Iterative solution methods for the simulation of flow in industrial glass furnaces

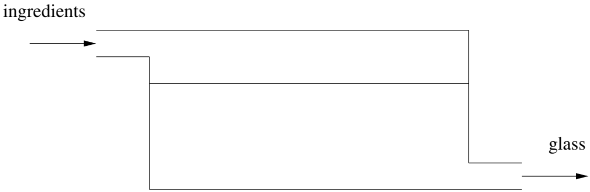
Eline Jonkers

Delft University of Technology

TNO Science and Industry

Outline

Industrial glass furnace



Mathematical simulation of flows

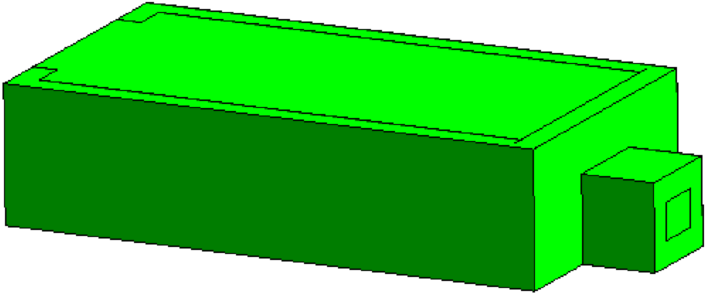
Why simulate?

- Physical experiments costly and time-consuming
- Certain physical quantities hard to measure

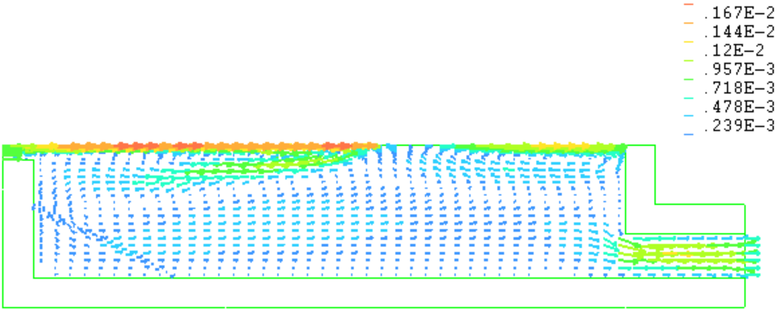
X-stream:

- CFD simulation package for glass industry
- Developed at TNO Science and Industry
- Lots of models available: combustion, turbulence, radiation, stirring, etc.

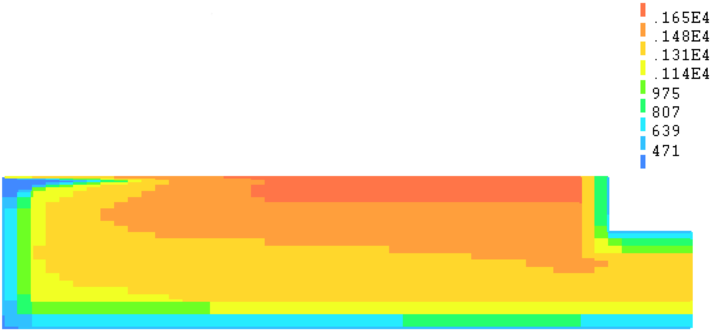
Geometry of a glass furnace



Simulation of velocities in a glass furnace

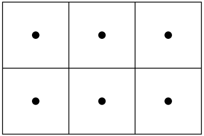


Simulation of temperature in a glass furnace

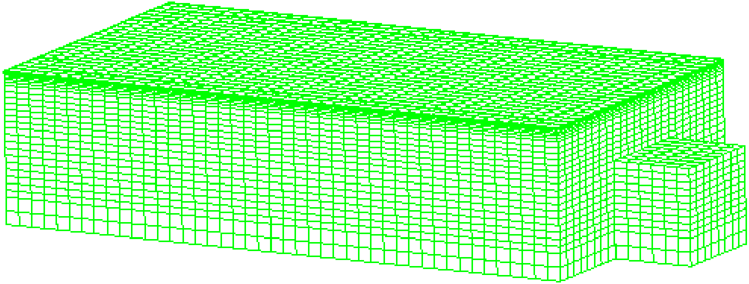


Solving mathematical flow model

- Partial differential equations arise from model
- Impossible to solve directly
- Transformation to finite number of difference equations
- Solve system of equations with iterative solution method



Grid of a glass furnace



Goal of the Master's project

Goal: improve X-stream algorithms

Focus: iterative solution methods combined with deflation

Purpose of deflation

System to be solved: $A\mathbf{x} = \mathbf{b}$

Condition number $\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$ (A SPD)

Smaller condition number $\longrightarrow$ faster convergence.

Deflation removes smallest eigenvalues A.

Basic idea of deflation

System to be solved: $A\mathbf{x} = \mathbf{b}$

$$P = I - AZ(Z^TAZ)^{-1}Z^T$$
$$Q = I - Z(Z^TAZ)^{-1}Z^TA \quad (PA = AQ)$$

$$\mathbf{x} = (I - Q)\mathbf{x} + Q\mathbf{x}$$

Calculation of $\mathbf{x}$:

- $(I - Q)\mathbf{x} = Z(Z^TAZ)^{-1}Z^TA\mathbf{x} = Z(Z^TAZ)^{-1}Z^T\mathbf{b}$
- Solve $PA\tilde{\mathbf{x}} = P\mathbf{b}$ for $\tilde{\mathbf{x}}$
- Premultiply result with Q

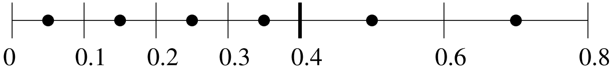
Choice of Z

Choice of Z important for convergence.

Possibilities:

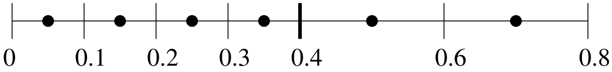
- Constant deflation (CD)
- Constant linear deflation based on grid numbering (CLD-ijk)
- Constant linear deflation based on grid coordinates (CLD-cartesian)

Constant deflation (CD)



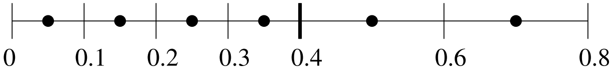
$$Z_{CD} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Constant linear deflation based on grid numbering (CLD-ijk)



$$Z_{CLD-ijk} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & 4 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

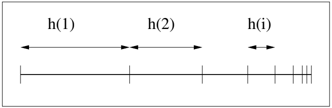
Constant linear deflation based on grid coordinates (CLD-cartesian)



$$Z_{CLD-cartesian} = \begin{pmatrix} 1 & 0.05 & 0 & 0 \\ 1 & 0.15 & 0 & 0 \\ 1 & 0.25 & 0 & 0 \\ 1 & 0.35 & 0 & 0 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 1 & 0.7 \end{pmatrix}$$

Experiments MATLAB

$$\frac{d^2\varphi}{dx^2} = x \sin x, \quad \varphi(0) = \varphi(\pi) = 0$$



Deflated GCR with 3 subdomains:

n	CD	CLD-ijk	CLD-cartesian
51	46	32	13
72	63	50	20

Condition number: $\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$

$\kappa(PA_{CLD-ijk}) = 1.4 \cdot 10^5$

$\kappa(PA_{CLD-cart}) = 8.9 \cdot 10^4$

Experiments X-stream

Input:

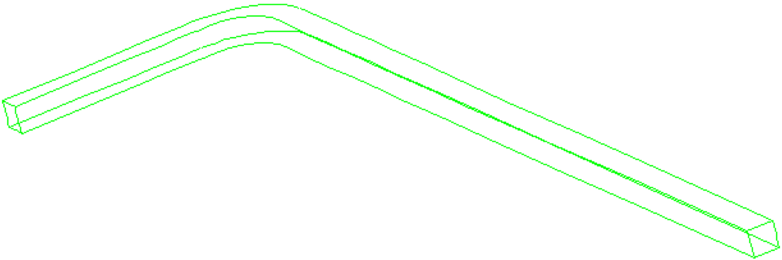
- number of inner iterations
- iterative method: SIP or GCR with CD, CLD-ijk or CLD-cartesian

Output:

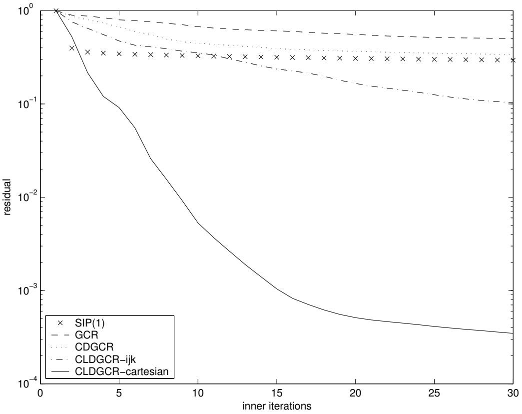
- residuals
- number of outer iterations
- wall-clock time

Note that the number of outer iterations and the wall-clock time depend on the residuals.

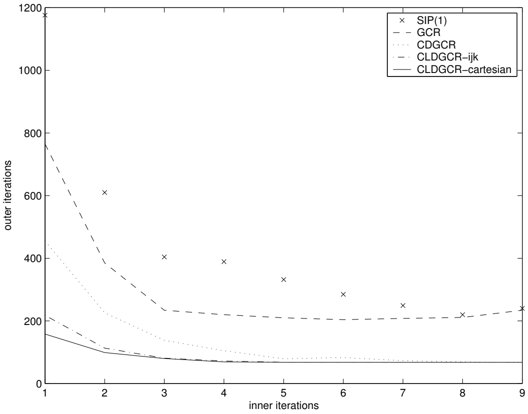
Test case channel



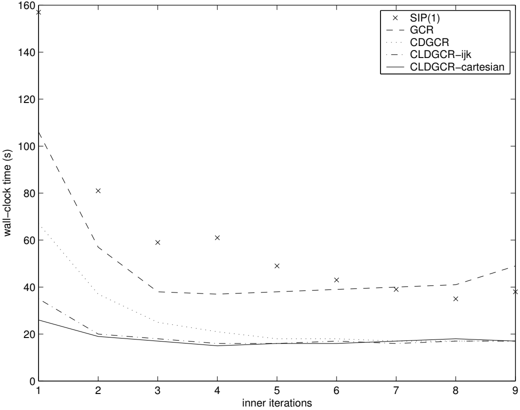
Residuals channel



Number of outer iterations channel



Wall-clock times channel



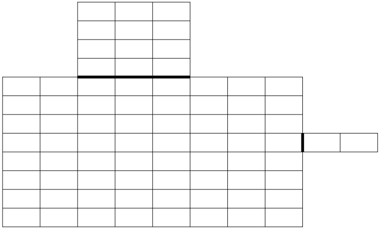
Results wall-clock times channel test case

inner it.	SIP(1) / GCR	SIP(1) / CDGCR	SIP(1) / CLDGCR-ijk	SIP(1) / CLDGCR-cart
1	1.5	2.3	4.5	6.0
2	1.4	2.2	4.1	4.3
6	1.1	2.4	2.5	2.7
9	0.8	2.2	2.2	2.2

Using optimal number of inner iterations for each method:

SIP(1) / GCR	SIP(1) / CDGCR	SIP(1) / CLDGCR-ijk	SIP(1) / CLDGCR-cart
0.9	2	2.1	2.3

Subdomain partitioning



$E = Z^T A Z$

600 subdomains:

E_{CD} 600×600-matrix

E_{CLD} 2400×2400-matrix

Conclusions

- GCR performs better with deflation
- Constant linear deflation performs better than constant deflation
- CLD-cartesian performs better than CLD-ijk
- More subdomains means less iterations
- Drawback of CLD: not suitable for large number of subdomains

Recommendations

Further research subdomain partitioning:

- How to decrease number of subdomains
- Implement a suitable sparse solver for unstructured matrices

Questions?