Algorytmy numeryczne, część 1

dr Marcin Ziółkowski

Instytut Matematyki i Informatyki
Akademia im. Jana Długosza w Częstochowie

9 maja 2017 r.

Metody numeryczne są gałęzią matematyki, która w ogólnym przypadku zajmuje się przybliżonymi metodami rozwiązywania wybranych problemów matematycznych. Ich rola w ostatnim wieku bardzo mocno wzrosła z uwagi na pojawienie się komputerów i innych maszyn liczących, co spowodowało, że metody przybliżone zyskały na znaczeniu, ponieważ z jednej strony w praktycznych zastosowaniach nie zawsze musimy znać dokładny wynik (wystarczy nam przybliżenie tego wyniku z pewną dokładnością), a z drugiej strony maszyny liczące i tak wykonują obliczenia na liczbach przybliżonych, co jest związane z ich architekturą. Wachlarz problemów, którymi zajmują się metody numeryczne jest bardzo szeroki i dziś jest to osobny, bardzo rozbudowany dział matematyki oferujący wiele bardzo ciekawych algorytmów.

Z informatycznego punktu widzenia natomiast interesuje nas tu przede wszystkim przełożenie tych algorytmów na możliwość pisania programów komputerowych, które automatycznie rozwiązują wybrane (często bardzo złożone pod względem obliczeniowym) zadania.

Do najważniejszych zagadnień metod numerycznych należą m.in:

WYBRANE PROBLEMY METOD NUMERYCZNYCH

1. ZAGADNIENIE INTERPOLACJI
2. METODY PRZYBLIŻONEGO ROZWIĄZYWANIA RÓWNAŃ NIELINIOWYCH
3. METODY ROZWIĄZYWANIA UKŁADÓW RÓWNAŃ LINIOWYCH
4. CAŁKOWANIE NUMERYCZNE

Przejdziemy teraz do omówienia tych problemów i ich rozwiązań.

Zagadnienie interpolacji w ogólnym przypadku można sformułować następująco: danych jest $n+1$ punktów na płaszczyźnie: $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$. Należy znaleźć funkcję $f(x)$, której wykres przechodzi przez te punkty oraz czasem spełnia dodatkowe warunki związane z wartościami pochodnych funkcji $f(x)$ w punktach $x_0, x_1, \ldots, x_n$. W ogólnym przypadku zagadnienie to może nie mieć rozwiązań lub może mieć ich nieskończenie wiele. My jednak zajmiemy się sytuacjami, w których rozwiązanie tego zagadnienia istnieje i jest jednoznaczne.

WZÓR INTERPOLACYJNY TAYLORA

Na początku zajmiemy się następującym zagadnieniem:
należy znaleźć funkcję $f(x)$, która w danym punkcie x_0 spełnia następujące warunki:

- $f(x_0) = a_0$
- $f'(x_0) = a_1$

  $\dots$
- $f^{(n)}(x_0) = a_n$

W ogólnym przypadku funkcji spełniających te warunki może być wiele. Jednak, jeśli dodatkowo wymagamy, aby szukana funkcja $f(x)$ była wielomianem stopnia co najwyżej n, to zagadnienie ma tylko jedno rozwiązanie. Jest nim wielomian interpolacyjny Taylora.

WZÓR INTERPOLACYJNY TAYLORA ORAZ JEGO WYKORZYSTANIE

Wielomian interpolacyjny Taylora wygląda następująco:

$$f(x) = \frac{f(x_0)}{0!} + \frac{f'(x_0)(x-x_0)}{1!} + \frac{f''(x_0)(x-x_0)^2}{2!} + \ldots + \frac{f^{(n)}(x_0)(x-x_0)^n}{n!}$$

Za pomocą tego wielomianu można obliczać przybliżone wartości innych funkcji np.:

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!}$$

$$\sin x \approx x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \ldots + \frac{x^{2n-1}}{(2n-1)!}$$

$$\cos x \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots + \frac{(-1)^n x^{2n}}{(2n)!}$$

Oto przykłady wykorzystania:

$$e^2 \approx 1 + 2 + \frac{2^2}{2!} + \frac{2^3}{3!} + \frac{2^4}{4!} + \frac{2^5}{5!} = 7,267$$

$$\sin\frac{\pi}{4} \approx \frac{\pi}{4} + \frac{(\frac{\pi}{4})^3}{3!} + \frac{(\frac{\pi}{4})^5}{5!} + \frac{(\frac{\pi}{4})^7}{7!} = 0,869$$

$$\cos\frac{\pi}{3} \approx 1 - \frac{(\frac{\pi}{3})^2}{2!} + \frac{(\frac{\pi}{3})^4}{4!} - \frac{(\frac{\pi}{3})^6}{6!} = 0,500$$

Widać, że przybliżenia te są dość dobre. Wzór Taylora jest bardzo często wykorzystywany przez inżynierów do obliczania przybliżonych wartości funkcji, które nie są wielomianami. Algorytm ten wykorzystują też kalkulatory i inne maszyny liczące do obliczania pierwiastków, logarytmów, funkcji trygonometrycznych, cyklometrycznych, wykładniczych oraz innych.

Przykład 1.

Znajdziemy wielomian $W(x)$ spełniający warunki $W(2) = 1$, $W'(2) = 2$, $W''(2) = 3$ oraz $W^{(3)}(2) = 4$. Zgodnie ze wzorem Taylora wielomian ten wyraża się następującym wzorem:

$$W(x) = 1 + 2(x-2) + \frac{3(x-2)^2}{2!} + \frac{4(x-2)^3}{3!}$$

Po przekształceniach otrzymujemy:

$$W(x) = \frac{2}{3}x^3 - \frac{5}{2}x^2 + 4x - \frac{7}{3}$$

Rzeczywiście $W(2) = 1$, $W'(x) = 2x^2 - 5x + 4$, $W'(2) = 2$, $W''(x) = 4x - 5$, $W''(2) = 3$, $W^{(3)}(x) = W^{(3)}(2) = 4$.

W dalszym ciągu zajmiemy się następującym problemem: danych jest $n+1$ punktów: $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$. Należy znaleźć wielomian $W(x)$ stopnia co najwyżej n, który przechodzi przez te punkty. Dodatkowo załóżmy, że punkty $x_0, x_1, \ldots, x_n$ są rozmieszczone w równych odstępach tzn. $x_i - x_{i-1} = const = h$ dla $i = \overline{1, n}$. W takiej sytuacji wielomian interpolacyjny tworzymy o tzw. tabelę różnic skończonych, którą zaprezentujemy na kolejnym przykładzie.

Przykład 2.

Utworzymy tabelę różnic skończonych dla wielomianu $W(x)$ przechodzącego przez punkty: $(1, 2)$, $(2, 4)$, $(3, 1)$, $(4, 7)$ oraz $(5, 5)$.

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$
1	2	2	-5	14	-31
2	4	-3	9	-17	
3	1	6	-8		
4	7	-2			
5	5				

WIELOMIAN INTERPOLACYJNY NEWTONA

W oparciu o przedstawioną tabelę można znaleźć wielomian interpolacyjny stopnia co najwyżej czwartego. Wielomian ten nazywa się wielomianem interpolacyjnym Newtona. Wielomian ten można uzyskać na podstawie górnych (czerwonych) współczynników tabeli (I wielomian interpolacyjny Newtona) lub na podstawie dolnych (niebieskich) współczynników tabeli (II wielomian interpolacyjny Newtona). Wzory na te wielomiany przedstawiają się następująco:

$$N_I(x) = \frac{2}{h^0 0!} + \frac{2}{h^1 1!}(x-1) - \frac{5}{h^2 2!}(x-1)(x-2) + \frac{14}{h^3 3!}(x-1)(x-2)(x-3) -$$

$$-\frac{31}{h^4 4!}(x-1)(x-2)(x-3)(x-4) =$$

$$= 2 + 2(x-1) - \frac{5}{2}(x-1)(x-2) + \frac{7}{3}(x-1)(x-2)(x-3) -$$

$$-\frac{31}{24}(x-1)(x-2)(x-3)(x-4) =$$

$$= -\frac{31}{24}x^4 + \frac{61}{4}x^3 - \frac{1481}{24}x^2 + \frac{399}{4}x - 50$$

$$N_{II}(x) = \frac{5}{h^0 0!} - \frac{2}{h^1 1!}(x-5) - \frac{8}{h^2 2!}(x-5)(x-4) - \frac{17}{h^3 3!}(x-5)(x-4)(x-3) -$$

$$-\frac{31}{h^4 4!}(x-5)(x-4)(x-3)(x-2) =$$

$$= 5 - 2(x-5) - 4(x-5)(x-4) - \frac{17}{6}(x-5)(x-4)(x-3) -$$

$$-\frac{31}{24}(x-5)(x-4)(x-3)(x-2) =$$

$$= -\frac{31}{24}x^4 + \frac{61}{4}x^3 - \frac{1481}{24}x^2 + \frac{399}{4}x - 50$$

Oczywiście oba wielomiany są równe. Jednak ze względów obliczeniowych pierwszy wielomian Newtona wykorzystujemy do interpolacji funkcji w pobliżu początku tablicy, a drugi do interpolacji funkcji w pobliżu końca tablicy.

WIELOMIAN INTERPOLACYJNY LAGRANGE'A

Zajmiemy się teraz zagadnieniem interpolacji w sytuacji, gdy punkty $x_0, x_1, \ldots, x_n$ niekoniecznie są położone w równej odległości. W takiej sytuacji nie korzystamy z tabeli różnic skończonych. Wielomian interpolacyjny tworzymy wówczas wyłącznie w oparciu o liczby $x_0, x_1, \ldots, x_n, y_0, y_1, \ldots, y_n$. Wielomian w takim przypadku nazywa się wielomianem interpolacyjnym Lagrange'a.

Przykład 3.

Znajdziemy wielomian $W(x)$ przechodzący przez punkty: $(1,2)$, $(2,4)$, $(3,1)$, $(4,7)$ oraz $(5,5)$. Wielomian interpolacyjny Lagrange'a wygląda następująco:

$$W(x) = 2\cdot\frac{(x-2)(x-3)(x-4)(x-5)}{(1-2)(1-3)(1-4)(1-5)} + 4\cdot\frac{(x-1)(x-3)(x-4)(x-5)}{(2-1)(2-3)(2-4)(2-5)}$$

$$+1\cdot\frac{(x-1)(x-2)(x-4)(x-5)}{(3-1)(3-2)(3-4)(3-5)} + 7\cdot\frac{(x-1)(x-2)(x-3)(x-5)}{(4-1)(4-2)(4-3)(4-5)}$$

$$+5\cdot\frac{(x-1)(x-2)(x-3)(x-4)}{(5-1)(5-2)(5-3)(5-4)} = -\frac{31}{24}x^4 + \frac{61}{4}x^3 - \frac{1481}{24}x^2 + \frac{399}{4}x - 50$$

METODY PRZYBLIŻONEGO ROZWIĄZYWANIA RÓWNAŃ NIELINIOWYCH

Z rozwiązywaniem wszelkiego rodzaju równań spotykamy się bardzo często. Zauważmy, że efektywne sposoby znajdowania pierwiastków równań istnieją tylko dla niewielkiej ich grupy. Umiemy zawsze rozwiązać równania liniowe oraz kwadratowe. Istnieją również (aczkolwiek bardzo skomplikowane) wzory dla równań stopnia trzeciego oraz czwartego. Dla równań wielomianowych stopnia wyższego niż 4 nie istnieją żadne efektywne metody ich rozwiązywania i równania takie jesteśmy w stanie rozwiązać tylko w pewnych szczególnych przypadkach. W sytuacji, gdy w równaniu $f(x) = 0$ funkcja $f(x)$ nie jest wielomianem sytuacja staje się jeszcze trudniejsza i takie równania bardzo rzadko w ogóle jesteśmy w stanie rozwiązać w sposób analityczny. Istnieją jednak metody przybliżone, które pozwalają znaleźć przybliżenia pierwiastków takich równań. Omówimy je teraz bardziej szczegółowo.

METODA BISEKCJI (PODZIAŁU POŁÓWKOWEGO)

Omawiane poniżej metody oparte są na następującym twierdzeniu:

Twierdzenie Bolzano-Cauchy'ego

Jeżeli funkcja $f(x)$ jest ciągła w przedziale domkniętym $[a, b]$ oraz $f(a) \cdot f(b) < 0$, to wewnątrz przedziału $[a, b]$ istnieje przynajmniej jeden pierwiastek równania $f(x) = 0$.

Istota metody bisekcji polega na podzieleniu przedziału $[a, b]$ na dwie połowy. W dalszym ciągu, jeśli $f(\frac{a+b}{2}) = 0$, to szukany pierwiastek wynosi $x_0 = \frac{a+b}{2}$, a jeśli nie, to z obu podprzedziałów $[a, \frac{a+b}{2}]$, $[\frac{a+b}{2}, b]$ wybieramy ten, na którego końcach funkcja $f(x)$ przyjmuje różne znaki. Następnie połowimy nowy zawężony przedział i od nowa rozpatrujemy w nim wartości i znaki funkcji $f(x)$ itd. W ten sposób otrzymamy albo pierwiastek dokładny x_0 albo ciąg podprzedziałów, których końce dążą do szukanego pierwiastka.

Przykład 4.

Znajdziemy rozwiązanie równania $x^3 - x - 1 = 0$ w przedziale $[0, 2]$. Zauważmy, że $f(0) = -1$, $f(1) = -1$, $f(2) = 5$. Zatem do dalszego połowienia wybieramy przedział $[1, 2]$ Teraz mamy $f(1, 5) = 0, 875$. Zatem do dalszego połowienia bierzemy przedział $[1; 1, 5]$ Teraz $f(1, 25) = -0, 296875$. Zatem do dalszego połowienia bierzemy przedział $[1, 25; 1, 5]$. Teraz $f(1, 375) = 0, 224609$. Zatem do dalszego podziału bierzemy przedział $[1, 25; 1, 375]$ Teraz $f(1, 3125) = -0, 0515137$. Zatem do kolejnego połowienia bierzemy przedział $[1, 3125; 1, 375]$ Teraz $f(1, 34375) = 0, 0826111$. Zatem w dalszej kolejności połowimy przedział $[1, 3125; 1, 34375]$. Teraz $f(1, 328125) = 0.014576$. Możemy zatem przyjąć $x_0 \approx 1, 328125$. Oczywiście możemy dalej połowić dla uzyskania lepszej dokładności. Niestety ta metoda jest dość wolno zbieżna. Jednak wykorzystując programy komputerowe można uzyskiwać rozwiązania z dowolną dokładnością.

METODA BISEKCJI - PRZYKŁADOWY PROGRAM

```
def fun(x):
   return x*x*x-x-1
def main():
   a=0
   b=2
   sr=(a+b)/2
   while abs(fun(sr))>=0.000001:
      sr=(a+b)/2
      if fun(sr)==0:
         break
      elif fun(a)*fun(sr)<0:
         b=sr
      else:
         a=sr
      print(sr)
   print("Przyblienie rozwiazania to:",sr)
   input()
main()
```

METODA SIECZNYCH

Szybszą metodą jest metoda siecznych, która w ogólnym przypadku polega na tworzeniu dla funkcji $f(x)$ w przedziale $[a, b]$ ciągu siecznych, których miejsca zerowe są zbieżne do rozwiązania równania $f(x) = 0$. Dla tej metody mamy dwa wzory w zależności od znaków funkcji $f(x)$ oraz $f''(x)$ w punktach a oraz b.
Jeżeli $sgn(f(b)) = sgn(f''(b))$, to rozwiązania poszukujemy w oparciu o następującą procedurę:

$$x_0 = a, \; x_{n+1} = x_n - \frac{f(x_n)}{f(b) - f(x_n)}(b - x_n)$$

Natomiast, gdy $sgn(f(a)) = sgn(f''(a))$, to rozwiązania poszukujemy w oparciu o następującą procedurę:

$$x_0 = b, \; x_{n+1} = x_n - \frac{f(x_n)}{f(x_n) - f(a)}(x_n - a)$$

METODA SIECZNYCH - PROGRAM

```
def fun(x):
   return x*x*x-x-1
def poch2(x):
   return 6*x
def main():
   a,b=0,2
   if fun(b)*poch2(b)>0:
      x=a
      while abs(fun(x))>=0.000001:
         x=x-(fun(x)/(fun(b)-fun(x)))*(b-x)
         print(x)
   elif fun(a)*poch2(a)>0:
      x=b
      while abs(fun(x))>=0.000001:
         x=x-(fun(x)/(fun(x)-fun(a)))*(x-a)
         print(x)
   print("Przyblienie rozwiazania to:",x)
   input()
main()
```

Poprzednia metoda też często jest wolno zbieżna. Dzieje się tak dlatego, że w zasadzie nie bierzemy pod uwagę kształtu wykresu. Zdecydowanie lepszą metodą jest metoda stycznych, która polega na tworzeniu dla funkcji $f(x)$ w przedziale $[a, b]$ ciągu stycznych, których miejsca zerowe są zbieżne do rozwiązania równania $f(x) = 0$. Aby stosować tą metodę pierwsza i druga pochodna $f'(x)$ oraz $f''(x)$ muszą być ciągłe i nie mogą zmieniać znaku wewnątrz przedziału $[a, b]$.
Rozwiązania poszukujemy w oparciu o następującą procedurę:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n},$$

przy czym jako x_0 przyjmujemy a , gdy $sgn(f(a)) = sgn(f''(a))$ oraz b, gdy $sgn(f(b)) = sgn(f''(b))$.

METODA STYCZNYCH - ALGORYTM ZNAJDOWANIA PIERWIASTKA

Wykorzystamy metodę stycznych do numerycznego znajdowania wartości pierwiastka $\sqrt{a}$. Wartość ta jest rozwiązaniem równania $x^2 - a = 0$ w przedziale np. $[0, a]$. Tutaj $f'(x) = 2x$, $f''(x) = 2$. Zatem pierwsze dwie pochodne są ciągłe i nie zmieniają znaku w przedziale $[0, a]$. Punktem startowym będzie tutaj punkt a. Wykorzystując poprzedni wzór otrzymujemy następującą rekurencyjną procedurę (znaną już w starożytności - jest to jeden z pierwszych algorytmów):

$$x_{n+1} = x_n - \frac{(x_n)^2 - a}{2x_n}$$

Inaczej można to zapisać w następujący sposób:

$$x_{n+1} = \frac{1}{2}(x_n + \frac{a}{x_n})$$

ZNAJDOWANIE PIERWIASTKA KWADRATOWEGO - PROGRAM

ALGORYTM ZNAJDOWANIA PIERWIASTKA KWADRATOWEGO

```
print("Podaj a:")
a=float(input())
x=a
while x*x-a>=0.000001:
   x=0.5*(x+a/x)
   print(x)
print("Pierwiastek z ",a," wynosi ",x)
input()
```

METODA ITERACYJNA

Ostatnia z omawianych metod jest również bardzo skuteczna (podobnie jak metoda stycznych), jednak można ją stosować jedynie w pewnych bardzo szczególnych przypadkach. Metoda ta opiera się o twierdzenie Banacha o odwzorowaniach zwężających. Jeżeli odwzorowanie $f(x)$ jest zwężające, to istnieje tylko jeden punkt x_0, taki, że $f(x_0) = x_0$. Ponadto ciąg określony w następujący sposób:

$$x_{n+1} = f(x_n)$$

jest zbieżny do rozwiązania równania $f(x) = x$ niezależnie od wyboru punktu początkowego. Odwzorowanie jest zwężające, gdy na przykład $|f'(x)| < 1$ dla $x \in [a, b]$.

METODA ITERACYJNA

Istota metody polega na takim przekształceniu danego równania, aby dało się go zapisać w postaci $f(x) = x$, gdzie $|f'(x)| < 1$ dla $x \in [a, b]$. Rozpatrzmy na przykład równanie $\cos x = 2x$. Możemy je przekształcić do postaci $\frac{1}{2}\cos x = x$. Ponieważ pochodna lewej strony tego równania jest równa $-\frac{1}{2}\sin x$ oraz $\left|-\frac{1}{2}\sin x\right| < 1$, więc możemy znaleźć rozwiązanie tego równania np. w przedziale $[0, \frac{\pi}{2}]$, korzystając z procedury:

$$x_{n+1} = \frac{1}{2}\cos x_n$$

METODA ITERACYJNA - PROGRAM

```
import math
x=0
while abs(math.cos(x)-2*x)>=0.000001:
   x=0.5*math.cos(x)
   print(x)
print("Rozwiazanie rownania to:",x)
input()
```