Rozwiązywanie układów równań liniowych metodami iteracyjnymi – transport ciepła

Michał Saturczak
Informatyka Stosowana, Akademia Górniczo-Hutnicza
Metody Numeryczne

15 marca 2024

1 Wstęp teoretyczny

Rozwiązywanie układów równań liniowych jest podstawowym zadaniem w wielu dziedzinach nauki i techniki. W kontekście problemów inżynierskich, takich jak transport ciepła, często stosuje się metody iteracyjne do rozwiązania układów równań liniowych.

Transport ciepła to proces, w którym energia w postaci ciepła jest przekazywana z jednego miejsca do drugiego. Problem jednowymiarowego transportu ciepła przez ścianę można zamodelować za pomocą równania różniczkowego drugiego rzędu, które jest równaniem transportu ciepła. Równanie to można dyskretyzować, co prowadzi do układu równań liniowych.

Pierwszym krokiem w procesie modelowania jest utworzenie siatki obliczeniowej, która reprezentuje ścianę. W jednowymiarowym problemie, siatka obliczeniowa sprowadza się do punktów położonych na osi x. Każdemu punktowi na tej osi przypisuje się wartość temperatury, reprezentowaną przez wektor t. Celem jest obliczenie tego wektora.

Równanie transportu ciepła dla problemu stacjonarnego (tzn. niezależnego od czasu) ma postać:

$$0 = \frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) \tag{1}$$

Można przekształcić to równanie na formę układu równań liniowych. Po dyskretyzacji i zastosowaniu metody różnic skończonych, otrzymujemy układ równań liniowych, który można zapisać w postaci $At = b$, gdzie A jest macierzą trójdiagonalną, t jest wektorem temperatur, a b jest wektorem prawych stron.

Rozwiązanie tego układu równań można znaleźć za pomocą różnych metod, zarówno bezpośrednich, jak i iteracyjnych. Jedną z metod iteracyjnych, którą można zastosować w tym przypadku, jest metoda największych spadków. W każdej iteracji tej metody oblicza się wektor reszt $r_k = b - At_k$,a następnie oblicza się kolejne przybliżenie rozwiązania $tk + 1 = t_k + \alpha_k r_k$, gdzie α_k jest długością kroku obliczaną na podstawie wektora reszt.

Metoda ta jest prostym i efektywnym narzędziem do rozwiązywania układów równań liniowych, szczególnie w kontekście problemów inżynierskich takich jak transport ciepła. Jednakże, jak każda metoda iteracyjna, wymaga ustalenia kryterium zbieżności, które określa, kiedy proces iteracyjny powinien zostać zatrzymany. W praktyce, proces iteracyjny jest zazwyczaj kontynuowany do momentu, gdy norma wektora reszt spadnie poniżej pewnego ustalonego progu, co oznacza, że osiągnięto wystarczająco dobre przybliżenie rzeczywistego rozwiązania.

2 Problem

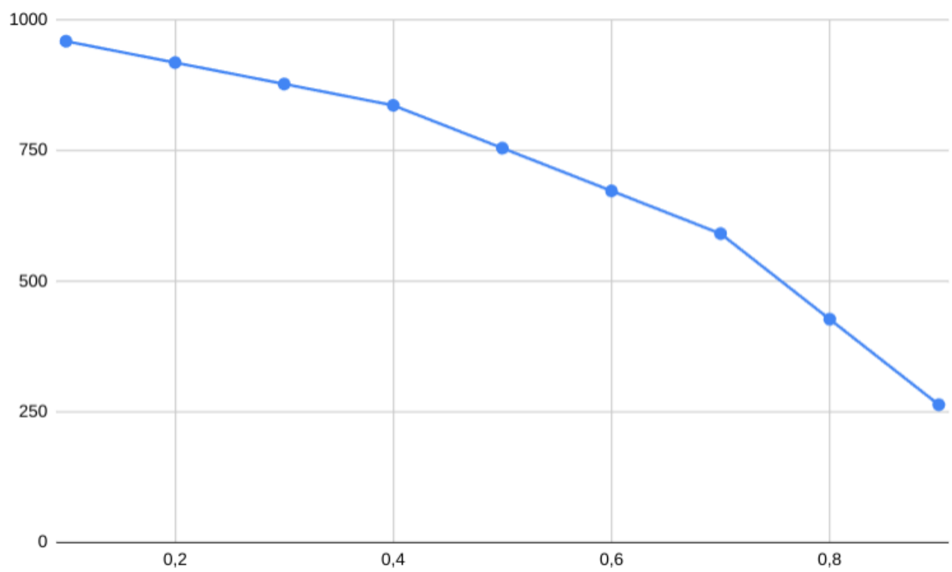
Funkcja rozwiązująca problem:

```
void result(gsl_matrix *A, gsl_vector *t, gsl_vector *b, const int n)
{
  gsl_vector *iloczyn = gsl_vector_calloc(n);
  gsl_vector *r_k = gsl_vector_calloc(n);
  gsl_vector *p = gsl_vector_calloc(n);
  float alfa = 0;
  float suma = 0;
  float mian = 0;
  float iloczyn_skal = 0;
  int licznik = 0;
    FILE *f;
  f = fopen("datanew.csv", "w");
  if (f == NULL)
  {
    perror("Nie udalo sie otworzyc pliku do zapisu");
    exit(1);
  }

  for (int x = 0; x < 60000; x++)
  {
    // iloczyn A * t_k
    for (int i = 0; i < n; i++)
    {
      suma = 0.0;
      for (int j = 0; j < n; j++)
      {
        suma += gsl_matrix_get(A, i, j) * gsl_vector_get(t, j);
      }
      gsl_vector_set(iloczyn, i, suma);
    }
    // r_k
    for (int i = 0; i < n; i++)
    {
      gsl_vector_set(r_k, i, gsl_vector_get(b, i) - gsl_vector_get(iloczyn, i));
    }


    // iloczyn r_kt * r_k // licznik w 2 równaniu
    iloczyn_skal = 0;
    for (int i = 0; i < n; i++)
      iloczyn_skal += gsl_vector_get(r_k, i) * gsl_vector_get(r_k, i);

    // sprawdzenie czy iloczyn nie wychozi poza zakres podany w zadaniu 10^-6
    if (sqrt(iloczyn_skal) < 0.000001)
    {
      return;
    }


    // iloczyn A*r_k
    for (int i = 0; i < n; ++i)
    {
      float s = 0;
      for (int j = 0; j < n; ++j)
      {
```

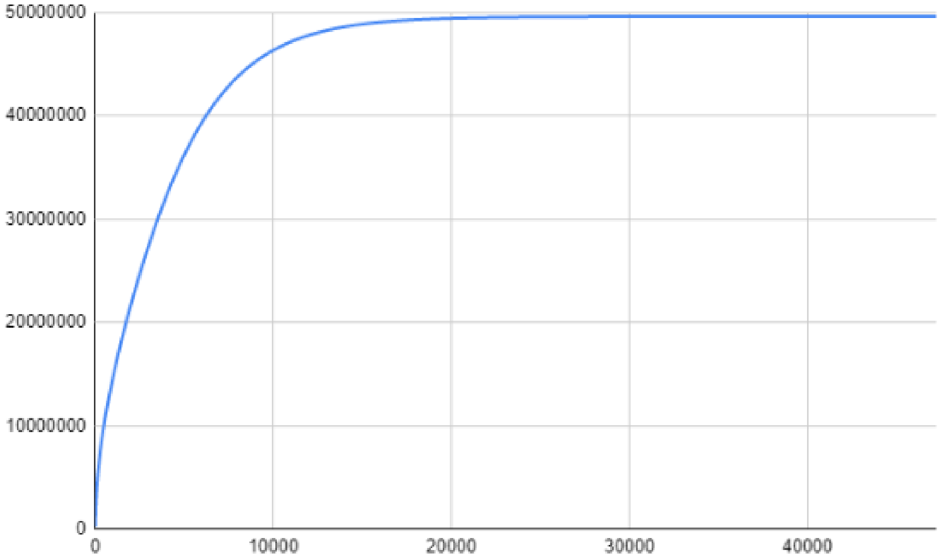
```
        s += gsl_matrix_get(A, i, j)*gsl_vector_get(r_k, j);
      }
      gsl_vector_set(p, i, s);
    }
    mian = 0;
    for (int i = 0; i < n; i++)
    {
      mian += gsl_vector_get(p, i) * gsl_vector_get(r_k, i);
    }

    alfa = (iloczyn_skal / mian);

    for (int i = 0; i < n; i++)
      gsl_vector_set(t, i, gsl_vector_get(t, i) + alfa * gsl_vector_get(r_k, i));

    float norma = 0;
    for (int i = 0; i < n; i++)
      norma += gsl_vector_get(t, i) * gsl_vector_get(t, i);
    fprintf(f,"%d\t%.2f\t%.2f\n", x, iloczyn_skal, sqrt(norma));
  }
  fclose(f);
}
```

3 Wyniki

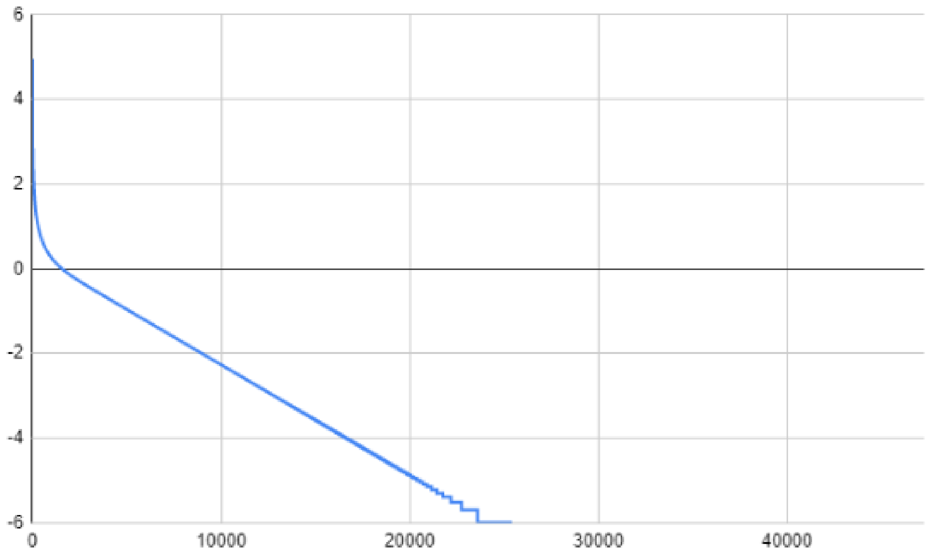
3.1 Wykres



3.2 Wykres $\|\mathbf{t}_k\|_2 = f(k)$



3.3 Wykres $\log(\|\mathbf{r}_k\|_2) = f(k)$



4 Wnioski

Zauważono że, otrzymane wyniki są charakteryzują się wysoką dokładnością. Metody stosowane do rozwiązywania podanych problemów nie należą do kategorii skomplikowanych, co dodatkowo podkreśla ich efektywność w kontekście rozważanych zastosowań.

Analizując przedstawione wykresy, można zauważyć, jak temperatura wzrasta na początku procesu transportu ciepła, a następnie utrzymuje się na stałym poziomie aż do końca procesu. Ta stabilność temperatury jest typowa dla problemów transportu ciepła w stanie stacjonarnym, gdzie po początkowym okresie wzrostu temperatura stabilizuje się i nie zmienia się już w dalszym czasie.

Dodatkowo, wykresy opisujące logarytmiczną zależność iloczynu skalarnego wektora reszt r z samym sobą pokazują, że wartość iloczynu skalarnego maleje wraz z kolejnymi iteracjami. To zjawisko jest zgodne z oczekiwaniem, że podczas iteracyjnego rozwiązania układu równań liniowych, reszty powinny dążyć do zera, co oznacza, że rozwiązanie staje się coraz bardziej dokładne.

Warto zauważyć, że dla pewnej ustalonej wartości, wykresy kończą się. Jest to związane z warunkiem zakończenia procesu iteracyjnego, który jest zazwyczaj zdefiniowany jako osiągnięcie przez normę wektora reszt wartości poniżej pewnego ustalonego progu. Może to również oznaczać, że proces iteracyjny zbiegł do dokładnego rozwiązania, co oznacza, że norma wektora reszt jest równa zero.

Wszystko to pokazuje, że metody iteracyjne są skutecznymi narzędziami do rozwiązywania problemów transportu ciepła, oferując dokładne rozwiązania przy stosunkowo niewielkim nakładzie obliczeniowym. Jest to szczególnie ważne w kontekście zastosowań inżynierskich, gdzie efektywność obliczeniowa i dokładność rozwiązania są kluczowe.