\right\} \quad \Rightarrow \quad S(f) = \frac{1}{2}h(f_0 + f_1)$$

Z wzoru na błąd interpolacji wynika, że kwadratura jest $N+1 = 2$ rzędu, a dokładnie przybliża wielomian $N = 1$ stopnia.

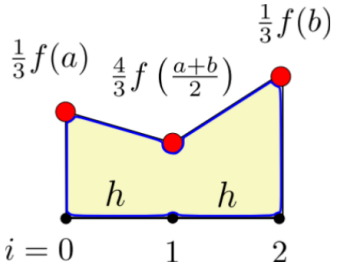
Błąd wyznaczenia przybliżonej wartości całki wynosi:

$$E(f) = \frac{1}{2!}\int_a^b (x-a)(x-b)f''(\xi)\,dx = -\frac{1}{12}h^3 f''(\xi), \quad \xi \in [a,b]$$

Zależność od h:

Na drugą pochodną nie mamy wpływu.

2.8.3 Wzór parabol/Simpsona (N=2)



$$A_k = h \cdot \frac{(-1)^{N-k}}{k!(N-k)!}\int_0^N t(t-1)\ldots(t-N)(t-k)\,dt$$

Dla $N = 2$:

$$\left.\begin{array}{ll} A_0 & = \frac{1}{2}h \\ A_1 & = \frac{4}{2}h \\ A_2 & = \frac{1}{2}h \end{array}\right\} \quad \Rightarrow \quad S(f) = \frac{1}{3}h(f_0 + 4f_1 + f_2)$$

Ponieważ N jest parzyste, więc kwadratura jest dokładna dla wielomianów stopnia $N+1$ i jest rzędu $N+2$. Dlaczego? Zgodnie z wzorem na błąd wzoru interpolacyjnego dostajemy:

$$E(f) \sim \int_a^b (x-a)(x-\frac{a+b}{2})(x-b)\,dx = 0$$

z powodu nieparzystości funkcji podcałkowej, ale nie ma powodu, aby błąd znikał dla dowolnej funkcji.

Dodajmy więc dodatkowy węzeł w $x = \frac{a+b}{2}$, który nie zmienia warunku interpolacji. Wówczas stopień wielomianu czynnikowego rośnie o 1:

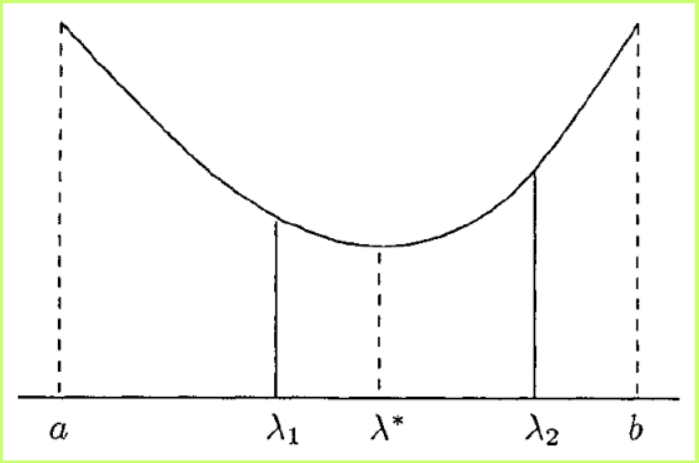
$$E(f) = \frac{f^{(4)}(\xi_1)}{4!}\int_a^b (x-a)(x-\frac{a+b}{2})^2(x-b)\,dx = -\frac{1}{90}h^5 f^{(4)}(\xi), \quad \xi \in [a,b]$$

(funkcja podcałkowa teraz jest parzysta).

2.9 Minimalizacja wartości funkcji

2.9.1 Metoda złotego podziału (metoda jednowymiarowa, niegradientowa)

1. Wstępnie wyznaczamy przedział $[a,b]$, w którym spodziewamy się minimum wartości funkcji.
2. W przedziale $[a,b]$ wyznaczamy dwa punkty λ_1, λ_2.
3. Jeśli $F(\lambda_2) > F(\lambda_1)$, to zmieniamy granice przedziału na $[a, \lambda_2]$.
4. Jeśli $F(\lambda_2) < F(\lambda_1)$, to zmieniamy granice przedziału na $[\lambda_1, b]$.
5. Proces podziału prowadzimy iteracyjnie, aż do spełnienia warunku $|a^i - b^i| < \epsilon$. Jako przybliżenie minimum możemy przyjąć $\lambda^* = \frac{a^i+b^i}{2}$.



Rys. Wyznaczanie kolejnych przybliżeń w metodzie złotego podziału

Pozostaje tylko kwestia, jak wyznaczyć punkty tak, aby wybór był optymalny, tzn. chcemy wykonać jak najmniejszą ilość podziałów. Punktem wyjścia jest zależność (złota proporcja/podział):

$$\frac{(\lambda_1 - a) + (b - \lambda_1)}{b - \lambda_1} = \frac{b - \lambda_1}{\lambda_1 - a} = \varphi$$

$$\lambda_1 = a + r^2 \cdot L$$

$$\lambda_2 = a + rL$$

Uzależniamy, b od a

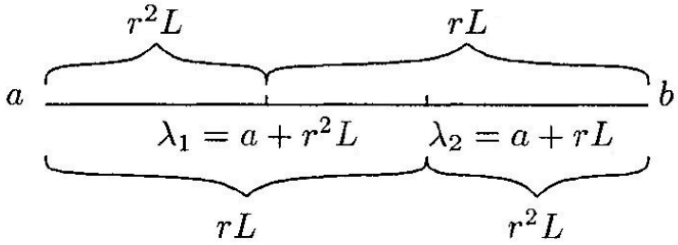
$$(b - a) = L \Rightarrow b = L + a$$

Po wstawieniu do równania otrzymujemy:

$$\frac{L}{L + a - \lambda_1} = \frac{L + a - \lambda_1}{\lambda_1 - a}$$

$$L(\lambda_1 - a) = (L - (\lambda_1 - a))^2$$

$$\lambda_1 - a = L \underbrace{\left(1 - \frac{\lambda_1 - a}{L}\right)^2}_{=r^2} = L \cdot r^2$$



Na podstawie rysunku możemy zapisać drugą relację.

$$b - \lambda_1 = L - (\lambda_1 - a)$$

$$b - \lambda_1 = L \underbrace{\left(1 - \frac{\lambda_1 - a}{L}\right)}_{=r} = Lr$$

Otrzymaliśmy dwie zależności:

$$\lambda_1 - a = L \cdot r^2, \quad b - \lambda_1 = Lr$$

Po wstawieniu ich do równania wyjściowego dostajemy równanie kwadratowe na r:

$$\frac{L \cdot r^2 + Lr}{Lr} = \frac{Lr}{L \cdot r^2} = \frac{1}{r} \Rightarrow r^2 + r - 1 = 0$$

I znajdujemy jego pierwiastki:

$$r_1 = \frac{\sqrt{5} - 1}{2} \approx 0.618034 \quad (> 0)$$

$$r_2 = \frac{-\sqrt{5} - 1}{2} \quad (< 0)$$

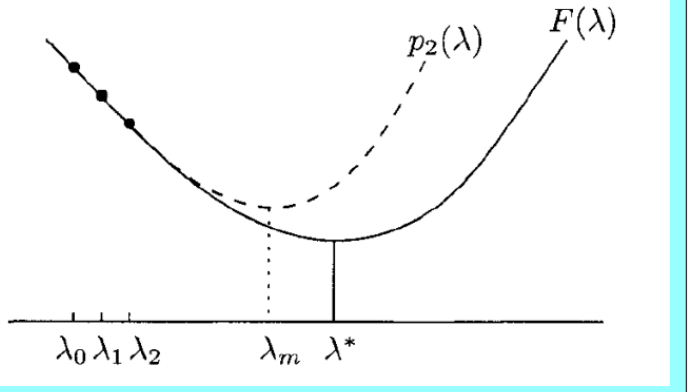
(Dodatni zachowujemy, ujemny - odrzucamy).

Po wyborze $r = r_1$, możemy określić wartości λ_1 i λ_2, zakładając ponadto, że oba punkty powinny być symetryczne względem krańców przedziału:

$$\lambda_1 = a + r^2 \cdot L$$

$$\lambda_2 = a + rL$$

2.9.2 Metoda interpolacji kwadratowej Powella



Rys. Wyznaczanie przybliżonego rozwiązania w metodzie Powell'a.

Przez trzy punkty: λ_1, λ_2, λ_3 prowadzimy wielomian 2 stopnia

$$p_2(\lambda) = F(\lambda_0) + F[\lambda_0, \lambda_1](\lambda - \lambda_0) + F[\lambda_0, \lambda_1, \lambda_2](\lambda - \lambda_0)(\lambda - \lambda_1)$$

gdzie:

$F(\lambda_n)$ - wartość funkcji
$F[\lambda_0, \lambda_1]$ - iloraz różnicowy 1 rzędu
$F[\lambda_0, \lambda_1, \lambda_2]$ - iloraz różnicowy 2 rzędu

Narzucamy warunek zerowania się pochodnej (spodziewamy się minimum)

$$\frac{dp_2}{d\lambda} = F[\lambda_0, \lambda_1] + 2\lambda F[\lambda_0, \lambda_1, \lambda_2] - F[\lambda_0, \lambda_1, \lambda_2](\lambda_0 + \lambda_1) = 0$$

Rozwiązując to równanie ze względu na λ otrzymamy:

$$\lambda_m = \frac{F[\lambda_0, \lambda_1, \lambda_2](\lambda_0 + \lambda_1) - F[\lambda_0, \lambda_1]}{2F[\lambda_0, \lambda_1, \lambda_2]} \approx \lambda^*$$

Aby znaleziony punkt był rzeczywistym minimum, iloraz $F[\lambda_0, \lambda_1, \lambda_2]$ musi spełniać warunek:

$$F[\lambda_0, \lambda_1, \lambda_2] > 0$$

Algorytm interpolacji Powella

1. Wybierz λ_0, oblicz:
$$F[\lambda_0 + h] < F[\lambda_0], \quad F[\lambda_0 + 2h] < F[\lambda_0 + h]$$
(ewentualnie zmień znak: $-h$, jeśli nierówności nie są spełnione)
2. Wyznacz λ_m, sprawdź czy jest minimum.
3. Jeśli $|\lambda_m - \lambda_n| > h$, odrzuć najdalej położony od λ_m punkt i ponownie wykonaj obliczenia z pkt. 2.

λ_n- najbliżej położony punkt względem λ_m.
Punkt λ_m akceptujemy jako minimum jeśli $|\lambda_m - \lambda_n| < \epsilon$.

2.9.3 Metoda Newtona dla funkcji kwadratowej w $\mathbb{R}^n$

Funkcję kwadratową definiujemy następująco:

$$f(x) = \frac{1}{2}x^T Ax + x^T b + c$$

gdzie: A jest pewną macierzą kwadratową oraz $x, b \in \mathbb{R}^n$ i $c \in \mathbb{R}$.
Jeśli macierz A jest symetryczna, to wówczas zachodzi:

$$\nabla f(x) = Ax + b$$

oraz

$$\nabla^2 f(x) = A \iff H(x) = A \quad \text{(hesjan)}$$

Jeśli A jest dodatniookreślona, to rozwiązanie można łatwo znaleźć, ponieważ:

$$\nabla f(x) = Ax + b = 0$$

$$x^* = -A^{-1}b$$

(macierz dodatniookreślona jest nieosobliwa i można ją odwrócić).
W metodzie Newtona zakładamy (x^* to rozwiązanie dokładne):

$$x^* = x^i + \delta$$

gdzie x to przybliżone rozwiązanie w i-tej iteracji.
Korzystając z rozwinięcia funkcji w szereg Taylora, możemy zapisać:

$$0 = \nabla f(x^*) = \nabla f(x^i + \delta) = \nabla f(x^i) + H(x^i)\delta + O(\|\delta\|^2)$$

Jeśli pominiemy wyrazy rzędu $\|\delta\|^2$, to:

$$\nabla f(x^i) + H(x^i)\delta = 0$$

W i-tej iteracji poprawiamy rozwiązanie, tj.

$$x^{(i+1)} = x^i + \delta$$

i ostatecznie:

$$x^{i+1} = x^i - H^{-1}(x^i)\nabla f(x^i)$$

Oczekujemy, że metoda Newtona będzie pracować również dla innych funkcji niż kwadratowe, tj. gdy badaną funkcję celu można lokalnie przybliżyć funkcją kwadratową.

Wadą metody jest konieczność wyznaczania hesjanu w każdym punkcie. Gdy hesjan staje się osobliwy, wówczas metoda przestaje działać, co może być spowodowane np. występowaniem błędów numerycznych.

2.10 Szybka transformacja Fouriera (FFT)

2.10.1 Algorytm radix-2

Najprostszy algorytm FFT to radix-2 (Cooley-Tukey) opracowany w latach 60 XX wieku w celu szybkiej analizy danych sejsmologicznych. Naszym zadaniem jest obliczenie współczynników transformaty Fouriera (DFT) c_k, ale wykonując jak najmniej obliczeń. Zakładamy, że całkowita liczba węzłów jest potęgą 2:

$$x_j = \frac{2\pi}{N} \cdot j, \quad j = 0, 1, 2, \ldots, N-1$$

$$N = 2^r, \quad r \in \mathbb{N}$$

$$c_k = \langle E_k, f \rangle = \sum_{j=0}^{N-1} E_k^*(x_j) f(x_j)$$

$$= \sum_{j=0}^{N-1} f(x_j) \cdot \exp(-I \cdot x_j \cdot k)$$

$$= \sum_{j=0}^{N-1} f_j \exp\left(-i\frac{2\pi}{N} jk\right)$$

Osobno grupujemy składniki parzyste $j = 2m$ i nieparzyste $j = 2m+1$:

$$c_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m} \exp\left(-I\frac{2\pi}{N}(2m)k\right) + \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \exp\left(-I\frac{2\pi}{N}(2m+1)k\right)$$

$$c_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m} \exp\left(-I\frac{2\pi}{N/2}mk\right) + \exp\left(-I\frac{2\pi}{N}k\right) \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \exp\left(-I\frac{2\pi}{N/2}mk\right)$$

$$\left.\begin{array}{l} p_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m} \exp\left(-I\frac{2\pi}{N/2}mk\right) \\ q_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \exp\left(-I\frac{2\pi}{N/2}mk\right) \\ \varphi_k = \exp\left(-I\frac{2\pi}{N}k\right) \end{array}\right\} \quad \Rightarrow \quad c_k = p_k + \varphi_k q_k$$

Korzystamy teraz z okresowości wyrazów p oraz q:

$$p_{k+\frac{N}{2}} = p_k$$
$$q_{k+\frac{N}{2}} = q_k$$

(nie musimy wyznaczać wszystkich współczynników - tylko połowę). Natomiast czynnik fazowy ma następującą własność:

$$\varphi_{k+\frac{N}{2}} = \exp\left(-I\frac{2\pi}{N}(k+\frac{N}{2})\right) = \exp\left(-I\frac{2\pi}{N}k\right) \cdot \exp\left(-I\frac{2\pi}{N} \cdot \frac{N}{2}\right) = -\exp\left(-I\frac{2\pi}{N}k\right) = -\varphi_k$$

Uwagi: a) Współczynniki p_k oraz q_k można wyliczyć dzięki DFT nakładem $O(\frac{N}{2})^2 = O(\frac{N^2}{4})$. b) Dodatkowo oszczędzamy czas wyznaczając tylko współczynniki dla $k < \frac{N}{2}$ ponieważ:

$$c_k = \begin{cases} p_k + \varphi_k q_k & \text{dla} \quad k < \frac{N}{2} \\ p_{k-\frac{N}{2}} - \varphi_k q_{k-\frac{N}{2}} & \text{dla} \quad k \geq \frac{N}{2} \end{cases}$$

c) Kolejnym krokiem w FFT jest podział sum w p_k oraz w q_k na sumy zawierające tylko elementy parzyste i nieparzyste. d) Po podziale liczba elementów w każdej z dwóch powstałych sum jest dwukrotnie mniejsza niż w elemencie macierzystym. e) Proces rekurencyjnego podziału kończymy, gdy liczba elementów jest równa 1.

2.10.2 Szybkie mnożenie wielomianów przy użyciu FFT

Chcemy obliczyć iloczyn dwóch wielomianów:

$$P(x) = \sum_{i=0}^{n-1} a_i x^i$$

$$Q(x) = \sum_{i=0}^{n-1} b_i x^i$$

Jeśli stopnie wielomianów są różne, to je wyrównujemy dodając do wielomianu niższego stopnia współczynniki równe 0. Iloczyn wielomianów:

$$R(x) = P(x)Q(x) = \sum_{i=0}^{n-1} a_i x^i \sum_{j=0}^{n-1} b_j x^j = \sum_{i,j=0}^{n-1} a_i b_j x^{i+j}$$

Dokonujemy reindeksacji wskaźników:

$$i + j = k \implies j = k - i$$

Po reindeksacji dostajemy:

$$c_k = \sum_{i=0}^{n-1} a_i b_{k-i}$$

$$R(x) = \sum_{k=0}^{2n-2} \underbrace{\left(\sum_{i=0}^{n-1} a_i b_{k-i} \right)}_{c_k} x^k = \sum_{k=0}^{2n-1} c_k x^k$$

$$c_k = \sum_{i=0}^{n-1} a_i b_{k-i}$$

$$c_{2n-1} = 0$$

Jeśli współczynniki wielomianów a_i oraz b_i potraktujemy jako współrzędne wektorów:

$$\mathbf{a} = [a_0, a_1, \ldots, a_{n-1}]$$

$$\mathbf{b} = [b_0, b_1, \ldots, b_{n-1}]$$

to wektor $\mathbf{c}$ jest ich splotem:

$$\mathbf{c} = \mathbf{a} * \mathbf{b}$$

Korzystając z definicji transformacji Fouriera dla splotu funkcji, możemy zapisać:

$$\mathbf{c} = FFT^{-1}\left[\mathrm{FFT}(\tilde{\mathbf{a}}) \cdot \mathrm{FFT}(\tilde{\mathbf{b}})\right]$$

$$\tilde{\mathbf{a}} = [a_0, a_1, \ldots, a_{n-1}, a_n, \ldots, a_{2n-1}]$$

$$\tilde{\mathbf{b}} = [b_0, b_1, \ldots, b_{n-1}, b_n, \ldots, b_{2n-1}]$$

Wektory $\mathbf{a}$ i $\mathbf{b}$ powiększamy, dodatkowe elementy zerujemy (bo ich nie ma):

$$a_i, b_i = 0 \quad \iff \quad i > n - 1$$

2.11 Generatory liczb pseudolosowych

2.11.1 Generatory liniowe

Generatory liniowe tworzą ciąg liczb według schematu:

$$X_{n+1} = (a_1 X_n + a_2 X_{n-1} + \ldots + a_k X_{n-k+1} + c) \mod m$$

gdzie: $a_1, a_2, \ldots, a_k, c, m$ są parametrami generatora (ustalone liczby).

Operację

$$r = (a \mod n) \quad a, n, r \in \mathbb{Z}$$

nazywamy dzieleniem modulo, a jej wynikiem jest reszta z dzielenia liczb całkowitych a przez n.

Lub inaczej: r jest kongruentne do a modulo n jeśli n jest dzielnikiem $a - r$.

$$a \equiv r \mod n \Rightarrow r = a - \left\lfloor \frac{a}{n} \right\rfloor n$$

Generatory wykorzystujące operację dzielenia modulo to generatory kongruentne lub kongruencyjne.

Przykład:

$$\begin{aligned} 19 &\mod 6 = 1 \\ 18 &\mod 6 = 0 \\ 17 &\mod 6 = 5 \\ 16 &\mod 6 = 4 \\ 15 &\mod 6 = 3 \\ 14 &\mod 6 = 2 \\ 13 &\mod 6 = 1 \\ 12 &\mod 6 = 0 \end{aligned}$$

Aby wygenerować ciąg liczb pseudolosowych należy zdefiniować jego parametry. Liczby $X_0, X_1, X_2, \ldots$ nazywamy ziarnem generatora (seed).

Dla bardziej rozbudowanych generatorów liczby te otrzymujemy z innego generatora lub np. używając zegara systemowego (X_0).

Najprostszy generator liniowy ma dwie odmiany:

- Generator multiplikatywny ($c = 0$)
- Generator mieszany ($c \neq 0$)

Maksymalny okres generatora liniowego to $(m - 1)$.

Najprostszy generator multiplikatywny

$$X_{i+1} = aX_{i-1} \mod m \implies k_i = \left\lfloor \frac{aX_{i-1}}{m} \right\rfloor \quad \text{dla i} \geq 1 \implies$$

$$\implies \begin{cases} X_1 = a \cdot X_0 - m \cdot k_1 \\ X_2 = a^2 \cdot X_0 - m \cdot k_2 - m \cdot k_1 \cdot a \\ \ldots \\ X_3 = a^3 \cdot X_0 - m \cdot k_3 - m \cdot k_2 \cdot a - m \cdot k_1 \cdot a^2 \end{cases}$$

Ostatnie równanie można zapisać w postaci:

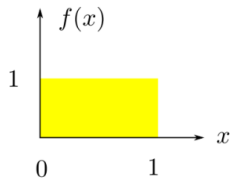
$$X_n = a^n \cdot X_0 \mod m$$

Wybór X_0 determinuje wszystkie liczby w generowanym ciągu (a i m są ustalone) działa w sposób całkowicie deterministyczny - całkowity brak losowości.

Taki generator jest NIEPRZYDATNY.

Podstawowe parametry statystyczne generatora o rozkładzie równomiernym $U(0,1)$

Zakładamy, że generator dostarcza liczb losowych o rozkładzie jednorodnym w zakresie $x \in (0,1)$, rozkład definiuje funkcja gęstości prawdopodobieństwa $f(x)$.



$$f(x) = 1, \quad x \in (0,1)$$

Jeśli generowany ciąg liczb jest niezależny, to wartość oczekiwana zmiennej losowej powinna wynosić:

$$\mu = \int_0^1 \underbrace{f(x)}_{=1} x\, dx = \left.\frac{x^2}{2}\right|_0^1 = \frac{1}{2}$$

Jej estymatorem jest średnia arytmetyczna:

$$\overline{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Wariancja rozkładu zdefiniowana jest jako drugi moment centralny:

$$\sigma^2 = \int_0^1 (x-\mu)^2\, dx = \frac{1}{12}$$

$$\overline{\sigma} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{\mu})^2$$

Jeśli parametry statystyczne generatora (ciągu generowanych przez niego liczb) odbiegają od powyższych wartości, to jest on nieprzydatny (lub warunkowo przydatny).

Ponadto współczynniki autokorelacji elementów ciągu powinny wynosić 0.

Funkcja autokorelacji

Opisuje zależność elementów ciągu od wyrazów poprzednich.

Definicja:

$$R_r = \frac{\mathbb{E}[(X_t - \mu)(X_s - \mu)]}{\sigma^2}$$

oraz wzór dla ciągu skończonego (r = s - t to przesunięcie elementów):

$$\overline{R_r} = \frac{1}{(N-1)\sigma^2}\sum_{i=1}^{N-r}(X_i - \mu)(X_{i+r} - \mu)$$

Inaczej: opisuje związek pomiędzy elementami dwóch szeregów - danego i przesuniętego o r.

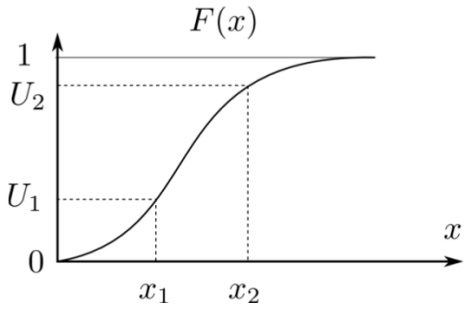
Dla rozkładu jednorodnego $R \sim 0$ oznacza brak korelacji pomiędzy elementami ciągu, czyli rozkład liczb pseudolosowych jest stochastyczny (brak korelacji).

W praktyce najsilniejsza jest korelacja między kilkoma kolejnymi liczbami pseudolosowymi, wyznacza się funkcję autokorelacji dla $r = 1, 2, 3, 4, 5, 6$.

2.11.2 Metoda odwracania dystrybuanty

Rozkład prawdopodobieństwa w sposób jednoznaczny określają dwie funkcje: dystrybuanta rozkładu i funkcja gęstości prawdopodobieństwa.

Dystrybuanta jest funkcją niemalejącą i prawostronnie ciągłą $F : \mathbb{R} \to \mathbb{R}$



$$\lim_{x\to-\infty} F(x) = 0 \quad \text{oraz} \quad \lim_{x\to\infty} F(x) = 1$$

Funkcja gęstości prawdopodobieństwa jest nieujemna i unormowana:

$$f(x) \geq 0 \quad \text{oraz} \quad \int_{-\infty}^{\infty} f(x)\,dx = 1$$

Dystrybuanta i fgp rozkładu są ze sobą ściśle związane (są miarą prawdopodobieństwa):

$$F(x) = \int_{-\infty}^{x} f(y)\,dy$$

Znajdźmy funkcję odwrotną dystrybuanty. Co wówczas uzyskamy?

$$U = F(x) \quad \Rightarrow \quad x = F^{-1}(U), \quad U \in [0,1], \quad x \in (-\infty, \infty)$$

Wstawiając jako argument do funkcji odwrotnej liczbę losową o rozkładzie jednorodnym $U(0,1)$, dokonujemy jej transformacji, uzyskując liczbę losową o rozkładzie zdefiniowanym przez dystrybuantę.

Przykład: metoda odwracania dystrybuanty - rozkład jednomianowy

Funkcja gęstości prawdopodobieństwa:

$$f(x) = (n+1)x^n, \quad x \in [0,1], \quad n = 1,2,3,\ldots$$

Dystrybuanta:

$$F(x) = (n+1)\int_0^x (x')^n\,dx' = (n+1)\frac{x^{n+1}}{n+1} = U$$

Generator o rozkładzie jednomianowym:

$$x = U^{\frac{1}{n+1}}, \quad x \in (0,1)$$

Przykład: metoda odwracania dystrybuanty - rozkład eksponencjalny

Funkcja gęstości prawdopodobieństwa rozkładu eksponencjalnego:

$$f(x) = e^{-x}, \quad x \in [0,\infty)$$

Dystrybuanta:

$$F(x) = \int_0^x e^{-x'}\,dx' = 1 - e^{-x} = U, \quad U \in (0,1)$$

$$e^{-x} = 1 - U \quad \Rightarrow \quad F^{-1}(x) = x = -\ln(1-U)$$

Generator o rozkładzie eksponencjalnym:

$$x = -\ln(1-U)$$

Przykład: metoda odwracania dystrybuanty - rozkład normalny $N(0,1)$

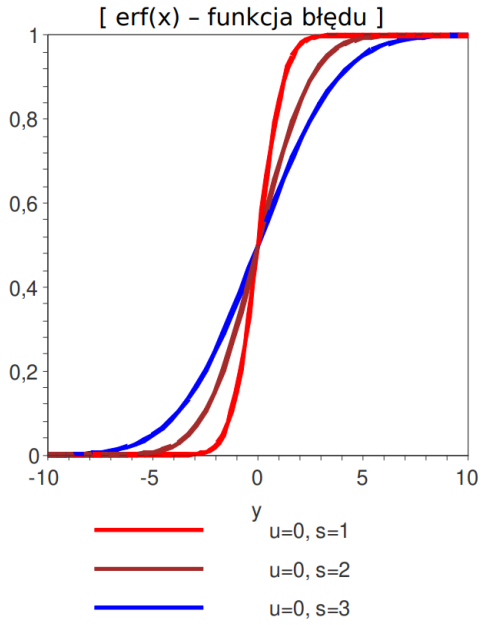
Funkcja gęstości prawdopodobieństwa to funkcja Gaussa:

$$f(x) = e^{-x^2}$$

Dystrybuanta jest funkcją błędu:

$$F(x) = \int_{-\infty}^{x} e^{-x'^2}\, dx' = \mathrm{erf}(x)$$

Szukanie funkcji odwrotnej $\mathrm{erf}(x)$ jest kosztowne, dlatego stosuje się metodę Boxa-Mullera.



Metoda Boxa-Mullera dla rozkładu normalnego

Definiujemy funkcję gęstości prawdopodobieństwa w 2D jako złożenie dwóch funkcji Gaussa:

$$f(x,y) = f(x)f(y) = e^{-\frac{x^2+y^2}{2}}, \quad x,y \in (-\infty,\infty)$$

Docelowo chcemy policzyć prawdopodobieństwo:

$$p(x,y) = f(x,y)\, dx\, dy$$

(czyli że wylosowana liczba znajdzie się w obszarze dxdy)
Przeprowadzamy transformację do współrzędnych biegunowych:

$$\begin{cases} x = r\cos(\theta), & r \in [0,\infty) \\ y = r\sin(\theta), & \theta \in [0,2\pi] \end{cases} \quad \Rightarrow \quad r^2 = x^2 + y^2$$

Prawdopodobieństwo określmy przy użyciu nowych zmienych – stosujemy prawo przenoszenia prawdopodobieństwa:

$$p = f(x,y)\, dx\, dy = f(r,\theta) r\, dr\, d\theta$$

A dodając separacje zmiennych:

$$p(r,\theta) = r \cdot e^{-\frac{r^2}{2}}\, dr\, d\theta$$

Wprowadzamy nową zmienną:

$$z = \frac{r^2}{2} \quad \Rightarrow \quad dz = r\, dr, \quad z \in [0,\infty)$$

$$p(z,\theta) = e^{-z}\, dz\, d\theta = f(z) = \, dz \cdot 1 \cdot d\theta$$

Uzyskaliśmy rozkład wykładniczy:

$$f(z) = e^{-z} \implies z = -\ln(1 - U_1), \quad U_1 \in (0, 1)$$

Następnie:

$$r = \sqrt{2z} = \sqrt{-2\ln(1 - U_1)}$$

Kąt θ ma rozkład jednorodny:

$$\theta = U_2 \cdot 2\pi, \quad U_2 \in (0, 1)$$

Dla pary (U_1, U_2) dostajemy parę liczb losowych (x, y) z rozkładu $N(0, 1)$:

$$X = r\cos(\theta) = \sqrt{-2\ln(1 - U_1)}\cos(2\pi U_2)$$

$$Y = r\sin(\theta) = \sqrt{-2\ln(1 - U_1)}\sin(2\pi U_2)$$

Jeśli chcemy zmienić parametry rozkładu normalnego, dokonujemy transformacji liniowej

$$X = x\sigma + \mu, \quad x \in N(0, 1) \Rightarrow X \in N(\mu, \sigma)$$

2.11.3 Testy zgodności z zadanym rozkładem - test chi-kwadrat

Badamy w nim hipotezę, że generowana zmienna losowa X ma rozkład prawdopodobieństwa o dystrybuancie F.

Jeżeli

$$F(a) = 0, \quad F(b) = 1$$

to możemy dokonać następującego podziału zbioru wartości zmiennej X:

$$a < a_1 < a_2 < \ldots < a_k = b \implies p_i = P\{a_{i-1} < X \leq a_i\}, \quad i = 1, 2, \ldots$$

Generujemy ciąg n liczb:

$$X_1, X_2, \ldots, X_n$$

Sprawdzamy ile z nich spełnia warunek $a_{i-1} < X \leq a_i$ i ich liczbę oznaczamy n_i. Statystyką testu jest:

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i}$$

Dla dużego n statystyka ta ma rozkład χ^2 o $k - 1$ stopniach swobody.

Uzyskaną wartość porównujemy z wartością graniczną dla rozkładu chi-kwadrat (np. korzystając z tabel statystycznych). Jeśli jest mniejsza od wartości granicznej, to hipotezy nie odrzucamy.

2.12 Całkowanie metodą Monte Carlo

2.12.1 Podstawowa metoda całkowania Monte Carlo

Interesuje nas wyznaczenie (a raczej estymacja) wartości oczekiwanej zmiennej losowej

$$z = z(x)$$

która jest funkcją wektora zmiennych (losowych):

$$x = [x_1, x_2, \ldots, x_m]$$

Rozkład prawdopodobieństwa zmiennej losowej z opisuje (nieznana) fgp $g(z)$

$$\int_{-\infty}^{\infty} g(z(x))\, dz = 1$$

a rozkład prawdopodobieństwa wektora x opisuje znana fgp $f(x)$

$$\int_V f(x)\, dx = 1$$

$$\langle z\rangle = \int_{-\infty}^{\infty} z g(z)\,dz = \int_V z(x) f(x)\,dx$$

Przy takich założeniach, zgodnie z CTG dystrybuanta wartości oczekiwanej ma rozkład normalny (niezależnie od rozkładu zmiennej losowej z):

$$\lim_{N\to\infty} P\left\{\frac{|\overline{z} - \langle z\rangle|}{\frac{\sigma}{\sqrt{N}}} \le \lambda\right\} = \frac{1}{\sqrt{2\pi}} \int_{-\lambda}^{\lambda} e^{-\frac{u^2}{2}}\,dt$$

Metodę Monte Carlo szacowania wartości całek w wersji podstawowej definiują wzory:
a) Wartość całki

$$I = \int_V z(x) f(x)\,dx \approx \frac{1}{N}\sum_{i=1}^{N} z(x_i), \quad x_i \sim f(x)$$

Uwaga: x - jest wektorem, którego składowe są niezależnymi zmiennymi losowymi o określonych funkcjach gęstości prawdopodobieństwa
b) Błąd oszacowania

$$\sigma(I) = \sqrt{\int_V (z - \langle z\rangle)^2 f(x)\,dx} \approx \frac{\sigma(z)}{\sqrt{N}}$$

Gdy obszarem całkowania jest określony nieregularny podzbiór przestrzeni $\mathbb{R}^n$, wówczas obliczaną całkę trzeba zapisać w zmienionej postaci:

$$V \subset \Omega \Longrightarrow I = \int_V z(x) f(x)\,dx = \int_\Omega 1_V(x) z(x) f(x)\,dx$$

gdzie $1_V(x)$ jest funkcją przynależności do zbioru:

$$1_V(x) = \begin{cases} 1 & \Longleftrightarrow\ x \in V \\ 0 & \Longleftrightarrow\ x \notin V \end{cases}$$

Kwadratura Monte Carlo (metoda orzeł-reszka)

$$I = \int_V z(x)\,dx = \int_\Omega 1_V(x) z(x)\,dx = \int_\Omega \underbrace{\Omega \frac{1}{\Omega}}_{f(x)} 1_V(x) z(x)\,dx \approx \frac{\Omega}{N}\sum_{i=1}^{N} 1_V(x) z(x), \quad x \sim f(x)$$

Warunek unormowania $f(x)$

$$\int_\Omega f(x)\,dx = \int_\Omega \frac{1}{\Omega}\,dx = 1$$

Uwagi:
- W powyższym przypadku zakładamy, że fgp jest stała w obszarze Ω.
- Kwadratura Monte Carlo - wzór podobny jak w całkowaniu numerycznym, ale tu położenia węzłów są losowane.

2.12.2 Sposób estymacji wartości oczekiwanej oraz odchylenia standardowego

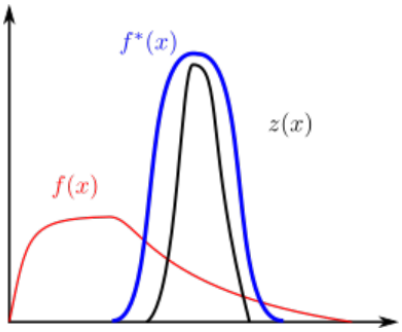
Metoda losowania ważonego

Pierwotna postać całki:

$$I = E(z) = \int_V G(x)\, f(x)\,dx$$

Pod całkę wprowadzamy nową funkcję gęstości prawdopodobieństwa $f^*(x)$:

$$I = E(z) = \int_V z(x)\, f(x)\, \frac{f^*(x)}{f^*(x)}\,dx$$



$$f^*(\vec{x}) \geq 0, \quad \int_V f^*(\vec{x}) d^n x = 1$$

Wprowadzamy nową zmienną losową y:

$$y(x) = z(x) \frac{f(x)}{f^*(x)}$$

Wówczas całkę I możemy obliczyć, losując wektor x z rozkładem $g_x(x)$ i sumując uzyskane wartości y_n:

$$I \approx \frac{1}{N} \sum_{i=1}^{N} y(x_i), \quad x_i \sim f^*(x)$$

Zmienna losowa y ma taką samą wartość oczekiwaną jak zmienna losowa G oraz wariancję zależną od nowej funkcji gęstości prawdopodobieństwa $g_x(x)$.

Wariancję estymatora całki można zmniejszyć, odpowiednio dobierając nową funkcję gęstości prawdopodobieństwa.

Metoda zmiennej kontrolnej

Metoda polega na dekompozycji całki:

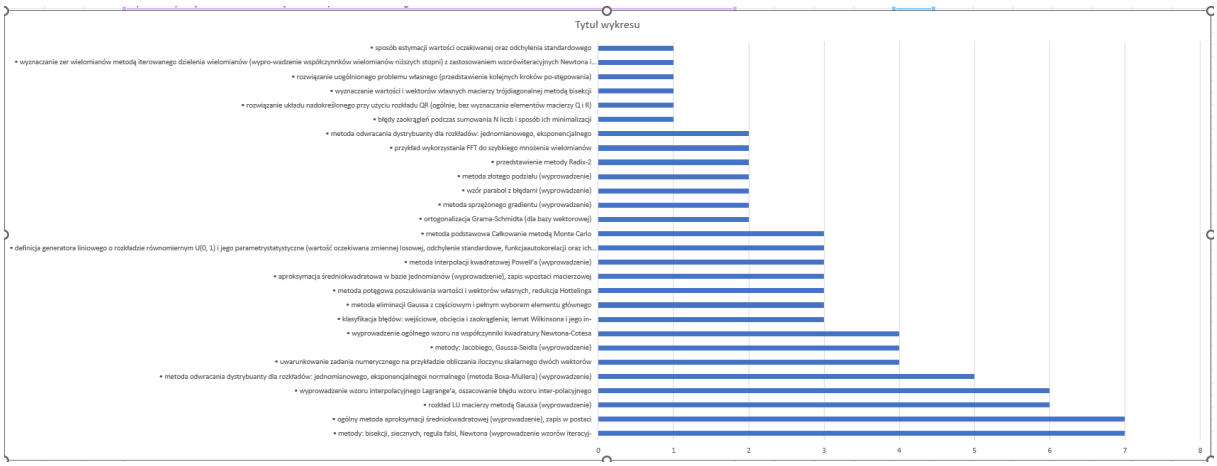
$$I = \int_V \left[G(x) + \underbrace{\hat{G}(x) - \hat{G}(x)}_{=0} \right] f(x)\, dx = \int_V \hat{G}(x)\, f(x)\, dx + \int_V \left[G(x) - \hat{G}(x) \right] f(x)\, dx$$

gdzie $\hat{G}(x)$ jest aproksymacją funkcji $G(x)$, umożliwiającą łatwe obliczenie pierwszego wyrazu po prawej stronie (analitycznie lub numerycznie).

Wariancja zmiennej losowej

$$y = G(x) - \hat{G}(x)$$

ma mniejszą wartość niż pierwotna zmienna (funkcja) $G(x)$.


Tytuł wykresu
• sposób estymacji wartości oczekiwanej oraz odchylenia standardowego
• wyznaczanie zer wielomianów metodą iterowanego dzielenia wielomianów (wypro-wadzenie współczynnków wielomianów niższych stopni) z zastosowaniem wzorówiteracyjnych Newtona i...
• rozwiązanie uogólnionego problemu własnego (przedstawienie kolejnych kroków po-stępowania)
• wyznaczanie wartości i wektorów własnych macierzy trójdiagonalnej metodą bisekcji
• rozwiązanie układu nadokreślonego przy użyciu rozkładu QR (ogólnie, bez wyznaczania elementów macierzy Q i R)
• błędy zaokrągleń podczas sumowania N liczb i sposób ich minimalizacji
• metoda odwracania dystrybuanty dla rozkładów: jednomianowego, eksponencjalnego
• przykład wykorzystania FFT do szybkiego mnożenia wielomianów
• przedstawienie metody Radio-2
• metoda złotego podziału (wyprowadzenie)
• wzór parabol z błędami (wyprowadzenie)
• metoda sprzężonego gradientu (wyprowadzenie)
• ortogonalizacja Grama-Schmidta (dla bazy wektorowej)
• metoda podstawowa Całkowanie metodą Monte Carlo
• definicja generatora liniowego o rozkładzie równomiernym U(0, 1) i jego parametrystatystyczne (wartość oczekiwana zmiennej losowej, odchylenie standardowe, funkcjaautokorelacji oraz ich...
• metoda interpolacji kwadratowej Powell'a (wyprowadzenie)
• aproksymacja średniokwadratowa w bazie jednomianów (wyprowadzenie), zapis wpostaci macierzowej
• metoda potęgowa poszukiwania wartości i wektorów własnych, redukcja Hottellinga
• metoda eliminacji Gaussa z częściowym i pełnym wyborem elementu głównego
• klasyfikacja błędów: wejściowe, obcięcia i zaokrąglenia; lemat Wilkinsona i jego in-
• wyprowadzenie ogólnego wzoru na współczynniki kwadratury Newtona-Cotesa
• metody: Jacobiego, Gaussa-Seidla (wyprowadzenie)
• uwarunkowanie zadania numerycznego na przykładzie obliczania iloczynu skalarnego dwóch wektorów
• metoda odwracania dystrybuanty dla rozkładów: jednomianowego, eksponencjalnegoi normalnego (metoda Boxa-Mullera) (wyprowadzenie)
• wyprowadzenie wzoru interpolacyjnego Lagrange'a, oszacowanie błędu wzoru inter-polacyjnego
• rozkład LU macierzy metodą Gaussa (wyprowadzenie)
• ogólny metoda aproksymacji średniokwadratowej (wyprowadzenie), zapis w postaci
• metody: bisekcji, siecznych, reguła falsi, Newtona (wyprowadzenie wzorów iteracyj-
0
1
2
3
4
5
6
7
8