VÍCEÚROVŇOVÉ METODY (NMNV571)

POZNÁMKY K PŘEDNÁŠCE

2023/2024, LETNÍ SEMESTR

JAN PAPEŽ

ITERAČNÍ METODY, REZIDUUM, CHYBA

řešíme $Au=f$ $\quad$ u … přesné řešení, A regulární, f … rhs

přibližné řešení: $u^{(i)}$ – chceme-li zdůraznit iterační index

$\qquad\qquad\qquad\quad v$ – chceme-li libovolnou aproximaci

chyba – hledaný vektor $\quad e = u - v \quad (e^{(i)} = u - u^{(i)}) \quad$ (algebraická chyba)

- známe-li chybu, pak $v + e = v + u - v = u$, a tedy dostáváme přesné řešení
- jestliže aproximujeme chybu $e^{(i)} \approx d^{(i)}$, pak můžeme definovat iterační proces $\quad u^{(i+1)} = u^{(i)} + d^{(i)} \approx u^{(i)} + e^{(i)} = u$

reziduum – spočitatelný vektor $\quad r = f - Av \quad (r^{(i)} = f - Au^{(i)})$

$r = 0 \iff v = u$, ale malé reziduum neznačí malou chybu

důležitý vztah:

$$r = f - Av = Au - Av = A(u-v) = Ae$$

$$\left.\begin{array}{l} e = A^{-1}r \Rightarrow \|e\| \le \|A^{-1}\|\cdot\|r\| \\ Au = f \Rightarrow \|f\| \le \|A\|\cdot\|u\| \end{array}\right\} \frac{\|e\|}{\|u\|} \le \frac{\|A\|\cdot\|A^{-1}\|\cdot\|r\|}{\|f\|} = \kappa(A)\cdot\frac{\|r\|}{\|f\|}$$

iterační zpřesnění

1) $r^{(i)} = f - Au^{(i)}$
2) $Ad^{(i)} = r^{(i)}$ $\quad$ řešíme (přibližně)
3) $u^{(i+1)} = u^{(i)} + d^{(i)}$

jestliže řešení v 2) můžeme zapsat jako $d^{(i)} = M^{-1}r^{(i)}$ $\quad (M^{-1} \approx A^{-1})$,

pak $\quad u^{(i+1)} = u^{(i)} + M^{-1}r^{(i)} = u^{(i)} + M^{-1}(f - Au^{(i)})$

vztah pro chybu

$$e^{(i+1)} = u - u^{(i+1)} = u - u^{(i)} - M^{-1}Ae^{(i)} = (I - M^{-1}A)e^{(i)}$$

je-li M^{-1} nezávislé na $u^{(i)}, r^{(i)}, i, \ldots$, pak se ~~jede~~ jedná o lineární metodu (stacionární metoda)

příklady takových metod: (později v nich budeme mluvit)

Jacobiho metoda

Gauss-Seidelova metoda, SOR

Richardsonova metoda

příklad nelineárních metod: Krylovské metody

(ale i v předpodmíněných metodách počítáme
$z = M^{-1} r$ ... předpodmíněné reziduum)

MODELOVÝ PROBLÉM A JEHO DISKRETIZACE

$$-\Delta u = f \quad v \quad \Omega = (0,1)^d \qquad (\text{lze uvažovat: navíc } +\sigma u,\ \sigma \geq 0)$$
$$u = 0 \quad na \quad \partial\Omega$$

- budeme používat $d=1$ pro ilustraci a $d=2$ pro grafy (jinak ale síla multigridu je předvedena v $d=3$)

- krajní podmínky lze zobecnit

- oblast Ω může být složitější, ale u mg – toto byl jeden z největších do jiných zemí!

<u>diskretizace v 1D</u>

$x_j = j \cdot h = \frac{j}{n}$, $j = 0, \dots, n$

označme aproximaci v uzlech $v_j \approx u(x_j)$, $v = [v_1, v_2, \dots, v_{n-1}]^T$

u'' nahradíme centrální diferencí $u''(x_j) \approx \dfrac{-u(x_{j-1}) + 2u(x_j) - u(x_{j+1})}{h^2}$

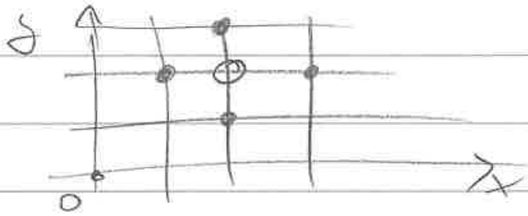
↳ dostáváme soustavu rovnic

$$v_0 = v_n = 0$$
$$\frac{-v_{j-1} + 2v_j - v_{j+1}}{h^2} = f(x_j) \qquad j = 1, 2, \dots, n-1$$
$$\vec{f} = [f(x_1), \dots, f(x_{n-1})]^T = [f_1, f_2, \dots, f_{n-1}]^T$$

↳ soustava

$$\frac{1}{h^2}\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} f_1 + v_0 \\ f_2 \\ \vdots \\ f_{n-1} + \underbrace{v_n}_{=0} \end{bmatrix}$$

(?) napřed rozmyslet, jak to vypadá bez zahrnutí okr. podmínek diskretizace v 2D



$$-\Delta u = -u_{xx} - u_{yy} = -\frac{\partial^2}{\partial x^2} u - \frac{\partial^2}{\partial y^2} u = f(x,y) \qquad 0 \le x,y \le 1$$

uzly sítě $(x_i, y_j) = (i \cdot h_x, j \cdot h_y)$ $\quad h_x = \frac{1}{m}$, $h_y = \frac{1}{n}$

$v_{ij} \approx u(x_i, y_j)$

centrální diference pro u_{xx} a u_{yy} dávají:

$$(*) \qquad \frac{-v_{i-1,j} + 2v_{ij} - v_{i+1,j}}{h_x^2} + \frac{-v_{i,j-1} + 2v_{ij} - v_{i,j+1}}{h_y^2} = f_{ij} = f(x_i, y_j)$$

$1 \le i \le m-1$, $1 \le j \le n-1$

okr. podmínky: $v_{0j} = v_{mj} = v_{i0} = v_{in} = 0$

$0 \le i \le m$, $0 \le j \le n$

máme $(m-1) \times (n-1)$ neznámých, které musíme nějak seřadit do vektoru
↳ buď "po řádcích", nebo "po sloupcích" (ostatní varianty teď nebudeme řešit)

nyní $\vec{v}_i = [v_{i1}, v_{i2}, \dots, v_{i,n-1}]^T \qquad 1 \le i \le m-1$

$\vec{f}_i = [f_{i1}, f_{i2}, \dots, f_{i,n-1}]^T$

Potom (*) můžeme zapsat maticově jako

$$\begin{bmatrix} B & -aI & & \\ -aI & B & \ddots & \\ & \ddots & \ddots & -aI \\ & & -aI & B \end{bmatrix} \begin{bmatrix} \vec{v}_1 \\ \vdots \\ \vec{v}_{m-1} \end{bmatrix} = \begin{bmatrix} \vec{f}_1 \\ \vdots \\ \vec{f}_{m-1} \end{bmatrix}$$

$$-aI = -\frac{1}{h_x^2} I_{n-1}, \qquad B = \frac{1}{h_y^2} \begin{bmatrix} 2\frac{(h_x^2 + h_y^2)}{h_x^2} & -1 & \\ -1 & \ddots & \ddots \\ & \ddots & \ddots \end{bmatrix}$$

+ jak to vypadá pro $m = n$, tj. $h_x = h_y$

KOREKCE NA HRUBÉ SÍTI

(30')

použití více sítí (obecně úrovní) je klíčovým prvkem multigridu
(síť/úroveň = obvykle podmnožina stupňů volnosti)
geometrický multigrid – existují různé hrubé sítě (odpovídá diskretizaci)
algebraický multigrid – úrovně vytváříme na základě nějakých algebraických vlastností (~~typicky se jedná o podmnožiny stupňů volnosti~~)

nejjednodušší případ (z pohledu vysvětlení i analýzy)
↳ dvouúrovňová metoda, two-grid method
- "jemná" síť, kde řešíme úlohu
- hrubá síť, která má pomoct k efektivnímu řešení

algoritmus může využít více sítí v úrovní, potřebujeme
- diskretizovanou úlohu na jemné i hrubé síti
  pro nás to budou matice A_1 a A_0
  (metodu se ~~x~~ dá značit obdobně A_ℓ a A_k, protože toto se lépe zobecňuje na více úrovní)
- "intergrid" operátory
  I_0^1 interpolace I_0 – "z kratšího vektoru delší"
  z vektoru odpovídajícímu hrubé síti vektor na jemné
  R_1^0 restrikce R_1 – "z delšího vektoru kratší"

důležitý předpoklad: problém na hrubé síti je menší než ten na jemné
(jinak si tím příliš nepomůžeme, jak uvidíme později)
$N_0 \ll N_1$

$A_1: \mathbb{R}^{N_1} \to \mathbb{R}^{N_1}$, $A_0: \mathbb{R}^{N_0} \to \mathbb{R}^{N_0}$, $I_0^1: \mathbb{R}^{N_0} \to \mathbb{R}^{N_1}$,
$R_1^0: \mathbb{R}^{N_1} \to \mathbb{R}^{N_0}$

Korekce na hrubé síti – myšlenka: problém zúžím na hrubé síti
a řešení použiju ke zpřesnění aproximace
tj. řešíme (přímou metodou, předp. že přesně)

$$A_0 d_0 = R_1^0 r \qquad \text{(proto chceme, aby } N_0 \ll N_1\text{)}$$

a doufáme, že

$$I_0^1 d_0 \approx e \qquad \text{interpolace } d_0 \text{ na jemnou síť}$$

(v zápisu pomocí iteračního zpřesnění: $u^{(i+1)} = u^{(i)} + I_0^1 A_0^{-1} R_1^0 r^{(i)}$)

Klíčová otázka: kdy můžeme čekat, že toto bude fungovat?
Interpolace i restrikce jsou obdélníkové matice, takže mají nějaký Range a Ker (jádro)

názorně (vágně): Potřebujeme, aby problém $A_0 d_0 = R_1 r$ byl "blízký" úloze $A_1 d = r$, aby zachytil "vše důležité"

algebraicky: • $\|I_0^1 d_0 - e\|$ může být (relativně) malé pouze tehdy, pokud e je blízko $\mathcal{R}(I_0)$
• pokud r je blízko $\mathrm{Ker}(R_1)$, pak $R_1^0 r$ je k ničemu

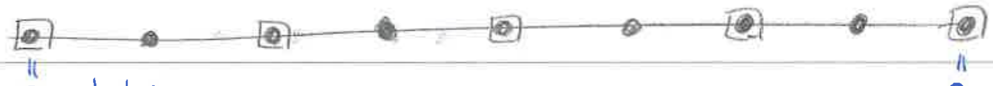
→ v důkazech konvergence uvidíme, jak se toto projeví
→ názorně to ale: motivuje k tomu, čím korekci na hrubé síti doplnit, abychom dostali efektivní řešič

Teď se podíváme na multigrid operace pro náš problém a diskretizaci.

INTERPOLACE A RESTRIKCE PRO KONEČNÉ DIFERENCE

Motivace v 1D:

jemná síť s krokem $h_1 = 1/n$
hrubší síť = každý druhý uzel, takže krok $h_0 = 2h_1$ $\Big\}$ $N_1 = 2N_0 + 1$ ~~[illegible]~~



• jemné
⊡ hrubé
uzly

0 okr. podm. 0

Interpolace – máme hodnoty v hrubých uzlech ⊡ a chceme dopočítat (snadno a levně) hodnoty mezi nimi
(to, že hodnoty v hrubých uzlech zachováme, je celkem přirozené)

někdy „prolongace"

VYSVĚTLIT NAPŘED GRAFICKY

$$I_0^1 \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_0 = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_7 \end{bmatrix}_1 = \begin{bmatrix} 1/2 & & \\ 1 & & \\ 1/2 & 1/2 & \\ & 1 & \\ & 1/2 & 1/2 \\ & & 1 \\ & & 1/2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_0$$

odp. hrubé síti

v uzlech mezi bereme 1/2 hodnoty zleva + 1/2 hodnoty zprava

Restrikce – máme hodnoty v jemných bodech a chceme určit hodnoty v ⊡

↳ nejsnazší volba: „injection" – vezmu hodnoty v ⊡, zbytek ignoruju

↳ jiná volba: „full weighting"

$$R_1^0 \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_7 \end{bmatrix}_1 = \begin{bmatrix} 1/4 & 1/2 & 1/4 & & & & \\ & & 1/4 & 1/2 & 1/4 & & \\ & & & & 1/4 & 1/2 & 1/4 \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_7 \end{bmatrix}_1 = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_0$$

navíc: $R_1 = \frac{1}{2} I_0^T$

To, že $R_1^0 = c I_0^{1T}$, $c \in \mathbb{R}$ se nazývá variational property, zjednodušuje to ~~te~~ teorii, protože to zachovává symetrii. Ukážeme si, že platí $A_0 = R_1^0 A_1 I_0^1$, což je užitečná vlastnost. Například proto, že korekce na hrubé síti jde vyjádřit jako

$$u^{(i+1)} = u^{(i)} + I_0^1 (R_1^0 A_1 I_0^1)^{-1} R_1^0 r^{(i)}$$

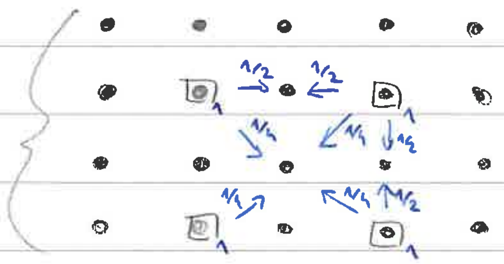
$$e^{(i+1)} = \underbrace{\left(I - I_0^1 (R_1^0 A_1 I_0^1)^{-1} R_1^0 A_1\right)}_{\text{tohle je projekce } (P^2 = P)} e^{(i)}$$

Jak vypadají interpolační vektory a jak vypadá restrikce vektorů? Označíme-li $V_0 \subset C(\Omega)$ prostor po částech lineárních spojitých fcí s hodnotami v hrubých uzlech, tak můžeme na interpolaci nahlížet jako na operátor $I_0^1 : V_0 \to V_0$. Chceme-li tedy $e \in \mathcal{R}(I_0^1)$, tak $\underline{e}$ musí být "hladká".

Interpolace a restrikce ve 2D (záleží na tom jak zvolíme hrubou síť!)

interpolace

obrázkem

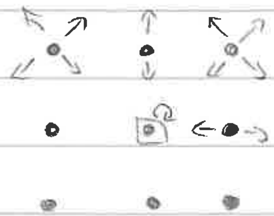


"stencil"

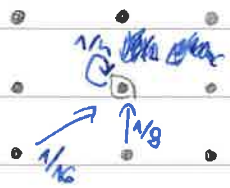
$$\begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}$$

hrubý uzel

restrikce



každý uzel (jemný) přispívá do 1, 2, nebo 4 hrubých uzlů
tomu odpovídají váhy (které se opět musí posčítat na jedničku)



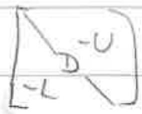
$$R_1 = \underbrace{1/4}_{= (1/2)^d} \cdot (I_0)^T$$

"coarsening factor = 2"

tato volba redukuje počet neznámých zhruba na 1/4 a zároveň každý jemný uzel má nějaký hrubý uzel "blízko"
(samozřejmě jsou možné i jiné varianty - nepravidelné v obou směrech, redukující neznámé více (méně, ...)
také stencil pro danou síť může být volen různě

(25')

STACIONÁRNÍ ITERAČNÍ METODY A SMOOTHING PROPERTY

štěpení matice $A = D - L - U$ $\qquad A =$ 

Jacobiho metoda

$$Au = f = Du - (L+U)u$$
$$\Rightarrow u = D^{-1}(L+U)u + D^{-1}f$$

$$u^{(i+1)} = R_J u^{(i)} + D^{-1}f := D^{-1}(L+U)u^{(i)} + D^{-1}f$$
$$R_J = D^{-1}(L+U)u^{(i)}$$

čemu to odpovídá pro naši úlohu a diskretizaci? (v 1D pro ilustraci)

j-tý řádek: $u_j^{(i+1)} = \frac{1}{2}\left(u_{j-1}^{(i)} + u_{j+1}^{(i)} + h^2 f_j\right)$

možná nevýhoda – uchovávání 2 celé vektory $u^{(i)}, u^{(i+1)}$

weighted (damped) Jacobi

varianta Jacobiho, kdy je nová aproximace lineární kombinací předchozí aproximace a aproximace dané Jacobiho metodou

$$\rightarrow u^{(*)} = R_J u^{(i)} + D^{-1}f$$
$$u^{(i+1)} = \omega \cdot u^{(*)} + (1-\omega)u^{(i)} = [\underbrace{(1-\omega)\cdot I + \omega R_J}_{R_\omega}]\, u^{(i)} + \omega D^{-1}f$$

Gauss – Seidel

$$(D-L)u = Uu + f$$
$$\Rightarrow u = (D-L)^{-1}Uu + (D-L)^{-1}f$$

$$u^{(i+1)} = \underbrace{(D-L)^{-1}U}_{R_{GS}}\, u^{(i)} + (D-L)^{-1}f$$

GS pro naši úlohu a diskretizaci:

j-tý řádek: $-1 \cdot u_{j-1}^{(i+1)} + 2\,u_j^{(i+1)} = u_{j+1}^{(i)} + h^2 f_j$

$$\Rightarrow \quad u_j^{(i+1)} = \frac{1}{2}\left(u_{j-1}^{(i+1)} + u_{j+1}^{(i)} + h^2 f_j \right)$$

neboli j-tý bod má aproximaci počítanou pomocí (j-1)ního již aktualizovaného bodu

=> v GS záleží na pořadí neznámých

=> v GS stačí ukládat 1 vektor, ve kterém přepisuju prvky

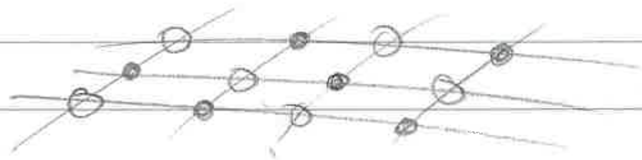
<u>red-black Gauss-Seidel</u>

varianta GS pro snazší paralelizaci

rozdělíme uzly do dvou skupin - černé a červené

v 1D

v 2D



tak, že pro aktualizaci všech černých uzlů potřebujeme jen hodnoty v červených a naopak

↳ pak není třeba aktualizovat hodnoty po jedné, ale můžeme to udělat ve dvou krocích vždy pro polovinu neznámých