NMB - Oefenzitting 7:
Iteratieve methoden

Simon Telen & Daan Camps

1 Convergentie CG

Opgave 1. Pen en papier

Ga voor jezelf na dat formule (38.9) op p. 298 van [Trefethen & Bau] wil zeggen dat voor alle veeltermen $p_n(z) \in P_n$ (veeltermen van graad $\leq n$ met $p_n(0) = 1$) het volgende geldt

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq \max_{z \in \Lambda(A)} |p_n(z)| \tag{1}$$

We kunnen aan de hand van deze formule voor bepaalde matrices de convergentie voorspellen aan de hand van een goed gekozen rij veeltermen. Er geldt ook nog

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^n \tag{2}$$

waarbij $\|x\|_A = \|A^{\frac{1}{2}} x\| = \sqrt{x^* A x}$.

Voorspel de convergentie voor matrices met de onderstaande eigenschappen. Veronderstel dat deze matrices symmetrisch-positief definiet zijn. Geef ook een schatting voor het aantal iteraties nodig om de A-norm van de fout ten opzichte van de initiële fout met een factor 10^{-3} te verkleinen. Tip: veeltermen van de vorm $p_n(z) = (1 - z/a)^n$ voldoen aan de voorwaarde $p_n(0) = 1$.

a) Eigenwaarden in het interval $(9, 11)$.

> **Oplossing.** $\lambda \in (9, 11)$
> *We kiezen als reeks veeltermen* $p_n(z) = (1 - z/10)^n$. *Het is duidelijk dat deze veeltermen hun maximum bereiken in de randpunten van het interval* $(9, 11)$. *We vinden dan ook*
>
> $$\frac{\|e_n\|_A}{\|e_0\|_A} \leq (0.1)^n$$
>
> *Om een reductie van* 10^{-3} *te bereiken moet dus gelden dat* $10^{-n} \leq 10^{-3}$ *of nog* $n \geq 3$.

b) Eigenwaarden in $(1, 1.5) \cup (399, 400)$. Gebruik eerst de formule met $\kappa(A)$. Verscherp deze schatting gebruik makend van de rij veeltermen $p_{3k}(z) = (1 - z/1.25)^k (1 - z/400)^{2k}$.

Oplossing. $\lambda \in (1, 1.5) \cup (399, 400)$.
Deze matrix is symmetrisch positief-definiet. De singuliere waarden zijn in dit geval gelijk aan de eigenwaarden. We weten dus dat $\kappa(A) = \frac{\sigma_1}{\sigma_m} \leq 400$. *We krijgen* $\frac{\|e_n\|_A}{\|e_0\|_A} \leq 2\left(\frac{19}{21}\right)^n$. *Voor het aantal iteraties zou dus gelden* $2\left(\frac{19}{21}\right)^n \leq 10^{-3}$ *of* $n \geq \left\lceil \frac{-3-\log_{10} 2}{\log_{10} \frac{19}{21}} \right\rceil = 76$. *Analyse aan de hand van de rij veeltermen* $p_n(z) = (1 - z/200.5)^n$ *zou ons nog slechtere convergentie doen vermoeden. Met de rij* $p_{3k}(z) = (1 - z/1.25)^k (1 - z/400)^{2k}$ *vinden we echter dat*

$$\frac{\|e_{3k}\|_A}{\|e_0\|_A} \leq \max p_{3k}(z) \leq \left(1 - \frac{1}{1.25}\right)^k = (0.2)^k$$

Er treedt per 3 iteraties een reductie op van 0.2. *We vinden dus* $n \geq 3 \left\lceil \frac{-3}{\log_{10}(0.2)} \right\rceil = 15$.

Opgave 2. Matlab

Controleer je bevindingen met numerieke experimenten. Je kan de volgende MATLAB-functies gebruiken. Gebruik `help <naam>` voor extra informatie.

Naam	info
`linspace.m`	`N` reële getallen uniform verdeeld in [`rmin`, `rmax`]
`eigint.m`	`N` willekeurige reële getallen uniform verdeeld in [`rmin`, `rmax`)
`eigcirk.m`	`N` willekeurige complexe getallen λ met $\lvert\lambda - \mathtt{c}\rvert < \mathtt{R}$
`willglv.m`	past een willekeurige gelijkvormigheidstransformatie toe op de diagonaalmatrix met opgegeven waarden
`willorth.m`	pas een willekeurige orthogonale transformatie toe op de diagonaalmatrix met opgeven waarden
`cg.m`	`help cg` (`M=eye(size(A))`)

Je kan bijvoorbeeld een matrix maken met 10 eigenwaarden in $(3, 4)$ en 5 eigenwaarden in $(7, 8)$ met het volgende bevel

```
A = willglv([eigint(3,4,10); eigint(7,8,5)]);
```

Gebruik je `willorth` dan is A, in dit geval, symmetrisch positief-definiet. Je kan de eigenwaarden bekijken met `plot(eig(A),'+')`.

Suggestie: Genereer een matrix A, kies een exacte oplossing x^*, bepaal het rechterlid $b = Ax^*$. Los het stelsel $Ax = b$ op met CG. Maak grafieken van de norm van de fout $(x - x^*)$, het residu $(b - Ax)$ en van de A-norm van de fout. Tip: maak hiervoor een m-bestand (of meerdere natuurlijk).

Probeer matrices die voldoen aan a) en b) uit de vorige opgave. Probeer verschillende liggingen van 1 en 2 intervallen. Formuleer enkele besluiten.

Oplossing. *Zie* `vergelijk.m`.

2 CG versus steilste helling

CG kan beschouwd worden als een methode om de functie $\phi(x) = \frac{1}{2}x^T Ax - x^T b$ te minimaliseren. Een andere manier is de methode van de steilste helling. Een iteratie bestaat dan uit het nemen van een stap in de richting van de negatieve gradiënt van $\phi(x)$.

Opgave 3. Pen en papier

a) Leid af dat $\nabla\phi(x) = -r$.

Oplossing. *We minimaliseren*

$$\phi(x) = \frac{1}{2}x^T Ax - x^T b$$

Als we dit uitschrijven krijgen we

$$\frac{1}{2}\sum_{i,j} x_i a_{i,j} x_j - \sum_j b_j x_j$$

Afleiden naar een component x_k geeft 3 gevallen één voor $i = k$, $j \neq k$, één voor $j = k$, $i \neq k$ en één voor $i = j = k$

$$\frac{1}{2}\sum_{j\neq k} a_{k,j} x_j + \frac{1}{2}\sum_{i\neq k} x_i a_{i,k} + a_{k,k}x_k - b_k$$

Aangezien A symmetrisch is, krijgen we

$$\nabla\phi(x) = \sum_i a_{k,j}x_j - b_k = Ax - b = -r$$

b) Bepaal de optimale staplengte voor de iteratie $x_n = x_{n-1} - \alpha_n \nabla\phi(x_{n-1})$.

Oplossing. *Een iteratiestap ziet er als volgt uit*

$$x_n = x_{n-1} + \alpha_n r_{n-1}$$

De waarde van de te minimaliseren functie in stap n wordt dus

$$\begin{aligned}
\phi(x_n) &= \phi(x_{n-1} + \alpha_n r_{n-1}) \\
&= \frac{1}{2}(x_{n-1} + \alpha_n r_{n-1})^T A(x_{n-1} + \alpha_n r_{n-1}) - (x_{n-1} + \alpha_n r_{n-1})^T b \\
&= \phi(x_{n-1}) + \alpha_n \left[\frac{1}{2}r_{n-1}^T Ax_{n-1} + \frac{1}{2}x_{n-1}^T Ar_{n-1} - r_{n-1}^T b\right] + \frac{1}{2}\alpha_n^2 r_{n-1}^T Ar_{n-1} \\
&= \phi(x_{n-1}) + \alpha_n r_{n-1}^T (Ax_{n-1} - b) + \frac{1}{2}\alpha_n^2 r_{n-1}^T Ar_{n-1} \\
&= \phi(x_{n-1}) - \alpha_n r_{n-1}^T r_{n-1} + \frac{1}{2}\alpha_n^2 r_{n-1}^T Ar_{n-1}
\end{aligned}$$

Bij de overgang van de derde naar de vierde lijn wordt gebruik gemaakt van de symmetrie van A. Afleiden naar α_n en gelijkstellen aan nul levert de volgende optimale staplengte op

$$\alpha_n = \frac{r_{n-1}^T r_{n-1}}{r_{n-1}^T Ar_{n-1}}$$

c) Toon aan dat de methode van de steilste helling convergeert.

Oplossing. *Gebruiken we de α_n uit de vorige vraag dan vinden we*

$$\begin{aligned}\phi(x_n) &= \phi(x_{n-1}) - \alpha_n r_{n-1}^T r_{n-1} + \frac{1}{2}\alpha_n^2 r_{n-1}^T A r_{n-1} \\ &= \phi(x_{n-1}) - \frac{(r_{n-1}^T r_{n-1})^2}{r_{n-1}^T A r_{n-1}} + \frac{1}{2}\frac{(r_{n-1}^T r_{n-1})^2}{(r_{n-1}^T A r_{n-1})^2} r_{n-1}^T A r_{n-1} \\ &= \phi(x_{n-1}) - \frac{1}{2}\frac{(r_{n-1}^T r_{n-1})^2}{r_{n-1}^T A r_{n-1}}\end{aligned}$$

Aangezien A symmetrisch positief-definiet is, krijgen we

$$\phi(x_n) \leq \phi(x_{n-1})$$

De gelijkheid geldt enkel als $r_{n-1} = 0$ m.a.w. bij convergentie. De iteratie zal dus convergeren.

Opgave 4. Matlab

a) Implementeer de methode van de steilste helling. (Een lus met 3 bevelen.)

Oplossing. *Zie* `steilstehelling.m`.

b) Vergelijk de convergentie van CG en de methode van de steilste helling voor een spd matrix met $\kappa = 10$. (Gebruik de norm van het residu.)

Oplossing. *Zie* `cg_sh.m`.

3 Eigenwaarden bepalen met de Arnoldi iteratie

Opgave 5. Matlab

Implementeer de Arnoldi iteratie. De iteratie hoeft geen rekening te houden met 'breakdown' (deling door 0). Bepaal bij elke iteratie de grootste en kleinste Ritz-eigenwaarden (efficiëntie is niet belangrijk, gebruik de standaard Matlab-functie). Construeer een matrix met eigenwaarden in het interval $(4, 5)$. Maak een grafiek van de absolute waarde van het verschil tussen de grootste Ritz-eigenwaarde en de grootste eigenwaarde. Idem voor de kleinste Ritz-eigenwaarde en de kleinste eigenwaarde. Doe nu hetzelfde maar vervang een van de eigenwaarden door 8 en vervolgens ook een door 2. Wat stel je vast? Wat zou je hieruit kunnen besluiten?

Oplossing. *Zie* `arnoldi.m` *en* `arnoldiew.m`.

4 The SIAM 100-Dollar, 100-Digit Challenge

Opgave 6. Zoek op het internet het artikel met als titel 'A Hundred-Dollar, Hundred-Digit Challenge'. Het is geschreven door de auteur van het tekstboek dat voor dit vak gebruikt wordt. Los probleem 7 uit dit artikel op.

Oplossing.

```
% challenge7.m

n = 20000;
P = 224737;
p = primes(P)';
m = floor(log2(n));
e = ones(n, m+1);
i = 2.^(0:m);
B = [ e p e ];
d = [ -i 0 i ];
A = spdiags(B, d, n, n);
b = eye(n, 1);

D = spdiags(p, 0, n, n);
L = triu(A);
x = rand(n, 1);
maxit = 400;
tol = 1e-15;

method = 3;

if (method ~= 2) && (method ~= 3)
  x1 = x(1);
  R = [];
  X = [];
  for it = 1:maxit
    r = b - A*x;
    if method == 1
      % gauss-seidel
      x = x + L \ r;
    else
      % jacobi
      x = x  + r ./ p;
    end
    R(it) = norm(r);
    X(it) = abs(x1 - x(1));
    %if (R(it) < tol * R(1)) || (X(it) < tol), break; end
    if (X(it) < tol), break; end
    x1 = x(1);
  end
  semilogy(1:length(R), R, 1:length(X), X)
else
```

Oplossing (vervolg).

```
    if method == 3
      % cg symmetric gauss-seidel
      M1 = L + D;
      M2 = D \ (D + L');
    else
      % cg diag
      M1 = D;
      M2 = [];
    end
    [x, flag, relres, iter, resvec] = pcg(A, b, tol, maxit, M1, M2);
    x1 = x(1);
    semilogy(resvec);
  end

  x1
```