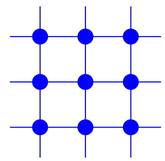
WISB356, Utrecht, 2 oktober 2012

Scientific Computing

Gerard Sleijpen
Rob Bisseling
Alessandro Sbrizzi

Universiteit Utrecht
Department of Mathematics

http://www.staff.science.uu.nl/~sleij101/

Aspect werkelijkheid	**Stroming grondwater**
Modelleer ⇓	⇓
Wiskundig model	$-\nabla(K\nabla\phi) = Q$ op Ω $-K\,\frac{\partial\phi}{\partial x}\cdot n = \gamma(\phi - \phi_0)$ op $\partial\Omega$
Discretiseer ⇓	⇓ Eindige differences, ...
Discreet model	 ... $\frac{\phi(x+\Delta x,y)-\phi(x-\Delta x,y)}{2\Delta x}$ ... $\mathbf{Ax} = \mathbf{b}$
⇓	⇓ Iteratieve lineaire solver
Computer model	Rekenschema voor het oplossen van $\mathbf{x}$ uit $\mathbf{Ax} = \mathbf{b}$
Implementeer ⇓	⇓ C++
Simulatie	Simulatie

WISB356, Utrecht, 2 oktober 2012

Iterative methoden voor lineaire vergelijkingen

Gerard Sleijpen

Universiteit Utrecht
Department of Mathematics

http://www.staff.science.uu.nl/~sleij101/

Program

- <u>B</u>asale <u>L</u>ineaire <u>A</u>lgebra Operatie<u>s</u>
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- <u>L</u>ocal <u>M</u>inimal <u>R</u>esiduals
- <u>G</u>eneralized <u>C</u>onjugate <u>R</u>esiduals
- Convergentie
- Krylov ruimte methoden

Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Basale **L**ineaire **A**lgebra operatie**S**

Basale **L**ineaire **A**lgebra operatie**S**:

- **MV**: $\mathbf{c} = \mathbf{A}\mathbf{u}$
- **AXPY** of **vector update**: $\mathbf{z} = \alpha\,\mathbf{x} + \mathbf{y}$
- **DOT** of **inproduct**: $(\mathbf{y}, \mathbf{z}) = \mathbf{z}^*\mathbf{y} = \mathbf{z} \bullet \mathbf{y}$

BLAS operaties kunnen snel uitgevoerd worden.

Computerfabrikanten zorgen ervoor dat BLAS software voor hun machine beschikbaar met een efficiëntie die voor hun machine geoptimaliseerd is.

Opmerking. De optimale MV is ook afhankelijk van de structuur van de matrix.

Basale Lineaire Algebra operatieS

BLAS operaties kunnen snel uitgevoerd worden.

We meten de kosten voor een operatie in

floating point operaties.

Floating point. $7.9487e{-}4 \quad = 794.87e{-}2$

Een **flop** is een scalaire operatie,
vermenigvuldiging $*$, deling $/$, optelling $+$, of aftrekking $-$.

In de numerieke lineaire algebra wordt vrijwel altijd
'n $+$ (of $-$) gecombineerd met 'n $*$.

Voorbeelden. 1 AXPY: $2n$ flop, 1 DOT: $2n$ flop
1 MV voor onze matrix op een 2-d gebied: $9n$ flop
3-d gebied: $13n$ flop

Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Gauss eliminatie (LU-decompositie)

$$
\begin{array}{rrcl}
 & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & 2x_1 + 3x_2 + 3x_3 & = & 5 \\
\frac{3}{4} & 3x_1 - 2x_2 - 7x_3 & = & 9
\end{array}
$$

Gauss eliminatie (LU-decompositie)

$$
\begin{array}{rrcl}
 & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & 2x_1 + 3x_2 + 3x_3 & = & 5 \\
\frac{3}{4} & 3x_1 - 2x_2 - 7x_3 & = & 9
\end{array}
$$

Gauss eliminatie (LU-decompositie)

$$
\begin{array}{rrcl}
 & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & 2x_1 + 3x_2 + 3x_3 & = & 5 \\
\frac{3}{4} & 3x_1 - 2x_2 - 7x_3 & = & 9
\end{array}
$$

$$
\begin{array}{rrcl}
 & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & 0x_1 + 1x_2 + 2x_3 & = & 3 \\
\frac{3}{4} & 3x_1 - 2x_2 - 7x_3 & = & 9
\end{array}
$$

Gauss eliminatie (LU-decompositie)

$$\begin{array}{rrcl} & 4x_1 + 4x_2 + 2x_3 & = & 6 \\ \frac{2}{4} & 2x_1 + 3x_2 + 3x_3 & = & 5 \\ \frac{3}{4} & 3x_1 - 2x_2 - 7x_3 & = & 9 \end{array}$$

$$\begin{array}{rrcl} & 4x_1 + 4x_2 + 2x_3 & = & 6 \\ \frac{2}{4} & 0x_1 + 1x_2 + 2x_3 & = & 3 \\ \frac{3}{4} & 0x_1 - 5x_2 - 8.5x_3 & = & 4.5 \end{array}$$

Gauss eliminatie (LU-decompositie)

$$
\begin{array}{rrcl}
 & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & 2x_1 + 3x_2 + 3x_3 & = & 5 \\
\frac{3}{4} & 3x_1 - 2x_2 - 7x_3 & = & 9
\end{array}
$$

$$
\begin{array}{rrrcl}
 & & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & & 0x_1 + 1x_2 + 2x_3 & = & 3 \\
\frac{3}{4} & -5 & 0x_1 - 5x_2 - 8.5x_3 & = & 4.5
\end{array}
$$

Gauss eliminatie (LU-decompositie)

$$\begin{array}{lrcl} & 4x_1 + 4x_2 + 2x_3 & = & 6 \\ \frac{2}{4} & 2x_1 + 3x_2 + 3x_3 & = & 5 \\ \frac{3}{4} & 3x_1 - 2x_2 - 7x_3 & = & 9 \end{array}$$

$$\begin{array}{llrcl} & & 4x_1 + 4x_2 + 2x_3 & = & 6 \\ \frac{2}{4} & & 0x_1 + 1x_2 + 2x_3 & = & 3 \\ \frac{3}{4} & -5 & 0x_1 - 5x_2 - 8.5x_3 & = & 4.5 \end{array}$$

Gauss eliminatie (LU-decompositie)

$$\begin{array}{lr} & 4x_1 + 4x_2 + 2x_3 = 6 \\ \frac{2}{4} & 2x_1 + 3x_2 + 3x_3 = 5 \\ \frac{3}{4} & 3x_1 - 2x_2 - 7x_3 = 9 \end{array}$$

$$\begin{array}{llr} & & 4x_1 + 4x_2 + 2x_3 = 6 \\ \frac{2}{4} & & 0x_1 + 1x_2 + 2x_3 = 3 \\ \frac{3}{4} & -5 & 0x_1 - 5x_2 - 8.5x_3 = 4.5 \end{array}$$

$$\begin{array}{llr} & & 4x_1 + 4x_2 + 2x_3 = 6 \\ \frac{2}{4} & & 0x_1 + 1x_2 + 2x_3 = 3 \\ \frac{3}{4} & -5 & 0x_1 + 0x_2 + 1.5x_3 = 19.5 \end{array}$$

Gauss eliminatie (LU-decompositie)

$$\begin{array}{lrcl}
 & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & 2x_1 + 3x_2 + 3x_3 & = & 5 \\
\frac{3}{4} & 3x_1 - 2x_2 - 7x_3 & = & 9
\end{array}$$

$$\begin{array}{llrcl}
 & & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & & 0x_1 + 1x_2 + 2x_3 & = & 3 \\
\frac{3}{4} & -5 & 0x_1 - 5x_2 - 8.5x_3 & = & 4.5
\end{array}$$

$$\begin{array}{llrcl}
 & & 4x_1 + 4x_2 + 2x_3 & = & 6 \\
\frac{2}{4} & & 0x_1 + 1x_2 + 2x_3 & = & 3 \\
\frac{3}{4} & -5 & 0x_1 + 0x_2 + 1.5x_3 & = & 19.5
\end{array}$$

Via terug substitutie volgt x_1, x_2, x_3

Gauss eliminatie (LU-decompositie)

$$\begin{array}{rl}
 & 4x_1 + 4x_2 + 2x_3 = 6 \\
\frac{2}{4} & 2x_1 + 3x_2 + 3x_3 = 5 \\
\frac{3}{4} & 3x_1 - 2x_2 - 7x_3 = 9
\end{array}$$

$$\begin{array}{rrl}
 & & 4x_1 + 4x_2 + 2x_3 = 6 \\
\frac{2}{4} & & 0x_1 + 1x_2 + 2x_3 = 3 \\
\frac{3}{4} & -5 & 0x_1 - 5x_2 - 8.5x_3 = 4.5
\end{array}$$

$$\begin{array}{rrl}
 & & 4x_1 + 4x_2 + 2x_3 = 6 \\
\frac{2}{4} & & 0x_1 + 1x_2 + 2x_3 = 3 \\
\frac{3}{4} & -5 & 0x_1 + 0x_2 + 1.5x_3 = 19.5
\end{array}$$

Gauss: voorwaartse eliminatie, terugwaartse substitutie

Gauss eliminatie (LU-decompositie)

In matrix taal: $\mathbf{Ax} = \mathbf{b}$

$$\begin{bmatrix} 4 & 4 & 2 \\ 2 & 3 & 3 \\ 3 & -2 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 9 \end{bmatrix}$$

Gauss eliminatie (LU-decompositie)

In matrix taal: $\mathbf{Ax} = \mathbf{b}$

$$\begin{bmatrix} 4 & 4 & 2 \\ 2 & 3 & 3 \\ 3 & -2 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 9 \end{bmatrix}$$

LU-decompositie: $\mathbf{A} = \mathbf{LU}$:

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{2}{4} & 1 & 0 \\ \frac{3}{4} & -5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 9 \end{bmatrix}$$

Gauss eliminatie (LU-decompositie)

In matrix taal: $\mathbf{Ax} = \mathbf{b}$

$$\begin{bmatrix} 4 & 4 & 2 \\ 2 & 3 & 3 \\ 3 & -2 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 9 \end{bmatrix}$$

LU-decompositie: $\mathbf{A} = \mathbf{LU}$:

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{2}{4} & 1 & 0 \\ \frac{3}{4} & -5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \\ 9 \end{bmatrix}$$

Het bepalen van $\mathbf{L}$ en $\mathbf{U}$ en het oplossen van $\mathbf{Ly} = \mathbf{b}$ correspondeert met voorwaartse eliminatie,
het oplossen van $\mathbf{Ux} = \mathbf{y}$ is terugwaartse substitutie

Gauss eliminatie (LU-decompositie)

LU-decompositie $\mathbf{A} = \mathbf{L}\mathbf{U}$, waarbij

$\mathbf{L}$ beneden driehoek met diag($\mathbf{L}$) $= \mathbf{I}$ en
$\mathbf{U}$ bovendriehoek.

De collectie (i, j) met $\mathbf{A}_{i,j} = 0$,
en $\mathbf{L}_{i,j} \neq 0$ (als $i < j$) of $\mathbf{U}_{i,j} \neq 0$ (als $i \geq j$) heet **fill**.

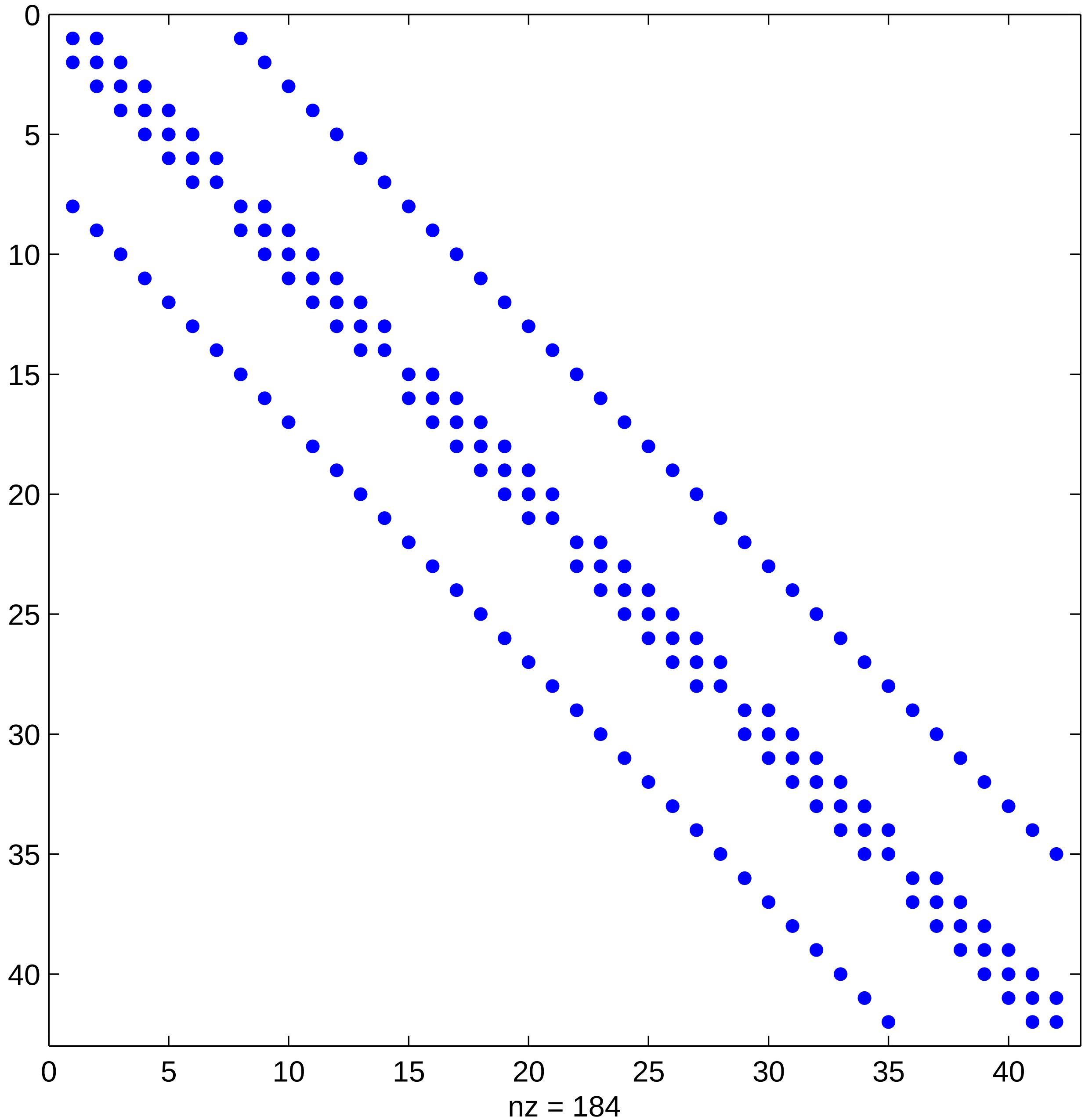
Gauss eliminatie (LU-decompositie)

LU-decompositie $\mathbf{A} = \mathbf{L}\mathbf{U}$, waarbij

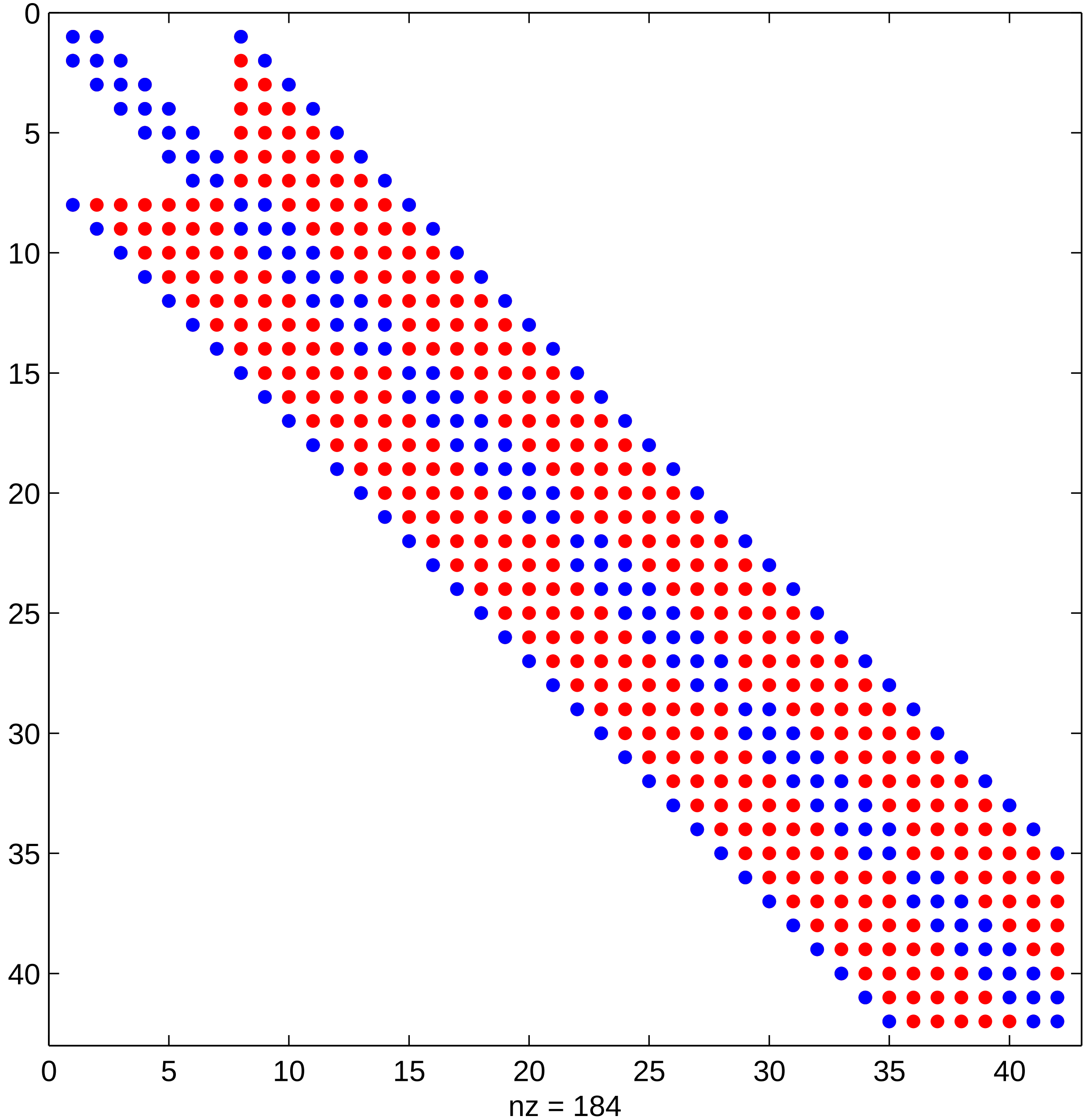
$\mathbf{L}$ beneden driehoek met $\mathrm{diag}(\mathbf{L}) = \mathbf{I}$ en
$\mathbf{U}$ bovendriehoek.

De collectie (i, j) met $\mathbf{A}_{i,j} = 0$,
en $\mathbf{L}_{i,j} \neq 0$ (als $i < j$) of $\mathbf{U}_{i,j} \neq 0$ (als $i \geq j$) heet **fill**.

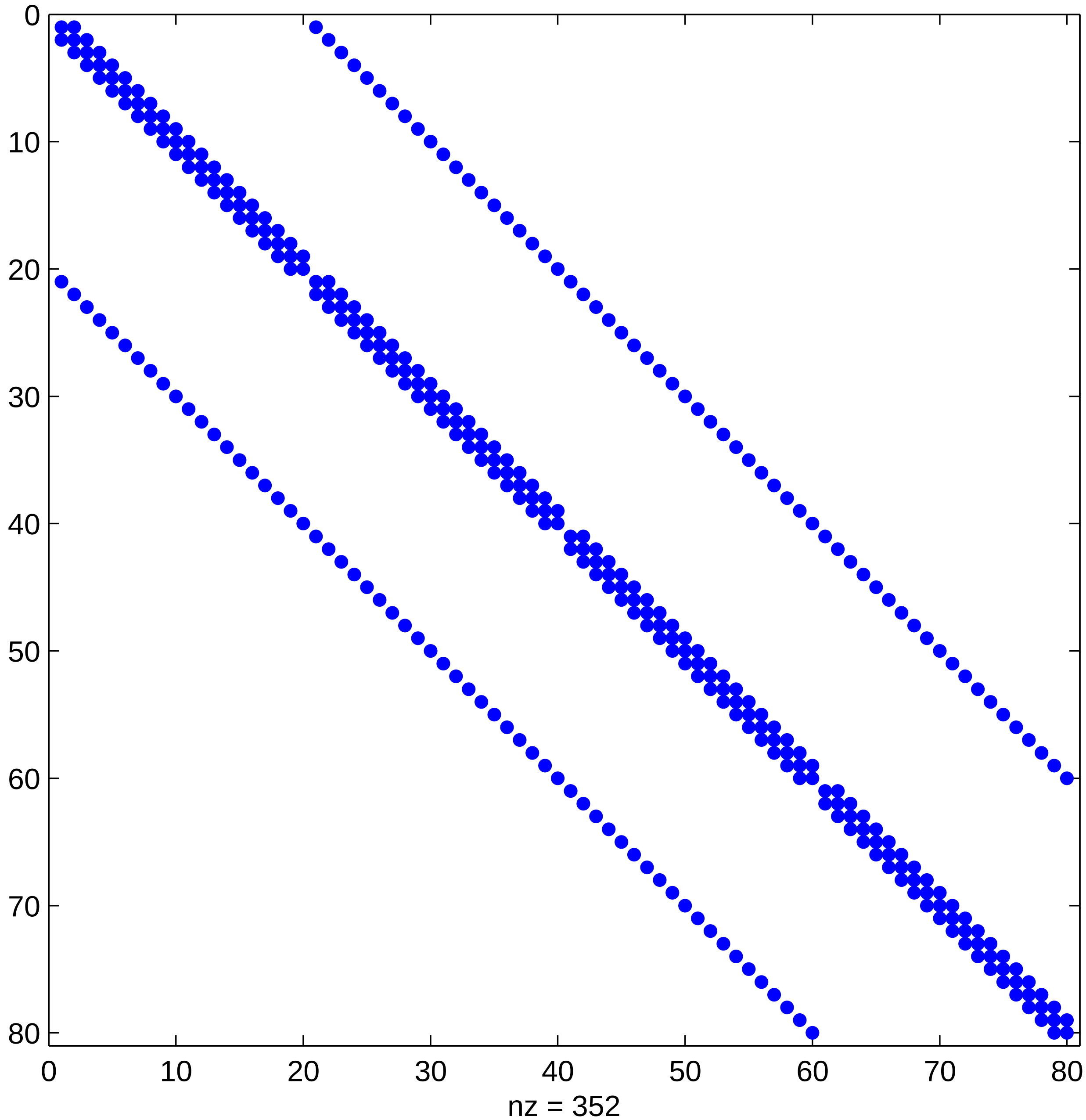
Gauss eliminatie (of LU-decompositie) is inefficiënt
als $\mathbf{A}$ **ijl** is (d.w.z., per rij maar een paar $\mathbf{A}_{i,j} \neq 0$)
en er veel fill optreedt.



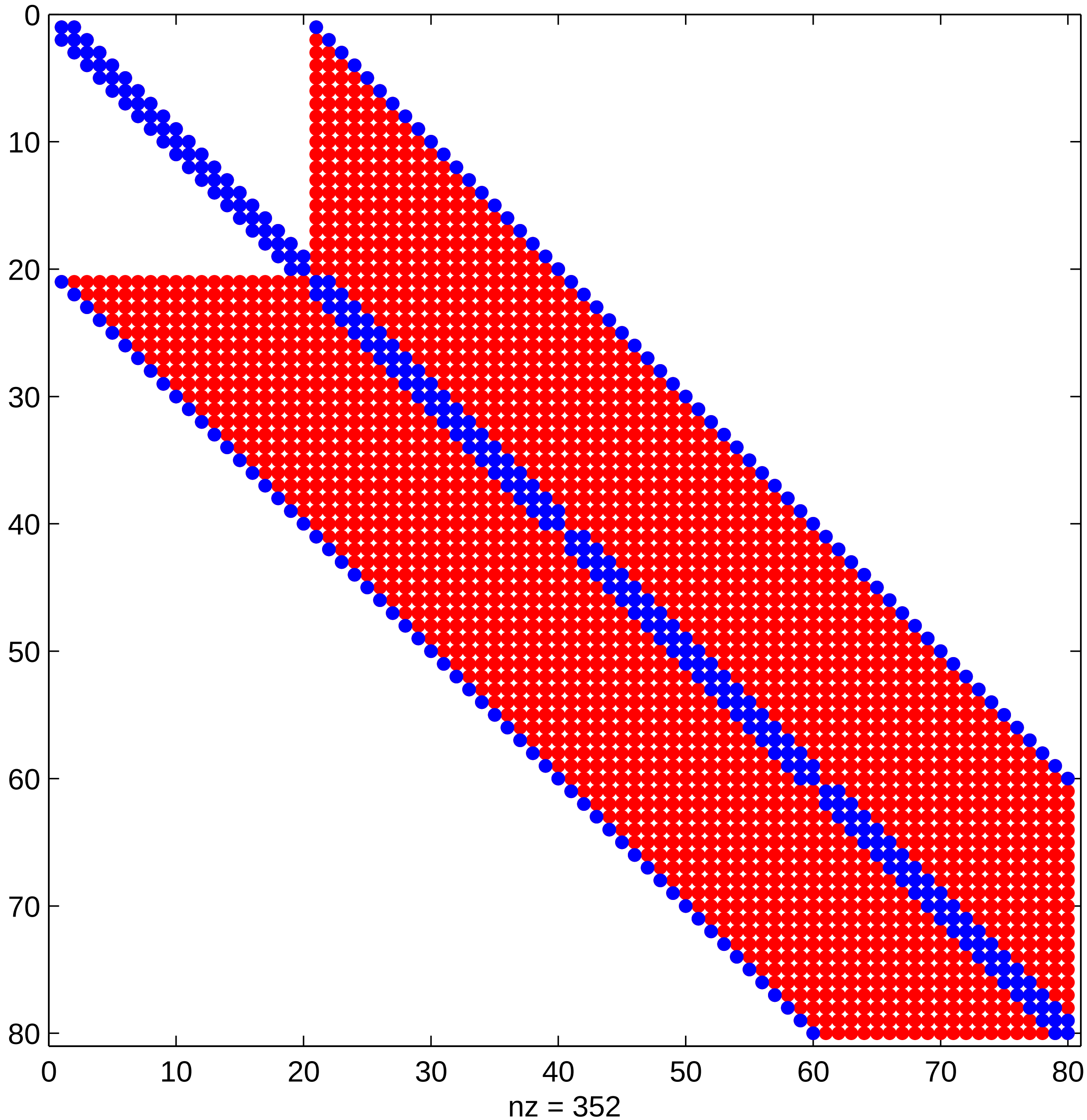
0
5
10
15
20
25
30
35
40
nz = 184



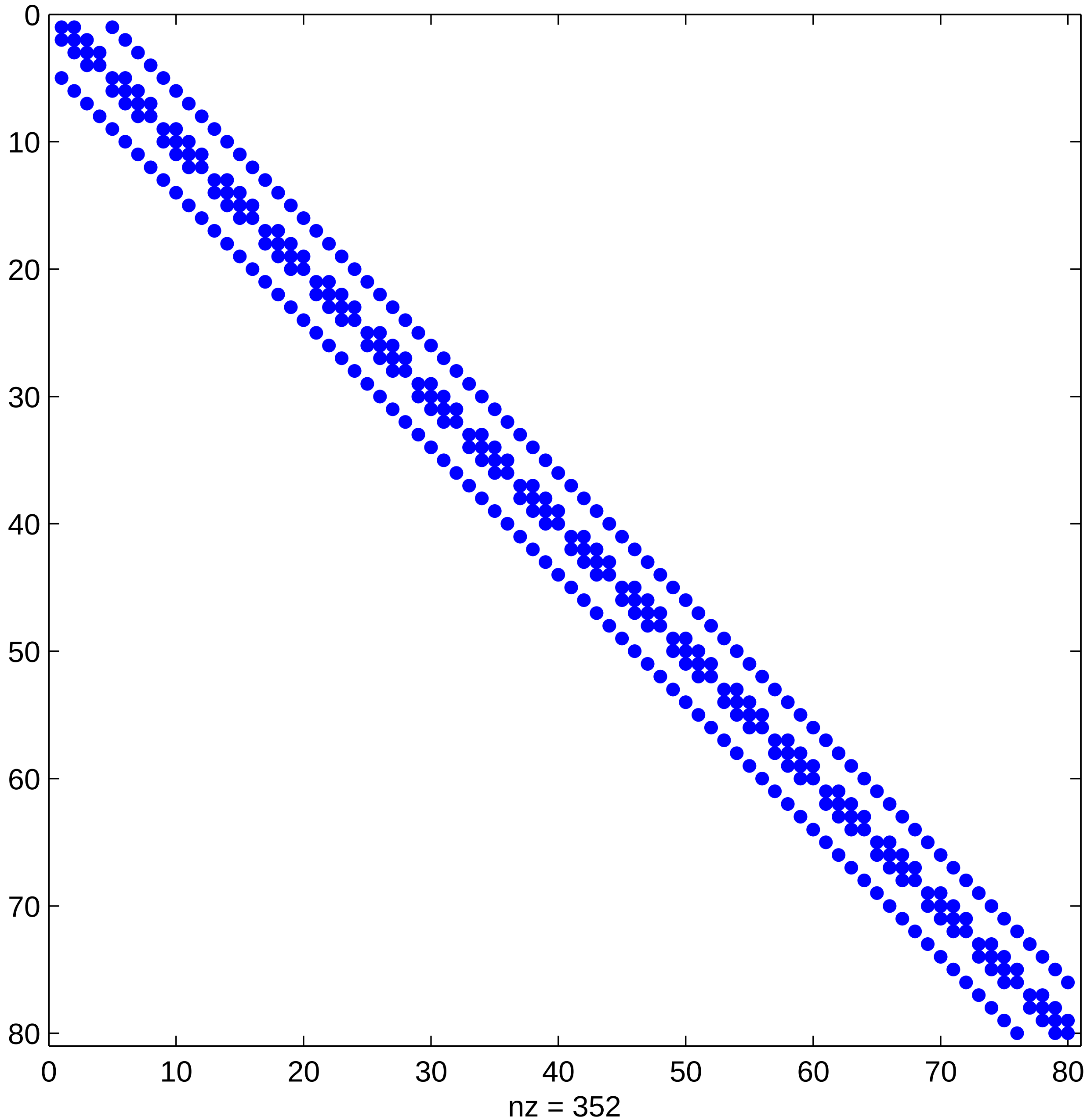
0
5
10
15
20
25
30
35
40
0
5
10
15
20
25
30
35
40
nz = 184



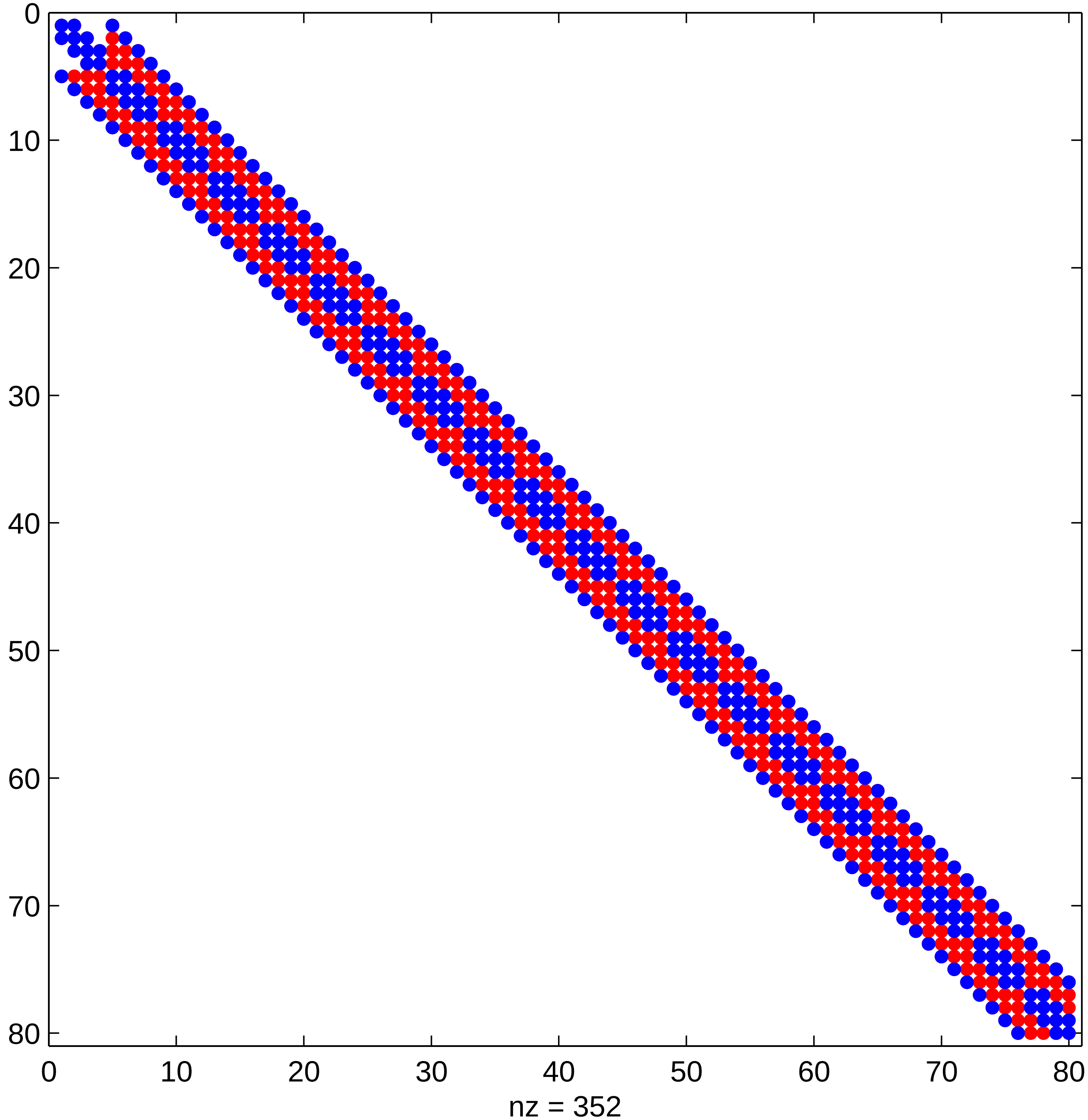
0
10
20
30
40
50
60
70
80
nz = 352



0
10
20
30
40
50
60
70
80
0
10
20
30
40
50
60
70
80
nz = 352



0
10
20
30
40
50
60
70
80
nz = 352


0
10
20
30
40
50
60
70
80
nz = 352

Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Gauss eliminatie (LU-decompositie)

Voor Grondwaterstroming/verspreiding van gif

Methode		Geheugen (in 8 bytes)	Werk (in flop)
MV	2d	$5(n_x n_y)$	$9(n_x n_y)$
Gauss	2d	$\approx 2n_x(n_x n_y)$	$2n_x^2(n_x n_y)$
MV	3d	$7(n_x n_y n_z)$	$13(n_x n_y n_z)$
Gauss	3d	$\approx 2(n_x n_y)(n_x n_y n_z)$	$2(n_x n_y)^2(n_x n_y n_z)$

Met $n_x = n_y = 300$ hebben we het over het verschil in opslag voor een matrix versus en L- en een U-factor

in 2-d: 3.6 Mb versus 430 Mb

in 3-d: 1.5 Gb versus 39 Tb

Gauss eliminatie (LU-decompositie)

Voor Grondwaterstroming/verspreiding van gif

`Methode`		Geheugen (in 8 bytes)	Werk (in flop)
MV	2d	$5(n_x n_y)$	$9(n_x n_y)$
Gauss	2d	$\approx 2n_x(n_x n_y)$	$2n_x^2(n_x n_y)$
MV	3d	$7(n_x n_y n_z)$	$13(n_x n_y n_z)$
Gauss	3d	$\approx 2(n_x n_y)(n_x n_y n_z)$	$2(n_x n_y)^2(n_x n_y n_z)$

Met $n_x = n_y = 300$ hebben we het over het verschil in rekentijd voor een MV versus Gauss eliminatie

in 2-d: 1 seconde versus 35 minuten

in 3-d: 1 seconde versus 40 jaar

Gauss eliminatie (LU-decompositie)

Voor Grondwaterstroming/verspreiding van gif

`Methode`		Geheugen (in 8 bytes)	Werk (in flop)
MV	2d	$5(n_x n_y)$	$9(n_x n_y)$
Gauss	2d	$\approx 2n_x(n_x n_y)$	$2n_x^2(n_x n_y)$
MV	3d	$7(n_x n_y n_z)$	$13(n_x n_y n_z)$
Gauss	3d	$\approx 2(n_x n_y)(n_x n_y n_z)$	$2(n_x n_y)^2(n_x n_y n_z)$

Gauss. + Maar één methode
+ Geeft exact antwoord
− **fill**: (duur mbt geheugen, rekenkosten)

Iteratief. − Methode per probleemtype te kiezen
− Benaderend antwoord
+ **alleen MV's, axpy's, dot's**

Gauss eliminatie (LU-decompositie)

Voor Grondwaterstroming/verspreiding van gif

Methode		Geheugen (in 8 bytes)	Werk (in flop)
MV	2d	$5(n_x n_y)$	$9(n_x n_y)$
Gauss	2d	$\approx 2n_x(n_x n_y)$	$2n_x^2(n_x n_y)$
MV	3d	$7(n_x n_y n_z)$	$13(n_x n_y n_z)$
Gauss	3d	$\approx 2(n_x n_y)(n_x n_y n_z)$	$2(n_x n_y)^2(n_x n_y n_z)$

Gauss. + Maar één methode (andere nummering als $n_y < n_x$?)
+ Geeft exact antwoord (je maakt ook afrondfouten!)
— fill: (duur mbt geheugen, rekenkosten)

Iteratief. – Methode per probleemtype te kiezen
– Benaderend antwoord (Kan heel nauwkeurig!)
+ alleen MV's, axpy's, dot's

Gauss eliminatie (LU-decompositie)

Voor Grondwaterstroming/verspreiding van gif

Methode		Geheugen (in 8 bytes)	Werk (in flop)
MV	2d	$5(n_x n_y)$	$9(n_x n_y)$
Gauss	2d	$\approx 2n_x(n_x n_y)$	$2n_x^2(n_x n_y)$
MV	3d	$7(n_x n_y n_z)$	$13(n_x n_y n_z)$
Gauss	3d	$\approx 2(n_x n_y)(n_x n_y n_z)$	$2(n_x n_y)^2(n_x n_y n_z)$

A is $n \times n$.

Vuistregel: (beste) Gauss is sneller dan (beste) iteratief

in 2 d als $n < 60\,000$ $\quad n_x = n_y = 240$

in 3 d als $n < 40\,000$ $\quad n_x = n_y = 65,\ n_z = 10$

Vuistregel geldt voor "algemene" praktische problemen

Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Iteratief

Observatie. We zijn niet echt geïnteresseerd in de **exacte** oplossing $\mathbf{x}$. We zijn al tevreden als de berekende oplossing goed is in pakweg 5 decimalen.

Iteratief

Observatie. We zijn niet echt geïnteresseerd in de **exacte** oplossing $\mathbf{x}$. We zijn al tevreden als de berekende oplossing goed is in pakweg 5 decimalen.

Idee. Als we een **benaderende** oplossing $\mathbf{x}_k$ hebben, construeer dan een beter benaderende oplossing uit $\mathbf{x}_k$ (en eventueel uit $\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots$ of andere vectoren die we al berekend hebben). Herhaal dit proces tot dat

de **fout** $\mathbf{e}_k \equiv \mathbf{x} - \mathbf{x}_k$ klein genoeg is.

Gebruik alleen **B**asale **L**ineaire **A**lgebra operatie**S**:

- **MV**s: $\mathbf{c} = \mathbf{A}\mathbf{u}$
- **AXPY**s of **vector update**s: $\mathbf{z} = \alpha\,\mathbf{x} + \mathbf{y}$
- **DOT**s of **inproduct**en: $\alpha = (\mathbf{y}, \mathbf{z})$

Iteratief

Observatie. We zijn niet echt geïnteresseerd in de **exacte** oplossing $\mathbf{x}$. We zijn al tevreden als de berekende oplossing goed is in pakweg 5 decimalen.

Idee. Als we een **benaderende** oplossing $\mathbf{x}_k$ hebben, construeer dan een beter benaderende oplossing uit $\mathbf{x}_k$ (en eventueel uit $\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots$ of andere vectoren die we al berekend hebben). Herhaal dit proces tot dat

de **fout** $\mathbf{e}_k \equiv \mathbf{x} - \mathbf{x}_k$ klein genoeg is.

Probleem. Hoe bepaal je de fout?

Iteratief

Observatie. We zijn niet echt geïnteresseerd in de **exacte** oplossing $\mathbf{x}$. We zijn al tevreden als de berekende oplossing goed is in pakweg 5 decimalen.

Idee. Als we een **benaderende** oplossing $\mathbf{x}_k$ hebben, construeer dan een beter benaderende oplossing uit $\mathbf{x}_k$ (en eventueel uit $\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots$ of andere vectoren die we al berekend hebben). Herhaal dit proces tot dat

de **fout** $\mathbf{e}_k \equiv \mathbf{x} - \mathbf{x}_k$ klein genoeg is.

Terminologie. **Residu:** $\mathbf{r}_k \equiv \mathbf{A}(\mathbf{x} - \mathbf{x}_k) = \mathbf{b} - \mathbf{A}\mathbf{x}_k$.

Residu kan berekend worden.
Stop als het residu klein genoeg is.

Iteratief

Observatie. We zijn niet echt geïnteresseerd in de **exacte** oplossing $\mathbf{x}$. We zijn al tevreden als de berekende oplossing goed is in pakweg 5 decimalen.

Idee. Als we een **benaderende** oplossing $\mathbf{x}_k$ hebben, construeer dan een beter benaderende oplossing uit $\mathbf{x}_k$ (en eventueel uit $\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots$ of andere vectoren die we al berekend hebben). Herhaal dit proces tot dat

de **fout** $\mathbf{e}_k \equiv \mathbf{x} - \mathbf{x}_k$ klein genoeg is.

Terminologie. **Residu:** $\mathbf{r}_k \equiv \mathbf{A}(\mathbf{x} - \mathbf{x}_k) = \mathbf{b} - \mathbf{A}\mathbf{x}_k$.

Probleem. Wat betekent "klein genoeg"?

Iteratief

Observatie. We zijn niet echt geïnteresseerd in de **exacte** oplossing $\mathbf{x}$. We zijn al tevreden als de berekende oplossing goed is in pakweg 5 decimalen.

Idee. Als we een **benaderende** oplossing $\mathbf{x}_k$ hebben, construeer dan een beter benaderende oplossing uit $\mathbf{x}_k$ (en eventueel uit $\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots$ of andere vectoren die we al berekend hebben). Herhaal dit proces tot dat

de **fout** $\mathbf{e}_k \equiv \mathbf{x} - \mathbf{x}_k$ klein genoeg is.

Terminologie. **Residu:** $\mathbf{r}_k \equiv \mathbf{A}(\mathbf{x} - \mathbf{x}_k) = \mathbf{b} - \mathbf{A}\mathbf{x}_k$.

Meet de fout, residu in de (geschaalde) 2-norm:

$$\|\mathbf{r}\|_2 \equiv \sqrt{(\mathbf{r}, \mathbf{r})} \text{ met } (\mathbf{y}, \mathbf{z}) = \textstyle\sum_{j=1}^{n} \overline{z_j}\, y_j = \mathbf{z}^*\mathbf{y} = \overline{\mathbf{z}}^{\mathsf{T}}\mathbf{y}.$$

Iteratief

Observatie. We zijn niet echt geïnteresseerd in de **exacte** oplossing $\mathbf{x}$. We zijn al tevreden als de berekende oplossing goed is in pakweg 5 decimalen.

Idee. Als we een **benaderende** oplossing $\mathbf{x}_k$ hebben, construeer dan een beter benaderende oplossing uit $\mathbf{x}_k$ (en eventueel uit $\mathbf{x}_{k-1}, \mathbf{x}_{k-2}, \ldots$ of andere vectoren die we al berekend hebben). Herhaal dit proces tot dat

de **fout** $\mathbf{e}_k \equiv \mathbf{x} - \mathbf{x}_k$ klein genoeg is.

Terminologie. **Residu:** $\mathbf{r}_k \equiv \mathbf{A}(\mathbf{x} - \mathbf{x}_k) = \mathbf{b} - \mathbf{A}\mathbf{x}_k$.

Meet de fout of residu in de (geschaalde) 2-norm:

$$\|\mathbf{r}\|_2 \equiv \sqrt{(\mathbf{r}, \mathbf{r})} \text{ met } (\mathbf{y}, \mathbf{z}) = \sum_{j=1}^{n} \frac{1}{n}\, \overline{z_j} y_j = \frac{1}{n}\mathbf{z}^* \mathbf{y}.$$

Iteratief

Kies α $$\mathbf{x}_{\text{opl}} = \mathbf{x}_{\text{opl}} + \alpha(\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{opl}})$$

Probleem van de vorm $\mathbf{x} = \phi(\mathbf{x})$.

Probeer op te lossen met successieve substitutie:

$$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k) = \mathbf{x}_k + \alpha\, \mathbf{r}_k$$

Iteratief

Richardson iteration

Choose $tol > 0$, $\mathbf{x}_0$, $k_{\max}$, α
Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
For $k = 0, 1, 2, \ldots, k_{\max}$
    Stop if $\|\mathbf{r}_k\|_2 \leq tol \|\mathbf{b}\|_2$
    $\mathbf{u}_k = \mathbf{r}_k$
    $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$
    Determine α
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\, \mathbf{u}_k$
    $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha\, \mathbf{c}_k$
end for

Iteratief

Richardson iteration

```
Choose tol > 0, x_0, k_max, α
Compute r_0 = b − A x_0
For k = 0, 1, 2, ..., k_max
    Stop if ||r_k||_2 ≤ tol ||b||_2
    u_k = r_k
    c_k = A u_k
    Determine α
    x_{k+1} = x_k + α u_k
    r_{k+1} = r_k − α c_k
end for
```

Iteratief

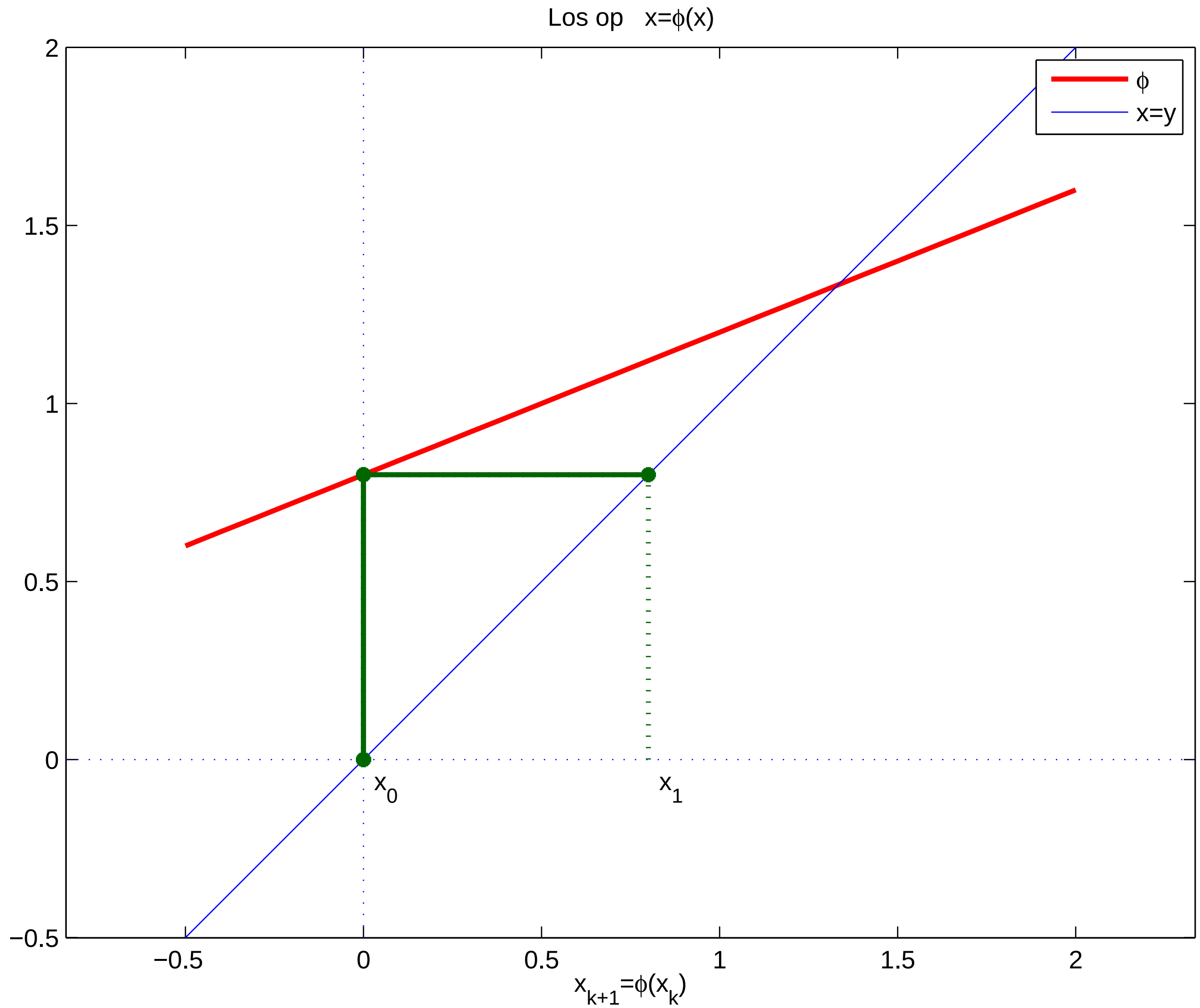
Kies α $$\mathbf{x}_{\text{opl}} = \mathbf{x}_{\text{opl}} + \alpha(\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{opl}})$$

Probleem van de vorm $\mathbf{x} = \phi(\mathbf{x})$.

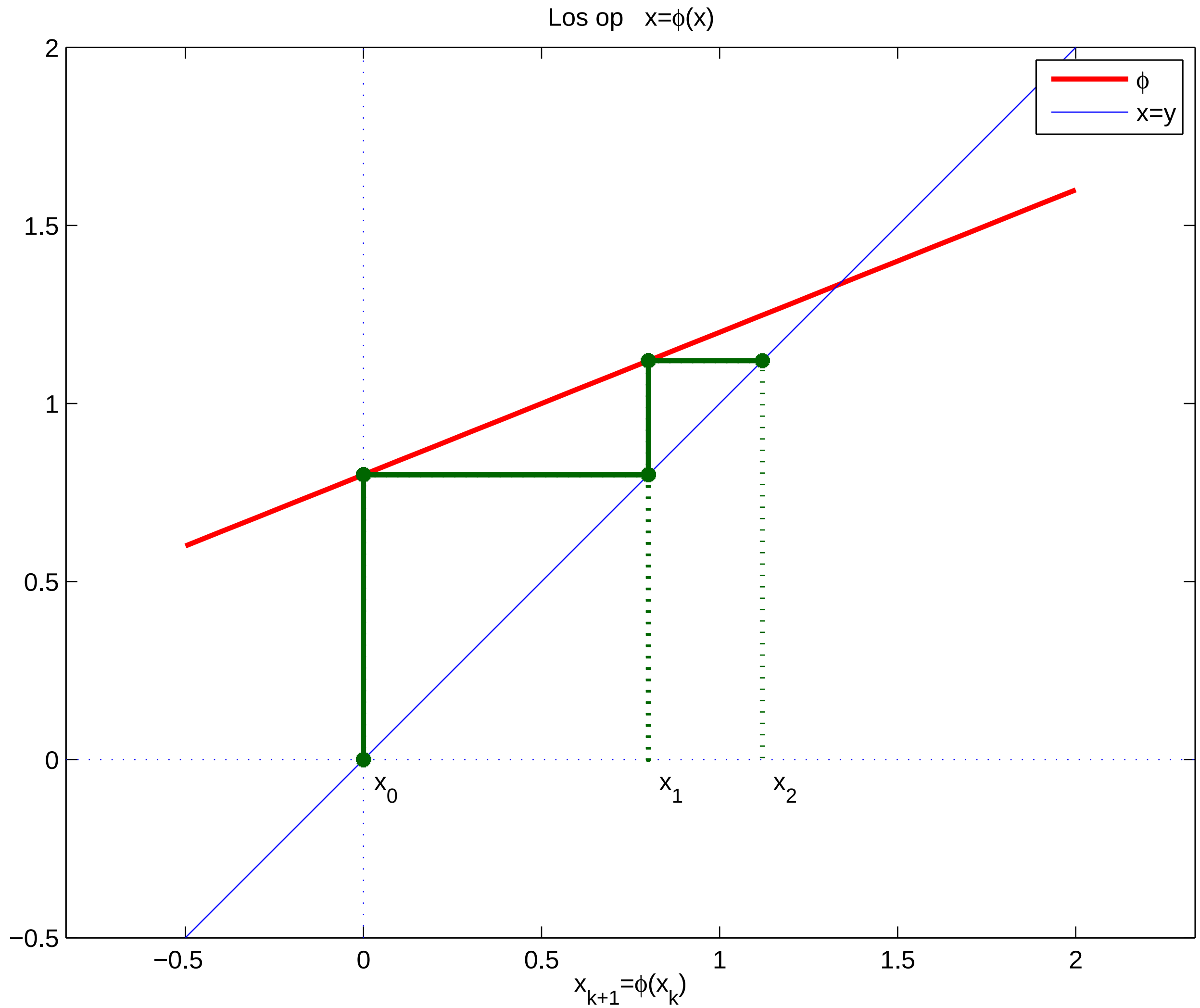
Probeer op te lossen met successieve substitutie:

$$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k) = \mathbf{x}_k + \alpha\,\mathbf{r}_k$$

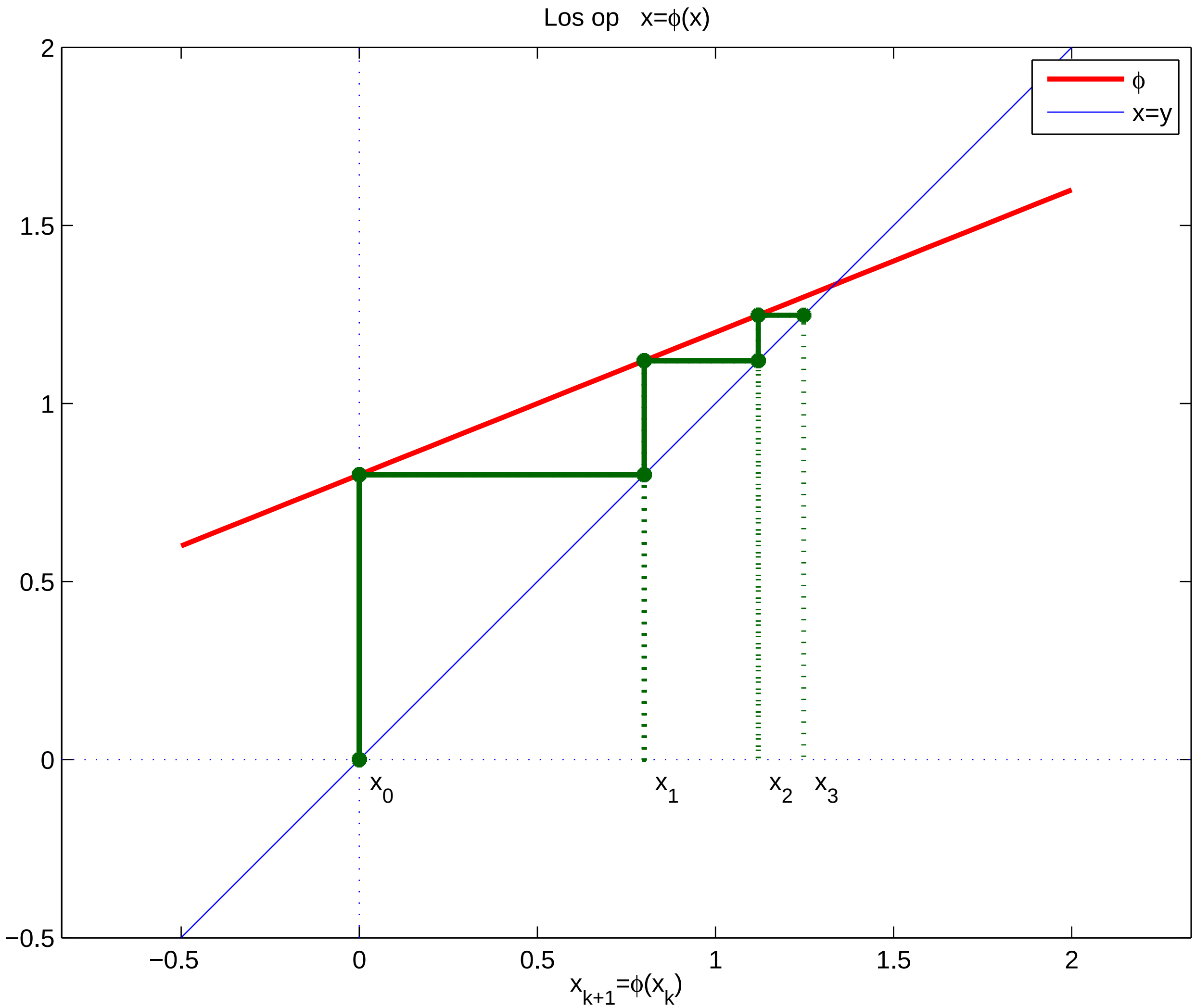
Convergeert dit (d.w.z., $\|\mathbf{r}_k\|_2 \to 0$ als $k \to \infty$)?



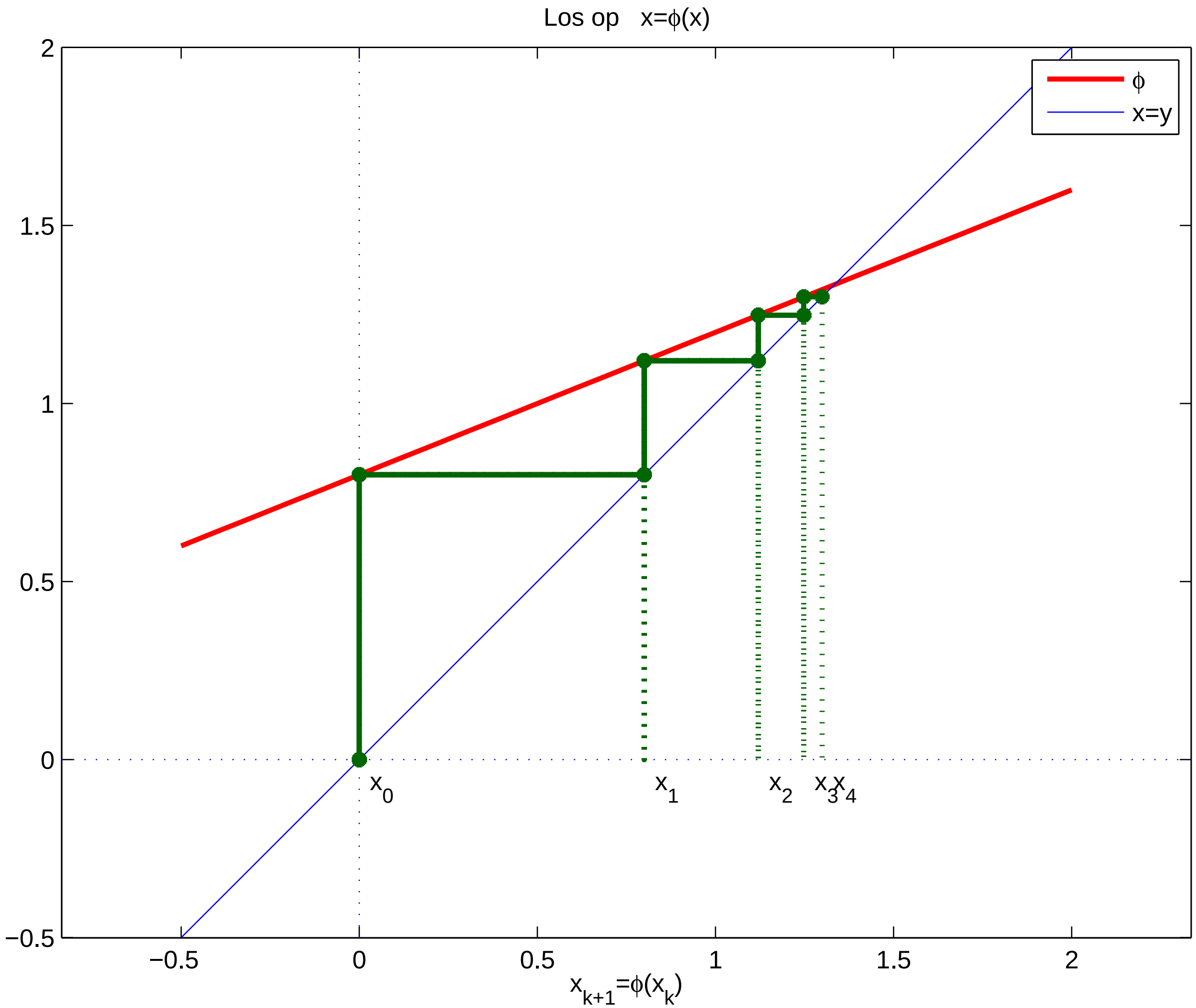
Los op x=ϕ(x)
ϕ
x=y
2
1.5
1
0.5
0
−0.5
−0.5
0
0.5
1
1.5
2
x_0
x_1
$x_{k+1}=\phi(x_k)$



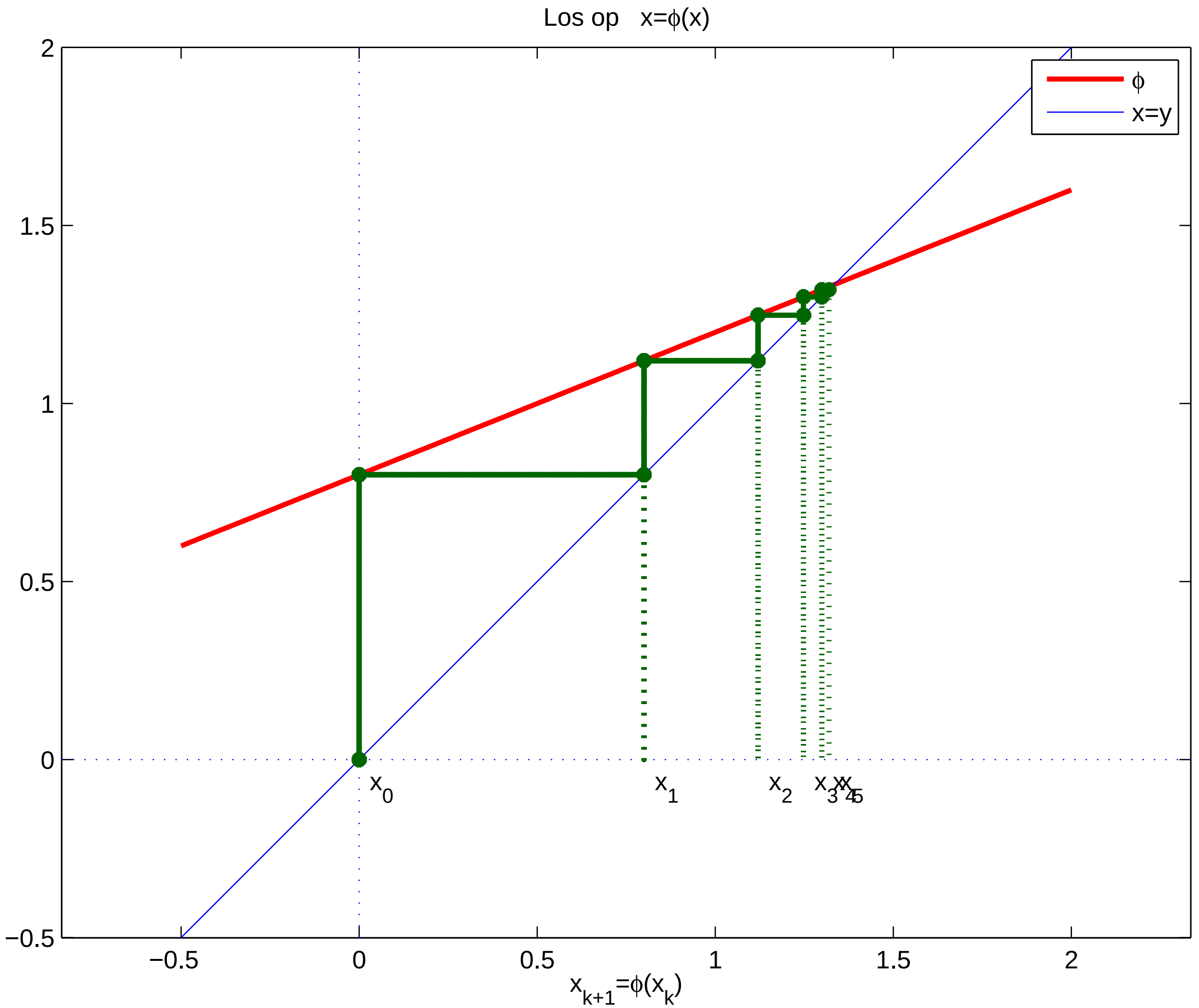
Los op x=ϕ(x)
ϕ
x=y
2
1.5
1
0.5
0
−0.5
−0.5
0
0.5
1
1.5
2
x0
x1
x2
xk+1=ϕ(xk)

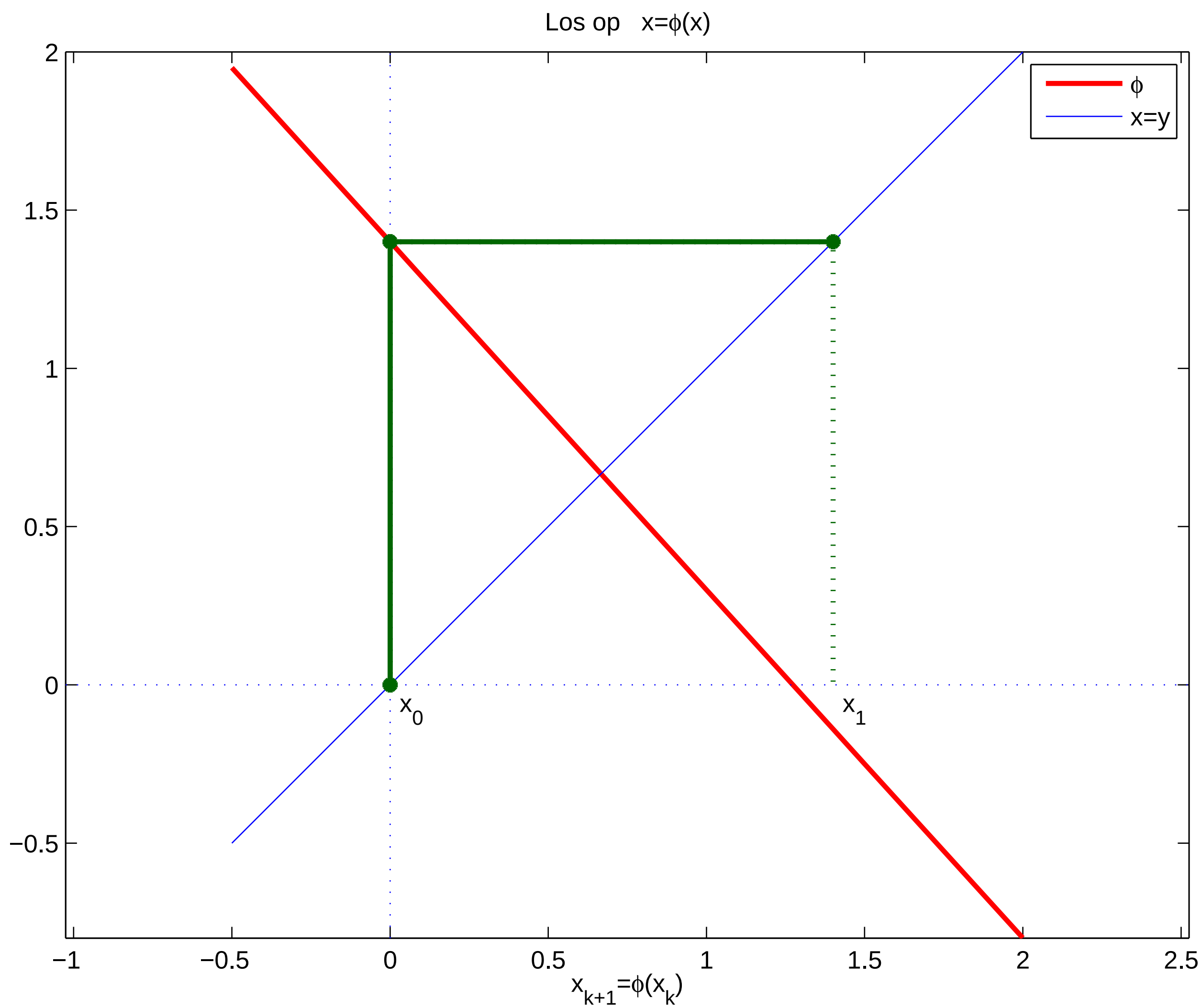


Los op x=ϕ(x)
ϕ
x=y
2
1.5
1
0.5
0
−0.5
−0.5
0
0.5
1
1.5
2
x_0
x_1
x_2
x_3
$x_{k+1}=\phi(x_k)$

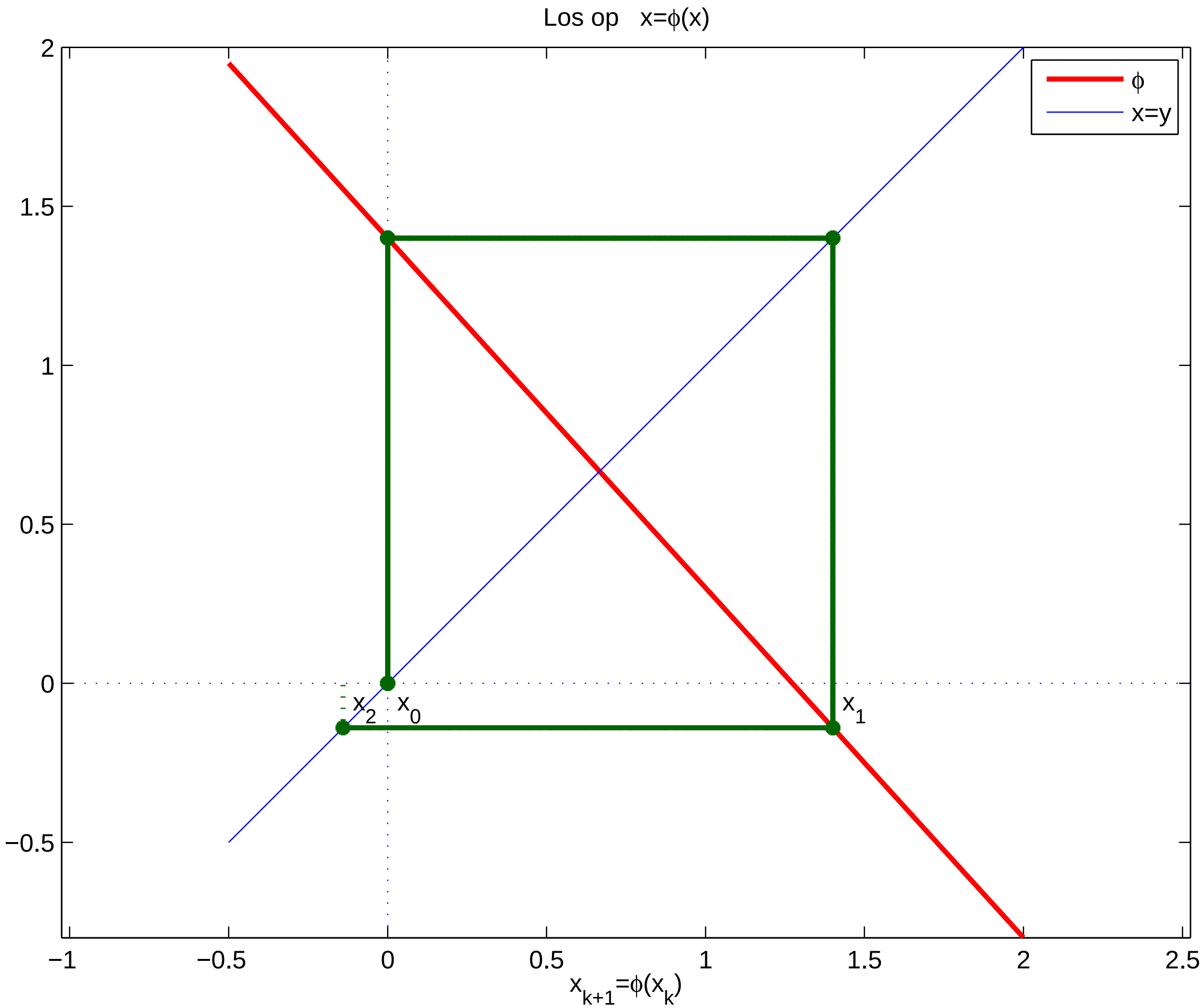


Los op x=ϕ(x)
ϕ
x=y
2
1.5
1
0.5
0
−0.5
−0.5
0
0.5
1
1.5
2
x_0
x_1
x_2
x_3
x_4
$x_{k+1}=\phi(x_k)$

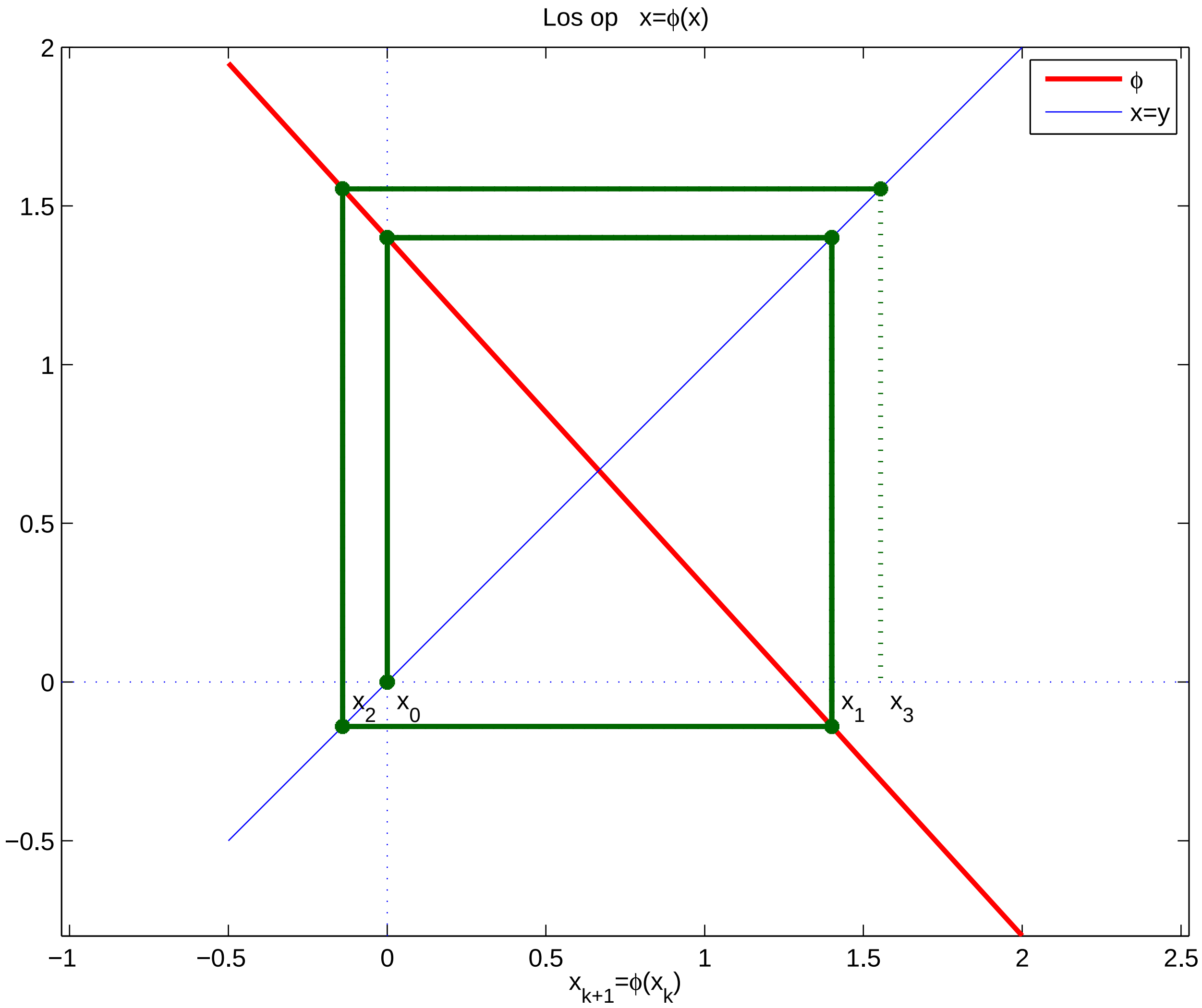

Los op x=ϕ(x)
ϕ
x=y
2
1.5
1
0.5
0
−0.5
−0.5
0
0.5
1
1.5
2
x0
x1
x2
x3
x4
x5
xk+1=ϕ(xk)



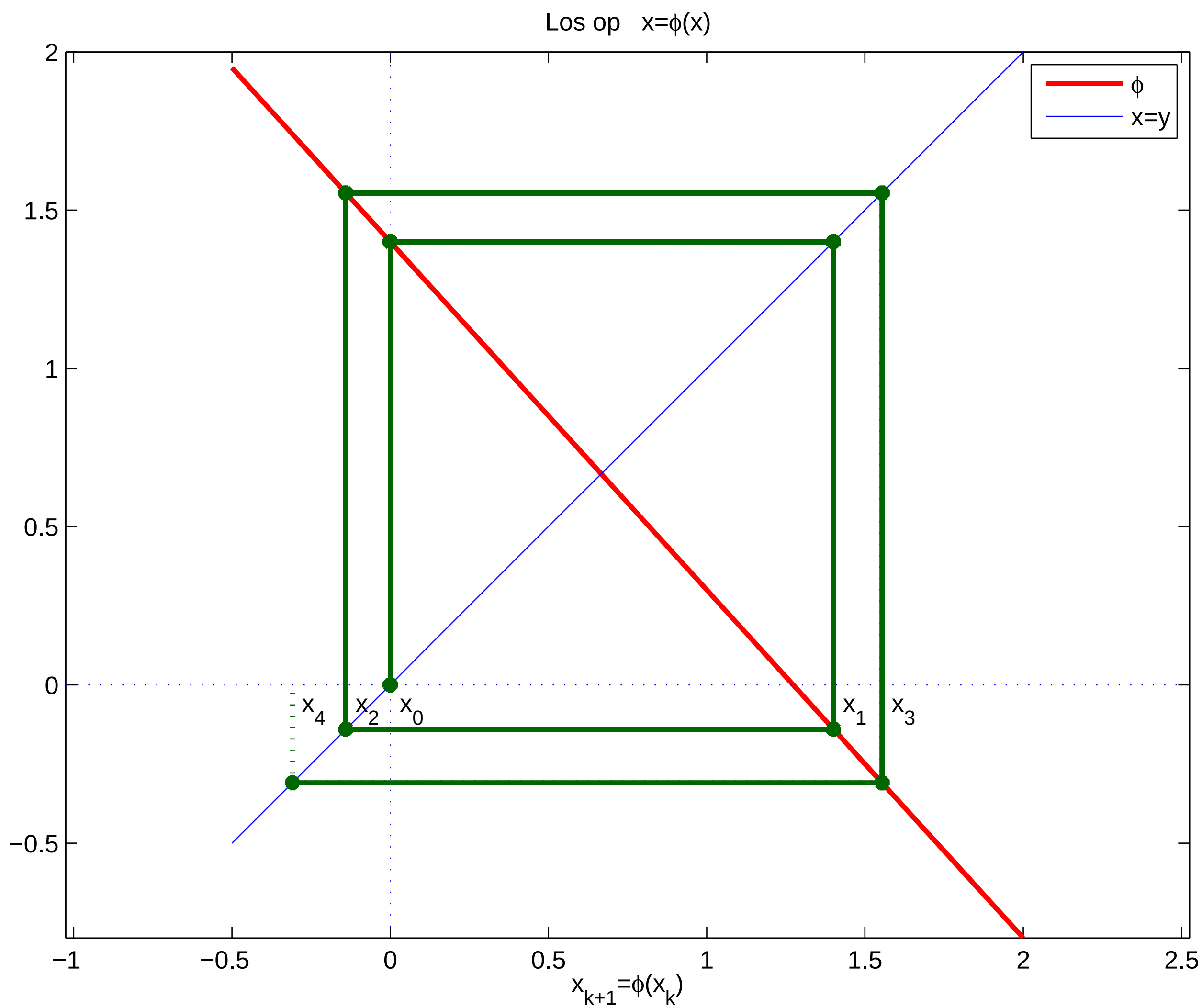
Los op   x=ϕ(x)
ϕ
x=y
x0
x1
2
1.5
1
0.5
0
−0.5
−1
−0.5
0
0.5
1
1.5
2
2.5
xk+1=ϕ(xk)



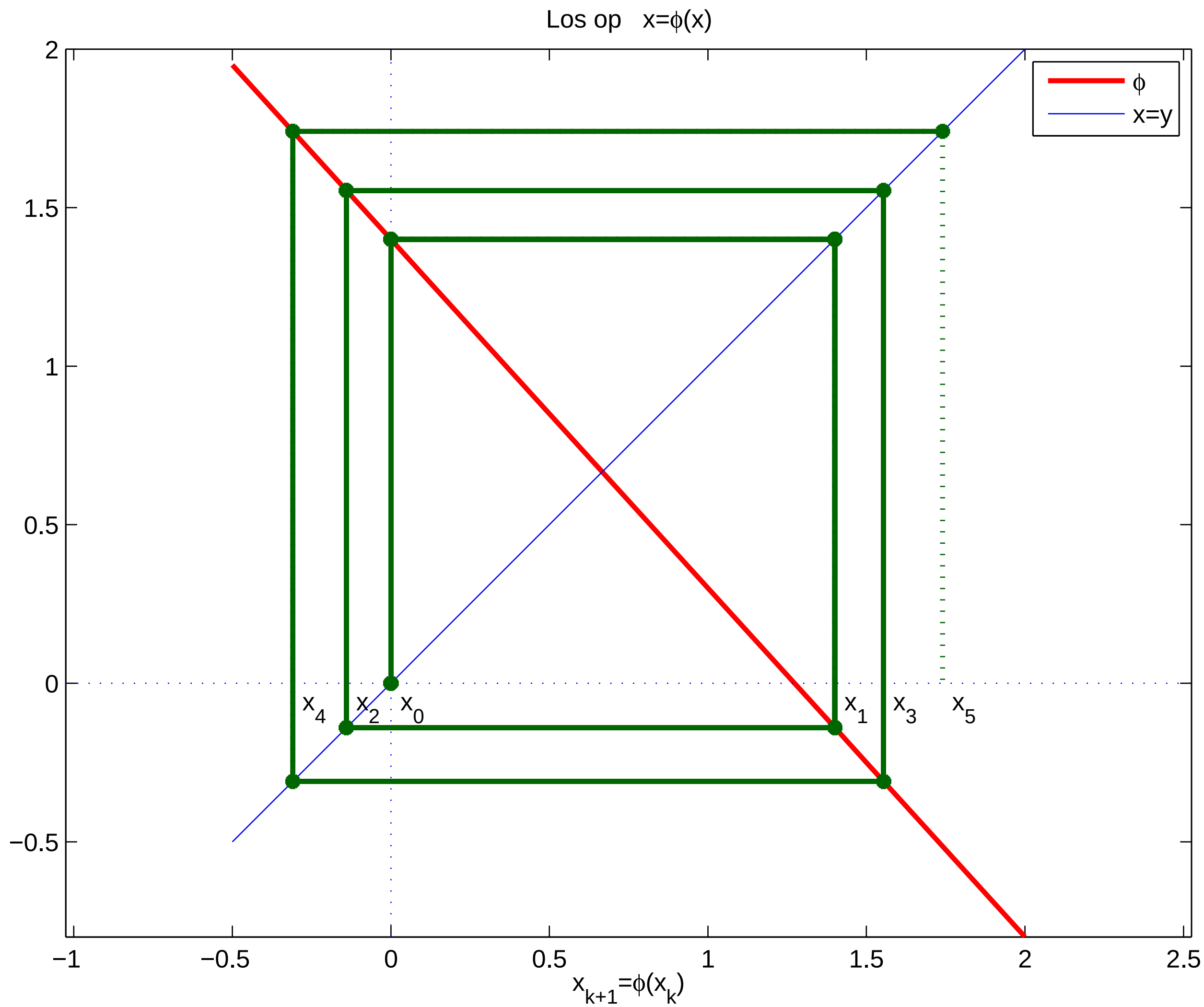
Los op x=ϕ(x)
2
1.5
1
0.5
0
−0.5
−1
−0.5
0
0.5
1
1.5
2
2.5
ϕ
x=y
x2
x0
x1
xk+1=ϕ(xk)



Los op x=ϕ(x)
ϕ
x=y
x2
x0
x1
x3
2
1.5
1
0.5
0
−0.5
−1
−0.5
0
0.5
1
1.5
2
2.5
x k+1=ϕ(x k)



Los op   x=ϕ(x)
ϕ
x=y
x_4
x_2
x_0
x_1
x_3
2
1.5
1
0.5
0
−0.5
−1
−0.5
0
0.5
1
1.5
2
2.5
x_{k+1}=ϕ(x_k)



Los op x=ϕ(x)
ϕ
x=y
2
1.5
1
0.5
0
−0.5
−1
−0.5
0
0.5
1
1.5
2
2.5
x4
x2
x0
x1
x3
x5
xk+1=ϕ(xk)

Iteratief

Kies α
$$\mathbf{x}_{\text{opl}} = \mathbf{x}_{\text{opl}} + \alpha(\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{opl}})$$

Probleem van de vorm $\mathbf{x} = \phi(\mathbf{x})$.

Probeer op te lossen met successieve substitutie:
$$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k) = \mathbf{x}_k + \alpha\,\mathbf{r}_k$$

$$\mathbf{e}_{k+1} = \mathbf{e}_k - \alpha\,\mathbf{r}_k = (\mathbf{I} - \alpha\mathbf{A})\mathbf{e}_k.$$

Voorbeeld. $n = 1$, $\mathbf{A} = [\lambda]$.

Dan $$\mathbf{e}_{k+1} = (1 - \alpha\,\lambda)\,\mathbf{e}_k = (1 - \alpha\,\lambda)^{k+1}\mathbf{e}_0 \to 0$$
$$\Leftrightarrow \qquad |1 - \alpha\,\lambda| < 1.$$

Iteratief

Kies α $\qquad \mathbf{x}_{\text{opl}} = \mathbf{x}_{\text{opl}} + \alpha(\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{opl}})$

Probleem van de vorm $\mathbf{x} = \phi(\mathbf{x})$.

Probeer op te lossen met successieve substitutie:

$$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k) = \mathbf{x}_k + \alpha\,\mathbf{r}_k$$

$$\mathbf{e}_{k+1} = \mathbf{e}_k - \alpha\,\mathbf{r}_k = (\mathbf{I} - \alpha\mathbf{A})\mathbf{e}_k.$$

Voorbeeld. $n = 1$, $\mathbf{A} = [\lambda]$.

Dan $\qquad \mathbf{e}_{k+1} = (1 - \alpha\,\lambda)\,\mathbf{e}_k = (1 - \alpha\,\lambda)^{k+1}\mathbf{e}_0 \to 0$

$$\Leftrightarrow \qquad |1 - \alpha\,\lambda| < 1.$$

Iteratief

Richardson iteration

Choose $tol > 0$, $\mathbf{x}_0$, $k_{\max}$, α
Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
For $k = 0, 1, 2, \ldots, k_{\max}$
    Stop if $\|\mathbf{r}_k\|_2 \leq tol\|\mathbf{b}\|_2$
    $\mathbf{u}_k = \mathbf{r}_k$
    $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$
    Determine α
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\,\mathbf{u}_k$
    $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha\,\mathbf{c}_k$
end for

Iteratief

Stelling. Richardson convergeert

als $\quad \alpha\,\lambda \in \{\zeta \in \mathbb{C} \mid |1-\zeta| < 1\} \qquad (*)$

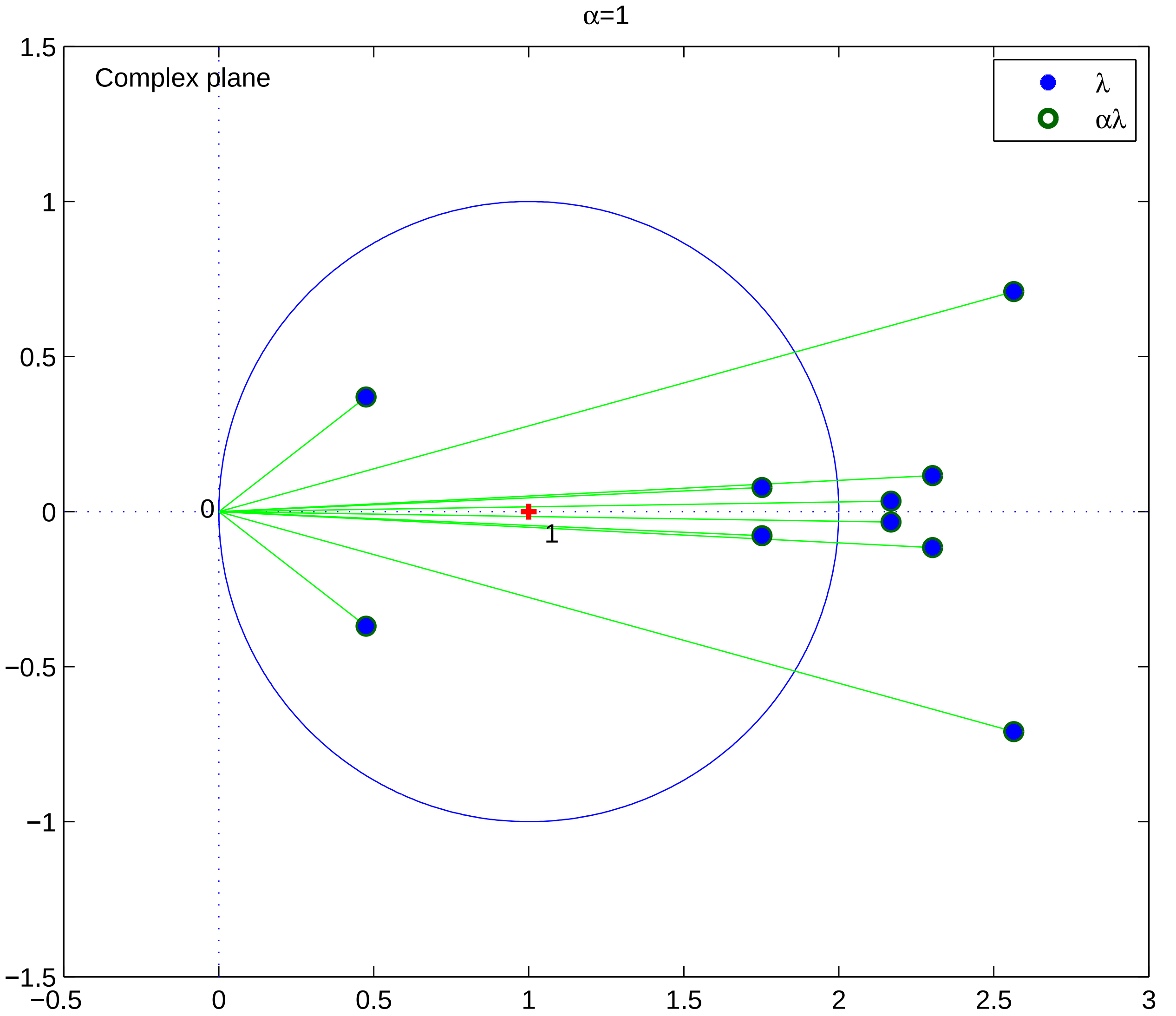
voor alle eigenwaarden λ van **A**

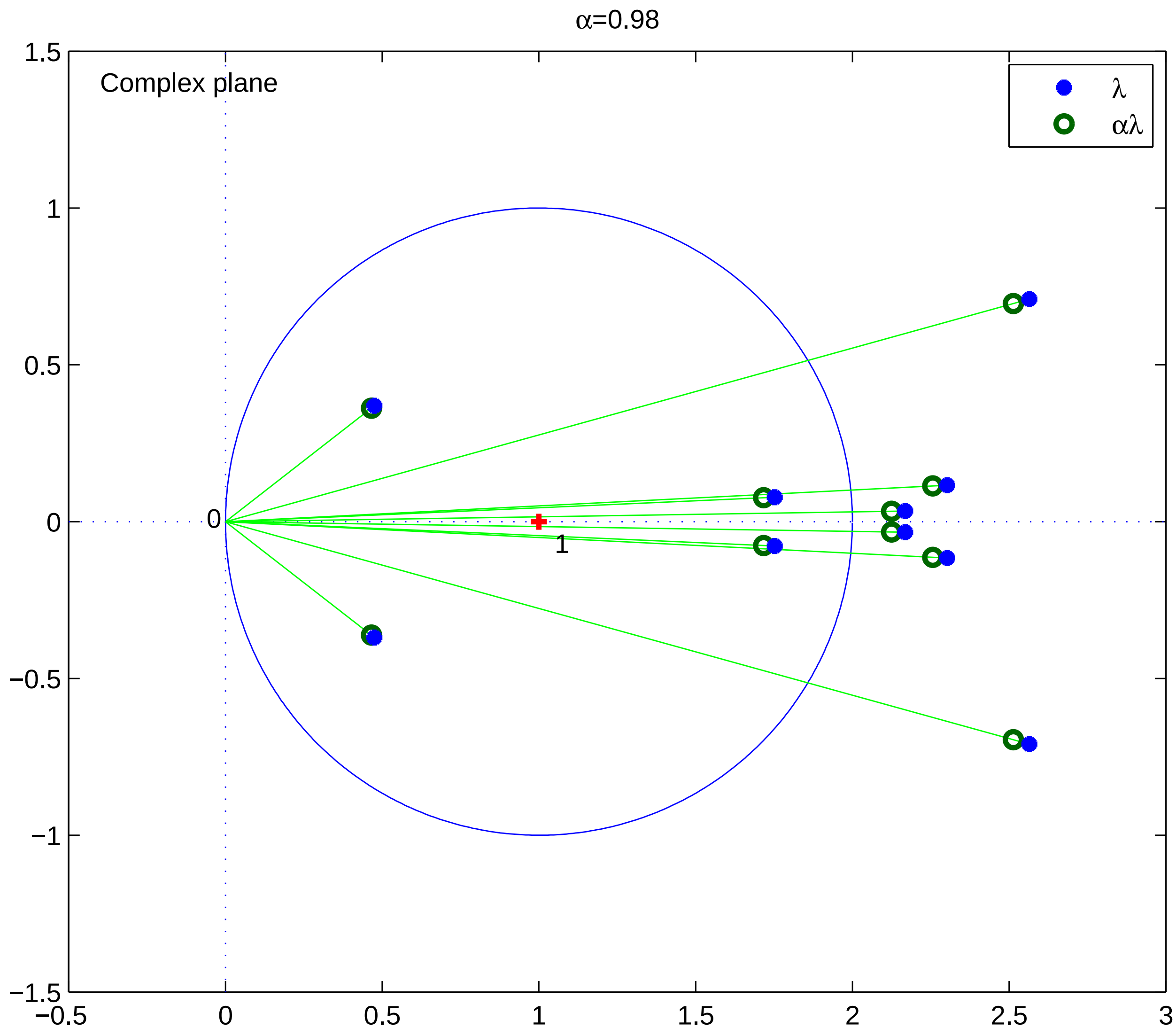
Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden van **A**, dan is er een $\alpha > 0$ zodat $(*)$ [*Bewijs: zie animatie*]

Voorbeeld. $\mathrm{Re}(\lambda) > 0$ voor alle eigw. van **A**:

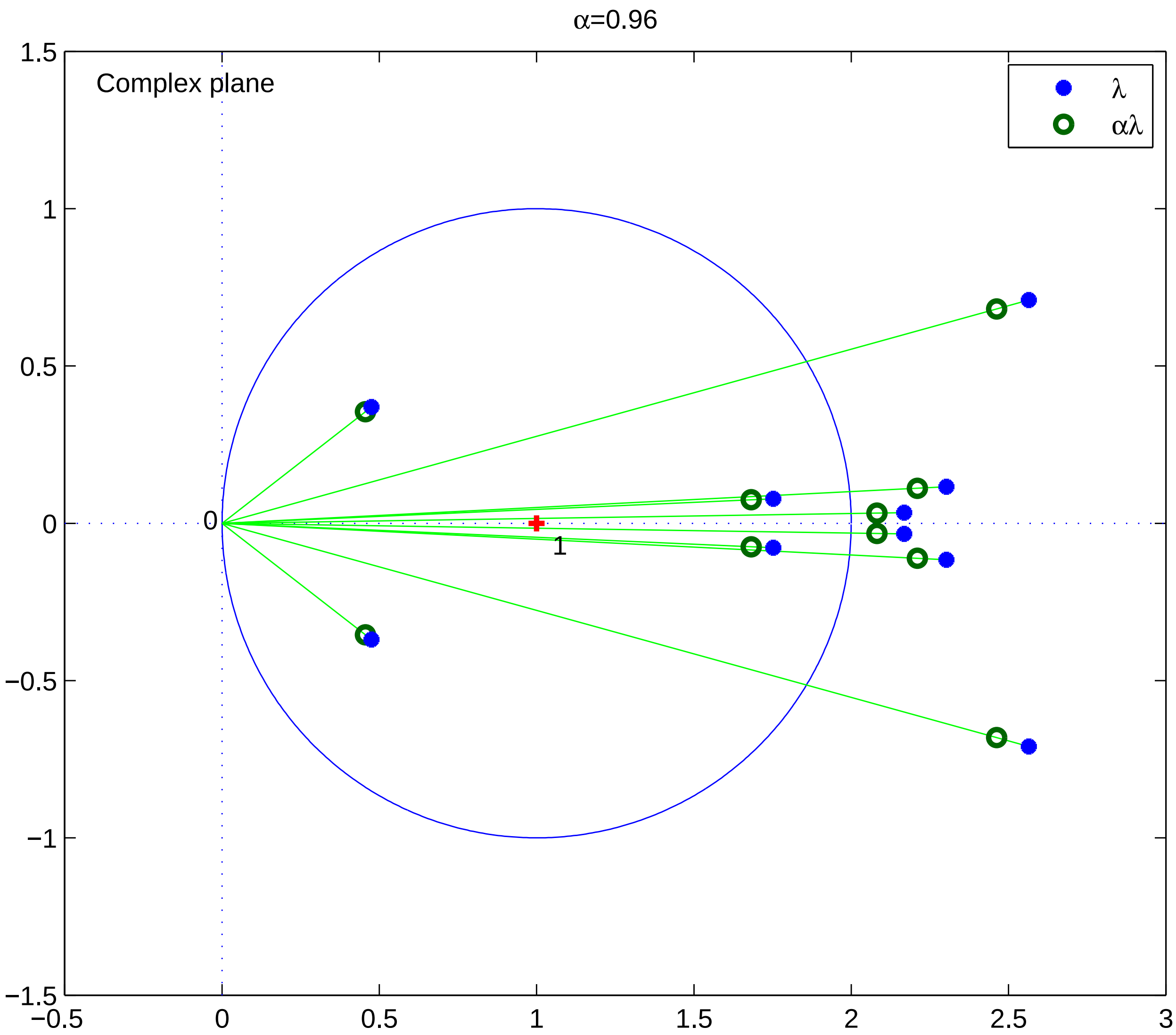
1) Grondwaterstroming (geschikte discretisatie)

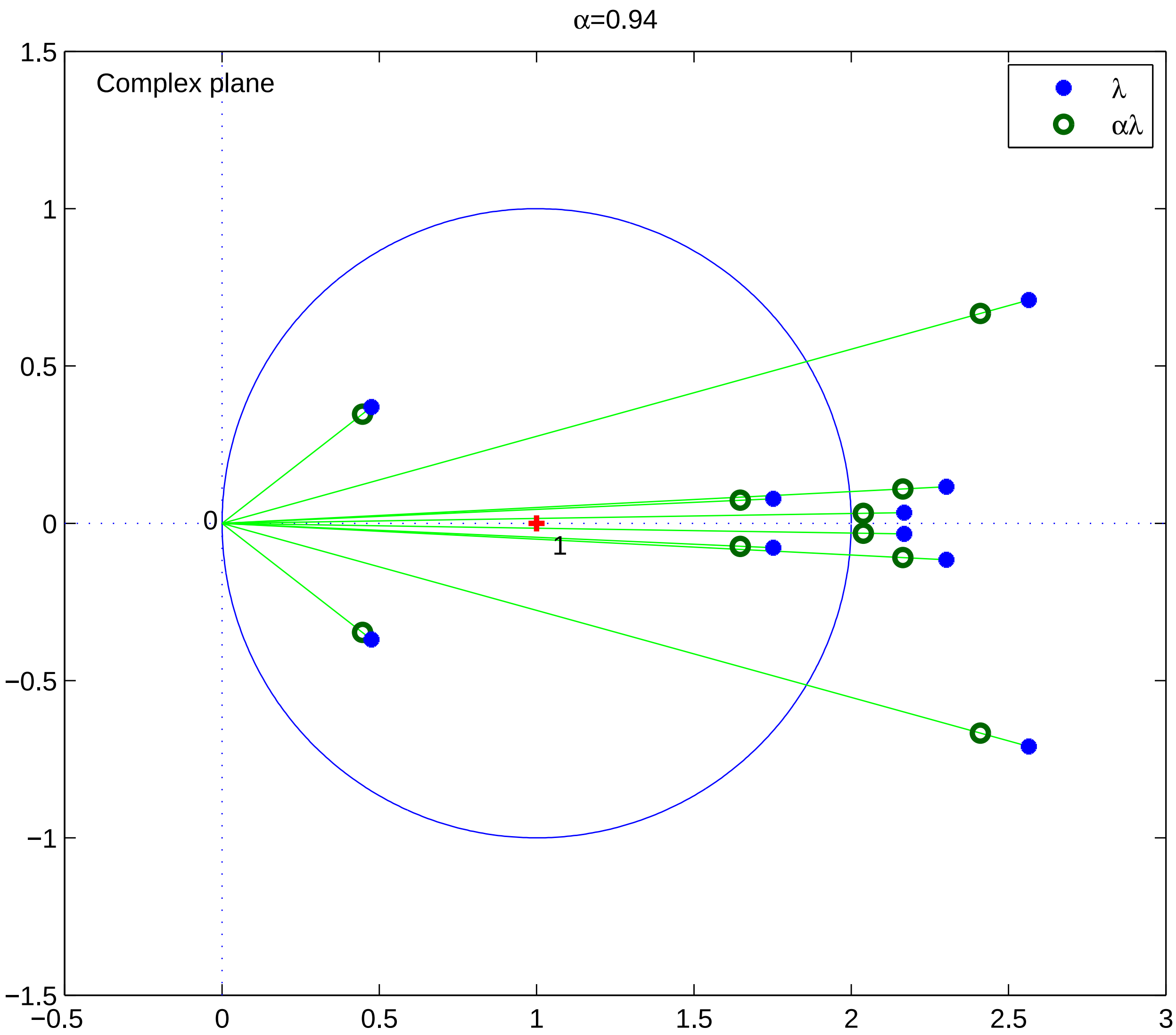
2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein


α=1
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

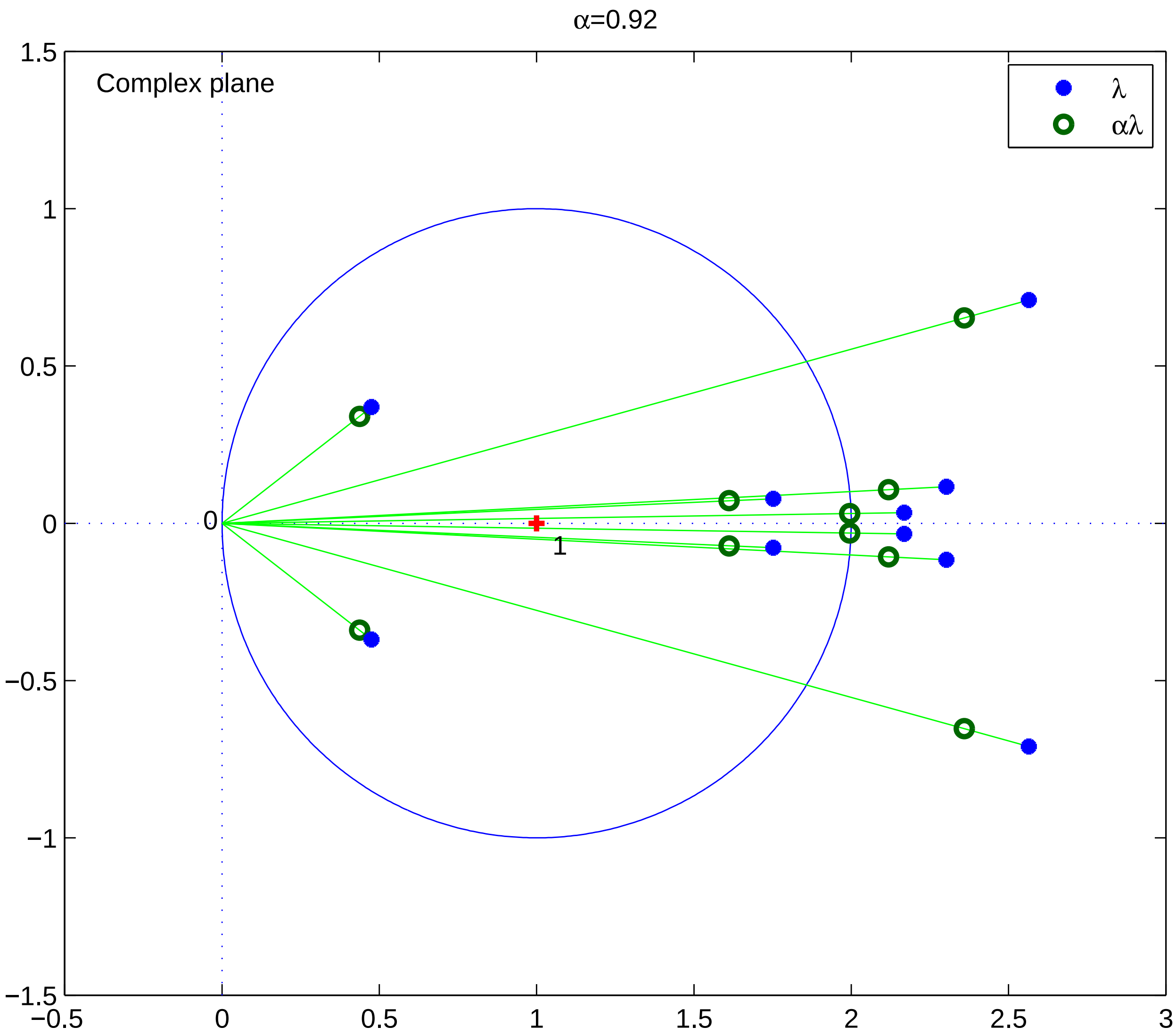


α=0.98
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

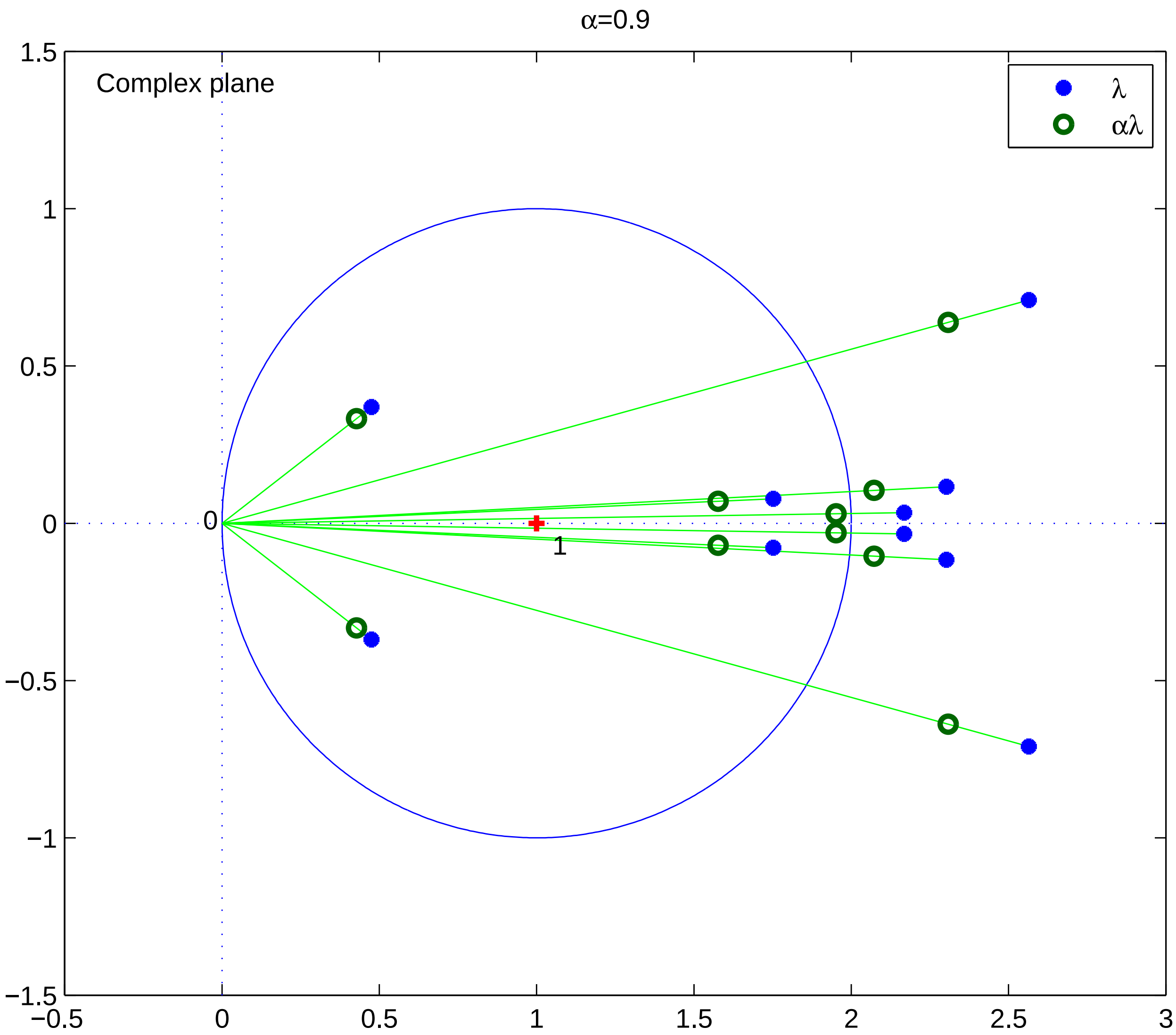

α=0.96
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3



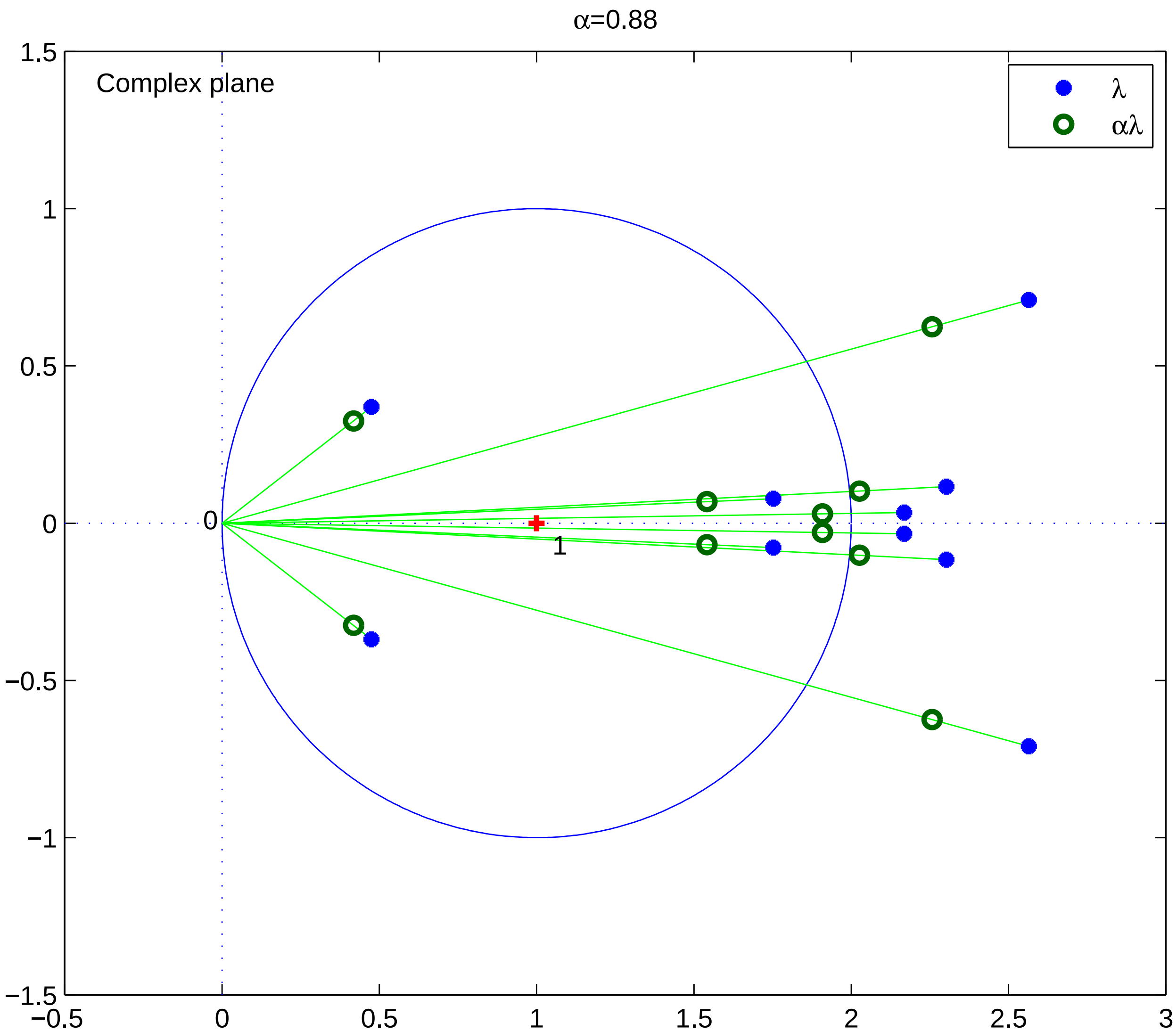
α=0.94
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

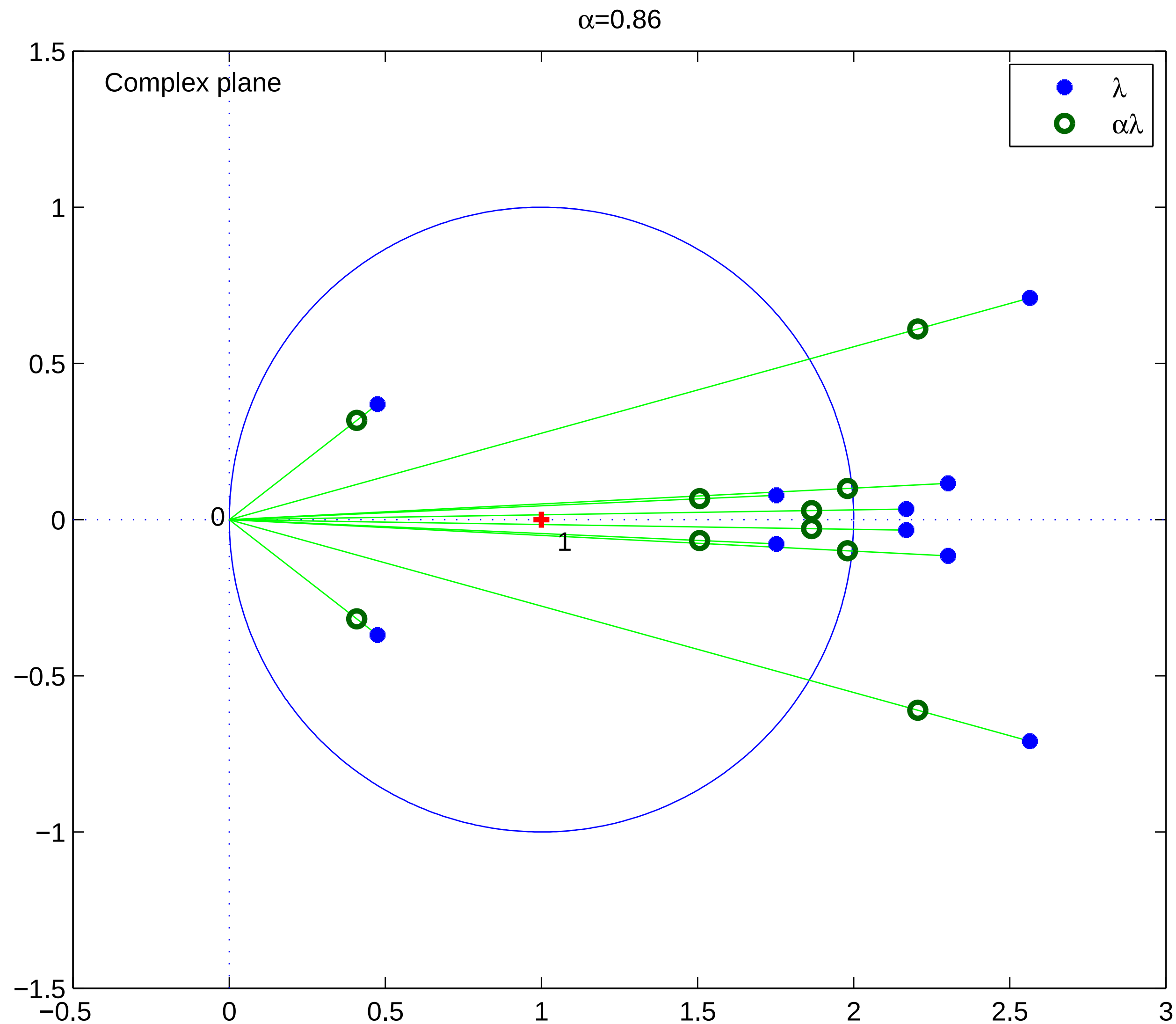


α=0.92
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

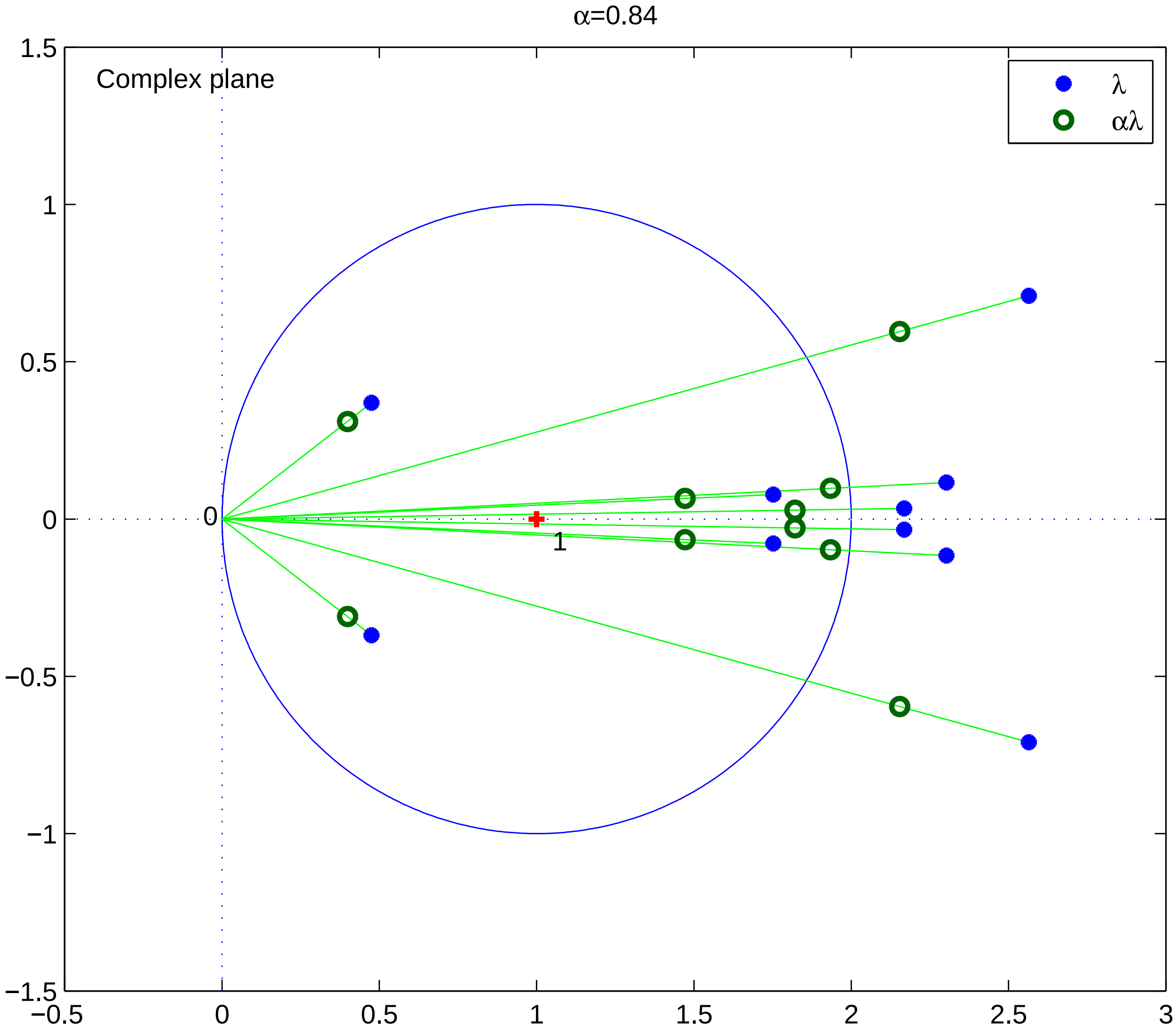


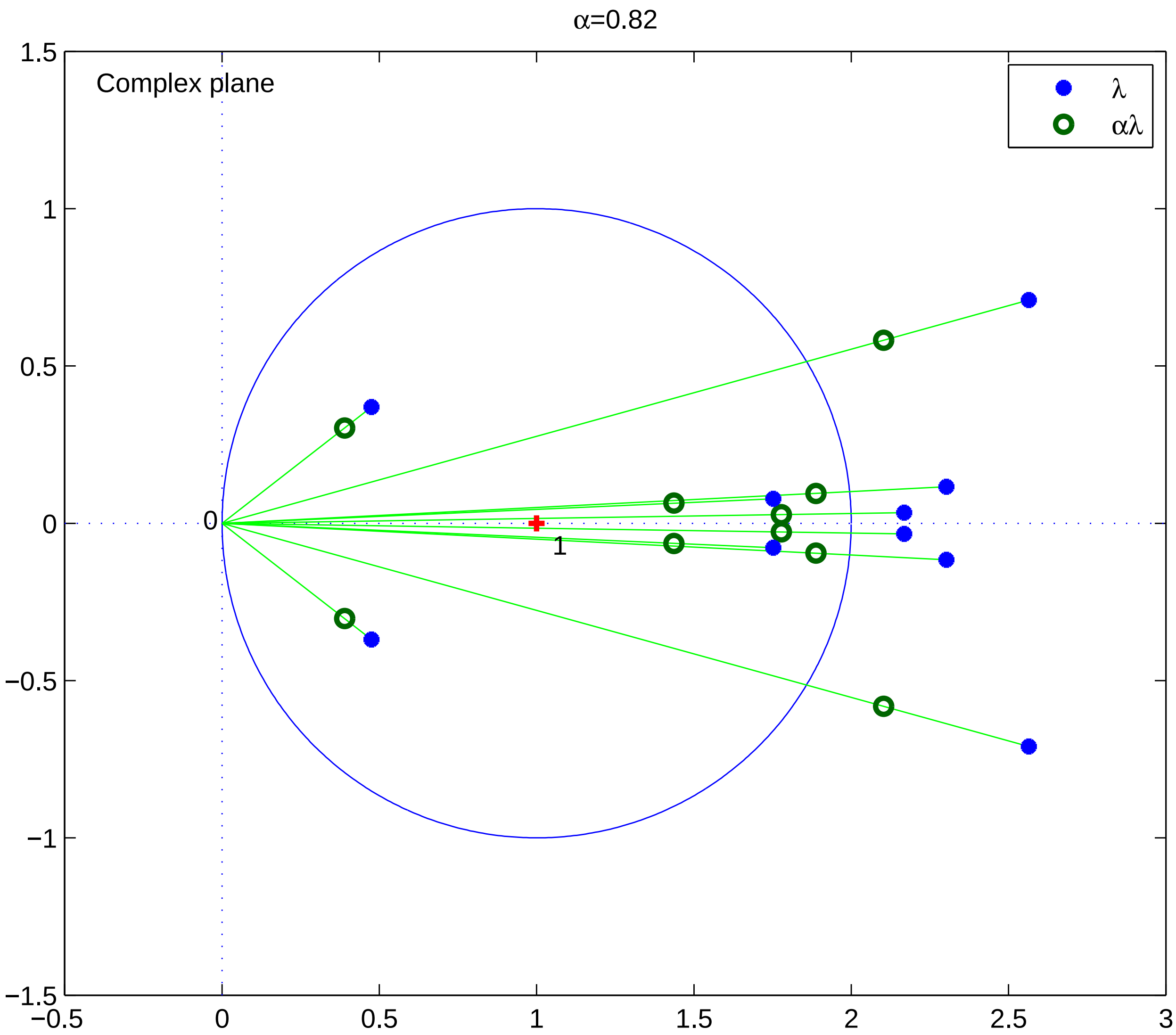
α=0.9
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3


α=0.88
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

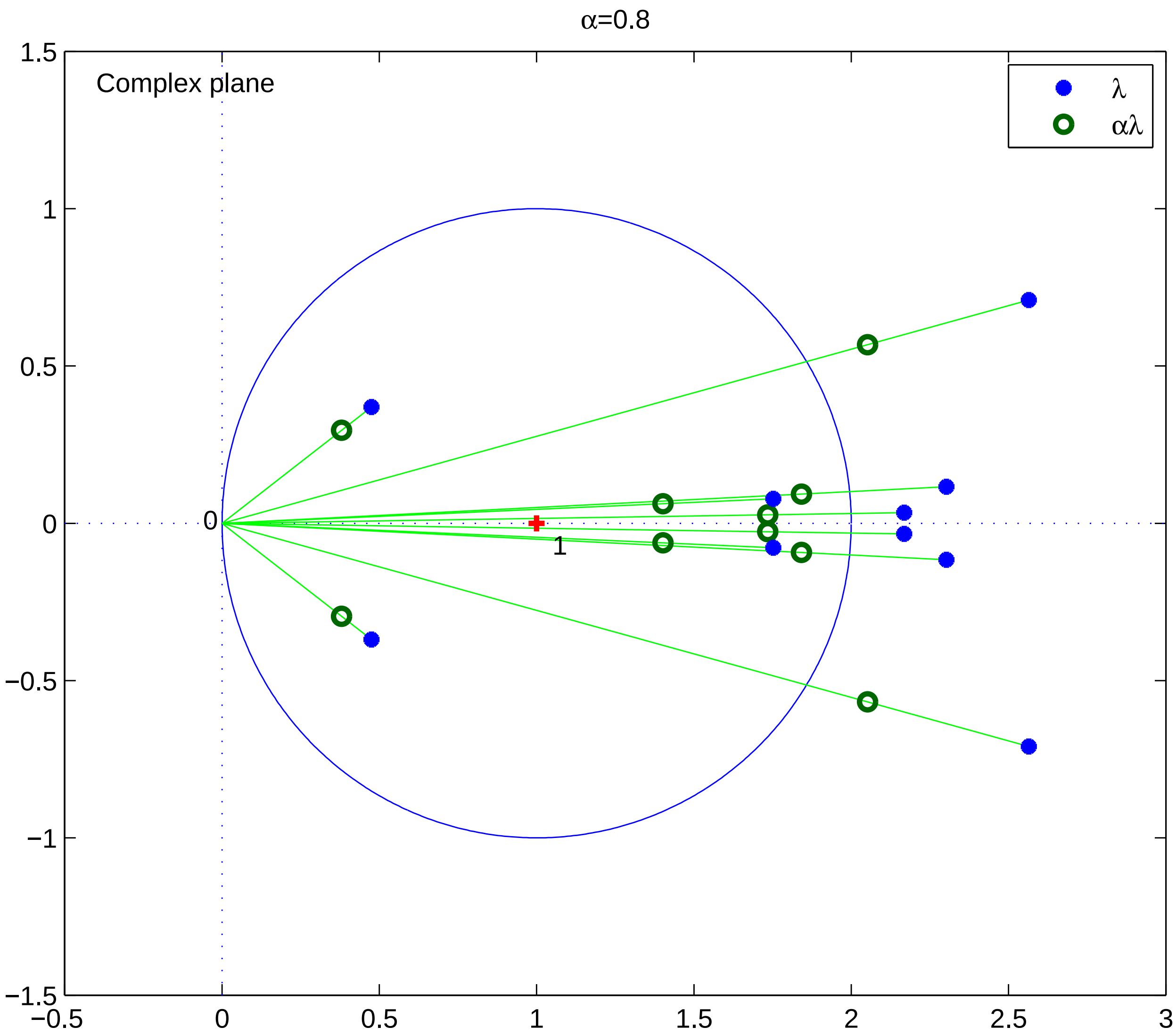


α=0.86
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

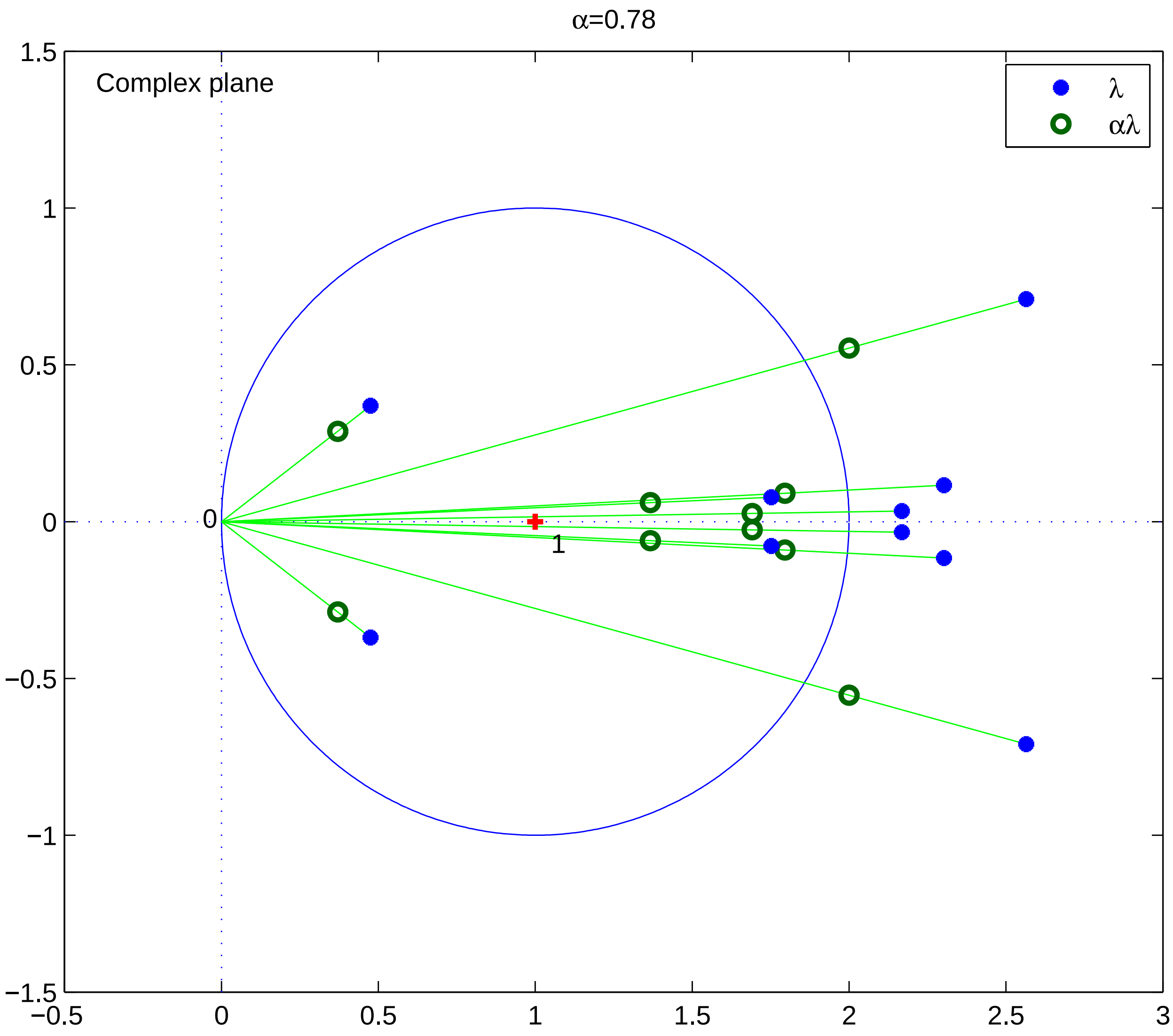

α=0.84
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3



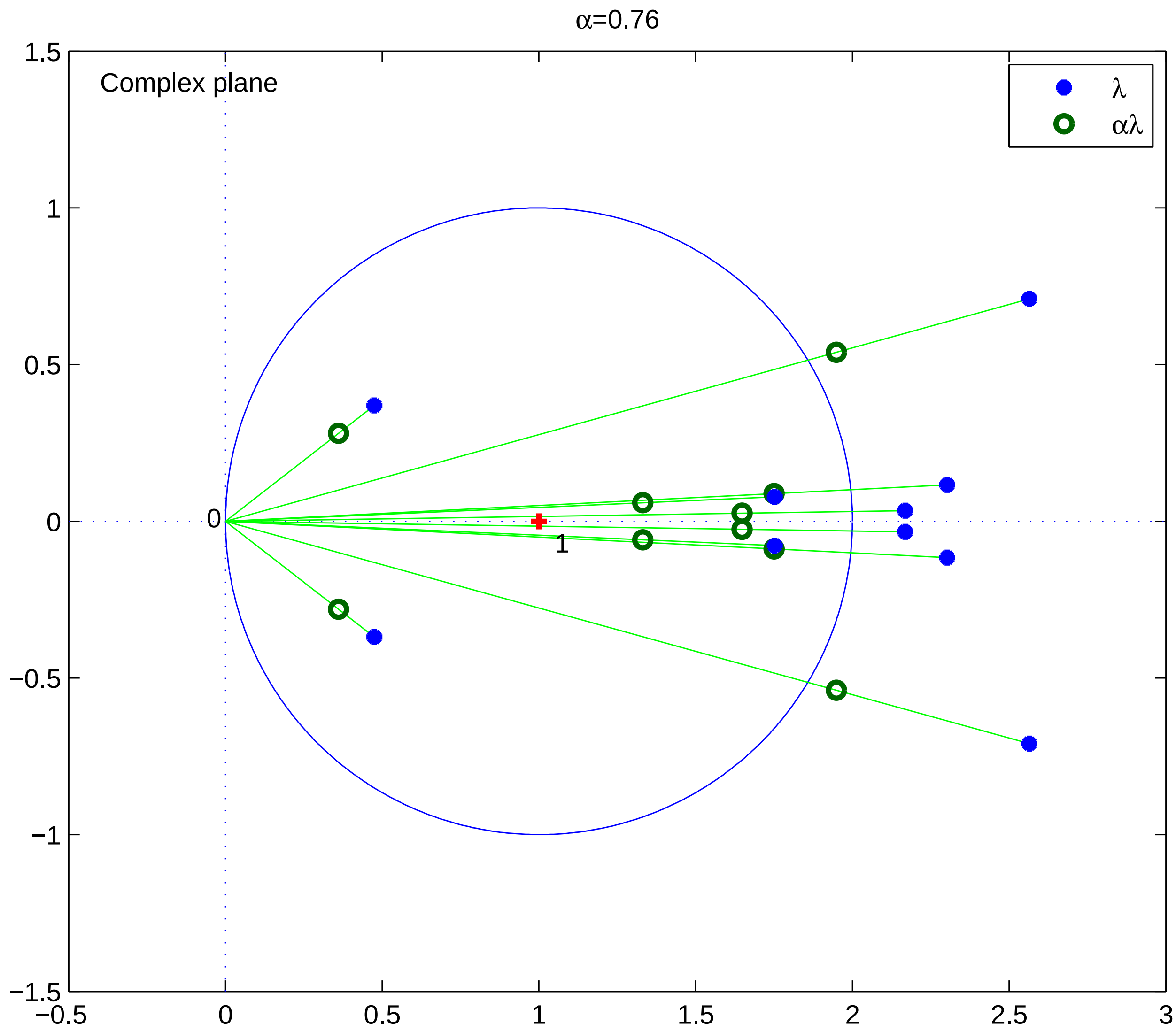
α=0.82
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3



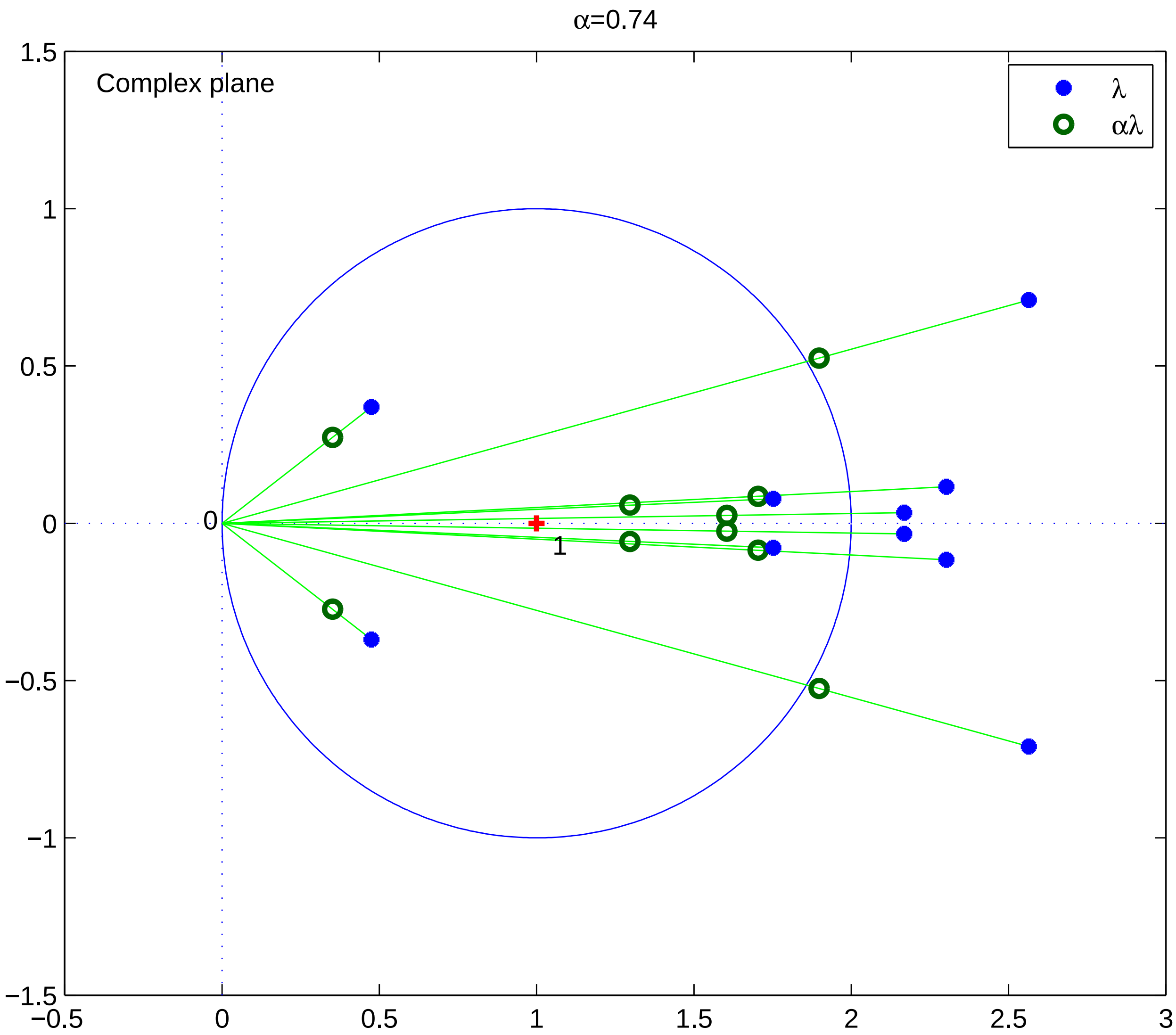
α=0.8
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

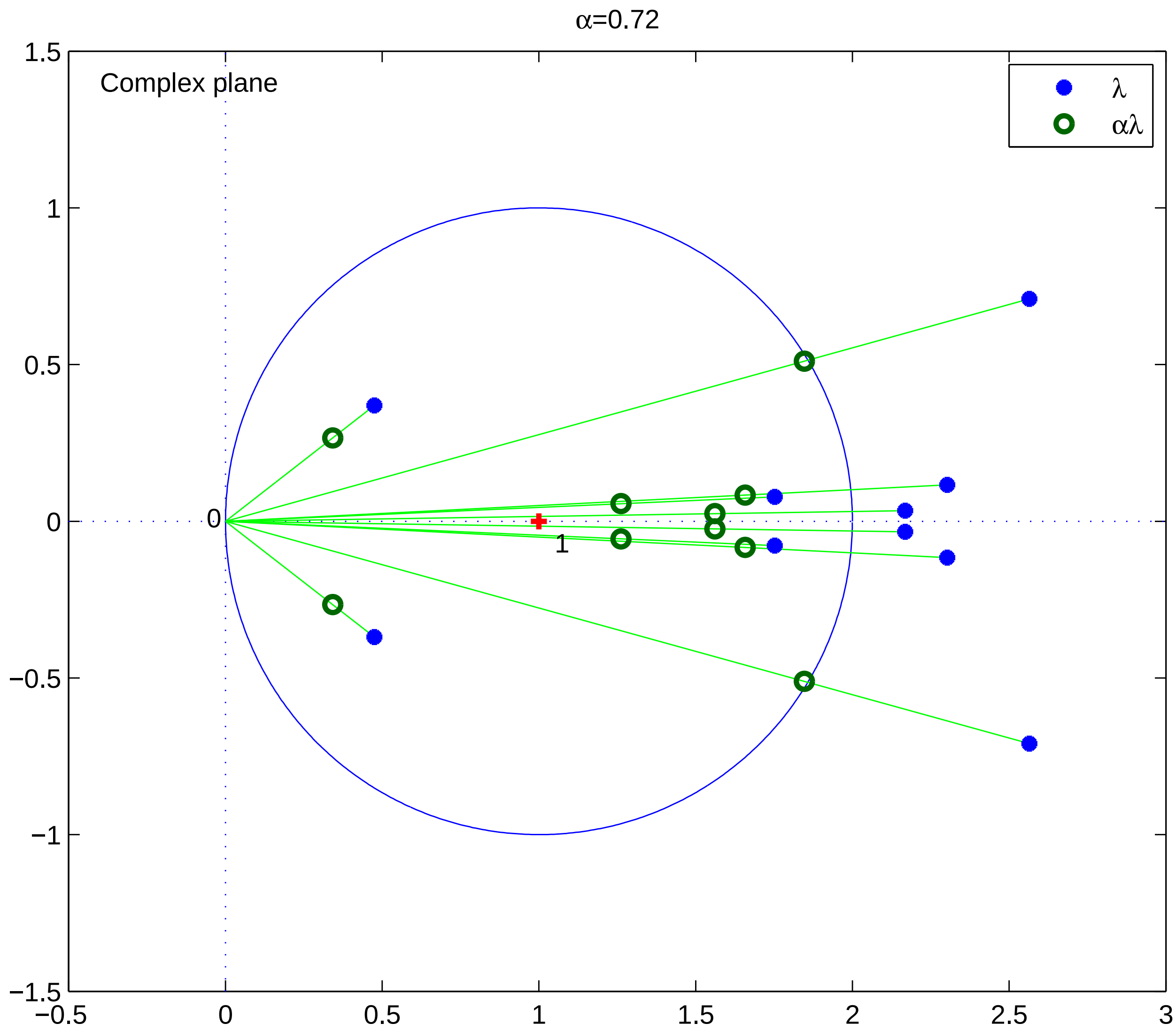


α=0.78
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

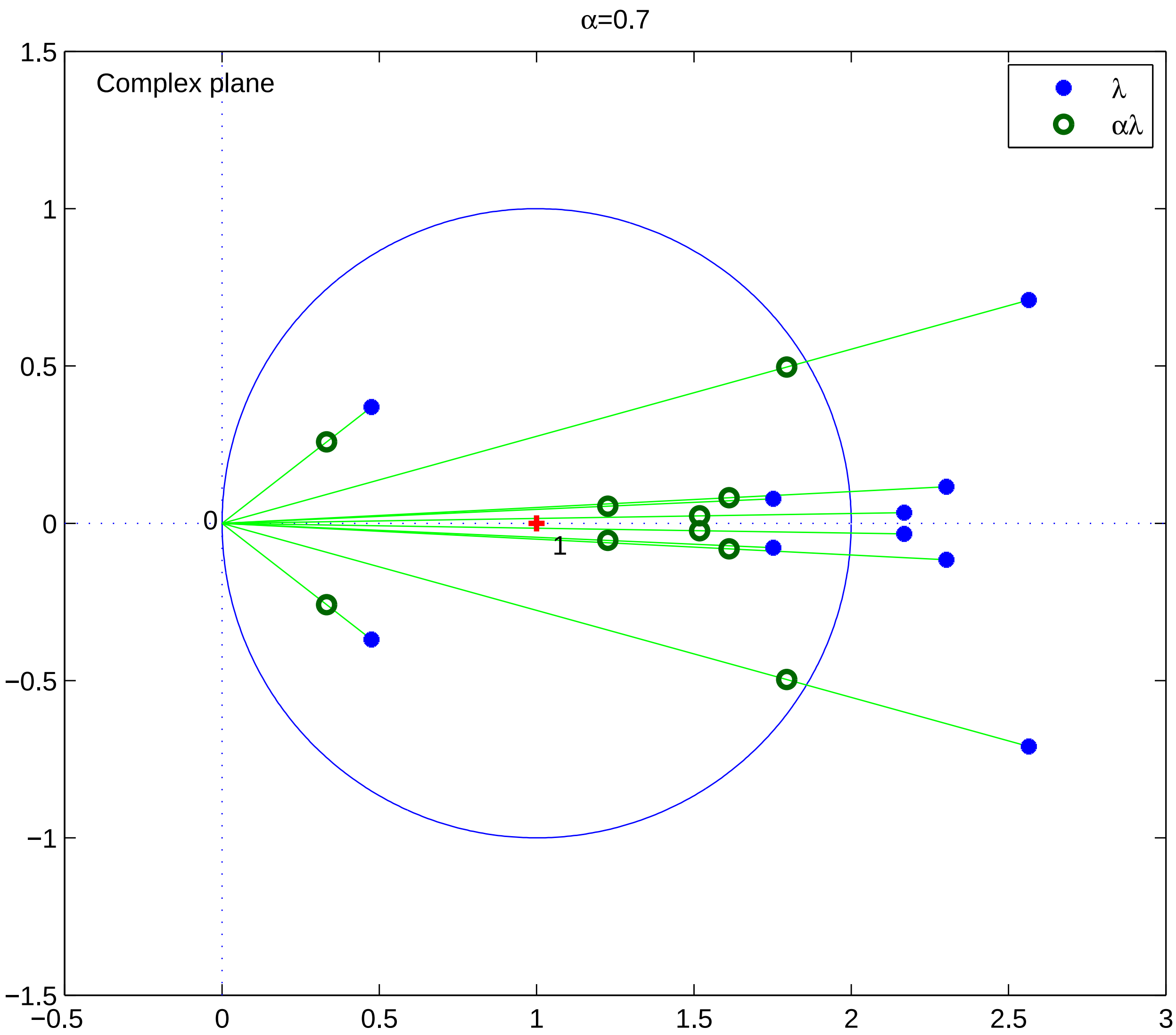


α=0.76
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

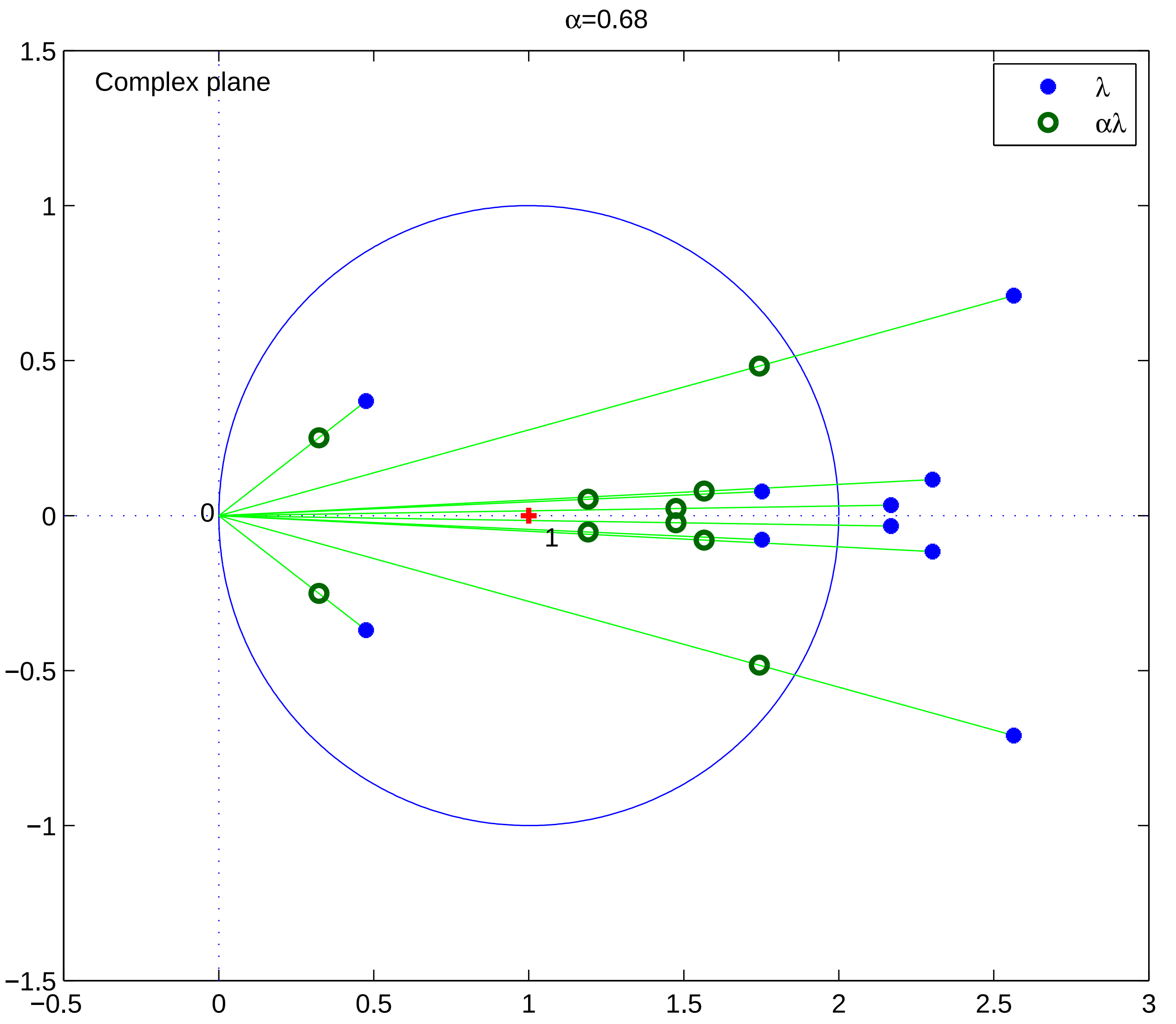

α=0.74
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3



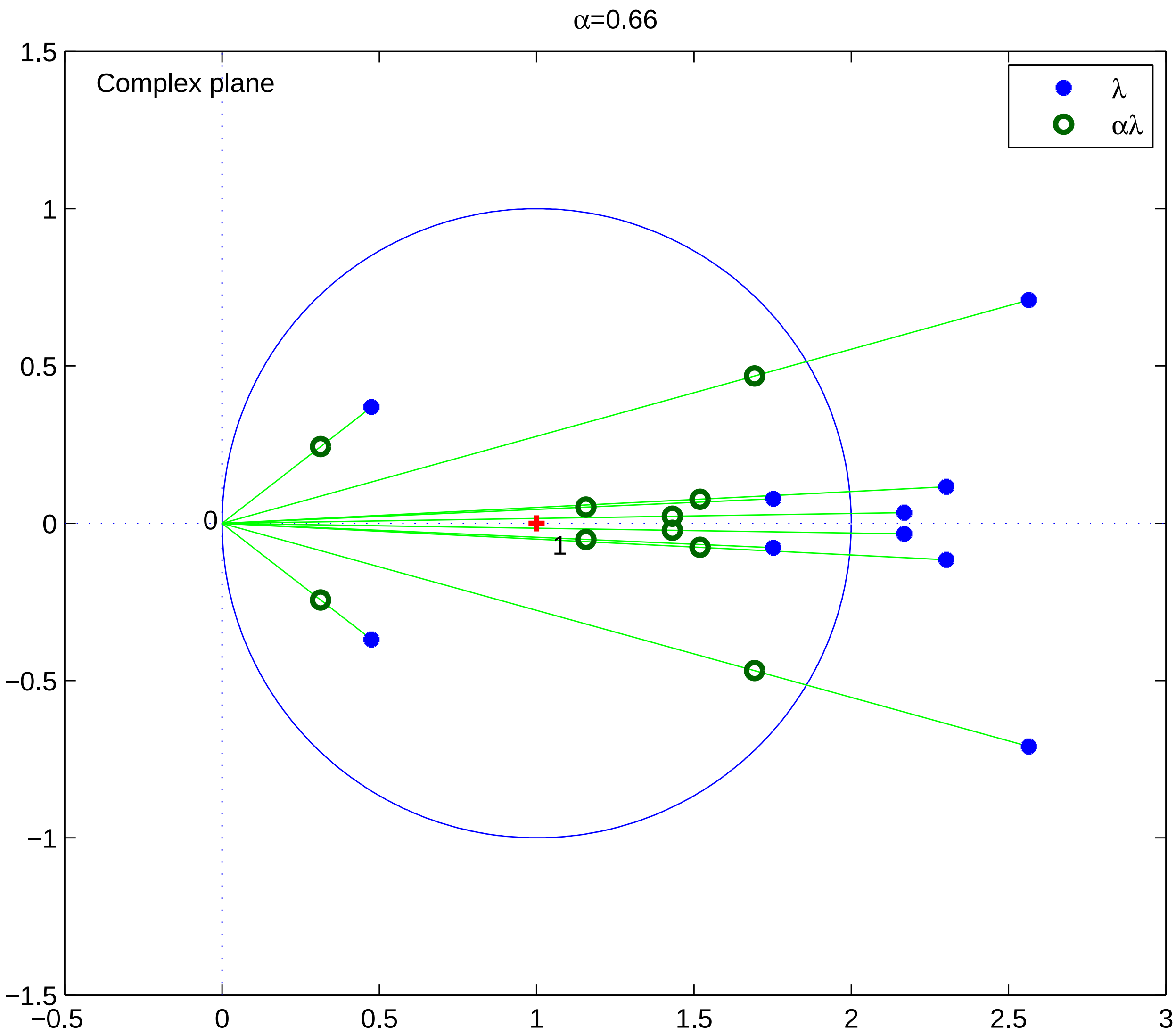
α=0.72
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3



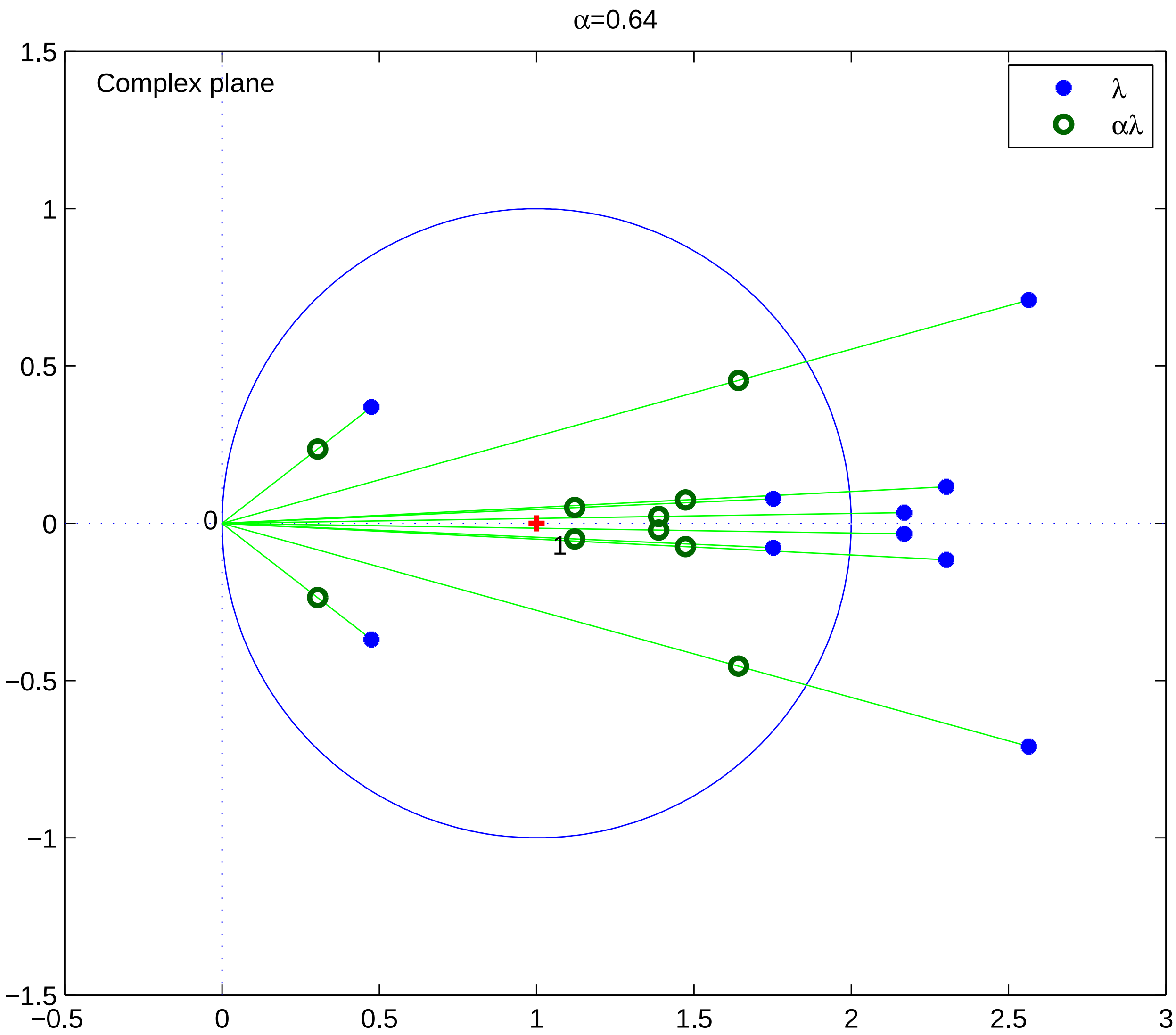
α=0.7
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3



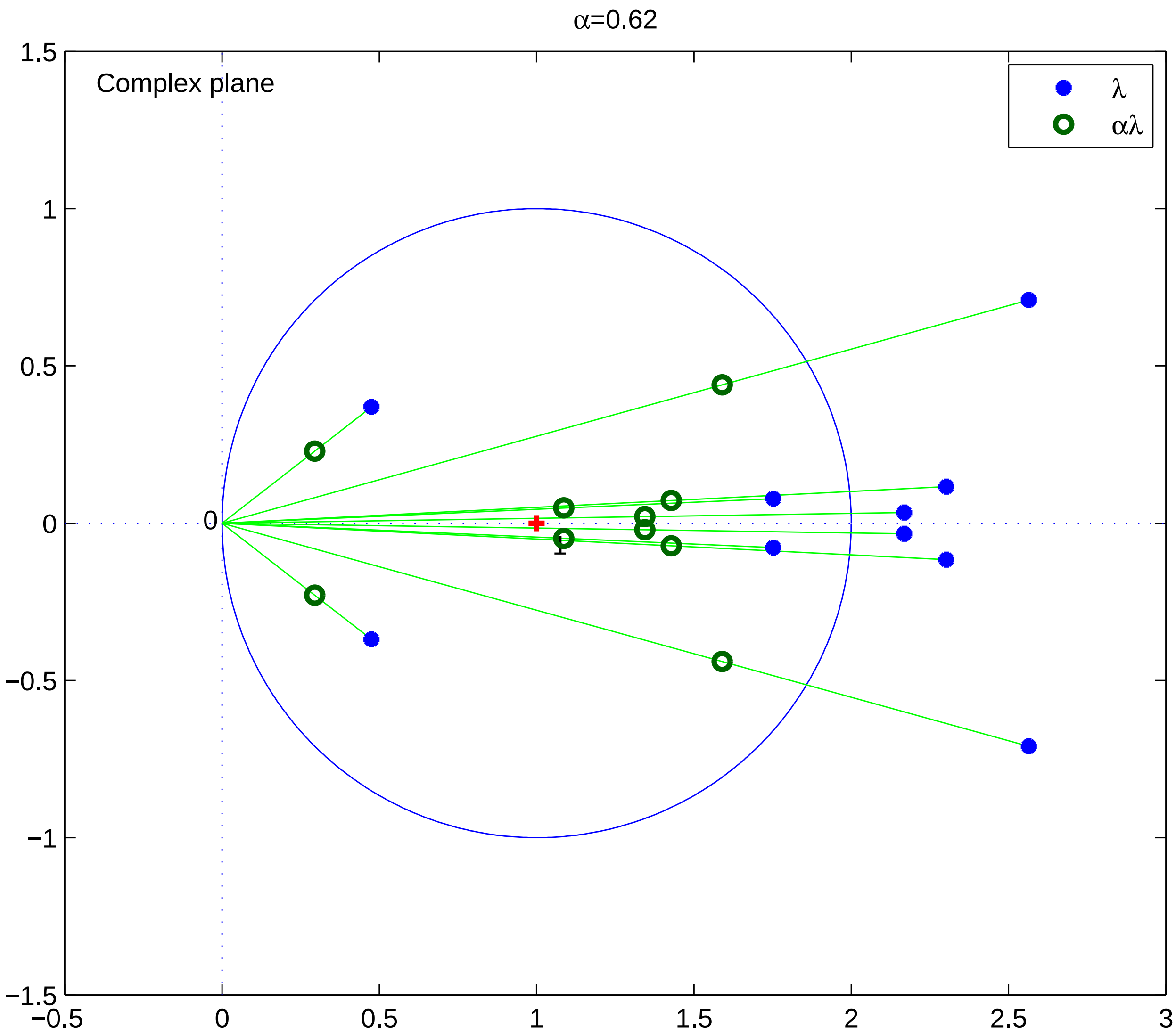
α=0.68
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

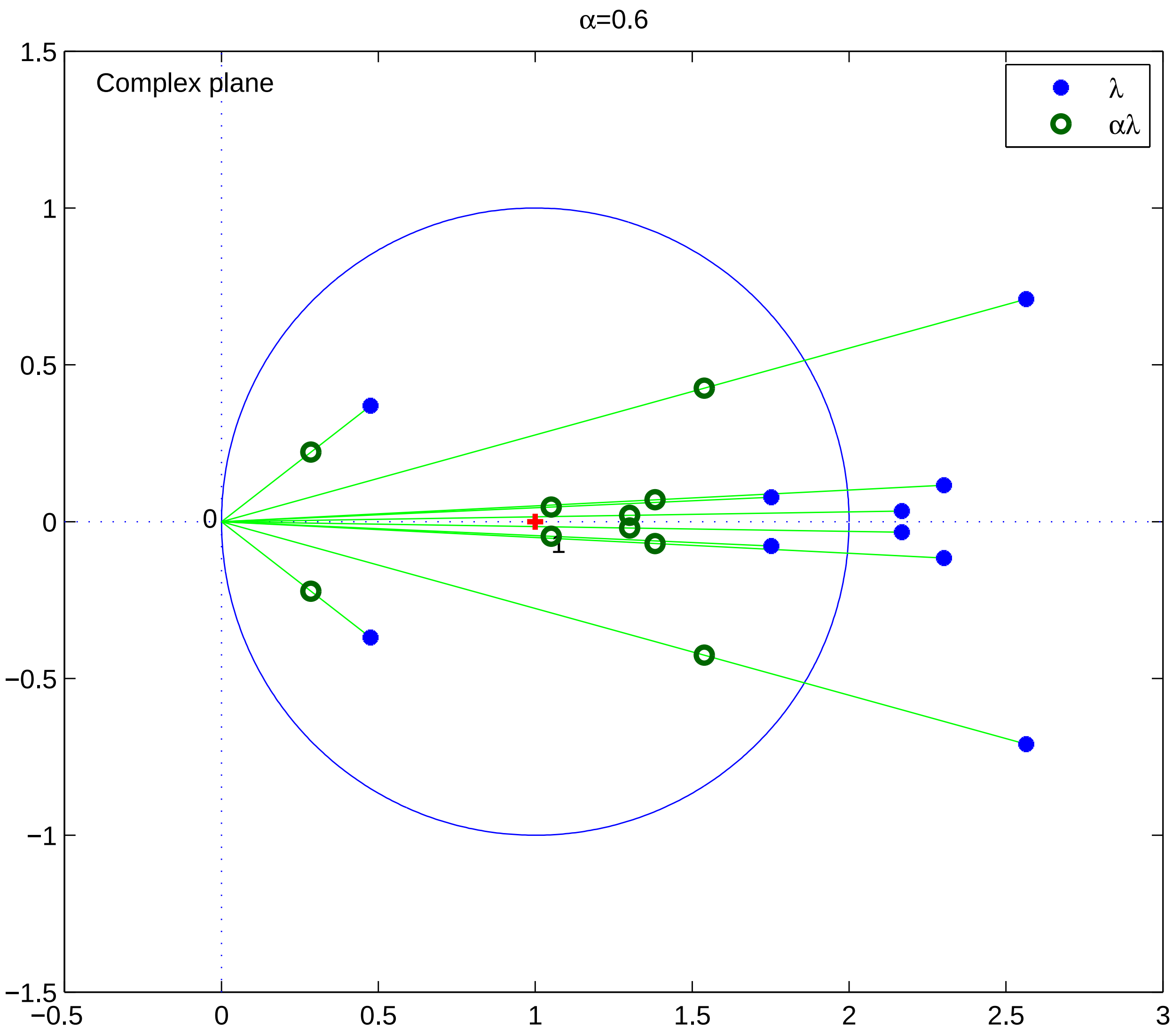


α=0.66
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3



α=0.64
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3


α=0.62
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3



α=0.6
Complex plane
λ
αλ
0
1
1.5
1
0.5
0
−0.5
−1
−1.5
−0.5
0
0.5
1
1.5
2
2.5
3

Iteratief

Stelling. Richardson convergeert

als $\quad \alpha\,\lambda \in \{\zeta \in \mathbb{C} \mid |1-\zeta| < 1\} \qquad (*)$

voor alle eigenwaarden λ van $\mathbf{A}$

Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden van $\mathbf{A}$, dan is er een $\alpha > 0$ zodat $(*)$ [*Bewijs: zie animatie*]

Voorbeeld. $\mathrm{Re}(\lambda) > 0$ voor alle eigw. van $\mathbf{A}$:
1) Grondwaterstroming (geschikte discretisatie)
2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein

Opmerking. Of het proces convergeert hangt niet af van de **initial guess** $\mathbf{x}_0$. Als het proces convergeert dan zal je wel minder stappen nodig hebben als $\mathbf{x}_0$ dichter bij de exacte oplossing $\mathbf{x}$ ligt.

Iteratief

Stelling. Richardson convergeert

$$\text{als} \qquad \alpha\,\lambda \in \{\zeta \in \mathbb{C} \mid |1-\zeta| < 1\} \qquad (*)$$

voor alle eigenwaarden λ van $\mathbf{A}$

Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden van $\mathbf{A}$, dan is er een $\alpha > 0$ zodat $(*)$ [*Bewijs: zie animatie*]

Voorbeeld. $\mathrm{Re}(\lambda) > 0$ voor alle eigw. van $\mathbf{A}$:

1) Grondwaterstroming (geschikte discretisatie)
2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein

Initial guess. $\mathbf{x}_0 = \mathbf{0}$ is een populaire keuze.

Als het je grondwater probleem opgelost hebt zonder pomp, dan zou je die oplossing kunnen gebruiken als $\mathbf{x}_0$ *voor het probleem met pomp.*

Iteratief

Stelling. Richardson convergeert

als $\quad \alpha\,\lambda \in \{\zeta \in \mathbb{C} \mid |1-\zeta| < 1\} \qquad (*)$

voor alle eigenwaarden λ van $\mathbf{A}$

Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden van $\mathbf{A}$, dan is er een $\alpha > 0$ zodat $(*)$ [*Bewijs: zie animatie*]

Voorbeeld. $\mathrm{Re}(\lambda) > 0$ voor alle eigw. van $\mathbf{A}$:
1) Grondwaterstroming (geschikte discretisatie)
2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein

Probleem. Hoe vinden we een geschikte α?

Iteratief

Stelling. Richardson convergeert

als $\quad \alpha\,\lambda \in \{\zeta \in \mathbb{C} \mid |1-\zeta| < 1\} \qquad (*)$

voor alle eigenwaarden λ van $\mathbf{A}$

Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden van $\mathbf{A}$, dan is er een $\alpha > 0$ zodat $(*)$ [*Bewijs: zie animatie*]

Voorbeeld. $\mathrm{Re}(\lambda) > 0$ voor alle eigw. van $\mathbf{A}$:
1) Grondwaterstroming (geschikte discretisatie)
2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein

Idee. Kies α optimaal per stap, afhankelijk van $\mathbf{x}_k$.

Iteratief

Richardson iteration

```
Choose tol > 0, x_0, k_max, α
Compute r_0 = b − A x_0
For k = 0, 1, 2, ..., k_max
    Stop if ||r_k||_2 ≤ tol ||b||_2
    u_k = r_k
    c_k = A u_k
    Determine α
    x_{k+1} = x_k + α u_k
    r_{k+1} = r_k − α c_k
end for
```

Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Iteratief

Stelling. Richardson convergeert

als $\quad \alpha\,\lambda \in \{\zeta \in \mathbb{C} \mid |1 - \zeta| < 1\} \qquad (*)$

voor alle eigenwaarden λ van $\mathbf{A}$

Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden van $\mathbf{A}$, dan is er een $\alpha > 0$ zodat $(*)$ [*Bewijs: zie animatie*]

Voorbeeld. $\mathrm{Re}(\lambda) > 0$ voor alle eigw. van $\mathbf{A}$:
1) Grondwaterstroming (geschikte discretisatie)
2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein

Idee. Kies α zo dat $\|\mathbf{x} - \mathbf{x}_{k+1}\|_2$ minimaal. Kan niet!

Iteratief

Stelling. Richardson convergeert

als $\quad \alpha\,\lambda \in \{\zeta \in \mathbb{C} \mid |1 - \zeta| < 1\} \qquad (*)$

voor alle eigenwaarden λ van $\mathbf{A}$

Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden van $\mathbf{A}$, dan is er een $\alpha > 0$ zodat $(*)$ [*Bewijs: zie animatie*]

Voorbeeld. $\mathrm{Re}(\lambda) > 0$ voor alle eigw. van $\mathbf{A}$:
1) Grondwaterstroming (geschikte discretisatie)
2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein

Idee. Kies α zo dat $\|\mathbf{r}_k - \alpha\,\mathbf{c}_k\|_2$ minimaal.

Stelling

$$\alpha = \mathrm{argmin}_{\widetilde{\alpha}} \|\mathbf{r}_k - \widetilde{\alpha}\mathbf{c}_k\|_2 \quad \Leftrightarrow$$

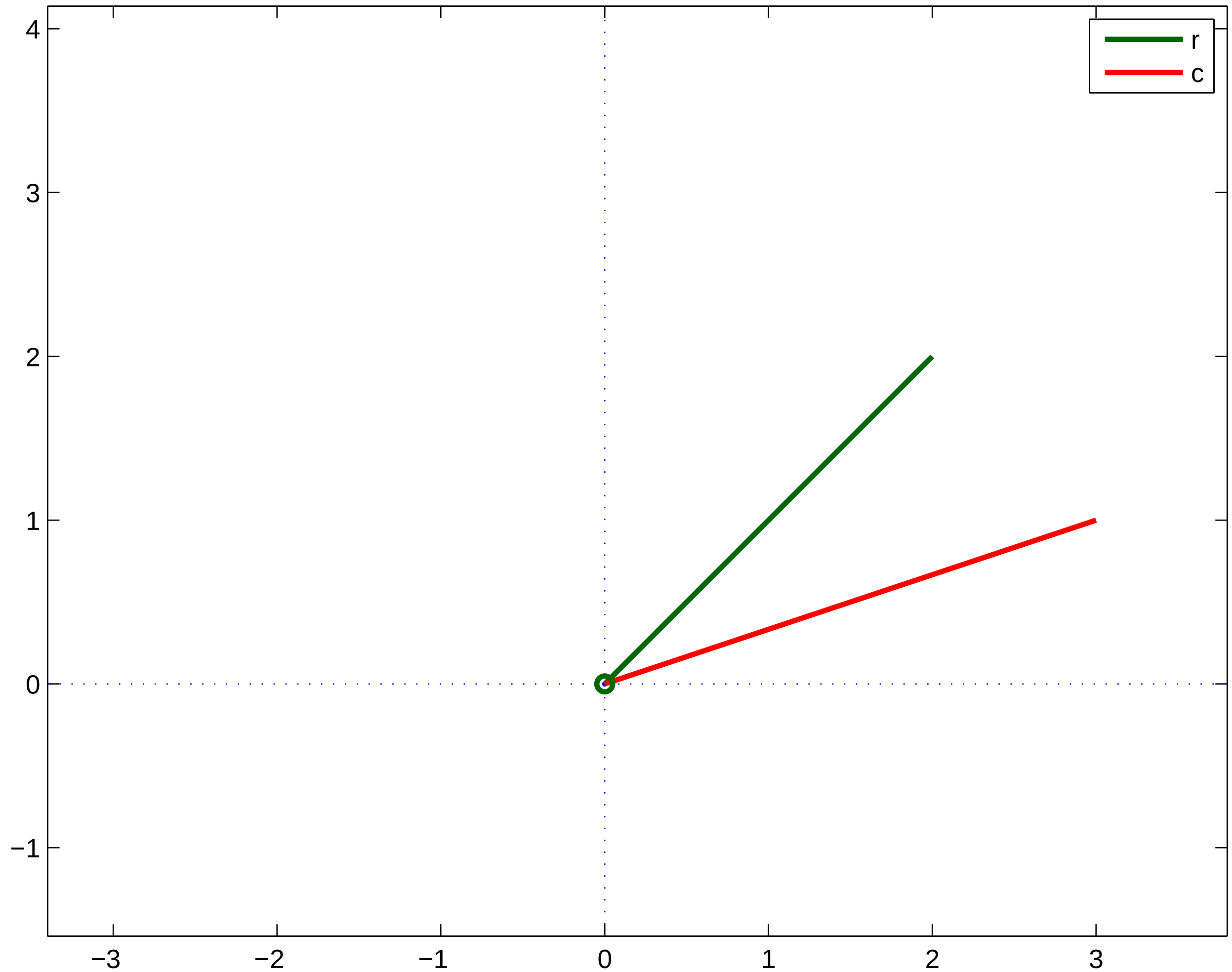
$$\mathbf{r}_k - \alpha\, \mathbf{c}_k \perp \mathbf{c}_k \quad \Leftrightarrow$$

$$\mathbf{c}_k^* \mathbf{r}_k - \alpha\, \mathbf{c}_k^* \mathbf{c}_k = 0 \quad \Leftrightarrow$$

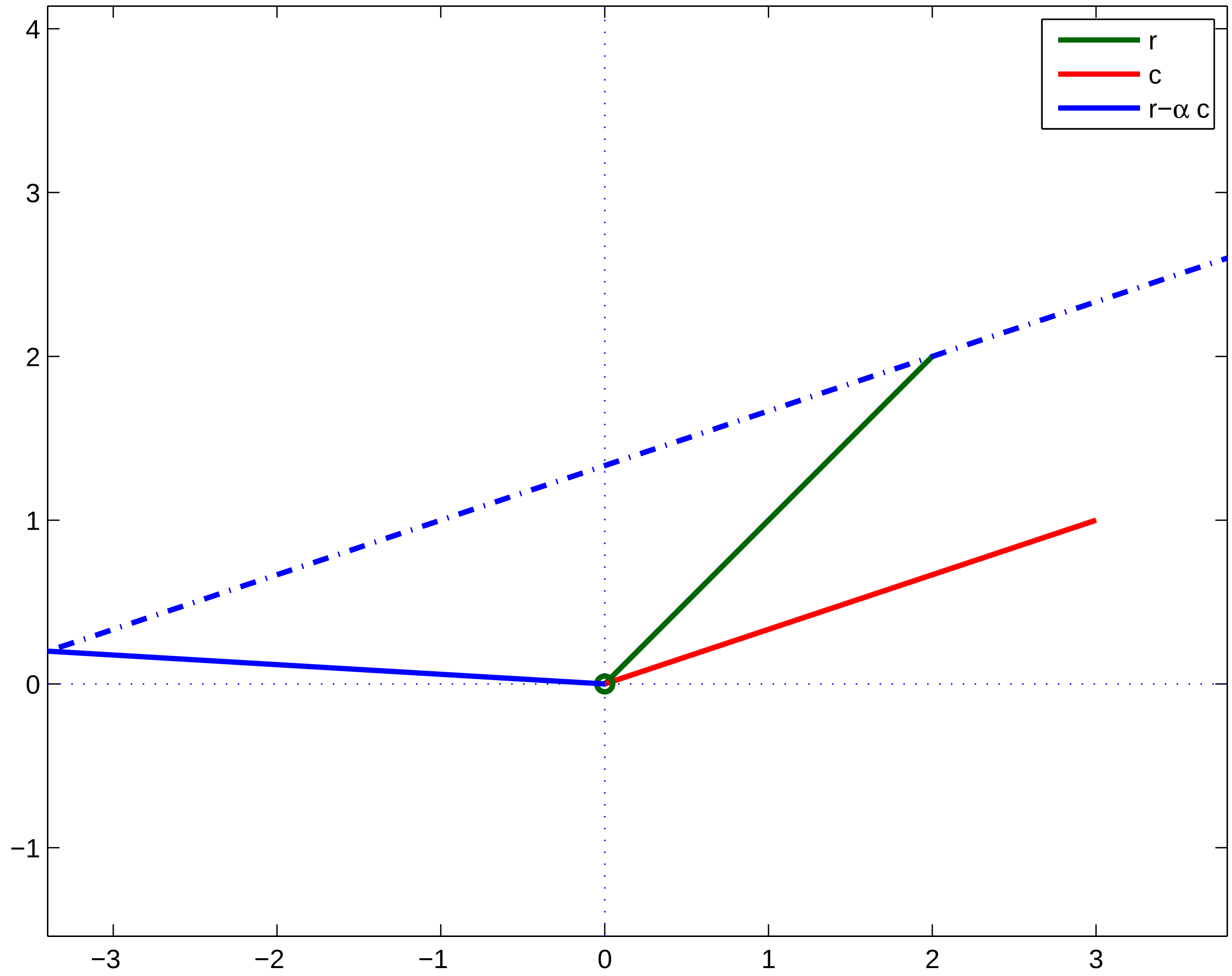
$$\mathbf{c}_k^* \mathbf{r}_k - \alpha\, \mathbf{c}_k^* \mathbf{c}_k = 0 \quad \Leftrightarrow$$

$$\alpha = \frac{\mathbf{c}_k^* \mathbf{r}_k}{\mathbf{c}_k^* \mathbf{c}_k}$$

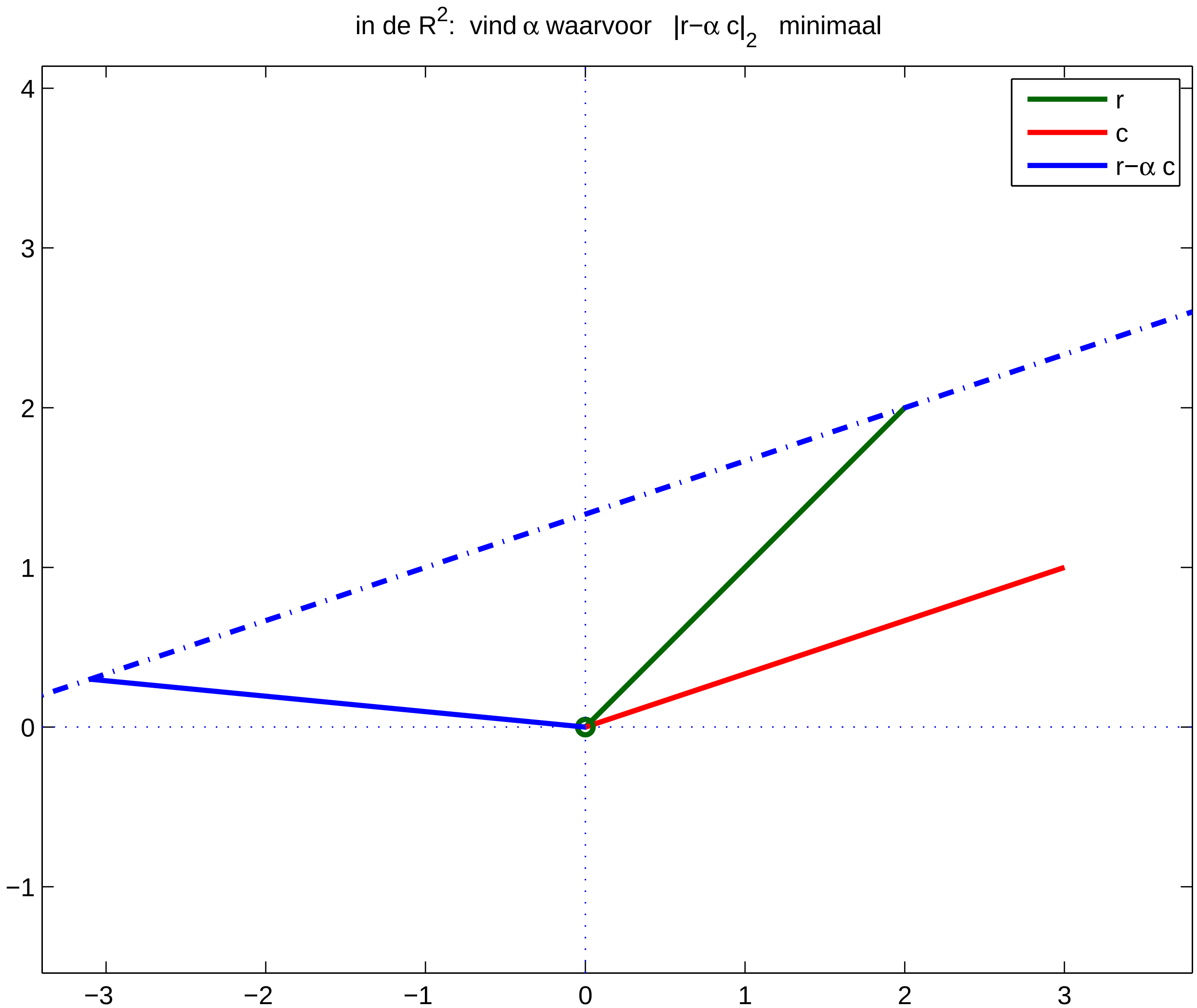
[zie ook plaatjes]

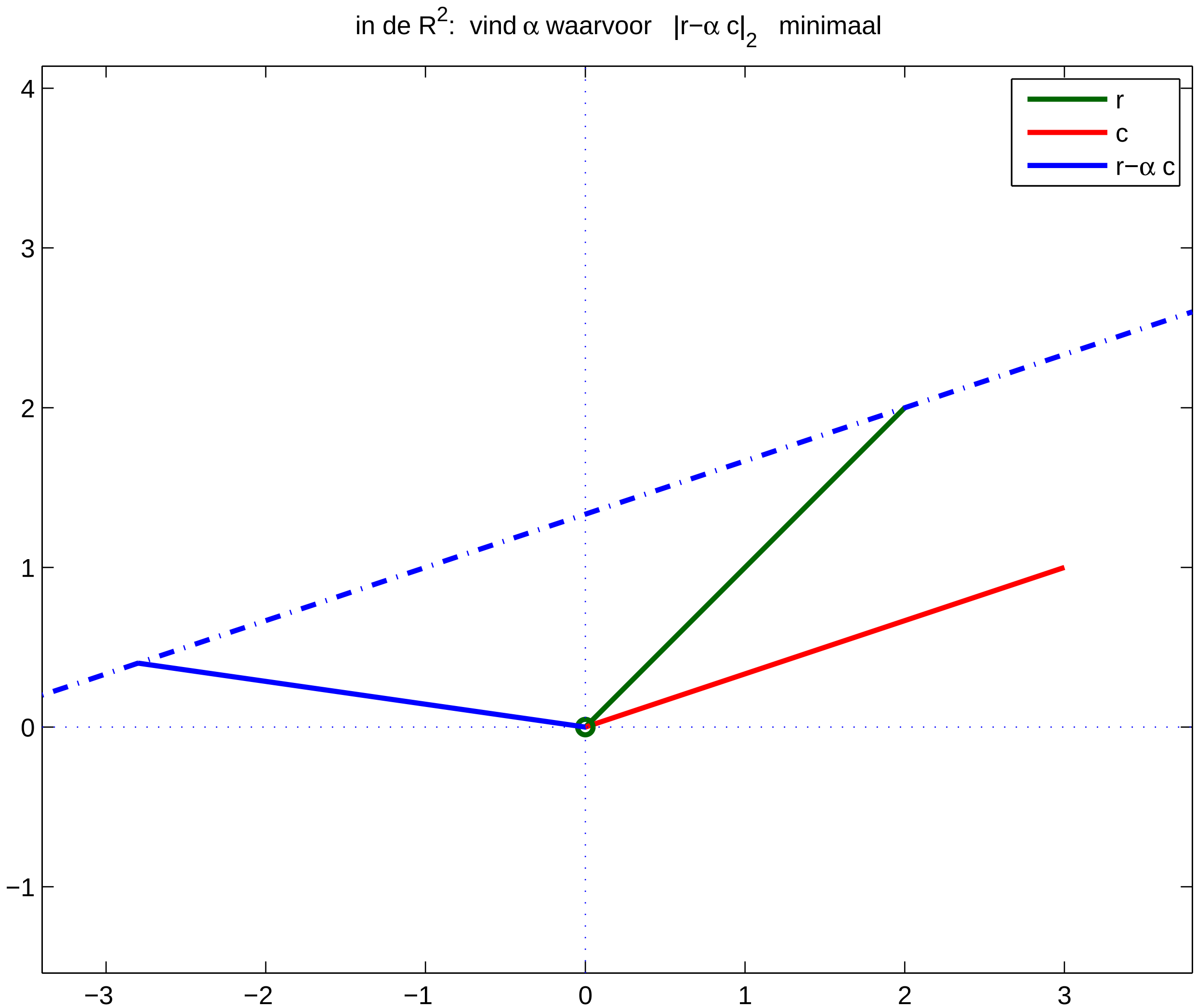


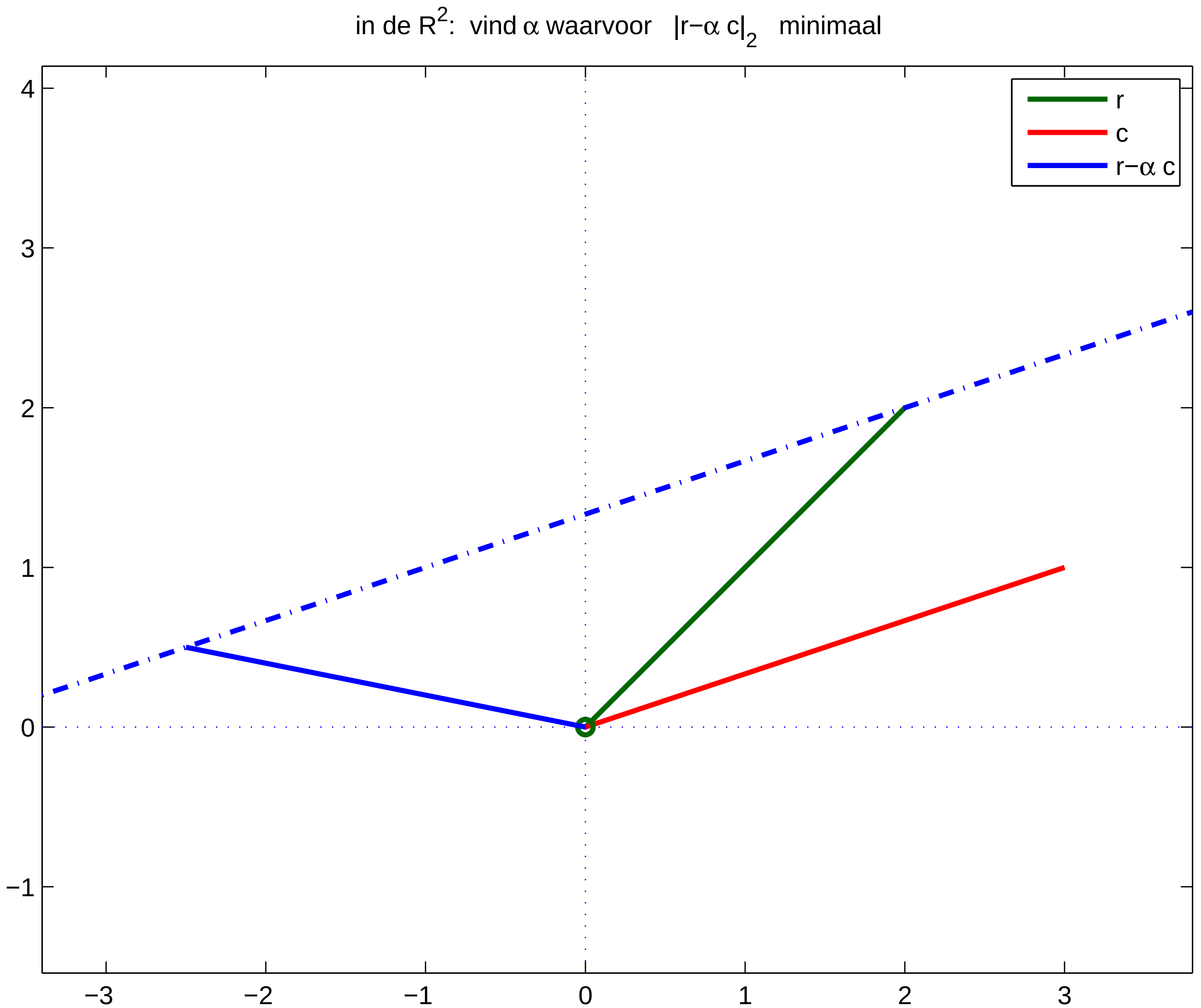
r
c
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3

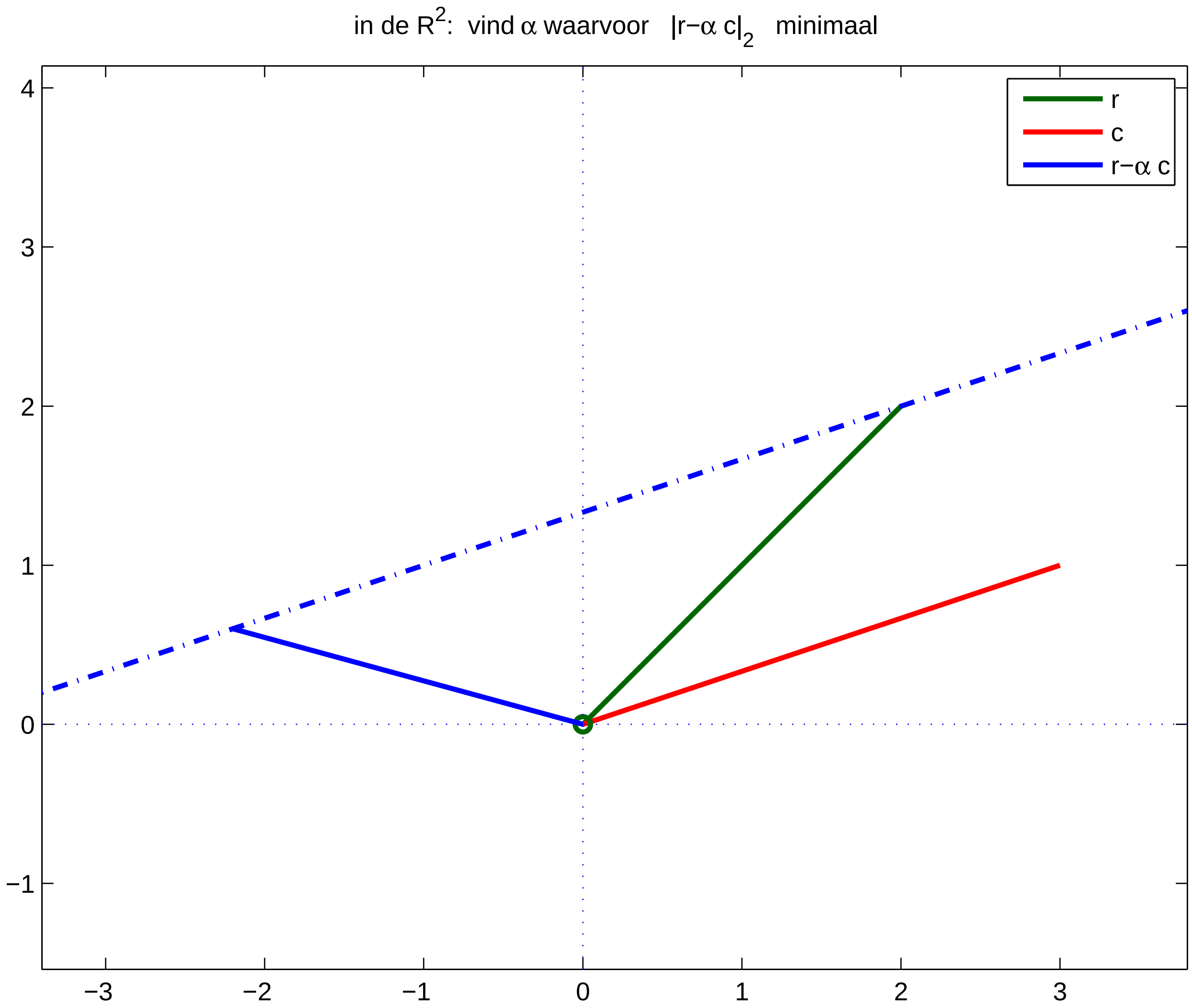


r
c
r−α c
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3

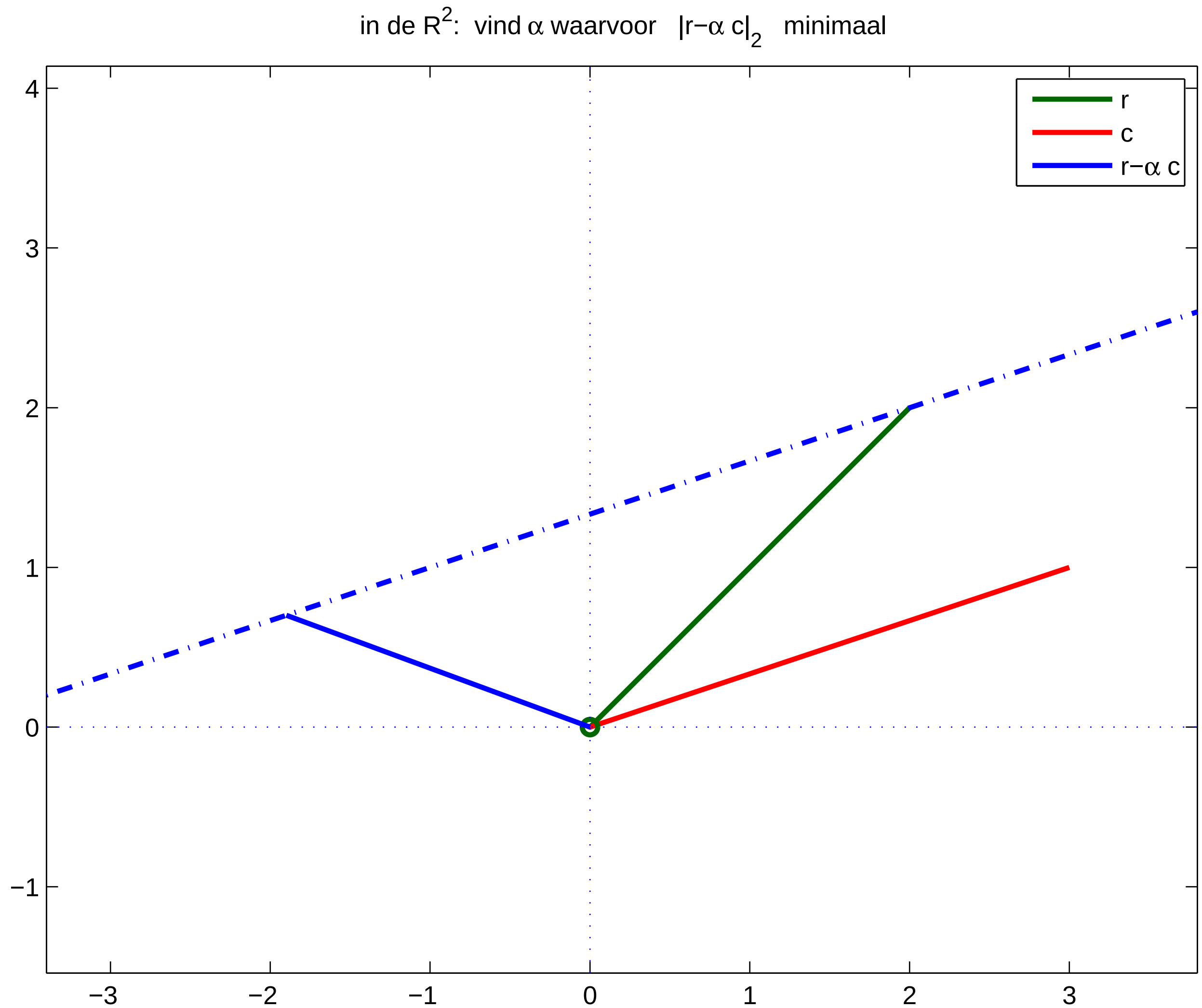

in de R2: vind α waarvoor |r−α c|2 minimaal
r
c
r−α c
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3

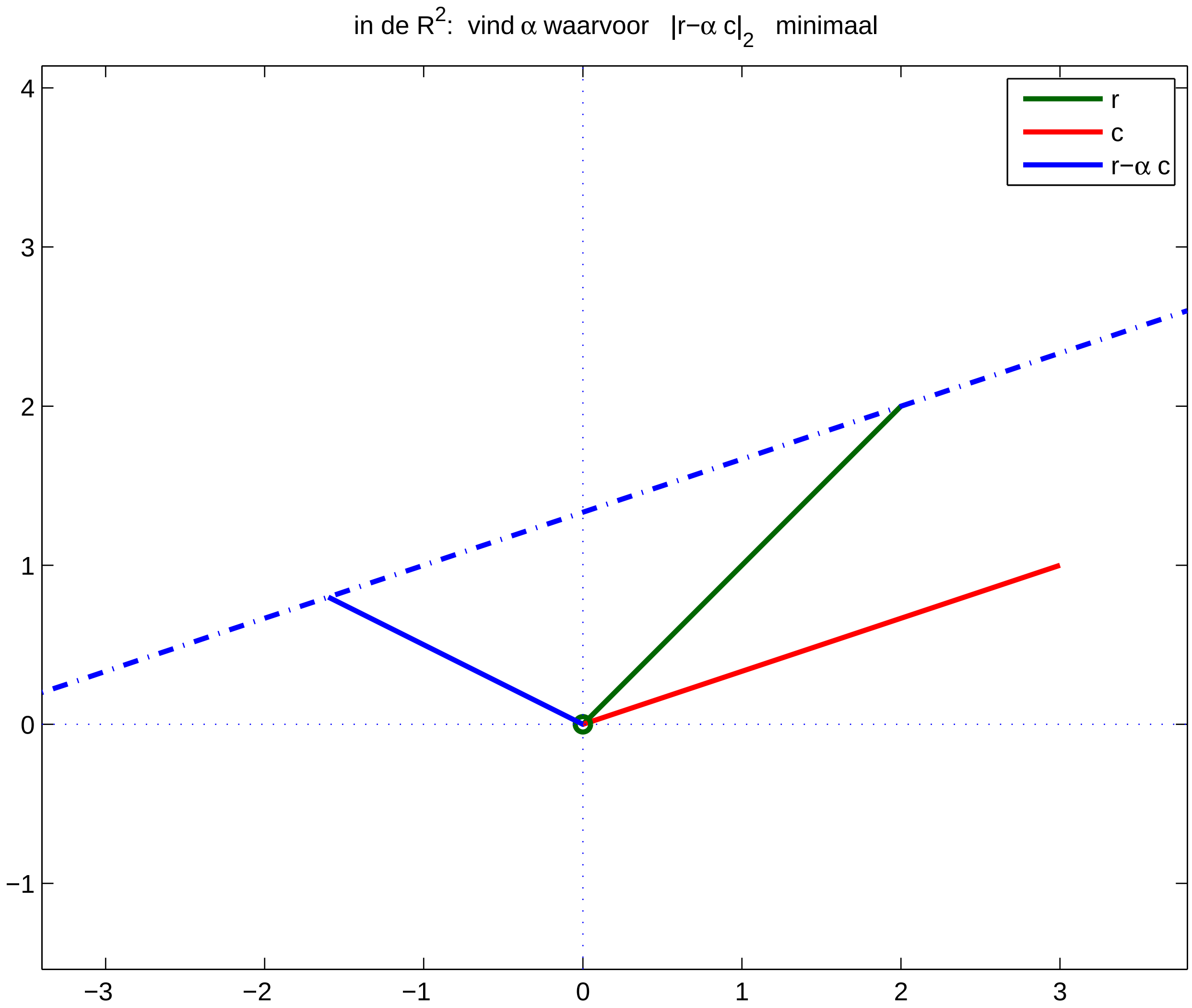

in de R2: vind α waarvoor |r−α c|2 minimaal
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3
r
c
r−α c

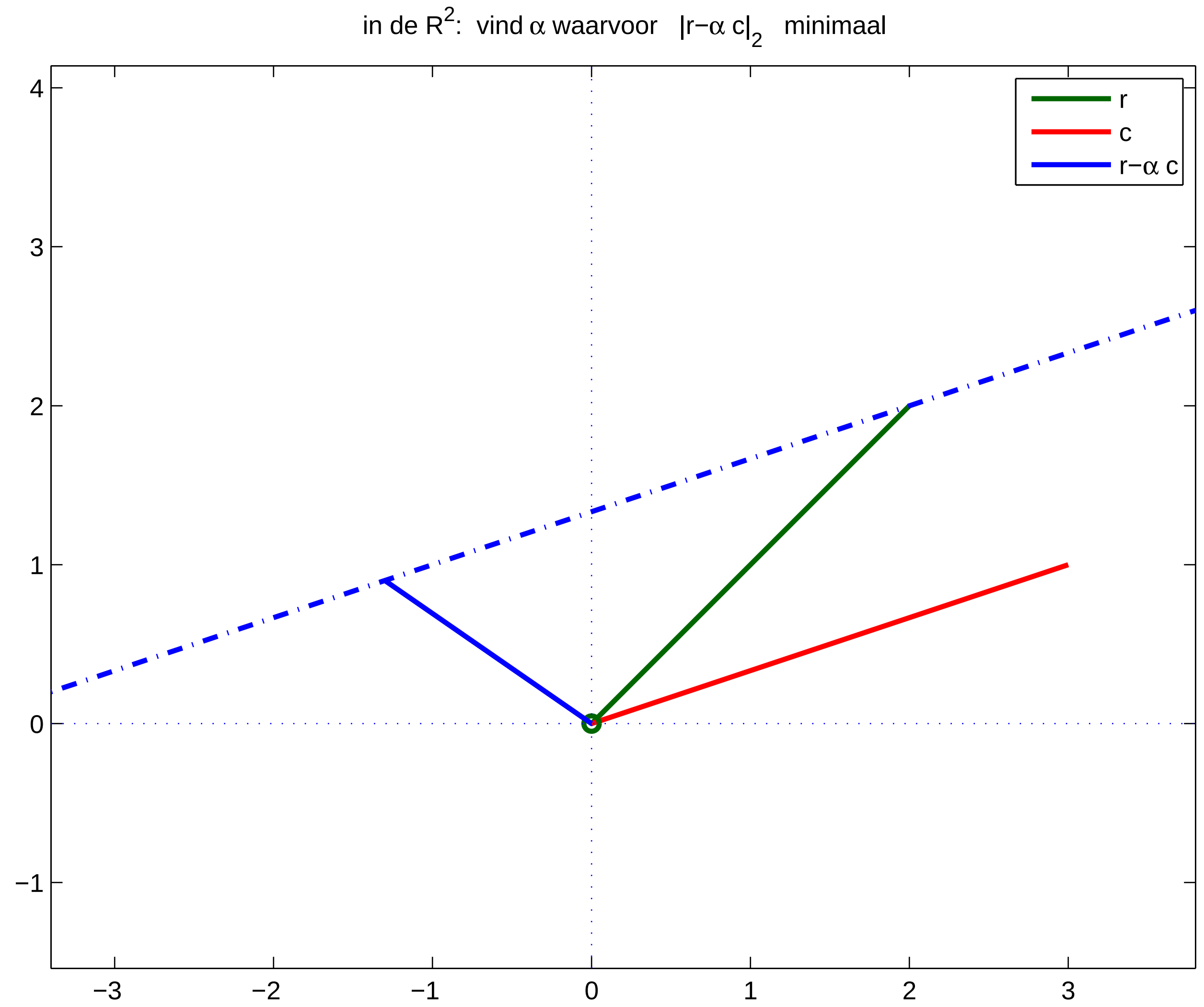

in de R^2: vind α waarvoor |r−α c|$_2$ minimaal
r
c
r−α c
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3

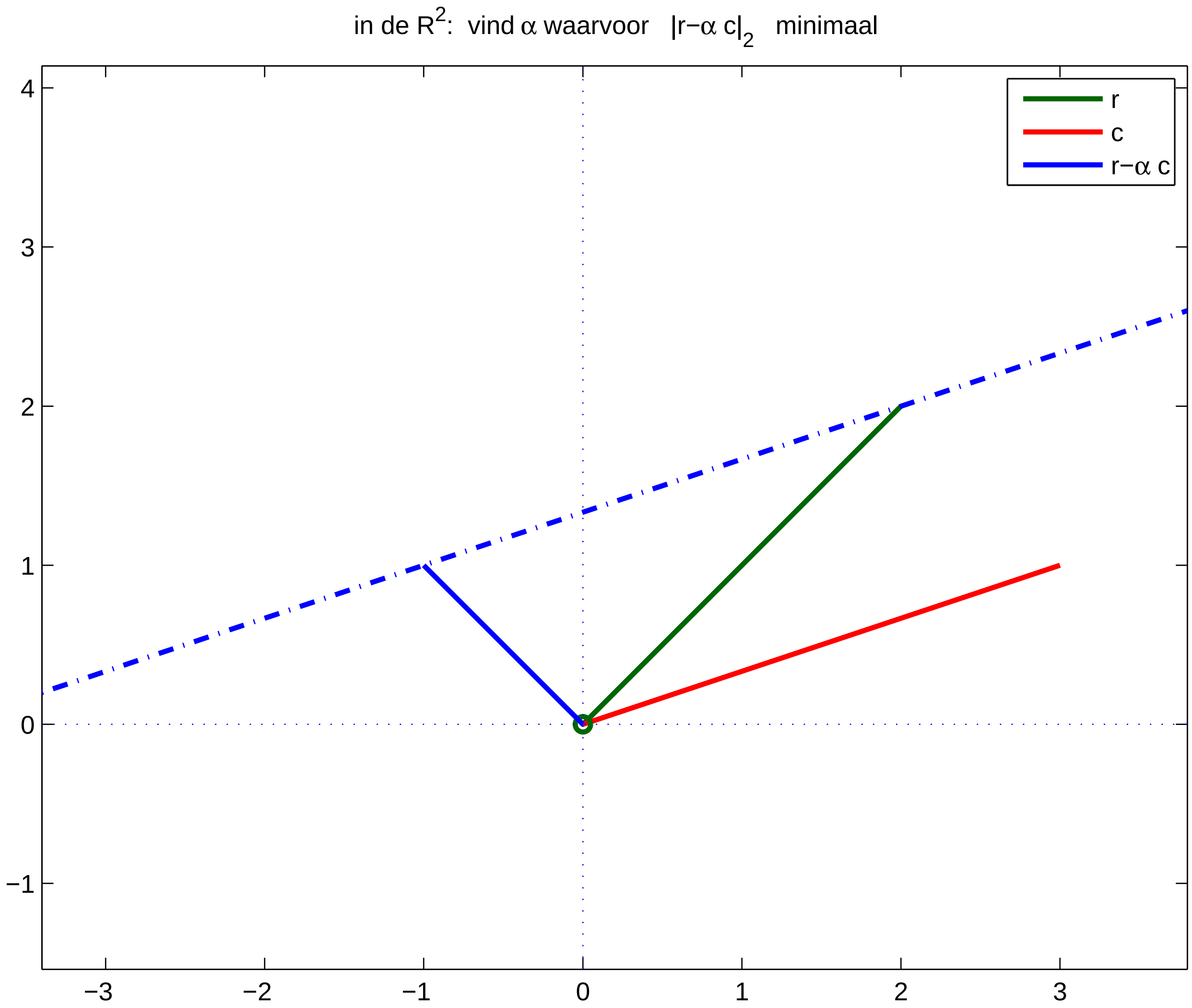


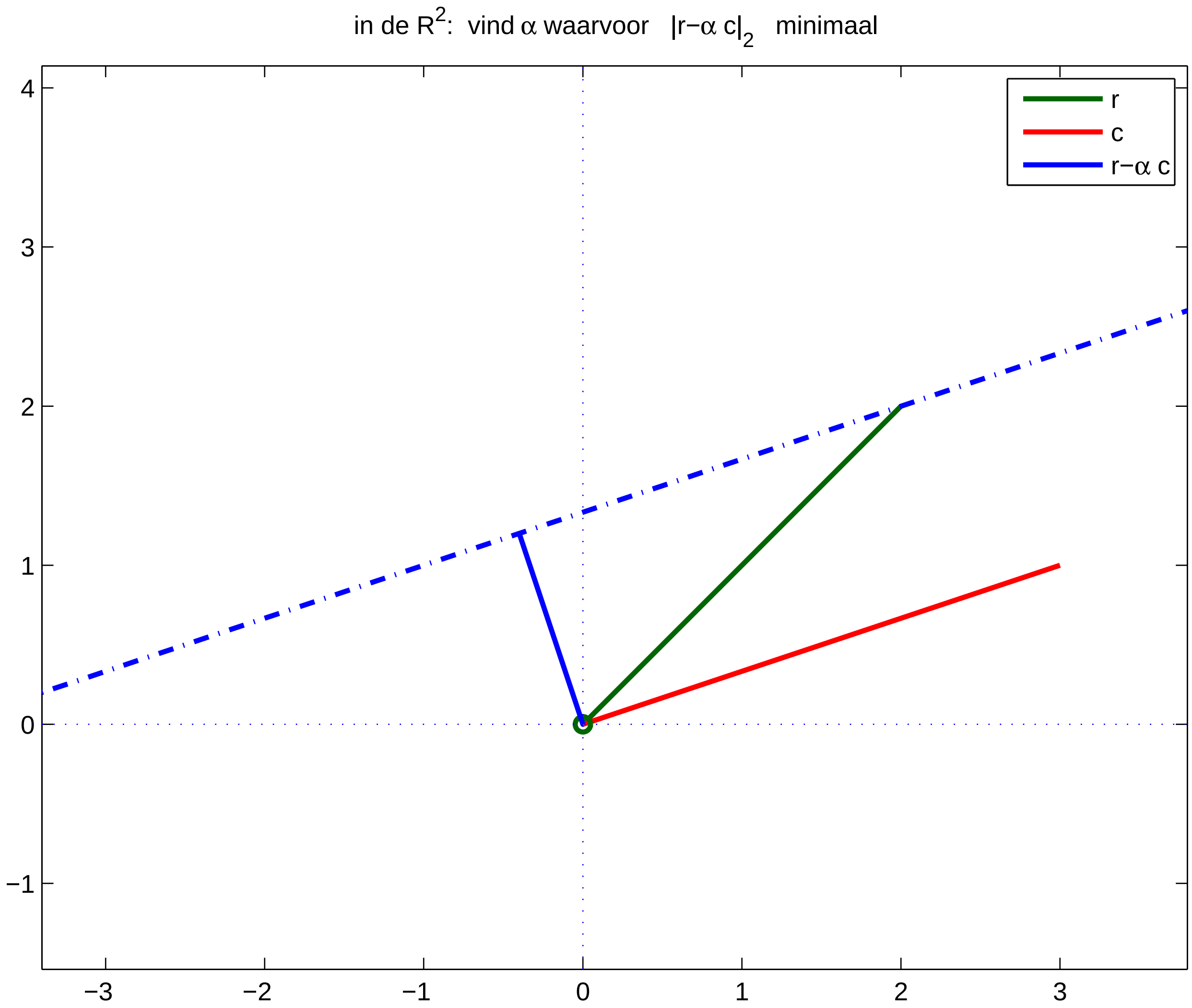
in de R2: vind α waarvoor |r−α c|2 minimaal
r
c
r−α c
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3


in de R^2: vind α waarvoor $|r-\alpha\ c|_2$ minimaal
r
c
r−α c


in de R²: vind α waarvoor |r−α c|₂ minimaal
r
c
r−α c
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3

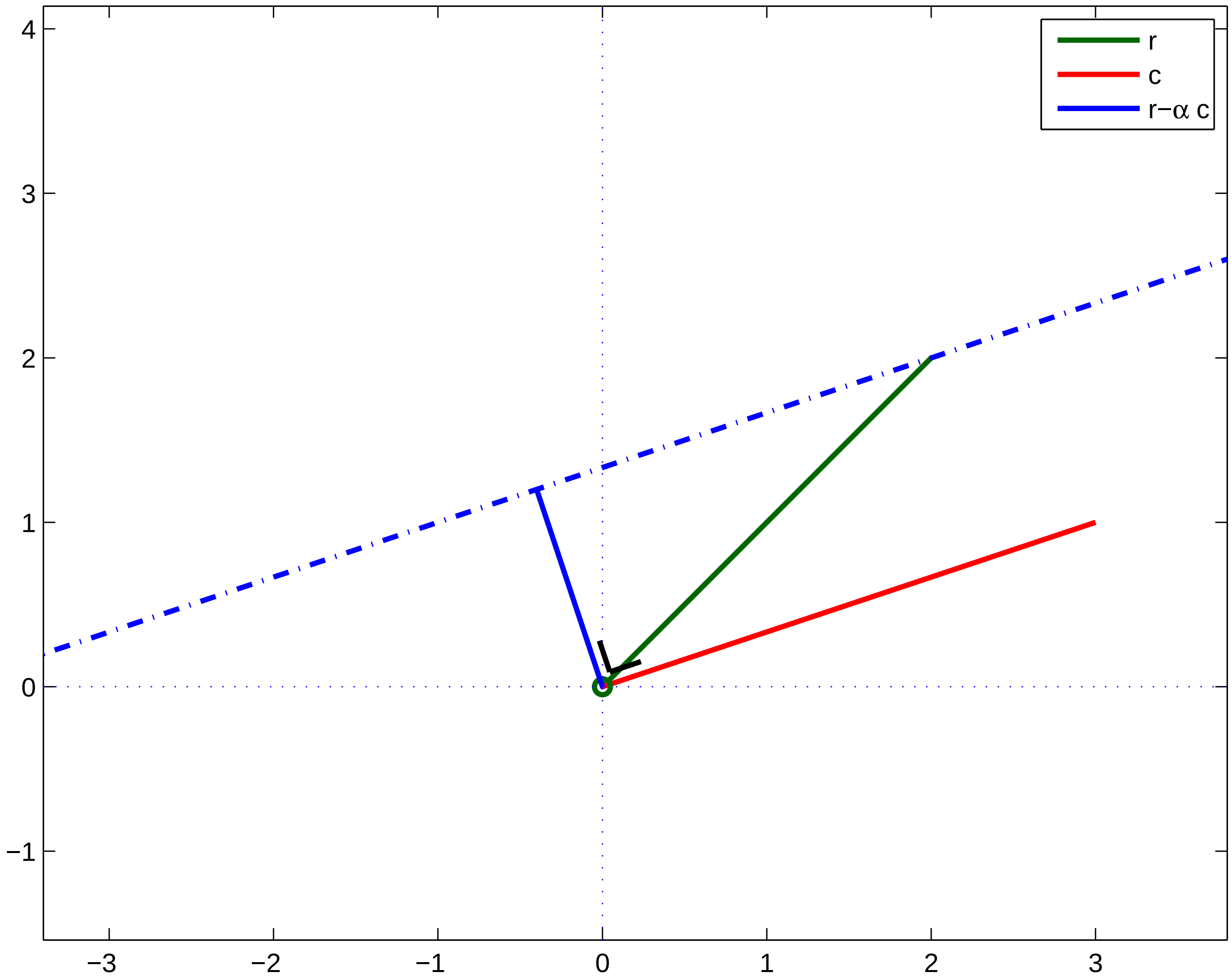

in de R2: vind α waarvoor |r−α c|2 minimaal
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3
r
c
r−α c


in de R^2: vind α waarvoor $|r-\alpha\ c|_2$ minimaal
r
c
r−α c
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3



in de R2: vind α waarvoor |r−α c|2 minimaal
r
c
r−α c
4
3
2
1
0
−1
−3
−2
−1
0
1
2
3

in de R^2: vind α waarvoor $|r-\alpha\, c|_2$ minimaal



Local Minimal Residuals

```
Choose tol > 0, x_0, k_max,
Compute r_0 = b - A x_0
For k = 0, 1, 2, ..., k_max
    Stop if ||r_k||_2 <= tol ||b||_2
    u_k = r_k
    c_k = A u_k
    sigma_k = c_k^* c_k,  alpha_k = c_k^* r_k / sigma_k
    x_{k+1} = x_k + alpha_k u_k
    r_{k+1} = r_k - alpha_k c_k
end for
```

Choose $tol > 0$, $\mathbf{x}_0$, $k_{\max}$,

Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

For $k = 0, 1, 2, \ldots, k_{\max}$

- Stop if $\|\mathbf{r}_k\|_2 \leq tol\|\mathbf{b}\|_2$
- $\mathbf{u}_k = \mathbf{r}_k$
- $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$
- $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha_k = \mathbf{c}_k^* \mathbf{r}_k / \sigma_k$
- $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{u}_k$
- $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{c}_k$

end for

Iteratief

Stelling. LMR convergeert als

Re(λ) > 0 voor alle eigenwaarden λ van **A**.

Voorbeeld. Re(λ) > 0 voor alle eigw. van **A**:

1) Grondwaterstroming (geschikte discretisatie)

2) Verspreiding gif, $c > 0$, en h_x, h_y voldoende klein

Local Minimal Residuals

```
Choose tol > 0, x_0, k_max,
Compute r_0 = b - A x_0
For k = 0, 1, 2, ..., k_max
    Stop if ||r_k||_2 <= tol ||b||_2
    u_k = r_k
    c_k = A u_k
    sigma_k = c_k^* c_k,  alpha_k = c_k^* r_k / sigma_k
    x_{k+1} = x_k + alpha_k u_k
    r_{k+1} = r_k - alpha_k c_k
end for
```

Choose $tol > 0$, $\mathbf{x}_0$, $k_{\max}$,

Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

For $k = 0, 1, 2, \ldots, k_{\max}$

- Stop if $\|\mathbf{r}_k\|_2 \leq tol \|\mathbf{b}\|_2$
- $\mathbf{u}_k = \mathbf{r}_k$
- $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$
- $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha_k = \mathbf{c}_k^* \mathbf{r}_k / \sigma_k$
- $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{u}_k$
- $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{c}_k$

end for

LMR (geheugen vriendelijk)

Choose $\mathit{tol} > 0$, $\mathbf{x}$, k_{max},
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
For $k = 0, 1, 2, \ldots, k_{\mathrm{max}}$

  Stop if $\|\mathbf{r}\|_2 \leq \mathit{tol}\|\mathbf{b}\|_2$
  $\mathbf{u} \leftarrow \mathbf{r}$
  $\mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$
  $\sigma \leftarrow \mathbf{c}^*\mathbf{c}$, $\alpha \leftarrow \mathbf{c}^*\mathbf{r}/\sigma$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$

end for

LMR (geheugen vriendelijk)

```
Choose tol > 0, x, k_max,
Compute r = b - Ax
For k = 0, 1, 2, ..., k_max
    Stop if ||r||_2 <= tol ||b||_2
    u <- r
    c <- Au
    σ <- c*c, α <- c*r/σ
    x <- x + α u
    r <- r - α c
end for
```

Choose $tol > 0$, $\mathbf{x}$, $k_{\max}$,

Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$

For $k = 0, 1, 2, \ldots, k_{\max}$

Stop if $\|\mathbf{r}\|_2 \leq tol \|\mathbf{b}\|_2$

$\mathbf{u} \leftarrow \mathbf{r}$

$\mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$

$\sigma \leftarrow \mathbf{c}^*\mathbf{c}$, $\alpha \leftarrow \mathbf{c}^*\mathbf{r}/\sigma$

$\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$

$\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$

end for

Loop

LMR (geheugen vriendelijk)

```
Choose tol > 0, x, k_max,
Compute r = b - Ax
For k = 0, 1, 2, ..., k_max
    Stop if ||r||_2 <= tol ||b||_2
    u <- r
    c <- Au
    σ <- c*c, α <- c*r/σ
    x <- x + α u
    r <- r - α c
end for
```

Choose $tol > 0$, $\mathbf{x}$, $k_{\max}$,
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
For $k = 0, 1, 2, \ldots, k_{\max}$

  Stop if $\|\mathbf{r}\|_2 \leq tol \|\mathbf{b}\|_2$
  $\mathbf{u} \leftarrow \mathbf{r}$
  $\mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$
  $\sigma \leftarrow \mathbf{c}^*\mathbf{c}$, $\alpha \leftarrow \mathbf{c}^*\mathbf{r}/\sigma$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$

end for

BLAS. Per stap: 1 MV, 2 AXPYs, 2 (of 3) DOTs

LMR (geheugen vriendelijk)

```
Choose tol > 0, x, k_max,
Compute r = b - Ax
For k = 0, 1, 2, ..., k_max
    Stop if ||r||_2 <= tol ||b||_2
    u <- r
    c <- Au
    σ <- c*c, α <- c*r/σ
    x <- x + α u
    r <- r - α c
end for
```

Choose $tol > 0$, $\mathbf{x}$, $k_{\max}$,
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
For $k = 0, 1, 2, \ldots, k_{\max}$

- Stop if $\|\mathbf{r}\|_2 \leq tol \|\mathbf{b}\|_2$
- $\mathbf{u} \leftarrow \mathbf{r}$
- $\mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$
- $\sigma \leftarrow \mathbf{c}^*\mathbf{c}$, $\alpha \leftarrow \mathbf{c}^*\mathbf{r}/\sigma$
- $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
- $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$

end for

Stopcriterium: $\|\mathbf{r}\|_2$ klein, of te veel stappen

LMR (geheugen vriendelijk)

```
Choose tol > 0, x, k_max,
Compute r = b - Ax
For k = 0, 1, 2, ..., k_max
    Stop if ||r||_2 <= tol ||b||_2
    u <- r
    c <- Au
    σ <- c*c, α <- c*r/σ
    x <- x + α u
    r <- r - α c
end for
```

$$
\begin{array}{l}
\texttt{Choose } \mathit{tol} > 0,\ \mathbf{x},\ k_{\max}, \\
\texttt{Compute } \mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} \\
\texttt{For } k = 0, 1, 2, \ldots, k_{\max} \\
\quad \texttt{Stop if } \|\mathbf{r}\|_2 \leq \mathit{tol} \|\mathbf{b}\|_2 \\
\quad \mathbf{u} \leftarrow \mathbf{r} \\
\quad \mathbf{c} \leftarrow \mathbf{A}\mathbf{u} \\
\quad \sigma \leftarrow \mathbf{c}^*\mathbf{c},\ \alpha \leftarrow \mathbf{c}^*\mathbf{r}/\sigma \\
\quad \mathbf{x} \leftarrow \mathbf{x} + \alpha\, \mathbf{u} \\
\quad \mathbf{r} \leftarrow \mathbf{r} - \alpha\, \mathbf{c} \\
\texttt{end for}
\end{array}
$$

Start: gewoonlijk kiest men $\mathbf{x}_0 = \mathbf{0}$.

LMR (geheugen vriendelijk)

```
Choose tol > 0, x, k_max,
Compute r = b - Ax
For k = 0, 1, 2, ..., k_max
    Stop if ||r||_2 <= tol ||b||_2
    u <- r
    c <- Au
    σ <- c*c, α <- c*r/σ
    x <- x + α u
    r <- r - α c
end for
```

Choose $tol > 0$, $\mathbf{x}$, $k_{\max}$,
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
For $k = 0, 1, 2, \ldots, k_{\max}$

  Stop if $\|\mathbf{r}\|_2 \leq tol \|\mathbf{b}\|_2$
  $\mathbf{u} \leftarrow \mathbf{r}$
  $\mathbf{c} \leftarrow \mathbf{A}\mathbf{u}$
  $\sigma \leftarrow \mathbf{c}^*\mathbf{c}$, $\alpha \leftarrow \mathbf{c}^*\mathbf{r}/\sigma$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}$

end for

Geheugen: op te slaan 4 (of 5?) vectoren

LMR heeft de ingrediënten die typisch zijn voor iteratieve methodes:

- Loop (iteratie stap).
- Paar BLAS operaties (MV, AXPYs, DOTs) per stap.
- Stopcriterium:

  stop als – residu voldoende klein is (success) of
  – als het aantal stappen te groot wordt (fail).
- Geheugen vriendelijk.

LMR heeft de ingrediënten die typisch zijn voor iteratieve methodes:

- Loop (iteratie stap).
- Paar BLAS operaties (MV, AXPYs, DOTs) per stap.
- Stopcriterium:

  stop als – residu voldoende klein is (success) of
  – als het aantal stappen te groot wordt (fail).
- Geheugen vriendelijk.

Belangrijke vragen (voor iedere iteratieve methode):

- Hoe snel convergeert het proces
  (d.w.z., wat is de kleinste k waarvoor $\|\mathbf{r}_k\|_2 \leq \mathit{tol}\|\mathbf{b}\|_2$)?
- Wat zijn de kosten per stap?

(de totale kosten volgen dan)

Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Generalized Conjugate Residuals

Idee.

In LMR is $\mathbf{r}_k$ alleen geminimaliseerd in de richting $\mathbf{c}_k$.
Echter de vectoren $\mathbf{c}_{k-1}, \mathbf{c}_{k-2}, \ldots$ zijn ook berekend.
Levert minimaliseren in al deze richtingen een kleiner residu?

Generalized Conjugate Residuals

Idee.

In LMR is $\mathbf{r}_k$ alleen geminimaliseerd in de richting $\mathbf{c}_k$.
Echter de vectoren $\mathbf{c}_{k-1}, \mathbf{c}_{k-2}, \ldots$ zijn ook berekend.
Levert minimaliseren in al deze richtingen een kleiner residu?

Stelling. (α_j) zo dat $\|\mathbf{r}_k - \sum_{j=0}^{k} \alpha_j \mathbf{c}_j\|_2$ minimaal

$$\Leftrightarrow \qquad \mathbf{r}_{k+1} \equiv \mathbf{r}_k - \sum_{j=0}^{k} \alpha_j \mathbf{c}_j \perp \mathbf{c}_i \text{ alle } i = 0, \ldots, k.$$

Generalized Conjugate Residuals

Idee.

In LMR is $\mathbf{r}_k$ alleen geminimaliseerd in de richting $\mathbf{c}_k$.
Echter de vectoren $\mathbf{c}_{k-1}, \mathbf{c}_{k-2}, \ldots$ zijn ook berekend.
Levert minimaliseren in al deze richtingen een kleiner residu?

Stelling. (α_j) zo dat $\|\mathbf{r}_k - \sum_{j=0}^{k} \alpha_j \mathbf{c}_j\|_2$ minimaal

$\Leftrightarrow \qquad \mathbf{r}_{k+1} \equiv \mathbf{r}_k - \sum_{j=0}^{k} \alpha_j \mathbf{c}_j \perp \mathbf{c}_i$ alle $i = 0, \ldots, k$.

Stelling. (α_j) zo dat $\|\mathbf{r}_k - \sum_{j=0}^{k} \alpha_j \mathbf{c}_j\|_2$ minimaal.

$\mathbf{r}_k, \mathbf{c}_k \perp \mathrm{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1}) \;\Rightarrow \mathbf{r}_{k+1} \equiv \mathbf{r}_k - \alpha_k \mathbf{c}_k \perp \mathbf{c}_k$

Dus $\alpha_0 = \ldots = \alpha_{k-1} = 0$ en $\alpha_k = \dfrac{\mathbf{c}_k^* \mathbf{r}_k}{\mathbf{c}_k^* \mathbf{c}_k}$

$\mathbf{r}_k$, $\mathbf{c}_k \perp \text{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1})$

Gebruik Gram-Schmidt om, voor iedere $k = 1, 2, \ldots$, de vector $\mathbf{c}_k$ loodrecht $\text{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1})$ te krijgen

Stelling [Modified Gram-Schmidt].
Gegeven vectoren $\widetilde{\mathbf{c}}_0, \widetilde{\mathbf{c}}_1, \ldots$.
Voor $k = 1, 2, \ldots$, bereken *(negeer lege loops)*

$$\boxed{\begin{array}{l} \mathbf{c}_k = \widetilde{\mathbf{c}}_k \\ \text{For } j = 0, 1, \ldots, k-1 \\ \qquad \beta_j = \mathbf{c}_j^* \widetilde{\mathbf{c}}_k / \sigma_j \\ \qquad \mathbf{c}_k \leftarrow \mathbf{c}_k - \beta_j \, \mathbf{c}_j \\ \text{end for} \\ \sigma_k = \mathbf{c}_k^* \mathbf{c}_k \end{array}}$$

Dan $\quad \text{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_k) = \text{Span}(\widetilde{\mathbf{c}}_0, \widetilde{\mathbf{c}}_1, \ldots, \widetilde{\mathbf{c}}_k)$ alle k

en $\mathbf{c}_i \perp \mathbf{c}_j$ alle $i, j, i \neq j$.

$\mathbf{r}_k, \mathbf{c}_k \perp \text{Span}(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1})$

Gebruik Gram-Schmidt om, voor iedere $k = 1, 2, \ldots,$ de vector $\mathbf{c}_k$ loodrecht Span$(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{k-1})$ te krijgen

Stelling [Modified Gram-Schmidt].

Gegeven vectoren $\widetilde{\mathbf{c}}_0, \widetilde{\mathbf{c}}_1, \ldots$.

Voor $k = 1, 2, \ldots,$ bereken

$$\boxed{\begin{array}{l} \mathbf{c}_k = \widetilde{\mathbf{c}}_k \\ \text{For } j = 0, 1, \ldots, k-1 \\ \qquad \beta_j = \mathbf{c}_j^* \widetilde{\mathbf{c}}_k / \sigma_j \\ \qquad \mathbf{c}_k \leftarrow \mathbf{c}_k - \beta_j \, \mathbf{c}_j \\ \text{end for} \\ \sigma_k = \mathbf{c}_k^* \mathbf{c}_k \end{array}}$$

Dan $\quad$ Span$(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_k) =$ Span$(\widetilde{\mathbf{c}}_0, \widetilde{\mathbf{c}}_1, \ldots, \widetilde{\mathbf{c}}_k)$ alle k

en $\mathbf{c}_i \perp \mathbf{c}_j$ alle $i, j, i \neq j$.

Generalized Conjugate Residuals

```
Choose tol > 0, x_0, k_max,
Compute r_0 = b - A x_0
For k = 0, 1, 2, ..., k_max
    Stop if ||r_k||_2 <= tol ||b||_2
    u_k = r_k
    c_k = A u_k
    For j = 0, 1, 2, ..., k-1
        β_j = c_j^* c_k / σ_j
        u_k ← u_k - β_j u_j
        c_k ← c_k - β_j c_j
    end for
    σ_k = c_k^* c_k,   α_k = c_k^* r_k / σ_k
    x_{k+1} = x_k + α_k u_k
    r_{k+1} = r_k - α_k c_k
end for
```

Choose $tol > 0$, $\mathbf{x}_0$, $k_{\max}$,

Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$

For $k = 0, 1, 2, \ldots, k_{\max}$

- Stop if $\|\mathbf{r}_k\|_2 \le tol\|\mathbf{b}\|_2$
- $\mathbf{u}_k = \mathbf{r}_k$
- $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$
- For $j = 0, 1, 2, \ldots, k-1$
  - $\beta_j = \mathbf{c}_j^* \mathbf{c}_k / \sigma_j$
  - $\mathbf{u}_k \leftarrow \mathbf{u}_k - \beta_j \mathbf{u}_j$
  - $\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta_j \mathbf{c}_j$
- end for
- $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha_k = \mathbf{c}_k^* \mathbf{r}_k / \sigma_k$
- $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{u}_k$
- $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{c}_k$

end for

Generalized Conjugate Residuals

Choose $tol > 0$, $\mathbf{x}_0$, $k_{\max}$,
Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$
For $k = 0, 1, 2, \ldots, k_{\max}$

    Stop if $\|\mathbf{r}_k\|_2 \leq tol\|\mathbf{b}\|_2$
    $\mathbf{u}_k = \mathbf{r}_k$
    $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$
    For $j = 0, 1, 2, \ldots, k-1$
        $\beta_j = \mathbf{c}_j^* \mathbf{c}_k / \sigma_j$
        $\mathbf{u}_k \leftarrow \mathbf{u}_k - \beta_j \mathbf{u}_j$
        $\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta_j \mathbf{c}_j$
    end for
    $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha_k = \mathbf{c}_k^* \mathbf{r}_k / \sigma_k$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{u}_k$
    $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{c}_k$

end for

GCR (geheugen vriendelijk)

`Choose` $tol > 0$, $\mathbf{x}$, $k_{\max}$,	
`Compute` $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$	
`For` $k = 0, 1, 2, \ldots, k_{\max}$	
`Stop if` $\|\mathbf{r}\|_2 \leq tol\|\mathbf{b}\|_2$	2n flop
$\mathbf{u}_k = \mathbf{r}$	
$\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$	13n flop (3d)
`For` $j = 0, 1, 2, \ldots, k-1$	
$\beta \leftarrow \mathbf{c}_j^* \mathbf{c}_k / \sigma_j$	2n flop
$\mathbf{u}_k \leftarrow \mathbf{u}_k - \beta\, \mathbf{u}_j$	2n flop
$\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta\, \mathbf{c}_j$	2n flop
`end for`	
$\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha \leftarrow \mathbf{c}_k^* \mathbf{r} / \sigma_k$	4n flop
$\mathbf{x} \leftarrow \mathbf{x} + \alpha\, \mathbf{u}_k$	2n flop
$\mathbf{r} \leftarrow \mathbf{r} - \alpha\, \mathbf{c}_k$	2n flop
`end for`	

totaal 3-d: 23n + 6nk flop

totaal 2-d: 19n + 6nk flop

GCR (geheugen vriendelijk)

Choose $tol > 0$, $\mathbf{x}$, $k_{\max}$,
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
For $k = 0, 1, 2, \ldots, k_{\max}$

  Stop if $\|\mathbf{r}\|_2 \leq tol\|\mathbf{b}\|_2$ — 2n flop
  $\mathbf{u}_k = \mathbf{r}$
  $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$ — 13n flop (3d)
  For $j = 0, 1, 2, \ldots, k-1$
    $\beta \leftarrow \mathbf{c}_j^* \mathbf{c}_k / \sigma_j$ — 2n flop
    $\mathbf{u}_k \leftarrow \mathbf{u}_k - \beta\, \mathbf{u}_j$ — 2n flop
    $\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta\, \mathbf{c}_j$ — 2n flop
  end for
  $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha \leftarrow \mathbf{c}_k^* \mathbf{r} / \sigma_k$ — 4n flop
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\, \mathbf{u}_k$ — 2n flop
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\, \mathbf{c}_k$ — 2n flop

end for

totaal 3-d: 23n + 6nk flop
totaal 2-d: 19n + 6nk flop

GCR (geheugen vriendelijk)

Choose $tol > 0$, $\mathbf{x}$, $k_{\max}$,
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
For $k = 0, 1, 2, \ldots, k_{\max}$
  Stop if $\|\mathbf{r}\|_2 \leq tol\|\mathbf{b}\|_2$ — 2n flop
  $\mathbf{u}_k = \mathbf{r}$
  $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$ — 13n flop (3d)
  For $j = 0, 1, 2, \ldots, k-1$
    $\beta \leftarrow \mathbf{c}_j^* \mathbf{c}_k / \sigma_j$ — 2n flop
    $\mathbf{u}_k \leftarrow \mathbf{u}_k - \beta\,\mathbf{u}_j$ — 2n flop
    $\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta\,\mathbf{c}_j$ — 2n flop
  end for
  $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha \leftarrow \mathbf{c}_k^* \mathbf{r} / \sigma_k$ — 4n flop
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u}_k$ — 2n flop
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c}_k$ — 2n flop
end for

totaal 3-d: 23n + 6nk flop
totaal 2-d: 19n + 6nk flop

GCR (geheugen vriendelijk)

```
Choose tol > 0, x, k_max,
Compute r = b - Ax
For k = 0, 1, 2, ..., k_max
    Stop if ||r||_2 <= tol ||b||_2          2n flop
    u_k = r
    c_k = A u_k                               13n flop (3d)
    For j = 0, 1, 2, ..., k-1
        β <- c_j^* c_k / σ_j                  2n flop
        u_k <- u_k - β u_j                    2n flop
        c_k <- c_k - β c_j                    2n flop
    end for
    σ_k = c_k^* c_k,  α <- c_k^* r / σ_k      4n flop
    x <- x + α u_k                            2n flop
    r <- r - α c_k                            2n flop
end for
```

totaal 3-d: 23n + 6nk flop

totaal 2-d: 19n + 6nk flop

GCR (Variant)

```
Choose tol > 0, x, k_max,
Compute r = b - Ax
For k = 0,1,2,...,k_max
    Stop if ||r||_2 <= tol ||b||_2
    u_k = r
    c <- A u_k
    For j = 0,1,2,...,k-1
        beta <- c_j^* c / sigma_j
        u_k <- u_k - beta u_j
    end for
    c_k = A u_k
    sigma_k = c_k^* c_k,  alpha <- c_k^* r / sigma_k
    x <- x + alpha u_k
    r <- r - alpha c_k
end for
```

Choose $tol > 0$, $\mathbf{x}$, $k_{\max}$,
Compute $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$
For $k = 0, 1, 2, \ldots, k_{\max}$
  Stop if $\|\mathbf{r}\|_2 \leq tol \|\mathbf{b}\|_2$
  $\mathbf{u}_k = \mathbf{r}$
  $\mathbf{c} \leftarrow \mathbf{A}\mathbf{u}_k$
  For $j = 0, 1, 2, \ldots, k-1$
    $\beta \leftarrow \mathbf{c}_j^* \mathbf{c} / \sigma_j$
    $\mathbf{u}_k \leftarrow \mathbf{u}_k - \beta\, \mathbf{u}_j$
  end for
  $\mathbf{c}_k = \mathbf{A}\mathbf{u}_k$
  $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\alpha \leftarrow \mathbf{c}_k^* \mathbf{r} / \sigma_k$
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\, \mathbf{u}_k$
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\, \mathbf{c}_k$
end for

Othogonaliseren versus orthonormaliseren

$\mathbf{c}_k = \widetilde{\mathbf{c}}_k$ For $j = 1, 2, \ldots, k-1$ $\quad\beta \leftarrow \mathbf{c}_j^* \mathbf{c}_k / \sigma_j$ $\quad\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta\, \mathbf{c}_j$ end for $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$	$\mathbf{c}_k = \widetilde{\mathbf{c}}_k$ For $j = 1, 2, \ldots, k-1$ $\quad\beta \leftarrow \mathbf{c}_j^* \mathbf{c}_k$ $\quad\mathbf{c}_k \leftarrow \mathbf{c}_k - \beta\, \mathbf{c}_j$ end for $\sigma_k = \mathbf{c}_k^* \mathbf{c}_k$, $\mathbf{c}_k \leftarrow \mathbf{c}_k / \sqrt{\sigma_k}$

Orthogonaliseren (versie links)

kost $4n(k-1) + 2n$ flop.

Ortho**norm**aliseren (=orthogo. + normeren, versie rechts)

kost $4n(k-1) + 3n$ flop

Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\mathsf{GCR}}\|_2 \le \|\mathbf{r}_k^{\mathsf{LMR}}\|_2 \le \|\mathbf{r}_k^{\mathsf{Richardson}}\|_2$$

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\text{GCR}}\|_2 \leq \|\mathbf{r}_k^{\text{LMR}}\|_2 \leq \|\mathbf{r}_k^{\text{Richardson}}\|_2$$

Stelling. $\varepsilon \in (0, 1]$. Richardson: α zo dat

$$\alpha\lambda \in \{\zeta \in \mathbb{C} \mid |1 - \zeta| < 1 - \varepsilon\} \text{ alle eig. } \lambda \text{ van } \mathbf{A} \qquad \Rightarrow$$

$$\frac{\|\mathbf{r}_k^{\text{Richardson}}\|_2}{\|\mathbf{r}_0\|_2} \leq (1 - \varepsilon)^k \leq \exp(-k\varepsilon)$$

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\text{GCR}}\|_2 \leq \|\mathbf{r}_k^{\text{LMR}}\|_2 \leq \|\mathbf{r}_k^{\text{Richardson}}\|_2$$

Stelling. $\varepsilon \in (0, 1]$. Richardson: α zo dat

$$\alpha\lambda \in \{\zeta \in \mathbb{C} \mid |1 - \zeta| < 1 - \varepsilon\} \text{ alle eig. } \lambda \text{ van } \mathbf{A} \qquad \Rightarrow$$

$$\frac{\|\mathbf{r}_k^{\text{Richardson}}\|_2}{\|\mathbf{r}_0\|_2} \leq (1 - \varepsilon)^k \leq \exp(-k\varepsilon)$$

Grondwatervergl.: $\lambda > 0$ alle eig. λ van $\mathbf{A}$.

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\mathrm{GCR}}\|_2 \leq \|\mathbf{r}_k^{\mathrm{LMR}}\|_2 \leq \|\mathbf{r}_k^{\mathrm{Richardson}}\|_2$$

Stelling. $\varepsilon \in (0,1]$. Richardson: α zo dat

$$\alpha\lambda \in \{\zeta \in \mathbb{C} \mid |1-\zeta| < 1-\varepsilon\} \text{ alle eig. } \lambda \text{ van } \mathbf{A} \qquad \Rightarrow$$

$$\frac{\|\mathbf{r}_k^{\mathrm{Richardson}}\|_2}{\|\mathbf{r}_0\|_2} \leq (1-\varepsilon)^k \leq \exp(-k\varepsilon)$$

Grondwatervergl.: $\lambda > 0$ alle eig. λ van $\mathbf{A}$.

Als $\lambda \in [\lambda_1, \lambda_n] \subset (0, \infty)$ $\forall \lambda$, dan optimale α:

$$\alpha^{\mathrm{opt}} = \frac{2}{\lambda_1+\lambda_n} \qquad \text{en} \qquad \varepsilon = \frac{2\,\lambda_1}{\lambda_1+\lambda_n} \leq 2\,\frac{\lambda_1}{\lambda_n}.$$

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\mathrm{GCR}}\|_2 \le \|\mathbf{r}_k^{\mathrm{LMR}}\|_2 \le \|\mathbf{r}_k^{\mathrm{Richardson}}\|_2$$

Grondwatervergl., met, voor Richardson, $\alpha = \alpha^{\mathrm{opt}}$

$$\rho_k^{\mathrm{LMR}} \le \rho_k^{\mathrm{Richardson}} \equiv \frac{\|\mathbf{r}_k^{\mathrm{Richardson}}\|_2}{\|\mathbf{r}_0\|_2} \le \exp\left(-2k\,\frac{\lambda_1}{\lambda_n}\right)$$

Opmerking. Schatting voor residue-reductie
is voor LMR **niet** beter dan voor optimale Richardson.
Echter in LMR wordt de "correcte" α vanzelf gevonden.

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\text{GCR}}\|_2 \leq \|\mathbf{r}_k^{\text{LMR}}\|_2 \leq \|\mathbf{r}_k^{\text{Richardson}}\|_2$$

Grondwatervergl., met, voor Richardson, $\alpha = \alpha^{\text{opt}}$

$$\rho_k^{\text{LMR}} \leq \rho_k^{\text{Richardson}} \equiv \frac{\|\mathbf{r}_k^{\text{Richardson}}\|_2}{\|\mathbf{r}_0\|_2} \leq \exp\left(-2k\,\frac{\lambda_1}{\lambda_n}\right)$$

$$\rho_k^{\text{GCR}} \leq \exp\left(-2k\sqrt{\frac{\lambda_1}{\lambda_n}}\right)$$

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\mathrm{GCR}}\|_2 \leq \|\mathbf{r}_k^{\mathrm{LMR}}\|_2 \leq \|\mathbf{r}_k^{\mathrm{Richardson}}\|_2$$

Grondwatervergl., met, voor Richardson, $\alpha = \alpha^{\mathrm{opt}}$

$$\rho_k^{\mathrm{LMR}} \leq \rho_k^{\mathrm{Richardson}} \equiv \frac{\|\mathbf{r}_k^{\mathrm{Richardson}}\|_2}{\|\mathbf{r}_0\|_2} \leq \exp\left(-2k\,\frac{\lambda_1}{\lambda_n}\right)$$

$$\rho_k^{\mathrm{GCR}} \leq \exp\left(-2k\sqrt{\frac{\lambda_1}{\lambda_n}}\right)$$

Voorbeeld. Met $\lambda_1 = 10^{-4}$ en $\lambda_n = 2$ is

$\rho_k^{\mathrm{LMR}} \leq 10^{-3}$ voor $k = 7\,10^4$ en

$\rho_k^{\mathrm{GCR}} \leq 10^{-3}$ voor $k = 490$.

Convergentie

Stelling. Met dezelfde $\mathbf{x}_0$:

$$\|\mathbf{r}_k^{\text{GCR}}\|_2 \leq \|\mathbf{r}_k^{\text{LMR}}\|_2 \leq \|\mathbf{r}_k^{\text{Richardson}}\|_2$$

Grondwatervergl., met, voor Richardson, $\alpha = \alpha^{\text{opt}}$

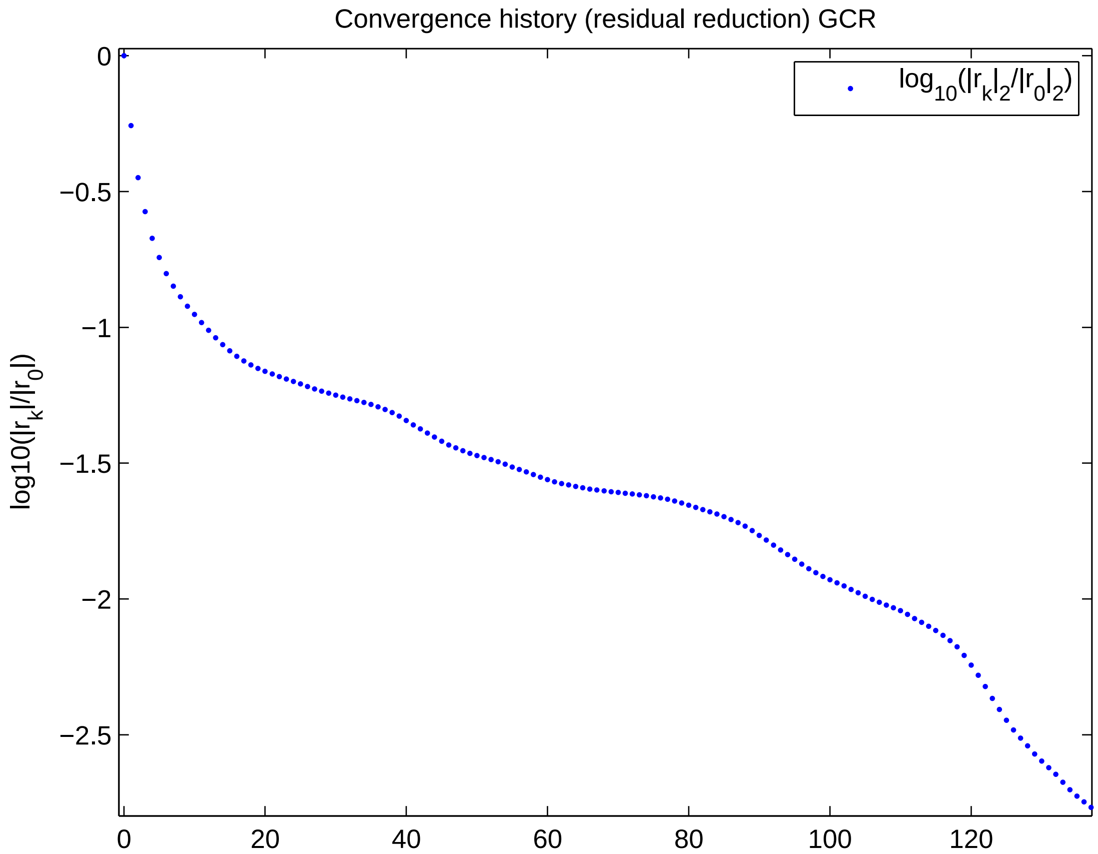
$$\rho_k^{\text{LMR}} \leq \rho_k^{\text{Richardson}} \equiv \frac{\|\mathbf{r}_k^{\text{Richardson}}\|_2}{\|\mathbf{r}_0\|_2} \leq \exp\left(-2k\,\frac{\lambda_1}{\lambda_n}\right)$$

$$\rho_k^{\text{GCR}} \leq \exp\left(-2k\sqrt{\frac{\lambda_1}{\lambda_n}}\right)$$

Grondwatervergl. Bij λ_1/λ_n moet je denken aan $h_x^2 + h_y^2$.

Opdracht. Probeer deze theoretische resultaten terug te zien in de praktijk.
Plot in je experimenten
de **residu reductie** of **convergentie historie**,
d.w.z., de grafiek van $k \rightsquigarrow \rho_k$ (op $\log_{10}$ schaal)

GCR op een 40 bij 30 rooster.



Program

- Basale Lineaire Algebra Operaties
- LU-decompositie
- Blas versus LU
- Iteratieve oplosmethoden
- Local Minimal Residuals
- Generalized Conjugate Residuals
- Convergentie
- Krylov ruimte methoden

Krylov ruimtes

De **Krylov ruimte** $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$ van orde k voorgebracht door $\mathbf{A}$ en $\mathbf{r}_0$ is

$$\begin{aligned}
\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) &= \text{Span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0) \\
&= \left\{ \sum_{j<k} \gamma_j \, \mathbf{A}^j \mathbf{r}_0 \mid \gamma_j \in \mathbb{C} \right\} \\
&= \{ P(\mathbf{A})\mathbf{r}_0 \mid P \text{ polynoom graad } < k \}
\end{aligned}$$

Krylov ruimtes

De **Krylov ruimte** $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$ van orde k voorgebracht door $\mathbf{A}$ en $\mathbf{r}_0$ is

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \;=\; \mathrm{Span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0)$$

Stelling. Richardson, LMR en GCR vinden hun benadering $\mathbf{x}_k$ in $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

Richardson, LMR en GCR zijn **Krylov ruimte methoden**.

Theorie gaat in feite over benaderen met polynomen ($P_k(\lambda)$ benadert λ^{-1} op het **spectrum** *van* $\mathbf{A}$*, d.w.z. de collectie van alle eigenwaarden van* $\mathbf{A}$*)*

Krylov ruimtes

De **Krylov ruimte** $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$ van orde k voorgebracht door $\mathbf{A}$ en $\mathbf{r}_0$ is

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \;=\; \mathrm{Span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0)$$

Stelling. Richardson, LMR en GCR vinden hun benadering $\mathbf{x}_k$ in $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

Opgave. Bewijs dat voor
zowel Richardson, als LMR en GCR geldt dat

$$\mathrm{Span}(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1}) = \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$$

$$\mathrm{Span}(\mathbf{c}_0, \ldots, \mathbf{c}_{k-1}) = \mathbf{A}\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$$

$\mathrm{Span}(\mathbf{u}_0, \ldots, \mathbf{u}_{k-1})$ is de **zoekruimte** (d.w.z., de ruimte waarin de updates voor de benaderende oplossing gezocht worden).

Krylov ruimtes

De **Krylov ruimte** $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$ van orde k voorgebracht door $\mathbf{A}$ en $\mathbf{r}_0$ is

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \; = \; \mathrm{Span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0)$$

Stelling. Richardson, LMR en GCR vinden hun benadering $\mathbf{x}_k$ in $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

GCR vindt de benadering met kleinste residu:

$$\|\mathbf{r}_k^{\mathrm{GCR}}\|_2 \leq \|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2 \text{ voor alle } \mathbf{y} \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \qquad .$$

Krylov ruimtes

De **Krylov ruimte** $\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$ van orde k voorgebracht door $\mathbf{A}$ en $\mathbf{r}_0$ is

$$\mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \;=\; \text{Span}(\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0)$$

Stelling. Richardson, LMR en GCR vinden hun benadering $\mathbf{x}_k$ in $\mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$.

GCR vindt de benadering met kleinste residu:

$$\|\mathbf{r}_k^{\text{GCR}}\|_2 \leq \|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2 \text{ voor alle } \mathbf{y} \in \mathbf{x}_0 + \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) \qquad .$$

GCR is een **minimale residu methode**.

Stagnatie en afbreken

Als in GCR $\mathbf{c}_k \perp \mathbf{r}_k$, dan
stagneert de methode, d.w.z., $\mathbf{r}_{k+1} = \mathbf{r}_k$ (waarom?)

Stagnatie en afbreken

Als in GCR $\mathbf{c}_k \perp \mathbf{r}_k$, dan
stagneert de methode, d.w.z., $\mathbf{r}_{k+1} = \mathbf{r}_k$ (waarom?)

dan is $\mathbf{c}_{k+1}$ (na orth.) $= \mathbf{0}$ (waarom?) en
breekt de methode **af**, d.w.z., er wordt gedeeld door 0 (nl., door $\sigma_{k+1} = 0$).

Stagnatie en afbreken

Als in GCR $\mathbf{c}_k \perp \mathbf{r}_k$, dan
stagneert de methode, d.w.z., $\mathbf{r}_{k+1} = \mathbf{r}_k$ (waarom?)

dan is $\mathbf{c}_{k+1}$ (na orth.) $= \mathbf{0}$ (waarom?) en
breekt de methode **af**, d.w.z., er wordt gedeeld door 0 (nl., door $\sigma_{k+1} = 0$).

Theoretisch kan stagnatie en afbreken optreden (zie opgave), praktisch gebeurt het niet.

Opgave. Voer het GCR proces (met de hand) uit voor

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ en } \mathbf{x}_0 = \mathbf{0}.$$

Opgave. Wat zijn de gevolgen van $\mathbf{c}_k \perp \mathbf{r}_k$ in LMR?

Stagnatie en afbreken

Als in GCR $\mathbf{c}_k \perp \mathbf{r}_k$, dan
stagneert de methode, d.w.z., $\mathbf{r}_{k+1} = \mathbf{r}_k$ (waarom?)

dan is $\mathbf{c}_{k+1}$ (na orth.) $= \mathbf{0}$ (waarom?) en
breekt de methode **af**, d.w.z., er wordt gedeeld door 0 (nl., door $\sigma_{k+1} = 0$).

Stelling. Als $\mathrm{Re}(\lambda) > 0$ voor alle eigenwaarden λ van $\mathbf{A}$, dan stagneert GCR niet.

Conclusies

Richardson, LMR, GCR zijn Krylov ruimten methoden (iteratief, alleen MVs, AXPYs & DOTs).

GCR is, wat betreft het aantal MVs, in zekere zin optimaal

Rich.: $+$ Geen DOTs, een paar AXPYs per MV

$-$ Langzame convergentie (in termen van MVs)

$-$ Geschikte α van te voren zelf bepalen
(kan divergeren bij "verkeerde" α)

LMR: $+$ Een paar AXPYs en DOTs/MV

$+$ Bepaalt automatisch geschikte α's

$-$ Langzame convergentie (in termen van MVs)

GCR: $-$ # AXPYs en DOTs/MV evenredig met k

$+$ Snelste convergentie (in termen van MVs)