

Potencjał Elektrostatyczny

Modelowanie komputerowe

Matejko Marek, Mazur Krzysztof, Paszkot Dawid

Fizyka Techniczna
Politechnika Krakowska

Plan Prezentacji

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszkot

Wstęp

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Rozwiązanie równania Szeregiem Fouriera

Metoda Różnic
Skończonych

Metoda Różnic Skończonych

Implementacja

Implementacja

Bibliografia

Bibliografia

Wstęp

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Rozwiązanie równania Szeregiem Fouriera

Metoda Różnic
Skończonych

Metoda Różnic Skończonych

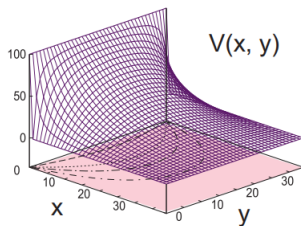
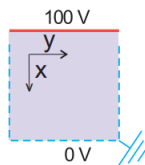
Implementacja

Implementacja

Bibliografia

Bibliografia

Znaleźć potencjał elektryczny we wszystkich punktach *wewnątrz* kwadratu wolnego od ładunku.



Dno i boki regionu składają się z drutów, które są „uziemiony” (utrzymywany w 0 V). Górny przewód jest podłączony do akumulatora, który utrzymuje go na stałym poziomie 100 V .

Eliptyczne równanie różniczkowe cząstkowe Laplace'a

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszko

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Metoda Różnic
Skończonych

Implementacja

Bibliografia

Z elektrodynamiki klasycznej wiadomo, że potencjał elektryczny $U(x)$ wynikający z ładunków statycznych spełnia równania różniczkowe Poissona:

$$\nabla^2 U(x) = -4\pi\rho(x)$$

gdzie $\rho(x)$ jest gęstością ładunku. W wolnych od ładunku rejonach przestrzeni, czyli rejonach, w których $\rho(x) = 0$, potencjał spełnia równanie Laplace'a:

$$\nabla^2 U(x) = 0$$

Eliptyczne równanie różniczkowe cząstkowe Laplace'a

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszko

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Metoda Różnic
Skończonych

Implementacja

Bibliografia

Oba te równania są eliptycznymi równaniami różniczkowymi cząstkowymi o postaci występującej w różnych zastosowaniach. My rozwiążemy je we współrzędnych prostokątnych dwuwymiarowych:

$$\frac{\partial U(x, y)}{\partial x^2} + \frac{\partial U(x, y)}{\partial y^2} = \begin{cases} 0, & \text{równanie Laplace'a,} \\ -4\pi\rho(x), & \text{równanie Poisson'a.} \end{cases}$$

W obu przypadkach widzimy, że potencjał zależy jednocześnie od x i y . Dla Laplace'a równania, ładunku, które są źródłem pola, wprowadzamy pośrednio, określając potencjalne wartości w jakimś rejonie przestrzeni; dla równania Poissona wpisują bezpośrednio.

Wstęp

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Rozwiązanie równania Szeregiem Fouriera

Metoda Różnic
Skończonych

Metoda Różnic Skończonych

Implementacja

Implementacja

Bibliografia

Bibliografia

Analityczne rozwiązanie równania Laplace'a istnieje w postaci nieskończonego szeregu

Rozdzielając zmienne, otrzymujemy

$$U(x, y) = X(x)Y(y) \Rightarrow \frac{d^2 X(x)/dx^2}{X(x)} + \frac{d^2 Y(y)/dy^2}{Y(y)} = 0$$

Aby powyższe równanie było spełnione, oba wyrażenia muszą być równe stałej

$$\frac{d^2 X(x)}{dx^2} + k^2 X(x) = 0, \quad \frac{d^2 Y(y)}{dy^2} - k^2 Y(y) = 0$$

Rozwiązanie równania Szeregiem Fouriera

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszko

Rozwiązania dla $X(x)$ są okresowe, a dla $Y(y)$ są wykładnicze

$$X(x) = A \sin kx + B \cos kx, \quad Y(y) = Ce^{ky} + De^{-ky}$$

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Metoda Różnic
Skończonych

Implementacja

Bibliografia

Warunki brzegowe

- ▶ $U(x = 0, y) = 0 \iff B = 0$
- ▶ $U(x = L, y) = 0 \iff kL = n\pi, n = 1, 2, \dots$
- ▶ $U(x, y = 0) = 0 \iff D = -C$

Rozwiązanie

$$X_n(x) = A_n \sin\left(\frac{n\pi}{L}x\right)$$

$$Y_n(y) = C(e^{k_n y} - e^{-k_n y}) = 2C \sinh \frac{n\pi}{L} y$$

Rozwiązanie równania Szeregiem Fouriera

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszko

Uwzględniając zasadę superpozycji otrzymujemy ogólne rozwiązanie

$$U(x, y) = \sum_{n=1}^{\infty} E_n \sin\left(\frac{n\pi}{L}x\right) \sinh\left(\frac{n\pi}{L}y\right)$$

Wartości E_n są ustalane przez pozostały warunek brzegowy
 $U(x, y = L) = 100V$

$$\sum_{n=1}^{\infty} E_n \sin\frac{n\pi}{L}x \sinh n\pi = 100V$$

E_n wyznaczamy mnożąc obie strony przez $\sin(m\pi/Lx)$,
gdzie $m \in \mathbb{Z}$, i całkując od 0 do L

$$\sum_n E_n \sinh n\pi \int_0^L dx \sin\frac{n\pi}{L}x \sin\frac{m\pi}{L}x = \int_0^L dx 100 \sin\frac{m\pi}{L}x$$

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Metoda Różnic
Skończonych

Implementacja

Bibliografia

Całka po lewej stronie równania jest niezerowa dla $n = m$, co daje

$$E_n = \begin{cases} 0 & \text{dla } n \text{ parzystych} \\ \frac{4(100)}{n\pi \sinh n\pi} & \text{dla } n \text{ nieparzystych} \end{cases}$$

Ostatecznie otrzymujemy potencjał w dowolnym punkcie (x,y) jako

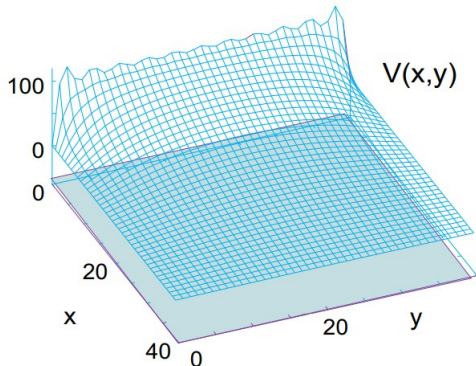
$$U(x, y) = \sum_{n=1,3,5,\dots}^{\infty} \frac{400}{n\pi} \sin\left(\frac{n\pi x}{L}\right) \frac{\sinh(n\pi y/L)}{\sinh(n\pi)}$$

Iloczyn oddzielnych funkcji x i y jest akceptowalnym rozwiązaniem równania Laplace'a, jednak są z tym problemy.

1. Realistyczne rozwiązanie może być nieskończoną sumą, ale sumy nie da się już oddzielić.
 - ▶ Dodatkowo sumowanie trzeba skończyć w pewnym punkcie, ale szereg zbiega się bardzo wolno
2. Funkcje \sinh przepelniają pamięć dla dużych n . Można to jednak złagodzić wyrażając ich iloraz w postaci eksponent i biorąc dużą granicę dla n

$$\frac{\sinh(n\pi y/L)}{\sinh(n\pi)} = \frac{e^{n\pi(y/L-1)} - e^{-n\pi(y/L+1)}}{1 - e^{-2n\pi}} \rightarrow e^{n\pi(y/L-1)}$$

3. Szereg rozwija oscylacje w punktach nieciągłych, czyli np. na rogach obszaru, tzw. Efekt Gibbsa



Rozwiązanie analityczne (szereg Fouriera) równania Laplace'a sumujące 21 wyrazów. Efekt Gibbsa prowadzi do oscylacji w pobliżu $x = 0$

Wstęp

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Rozwiązanie równania Szeregiem Fouriera

Metoda Różnic
Skończonych

Metoda Różnic Skończonych

Implementacja

Implementacja

Bibliografia

Bibliografia

Aby rozwiązać numerycznie nasze PDE, dzielimy przestrzeń na siatkę i rozwiązujemy dla każdego miejsca na siatce. Ponieważ będziemy wyrażać pochodne w warunkach różnic skończonych w wartościach U w miejscach sieci, nazywamy to metodą różnic skończonych.

Aby wyprowadzić algorytm różnic skończonych dla rozwiązania numerycznego zaczynamy od dodania dwóch rozwinięć Taylora potencjału po prawej i lewej stronie (x, y) oraz powyżej i poniżej (x, y) :

$$U(x + \Delta x, y) = U(x, y) + \frac{\partial U}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 U}{\partial x^2} (\Delta x)^2 + \dots,$$

$$U(x - \Delta x, y) = U(x, y) - \frac{\partial U}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 U}{\partial x^2} (\Delta x)^2 - \dots$$

Wszystkie nieparzyste wyrazy znoszą się, gdy dodamy te równania, i otrzymujemy przybliżenie różnicy środkowej dla drugiej pochodnej cząstkowej rzędu Δ^4

$$\frac{\partial^2 U(x, y)}{\partial x^2} \simeq \frac{U(x + \Delta x, y) + U(x - \Delta x, y) - 2U(x, y)}{(\Delta x)^2}$$

$$\frac{\partial^2 U(x, y)}{\partial y^2} \simeq \frac{U(x, y + \Delta y) + U(x, y - \Delta y) - 2U(x, y)}{(\Delta y)^2}$$

Podstawienie obu tych przybliżeń do równania Poissona prowadzi do postaci różnicy skończonej PDE. Zakładamy, że siatki x i y są w równych odstępach $\Delta x = \Delta y = \Delta$ więc algorytm przyjmuje prostą postać.

$$U(x + \Delta, y) + U(x - \Delta, y) + U(x, y + \Delta) + U(x, y - \Delta) - 4U(x, y) = -4\pi\rho.$$

Zastosujemy teraz rozwiązanie algebraiczne dla $U(x, y)$.

$$4U(x, y) \simeq U(x + \Delta, y) + U(x - \Delta, y) + U(x, y + \Delta) + U(x, y - \Delta) + 4\pi\rho(x, y)\Delta^2,$$

Gdzie pominielibyśmy wyraz $p(x)$ dla równania Laplace'a.

W warunkach lokalizacji na naszej sieci, zmienne x i y wyglądają

$$x = x_0 + i\Delta, \quad y = y_0 + j\Delta, \quad i, j = 0, \dots, N_{\max}-1,$$

W których umieściliśmy naszą sieć w kwadracie o boku L .
Algorytm skończonej różnicy przekształcił się w

$$U_{i,j} = \frac{1}{4} [U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1}] + \pi\rho(i\Delta, j\Delta)\Delta^2.$$

Istnieje wiele sposobów iteracji naszego algorytmu, w celu przekształcenia warunków brzegowych w rozwiązanie. Jego najbardziej podstawową formą jest metoda Jacobiego.

Raczej oczywiste ulepszenie metody Jacobiego wykorzystuje zaktualizowane domysły dotyczące potencjału w algorytmie, gdy tylko są one dostępne.

Technika ta, znana jako metoda Gaussa–Seidela (GS), zwykle prowadzi ona do przyspieszonej konwergencji, co z kolei prowadzi do mniejszego błędu zaokrąglania. Zużywa również mniej pamięci, ponieważ nie ma potrzeby przechowywania dwóch generacji domysłów.

$$U_{i,j}^{(\text{new})} = \frac{1}{4} \left[U_{i+1,j}^{(\text{old})} + U_{i-1,j}^{(\text{new})} + U_{i,j+1}^{(\text{old})} + U_{i,j-1}^{(\text{new})} \right]$$

Spis treści

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszko

Wstęp

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Rozwiązanie równania Szeregiem Fouriera

Metoda Różnic
Skończonych

Metoda Różnic Skończonych

Implementacja

Implementacja

Bibliografia

Bibliografia

Implementacja - Kod

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszko

Wstęp

Rozwiązanie
Równania
Szeregiem
Fouriera

Metoda Różnic
Skończonych

Implementacja

Bibliografia

```
1 # from numpy import *
2 import numpy as np
3 import matplotlib.pyplot as p;
4 from mpl_toolkits.mplot3d import Axes3D
5
6 print("Initializing")
7 Nmax = 100; Niter = 70; V = np.zeros((Nmax, Nmax), float) # float maybe Float
8
9 print( "Working hard, wait for the figure while I count to 60")
10 for k in range(0, Nmax-1): V[k,0] = 100.0 # line at 100V
11
12 for iter in range(Niter): # iterations over algorithm
13     if iter % 10 == 0: print( iter)
14     for i in range(1, Nmax-2):
15         for j in range(1,Nmax-2): V[i,j] = 0.25*(V[i-1,j]+V[i+1,j]+V[i,j-1]+V[i,j+1])
16
17 x = range(0, Nmax-1, 2); y = range(0, 50, 2) # plot every other point
18 X, Y = p.meshgrid(x,y)
19
20 def functz(V): # Function returns V(x, y)
21     z = V[X,Y]
22     return z
23
24 Z = functz(V)
25 fig = p.figure() # Create figure
26 ax = Axes3D(fig) # plot axes
27 ax.plot_wireframe(X, Y, Z, color = 'r') # red wireframe
28 ax.set_xlabel('X') # label axes
29 ax.set_ylabel('Y')
30 ax.set_zlabel('Potential')
31 p.show() # display fig, close shell to quit
```

Implementacja - Wykres

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszko

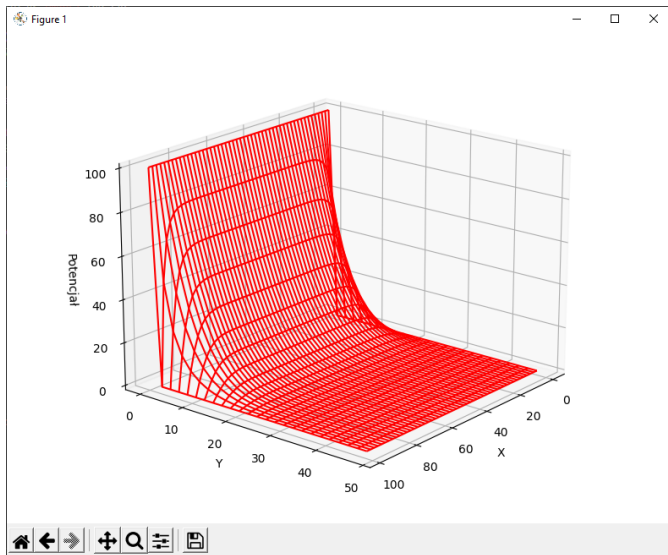
Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Metoda Różnic
Skończonych

Implementacja

Bibliografia



Spis treści

Potencjał
Elektrostatyczny

Matejko, Mazur,
Paszko

Wstęp

Wstęp

Rozwiązanie
równania
Szeregiem
Fouriera

Rozwiązanie równania Szeregiem Fouriera

Metoda Różnic
Skończonych

Metoda Różnic Skończonych

Implementacja

Implementacja

Bibliografia

Bibliografia

[1] <https://pl.wikipedia.org/wiki/Python>

[2] 1ElektrostatykaICieplo.pdf