WISB356, Utrecht, 18 september 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 ⇓ ⇓

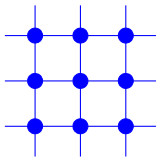
Wiskundig model

$-\nabla(K\nabla\phi) = Q$ op Ω

$-K\frac{\partial\phi}{\partial x}\cdot n = \gamma(\phi - \phi_0)$ op $\partial\Omega$

Discretiseer ⇓ ⇓ Eindige differences, . . .

Discreet model



$\ldots \quad \frac{\phi(x+\Delta x,y)-\phi(x-\Delta x,y)}{2\Delta x} \quad \ldots$

$\mathbf{Ax} = \mathbf{b}$

⇓ ⇓ Iteratieve lineaire solver

Computer model

Rekenschema voor het oplossen van $\mathbf{x}$ uit $\mathbf{Ax} = \mathbf{b}$

Implementeer ⇓ ⇓ C++

Simulatie

Simulatie

WISB356, Utrecht, 18 september 2012

Matrix-vector vergelijking voor Grondwatervergelijkingen

Gerard Sleijpen

Universiteit Utrecht
Department of Mathematics

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

Programma

- Een stelsel vergelijkingen voor roosterfunctiewaarden
- Een matrix-vector vergelijking
- De structuur van de matrix
- Matlab en ijle matrices
  - Andere ijle formaten

Programma

- Een stelsel vergelijkingen voor roosterfunctiewaarden
- Een matrix-vector vergelijking
- De structuur van de matrix
- Matlab en ijle matrices
  - Andere ijle formaten

Het stelsel vergelijkingen

Voor ieder inwendig roosterpunt $p_{ij} \equiv (x_{ij}, y_{ij})$ zijn we geïnteresseerd in de (onbekende) functiewaarde

$$\psi_{ij}$$

en beschikken we over de lineaire vergelijking

$$\alpha_{ij}^{\text{west}}\,\psi_{i-1,j} + \alpha_{ij}^{\text{cent}}\,\psi_{ij} + \alpha_{ij}^{\text{oost}}\,\psi_{i+1,j} + \alpha_{ij}^{\text{zuid}}\,\psi_{i,j-1} + \alpha_{ij}^{\text{noord}}\,\psi_{i,j+1} = f_{ij}$$

De stencil coëfficiënten α_{ij}^{xxx} zijn bekend, d.w.z, kunnen door de computer berekend worden uit de model gegevens (a, b, c, u, v, f en randvoorwaarden).

We hebben even veel vergelijkingen als onbekende (bij ieder inwendig roosterpunt een) en kunnen hopen dat dit oplosbaar is.

Het stelsel vergelijkingen

Voor ieder inwendig roosterpunt $p_{ij} \equiv (x_{ij}, y_{ij})$ zijn we geïnteresseerd in de (onbekende) functiewaarde

$$\psi_{ij}$$

en beschikken we over de lineaire vergelijking

$$\alpha_{ij}^{\text{west}}\,\psi_{i-1,j} + \alpha_{ij}^{\text{cent}}\,\psi_{ij} + \alpha_{ij}^{\text{oost}}\,\psi_{i+1,j} + \alpha_{ij}^{\text{zuid}}\,\psi_{i,j-1} + \alpha_{ij}^{\text{noord}}\,\psi_{i,j+1} = f_{ij}$$

De stencil coëfficiënten α_{ij}^{xxx} zijn bekend.

Opmerking. De vergelijking "koppelt" expliciet de functiewaarde ψ_{ij} met onder andere de functiewaarde $\psi_{i-1,j}$ middels de coëfficiënt $\alpha_{ij}^{\text{west}}$. Impliciet geeft deze vergelijking ook een koppeling tussen ψ_{ij} en bijvoorbeeld $\psi_{i-2,\text{J}}$ (voor $i = 3, \ldots, n_x$). Immers de vergelijking verandert niet door er een term als $0 \cdot \psi_{i-1,j}$ bij op te tellen.

Het stelsel vergelijkingen

Voor ieder inwendig roosterpunt $p_{ij} \equiv (x_{ij}, y_{ij})$ zijn we geïnteresseerd in de (onbekende) functiewaarde

$$\psi_{ij}$$

en beschikken we over de lineaire vergelijking

$$\alpha_{ij}^{\text{west}}\,\psi_{i-1,j} + \alpha_{ij}^{\text{cent}}\,\psi_{ij} + \alpha_{ij}^{\text{oost}}\,\psi_{i+1,j} + \alpha_{ij}^{\text{zuid}}\,\psi_{i,j-1} + \alpha_{ij}^{\text{noord}}\,\psi_{i,j+1} = f_{ij}$$

De stencil coëfficiënten α_{ij}^{xxx} zijn bekend.

Er zijn vele algoritmes om stelsels lineaire vergelijkingen numeriek op te lossen. Om gebruik te kunnen maken van deze algoritmes moeten we het stelsel in een standaard vorm schrijven.

Programma

- Een stelsel vergelijkingen voor roosterfunctiewaarden
- Een matrix-vector vergelijking
- De structuur van de matrix
- Matlab en ijle matrices
  - Andere ijle formaten

Het stelsel vergelijkingen

Voor ieder inwendig roosterpunt $p_{ij} \equiv (x_{ij}, y_{ij})$ zijn we geïnteresseerd in de (onbekende) functiewaarde

$$\psi_{ij}$$

en beschikken we over de lineaire vergelijking

$$\alpha_{ij}^{\text{west}}\,\psi_{i-1,j} + \alpha_{ij}^{\text{cent}}\,\psi_{ij} + \alpha_{ij}^{\text{oost}}\,\psi_{i+1,j} + \alpha_{ij}^{\text{zuid}}\,\psi_{i,j-1} + \alpha_{ij}^{\text{noord}}\,\psi_{i,j+1} = f_{ij}$$

De stencil coëfficiënten α_{ij}^{xxx} zijn bekend.

We schrijven het stelsel als één **matrix-vector vergelijking**

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

waarbij we voor de bekende vierkante matrix **A** en de bekende vector **b** de onbekende vector **x** moeten oplossen.

Het stelsel vergelijkingen

Voor ieder inwendig roosterpunt $p_{ij} \equiv (x_{ij}, y_{ij})$ zijn we geïnteresseerd in de (onbekende) functiewaarde

$$\psi_{ij}$$

en beschikken we over de lineaire vergelijking

$$\alpha_{ij}^{\text{west}}\,\psi_{i-1,j} + \alpha_{ij}^{\text{cent}}\,\psi_{ij} + \alpha_{ij}^{\text{oost}}\,\psi_{i+1,j} + \alpha_{ij}^{\text{zuid}}\,\psi_{i,j-1} + \alpha_{ij}^{\text{noord}}\,\psi_{i,j+1} = f_{ij}$$

De stencil coëfficiënten α_{ij}^{xxx} zijn bekend.

We schrijven het stelsel als één **matrix-vector vergelijking**

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

We moeten hiertoe identificeren iedere

- ψ_{ij} met een coördinaat x_k van **x**,
- f_{ij} met een coördinaat b_k van **b**,
- α_{ij}^{xxx} met een matrix coëfficiënt $A_{k,\ell}$ van **A**.

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij} (en daarmee de onbekende ψ_{ij} en de vergelijkingen).

We **kiezen** voor de nummering

van west naar oost en van zuid naar noord.

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Opgave. Ga na dat je hiermee
van oost naar west en dan van zuid naar noord nummert.

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Rekenschema.

$k = 1$
for $j = 1, \ldots, n_y$
  for $i = 1, \ldots, n_x$
    ...
    $b_k = f_{ij}$, $\psi_{ij} = x_k$
    $k = k + 1$
  end for
end for

In groen: van roosterfunctie naar vector.
In rood: van vector naar roosterfunctie

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Rekenschema.

$k = 1$
`for` $j = 1, \ldots, n_y$
  `for` $i = 1, \ldots, n_x$
    $\ldots$
    $b_k = f_{ij}$, $\psi_{ij} = x_k$
    $k = k + 1$
  `end for`
`end for`

Programmeer tip. Als je dit rekenschema gebruikt in MATLAB om grote arrays te 'vullen' (zoals de vector **x** of de matrix **A**) dan is het verstandig om voor dat je de loop ingaat geheugenruimte te **alloceren** (reserveren)
(bijvoorbeeld door `x=zeros(n,1);` met `n` $\equiv n_x\, n_y$).

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Rekenschema.

$k = 1$
for $j = 1, \ldots, n_y$
  for $i = 1, \ldots, n_x$
    ...
    $b_k = f_{ij}$, $\psi_{ij} = x_k$
    $k = k + 1$
  end for
end for

MATLAB tool: Matlab heeft een tool om een vector te maken van een roosterfunctie en visa versa :-) Er is geen tool om matrices te maken uit een set van stencils :-(

```
psi=reshape(x,nx,ny);
x=reshape(psi,nx*ny,1);
```

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Naar een matrix. De (i,j)-de vergelijking voor rooster-functies correspondeert met de vergelijking die de k-de rij van **A** definieert.

Zo is voor $i = 2, \ldots, n_x$ de stencilcoëfficiënt $\alpha_{ij}^{\text{west}}$ gelijk aan de matrixcoëfficiënt $A_{k,k-1}$.

Opgave. Waarom sluiten we $i = 1$ uit?

Wat is de waarde van $A_{k,k-1}$ in geval $i = 1$ (en $j > 1$)?

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Naar een matrix. De (i,j)-de vergelijking voor rooster-functies correspondeert met de vergelijking die de k-de rij van $\mathbf{A}$ definieert.

Zo is voor $i = 2, \ldots, n_x$ de stencilcoëfficiënt $\alpha_{ij}^{\text{west}}$ gelijk aan de matrixcoëfficiënt $A_{k,k-1}$.

Opgave. Met welke matrixcoëfficiënt correspondeert $\alpha_{ij}^{\text{noord}}$?

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Naar een matrix. De (i,j)-de vergelijking voor rooster-functies correspondeert met de vergelijking die de k-de rij van **A** definieert.

Zo is voor $i = 2, \ldots, n_x$ de stencilcoëfficiënt $\alpha_{ij}^{\text{west}}$ gelijk aan de matrixcoëfficiënt $A_{k,k-1}$.

De (i,j)-de vergelijking koppelt
ψ_{ij} **niet** met $\psi_{i'j'}$ als $|i-i'| > 1$ of $|j-j'| > 1$:
de corresponderende matrixcoëfficiënt $A_{k,\ell}$
(met $\ell = (j'-1)n_x + i'$) is daarom 0:
$A_{k,\ell} = 0$ als $|k-\ell| \notin \{-n_x, -1, 0, 1, n_x\}$.

Opgave. Ga dit na.

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Naar een matrix. De (i,j)-de vergelijking voor rooster-functies correspondeert met de vergelijking die de k-de rij van **A** definieert.

Zo is voor $i = 2, \ldots, n_x$ de stencilcoëfficiënt $\alpha_{ij}^{\text{west}}$ gelijk aan de matrixcoëfficiënt $A_{k,k-1}$.

De (i,j)-de vergelijking koppelt
ψ_{ij} **niet** met $\psi_{i'j'}$ als $|i-i'| > 1$ of $|j-j'| > 1$:
de corresponderende matrixcoëfficiënt $A_{k,\ell}$
(met $\ell = (j'-1)n_x + i'$) is daarom 0:
$A_{k,\ell} = 0$ als $|k-\ell| \notin \{-n_x, -1, 0, 1, n_x\}$.

Waarschuwing. Verwar niet de indices van de matrix-coëfficiënten met de indices van de koppelingscoëfficiënten: $A_{11} = \alpha_{11}^{\text{cent}}$ maar verder is er geen relatie tussen A_{ij} en α_{ij}^{xxx}. Zo is $A_{22} = \alpha_{12}^{\text{cent}}$.

Van stelsel naar matrix

Om te kunnen identificeren: nummer de rooster punten p_{ij}
Met $k \equiv (j-1)n_x + i$ is p_{ij} het k-de roosterpunt.

Naar een matrix. De (i,j)-de vergelijking voor rooster-functies correspondeert met de vergelijking die de k-de rij van **A** definieert.

Zo is voor $i = 2, \ldots, n_x$ de stencilcoëfficiënt $\alpha_{ij}^{\text{west}}$ gelijk aan de matrixcoëfficiënt $A_{k,k-1}$.

De (i,j)-de vergelijking koppelt
ψ_{ij} **niet** met $\psi_{i'j'}$ als $|i-i'| > 1$ of $|j-j'| > 1$:
de corresponderende matrixcoëfficiënt $A_{k,\ell}$
(met $\ell = (j'-1)n_x + i'$) is daarom 0:
$A_{k,\ell} = 0$ als $|k-\ell| \notin \{-n_x, -1, 0, 1, n_x\}$.

Opmerking. De matrix heeft afmeting $n \times n$,
met $n \equiv n_x n_y$ (in 2-d en in 3-d $n = n_x n_y n_z$).
In realistische situaties is n groot (10^6-10^8) en
is n^2, het aantal matrixcoëfficiënten, gigantisch (1 Tera – 10 000 Tera).

Programma

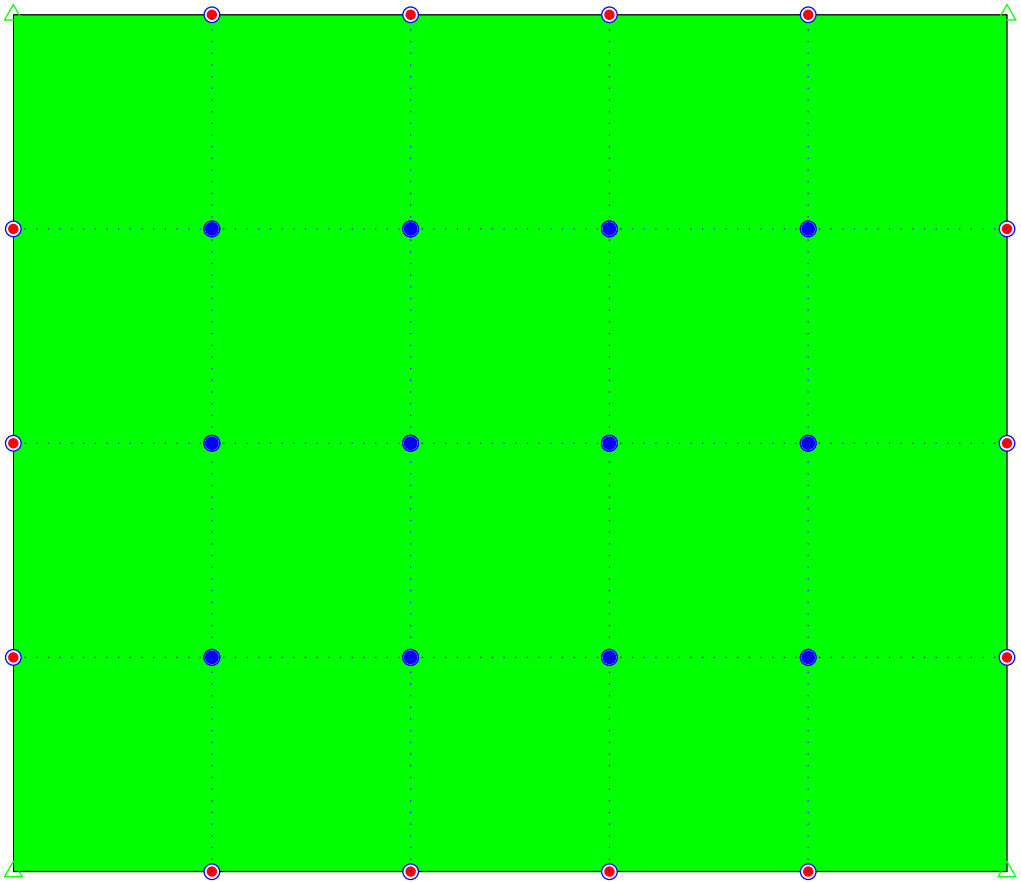
- Een stelsel vergelijkingen voor roosterfunctiewaarden
- Een matrix-vector vergelijking
- De structuur van de matrix
- Matlab en ijle matrices
  - Andere ijle formaten

De structuur van de matrix

De effectiviteit van numerieke methode hangt vaak af van de structuur van de matrix.

Sommige numerieke methoden proberen eerst de structuur te ‘verbeteren’ voordat ze aan het rekenen gaan.

De structuur van de matrix



De structuur van de matrix

$$\begin{bmatrix}
\alpha^{c}_{11} & \alpha^{o}_{11} & 0 & 0 & \alpha^{n}_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha^{w}_{12} & \alpha^{c}_{12} & \alpha^{o}_{12} & 0 & 0 & \alpha^{n}_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha^{w}_{13} & \alpha^{c}_{13} & \alpha^{o}_{13} & 0 & 0 & \alpha^{n}_{13} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \alpha^{w}_{14} & \alpha^{c}_{14} & 0 & 0 & 0 & \alpha^{n}_{14} & 0 & 0 & 0 & 0 \\
\alpha^{z}_{21} & 0 & 0 & 0 & \alpha^{c}_{21} & \alpha^{o}_{21} & 0 & 0 & \alpha^{n}_{21} & 0 & 0 & 0 \\
0 & \alpha^{z}_{22} & 0 & 0 & \alpha^{w}_{22} & \alpha^{c}_{22} & \alpha^{o}_{22} & 0 & 0 & \alpha^{n}_{22} & 0 & 0 \\
0 & 0 & \alpha^{z}_{23} & 0 & 0 & \alpha^{w}_{23} & \alpha^{c}_{23} & \alpha^{o}_{23} & 0 & 0 & \alpha^{n}_{23} & 0 \\
0 & 0 & 0 & \alpha^{z}_{24} & 0 & 0 & \alpha^{w}_{24} & \alpha^{c}_{24} & 0 & 0 & 0 & \alpha^{n}_{24} \\
0 & 0 & 0 & 0 & \alpha^{z}_{31} & 0 & 0 & 0 & \alpha^{c}_{31} & \alpha^{o}_{31} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \alpha^{z}_{32} & 0 & 0 & \alpha^{w}_{32} & \alpha^{c}_{32} & \alpha^{o}_{32} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha^{z}_{33} & 0 & 0 & \alpha^{w}_{33} & \alpha^{c}_{33} & \alpha^{o}_{33} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{z}_{34} & 0 & 0 & \alpha^{w}_{34} & \alpha^{c}_{34}
\end{bmatrix}$$

De structuur van de matrix

$$
\left[\begin{array}{cccc|cccc|cccc}
\alpha^{c}_{11} & \alpha^{o}_{11} & 0 & 0 & \alpha^{n}_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\alpha^{w}_{12} & \alpha^{c}_{12} & \alpha^{o}_{12} & 0 & 0 & \alpha^{n}_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \alpha^{w}_{13} & \alpha^{c}_{13} & \alpha^{o}_{13} & 0 & 0 & \alpha^{n}_{13} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \alpha^{w}_{14} & \alpha^{c}_{14} & 0 & 0 & 0 & \alpha^{n}_{14} & 0 & 0 & 0 & 0 \\
\hline
\alpha^{z}_{21} & 0 & 0 & 0 & \alpha^{c}_{21} & \alpha^{o}_{21} & 0 & 0 & \alpha^{n}_{21} & 0 & 0 & 0 \\
0 & \alpha^{z}_{22} & 0 & 0 & \alpha^{w}_{22} & \alpha^{c}_{22} & \alpha^{o}_{22} & 0 & 0 & \alpha^{n}_{22} & 0 & 0 \\
0 & 0 & \alpha^{z}_{23} & 0 & 0 & \alpha^{w}_{23} & \alpha^{c}_{23} & \alpha^{o}_{23} & 0 & 0 & \alpha^{n}_{23} & 0 \\
0 & 0 & 0 & \alpha^{z}_{24} & 0 & 0 & \alpha^{w}_{24} & \alpha^{c}_{24} & 0 & 0 & 0 & \alpha^{n}_{24} \\
\hline
0 & 0 & 0 & 0 & \alpha^{z}_{31} & 0 & 0 & 0 & \alpha^{c}_{31} & \alpha^{o}_{31} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \alpha^{z}_{32} & 0 & 0 & \alpha^{w}_{32} & \alpha^{c}_{32} & \alpha^{o}_{32} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha^{z}_{33} & 0 & 0 & \alpha^{w}_{33} & \alpha^{c}_{33} & \alpha^{o}_{33} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{z}_{34} & 0 & 0 & \alpha^{w}_{34} & \alpha^{c}_{34}
\end{array}\right]
$$

De structuur van de matrix

$$
\left[\begin{array}{cccc|cccc|cccc}
\alpha^{c}_{11} & \alpha^{o}_{11} & & & \alpha^{n}_{11} & & & & & & & \\
\alpha^{w}_{12} & \alpha^{c}_{12} & \alpha^{o}_{12} & & & \alpha^{n}_{12} & & & & & & \\
 & \alpha^{w}_{13} & \alpha^{c}_{13} & \alpha^{o}_{13} & & & \alpha^{n}_{13} & & & & & \\
 & & \alpha^{w}_{14} & \alpha^{c}_{14} & & & & \alpha^{n}_{14} & & & & \\
\hline
\alpha^{z}_{21} & & & & \alpha^{c}_{21} & \alpha^{o}_{21} & & & \alpha^{n}_{21} & & & \\
 & \alpha^{z}_{22} & & & \alpha^{w}_{22} & \alpha^{c}_{22} & \alpha^{o}_{22} & & & \alpha^{n}_{22} & & \\
 & & \alpha^{z}_{23} & & & \alpha^{w}_{23} & \alpha^{c}_{23} & \alpha^{o}_{23} & & & \alpha^{n}_{23} & \\
 & & & \alpha^{z}_{24} & & & \alpha^{w}_{24} & \alpha^{c}_{24} & & & & \alpha^{n}_{24} \\
\hline
 & & & & \alpha^{z}_{31} & & & & \alpha^{c}_{31} & \alpha^{o}_{31} & & \\
 & & & & & \alpha^{z}_{32} & & & \alpha^{w}_{32} & \alpha^{c}_{32} & \alpha^{o}_{32} & \\
 & & & & & & \alpha^{z}_{33} & & & \alpha^{w}_{33} & \alpha^{c}_{33} & \alpha^{o}_{33} \\
 & & & & & & & \alpha^{z}_{34} & & & \alpha^{w}_{34} & \alpha^{c}_{34}
\end{array}\right]
$$

De matrix heeft een **blok structuur**.

De blokken zijn 4×4 groot en

de matrix bestaat uit 3×3 blokken.

De structuur van de matrix

$$
\left[\begin{array}{cccc|cccc|cccc}
\alpha^{c}_{11} & \alpha^{o}_{11} & & & \alpha^{n}_{11} & & & & & & & \\
\alpha^{w}_{12} & \alpha^{c}_{12} & \alpha^{o}_{12} & & & \alpha^{n}_{12} & & & & & & \\
 & \alpha^{w}_{13} & \alpha^{c}_{13} & \alpha^{o}_{13} & & & \alpha^{n}_{13} & & & & & \\
 & & \alpha^{w}_{14} & \alpha^{c}_{14} & & & & \alpha^{n}_{14} & & & & \\
\hline
\alpha^{z}_{21} & & & & \alpha^{c}_{21} & \alpha^{o}_{21} & & & \alpha^{n}_{21} & & & \\
 & \alpha^{z}_{22} & & & \alpha^{w}_{22} & \alpha^{c}_{22} & \alpha^{o}_{22} & & & \alpha^{n}_{22} & & \\
 & & \alpha^{z}_{23} & & & \alpha^{w}_{23} & \alpha^{c}_{23} & \alpha^{o}_{23} & & & \alpha^{n}_{23} & \\
 & & & \alpha^{z}_{24} & & & \alpha^{w}_{24} & \alpha^{c}_{24} & & & & \alpha^{n}_{24} \\
\hline
 & & & & \alpha^{z}_{31} & & & & \alpha^{c}_{31} & \alpha^{o}_{31} & & \\
 & & & & & \alpha^{z}_{32} & & & \alpha^{w}_{32} & \alpha^{c}_{32} & \alpha^{o}_{32} & \\
 & & & & & & \alpha^{z}_{33} & & & \alpha^{w}_{33} & \alpha^{c}_{33} & \alpha^{o}_{33} \\
 & & & & & & & \alpha^{z}_{34} & & & \alpha^{w}_{34} & \alpha^{c}_{34}
\end{array}\right]
$$

De matrix heeft een **blok structuur**.

De blokken zijn $n_x \times n_x$ groot en

de matrix bestaat uit $n_y \times n_y$ blokken.

De structuur van de matrix

$$
\begin{bmatrix}
C_1 & D_1^{\text{noord}} & 0 & & & & & \\
D_2^{\text{zuid}} & C_2 & D_2^{\text{noord}} & 0 & & & & \\
0 & D_3^{\text{zuid}} & C_3 & D_3^{\text{noord}} & 0 & & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & 0 & D_j^{\text{zuid}} & C_j & D_j^{\text{noord}} & 0 & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots \\
 & & & & 0 & D_{n_y-1}^{\text{zuid}} & C_{n_y-1} & D_{n_y-1}^{\text{noord}} \\
 & & & & & 0 & D_{n_y}^{\text{noord}} & C_{n_y}
\end{bmatrix}
$$

De matrix is een $n_y \times n_y$ **blok matrix** met blokken van afmeting $n_x \times n_x$. Deze matrix is **blok tri-diagonaal**.

De diagonaal blokken C_j zijn tri-diagonale matrices.
De co-diagonaal blokken D_j^{zuid} en D_j^{noord} zijn diagonale matrices.

De structuur van de matrix

$$
\begin{bmatrix}
C_1 & D_1^{\text{noord}} & 0 & & & & & \\
D_2^{\text{zuid}} & C_2 & D_2^{\text{noord}} & 0 & & & & \\
0 & D_3^{\text{zuid}} & C_3 & D_3^{\text{noord}} & 0 & & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & 0 & D_j^{\text{zuid}} & C_j & D_j^{\text{noord}} & 0 & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots \\
 & & & & 0 & D_{n_y-1}^{\text{zuid}} & C_{n_y-1} & D_{n_y-1}^{\text{noord}} \\
 & & & & & 0 & D_{n_y}^{\text{noord}} & C_{n_y}
\end{bmatrix}
$$

De matrix is een $n_y \times n_y$ **blok matrix** met blokken van afmeting $n_x \times n_x$. Deze matrix is **blok tri-diagonaal**.

Het blok C_j representeert de koppelingen in de west-oost richting op de j-de west-oost roosterlijn.
Het blok D_j^{zuid} representeert de koppeling tussen de j-de west-oost roosterlijn en de lijn ten zuiden daarvan.

De structuur van de matrix

$$
\begin{bmatrix}
C_1 & D_1^{\text{noord}} & 0 & & & & & \\
D_2^{\text{zuid}} & C_2 & D_2^{\text{noord}} & 0 & & & & \\
0 & D_3^{\text{zuid}} & C_3 & D_3^{\text{noord}} & 0 & & & \\
 & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & 0 & D_j^{\text{zuid}} & C_j & D_j^{\text{noord}} & 0 & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots \\
 & & & & 0 & D_{n_y-1}^{\text{zuid}} & C_{n_y-1} & D_{n_y-1}^{\text{noord}} \\
 & & & & & 0 & D_{n_y}^{\text{noord}} & C_{n_y}
\end{bmatrix}
$$

De matrix is een $n_y \times n_y$ **blok matrix** met blokken van afmeting $n_x \times n_x$. Deze matrix is **blok tri-diagonaal**.

Een matrix van een 3 dimensionaal model is ook blok tri-diagonaal. Het is een $n_z \times n_z$ blok matrix met blokken van $n_x n_y \times n_x n_y$. De co-diagonaal blokken koppelen horizontale vlakken, de diagonale blokken representeren de een 2-d model in een horizontaal vlak.

De structuur van de matrix

$$
\left[\begin{array}{cccc|cccc|cccc}
\alpha^{c}_{11} & \alpha^{o}_{11} & & & \alpha^{n}_{11} & & & & & & & \\
\alpha^{w}_{12} & \alpha^{c}_{12} & \alpha^{o}_{12} & & & \alpha^{n}_{12} & & & & & & \\
 & \alpha^{w}_{13} & \alpha^{c}_{13} & \alpha^{o}_{13} & & & \alpha^{n}_{13} & & & & & \\
 & & \alpha^{w}_{14} & \alpha^{c}_{14} & & & & \alpha^{n}_{14} & & & & \\
\hline
\alpha^{z}_{21} & & & & \alpha^{c}_{21} & \alpha^{o}_{21} & & & \alpha^{n}_{21} & & & \\
 & \alpha^{z}_{22} & & & \alpha^{w}_{22} & \alpha^{c}_{22} & \alpha^{o}_{22} & & & \alpha^{n}_{22} & & \\
 & & \alpha^{z}_{23} & & & \alpha^{w}_{23} & \alpha^{c}_{23} & \alpha^{o}_{23} & & & \alpha^{n}_{23} & \\
 & & & \alpha^{z}_{24} & & & \alpha^{w}_{24} & \alpha^{c}_{24} & & & & \alpha^{n}_{24} \\
\hline
 & & & & \alpha^{z}_{31} & & & & \alpha^{c}_{31} & \alpha^{o}_{31} & & \\
 & & & & & \alpha^{z}_{32} & & & \alpha^{w}_{32} & \alpha^{c}_{32} & \alpha^{o}_{32} & \\
 & & & & & & \alpha^{z}_{33} & & & \alpha^{w}_{33} & \alpha^{c}_{33} & \alpha^{o}_{33} \\
 & & & & & & & \alpha^{z}_{34} & & & \alpha^{w}_{34} & \alpha^{c}_{34}
\end{array}\right]
$$

De matrix heeft 5 diagonalen; een **penta-diagonaal** matrix.
De 1-d variant heeft 3 diagonalen; een **tri-diagonaal** matrix.
De 3-d variant heeft 7 diagonalen.

De structuur van de matrix

$$
\left[\begin{array}{cccc|cccc|cccc}
\alpha^{c}_{11} & \alpha^{o}_{11} & & & \alpha^{n}_{11} & & & & & & & \\
\alpha^{w}_{12} & \alpha^{c}_{12} & \alpha^{o}_{12} & & & \alpha^{n}_{12} & & & & & & \\
 & \alpha^{w}_{13} & \alpha^{c}_{13} & \alpha^{o}_{13} & & & \alpha^{n}_{13} & & & & & \\
 & & \alpha^{w}_{14} & \alpha^{c}_{14} & & & & \alpha^{n}_{14} & & & & \\
\hline
\alpha^{z}_{21} & & & & \alpha^{c}_{21} & \alpha^{o}_{21} & & & \alpha^{n}_{21} & & & \\
 & \alpha^{z}_{22} & & & \alpha^{w}_{22} & \alpha^{c}_{22} & \alpha^{o}_{22} & & & \alpha^{n}_{22} & & \\
 & & \alpha^{z}_{23} & & & \alpha^{w}_{23} & \alpha^{c}_{23} & \alpha^{o}_{23} & & & \alpha^{n}_{23} & \\
 & & & \alpha^{z}_{24} & & & \alpha^{w}_{24} & \alpha^{c}_{24} & & & & \alpha^{n}_{24} \\
\hline
 & & & & \alpha^{z}_{31} & & & & \alpha^{c}_{31} & \alpha^{o}_{31} & & \\
 & & & & & \alpha^{z}_{32} & & & \alpha^{w}_{32} & \alpha^{c}_{32} & \alpha^{o}_{32} & \\
 & & & & & & \alpha^{z}_{33} & & & \alpha^{w}_{33} & \alpha^{c}_{33} & \alpha^{o}_{33} \\
 & & & & & & & \alpha^{z}_{34} & & & \alpha^{w}_{34} & \alpha^{c}_{34}
\end{array}\right]
$$

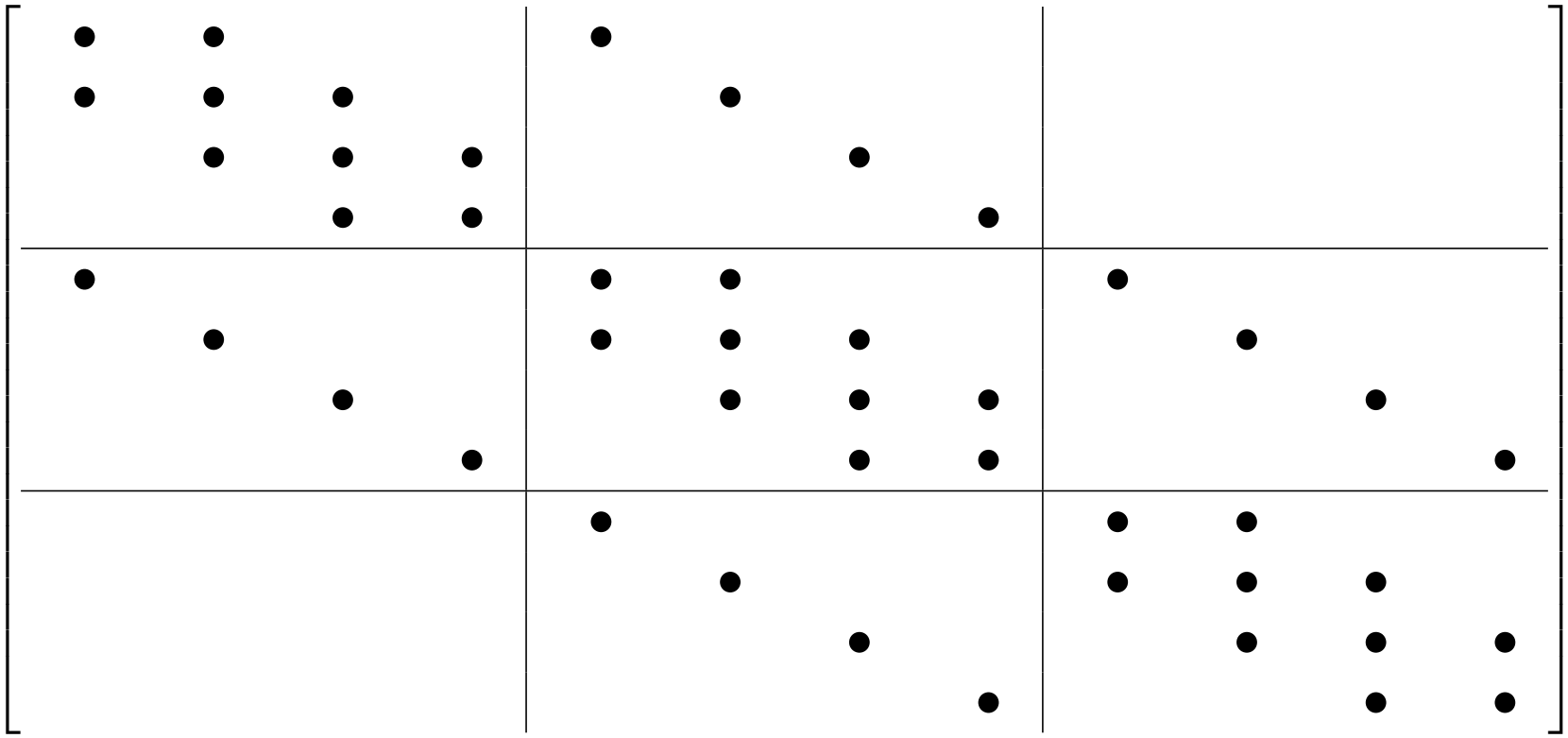
De matrix heeft **bandbreedte** n_x, dat wil zeggen

$$n_k = \min\{k \mid A_{ij} = 0 \text{ alle } i,j \text{ waarvoor } |i-j| > k\}.$$

De coëfficiënten A_{ij} met $|i-j| \leq n_x$ vormen de **band** van de matrix.

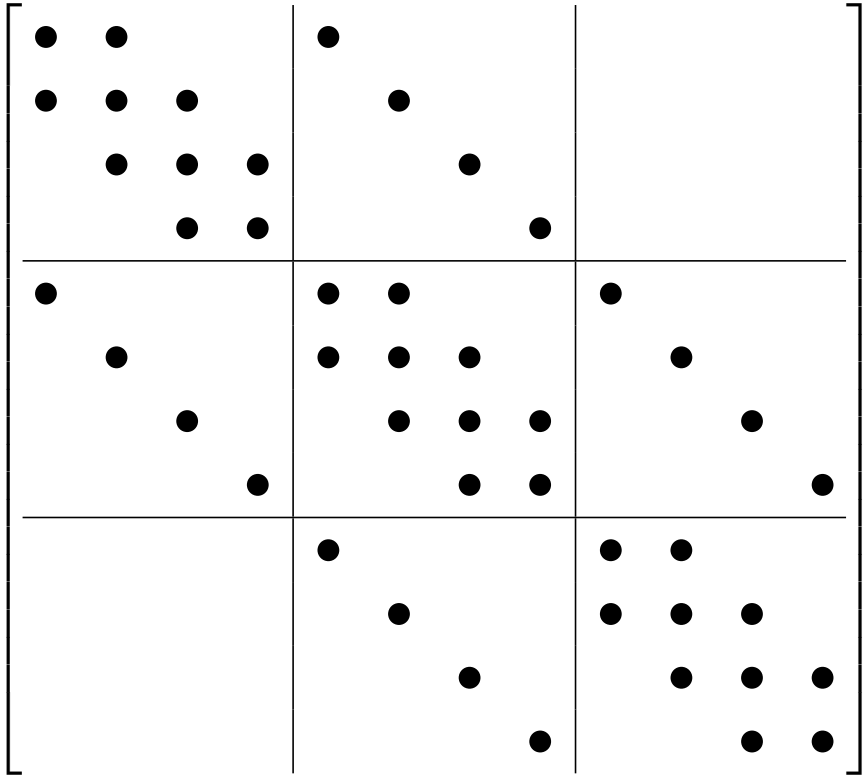
De band is gewoonlijk smal: $n_x \ll n \equiv n_x n_y$

De structuur van de matrix



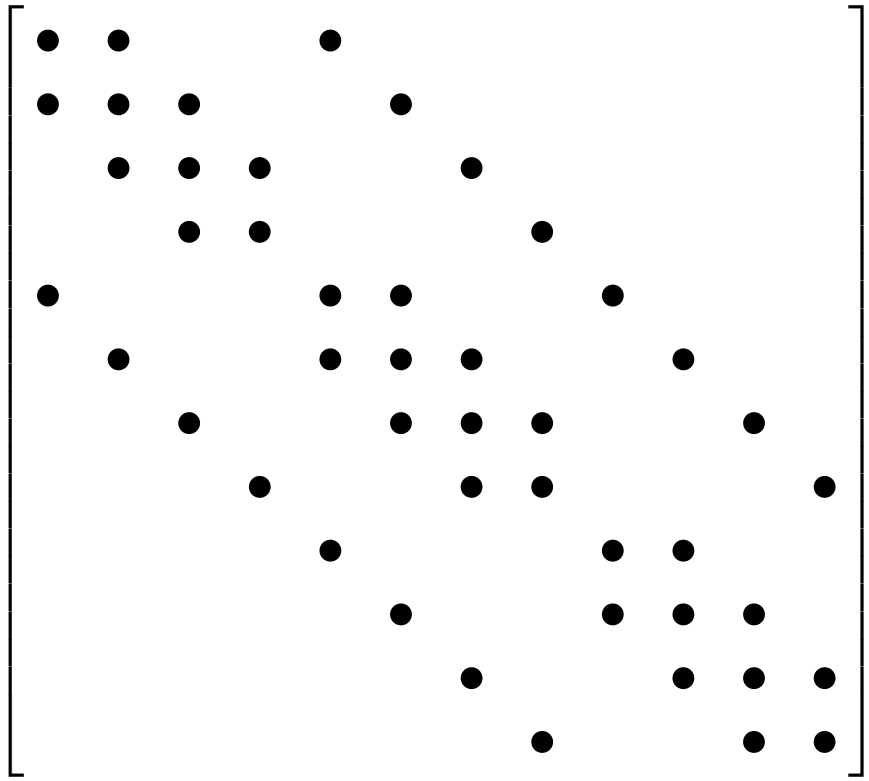
De ‘bullets’ geven aan
waar in de matrix de coëfficiënten $\neq 0$ staan.
De matrix is **ijl**: slechts een paar niet-nullen in iedere rij.

De structuur van de matrix



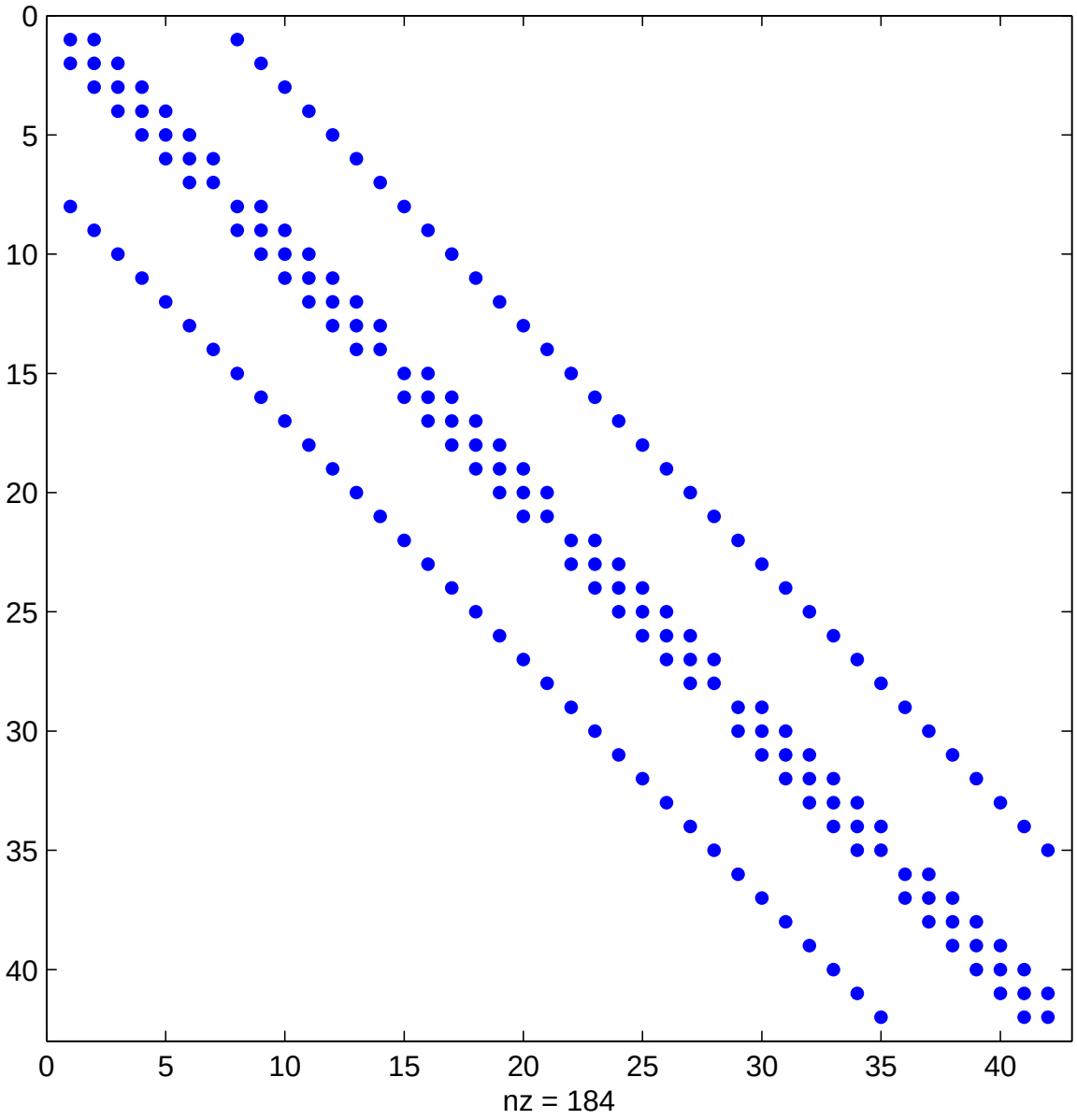
De verzameling $\{(k,\ell) \mid A_{k,\ell} \neq 0\}$
is het **ijlheidspatroon** van **A**.
Het plaatje representeert dus het ijlheidspatroon.

De structuur van de matrix



Matlab tool. `spy(A);`

De structuur van de matrix



De structuur van de matrix

De effectiviteit van numerieke methode hangt vaak af van de structuur van de matrix.

Sommige numerieke methoden proberen eerst de structuur te 'verbeteren' voordat ze aan het rekenen gaan.

In ons geval hangt de structuur niet alleen af van het model en van de discretisatie, maar ook van de gekozen nummering.

Opgave. Beschouw de **zwart-wit** nummering (red-black ordering), dat wil zeggen, 'kleur' de roosterpunten beurtelings zwart en wit, zoals de velden van een schaakbord,: $p_{1,1}$ is wit; witte punten hebben alleen zwarte buren, zwarte buren hebben alleen witte buren. Nummer nu eerst de witte punten (van west naar oost en van zuid naar noord) en nummer daarna de zwarte.

Analyseer de structuur van de resulterende matrix.

Programma

- Een stelsel vergelijkingen voor roosterfunctiewaarden
- Een matrix-vector vergelijking
- De structuur van de matrix
- Matlab en ijle matrices
  - Andere ijle formaten

Hoe sla je een ijle matrix op?

$n_2 \equiv n \equiv n_x n_y$ in 2d en $n_3 \equiv n \equiv n_x n_y n_z$ in 3d.

In de getallenvoorbeelden hier is $n_y = n_x = 1000$ en $n_z = 20$

Standaard sla je en matrix in de computer (in Matlab) op als een $n \times n$ array.

In ons geval is n groot ($n_2 = 10^6$, $n_3 = 20\,10^6$) en past zo'n array van 8-bytes getallen (n^2 getallen van 8 bytes elk) niet in het (cash) geheugen (8Tb resp., 3 200Tb).

Hoe sla je een ijle matrix op?

$n_2 \equiv n \equiv n_x n_y$ in 2d en $n_3 \equiv n \equiv n_x n_y n_z$ in 3d.

In de getallenvoorbeelden hier is $n_y = n_x = 1000$ en $n_z = 20$

Standaard sla je en matrix in de computer (in Matlab) op als een $n \times n$ array.

In ons geval is n groot ($n_2 = 10^6$, $n_3 = 20\,10^6$) en past zo'n array van 8-bytes getallen (n^2 getallen van 8 bytes elk) niet in het (cash) geheugen (8Tb resp., 3 200Tb).

Compressed storage format. Gelukkig is **A** ijl. We hoeven alleen de coëfficiënten $A_{k\ell} \neq 0$ op te slaan en de locatie (k, ℓ). Er zijn verschillende manieren om dat te doen. We kiezen voor het formaat van Matlab.

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

Zorg ervoor dat iedere i, j, A_{ij} met $A_{ij} \neq 0$ in de lijst voorkomt.

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

Matlab tool. Met het Matlab commando

```
A=sparse(I,J,V);
```

definieer je uit `I,J,V` een ijle matrix `A`.

Verdere operaties programmeer je als voor volle matrices, zo voert `b=A*x` de matrix-vector vermenigvuldiging $\mathbf{b} = \mathbf{Ax}$ uit.

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

Matlab tool. Met het Matlab commando

```
A=sparse(I,J,V);
```

definieer je uit `I,J,V` een ijle matrix `A`.

Opmerking. `V` mag nullen bevatten: Matlab verwijdert die bij het uitvoeren van `sparse`.

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

Matlab tool. Met het Matlab commando

```
[I,J,V]=sparse(A);
```

definieer je uit de ijle matrix `A` de lijst van niet-nul coëfficiënten met bijbehorende locatie indices.

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

De vectoren I en J bestaan uit natuurlijke getallen. Dat gebruikt matlab niet. Matlab slaat dus 3 maal 5 n getallen op (van 8 byte elk) (in 3-d, 3 maal 7 n getallen). Hoewel het feit dat I en J uit integers bestaat niet wordt uitgebuit is dit nog altijd aanzienlijk minder dan n^2 getallen.

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

Voordeel geheugenbeslag.

$15\,n_2$ versus n_2^2 (120Mb versus 8Tb)

$21\,n_3$ versus n_3^2 (3.36 Gb versus 3 200 Tb).

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

Voordeel rekenkosten.

Vermenigvuldigen met 0 levert 0. Dat hoef je niet door de computer te laten uitrekenen.

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

Voordeel rekenkosten.

1 **flop** is een floating point operatie ($+$, $-$, $*$, $/$).

Kosten **Ax**

- voor een volle matrix: $\approx 2n^2$ flop
- voor onze ijle matrix; $9n_2$ flop, $13n_3$ flop.

$9n_2$ versus $2n_2^2$ is 1 sec versus 2.2 maanden.

Matlab's compressed storage format

Er zijn `nnz` coëfficiënten $A_{i,j} \neq 0$ (number of non zeros):

$$\texttt{nnz} \equiv \#\{(i,j) \mid A_{i,j} \neq 0\}.$$

In Matlab kan je een ijle matrix (**sparse matrix**) beschrijven met drie kolom vectoren `I,J,V` van gelijke lengte `nnz`, waarbij met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`:

- `I` is de vector van rij indices,
- `J` is de vector van kolom indices en
- `V` is de vector van de waardes van de bijbehorende matrixcoëfficiënt.

Opdracht. Schrijf een Matlab functie routine

```
[A,b]=vectoriseer(alphac,alphaw,alphao,alphaz,alphan,f);
```

die uit de stencil coëfficiënten α_{ij}^{xxx} en f_{ij} de **ijle** matrix **A** en de rechterlid vector **b** berekent.

`I,J,V`. Met $i =$ `I(k)`, $j =$ `J(k)` is $A_{ij} =$ `V(k)`.

Matlab's `b=A*x`. Achter de schermen runt matlab

```
b=zeros(n,1);
for k=1:nnnz;
   b(I(k))=b(I(k))+V(k)*x(I(k));
end
```

De performance kan verbeteren door de getallen in de drie vectoren `I,J,V` anders te nummeren: vindt een **permutatie** `P` (een bijectie van $N \equiv \{1, 2, \ldots,$ `nnz`$\}$ naar N) en werk met `I(P),J(P),V(P)`.

Verbetering kan komen door een beter gebruik van het cache geheugen. Dit zou je kunnen bereiken door de te hernummeren zodat de rij indices (of kolom indices) in stijgende grootte verschijnen.

Programma

- Een stelsel vergelijkingen voor roosterfunctiewaarden
- Een matrix-vector vergelijking
- De structuur van de matrix
- Matlab en ijle matrices
- Andere ijle formaten

Andere compressed strorage formats

In 2d is onze matrix een penta-diagonaal matrix. Vijf vectoren opslaan van diagonaal coëfficiënten plus de bijbehorende locatie van de diagonalen is ook een efficiënte manier om onze matrix te representeren.

Matlab tool.

```
A=spdiags([d1,d2,d3,d4,d5],[k1,k2,k3,k4,k5],n.n);
```

genereert een ijle `n` bij `n` matrix `A` met op de `k1`-de co-diagonaal de coëfficiënten uit de kolomvector `d1`, etc. Hierbij is `k=[k1,k2,k3,k4,k5]` een rij van 5 gehele getallen, `d1`, . . . `d5` zijn kolomvectoren van lengte n. De hoofddiagonaal $\{A_{ii}\}$ van de matrix is de 0-de diagonaal, $\{A_{ij} \mid j - i = k\}$ is de k-de co-diagonaal.

Andere compressed strorage formats

In 2d is onze matrix een penta-diagonaal matrix. Vijf vectoren opslaan van diagonaal coëfficiënten plus de bijbehorende locatie van de diagonalen is ook een efficiënte manier om onze matrix te representeren.

Waarschuwing. Als je Matlab's `spdiags` gebruikt om een ijle matrix `A` te definiëren voor gebruik in Matlab dan moet je er rekening mee houden dat de twee co-diagionalen van **A** (dus die met `k2=-1` en `k4=1`) ook nullen bevatten.

Compressed row storage

Van matrix A $= (A_{ij})$ naar compressed row storage `Jc,Vc`.

```
k=n+2;
for i=1:n
    Jc(i)=k; Vc(i)=A_ii;
    for j=1:n
       if A_ij ≠ 0, Jc(k)=j; Vc(k)=A_ij; k=k+1; end
    end
end
Jc(n+1)=k; Vc(n+1)=0;
```

Twee kolom vectoren `Jc,Vc`, beide van lengte `nnz+1`.
`Jc` bevat natuurlijke getallen, `Vc` bevat de waarde van de niet-nul coëfficiënten van **A** (behalve `Vc(n+1)`).

Compressed row storage

Van matrix A $= (A_{ij})$ naar compressed row storage `Jc,Vc`.

```
k=n+2;
for i=1:n
    Jc(i)=k; Vc(i)=A_ii;
    for j=1:n
       if A_ij ≠ 0, Jc(k)=j; Vc(k)=A_ij; k=k+1; end
    end
end
Jc(n+1)=k; Vc(n+1)=0;
```

De eerste n elementen van `Vc` bevatten de diagonaal coëfficiënten van **A** (ook als sommige ervan de waarde 0 hebben).
De elementen vanaf `n+2` in `Vc` bevatten de nut-nullen van **A** buiten diagonaal elementen, genummerd per rij (de hele rij doorlopen, dan naar de volgende rij).

Compressed row storage

Van matrix A $= (A_{ij})$ naar compressed row storage `Jc,Vc`.

```
k=n+2;
for i=1:n
    Jc(i)=k; Vc(i)=A_ii;
    for j=1:n
       if A_ij ≠ 0, Jc(k)=j; Vc(k)=A_ij; k=k+1; end
    end
end
Jc(n+1)=k; Vc(n+1)=0;
```

`Jc` bevat de rij en kolom index informatie.
Voor `k>n+1` is `Jc(k)` de kolom index (Jc(k)$=j$ als `Vc(k)`$=A_{ij}$).
Voor `i<n+1` vertelt `Jc(i)` waar de i-de rij begint (en dus waar de $i-1$-ste eindigt):
de i-de rij is te vinden voor `k=Jc(i):Jc(i+1)-1`.

Compressed row storage

`b=A*x` in CRS

```
n=Jc(1)-1; b=zeros(n,1);
for i=1:n
   b(i)=Vc(i)*x(i);
   for k=Jc(i):Jc(i+1)-1
      j=Jc(k); Aij=Vc(k); b(i)=b(i)+Aij*x(j);
   end
end
```

Voordeel. De waarden `Aij` liggen achter elkaar in het geheugen (voordeel in het gebruik van het cache geheugen).

Nadeel. Bovenstaand voordeel gaat niet op voor `b=A'*x` (vermenigvuldigen met A' $= \mathbf{A}^* \equiv (\overline{\mathbf{A}})^\mathsf{T}$).

Compressed row storage

Van CRS naar Matlab's storage. `Jc,Vc`$\rightsquigarrow$`Im,Jm,Vm`

```
n=Jc(1)-1; nnz=Jc(n)-1;
Im=zeros(nnz,1); Jm=Im; Vm=Im;
for i=1:n
   Im(i)=i; Jm(i)=i; Vm(i)=Vc(i);
   for k=Jc(i):Jc(i+1)-1
      j=Jc(k); Aij=Vc(k);
      Im(k-1)=i; Jm(k-1)=j; Vm(k-1)=Aij;
   end
end
```

Omgekeerd is lastiger:
de informatie in `Im,Jm,Vm` is niet gesorteerd per rij.

Compressed row storage

Matlab's storage versus CRS.

CRS. Minimaal geheugen beslag
snelle matrix-vector vermenigvuldiging

Matlab. Geen verschil tussen $\mathbf{A}$ en $\mathbf{A}^*$