Tomáš Ondruch, 3eMAI/2

Zobecněná metoda minimálních reziduí (GMRES)

Úvod

Zobecněná metoda minimálních reziduí (anglicky *Generalized Minimal Residual Method*, odtud zkratka *GMRES*) se řadí mezi pokročilé iterační metody pro výpočet soustav lineárních rovnic tvaru $A\boldsymbol{x} = \boldsymbol{b}$. Jedná se o zobecněnou formu algoritmu jednokrokové *metody minimálních reziduí* (označení *MR,* někdy také *MINRES*). Metodu uvedli v roce 1986 američtí matematici Yousef Saad a Martin H. Schultz.

Princip metody

V tomto odstavci bude stručně popsán princip metody a budou zmíněny vybrané pojmy, které se k dané metodě pojí. Metoda spočívá v minimalizaci normy vektoru reziduí $\|r_k\|^2$, kde index k se vztahuje k hodnotě v k-tém kroku algoritmu. Při hledání řešení se pracuje s prvky *Krylovova podprostoru* $K_k(A, r_0)$, jehož přirozená báze je tvaru $\{ r_0, Ar_0, \dots, A^{k-1}r_0 \}$. Tato báze je však špatně podmíněná, pomocí algoritmu *AGS (Arnoldi-Gram-Schmidt)* je tedy v Krylovově podprostoru K_k sestrojena ortonormální báze $\{v_i\}_{i=1}^{k}$. V dalším postupu je sestrojena *horní Hessenbergova matice* $\bar{H}_k$ typu $(k+1, k)$ a následně proveden její QR *rozklad* $\bar{H}_k = Q_{k+1}\bar{R}_k$ (v přiloženém programu pro MatLab byl zvolen postup QR rozkladu pomocí *Givensova algoritmu*), kde Q_{k+1} je ortonormální matice řádu $k+1$ a $\bar{R}_k$ je horní trojúhelníková matice. S využitím získaných matic Q_{k+1} a $\bar{R}_k$ a normy počátečního rezidua $\|r_0\|$ lze ve finální části srovnat velikost k-tého rezidua r_k s požadovanou přesností výsledku ε. V případě, že $\|r_k\| < \varepsilon$ je algoritmus ukončen a je stanoveno řešení $x \approx x_k$, v opačném případě se přejde k dalšímu kroku. Algoritmus GMRES je obecně m-kroková metoda.

Využití metody

Základní předpoklad použití metody je, že při řešení soustavy $A\boldsymbol{x} = \boldsymbol{b}$ je A čtvercovou maticí. Její použití je velmi vhodné v případě, že A není symetrická. V případě velkého počtu kroků m je však pro výpočty nutné, aby V_m (matice ortonormálních bázových vektorů Krylovova podprostoru) a $\bar{H}_m$ (horní Hessenbergova matice) zůstaly uloženy v paměti počítače. Také z tohoto důvodu může následně rychlost výpočtu úlohy výrazně poklesnout. Rychlost konvergence výpočtu lze zvýšit použitím předpodmiňovací matice M – řešená rovnice pak bude tvaru $M^{-1}A\boldsymbol{x} = M^{-1}\boldsymbol{b}$**.** Jednou z možností při sestrojování matice M je využití neúplného LU rozkladu matice A; tato modifikace však v kódu přiloženého programu není implementována.

Mezi výhody zobecněné metody minimálních reziduí lze zařadit její schopnost řešit soustavy s nesymetrickou čtvercovou maticí A a schopnosti výpočtu přesného řešení pro počet kroků $m = n$, kde n je řád matice A (při splněné podmínce dobré podmíněnosti matice). Největší nevýhodou jsou naopak vysoké nároky na paměť výpočetní techniky a celkový výpočetní čas.

Některé další metody pro řešení soustav lineárních rovnic

Pravděpodobně nejznámějším zástupcem pro řešení soustav lineárních rovnic je *Gaussova eliminační metoda*. Jeho varianta *GEMz*, tj. bez výběru hlavního prvku je však náchylná na případ, kdy při výpočtu multiplikátoru algoritmus začne pracovat s nulovým pivotem – v takovém případě

výpočet havaruje. Omezení metody také spočívá ve skutečnosti, že metodou lze řešit pouze soustavy, jejichž matice je buďto ryze diagonálně dominantní nebo pozitivně definitní. Modifikace *Gaussovy eliminační metody s částečným výběrem hlavního prvku* je již propracovanější, ovšem úspěšný výpočet závisí na dobré podmíněnosti matice soustavy.

Mezi klasické iterační metody se řadí *Jacobiho metoda*, jejíž nutnou podmínkou konvergence je ryzí diagonální dominance matice soustavy, a *Gauss-Seidlova metoda* – u ní je pro úspěšnou konvergenci nutné splnění jedné ze dvou podmínek, a to podmínky ryzí diagonální dominance nebo pozitivní definitnosti matice soustavy. Pro zlepšení konvergence těchto metod se užívají jejich relaxační formy.

Metoda sdružených gradientů se stejně jako GMRES řadí mezi pokročilé iterační metody. Tato metoda je pravděpodobně nejpoužívanější pro řešení pozitivně definitních matic. Její nevýhody jsou podobné jako u zobecněné metody minimálních reziduí, tj. přísné požadavky na paměť počítače a rychlost výpočtu. Také zde se pro rychlejší konvergenci úloha modifikuje s použitím předpodmiňovací matice. Pravděpodobně nejvýraznějším „konkurentem“ algoritmu *GMRES* je *metoda bikonjugovaných gradientů*, která se rovněž používá při řešení soustav s nesymetrickou maticí, případně je také velmi vhodná pro řešení soustav s řídkou maticí. Nevýhodou této metody však je skutečnost, že její konvergence není obecně zaručena – metoda je numericky nestabilní.

Vypracovaný program

V prostředí MatLab byl vytvořen program, jehož algoritmus realizuje výpočet řešení soustav lineárních rovnic právě pomocí algoritmu *GMRES.* Při tvorbě algoritmu byly použity poznámky a stručné úryvky kódu, které jsou k nalezení v pdf-dokumentu *Vybrané statě z numerických metod*, který vytvořil doc. RNDr. Libor Čermák, CSc. Funkčnost programu byla mimo jiné ověřena na vzorovém příkladu, kterému se věnuje následující část textu.

Vzorový příklad – nastavení a popis fáze testování

Veškeré vstupy, tj. matice A, vektor pravé strany b, počáteční vektor řešení x_0 a požadovaná tolerance ε, jsou pro jednoduchost a bezproblémové spouštění skriptu obsaženy přímo ve zdrojovém kódu programu. Potřebné výstupy, kterými jsou počet provedených iterací a nalezené řešení x, jsou pro uživatele připraveny k uložení.

Vzorové volání programu v prostředí MatLab:

```
[reseni, reps]=gmresOndruch
```

, kde `reseni` představuje vektor nalezeného řešení, proměnná `reps` zaznamená počet potřebných iterací k nalezení řešení, neboli počet „restartů“ algoritmu.

Pro důkladné ověření funkčnosti programu byl skript ve fázi ladění testován na různé varianty zadání. Bylo také provedeno několik ověřovacích výpočtů, jejichž vstupy byly náhodně generovány následujícím způsobem:

```
A=randi([-10 10],10,10);
```

```
b=randi([-10 10],10,1);
```

Při testování s použitím těchto vstupů bylo ověřeno, že při zadané toleranci ε=5e−14 byl u velkého počtu výpočtů počet restartů v rozmezí $1 <$ reps < 100, kde hodnota 100 představuje nastavený počet maximálních restartů, ve skriptu pojmenovaný maxReps. Tato hodnota tolerance byla zvolena také pro vzorový příklad, neboť názorně demonstruje funkčnost programu v jeho schopnosti provádět nové restarty výpočtu. Počáteční vektor řešení x_0 byl nastaven jako vektor nulový.

Vzorový příklad - zadání

Z testování při výše zadaných parametrech byl vybrán jeden vzorový příklad, který demonstruje funkčnost vytvořeného programu. Jedná se o řešení soustavy deseti lineárních rovnic, jejíž parametry byly vytvořeny náhodným generováním (viz předchozí odstavec). Vstupními daty jsou:

```
A = [-5 -2 3 -7 -6 -9 -9 -5 -4 -2; -10 1 -6 8 7 -4 -9 -9 8  8; 7 -4
2 0 -7 -7 9 -10 0 0; 6 -10 10 -6 9 9 2 2 0 9; -8 -5 -4 -5 6 -8 10 9
-1 -7;
-8 3 -8 4 9 -9 -7 5 -3 -4; -3 -5 5 6 -6 3 9 -9 -2 7; -2 -7 9 0 -1 -1
3 8 -2 6; -5 6 -3 9 3 -2 -2 -9 2 9; -10 7 -8 3 10 7 -8 7 -9 -6];


b=[-3; 4; 9; -1; 2; 4; 6; 8; -1; 5];


epsil=5e-14;


x0=zeros(length(b), 1);
```

Dalšími volitelnými parametry úlohy mohou být počet maximálních restartů maxReps, ve skriptu nastavený na hodnotu 100, případně počet kroků m v jedné iteraci, který ve vypracovaném programu je nastaven na hodnotu rovnou počtu rovnic soustavy.

Vzorový příklad – výsledky

Po spuštění programu v okně MatLabu se v okně vypisuje hodnota aktuální prováděné iterace reps. Po ukončení výpočtu vzorového příkladu se vypíší výsledky:

```
reseni =
   8.099282752442452
 -16.472713526609457
 -48.091370132338113
 -14.150668463106921
 -18.924055304130011
   2.300375103102610
  -2.193947800257793
  11.402332431373045
 -15.931299108876184
  34.764801256778490


reps = 27
```

Výsledky provedeného výpočtu byly porovnány s výstupy po zadání příkazu `linsolve(A,b)`, kde A, b jsou stejné jako v případě výpočtu algoritmem GMRES. Výsledný vektor řešení je tvaru

```
resenilinsolve =
   8.099282752442493
 -16.472713526609553
 -48.091370132338398
 -14.150668463107026
 -18.924055304130118
   2.300375103102623
  -2.193947800257802
  11.402332431373102
 -15.931299108876276
  34.764801256778689
```

Při srovnání získaných řešení si lze povšimnout, že hodnoty vektoru získaného metodou *GMRES* jsou ve velmi dobré shodě s hodnotami pro vektor `resenilinsolve`. Hodnoty se liší až na třináctém desetinném místě a lze tedy usoudit, že výpočet proběhnul při splnění požadované tolerance ε.

Poznámka: Program byl rovněž testován na případ „neřešitelné“ soustavy, např. pro vstupy

```
A = [1 1; 1 1];
b = [2; 3];
```

Pro tato data program správně vyhodnotil neřešitelnost soustavy, kdy po provedení maximálního počtu iterací vypsal přednastavené hlášení o neúspěšném výpočtu.

Jelikož při testování programu nebylo objeveno zadání, pro které by program při výpočtu selhal, lze považovat vypracovaný skript za funkční a připravený k praktickému použití, byť pro řešení rozsáhlejších soustav připadá v úvahu optimalizace kódu k dosažení nižší časové a paměťové náročnosti.