

Zápočtová úloha č.2

Nail Sultanbekov

8. dubna 2024

Abstrakt

Cílem tohoto protokolu je pochopit a implementovat metodu konečných diferencí pro řešení Poissonovy rovnice. Budeme tady používat Jacobiho a Gauss-Seidelovu iterační metody pro řešení lineárních soustav a srovnávat jejich chyby v případě Poissonovy rovnici.

Numerické metody

Kratký popis metody konečných diferencí

Metoda konečných diferencí je numerická metoda používaná pro aproximaci řešení parciálních diferenciálních rovnic. Je založena na diskretizaci oblasti do sítě bodů a nahrazení derivací v diferenciální rovnici aproximacími konečných rozdílů.

Definujme následující diferenciální rovnici:

$$\frac{d^2u}{dx^2} = f(x)$$

kde $u(x)$ je neznámá funkce a $f(x)$ je daná pravá strana. Cílem je nalézt řešení $u(x)$, které tuto rovnici splňuje.

Pro aplikaci metody konečných diferencí diskretizujeme oblast do mřížky s body x_i , kde $i = 0, 1, \dots, M$. Nechť h jsou mezikrokové vzdálenosti.

Druhou derivace můžeme aproximovat pomocí centrální diferencí:

$$\frac{d^2u}{dx^2} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

Dosazením těchto aproximací do původní PDR získáme systém algebraických rovnic, které obsahují neznámé hodnoty u v každém bodě mřížky. Tento systém lze pak řešit pomocí iterativních metod nebo přímých solverů.

Přesnost metody konečných diferencí závisí na volbě mezikrokových vzdáleností h . S klesajícími mezikrokovými vzdálenostmi se aproximace stává přesnější, ale za cenu zvýšené výpočetní náročnosti.

Úvod do metod Jacobiho a Gauss-Seidela

Metody Jacobiho a Gauss-Seidela jsou iterativní numerické metody používané k řešení soustav lineárních rovnic $Ax = b$.

Jakobiho metoda

Metoda Jacobiho je založena na iterativním přístupu, při kterém se v každém kroku aktualizují hodnoty neznámých proměnných založené pouze na jejich předchozích hodnotách. Konkrétně se jedná o následující krok:

$$x_{k+1} = D^{-1} (b - (L + U)x_k),$$

kde $A = D + L + U$, D je diagonální matice, L je dolní trojúhelníková a U je horní trojúhelníková matice.

Gauss-Seidlová metoda

Gauss-Seidlová metoda je podobná Jakobího, avšak v každém kroku hodnoty x_{k+1} se aktualizují trochu jinak:

$$x_{k+1} = L^{-1} (b - Ux_k)$$

Srovnání metod

V této sekci prezentuji výsledky experimentů, které jsem provedl za účelem porovnání a analýzy chyb a konvergence metod používaných k řešení lineárních rovnic. Pro každou metodu jsem připravil několik tabulek s naměřenými daty z experimentů, které obsahují maximální chybu, chybu řešení a počet iterací potřebných pro dosažení konvergence v závislosti na velikosti kroku h .

Najprve jsem zkusil rozebrat pravou stranu $f(x) = \sin(x) - \cos(2x)$. Přesné řešení je tedy

$$u(x) = \frac{(x - 2x \sin(5) + 10 \sin(x) - 5 \cos^2(x) + x \cos^2(5) + 5)}{10}$$

Pro každou velikost kroku metody jsem zaznamenal následující údaje:

1. $\|Ax - b\|_{max}$ maximova norma při řešení lineární soustavy po konvergenci(nebo po dosažení nejvyššího počtu iterací).
2. $\|u_x - u(x_i)\|$ rozdíl mezi teoretickými a vypočtenými hodnotami.
3. Počet iterací, které byly potřebné pro dosažení konvergence. Maximálně 10^5 .
4. Časová náročnost, tedy doba, kterou jednotlivé metody potřebovaly k dosažení konvergence.

Step Values	h = 0.5	h = 0.1	h = 0.05	h = 0.01
Maximum error	3.3 e-12	2.65e-13	2.00e-13	1.25e-05
$u(x_i) - u_i$ error	0.743	0.14	0.07	0.237
Iterations	400-500	11800-11900	46100-46200	100000
Time(s)	0.01	0.92	3.68	94.04

Tabulka 1: Jakobího metoda pro $f(x) = \sin(x) - \cos(2x)$

Step Values	h = 0.5	h = 0.1	h = 0.05	h = 0.01
Maximum error	2.60E-12	2.80E-13	1.42E-13	1.25E-06
$u(x_i) - u_i$ error	0.743	0.140	0.070	0.041
Iterations	200-300	5800-5900	23000-23100	100000
Time(s)	0.09	3.53	15.93	292.42

Tabulka 2: Gauss-Seidlová metoda pro $f(x) = \sin(x) - \cos(2x)$

Step \ Values	h = 0.5	h = 0.1	h = 0.05	h = 0.01
Maximum error	2.22E-16	4.66E-16	5.25E-16	6.84E-16
$u(x_i) - u_i$ error	0.743	0.140	0.070	0,014

Tabulka 3: Build-in(numpy) metoda pro $f(x) = \sin(x) - \cos(2x)$

V případě iteračních metod se mi nepodařilo dosáhnout nastavené přesnosti 10^{-12} pro velikost kroku 0.01. Ale pro standardní solver numpy je vidět lineární závislost $u(x_i) - u_i$ chyby na kroku metody h viz tabulka 3.

Ted' koukneme na konstantní pravou stranu $f(x) = 1$. Řešení tedy:

$$u(x) = \frac{(27 - 5x)x}{10}$$

Step \ Values	h = 0.5	h = 0.1	h = 0.05	h = 0.01
Maximum error	7.00e-12	2.00e-13	2.00e-13	2.00e-05
$u(x_i) - u_i$ error	0.586	0.124	0.062	0.543
Iterations	400-500	12100-12200	47700-47800	100000
Time(s)	0.01	0.83	3.82	96.26

Tabulka 4: Jakobiho metoda pro konstantní pravou stranu

Step \ Values	h = 0.5	h = 0.1	h = 0.05	h = 0.01
Maximum error	7.00E-12	3.00E-13	2.00E-13	2.89E-06
$u(x_i) - u_i$ error	0.586	0.124	0.062	0.086
Iterations	200-300	6000-6100	23800-23900	100000
Time(s)	0.01	3.37	16.65	303.71

Tabulka 5: Gauss-Seidlová metoda pro konstantní pravou stranu

Step \ Values	h = 0.5	h = 0.1	h = 0.05	h = 0.01
Maximum error	8,00E-16	1,10E-15	1,23E-15	1,54E-15
$u(x_i) - u_i$ error	0,586	0,124	0,062	0,012

Tabulka 6: Build-in(numpy) metoda pro konstantní pravou stranu

Pro konstantní pravou stranu jsem nenašel nic zajímavého...