

Een krachtige waterkwaliteitssimulator

Een snelle en robuuste eindige volume methode voor de advectie-diffusie vergelijking.

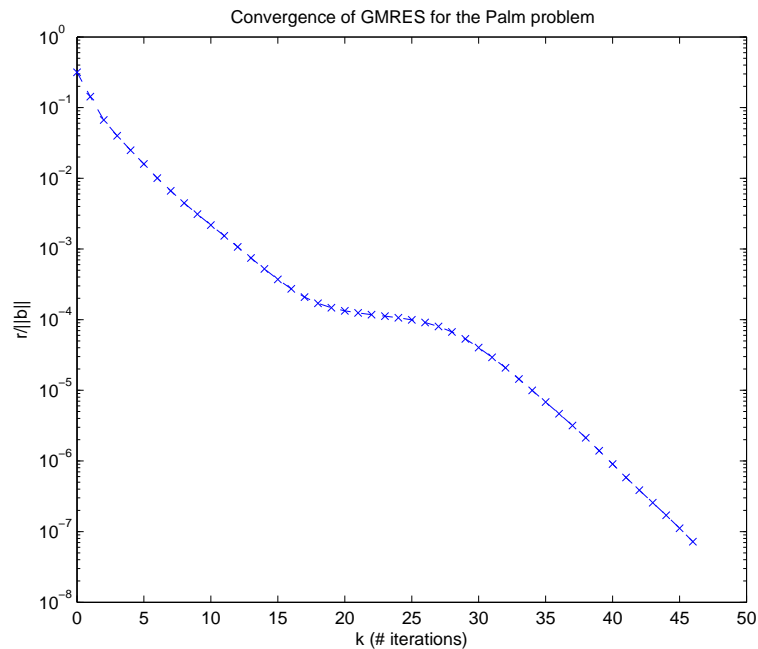


Paulien van Slingerland

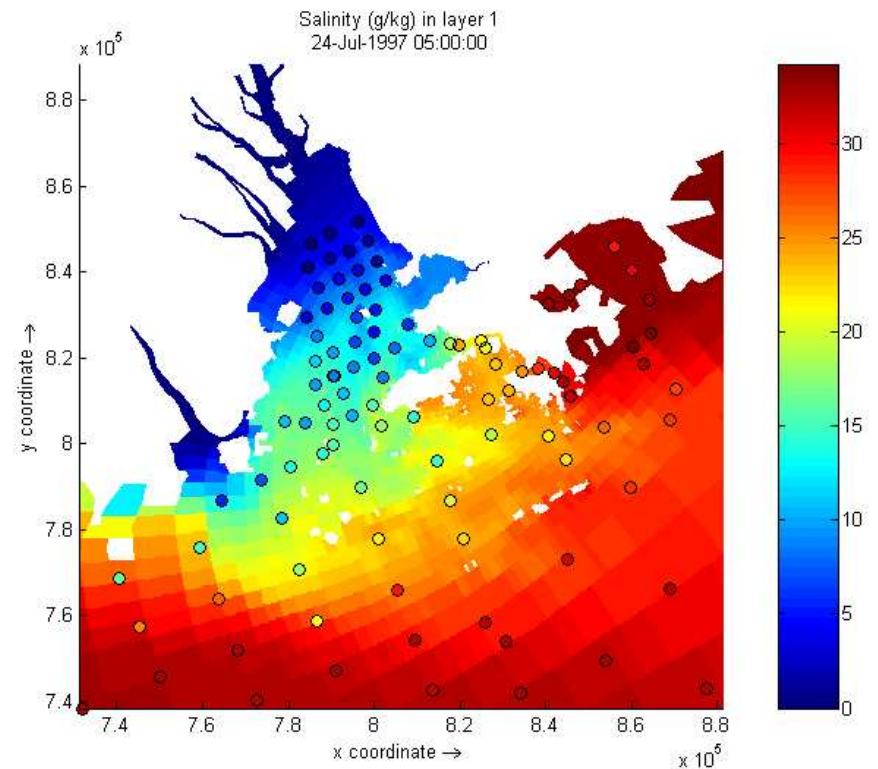
7 december 2006

Na een korte introductie over Delft3D-WAQ komen **twee hoofdproblemen** aan bod.

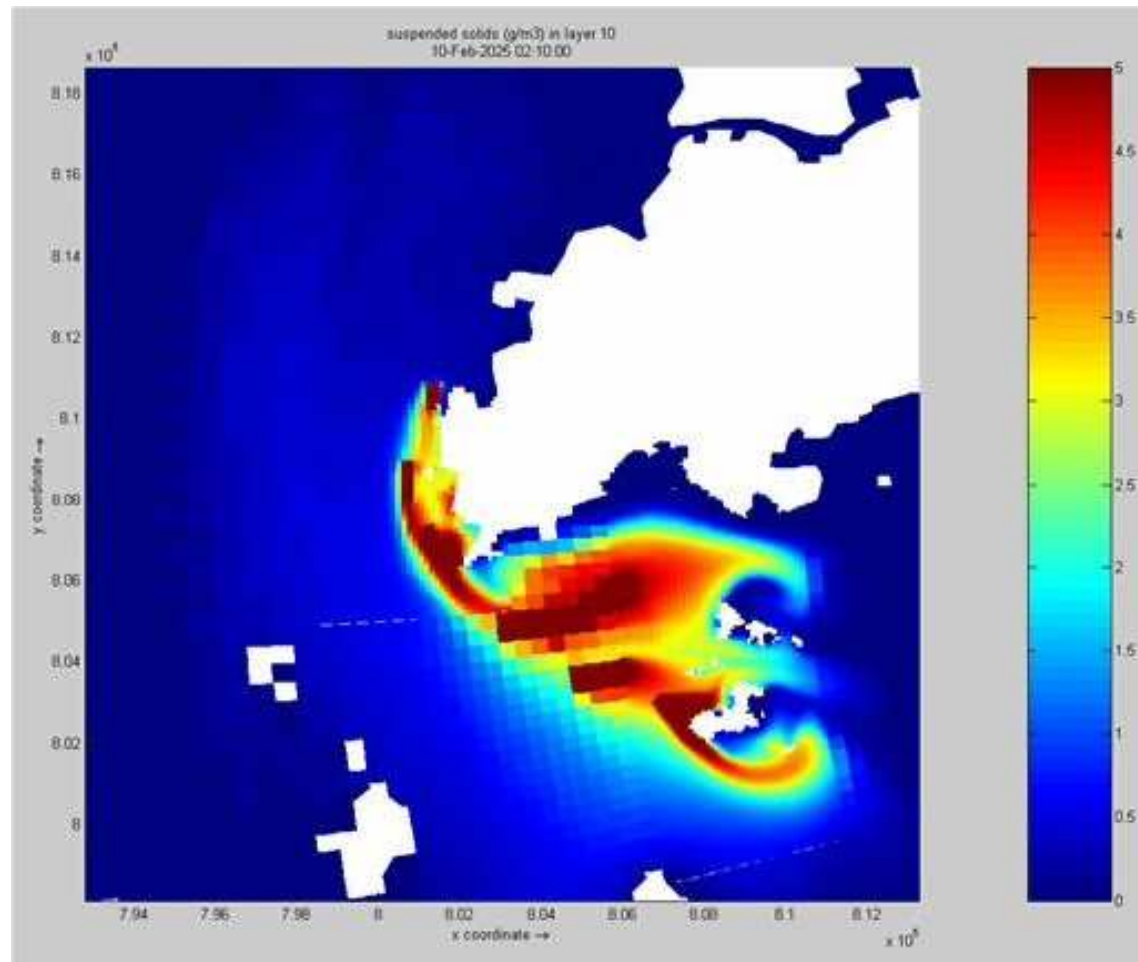
I: Trage convergentie van GMRES



II: Numerieke diffusie of strenge stabiliteitseis

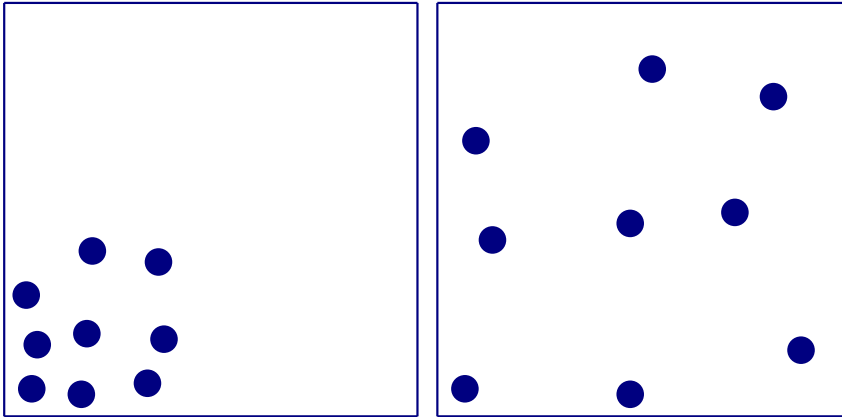


Een **toepassing** van WAQ is een simulatie van wolken slibdeeltjes in de zee bij Hong Kong.

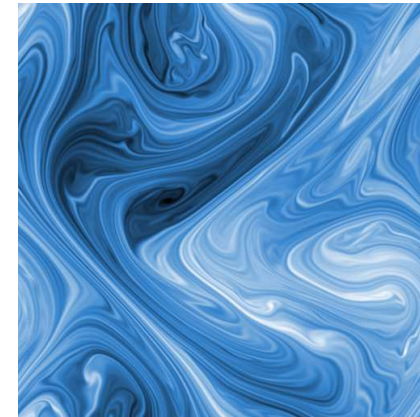


Waterkwaliteit wordt beïnvloed door **transport** en waterkwaliteitsprocessen.

Moleculaire diffusie



Turbulente diffusie



Advectie

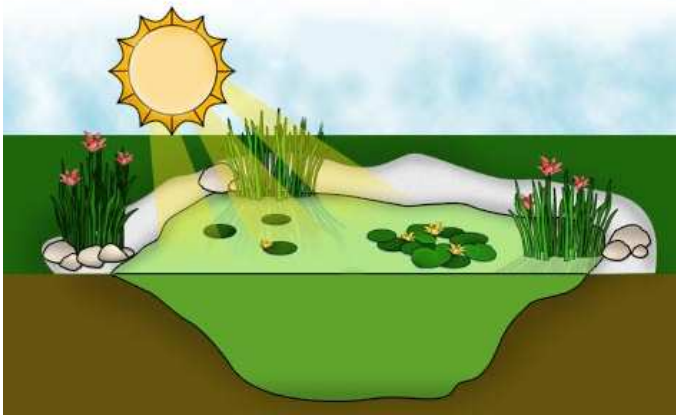


Overig transport

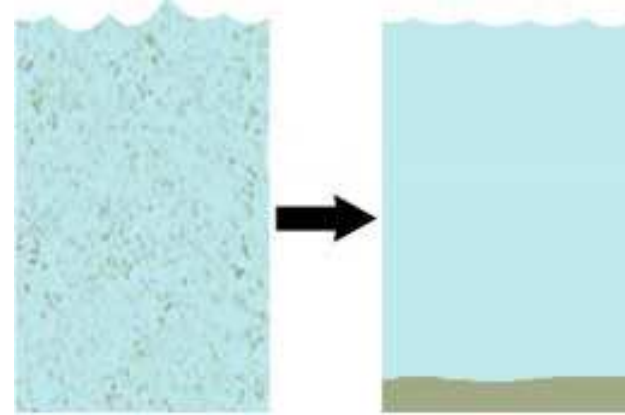


Waterkwaliteit wordt beïnvloed door transport en **waterkwaliteitsprocessen.**

Fotosynthese



Sedimentatie



Sterfte van bacteriën



Algengroei



Waterkwaliteit kan gemodelleerd worden aan de hand van de **advectie-diffusie vergelijking**:

$$\underbrace{\frac{\partial c}{\partial t}}_{\text{verandering}} = \underbrace{-\nabla \cdot (\underline{\mathbf{u}}c - d\nabla c)}_{\text{transport}} + \underbrace{p}_{\text{processen}}$$

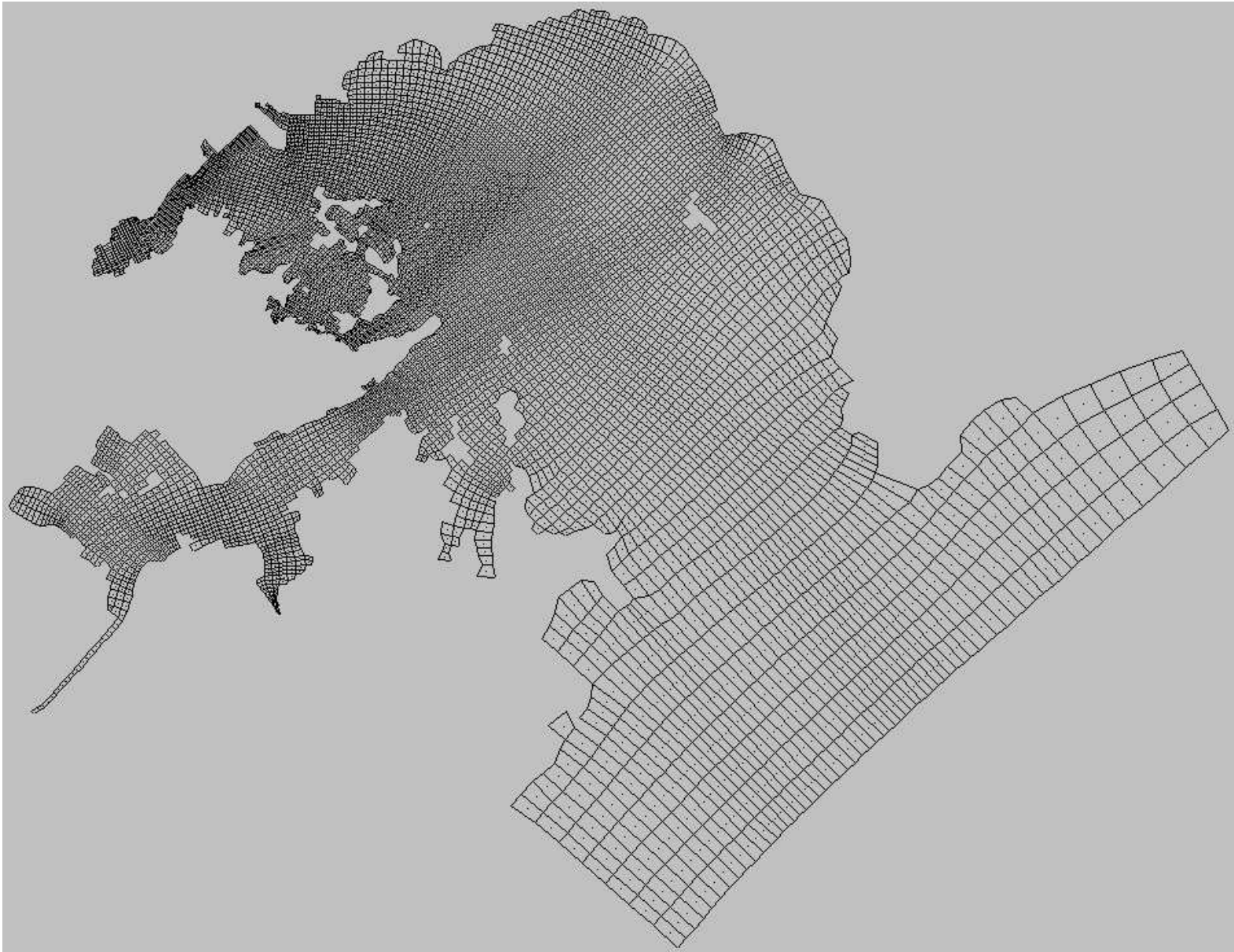
c concentratie

$\underline{\mathbf{u}}$ snelheid

d diffusiecoefficient

p waterkwaliteitsprocessen

Het **grid** van WAQ volgt uit het samennemen van gridcellen van het gestructureerde grid van **FLOW**.

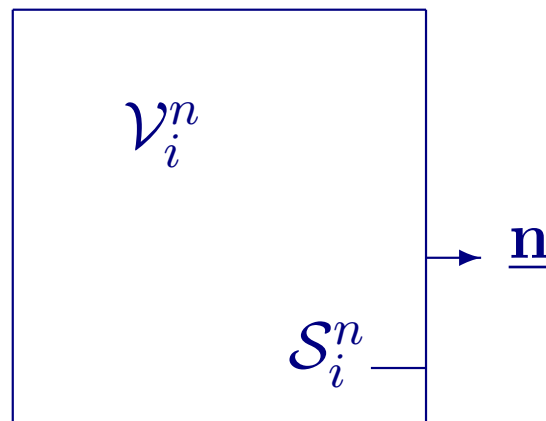


Het **grid** van **WAQ** volgt uit het samennemen van gridcellen van het gestructureerde grid van FLOW.



De eindige volume methode volgt uit het benaderen van de **integraalvorm** voor elke gridcel.

$$\frac{\partial c}{\partial t} = -\nabla \cdot (\underline{\mathbf{u}}c - d\nabla c) + p$$
$$\frac{d}{dt} \int_{\mathcal{V}_i} c d\underline{\mathbf{x}} = - \int_{\mathcal{S}_i} (\underline{\mathbf{u}}c - d\nabla c) \cdot \underline{\mathbf{n}} d\underline{\mathbf{x}} + \int_{\mathcal{V}_i} p d\underline{\mathbf{x}}$$



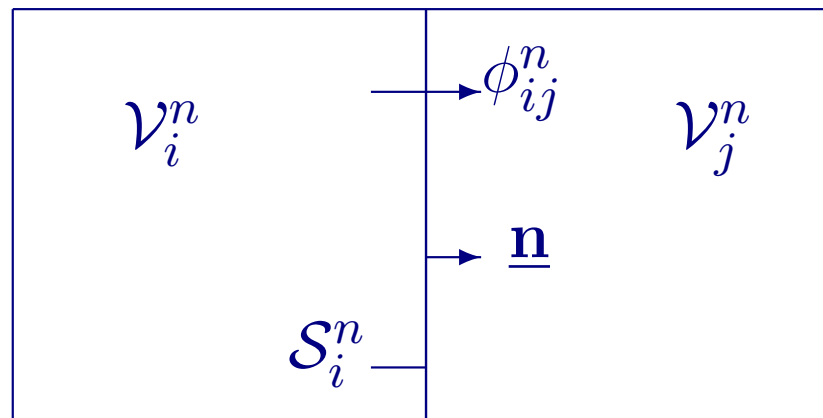
De **eindige volume methode** volgt uit het benaderen van de integraalvorm voor elke gridcel.

$$\frac{\partial c}{\partial t} = -\nabla \cdot (\underline{\mathbf{u}}c - d\nabla c) + p$$

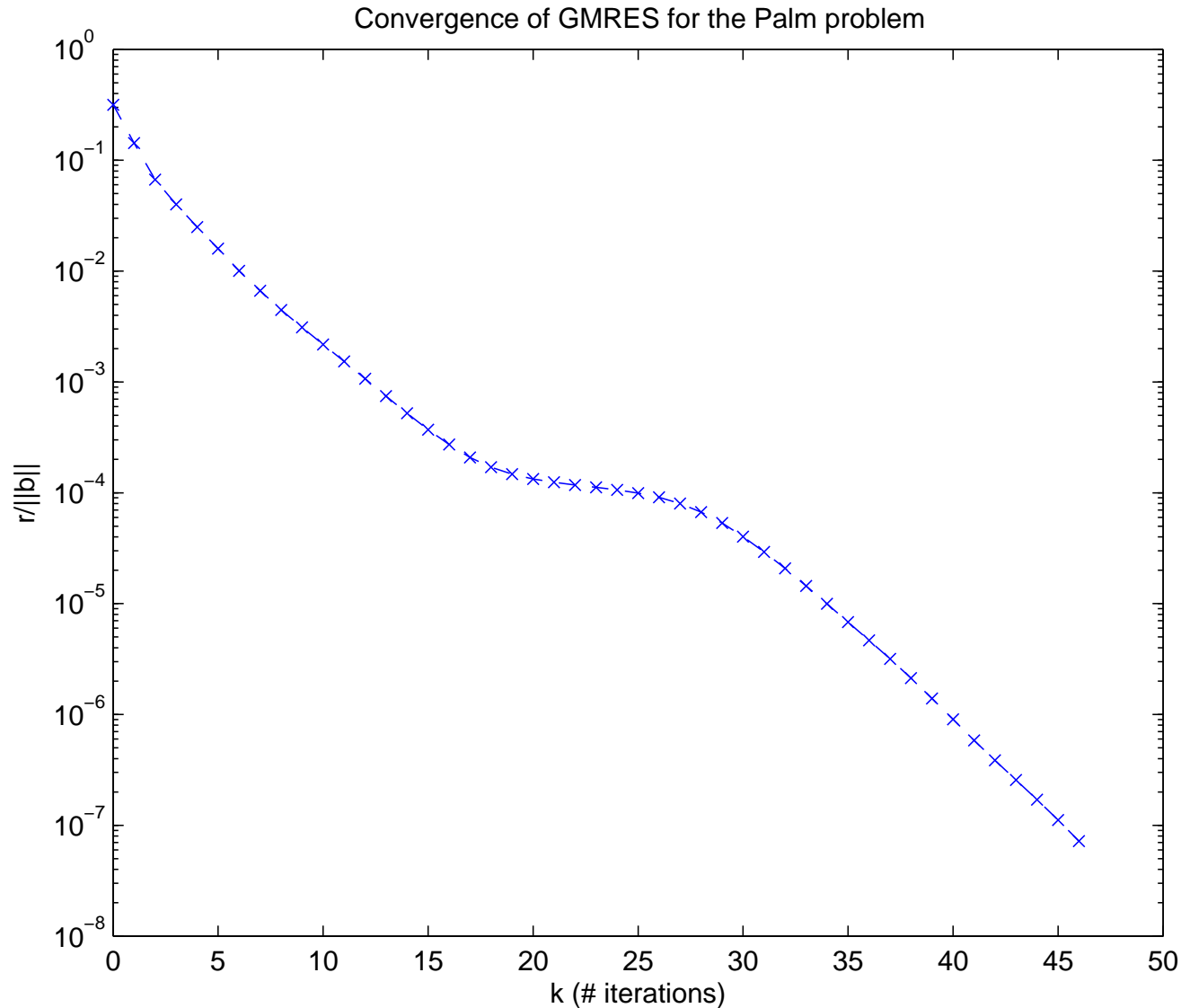
$$\frac{d}{dt} \int_{\mathcal{V}_i} c \, d\underline{\mathbf{x}} = - \int_{\mathcal{S}_i} (\underline{\mathbf{u}}c - d\nabla c) \cdot \underline{\mathbf{n}} \, d\underline{\mathbf{x}} + \int_{\mathcal{V}_i} p \, d\underline{\mathbf{x}}$$

$$\frac{|\mathcal{V}_i^n| \bar{c}_i^n - |\mathcal{V}_i^{n-1}| \bar{c}_i^{n-1}}{t_n - t_{n-1}} = \theta a_i^n + (1 - \theta) a_i^{n-1}$$

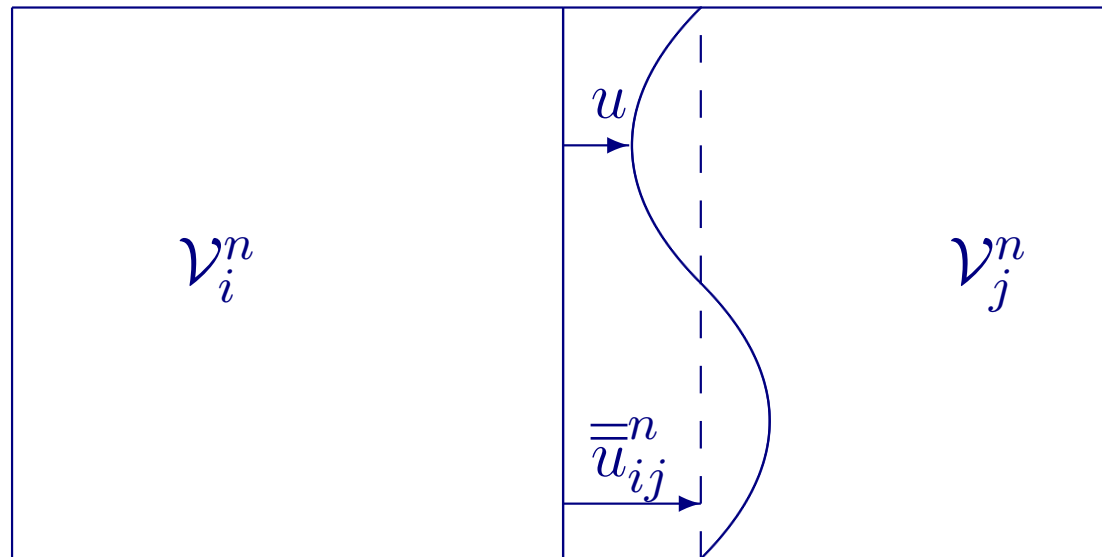
$$a_i^n = - \sum_j |\mathcal{S}_{ij}^n| \phi_{ij}^n + |\mathcal{V}_i^n| \bar{p}_i^n + \text{rvw}$$



Probleem I: GMRES convergeert te traag voor diffusie gedomineerde problemen.

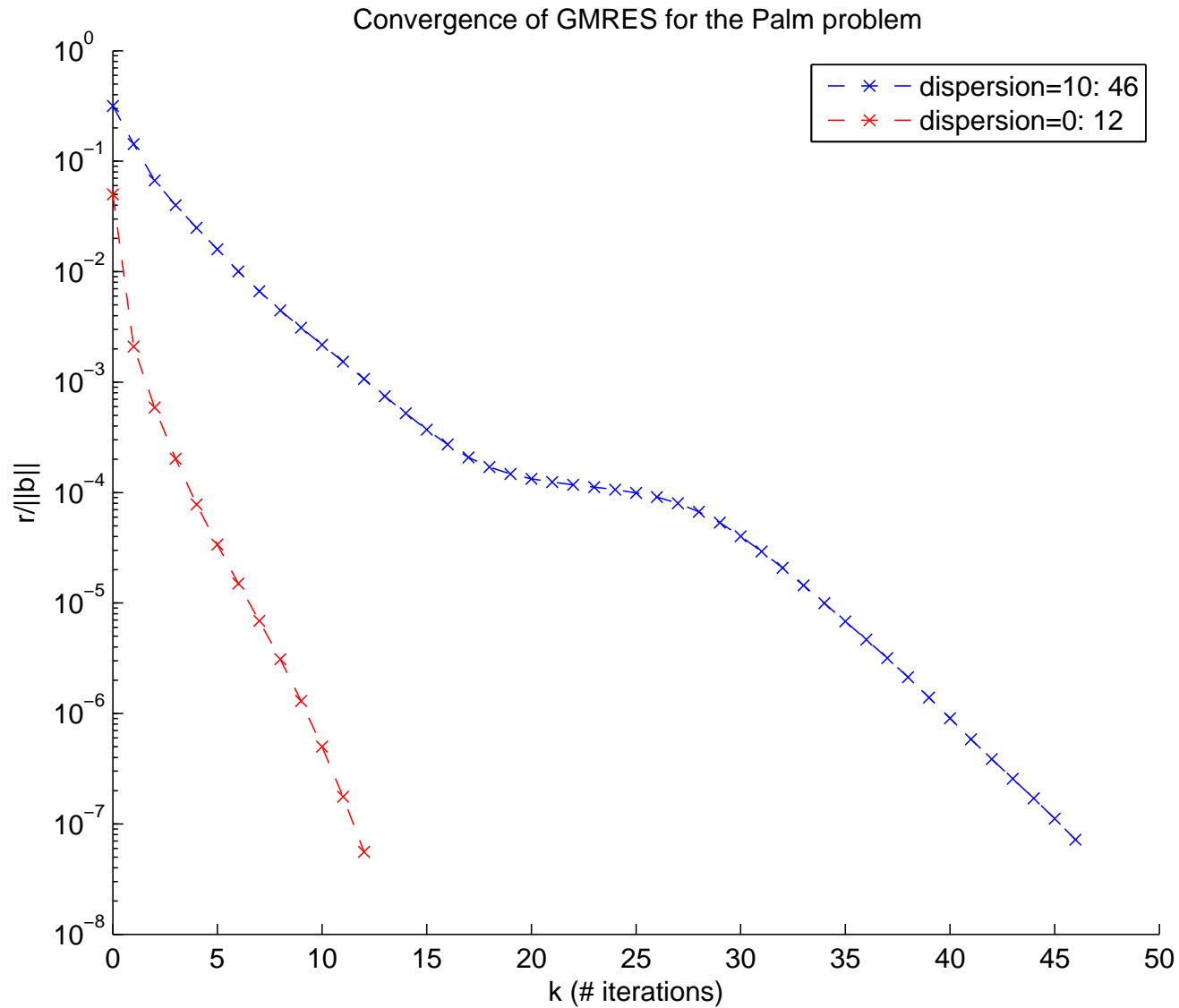


Diffusie dominantie ontstaat i.h.a. doordat het midden van de snelheid gecompenseerd wordt door extra diffusie.



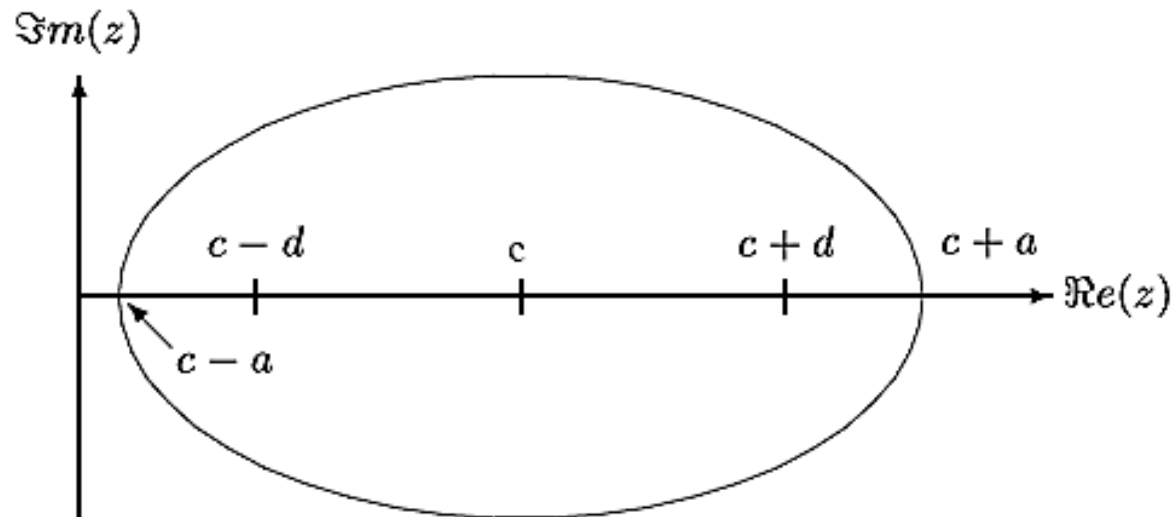
dimensie	extra diffusie
1	1000
2	10
3	1

GMRES convergeert veel sneller **zonder dispersie**.



Convergentie van GMRES is beter naarmate de **eigenwaarden** dichter bij elkaar liggen.

$$\underbrace{\frac{\|\underline{\mathbf{b}} - A\underline{\mathbf{x}}_k\|_2}{\|\underline{\mathbf{b}} - A\underline{\mathbf{x}}_0\|_2}}_{\text{relatieve fout}} \approx \underbrace{\kappa_2}_{\text{conditiegetal}} \left| \frac{a + \sqrt{a^2 - d^2}}{c + \sqrt{c^2 - d^2}} \right|^k$$



Preconditioneren van $A\underline{\mathbf{x}} = \underline{\mathbf{b}}$ kan GMRES versnellen door het spectrum te verbeteren.

$$\underbrace{P^{-1}A}_{\text{gunstiger spectrum}} \quad \underline{\mathbf{x}} = P^{-1}\underline{\mathbf{b}}$$

Preconditioneren van $A\underline{\mathbf{x}} = \underline{\mathbf{b}}$ kan GMRES versnellen door het spectrum te verbeteren.

$$\underbrace{P^{-1}A}_{\text{gunstiger spectrum}} \quad \underline{\mathbf{x}} = P^{-1}\underline{\mathbf{b}}$$

	huidige preconditioner	ILU preconditioner
factorisatie	$A = L + D + U$	$A \approx \tilde{L}\tilde{U}$
preconditioner	$P = \underbrace{(D + L)}_{\begin{smallmatrix} \diagdown \\ \underline{\quad} \end{smallmatrix}} \underbrace{D^{-1}}_{\diagdown} \underbrace{(D + U)}_{\begin{smallmatrix} \overline{\quad} \\ \diagup \end{smallmatrix}}$	$P = \underbrace{\tilde{L}}_{\begin{smallmatrix} \diagdown \\ \underline{\quad} \end{smallmatrix}} \underbrace{\tilde{U}}_{\begin{smallmatrix} \overline{\quad} \\ \diagup \end{smallmatrix}}$

Preconditioneren van $A\underline{\mathbf{x}} = \underline{\mathbf{b}}$ kan GMRES versnellen door het spectrum te verbeteren.

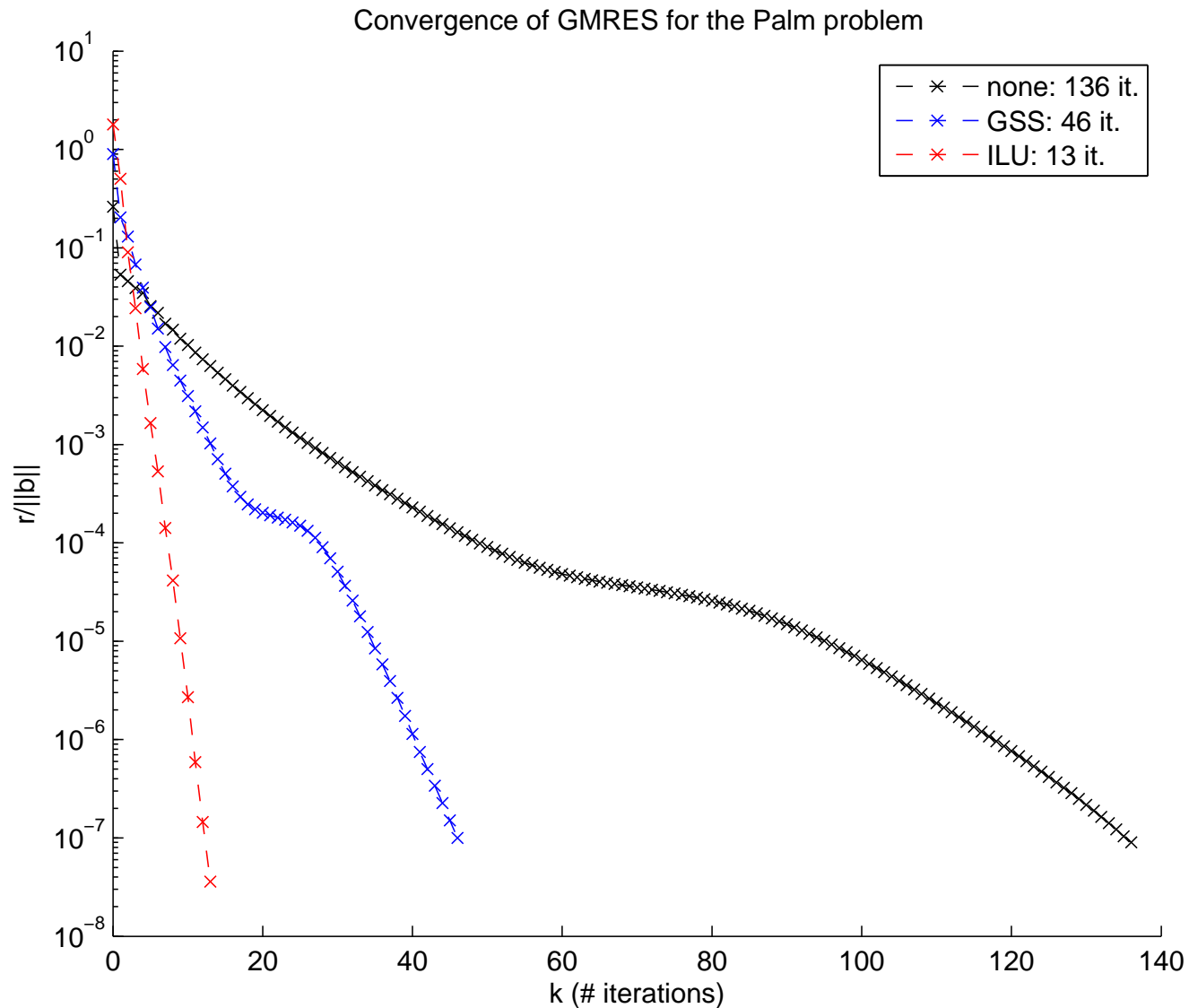
$$\underbrace{P^{-1}A}_{\text{gunstiger spectrum}} \quad \underline{\mathbf{x}} = P^{-1}\underline{\mathbf{b}}$$

	huidige preconditioner	ILU preconditioner
factorisatie	$A = L + D + U$	$A \approx \tilde{L}\tilde{U}$
preconditioner	$P = \underbrace{(D+L)}_{\begin{smallmatrix} \diagdown \\ \underline{\quad} \end{smallmatrix}} \underbrace{D^{-1}}_{\diagdown} \underbrace{(D+U)}_{\begin{smallmatrix} \overline{\quad} \\ \diagup \end{smallmatrix}}$	$P = \underbrace{\tilde{L}}_{\begin{smallmatrix} \diagdown \\ \underline{\quad} \end{smallmatrix}} \underbrace{\tilde{U}}_{\begin{smallmatrix} \overline{\quad} \\ \diagup \end{smallmatrix}}$

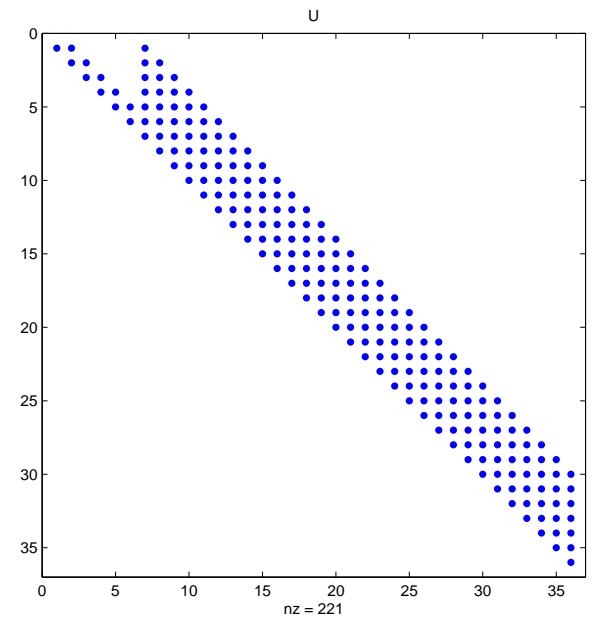
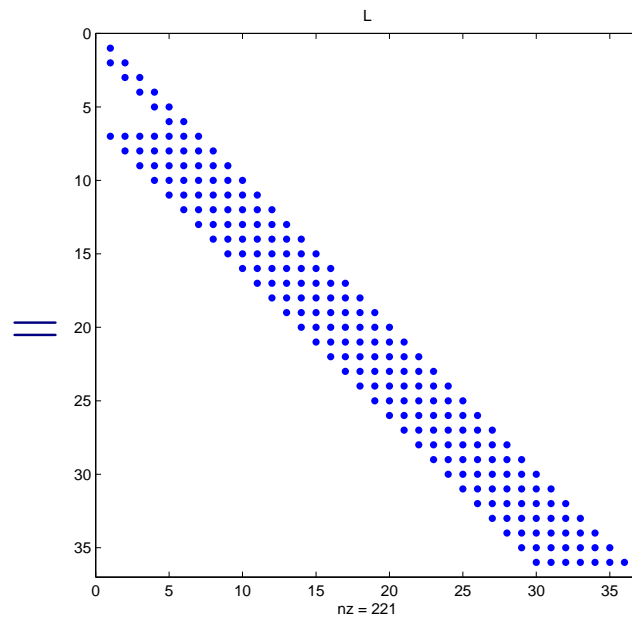
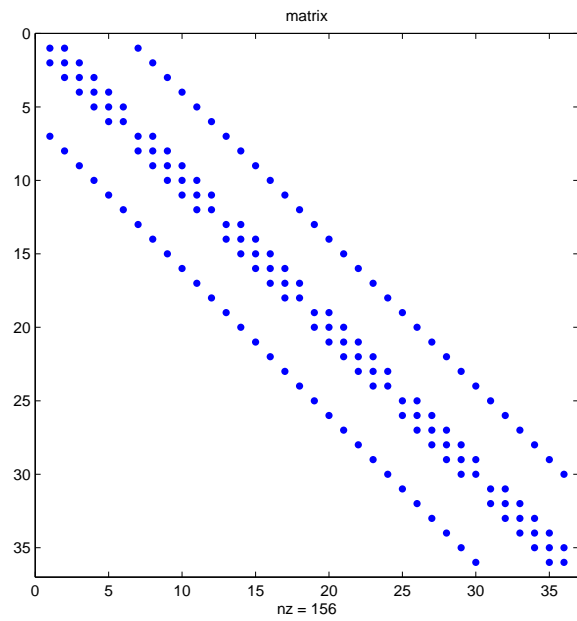
De **totale hoeveelheid werk** hangt af van het aantal iteraties, de constructie van P en het oplossen van stelsels $P^{-1}\underline{\mathbf{v}}$:

$$1 \times \text{'constructie'} + (2 + \text{'\# iteraties'}) \times \text{'stelsel oplossen'}$$

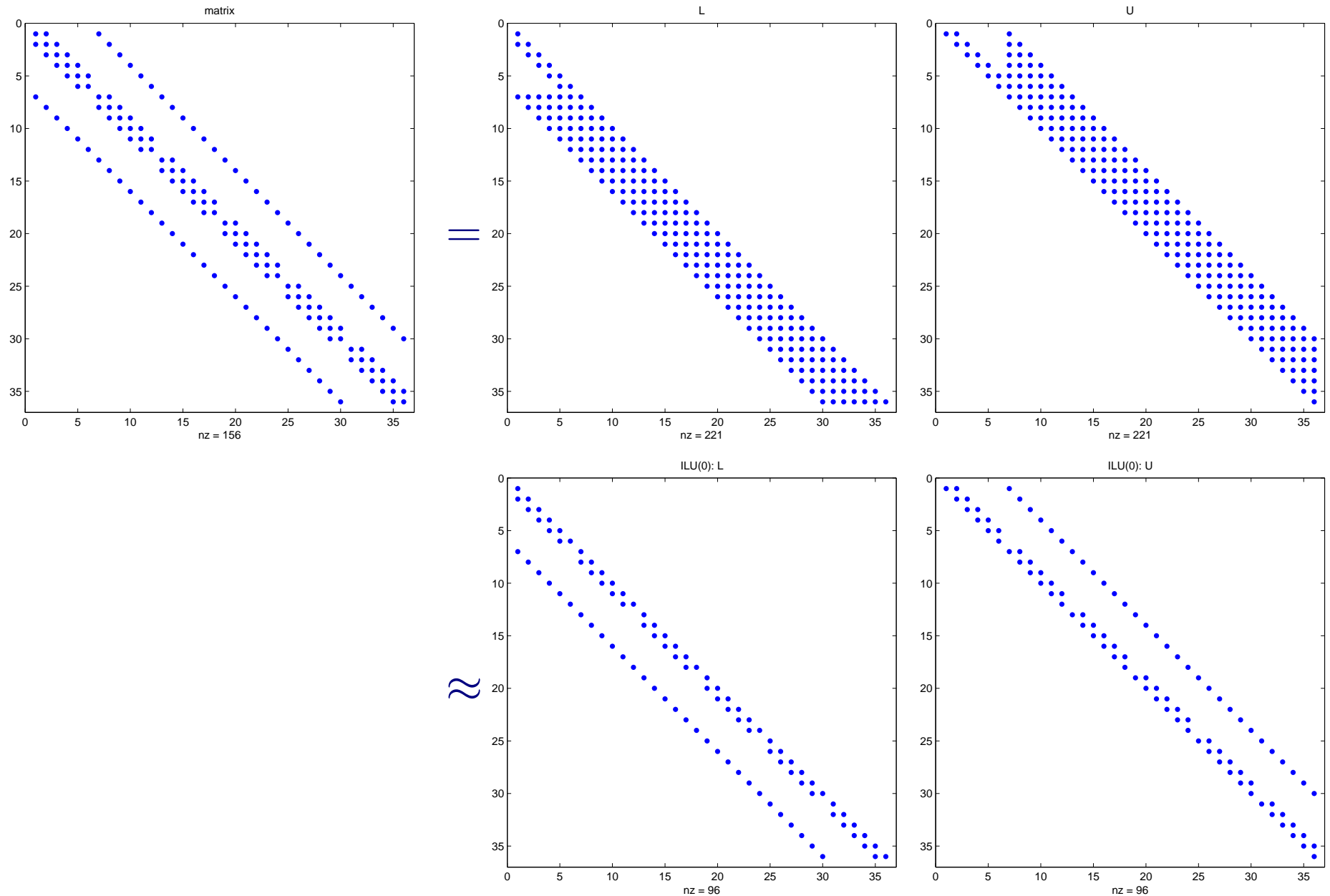
ILU(p) preconditioning levert **minder iteraties** dan de huidige preconditionering!



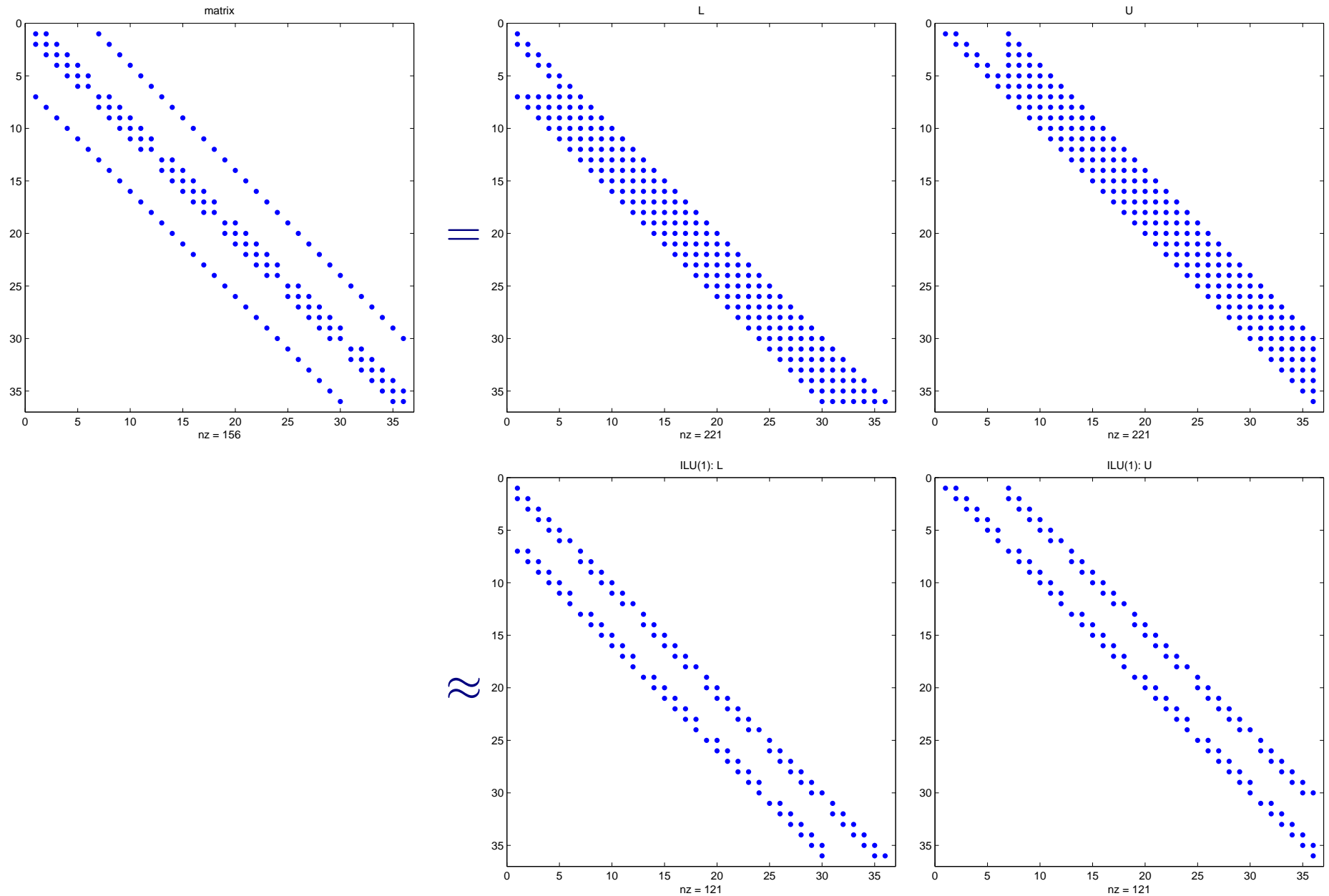
Een LU factorisatie is duur vanwege fill-in:



Een **incomplete LU factorisatie** neemt sommige elementen niet mee in de berekening:



Een **incomplete LU factorisatie** neemt sommige elementen niet mee in de berekening:

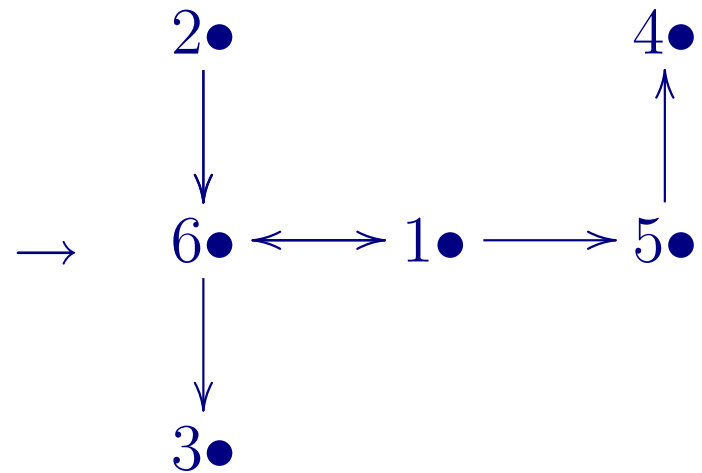


Vooraf **herstructureren** kan de kwaliteit van een preconditionering verbeteren.

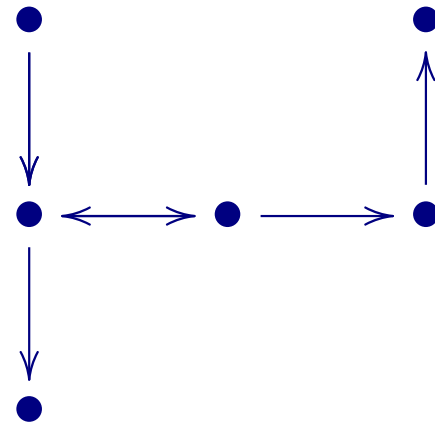
$$\begin{bmatrix} a_{11} & & & & a_{15} & a_{16} \\ & a_{22} & & & & a_{26} \\ & & a_{33} & & & \\ & & & a_{44} & & \\ & & & a_{54} & a_{55} & \\ a_{61} & & a_{63} & & & a_{66} \end{bmatrix} \xrightarrow{?} \begin{bmatrix} * & * & * & & & \\ & * & & * & & \\ * & & * & & & * \\ & & & * & & \\ & & * & & * & \\ & & & & & * \end{bmatrix}$$

Herordenen kan plaatsvinden door de knopen van de **adjacency graph** te hernoemen.

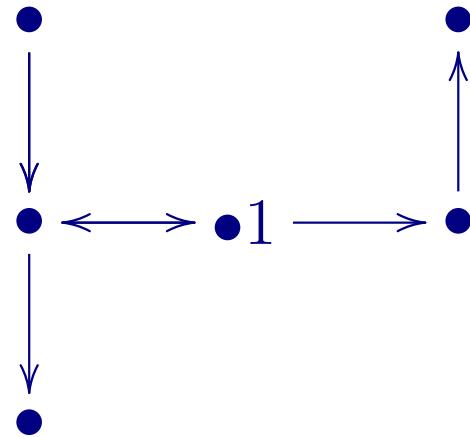
$$\begin{bmatrix} a_{11} & & & & a_{15} & a_{16} \\ & a_{22} & & & & a_{26} \\ & & a_{33} & & & \\ & & & a_{44} & & \\ & & & a_{54} & a_{55} & \\ a_{61} & a_{63} & & & & a_{66} \end{bmatrix}$$



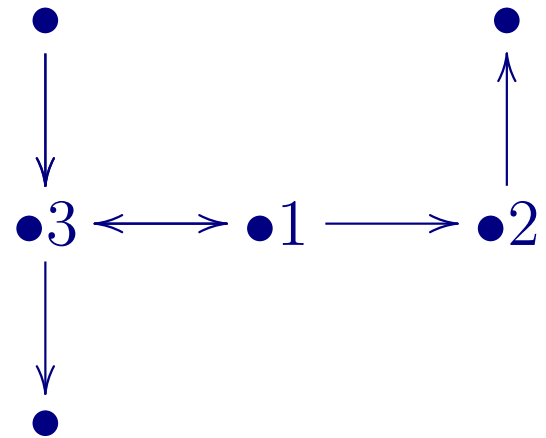
Herordenen kan plaatsvinden door de knopen van de adjacency graph te **hernummeren**.



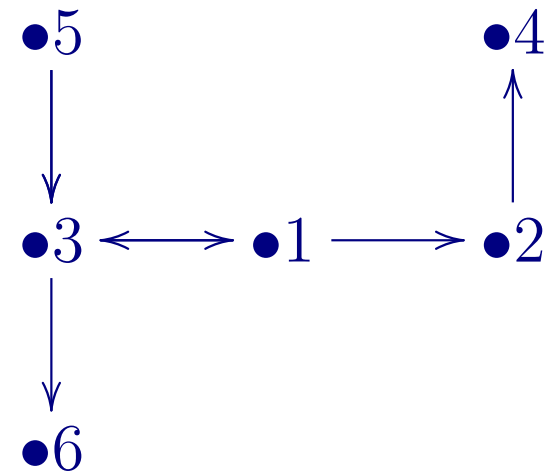
Herordenen kan plaatsvinden door de knopen van de adjacency graph te **hernummeren**.



Herordenen kan plaatsvinden door de knopen van de adjacency graph te **hernummeren**.



Herordenen kan plaatsvinden door de knopen van de adjacency graph te **hernummeren**.



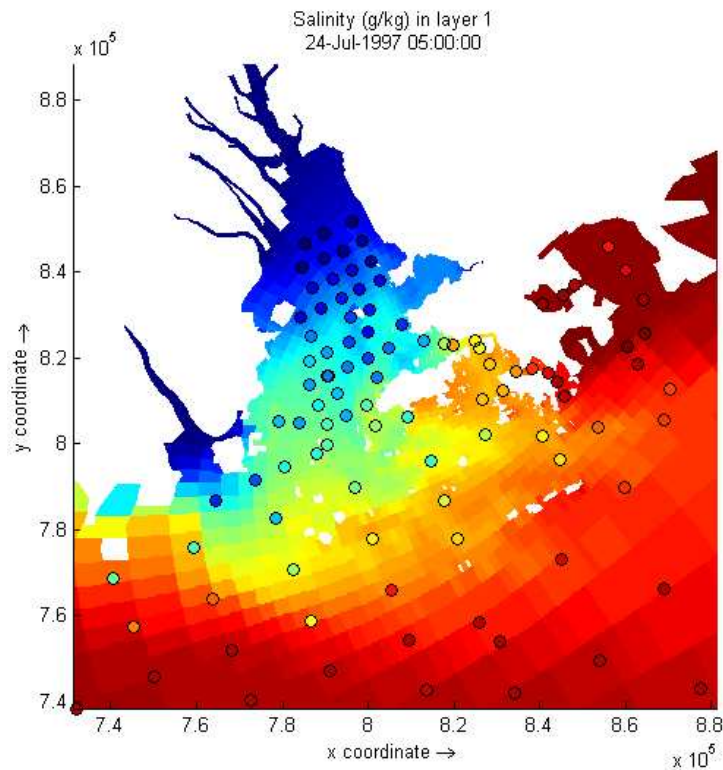
Wat moet **nader onderzocht** worden om de convergentiesnelheid van GMRES te optimaliseren?

1. Welke iteratieve methode/ herordening/ preconditionering combinatie is optimaal?
2. Is het zinvol om dezelfde - relatief dure - preconditioner voor meerdere tijdstappen te gebruiken?

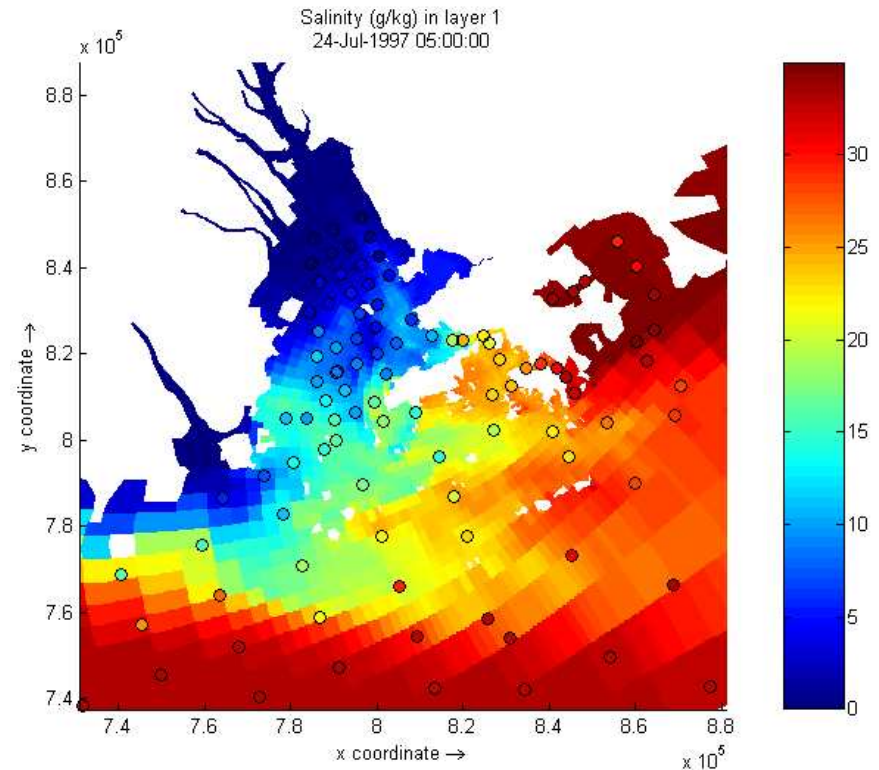


Probleem II: De huidige numerieke schema's zijn te traag of te onnauwkeurig.

Schema 16 van WAQ
20 minuten rekentijd

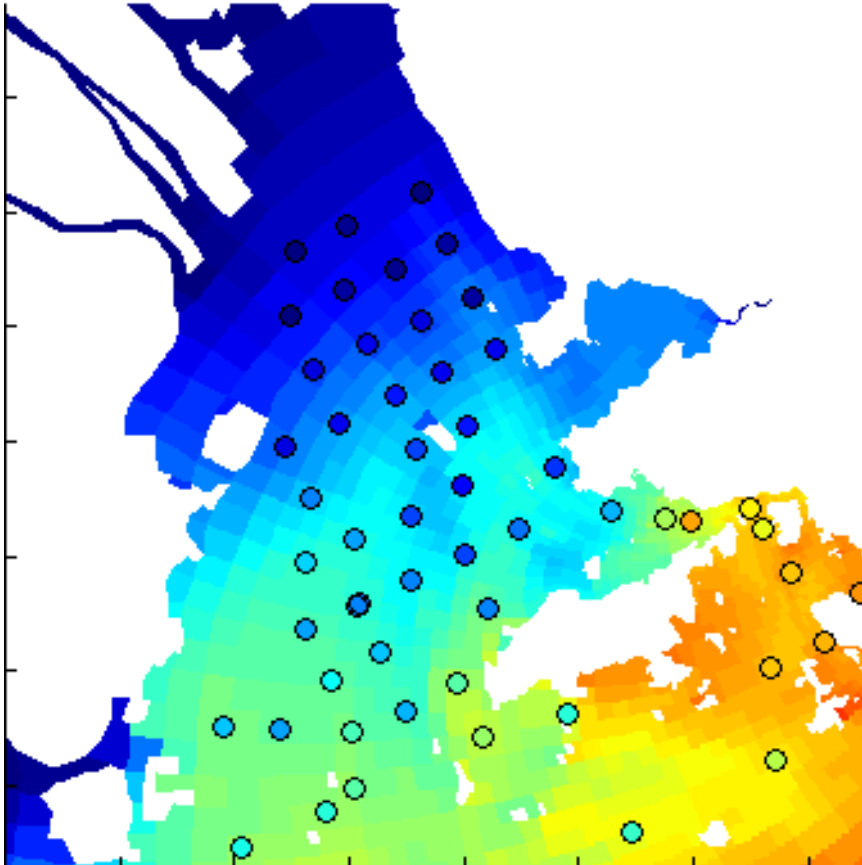


Schema 12 van WAQ
130 minuten rekentijd

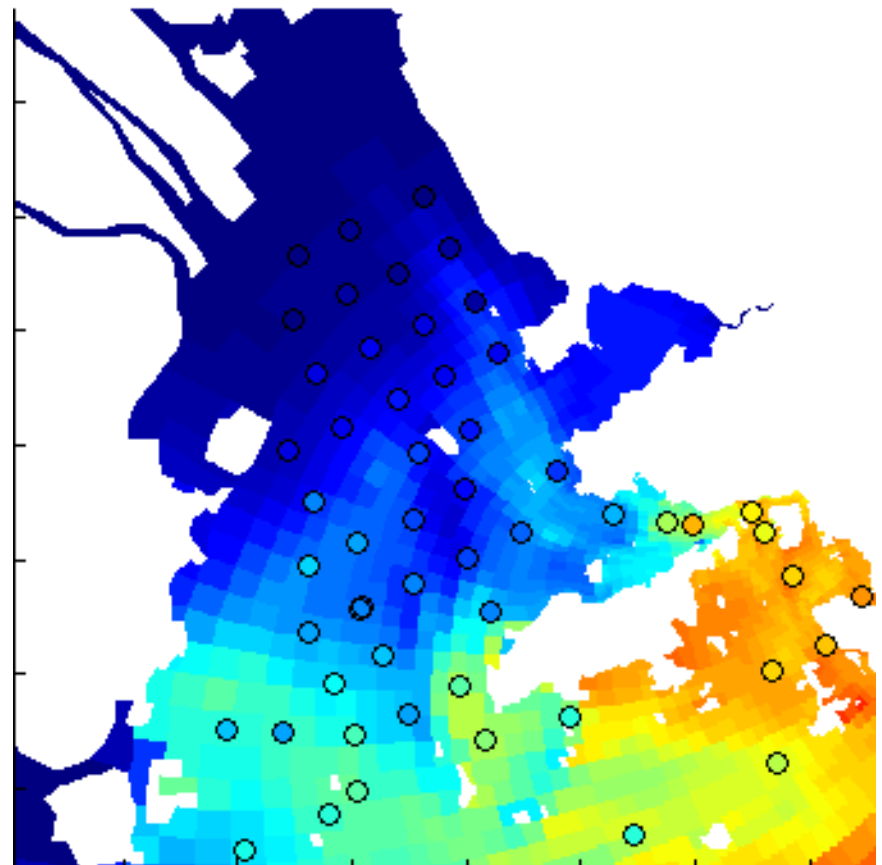


Probleem II: De huidige numerieke schema's zijn te **traag**
of te **onnauwkeurig**.

Schema 16 van WAQ
20 minuten rekentijd



Schema 12 van WAQ
130 minuten rekentijd



De **numerieke diffusiecoëfficiënt** van het eerste orde upwind schema voor de 1D advectionvergelijking luidt:

constante, positieve snelheid

constante tijdstap

$$\frac{u\Delta x}{2} \left(1 - (1 - 2\theta) \frac{u\Delta t}{\Delta x} \right)$$

equidistant grid

De numerieke diffusiecoëfficiënt van het eerste orde upwind schema voor de 1D advectionvergelijking luidt:

constante, positieve snelheid

constante tijdstap

$$\frac{u\Delta x}{2} \left(1 - (1 - 2\theta) \frac{u\Delta t}{\Delta x} \right)$$

equidistant grid

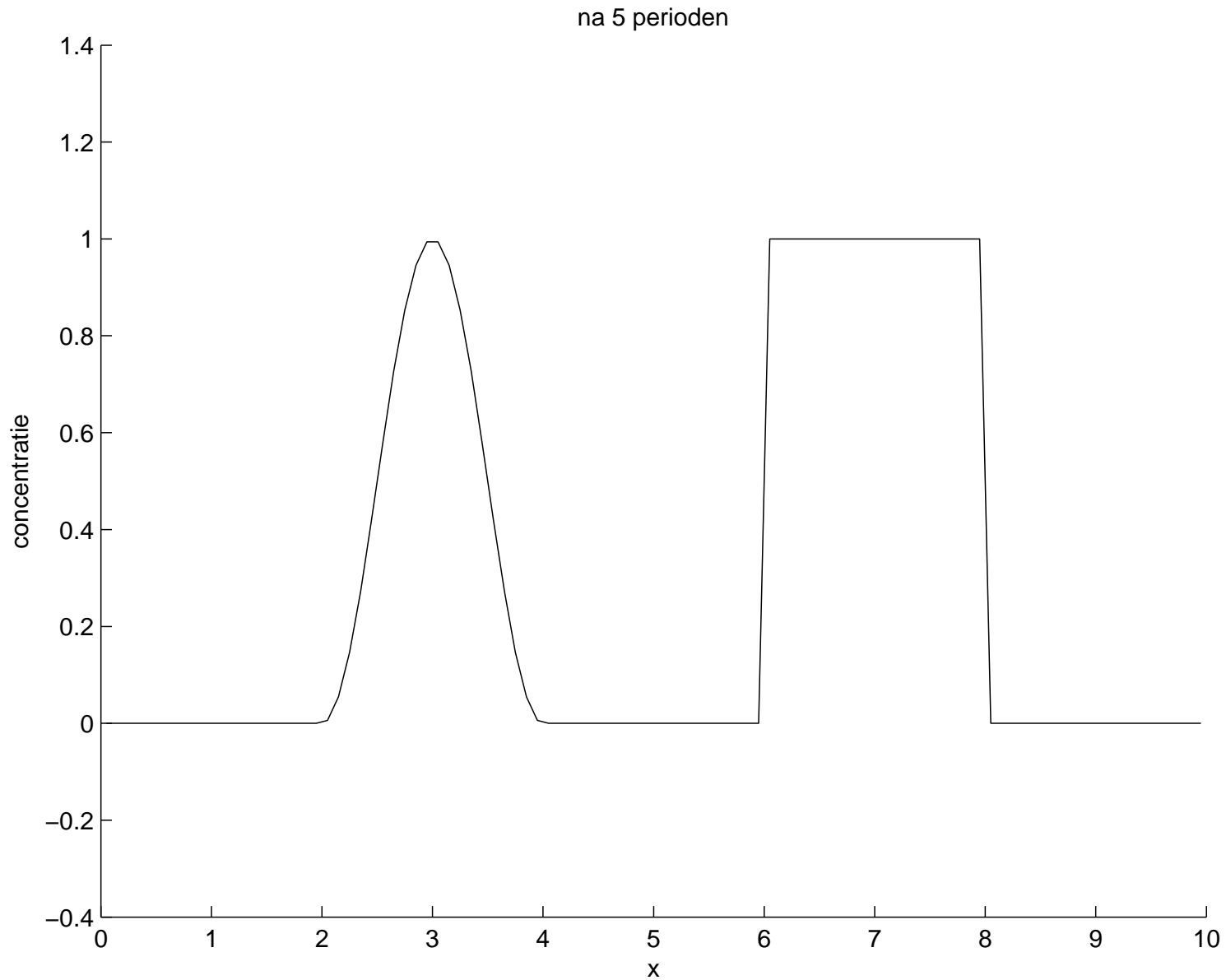
volledig impliciet ($\theta = 1$):

$$\frac{u\Delta x}{2} \left(1 + \frac{u\Delta t}{\Delta x} \right)$$

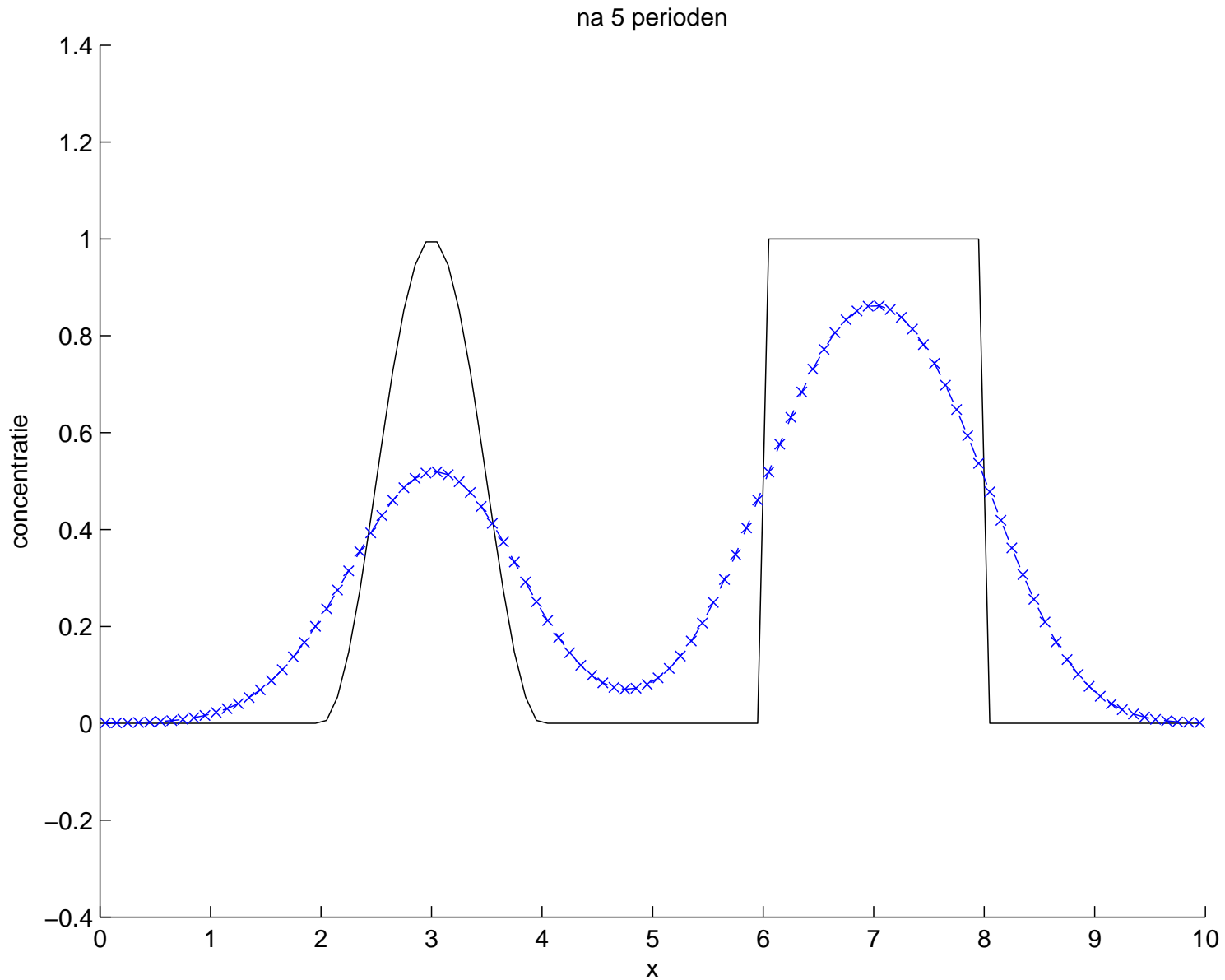
volledig expliciet ($\theta = 0$):

$$\frac{u\Delta x}{2} \left(1 - \frac{u\Delta t}{\Delta x} \right)$$

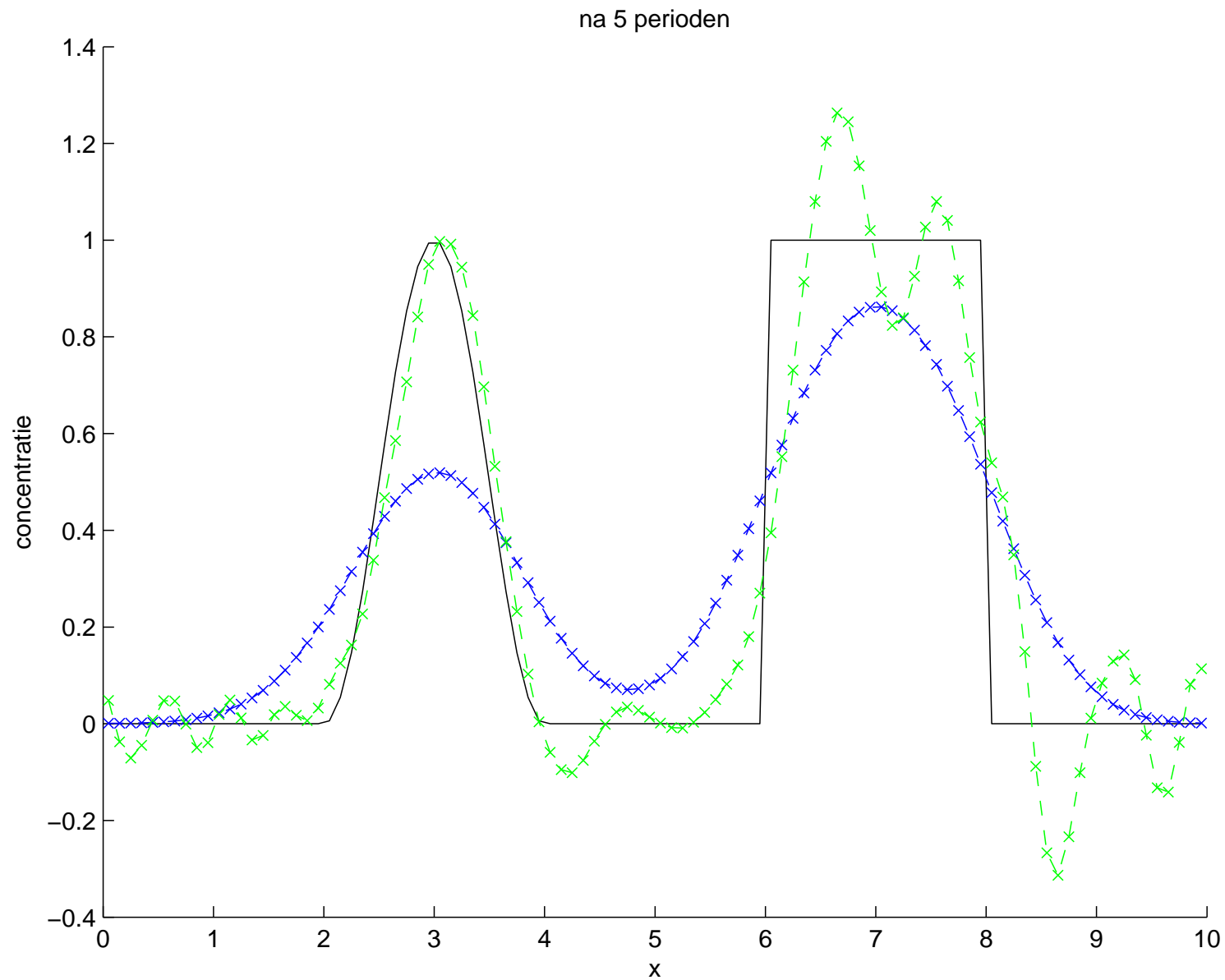
Het **flux correcting transport** algoritme levert een monotone methode met minder numerieke diffusie.



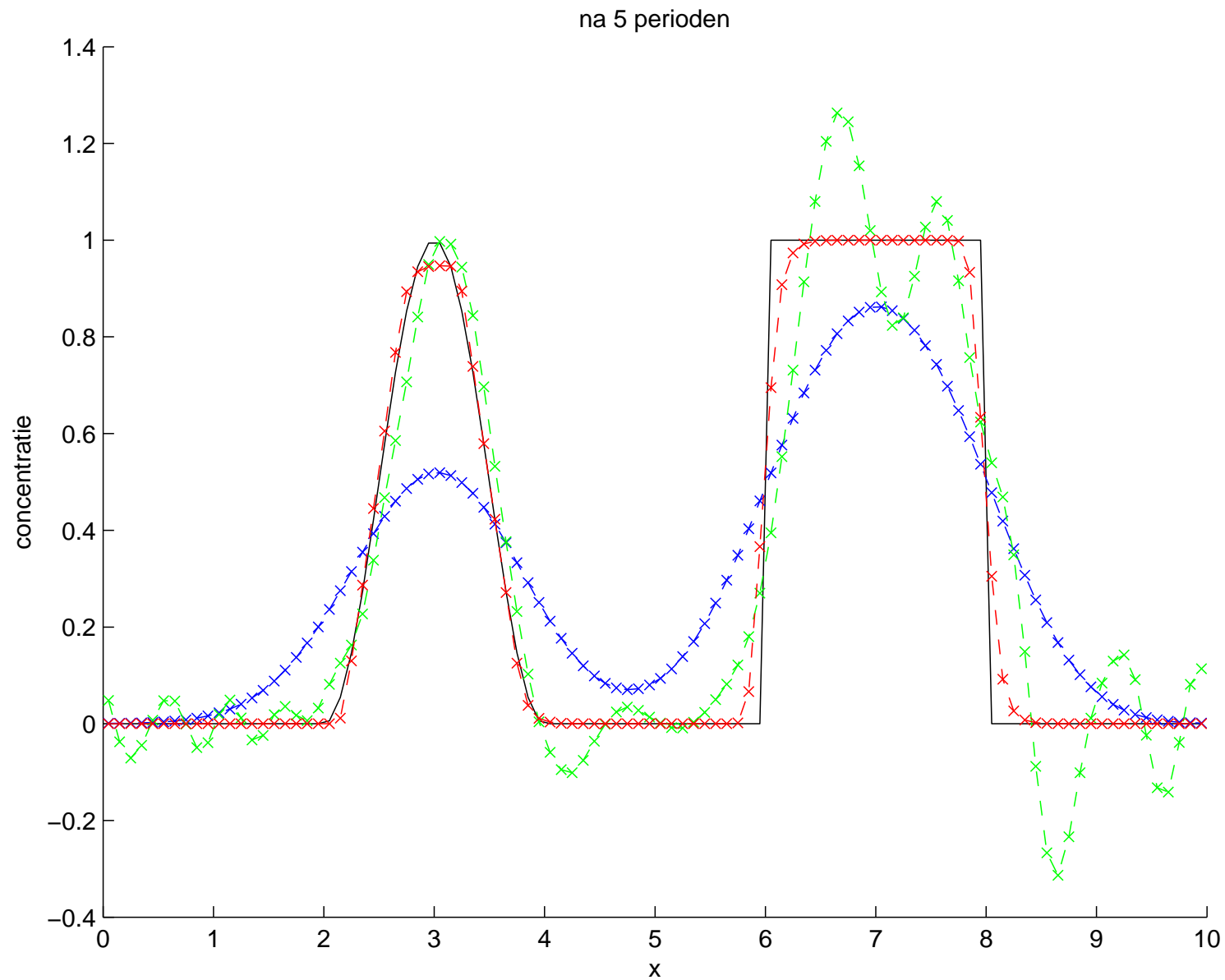
Het **flux correcting transport** algoritme levert een monotone methode met minder numerieke diffusie.



Het **flux correcting transport** algoritme levert een monotone methode met minder numerieke diffusie.



Het **flux correcting transport** algoritme levert een monotone methode met minder numerieke diffusie.



Het **flux correcting transport** algoritme is een compromis tussen een monotone eerste en een hogere orde methode.

$$\boxed{\phi_{ij}^n} = \underbrace{\hat{\phi}_{ij}^n}_{\text{eerste orde flux}} + \underbrace{l_{ij}^n}_{\text{limiter}} \underbrace{\left(\tilde{\phi}_{ij}^n - \hat{\phi}_{ij}^n \right)}_{\text{hogere orde flux correctie}}$$

Herinner de eindige volume methode:

$$\frac{|\mathcal{V}_i^n| \bar{c}_i^n - |\mathcal{V}_i^{n-1}| \bar{c}_i^{n-1}}{t_n - t_{n-1}} = \theta a_i^n + (1 - \theta) a_i^{n-1}$$

$$a_i^n = - \sum_j |\mathcal{S}_{ij}^n| \boxed{\phi_{ij}^n} + |\mathcal{V}_i^n| \bar{p}_i^n + \text{rvw}$$

Niet-lineaire termen kunnen expliciet benaderd worden.

eerste orde schema met $\bar{p}_i^n \approx \bar{p}_i^{n-1}$

\downarrow
 $\hat{\bar{c}}_i^n$

eerste orde flux $\hat{\phi}_{ij}^n$

\downarrow

limiter $l_{ij}^n \longrightarrow \phi_{ij}^n \longrightarrow \bar{c}_i^n$

\uparrow

hogere orde flux $\tilde{\phi}_{ij}^n$

\uparrow
 $\tilde{\bar{c}}_i^n$

hogere orde schema met $\bar{p}_i^n \approx \bar{p}_i^{n-1}$

Wat moet **nader onderzocht** worden om een nauwkeurige en robuuste FCT methode te ontwikkelen?

1. Welke limiters en numerieke fluxen zijn geschikt?
2. Is de bovengrens voor de tijdstap voldoende groot?
3. Zorgt een grote tijdstap niet voor teveel limiten, waardoor de eerste orde methode opnieuw verkregen wordt?

