Kacper Kula

Zadanie numeryczne nr 3

Omówienie

Problem polega na rozwiązaniu układu macierzowego $y = A^{-1}x$, wybierając odpowiednią, czyli najbardziej optymalną metodę, wykorzystując przy tym charakterystyczną budowę macierzy.

Macierz A jest zdefiniowana następująco:

$$\mathbf{A} = \begin{pmatrix} 1.2 & \frac{0.1}{1} & \frac{0.15}{1^2} & & & & & & \\ 0.2 & 1.2 & \frac{0.1}{2} & \frac{0.15}{2^2} & & & & & \\ & 0.2 & 1.2 & \frac{0.1}{3} & \frac{0.15}{3^2} & & & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ & & & & & 0.2 & 1.2 & \frac{0.1}{N-2} & \frac{0.15}{(N-2)^2} \\ & & & & & & 0.2 & 1.2 & \frac{0.1}{N-1} \\ & & & & & & & 0.2 & 1.2 \end{pmatrix}$$

Wektor x zdefiniowany jest następująco:
$x = (1, 2, 3, \dots, N)^T$

Ponadto, należy rozwiązać wyznacznik macierzy A oraz stworzyć wykres zależności czasu działania utworzonego programu od parametru N.

Narzędziem do napisania programu rozwiązującego zadanie jest język Python oraz biblioteka pythona służąca do tworzenia wykresów – matplotlib.
Do sprawdzenia poprawności programu wykorzystana została biblioteka z zakresu algebry liniowej - numpy.

Rozwiązanie układu macierzowego

Macierze, w których liczba miejsc niezerowych rośnie wolniej niż wymiar macierzy, nazywamy macierzami rzadkimi.
Wykorzystując odpowiednie algorytmy do operowania na macierzach rzadkich, możemy zaoszczędzić czas wykonania i pamięć potrzebną na obliczenia.

Macierz A jest typem macierzy wstęgowej lub pasmowej, czyli taką, której wszystkie elementy są zerami oprócz jej diagonali i elementów wokół niej.

Aby efektywnie obliczyć równanie $y=A^{-1}x$ należy:

(1) Zastosować rozkład LU dla macierzy A

(2) Rozwiązać układ przy pomocy metody „back substitution"

Dla ogólnego przypadku złożoność czasowa (1) wynosi $O(n^3)$, a dla punktu (2) wynosi $O(n^2)$, czyli całkowita złożoność czasowa to $O(n^3)+O(n^2)=O(n^3)$.

Wykorzystując jednak strukturę macierzy A i stosując odpowiednią metodę, która pomija obliczenia dla elementów zerowych, dla (1), jak i (2) możemy osiągnąć złożoność czaową $O(n)$.

Rozkład macierzy A do postaci LU

Wzory ogólne na poszczególne elementy macierzy rozkładu przedstawiają się następująco :

dla i ∈ {1, 2, …, N}

- $u_{ij}=a_{ij}-\sum_{k=1}^{i-1} l_{ik}*u_{kj}$, j ∈ {i, i+1, …, N}
- $l_{ij}=\frac{1}{u_{ii}}*(a_{ji}-\sum_{k=1}^{i-1} l_{jk}*u_{ki})$, j ∈ {i + 1, i+2, …, N}

Z budowy macierzy A wynika jednak, że jedyne niezerowe elementy mają postać:

dla i ∈ {1, 2, …, N}

$u_{ii}, u_{i\,i+1}, u_{i\,i+2}, l_{i+1\,i}, l_{i\,i}=1$

Możemy więc wyprowadzić następujące wzory:

- $u_{ii}=a_{ii}-l_{i\,i-1}*u_{i-1\,i}$
- $l_{i+1\,i}=\frac{1}{u_{ii}}(a_{i\,i+1})$
- $u_{i\,i+1}=a_{i\,i+1}-l_{i\,i-1}*u_{i-1\,i+1}$
- $u_{i\,i+2}=a_{i\,i+2}$

Do rozkładu LU zawsze będziemy N-razy wykonywać 4 operację, co daje nam liniową złożoność problemu.

Ponadto, wiedząc, że pod diagonalą znajduję się tylko jedna „wstążka", a nad diagonalą tylko dwie „wstążki", jedyna potrzebna nam pamięć dla elementów macierzy L i U to 4 tablice o długości N.

Rozwiązanie układu $y=A^{-1}x$ metodą „back substitution”

Aby efektywnie rozwiązać układ z zadania, należy wykorzystać wcześniej wyliczoną postać LU.
Postawiając do wzoru $Ay=x$ otrzymujemy: $LUy=x$, podstawiając $Uy=z$, tworzymy układ równań:

- $Lz=x$
- $Uy=z$

Dla ogólnego przypadku wzór na składowe wektora wynikowego to:

$$x_n=\frac{b_{nn}-\sum_{i=1}^{n-1} l_{ni}*x_i}{l_{nn}}$$

Wiedząc jednak, które elementy są niezerowe, dla każdej składowej $x_1, x_2, \ldots, x_n$ wykonamy tylko skończoną liczbę operacji, zamiast n liczby operacji, co prowadzi do osiągnięcia złożoności czasowej $O(n)$.

Obliczanie wyznacznika macierzy A

Aby z łatwością obliczyć wyznacznik macierzy A, możemy wykorzystać wcześniej wyliczoną postać LU tej macierzy.

Z własności wyznacznika: $det(A)=det(LU)=det(L)*det(U)$.
Wiemy również, że macierz L jest macierzą trójkątną dolną, a macierz U macierzą trójkątną górną.

Wyznacznik macierzy trójkątnych obliczamy z wzoru: $det(B)=\prod^{n} b_{ii}$.
Jednak wiemy, że elementy diagonalne macierzy L wynoszą 1, zatem $det(L)=1$.

Podsumowując: $det(A)=det(LU)=det(L)*det(U)=1*det(U)=\prod^{n} u_{ii}$

Porównanie i sprawdzenie wyników

Przy pomocy funkcji matmul() z biblioteki numpy wymnożyłem macierze L i U, a następnie porównałem wszystkie elementy powstałej macierzy z elementami oryginalnej macierzy A, z dokładności do pięciu miejsc po przecinku przy pomocy funkcji allclose().
<u>Macierze okazały się równe.</u>

Wyznacznik macierzy A obliczony przy pomocy funkcji linalg.get() z biblioteki numpy wynosi:
$det(A)=6141973498.857896$
Wyznacznik macierzy A obliczony przy pomocy opisanej metody wynosi:
$det(A)=6141973498.857843$

Przy pomocy funkcji linalg.solve() z biblioteki numpy obliczyłem układ macierzowy $y=A^{-1}x$, a następnie porównałem wynik z wektorem, który otrzymałem stosując opisane metody, z dokładnością do pięciu miejsc po przecinku.
<u>Wektory okazały się równe.</u>

Wektor wynikowy ma postać:

```
y = (0.44870083  1.41327329  2.13487785  2.86901327  3.59148857  4.31160496
   5.02982717  5.74701146  6.46350369  7.17952596  7.8952126   8.61065186
   9.32590362 10.04100995 10.75600127 11.47090008 12.18572339 12.90048431
  13.61519305 14.32985776 15.04448494 15.7590799  16.47364698 17.18818978
  17.90271132 18.61721411 19.3317003  20.04617173 20.76062997 21.47507637
  22.18951211 22.90393821 23.61835557 24.33276499 25.04716715 25.76156269
  26.47595215 27.19033601 27.90471473 28.61908869 29.33345825 30.04782373
  30.76218542 31.47654359 32.19089847 32.90525028 33.61959923 34.33394549
  35.04828923 35.76263061 36.47696977 37.19130683 37.90564193 38.61997516
  39.33430663 40.04863645 40.76296469 41.47729144 42.19161678 42.90594078
  43.6202635  44.33458501 45.04890537 45.76322462 46.47754284 47.19186005
  47.90617631 48.62049166 49.33480613 50.04911978 50.76343263 51.47774471
  52.19205606 52.90636671 53.62067669 54.33498602 55.04929473 55.76360284
  56.47791038 57.19221737 57.90652382 58.62082977 59.33513522 60.04944019
  60.76374471 61.47804879 62.19235244 62.90665567 63.62095851 64.33526096
  65.04956305 65.76386477 66.47816614 67.19246718 67.90676789 68.62106829
  69.33536838 70.04966818 70.76396768 71.47826691 72.19256587 72.90686457
  73.62116301 74.33546121 75.04975916 75.76405688 76.47835438 77.19265165
  77.90694872 78.62124557 79.33554222 80.04983867 80.76413494 81.47843101
  82.19272691 82.90702262 83.62131817 84.33561355 85.04990876 85.76420381
  86.4784987  87.19279247 87.90778524 88.68203579)^T
```

Czas działania programu

Średni czas wykonania programu dla N = 124 wyniósł: 0.1215 milisekundy. Przeprowadzone zostało 100 testów i wyciągnięta średnia czasu.

Program wykonałem również dla N = 10,0 200, 300, ..., 10000.
Czas liczony był <u>tylko</u> przy operacjach rozkładu macierzy do postaci LU i rozwiązania układu macierzowego metodą „back substitution".

Dla otrzymanych rezultatów sporządziłem wykres zależności czasu działania programu od parametru N.

Spodziewaliśmy się zależności czasowej O(n) - tę zależność potwierdza nam widziana linia prosta na wykresie, po której rozkładają się wyniki poszczególnych testów.