



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Adam Piskalla

Deflated Conjugate Gradient Method

Katedra numerické matematiky

Vedoucí bakalářské práce: RNDr. Jan Papež, Ph.D.

Studijní program: Matematické modelování

Studijní obor: Matematické modelování

Praha 2023

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Děkuji doktoru Papežovi za trpělivé vedení této práce, podnětné komentáře a čas, který věnoval důkladným kontrolám textu.

Název práce: Deflated Conjugate Gradient Method

Autor: Adam Piskalla

Katedra: Katedra numerické matematiky

Vedoucí bakalářské práce: RNDr. Jan Papež, Ph.D., Matematický ústav Akademie věd České republiky

Abstrakt: Metoda sdružených gradientů je jednou ze základních iteračních metod pro řešení soustav lineárních algebraických rovnic se symetrickou pozitivně definitní maticí. V práci uvádíme dvě různá odvození této metody a ukazujeme některé její vlastnosti. V situacích, kdy metoda konverguje pomalu či téměř stagnuje, se obvykle používají techniky, které transformují původní soustavu s cílem konvergenci urychlit. Jednou z nich je předpodmínění, u kterého stručně uvádíme základní myšlenku a algoritmus předpodmíněných sdružených gradientů. Podrobněji se pak zaměřujeme na techniku tak zvané deflace. Představujeme kontext, v jakém byla popsána v literatuře, a komentujeme různé přístupy k odvození algoritmu deflated CG. Vysvětlujeme princip deflace a algoritmus detailně odvozujeme, přičemž popisujeme i kroky, které v literatuře nebývají explicitně uvedeny nebo podrobně rozebrány. Vliv deflace na rychlost konvergence ilustrujeme na jednoduchých numerických experimentech.

Klíčová slova: metoda sdružených gradientů, předpodmínění, deflace

Title: Deflated Conjugate Gradient Method

Author: Adam Piskalla

Department: Department of Numerical Mathematics

Supervisor: RNDr. Jan Papež, Ph.D., Institute of Mathematics of the Czech Academy of Sciences

Abstract: Conjugate gradient method is one of the basic iterative methods for solving systems of linear algebraic equations with a symmetric positive definite matrix. We present two different derivations of the method and show some its properties. In situations where the method converges slowly or almost stagnates, techniques that transform the original system are usually used to speed up the convergence. Among them there is a preconditioning, for which we briefly present the basic idea and algorithm of preconditioned conjugate gradients. We then focus in more detail on the so-called deflation. We present the context in which it has been described in the literature, and comment on various approaches to the derivation of the deflated CG algorithm. We explain the principle of deflation and derive thoroughly the algorithm, describing steps that are not explicitly stated or discussed in detail in the literature. On simple numerical experiments we illustrate the effect of the deflation on the convergence rate.

Keywords: conjugate gradient method, preconditioning, deflation

Obsah

Úvod	2
1 Metoda sdružených gradientů	3
1.1 CG jako minimalizace kvadratického funkcionálu	3
1.2 CG jako projekční metoda	4
1.3 Algoritmus	8
2 Transformace úlohy: předpodmínění a deflace	11
2.1 Předpodmínění	11
2.2 Deflace	12
2.2.1 Deflated CG v literatuře	12
2.2.2 Odvození algoritmu	13
2.2.3 Vlastnosti deflated CG	16
3 Numerické experimenty	19
Závěr	23
Seznam použité literatury	24

Úvod

Řešení soustavy lineárních algebraických rovnic je jednou ze základních úloh vědecko-technických výpočtů. Objevují se například v matematických modelech při diskretizaci parciálních diferenciálních rovnic. Takovéto soustavy bývají obvykle velké, a je tedy nutné je řešit iteračními metodami.

V úlohách, kdy je matice soustavy symetrická pozitivně definitní, je často používána metoda sdružených gradientů (CG), která byla poprvé navržena v Hestenes a Stiefel (1952). CG má řadu výhod: minimalizuje chybu na určitém podprostoru, není paměťově (příliš) náročná a pro její implementaci stačí operace matice soustavy krát vektor.

V mnoha případech však metoda sdružených gradientů konverguje velmi pomalu. Proto byly odvozeny různé techniky pro urychlení konvergence. Standardní technikou je předpodmínění, jehož podstatou je transformace původní úlohy do nové soustavy, která je lépe a rychleji řešitelná. Další možností je pak tak zvaná deflace, jejímž principem je konstrukce vhodného deflačního podprostoru (obvykle malé dimenze) a řešení úlohy na jejím ortogonálním doplňku. Na popis deflace a jejího odvození se zaměřuje tato práce.

V úvodní kapitole představíme dvě možná odvození metody sdružených gradientů a ukážeme vlastnosti výsledného algoritmu. V druhé kapitole nejprve popíšeme předpodmínění a uvedeme algoritmus předpodmíněných sdružených gradientů. Poté se zaměříme na techniku deflace, která byla poprvé uvedena v Nicolaidese (1987), krátce poté v Dostál (1988) a později například v Saad a kol. (2000). Představíme kontext těchto článků a okomentujeme různé přístupy k odvození algoritmu, který bývá v literatuře nazývaný deflated CG. Popíšeme princip deflace a detailně odvodíme algoritmus deflated CG, včetně kroků, které ve zmíněných článcích nejsou explicitně uvedeny nebo podrobně rozebrány. Rovněž ukážeme některé vlastnosti deflace. Algoritmy CG a deflated CG implementujeme v MATLABu a na jednoduchých příkladech budeme sledovat vliv deflace na rychlost konvergence. Výsledky těchto numerických experimentů představíme a okomentujeme v závěrečné kapitole.

1. Metoda sdružených gradientů

V této kapitole ukážeme dva způsoby odvození metody sdružených gradientů, konkrétně pomocí minimalizace kvadratického funkcionálu a jako projekční metodu. Uvedeme některé její vlastnosti a algoritmus, kterým bývá metoda standardně implementována. Budeme postupovat podobně jako v Liesen a Strakoš (2013, sekce 2.5.1 a 2.5.3).

Metoda sdružených gradientů (zkráceně CG z anglického názvu conjugate gradient method) je iterační metoda pro řešení soustav lineárních algebraických rovnic

$$Ax = b, \quad (1.1)$$

kde $A \in \mathbb{R}^{n \times n}$ je symetrická pozitivně definitní (SPD) matice a $b \in \mathbb{R}^n$ je reálný vektor. CG lze formulovat i pro komplexní data A, b . V této práci ji pro jednoduchost uvažujeme pouze pro reálné vstupy.

1.1 CG jako minimalizace kvadratického funkcionálu

Uvažujme funkcionál $F : \mathbb{R}^n \rightarrow \mathbb{R}$,

$$F(x) = \frac{1}{2}x^T Ax - x^T b. \quad (1.2)$$

Matice A je SPD, tudíž tento funkcionál má právě jeden extrém a jedná se o minimum. Platí $\nabla F(x) = Ax - b$ a z nutné podmínky pro minimum $\nabla F(x) = 0$ dostáváme právě soustavu (1.1). Řešení soustavy (1.1) je tedy ekvivalentní minimalizaci funkcionálu (1.2). Označme x_* (přesné) řešení (1.1) (nebo ekvivalentně bod minima (1.2)) a x_k jeho aproximaci. Potom

$$\begin{aligned} F(x_k) &= \frac{1}{2}x_k^T Ax_k - x_k^T Ax_* \left(+ \frac{1}{2}x_*^T Ax_* - \frac{1}{2}x_*^T Ax_* \right) \\ &= \frac{1}{2}(x_* - x_k)^T A(x_* - x_k) - \frac{1}{2}x_*^T Ax_* \\ &= \frac{1}{2}\|x_* - x_k\|_A^2 - \frac{1}{2}\|x_*\|_A^2, \end{aligned}$$

kde $\|x\|_A = \sqrt{x^T Ax}$ je tzv. energetická nebo A -norma. Minimalizace daného funkcionálu tak odpovídá minimalizaci A -normy chyby $\|x_* - x_k\|_A$.

Označme $x_0 \in \mathbb{R}^n$ počáteční aproximaci řešení a konstruujme posloupnost aproximací následující rekurencí

$$x_k = x_{k-1} + \gamma_{k-1}p_{k-1}, \quad k = 1, 2, \dots,$$

kde $\gamma_{k-1} \in \mathbb{R}$ je délka kroku a $p_{k-1} \in \mathbb{R}^n$ je směrový vektor k -tého kroku. Předpokládejme, že vektor p_{k-1} známe a chceme odvodit vztah pro koeficient γ_{k-1} . Budeme jej volit takový, aby minimalizoval

$$\|x_* - x_k\|_A^2 = \|x_* - x_{k-1}\|_A^2 - 2\gamma_{k-1}p_{k-1}^T r_{k-1} + \gamma_{k-1}^2 p_{k-1}^T A p_{k-1}.$$

Zderivováním podle γ_{k-1} a položením derivace rovno nule dostaneme

$$\gamma_{k-1} = \frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}.$$

Tato volba navíc dává

$$\begin{aligned} p_{k-1}^T r_k &= p_{k-1}^T (b - Ax_k) = p_{k-1}^T (b - A(x_{k-1} + \gamma_{k-1} p_{k-1})) \\ &= p_{k-1}^T (r_{k-1} - \gamma_{k-1} A p_{k-1}) = 0. \end{aligned} \quad (1.3)$$

Vektory p_{k-1} a r_k jsou tedy kolmé.

Zbývá zvolit směrové vektory p_k . Nejjednodušší volba $p_k = r_k$ pro $k = 1, 2, \dots$ odpovídá metodě největšího spádu a nezaručuje žádnou minimalizační vlastnost. Volme proto pouze $p_0 = r_0$ a v dalších krocích konstruujeme vektory následovně

$$p_k = r_k + \delta_k p_{k-1},$$

kde δ_k je zatím blíže neurčený skalár. Budeme jej volit tak, aby dva po sobě následující směrové vektory byly A -ortogonální, tzn. aby $p_{k-1}^T A p_k = 0$. Z požadavku na A -ortogonalitu vektorů p_{k-1}, p_k dostaneme pro koeficient δ_k vztah

$$\delta_k = -\frac{p_{k-1}^T A r_k}{p_{k-1}^T A p_{k-1}}.$$

Tímto je již postup jednoznačně určen.

Lze ukázat, že tato lokální A -ortogonalita již zaručuje globální A -ortogonalitu všech směrových vektorů, tzn. $p_i^T A p_j = 0, i \neq j$, viz Hestenes a Stiefel (1952, Theorem 4.1). Díky tomu je v každém kroku

$$\|x_\star - x_k\|_A = \min_{z \in x_0 + \text{span}\{p_0, \dots, p_{k-1}\}} \|x_\star - z\|_A.$$

Aproximace řešení v prostoru generovaném vektory p_0, \dots, p_{k-1} je tedy nejlepší možná ve smyslu minimální A -normy chyby $\|x_\star - x_k\|_A$. Indukcí lze navíc ukázat, že

$$\text{span}\{p_0, \dots, p_{k-1}\} = \mathcal{K}_k(A, r_0),$$

kde $\mathcal{K}_k(A, v)$ je k -tý Krylovův podprostor příslušný matici A a vektoru v , neboli $\mathcal{K}_k(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{k-1}v\}$. Minimalizační vlastnost tak můžeme psát ve tvaru

$$\|x_\star - x_k\|_A = \min_{z \in x_0 + \mathcal{K}_k(A, r_0)} \|x_\star - z\|_A. \quad (1.4)$$

1.2 CG jako projekční metoda

Projekční metody jsou iterační metody pro hledání řešení soustav lineárních algebraických rovnic (Saad (2003, kapitola 5)). Jejich princip spočívá v hledání k -té aproximace x_k v podprostoru $\mathcal{S}_k \subset \mathbb{R}^n$ dimenze k („search space“). Dále uvažujeme prostor podmínek $\mathcal{C}_k \subset \mathbb{R}^n$ („constraints space“) rovněž dimenze k a vyžadujeme ortogonalitu (ve smyslu standardního skalárního součinu) rezidua $r_k = b - Ax_k$ na \mathcal{C}_k . V metodě sdružených gradientů je

$$\mathcal{S}_k = \mathcal{C}_k = \mathcal{K}_k(A, r_0).$$

Pro A symetrickou pozitivně definitní tato volba zajišťuje existenci a jednoznačnost aproximace x_k , viz například Liesen a Strakoš (2013, Theorem 2.3.1).

V každém kroku tedy hledáme aproximaci řešení

$$x_k \in x_0 + \mathcal{K}_k(A, r_0), \quad (1.5)$$

kde $x_0 \in \mathbb{R}^n$ je počáteční aproximace, a vyžadujeme

$$r_k \perp \mathcal{K}_k(A, r_0). \quad (1.6)$$

Uvažujme ortonormální bázi $V_k = [v_1, \dots, v_k]$ prostoru $\mathcal{K}_k(A, r_0)$ získanou Lanczosovým algoritmem aplikovaným na matici A a vektor r_0 , viz například Liesen a Strakoš (2013, sekce 2.4.1). Platí

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T, \quad (1.7)$$

kde e_k je k -tý sloupec jednotkové matice řádu k a β_{k+1} je normalizační koeficient z Lanczosova algoritmu. Díky ortogonalitě sloupců matice V_k platí

$$T_k = V_k^T A V_k \in \mathbb{R}^{k \times k}.$$

Matice T_k je tedy SPD, navíc je tridiagonální a prvky β_2, \dots, β_k mimo diagonálu jsou kladné. Můžeme uvažovat rozklad $T_k = L_k D_k L_k^T$, kde

$$L_k = \begin{bmatrix} 1 & & & & \\ \ell_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & & \ell_{k-1} & 1 \end{bmatrix} \in \mathbb{R}^{k \times k}$$

a $D_k \in \mathbb{R}^{k \times k}$ je diagonální s kladnými prvky d_1, \dots, d_k .

Jelikož V_k tvoří bázi $\mathcal{K}_k(A, r_0)$, lze přibližné řešení x_k v k -tém kroku vyjádřit jako

$$x_k = x_0 + V_k y_k \quad (1.8)$$

pro vhodný vektor $y_k \in \mathbb{R}^k$. Ten dokážeme určit z podmínky (1.6)

$$\begin{aligned} 0 &= V_k^T r_k = V_k^T (b - A(x_0 + V_k y_k)) = V_k^T r_0 - V_k^T A V_k y_k \\ &= V_k^T (||r_0|| v_1) - T_k y_k = ||r_0|| e_1 - T_k y_k, \end{aligned}$$

kde $||r_0|| = \sqrt{r_0^T r_0}$ označuje standardní euklidovskou normu. Poslední rovnost vyplývá z ortonormality sloupců V_k . Výraz (1.8) tak můžeme upravit na

$$x_k = x_0 + V_k T_k^{-1} (||r_0|| e_1). \quad (1.9)$$

S využitím (1.7) a (1.9) ukážeme důležitou vlastnost reziduového vektoru r_k , a to že je násobkem bázového vektoru v_{k+1}

$$\begin{aligned} r_k &= b - A x_k = b - A(x_0 + V_k T_k^{-1} ||r_0|| e_1) \\ &= r_0 - (V_k T_k + \beta_{k+1} v_{k+1} e_k^T) T_k^{-1} ||r_0|| e_1 \\ &= r_0 - ||r_0|| v_1 - \beta_{k+1} ||r_0|| (e_k^T T_k^{-1} e_1) v_{k+1} \\ &= -(\beta_{k+1} ||r_0|| e_k^T T_k^{-1} e_1) v_{k+1}. \end{aligned} \quad (1.10)$$

Poslední rovnost platí díky tomu, že $v_1 = r_0/||r_0||$, vektor r_0 jsme zvolili jako počáteční vektor v Lanczosově algoritmu. Z (1.10) vyplývá, že $\text{span}\{r_0, \dots, r_k\} = \text{span}\{v_1, \dots, v_{k+1}\}$ a vektory r_0, \dots, r_k tedy tvoří ortogonální bázi Krylovova prostoru $\mathcal{K}_{k+1}(A, r_0)$.

Nyní budeme pokračovat v úpravách vztahu (1.9) a ukážeme, že posloupnost aproximací řešení je možné konstruovat rekurentně. Definujme nejprve matici

$$\widehat{P}_k = [\widehat{p}_0, \dots, \widehat{p}_{k-1}] := V_k L_k^{-T} \in \mathbb{R}^{n \times k}.$$

Pak můžeme psát

$$V_k = \widehat{P}_k L_k^T = [\widehat{p}_0, \dots, \widehat{p}_{k-1}] \begin{bmatrix} 1 & \ell_1 & & \\ & \ddots & \ddots & \\ & & 1 & \ell_{k-1} \\ & & & 1 \end{bmatrix}.$$

Díky bidiagonální struktuře matice L_k^T platí $v_{j+1} = \widehat{p}_j + \ell_j \widehat{p}_{j-1}$, respektive

$$\widehat{p}_j = v_{j+1} - \ell_j \widehat{p}_{j-1}, \quad j = 0, 1, \dots, k-1, \quad (1.11)$$

kde uvažujeme $\widehat{p}_{-1} = 0$ a $\ell_0 = 0$. Získali jsme tak rekurentní vztah pro výpočet vektorů \widehat{p}_j . Přímou z definice matice \widehat{P}_k dostáváme

$$\widehat{P}_k^T A \widehat{P}_k = L_k^{-1} V_k^T A V_k L_k^{-T} = L_k^{-1} T_k L_k^{-T} = D_k,$$

vektory \widehat{p}_j jsou tedy A -ortogonální. Dále z jejich konstrukce vyplývá, že

$$\text{span}\{\widehat{p}_0, \dots, \widehat{p}_{k-1}\} = \text{span}\{v_1, \dots, v_k\} = \mathcal{K}_k(A, r_0).$$

Upravme nyní vztah (1.9) pro aproximaci řešení

$$\begin{aligned} x_k &= x_0 + V_k T_k^{-1} (||r_0|| e_1) = x_0 + (V_k L_k^{-T}) (||r_0|| D_k^{-1} L_k^{-1} e_1) \\ &= x_0 + \widehat{P}_k (||r_0|| D_k^{-1} L_k^{-1} e_1) = x_0 + \widehat{P}_k \widehat{c}_k, \end{aligned} \quad (1.12)$$

kde jsme označili

$$\widehat{c}_k = \begin{bmatrix} c_k^{(1)} \\ \vdots \\ c_k^{(k)} \end{bmatrix} := ||r_0|| D_k^{-1} L_k^{-1} e_1 \in \mathbb{R}^k.$$

Vektor \widehat{c}_k získáme řešením soustavy $L_k D_k \widehat{c}_k = ||r_0|| e_1$. Rozepišme nejprve podrobněji součin matic $L_k D_k$

$$L_k D_k = \left[\begin{array}{cc|c} 1 & & \\ \ell_1 & 1 & \\ & \ddots & \ddots \\ & & \ell_{k-1} & 1 \end{array} \right] \left[\begin{array}{c|c} d_1 & \\ \vdots & \\ d_{k-1} & \\ \hline & d_k \end{array} \right] = \left[\begin{array}{c|c} L_{k-1} D_{k-1} & 0 \\ \hline \ell_{k-1} d_{k-1} e_{k-1}^T & d_k \end{array} \right].$$

Soustava má tedy tvar

$$\left[\begin{array}{c|c} L_{k-1} D_{k-1} & 0 \\ \hline \ell_{k-1} d_{k-1} e_{k-1}^T & d_k \end{array} \right] \begin{bmatrix} c_k^{(1)} \\ \vdots \\ c_k^{(k-1)} \\ \hline c_k^{(k)} \end{bmatrix} = ||r_0|| e_1.$$

Z této struktury je patrné, že pro vektor \widehat{c}_k platí

$$\widehat{c}_k = \begin{bmatrix} \widehat{c}_{k-1} \\ c_k^{(k)} \end{bmatrix}, \quad (1.13)$$

přičemž poslední prvek $c_k^{(k)}$ snadno určíme z rovnosti

$$c_{k-1}^{(k-1)} \ell_{k-1} d_{k-1} + c_k^{(k)} d_k = 0.$$

S použitím (1.13) dále upravíme vztah (1.12) pro x_k

$$\begin{aligned} x_k &= x_0 + \widehat{P}_k \widehat{c}_k = x_0 + \widehat{P}_{k-1} \widehat{c}_{k-1} + c_k^{(k)} \widehat{p}_{k-1} \\ &= x_{k-1} + c_k^{(k)} \widehat{p}_{k-1}. \end{aligned} \quad (1.14)$$

Novou aproximaci x_k tedy získáme z předchozí x_{k-1} přičtením vektoru $c_k^{(k)} \widehat{p}_{k-1}$. Můžeme ještě pokračovat v úpravách. Z (1.11) vyplývá, že $\widehat{p}_j \in \text{span}\{v_{j+1}, \widehat{p}_{j-1}\}$ a díky (1.10)

$$\widehat{p}_j \in \text{span}\{r_j, \widehat{p}_{j-1}\}$$

Stejně jako v předchozí sekci dále uvažujme posloupnost vektorů p_j ,

$$p_j = r_j + \delta_j p_{j-1}, \quad (1.15)$$

kde $p_0 := r_0$ a vyžadujeme, aby vektory p_j a p_{j-1} byly A -ortogonální. To je splněno pro

$$\delta_j = -\frac{p_{j-1}^T A r_j}{p_{j-1}^T A p_{j-1}}.$$

Takto získané vektory p_j jsou skalárními násobky příslušných vektorů \widehat{p}_j . To lze dokázat indukcí: Pro $j = 0$ je tvrzení zřejmé díky (1.10). Vektory p_1 a \widehat{p}_1 jsou lineárními kombinacemi p_0 a r_1 . Navíc oba jsou A -ortogonální k vektoru p_0 . Musí tedy být násobkem jeden druhého. Stejným argumentem dokážeme i indukční krok pro obecné j . Předpokládejme, že tvrzení platí pro p_0, \dots, p_{j-1} . Vektory p_j a \widehat{p}_j jsou lineárními kombinacemi p_{j-1} a r_j a jsou A -ortogonální k p_{j-1} . Vektor p_j tak musí být násobkem \widehat{p}_j a tvrzení platí pro obecné j .

Vztah (1.14) tak přechází do tvaru

$$x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$$

pro vhodný koeficient α_{k-1} a pro reziduum r_k platí

$$r_k = b - A(x_{k-1} + \alpha_{k-1} p_{k-1}) = r_{k-1} - \alpha_{k-1} A p_{k-1}. \quad (1.16)$$

Koeficient α_{k-1} určíme z ortogonalit reziduí,

$$r_{k-1}^T r_k = r_{k-1}^T (r_{k-1} - \alpha_{k-1} A p_{k-1}) = 0$$

je splněno pro

$$\alpha_{k-1} = \frac{r_{k-1}^T r_{k-1}}{r_{k-1}^T A p_{k-1}}.$$

Ačkoli jsme postup konstrukce aproximací řešení odvodili s použitím rozkladu $T_k = L_k D_k L_k^T$, dokázali jsme jej díky přeškálování vektorů \widehat{p}_j modifikovat tak, že rozklad nepotřebujeme explicitně znát. Stačí nám počáteční odhad řešení x_0 a všechny vektory potřebné v dalších krocích jsou generovány rekurentně.

1.3 Algoritmus

Na závěr kapitoly uvedeme standardní implementaci metody sdružených gradientů, která je navržena i v původním článku Hestenes a Stiefel (1952). Nejprve provedeme úpravy koeficientů. Ukážeme, že koeficienty α_{k-1} a γ_{k-1} z odvození v sekcích 1.1 a 1.2 jsou stejné. Dále vyjádříme γ_{k-1} a δ_k ve tvaru vhodném pro implementaci.

Z (1.15) vyplývá, že $r_{k-1} = p_{k-1} - \delta_{k-1}p_{k-2}$. Potom

$$\alpha_{k-1} = \frac{r_{k-1}^T r_{k-1}}{r_{k-1}^T A p_{k-1}} = \frac{\|r_{k-1}\|^2}{(p_{k-1} - \delta_{k-1}p_{k-2})^T A p_{k-1}} = \frac{\|r_{k-1}\|^2}{p_{k-1}^T A p_{k-1}},$$

kde poslední rovnost vyplývá z A -ortogonalit p_{k-1} a p_{k-2} . Dále

$$\gamma_{k-1} = \frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}} = \frac{(r_{k-1} + \delta_{k-1}p_{k-2})^T r_{k-1}}{p_{k-1}^T A p_{k-1}} = \frac{\|r_{k-1}\|^2}{p_{k-1}^T A p_{k-1}}.$$

Poslední rovnost plyne z (1.3). Platí tedy, že $\alpha_{k-1} = \gamma_{k-1}$. Pro úpravu koeficientu δ_k nejprve přepíšeme vztah (1.16) do tvaru

$$-A p_{k-1} = \frac{1}{\alpha_{k-1}}(r_k - r_{k-1}).$$

S využitím ortogonalit reziduí dostáváme

$$\delta_k = -\frac{p_{k-1}^T A r_k}{p_{k-1}^T A p_{k-1}} = -\frac{(A p_{k-1})^T r_k}{p_{k-1}^T A p_{k-1}} = \frac{1}{\alpha_{k-1}} \frac{(r_k - r_{k-1})^T r_k}{p_{k-1}^T A p_{k-1}} = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}.$$

Nyní můžeme zapsat výsledný algoritmus.

Algoritmus 1 Metoda sdružených gradientů

Vstup: SPD matice $A \in \mathbb{R}^{n \times n}$, pravá strana $b \in \mathbb{R}^n$, počáteční odhad řešení $x_0 \in \mathbb{R}^n$

Výstup: aproximace řešení x_k

```

 $r_0 = b - Ax_0$ 
 $p_0 = r_0$ 
for  $k = 1, 2, \dots$  do
   $\gamma_{k-1} = \frac{\|r_{k-1}\|^2}{p_{k-1}^T A p_{k-1}}$ 
   $x_k = x_{k-1} + \gamma_{k-1} p_{k-1}$ 
   $r_k = r_{k-1} - \gamma_{k-1} A p_{k-1}$ 
   $\delta_k = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}$ 
   $p_k = r_k + \delta_k p_{k-1}$ 
end for

```

V přesné aritmetice Algoritmus 1 vždy nalezne přesné řešení v konečném počtu kroků. K zastavení dojde, pokud $x_k = x_{k-1}$. To nastane, pokud $\gamma_{k-1} = 0$ nebo $p_{k-1} = 0$. V prvním případě ze vztahu pro γ_{k-1} okamžitě dostáváme, že $r_{k-1} = 0$, a tedy $Ax_{k-1} = b$. Dále ukážeme, že $p_{k-1} = 0$ pouze pokud $r_{k-1} = 0$. Předpokládejme, že $r_{k-1} \neq 0$. Z (1.3) vyplývá, že $r_{k-1} \perp p_{k-2}$, a tedy vektory r_{k-1} a p_{k-2} jsou lineárně nezávislé. Z tohoto faktu a konstrukce p_{k-1} již plyne, že

rovněž $p_{k-1} \neq 0$. V obou uvažovaných případech zastavení algoritmu dostáváme, že $r_{k-1} = 0$. Algoritmus se tak zastaví až s přesným řešením.

V praxi obvykle není potřeba znát přesné řešení, stačí dostatečně přesná aproximace. Zajímá nás tedy rychlost konvergence metody. Připomeňme minimalizační vlastnost (1.4)

$$\|x_\star - x_k\|_A = \min_{z \in x_0 + \mathcal{K}_k(A, r_0)} \|x_\star - z\|_A.$$

Libovolný vektor $v \in \mathcal{K}_k(A, r_0)$ lze z definice Krylova prostoru vyjádřit ve tvaru $v = a_0 r_0 + a_1 A r_0 + \dots + a_{k-1} A^{k-1} r_0$ pro vhodné koeficienty a_0, a_1, \dots, a_{k-1} , neboli $v = s(A)r_0$, kde s je polynom stupně nejvýše $k-1$. Můžeme tedy psát (viz (1.5))

$$x_k = x_0 + q(A)r_0$$

pro vhodný polynom q stupně nejvýše $k-1$. Podobně můžeme vyjádřit i chybu v k -té aproximaci

$$\begin{aligned} \varepsilon_k &= x_\star - x_k = x_\star - x_0 - (x_k - x_0) = \varepsilon_0 - q(A)r_0 = \varepsilon_0 - q(A)A(x_\star - x_0) \\ &= (I_n - q(A)A)\varepsilon_0 = p(A)\varepsilon_0, \end{aligned}$$

kde I_n je jednotková matice řádu n a p je polynom stupně nejvýše k , pro který platí $p(0) = 1$. Jak jsme ukázali v (1.4), A -norma chyby $\|\varepsilon_k\|_A$ je minimální vzhledem k množině $x_0 + \mathcal{K}_k(A, r_0)$, musí proto platit

$$\|\varepsilon_k\|_A = \min_{p \in \Pi_k} \|p(A)\varepsilon_0\|_A,$$

kde Π_k označuje množinu polynomů stupně nejvýše k , které v nule nabývají hodnoty 1. Platí

$$\|p(A)\varepsilon_0\|_A = \|p(A)A^{1/2}\varepsilon_0\| \leq \|p(A)\| \|A^{1/2}\varepsilon_0\| = \|p(A)\| \|\varepsilon_0\|_A,$$

kde $\|p(A)\|$ označuje spektrální normu matice, a tedy

$$\frac{\|\varepsilon_k\|_A}{\|\varepsilon_0\|_A} \leq \min_{p \in \Pi_k} \|p(A)\|. \quad (1.17)$$

Jelikož matice A je SPD, můžeme uvažovat její spektrální rozklad $A = Q\Lambda Q^T$, kde $\Lambda \in \mathbb{R}^{n \times n}$ je diagonální s vlastními čísly matice A na diagonále, $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, a $Q \in \mathbb{R}^{n \times n}$ je ortogonální, tedy $QQ^T = Q^TQ = I_n$. Potom

$$\|p(A)\| = \|p(Q\Lambda Q^T)\| = \|Qp(\Lambda)Q^T\| = \|p(\Lambda)\| = \max_{i=1, \dots, n} |p(\lambda_i)|. \quad (1.18)$$

Dosazením (1.18) do (1.17) dostáváme odhad

$$\frac{\|\varepsilon_k\|_A}{\|\varepsilon_0\|_A} \leq \min_{p \in \Pi_k} \max_{i=1, \dots, n} |p(\lambda_i)| \leq \min_{p \in \Pi_k} \max_{\lambda \in [\lambda_1, \lambda_n]} |p(\lambda)|.$$

V poslední nerovnosti jsme přešli od diskrétního spektra $\lambda_1 \leq \dots \leq \lambda_n$ k intervalu $[\lambda_1, \lambda_n]$. S využitím vlastností posunutých a škálovaných Čebyševových

polynomů získáme odhad ve tvaru (pro podrobnější postup a důkaz viz například Greenbaum (1997, Theorem 3.1.1))

$$\frac{\|\varepsilon_k\|_A}{\|\varepsilon_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k, \quad (1.19)$$

kde $\kappa(A) = \|A\| \|A^{-1}\| = \lambda_n/\lambda_1$ označuje číslo podmíněnosti matice A .

Pro dobře podmíněné matice tak máme zaručenu rychlou konvergenci metody. To ovšem neznamená, že pro špatně podmíněné matice bude vždy metoda konvergovat pomalu. Odhad (1.19) totiž nebere v potaz rozložení vlastních čísel uvnitř intervalu, na kterém rychlost konvergence rovněž závisí (viz například Liesen a Strakoš (2013, sekce 5.6.3)), a bývá často velmi nadhodnocený. Zároveň nižší číslo podmíněnosti nutně nezaručuje rychlejší konvergenci.

2. Transformace úlohy: předpodmínění a deflace

V řadě úloh metoda sdružených gradientů konverguje velmi pomalu či téměř stagnuje. V této kapitole se budeme zabývat dvěma technikami, předpodmíněním a deflací, které transformují původní úlohu s cílem konvergenci CG urychlit.

2.1 Předpodmínění

Předpodmínění je jednou ze standardních a často využívaných technik pro urychlení konvergence iteračních metod. Její myšlenkou je transformace původní úlohy $Ax = b$ do nové soustavy $\hat{A}\hat{x} = \hat{b}$, která je lépe a rychleji řešitelná. Obvykle je cílem, aby matice \hat{A} byla lépe podmíněná nebo měla vhodnější spektrum než původní matice A . Pro regulární matici $L \in \mathbb{R}^{n \times n}$ můžeme soustavu (1.1) upravit do tvaru

$$(L^{-1}AL^{-T})(L^T x) = L^{-1}b. \quad (2.1)$$

Označme $\hat{A} := L^{-1}AL^{-T}$, $\hat{x} := L^T x$ a $\hat{b} := L^{-1}b$ a přepíšme (2.1) do tvaru

$$\hat{A}\hat{x} = \hat{b}. \quad (2.2)$$

Matice \hat{A} je SPD a na soustavu (2.2) tak můžeme aplikovat CG. V k -tém kroku získáme aproximace \hat{x}_k a rezidua \hat{r}_k příslušné modifikované soustavě (2.2). Nás však zajímají vektory vztahující se k původní soustavě $Ax = b$, a proto nyní odvodíme algoritmus *předpodmíněných sdružených gradientů* (PCG) počítající přímo aproximace x_k přesného řešení x_* . Pro přesná řešení x_* a \hat{x}_* původní a modifikované soustavy platí $x_* = L^{-T}\hat{x}_*$. Položme $x_k := L^{-T}\hat{x}_k$, čímž získáme aproximace řešení x_* , a $r_k := L\hat{r}_k$. Platí

$$r_k = L\hat{r}_k = L(\hat{b} - \hat{A}\hat{x}_k) = L(L^{-1}b - L^{-1}AL^{-T}(L^T x_k)) = b - Ax_k,$$

a tedy r_k je tedy skutečně reziduum příslušné x_k a vztahující se k původní soustavě. Nakonec uvažujme vektory $p_k := L^{-T}\hat{p}_k$, kde \hat{p}_k je směrový vektor v CG pro soustavu (2.2), a $z_k := L^{-T}L^{-1}r_k$. Pomocí těchto vektorů nyní zapíšeme algoritmus PCG ekvivalentní standardní metodě CG aplikované na (2.1).

Algoritmus 2 Předpodmíněná metoda sdružených gradientů (PCG)

Vstup: SPD matice $A \in \mathbb{R}^{n \times n}$, pravá strana $b \in \mathbb{R}^n$, počáteční odhad řešení $x_0 \in \mathbb{R}^n$, regulární matice $L \in \mathbb{R}^{n \times n}$

Výstup: aproximace řešení x_k

```
r0 = b - Ax0
z0 = L-TL-1r0
p0 = z0
for k = 1, 2, ... do
     $\hat{\gamma}_{k-1} = \frac{z_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$ 
    xk = xk-1 +  $\hat{\gamma}_{k-1} p_{k-1}$ 
    rk = rk-1 -  $\hat{\gamma}_{k-1} A p_{k-1}$ 
    zk = L-TL-1rk
     $\hat{\delta}_k = \frac{z_k^T r_k}{z_{k-1}^T r_{k-1}}$ 
    pk = zk +  $\hat{\delta}_k p_{k-1}$ 
end for
```

Matice L musí být volena tak, aby CG pro (2.1) skutečně konvergovala rychleji a aby efekt předpodmínění nebyl vykoupěn výrazně vyššími výpočetními náklady například na řešení soustav $z_k = L^{-T}L^{-1}r_k$. Často se pro předpodmínění používá neúplný Choleského rozklad matice A dávající řídkou, regulární, dolní trojúhelníkovou matici $L \in \mathbb{R}^{n \times n}$, která je ideálně blízka (přesnému) Choleského faktoru \tilde{L} matice A ve smyslu $LL^T \approx \tilde{L}\tilde{L}^T = A$.

2.2 Deflace

Další technikou používanou pro urychlení konvergence CG je takzvaná deflace. Principem deflace v tomto kontextu je konstrukce vhodného podprostoru malé dimenze m a řešení úlohy metodou sdružených gradientů na podprostoru dimenze $n - m$. Výsledný algoritmus bývá v literatuře označován jako deflated CG.

V této sekci nejprve okomentujeme vybrané články, které se deflated CG zabývají, konkrétně Nicolaidese (1987), Dostál (1988) a Saad a kol. (2000). Představíme kontext uvedených článků a krátce popíšeme způsoby odvození algoritmu v jednotlivých člancích. Dále podrobně odvodíme algoritmus deflated CG s využitím tzv. sdružených projektorů a ukážeme jeho vlastnosti.

2.2.1 Deflated CG v literatuře

Poprvé byl algoritmus navržen v Nicolaidese (1987). Při odvození je uvažována deflace rezidua, kdy ve standardním algoritmu CG je reziduum r_k nahrazeno vektorem $r_k - Ec_k$, kde $E \in \mathbb{R}^{n \times m}$, $m < n$, je pevně zvolená matice s lineárně nezávislými sloupci a $c_k \in \mathbb{R}^m$ je vektor minimalizující A -normu $\|r_k - Ec_k\|_A$. V našem značení odpovídá $\text{Range}(E)$ podprostoru \mathcal{U} .

Algoritmus je aplikován na řešení soustavy vzniklé diskretizací Poissonovy úlohy $-\Delta u = f$ s okrajovými podmínkami na oblasti $\Omega \subset \mathbb{R}^2$. Sloupce matice E jsou pak vektory složené z jedniček a nul odpovídající rozkladu jednotky určenému

rozkladem oblasti Ω na disjunktní množiny $\Omega_1, \Omega_2, \dots, \Omega_i$. Počet těchto množin je roven počtu sloupců matice E , a tedy i dimenzi podprostoru \mathcal{U} .

Krátce po Nicolaides (1987) byl matematicky ekvivalentní algoritmus publikován v Dostál (1988). V tomto článku jsou ukázány vlastnosti deflated CG a jeho použití pro řešení úlohy linearizované elasticity ve 2D. Bázové vektory podprostoru \mathcal{U} agregují některé neznámé a obsahují, podobně jako v Nicolaides (1987), jedničky a nuly. K volbě podprostoru \mathcal{U} autor poznamenává: „Techniku deflace lze snadno zobecnit tak, aby zahrnovala jakoukoli další informaci o řešení soustavy (1.1) a vlastních vektorech matice A . V praxi tyto informace získáme ze zkušenosti, analýzy modelových problémů nebo fyzikální podstaty problému.“ (Dostál (1988, sekce 3.A.i), str. 317))

Odlišný postup odvození algoritmu je uveden v Saad a kol. (2000). Zde je nejprve uveden deflated Lanczos algorithm. Jeho princip spočívá v nahrazení matice A v klasickém Lanczosově algoritmu maticí $B = A - AW(W^T AW)^{-1}W^T A$, kde $W \in \mathbb{R}^{n \times m}$ je daná matice s lineárně nezávislými sloupci. Tím je konstruována ortonormální báze $V_j = \{v_1, \dots, v_j\}$ prostoru $\mathcal{K}_j(B, v_1)$ taková, že $v_i^T W = 0$, $i = 1, \dots, j$. Z deflated Lanczos je poté, podobně jako jsme to udělali my v sekci 1.2, odvozen deflated CG minimalizující A -normu chyby na množině $x_0 + \text{span}\{W, V_j\}$. V článku jsou prezentovány výsledky numerických experimentů, ve kterých je uvažována posloupnost úloh se stejnou maticí A a měnící se pravou stranou $b^{(i)}$. Z podprostorů budovaných při řešení jednotlivých soustav jsou extrahovány aproximace vlastních vektorů příslušných nejmenším vlastním číslům matice A . Ty jsou poté použity pro deflaci při řešení systému s novou pravou stranou.

2.2.2 Odvození algoritmu

Při odvozování algoritmu deflated CG využijeme vlastností tzv. sdružených projektorů (anglicky conjugate projector). Matici $P \in \mathbb{R}^{n \times n}$ nazveme sdruženým projektorem na podprostor $\mathcal{P} \subset \mathbb{R}^n$, pokud $\mathcal{P} = \text{Range}(P)$, $P^2 = P$ a $P^T A(I_n - P) = 0$. Z poslední vlastnosti zároveň plyne, že $P^T AP = AP$ a P^T je tedy projektor na podprostor $A\mathcal{P}$.

S využitím uvedených vlastností nyní odvodíme algoritmus deflated CG. Uvažujme matici $U \in \mathbb{R}^{n \times m}$, $m < n$, s lineárně nezávislými sloupci. Označme $\mathcal{U} := \text{Range}(U)$. Dále definujme matici P

$$P := U(U^T AU)^{-1}U^T A \in \mathbb{R}^{n \times n}.$$

Přímým výpočtem s využitím symetrie matice A lze ukázat, že $P^2 = P$ a $P^T AP = AP$ a P je tedy sdružený projektor na podprostor \mathcal{U} . Dále označme $Q := I_n - P$ a $\mathcal{V} := \text{Range}(Q)$. Matice Q je sdružený projektor na \mathcal{V} , jelikož

$$\begin{aligned} Q^2 &= (I_n - P)^2 = I_n - 2P + P^2 = I_n - P = Q, \\ Q^T A(I_n - Q) &= (I_n - P^T)AP = AP - P^T AP = 0. \end{aligned} \quad (2.3)$$

Zároveň platí $\mathcal{U} + \mathcal{V} = \mathbb{R}^n$ a $\mathcal{U} \cap \mathcal{V} = \emptyset$. Přesné řešení x_* tak můžeme rozložit na složky odpovídající projekcím maticemi P a Q , neboli

$$x_* = x_*|_{\mathcal{V}} + x_*|_{\mathcal{U}} = Qx_* + Px_*.$$

Můžeme tedy psát

$$Ax_\star = A(Px_\star + Qx_\star) = b$$

a jednoduchou úpravou dostáváme

$$AQx_\star = b - APx_\star. \quad (2.4)$$

Poznamenejme, že vektor Px_\star můžeme spočítat i přesto, že přesné řešení x_\star neznáme. Z definice matice P a s využitím $x_\star = A^{-1}b$ totiž dostáváme

$$Px_\star = U(U^T AU)^{-1}U^T Ax_\star = U(U^T AU)^{-1}U^T b.$$

Vztah (2.4) tak můžeme upravit do tvaru

$$AQx_\star = b - AU(U^T AU)^{-1}U^T b = (I_n - P^T)b = Q^T b = \tilde{b},$$

kde jsme označili $\tilde{b} := Q^T b = b - APx_\star$.

Uvažujme nyní soustavu

$$AQ\tilde{x} = \tilde{b} \quad (2.5)$$

s neznámou \tilde{x} . Jelikož $\tilde{b} = Q^T b \in A\mathcal{V}$, je tato soustava kompatibilní a má řešení. Z (2.3) plyne, že $AQ = Q^T AQ$ a tedy matice AQ je symetrická. Na prostoru $A\mathcal{V}$ je navíc pozitivně definitní. K důkazu budeme potřebovat následující pozorování. Je-li Q sdružený projektor na prostor \mathcal{V} a $v \in A\mathcal{V}$, potom

$$\|Qv\| \geq \|v\|. \quad (2.6)$$

To dokážeme s využitím vztahů $Q^T v = v$ a $v^T v = (Q^T v)^T v = v^T Qv = v^T (Q^T v)$. Dostáváme

$$\begin{aligned} \|Qv\|^2 &= v^T Q^T Qv = v^T ((Q^T - I_n) + I_n)((Q - I_n) + I_n)v \\ &= \|(Q - I_n)v\|^2 + v^T (Q^T - I_n)v + v^T (Q - I_n)v + \|v\|^2 \\ &= \|(Q - I_n)v\|^2 + v^T Q^T v - v^T v + v^T Qv - v^T v + \|v\|^2 \\ &= \|(Q - I_n)v\|^2 + \|v\|^2, \end{aligned}$$

z čehož již vyplývá nerovnost (2.6).

Pro $v \in A\mathcal{V}$, $v \neq 0$ tedy platí

$$v^T AQv = v^T Q^T AQv = z^T Az > 0, \quad (2.7)$$

kde jsme označili $z := Qv$ a díky (2.6) platí, že pokud $v \neq 0$, pak i $z \neq 0$. Poslední nerovnost pak plyne z pozitivní definitnosti matice A . Ukázali jsme, že matice AQ je na prostoru $A\mathcal{V}$ symetrická pozitivně definitní. Soustava (2.5) má tedy na $A\mathcal{V}$ jednoznačné řešení, které označíme \tilde{x}_\star .

Na úlohu (2.5) nyní aplikujme standardní algoritmus CG a jako počáteční aproximaci zvolme $\tilde{x}_0 = 0$.

Algoritmus 3 CG aplikované na úlohu $AQ\tilde{x} = \tilde{b} = b - APx_*$

$$\begin{aligned} \tilde{r}_0 &= \tilde{b} \\ \tilde{p}_0 &= \tilde{r}_0 \\ \text{for } k &= 1, 2, \dots \text{ do} \\ \tilde{\gamma}_{k-1} &= \frac{\|\tilde{r}_{k-1}\|^2}{\tilde{p}_{k-1}^T A Q \tilde{p}_{k-1}} \\ \tilde{x}_k &= \tilde{x}_{k-1} + \tilde{\gamma}_{k-1} \tilde{p}_{k-1} \\ \tilde{r}_k &= \tilde{r}_{k-1} - \tilde{\gamma}_{k-1} A Q \tilde{p}_{k-1} \\ \tilde{\delta}_k &= \frac{\|\tilde{r}_k\|^2}{\|\tilde{r}_{k-1}\|^2} \\ \tilde{p}_k &= \tilde{r}_k + \tilde{\delta}_k \tilde{p}_{k-1} \\ \text{end for} \end{aligned}$$

Analogie minimalizační vlastnosti (1.4) pro aproximace \tilde{x}_k generované Algoritmem 3 je

$$\|Q(\tilde{x}_* - \tilde{x}_k)\|_A = \min_{\tilde{z} \in \tilde{x}_0 + \mathcal{K}_k(Q^T A Q, \tilde{r}_0)} \|Q(\tilde{x}_* - \tilde{z})\|_A. \quad (2.8)$$

Jelikož je matice $Q^T A Q$ singulární, místo „ $\|v\|_{Q^T A Q}$ “ píšeme $\|Qv\|_A$.

Nyní budeme chtít algoritmus upravit tak, abychom dostali aproximace x_k příslušné původní soustavě $Ax = b$. Platí

$$b = \tilde{b} + APx_* = AQ\tilde{x}_* + APx_* = A(Q\tilde{x}_* + Px_*).$$

Aproximace x_k proto budeme konstruovat následovně

$$x_k := Q\tilde{x}_k + Px_*.$$

Potom pro reziduum r_k příslušné x_k vztahující se k soustavě $Ax = b$ platí

$$r_k = b - Ax_k = b - APx_* - AQ\tilde{x}_k = \tilde{b} - AQ\tilde{x}_k = \tilde{r}_k. \quad (2.9)$$

Dále s využitím rekurence pro \tilde{x}_k dostáváme

$$x_k = Px_* + Q\tilde{x}_{k-1} + \tilde{\gamma}_{k-1} Q\tilde{p}_{k-1} = x_{k-1} + \tilde{\gamma}_{k-1} Q\tilde{p}_{k-1}.$$

Položme $p_k := Q\tilde{p}_k$ a upravme vztah pro koeficient $\tilde{\gamma}_{k-1}$

$$\tilde{\gamma}_{k-1} = \frac{\|\tilde{r}_{k-1}\|^2}{\tilde{p}_{k-1}^T A Q \tilde{p}_{k-1}} = \frac{\|r_{k-1}\|^2}{\tilde{p}_{k-1}^T Q^T A Q \tilde{p}_{k-1}} = \frac{\|r_{k-1}\|^2}{p_{k-1}^T A p_{k-1}}.$$

Zbývá diskutovat volbu počáteční aproximace x_0 . Potřebujeme zaručit, že počáteční reziduum r_0 leží v podprostoru \mathcal{AV} . Výše zmíněné volbě $\tilde{x}_0 = 0$ odpovídá $x_0 = Px_* = U(U^T A U)^{-1} U^T b$ a $r_0 = b - APx_* = \tilde{b} \in \mathcal{AV}$. Poznamenejme, že se jedná o počáteční aproximaci uvažovanou v Dostál (1988). Obecně můžeme uvažovat x_0 ve tvaru (viz Saad a kol. (2000, vztah 3.12))

$$\begin{aligned} x_0 &= x_{-1} + U(U^T A U)^{-1} U^T (b - Ax_{-1}) = (I_n - U(U^T A U)^{-1} U^T A) x_{-1} + Px_* \\ &= Qx_{-1} + Px_*, \end{aligned} \quad (2.10)$$

kde $x_{-1} \in \mathbb{R}^n$ je libovolný vektor. První člen v posledním součtu odpovídá volbě $\tilde{x}_0 \neq 0$ v Algoritmu 3.

S využitím výše zavedených vektorů a odvozených vztahů zapíšeme výsledný algoritmus deflated CG pro soustavu $Ax = b$.

Algoritmus 4 Deflated CG

Vstup: SPD matice $A \in \mathbb{R}^{n \times n}$, pravá strana $b \in \mathbb{R}^n$, matice $U \in \mathbb{R}^{n \times m}$, $m < n$ s lineárně nezávislými sloupci, počáteční odhad řešení $x_{-1} \in \mathbb{R}^n$

Výstup: aproximace řešení x_k

$$x_0 = x_{-1} + U(U^T A U)^{-1} U^T (b - A x_{-1})$$

$$r_0 = b - A x_0$$

$$\tilde{p}_0 = r_0$$

$$Q = I_n - U(U^T A U)^{-1} U^T A$$

for $k = 1, 2, \dots$ **do**

$$p_{k-1} = Q \tilde{p}_{k-1}$$

$$\gamma_{k-1} = \frac{\|r_{k-1}\|^2}{p_{k-1}^T A p_{k-1}}$$

$$x_k = x_{k-1} + \gamma_{k-1} p_{k-1}$$

$$r_k = r_{k-1} - \gamma_{k-1} A p_{k-1}$$

$$\delta_k = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}$$

$$\tilde{p}_k = r_k + \delta_k \tilde{p}_{k-1}$$

end for

2.2.3 Vlastnosti deflated CG

Algoritmus 4 je ekvivalentní algoritmu uvedenému v (Dostál, 1988, sekce 3). Liší se pouze ve způsobu výpočtu koeficientů námi označených jako γ_k, δ_k . Ekvivalenci uvedených tvarů koeficientů jsme však ukázali v úvodu sekce 1.3. Stejně tak je námi odvozený algoritmus ekvivalentní Saad a kol. (2000, Algorithm 3.5), kde nejsou generovány vektory \tilde{p}_k , ale rovnou jsou aktualizovány vektory p_k .

Z konstrukce vyplývá, že algoritmus deflated CG je dobře definován, viz rovněž Dostál (1988, Theorem 1) a Saad a kol. (2000, Theorem 4.2). Dále pro aproximace řešení x_k platí minimalizační vlastnost podobná (1.4),

$$\|x_\star - x_k\|_A = \min_{z \in x_0 + \mathcal{K}_k(QAQ, Qr_0) + \mathcal{U}} \|x_\star - z\|_A,$$

kterou nyní dokážeme. Podoba prostoru vyplývá ze vztahu mezi aproximacemi x_k a \tilde{x}_k . Aproximace \tilde{x}_k leží v prostoru $\tilde{x}_0 + \mathcal{K}_k(AQ, \tilde{r}_0)$. Z definice $x_k = Q\tilde{x}_k + Px_\star$ a s využitím $Q^2 = Q$, $r_0 = \tilde{r}_0$ a $\text{Range}(P) = \mathcal{U}$ pak dostáváme $x_k \in x_0 + \mathcal{K}_k(QAQ, Qr_0) + \mathcal{U}$. Dále platí

$$\|x_\star - x_k\|_A = \|x_\star - Px_\star - Q\tilde{x}_k\|_A = \|Qx_\star - Q\tilde{x}_k\|_A = \|Q\tilde{x}_\star - Q\tilde{x}_k\|_A, \quad (2.11)$$

kde jsme v poslední rovnosti využili, že $Qx_\star = Q\tilde{x}_\star$, což plyne z (2.4), (2.5) a regularity matice A . Uvažujme nyní libovolný vektor $z \in x_0 + \mathcal{K}_k(QAQ, Qr_0) + \mathcal{U}$. Protože $x_0 = Qx_{-1} + Px_\star$, viz (2.10), kde $x_{-1} = \tilde{x}_0$ je (nenulová) počáteční aproximace v Algoritmu 3, můžeme vektor z uvažovat ve tvaru $z = Q\tilde{z} + u$, kde $u \in \mathcal{U}$ a $\tilde{z} \in \tilde{x}_0 + \mathcal{K}_k(AQ, \tilde{r}_0)$. S využitím vlastností sdružených projektorů P a Q , zejména $P^T A Q = 0$,

$$\begin{aligned} \|x_\star - z\|_A^2 &= \|P(x_\star - z)\|_A^2 + \|Q(x_\star - z)\|_A^2 \\ &= \|Px_\star - u\|_A^2 + \|Qx_\star - Q\tilde{z}\|_A^2. \end{aligned} \quad (2.12)$$

Člen $\|Px_\star - u\|_A^2$ v předchozím součtu je větší nebo roven nule a je nulový právě tehdy, když $u = Px_\star$. Z minimalizační vlastnosti (2.8) pak pro každé $\tilde{z} \in \tilde{x}_0 + \mathcal{K}_k(AQ, \tilde{r}_0)$ platí

$$\|Qx_\star - Q\tilde{z}\|_A^2 \geq \|Q\tilde{x}_\star - Q\tilde{x}_k\|_A^2, \quad (2.13)$$

kde jsme opět využili $Qx_\star = Q\tilde{x}_\star$. Celkově tak s využitím (2.11), (2.12) a (2.13) dostáváme

$$\|x_\star - z\|_A \geq \|x_\star - x_k\|_A$$

pro libovolné $z \in x_0 + \mathcal{K}_k(QAQ, Qr_0) + \mathcal{U}$.

Nyní budeme diskutovat konvergenční vlastnosti deflated CG. Jelikož platí (2.9), Algoritmy 3 a 4 generují stejná rezidua. Rovnost (2.11) pak popisuje vztah mezi A -normami chyb v Algoritmech 3 a 4. Pro odhad A -normy chyby v k -tém kroku tak můžeme využít odhad (1.19) odvozený v sekci 1.3, kde budeme uvažovat číslo podmíněnosti matice $AQ = Q^T AQ$ na podprostoru $A\mathcal{V}$, které pro zjednodušení zápisu označíme $\kappa(Q^T AQ|A\mathcal{V})$. Jak jsme ukázali v (2.7), na prostoru $A\mathcal{V}$ je matice $Q^T AQ$ pozitivně definitní. Dostáváme tak (viz rovněž Saad a kol. (2000, Theorem 4.3))

$$\frac{\|x_\star - x_k\|_A}{\|x_\star - x_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa(Q^T AQ|A\mathcal{V})} - 1}{\sqrt{\kappa(Q^T AQ|A\mathcal{V})} + 1} \right)^k.$$

Zároveň podobně jako v Dostál (1988, sekce 5) s využitím vlastností Rayleighova podílu ukážeme, že $\kappa(Q^T AQ|A\mathcal{V}) \leq \kappa(A)$. Pro symetrickou matici $B \in \mathbb{R}^{n \times n}$ a nenulový vektor $w \in \mathbb{R}^n$ definujeme Rayleighův podíl jako

$$R(B, w) := \frac{w^T B w}{w^T w}.$$

Označíme-li μ_{\min} a μ_{\max} nejmenší a největší vlastní číslo matice B , pak pro každý nenulový vektor w platí (viz Saad (2003, vztahy (1.38) a (1.40)))

$$\mu_{\min} \leq R(B, w) \leq \mu_{\max}, \quad (2.14)$$

přičemž dolní i horní meze se nabývá pro vlastní vektor příslušný μ_{\min} , resp. μ_{\max} . Pro libovolný vektor $v \in A\mathcal{V}$ platí

$$R(Q^T AQ, v) = \frac{v^T Q^T A Q v}{\|v\|^2} \geq \frac{(Qv)^T A (Qv)}{\|Qv\|^2} = R(A, Qv) \geq \lambda_1,$$

kde jsme v první nerovnosti využili (2.6), a

$$\begin{aligned} R(Q^T AQ, v) &= \frac{v^T Q^T A Q v}{\|v\|^2} \leq \frac{v^T Q^T A Q v}{\|v\|^2} + \frac{v^T P^T A P v}{\|v\|^2} \\ &= \frac{v^T A v}{\|v\|^2} = R(A, v) \leq \lambda_n. \end{aligned}$$

Poslední nerovnosti v obou výrazech vyplývají z (2.14) pro matici A a vektor v . Označme dále $\tilde{\lambda}_{\min}$ a $\tilde{\lambda}_{\max}$ nejmenší a největší vlastní číslo matice $Q^T AQ$ na podprostoru $A\mathcal{V}$. Pro $v \in A\mathcal{V}$ pak podle (2.14) platí

$$\tilde{\lambda}_{\min} \leq R(Q^T AQ, v) \leq \tilde{\lambda}_{\max}$$

a krajních hodnot se nabývá. Musí proto platit $\tilde{\lambda}_{\min} \geq \lambda_1$ a $\tilde{\lambda}_{\max} \leq \lambda_n$, z čehož již plyne $\kappa(Q^T A Q|A\mathcal{V}) \leq \kappa(A)$.

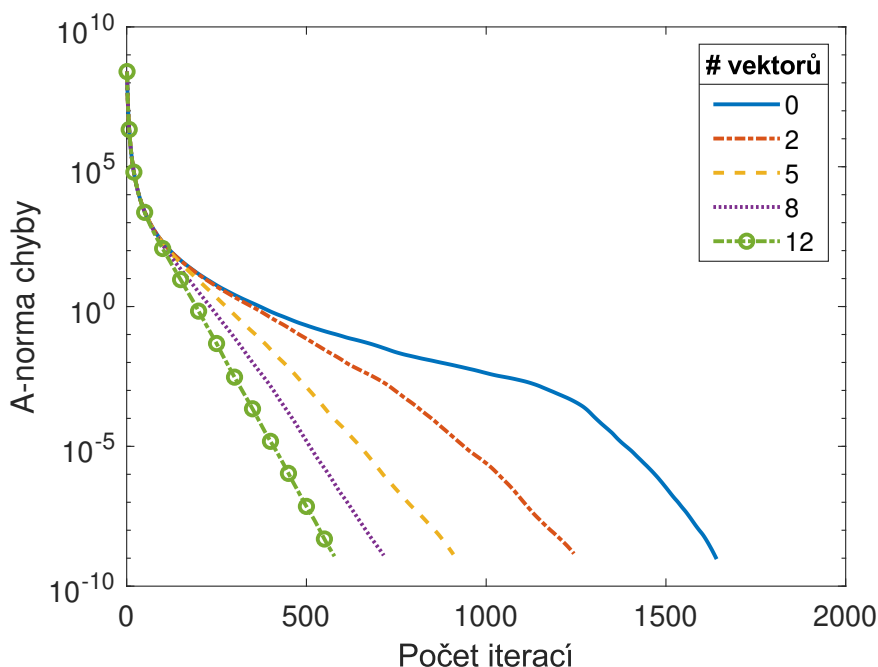
V případě, kdy $\mathcal{U} = \text{span}\{v_1, \dots, v_m\}$, kde v_1, \dots, v_m jsou vlastní vektory příslušné m nejmenším vlastním číslům matice A , je $\kappa(Q^T A Q|A\mathcal{V}) = \lambda_n/\lambda_{m+1}$. Toho využijeme v části s numerickými experimenty a analogická situace je rovněž uvažována v Saad a kol. (2000).

Na závěr poznamenejme, že deflaci lze rovněž zkombinovat s předpodmíněním původní úlohy (1.1). Výsledný algoritmus Preconditioned Deflated CG je uveden například v Saad a kol. (2000, Algorithm 3.6).

3. Numerické experimenty

V závěrečné kapitole představíme výsledky numerických experimentů, na kterých jsme zkoumali vliv volby různých podprostorů \mathcal{U} na rychlost konvergence. Experimenty jsme prováděli v MATLABu a využili jsme vlastní implementace algoritmů CG a deflated CG. Jako počáteční aproximace jsme uvažovali nulový vektor a jako zastavovací kritérium zvolili relativní normu rezidua $\|r_k\|/\|r_0\|$ menší než 10^{-10} . V grafech vykreslujeme závislost A -normy chyby na počtu iterací. Zvolili jsme proto přesné řešení jako vektor jedniček a jemu odpovídající pravou stranu. Experimenty jsme prováděli na matici Trefethen 20000 řádu $n = 20000$ ze sbírky SuiteSparse Matrix Collection¹.

V prvním experimentu jsme jako sloupce matice U neboli bázi prostoru \mathcal{U} zvolili vlastní vektory příslušné několika nejmenším vlastním číslům. Jejich počet jsme postupně zvyšovali, nula odpovídá standardnímu CG bez deflace.



Obrázek 3.1: Vliv deflace vlastních vektorů příslušných nejmenším vlastním číslům na konvergenci

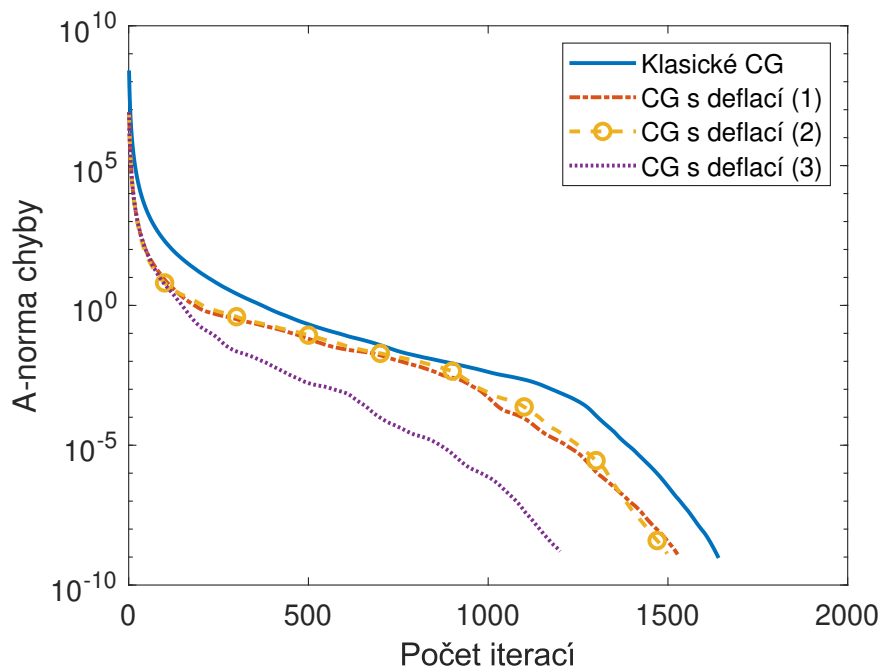
Z Obrázku 3.1 je patrné, že s rozšiřováním prostoru \mathcal{U} se počet iterací potřebný k dosažení požadované přesnosti postupně snižuje. V případě osmi vektorů již stačila méně než polovina iterací oproti standardnímu CG. Abychom vysvětlili možný důvod, spočítali jsme vlastní čísla matice A a odpovídající číslo podmíněnosti $\kappa(Q^T A Q|A\mathcal{V})$. To se výrazně snižuje (viz Tabulka 3.1), a je proto pochopitelné (ne však zaručené), že dochází ke zrychlení konvergence.

¹Davis a Hu (2011), <http://sparse.tamu.edu/>

počet vektorů	0	2	5	8	12
$\kappa(Q^T A Q A\mathcal{V})$	200 559	45 859	17 050	9 695	5 523

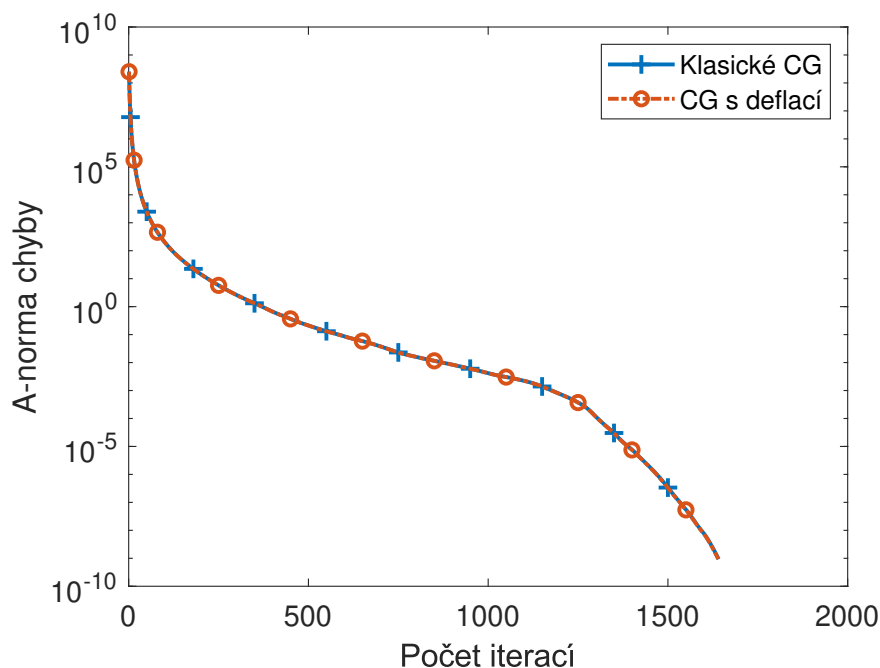
Tabulka 3.1: Číslo podmíněnosti matice $Q^T A Q|A\mathcal{V}$ při deflaci různého počtu vlastních vektorů matice A odpovídajících nejmenším vlastním číslům

Dále jsme jako sloupce matice U zvolili 10 náhodných vektorů. Na Obrázku 3.2 jsou vykresleny konvergenční křivky pro tři různé množiny vektorů. Jelikož jsme uvažovali náhodné vektory, není překvapivé, že se výsledné křivky liší. V jednom případě (křivka (3)) jsme dosáhli poměrně výrazného snížení počtu iterací. Jak však ukazují další dva případy, jedná se spíše o výjimku a obecně nelze předpokládat výrazný efekt.



Obrázek 3.2: Deflace tří různých sad náhodných vektorů

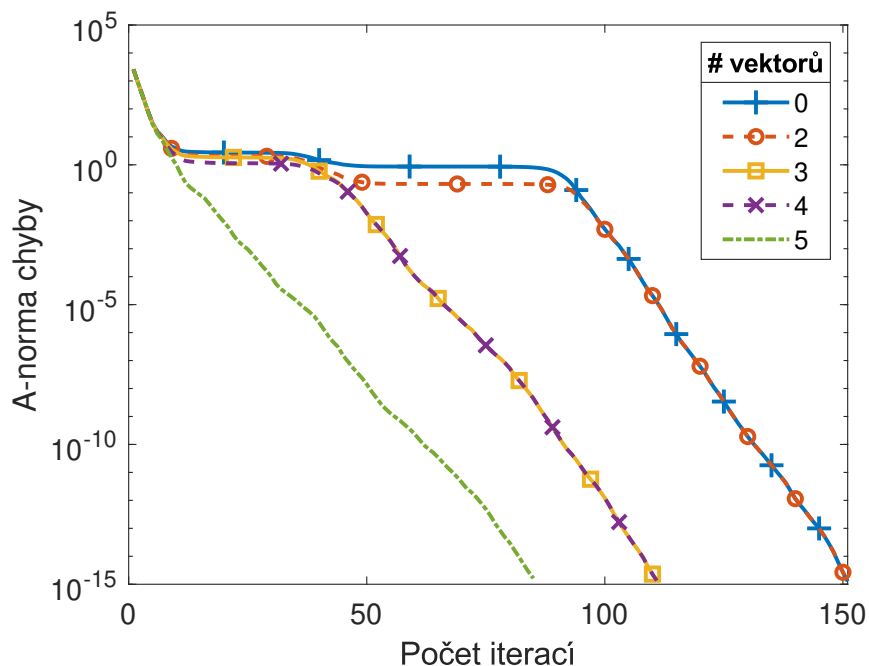
Zatímco při deflaci vlastních vektorů příslušných nejmenším vlastním číslům stačily pouze dva k výraznému snížení počtu iterací, v případě vlastních vektorů příslušných největším vlastním číslům jsme podobného efektu nedosáhli. Dokonce ani 16 vektorů konvergenci prakticky vůbec neurychlilo (viz Obrázek 3.3).



Obrázek 3.3: Deflace 16 vlastních vektorů příslušných největším vlastním číslům

Předchozí pozorování opět doplníme o výpočet vlastních čísel. Hodnoty 1., 16. a 17. největšího jsou po řadě (zaokrouhlo na jednotky) 224737, 224569 a 224563. Navíc ještě sté největší vlastní číslo je rovno 223469, a tedy vlastní čísla jsou výrazně klastrovaná u horního okraje spektra. Zdá se tedy, že pokud do podprostoru \mathcal{U} nezahrneme všechny vektory příslušné podobně velkým (klastrovaným) vlastním číslům, nemůžeme očekávat zrychlení konvergence. Pro doplnění, nejmenší vlastní číslo je 1,12. Číslo podmíněnosti $\kappa(Q^T A Q | A \mathcal{V})$ se tak při deflaci 16 vektorů příliš nesnížilo.

Vliv vlastních čísel tvořících klastry ilustrujeme ještě na jednoduchém příkladu s maticí řádu $n = 1000$ s námi zvoleným spektrem. Uvažovali jsme tři nejmenší vlastní čísla stejná, $\lambda_1 = \lambda_2 = \lambda_3 = 0,001$, a následující dvě $\lambda_4 = \lambda_5 = 0,05$. Zbytek spektra tvořila náhodná čísla z intervalu $[10, 1000]$. Bázi podprostoru \mathcal{U} postupně tvořilo 2, 3, 4 a 5 vlastních vektorů příslušných nejmenším vlastním číslům. Nula opět odpovídá standardnímu CG bez deflace.



Obrázek 3.4: Matice se zvoleným spektrem, deflace vlastních vektorů příslušných nejmenším vlastním číslům

Na Obrázku 3.4 vidíme, že ke zrychlení dojde až při deflaci tří vektorů. Pokud do podprostoru \mathcal{U} zahrneme vektory pouze dva, řeší standardní CG a deflated CG soustavy se stejným číslem podmíněnosti $\kappa(A) = \kappa(Q^T A Q|_{\mathcal{AV}})$ a podprostor odpovídající vlastnímu číslu λ_3 zřejmě „brzdí“ konvergenci. Podobné chování pozorujeme i v případě čtyř a pěti vektorů, deflace čtvrtého vektoru konvergenci neurychlila. V situaci, kdy jsou vlastní čísla klastrovaná, tak podprostor \mathcal{U} musí zahrnovat všechny odpovídající vlastní vektory.

Celkově můžeme shrnout, že podoba prostoru \mathcal{U} a spektra matice A má na případné urychlení konvergence zásadní vliv. Při vhodné volbě se počet iterací potřebný k dosažení požadované přesnosti poměrně výrazně snížil. V našich případech se jako nejefektivnější ukázala deflace vlastních vektorů příslušných vlastním číslům tvořící klastry výrazně oddělené od zbytku spektra.

Závěr

Uvažovali jsme soustavu lineárních algebraických rovnic se symetrickou pozitivně definitní maticí. Popsali jsme metodu sdružených gradientů, iterační metodu pro řešení takovýchto soustav. Uvedli jsme její odvození a dvě techniky, předpodmínění a deflaci, které mají za cíl urychlit konvergenci metody.

Detailněji jsme se zaměřili na deflaci. Okomentovali jsme vybrané články zabývající se touto technikou, vysvětlili jsme její princip a důkladně odvodili algoritmus deflated CG. Rozebrali jsme i kroky, které nejsou v literatuře příliš podrobně vysvětleny a ukázali některé vlastnosti.

S využitím vlastní implementace algoritmů CG a deflated CG v MATLABu jsme na konkrétních příkladech zkoumali efekt deflace, zejména vliv různé volby deflačního podprostoru. Ilustrovali jsme situace, kdy je deflace účinná, ale i případ, kdy k požadovanému urychlení konvergence nedošlo. Jako nejefektivnější se v našich příkladech ukázala deflace vlastních vektorů příslušných vlastním číslem tvořící jasně oddělené klastry, kdy jsme do deflačního podprostoru zahrnuli všechny vektory odpovídající danému klastru.

Seznam použité literatury

- DAVIS, T. A. a HU, Y. (2011). The University of Florida sparse matrix collection. *ACM Trans. Math. Softw.*, **38**(1), 25. ISSN 0098-3500. doi: 10.1145/2049662.2049663. Id/No 1.
- DOSTÁL, Z. (1988). Conjugate gradient method with preconditioning by projector. *Int. J. Comput. Math.*, **23**(3-4), 315–323. ISSN 0020-7160. doi: 10.1080/00207168808803625.
- GREENBAUM, A. (1997). *Iterative methods for solving linear systems*, volume 17 of *Front. Appl. Math.* Philadelphia, PA: SIAM Society for Industrial and Applied Mathematics. ISBN 0-89871-396-X.
- HESTENES, M. R. a STIEFEL, E. (1952). Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, **49**, 409–436. ISSN 0160-1741. doi: 10.6028/jres.049.044.
- LIESEN, J. a STRAKOŠ, Z. (2013). *Krylov subspace methods. Principles and analysis*. Oxford: Oxford University Press. ISBN 978-0-19-965541-0.
- NICOLAIDES, R. A. (1987). Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, **24**, 355–365. ISSN 0036-1429. doi: 10.1137/0724027.
- SAAD, Y., YEUNG, M., ERHEL, J. a GUYOMARC'H, F. (2000). A deflated version of the conjugate gradient algorithm. *SIAM J. Sci. Comput.*, **21**(5), 1909–1926. ISSN 1064-8275. doi: 10.1137/S1064829598339761.
- SAAD, Y. (2003). *Iterative methods for sparse linear systems*. Philadelphia, PA: SIAM Society for Industrial and Applied Mathematics, 2nd ed. edition. ISBN 0-89871-534-2.