

Modellering van de ruimtelijke lucht en temperatuur verdeling in een semigesloten kas

Naam vak : Afstudeervak Meet-, Regel- en Systeemtechniek
Nummer : SCO-80436
Omvang : 36 ects
Datum : Oktober 2009

Student : A.R. Steennis
Registratienummer : 85-06-07-799-080
Opleiding : Agricultural and Bioresource Engineering

Begeleider(s) : L.G. van Willigenburg, J.B. Campen
Examinatoren : Prof.dr.ir. G. van Straten,
Leerstoelgroep : Meet-, Regel- en Systeemtechniek
Adres : Bornsesteeg 59
6708 PD Wageningen
Tel: (0317) 48 21 24
Fax: (0317) 48 49 57



Samenvatting

Semigesloten kasconcepten zijn de laatste jaren sterk in ontwikkeling. De reden hiervan is dat semigesloten kassen minder afhankelijk zijn van fossiele energie en zelfs de mogelijkheid hebben om energieleverend te zijn. Een belangrijke stap die moet worden gemaakt bij het ontwikkelen van semigesloten kassen is het optimaliseren van klimaatbeheersing. Om bijvoorbeeld de temperatuurbeheersing te optimaliseren is het nodig om een model te ontwikkelen dat op verschillende plaatsen in de kas de temperatuur kan berekenen in de tijd. Het doel van dit onderzoek was dan ook: het ontwikkelen van een dynamisch model dat de temperatuur beschrijft in een kassysteem.

Aan de basis van het ontwikkelen van het model staan fundamentele fysische behoudswetten zoals het behoud van massa en het behoud van energie. Op basis van deze wetten is een model in toestandsvorm opgesteld. Het bleek echter dat het model in deze vorm stijf is en daardoor leidde dat tot divergentie bij numerieke integratie. In de fysica is deze problematiek nauwkeurig bestudeerd en heeft het geleid tot numerieke oplossingen die op een intelligente manier de fysische interpretatie van het model gebruiken bij het numeriek oplossen, zoals het SIMPLE algoritme.

In dit onderzoek is gekozen om vervolgens het SIMPLE algoritme toe te passen op de behoudswetten om tot een numerieke oplossing te komen. Wanneer het SIMPLE algoritme wordt toegepast blijken de resultaten ervan fysisch goed verklaarbaar. Zo blijkt een initieel aangebrachte warme spot in verloop van tijd te stijgen en een koude spot te dalen. Daarnaast blijkt dat luchtstroming circuleert in een gesloten ruimte wat eveneens fysisch verklaarbaar is aangezien de luchtstroming nabij de wanden een andere weg zal moeten vinden. Vervolgens is het model gevalideerd met het officiële CFD pakket genaamd FLUENT. Het blijkt dat de oplossing van het model redelijk goed overeen komt met de oplossing van FLUENT. Hiervoor moet wel een kleinere discretisatie tijd worden gekozen ten opzichte van FLUENT. Daarnaast heeft het model langere rekestijden nodig dan FLUENT.

Inhoudsopgave

1	Inleiding	4
1.1	Doelstelling	4
1.2	Opbouw Rapport	4
2	Eigenschappen van de kas	5
2.1	Kasmodellering	5
2.2	Eigenschappen	5
3	Fysische model vergelijkingen	7
3.1	Stromingsleer	7
3.2	Behoudswetten	8
3.2.1	Continuïteitsvergelijking	8
3.2.2	Impulsvergelijking	9
3.2.3	Energievergelijking	11
3.3	Opstellen van een extra vergelijking	13
3.3.1	Dichtheid als functie van de temperatuur	14
3.3.2	$\rho(T)$ in continuïteitsvergelijking	14
3.3.3	$\rho(T)$ in impulsvergelijkingen	15
3.3.4	$\rho(T)$ in energievergelijking	16
3.3.5	Model met vijf toestandsvariabelen	16
4	Toestandsvorm	18
4.1	Omschrijven naar toestandsvorm	18
4.2	Numerieke oplossingsmethode	20
4.2.1	Boussinesq benadering	21
5	Numeriek oplossen van de vergelijkingen	23
5.1	Ruimtelijke discretisatie	23
5.2	Oplossen van de vergelijkingen	25
5.2.1	Matrixmethode	25
5.2.2	Onderscheiden van variabelen	25
5.3	Discretiseren van de vergelijkingen	26
5.3.1	Discretisatie van de impulsvergelijkingen	26
5.3.2	Discretisatie van de continuïteitsvergelijking	29
5.3.3	Discretisatie van de energievergelijking	29
5.3.4	Set vergelijkingen	31

6	Het SIMPLE algoritme	32
6.1	Introductie SIMPLE algoritme	32
6.2	Het SIMPLE algoritme	33
6.2.1	Stap 1: impulsvergelijkingen	33
6.2.2	Stap 2: druk correctievergelijking	34
6.2.3	Stap 3: het corrigeren van de velden	36
6.2.4	Stap 4: het oplossen van de energievergelijking	38
7	Model	39
7.1	Bepalen van de parameters	39
7.1.1	Discretisatie tijd	39
7.1.2	Onderrelaxatie factoren	41
7.1.3	Convergentie criterium	42
7.2	Modellering met een grid van 5 bij 5	42
7.2.1	Afwijkende subruimte onderin	42
7.2.2	Afwijkende subruimte bovenin	44
7.2.3	Afwijkende subruimtes op meerder plaatsen	45
7.3	Validatie model met CFD pakket	45
7.3.1	Grid van 10 bij 10	46
7.3.2	Grid van 25 bij 25	47
7.3.3	Grid van 50 bij 50	48
8	Conclusie en aanbevelingen	50
8.1	Conclusie	50
8.2	Aanbevelingen	50
8.2.1	Rekentijd	51
8.2.2	Nauwkeurigheid	51
8.2.3	Optimal Control	51
9	Literatuur	52
10	Bijlagen	53
10.1	Symbolenlijst	53
10.2	Quasi-statisch maken van vergelijkingen	54
10.3	Afgeleide term in impulsvergelijkingen	54
10.4	Randvoorwaarden	55
10.5	Matlab Model	60

Hoofdstuk 1

Inleiding

1.1 Doelstelling

De laatste jaren is er steeds meer de trend om semigesloten kassystemen te ontwikkelen. De reden hiervan is dat semigesloten kassen minder afhankelijk zijn van fossiele energie en zelfs de mogelijkheid hebben om energieleverend te zijn. Een belangrijke stap die moet worden gemaakt bij het ontwikkelen van semigesloten kassen is het optimaliseren van de klimaatbeheersing. Zo zijn er luchtbehandelingskasten ontwikkeld die kunnen koelen, verwarmen, bevochtigen en ontvochtigen en daarmee het klimaat, en dus ook de temperatuur, kunnen aanpassen.

Om de temperatuurbeheersing in de kas te optimaliseren is het nodig een model te ontwikkelen dat op verschillende plaatsen in de kas de temperatuur kan berekenen in de tijd. Een dergelijk meerlagen model is nodig om een betere voorspelling te doen van de invloed van de luchtbehandelingskasten op de temperatuur op verschillende plaatsen in de kas. Daarnaast is een dergelijk model geschikt om een geavanceerde regeling toe te kunnen passen. Het doel van dit onderzoek is daarom ook: het ontwikkelen van een dynamisch model dat de temperatuur beschrijft in een kassysteem.

1.2 Opbouw Rapport

In hoofdstuk 2 wordt kort uitgelegd wat de eigenschappen zijn van de kas voor het opstellen van het model. In hoofdstuk 3 worden de fysische vergelijkingen opgesteld die nodig zijn om de processen die in een kas plaatsvinden te modelleren. In hoofdstuk 4 worden de vergelijkingen die zijn opgesteld zodanig verwerkt dat ze gebruikt kunnen worden voor numerieke integratie. Uiteindelijk blijkt dat er toch voor een andere oplossingsmethode moet worden gekozen, namelijk het SIMPLE algoritme. In hoofdstuk 5 worden de vergelijkingen ruimtelijk en in de tijd gediscrètiseerd. In hoofdstuk 6 wordt het SIMPLE algoritme uitgelegd en uitgewerkt. In hoofdstuk 7 wordt het model wat is opgesteld in de hoofdstukken daarvoor geprogrammeerd met Matlab en gerund voor verschillende test situaties. In hoofdstuk 8 wordt vervolgens een conclusie getrokken over het gehele onderzoek. Daarnaast worden er in dit hoofdstuk aanbevelingen gedaan om het model te verbeteren en eventueel wat elementen hieruit zijn voor vervolgonderzoek.

Hoofdstuk 2

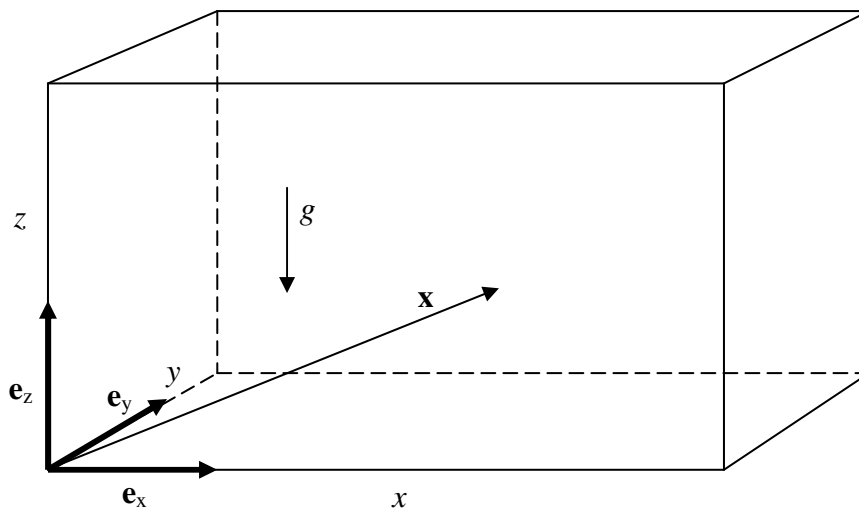
Eigenschappen van de kas

2.1 Kasmodellering

De temperatuurverdeling in een kas is al meerdere jaren een onderwerp van discussie. Er bestaan veel vragen over de gewenste temperatuurverhouding tussen de top van het gewas, de vruchten en de wortels. Ook in semi-gesloten kassen, waarbij de temperatuurbeheersing gerealiseerd wordt met apart geplaatste luchtbehandelingskasten, is de juiste temperatuurverhouding onderwerp van discussie. Echter, de temperatuurbeheersing in een semi-gesloten kas is vaak gebaseerd op een model met homogene temperatuur. Om de juiste temperatuurverhouding in een kas te realiseren zal er geavanceerde regeltechniek gebruikt moeten worden die deze temperatuurverhouding kan realiseren en daarvoor is dan ook een model nodig wat de temperatuur beschrijft op verschillende punten in de ruimte.

2.2 Eigenschappen

Om de temperatuur in een kas te modelleren moet eerst worden bepaald wat de ruimte is waarbinnen wordt gemodelleerd en welke eigenschappen er aan verbonden zijn. De ruimte waarover wordt gesproken is een kas, wat in dit onderzoek betekent dat we te maken hebben met een afgesloten ruimte, zie de figuur hieronder:



Figuur 2.1: Kassysteem met gegeven notaties

Om de temperatuurverdeling in een ruimte te modelleren zal ook de luchtstroming moeten worden gemodelleerd. In dit onderzoek wordt er niet van uit gegaan dat er zich in of

uitgangen bevinden in de ruimte. Dit betekent dat de luchtstroming, die kan ontstaan door een initieel uit evenwicht getrokken temperatuurverdeling, niet in of uit de ruimte kan stromen maar altijd binnen de ruimte zal blijven. De ruimte bestaat dus uit een afgesloten geheel waarbij de luchtstroming niet door de wanden heen kan. Er wordt uitgegaan van de no-slip conditie bij de wanden wat betekent dat de luchtstroming bij de wanden geen snelheid heeft.

In figuur 2.1 komen de volgende notaties voor:

x de afstand in de x -richting in [m]

y de afstand in de y -richting in [m]

z de afstand in de z -richting in [m]

\mathbf{e}_x de eenheidsvector in de x -richting

\mathbf{e}_y de eenheidsvector in de y -richting

\mathbf{e}_z de eenheidsvector in de z -richting

\mathbf{x} de positievector in [m]

g de versnelling door gravitatie in [m s^{-2}]

In de ruimte bevindt zich atmosferische lucht. Dit betekent dat we te maken hebben met een compressiegevoelig medium. Hierdoor kan de dichtheid van de lucht een andere waarde aannemen. Wanneer dit effect niet wordt meegenomen is het niet mogelijk om een luchtstroming te laten ontstaan door het effect van temperatuurverandering, wat wel gewenst is omdat dit effect in een kas optreedt.

Hoofdstuk 3

Fysische model vergelijkingen

In dit hoofdstuk worden de vergelijkingen opgesteld die nodig zijn om de processen die in een kas plaatsvinden te modelleren. In 3.1 staat welke soort vergelijkingen er worden opgesteld en in welke vorm ze uiteindelijk moeten komen zodat ze geschikt zijn voor numerieke integratie. In 3.2 worden de vergelijkingen daadwerkelijk opgesteld en worden argumenten gegeven voor de benaderingen en verwaarlozingen die worden toegepast. Tenslotte, in 3.3, blijkt dat er nog een vergelijking mist om tot een geschikte totale set vergelijkingen te komen. In dit gedeelte wordt deze vergelijking dan ook opgesteld en toegevoegd.

3.1 Stromingsleer

Stromingsleer is het vakgebied dat de stroming van gassen en vloeistoffen beschrijft door middel van behoudswetten. De behoudswetten die nodig zijn om tot een volledige beschrijving van de processen die plaatsvinden in een kas te komen zijn de volgende: wet van behoud van massa, wet van behoud van impuls en de wet van behoud van energie. Bij deze drie wetten gaat men er van uit dat er een bepaalde hoeveelheid massa is (behoud van massa), in een afgesloten systeem, die in beweging is (behoud van impuls) en daardoor temperatuursveranderingen veroorzaakt (behoud van energie).

Deze wetten kunnen dus worden gebruikt om de processen te beschrijven en tot een model te komen. Om tot een geschikt model te komen heeft het de voorkeur dat de vergelijkingen in toestandsvorm komen te staan. Een toestandsvorm heeft onder andere de volgende voordelen:

1. Modellen in toestandsvorm zijn zeer geschikt voor numerieke integratie.
2. Oorzaak en gevolg zijn duidelijk zichtbaar bij een model in toestandsvorm.
3. Input, toestandsvariabelen en parameters zijn duidelijk te onderscheiden in een toestandsvorm.

Een voorbeeld van een toestandsvorm is als volgt:

$$\begin{aligned}\frac{dx_1}{dt} &= p_1 x_1 + p_2 x_2 \\ \frac{dx_2}{dt} &= p_3 x_1 - p_4 x_2\end{aligned}\tag{3.1}$$

In dit stelsel vergelijkingen zijn twee toestanden te herkennen, namelijk x_1 en x_2 . De parameters p_{1-4} hebben per definitie een constante waarde. De beschrijving hierboven bestaat uit twee 1^e orde differentiaal vergelijkingen en twee toestanden waardoor het aantal vergelijkingen gelijk is aan het aantal toestanden. Voor meer informatie over de toestandsvorm, zie *Lecture Notes of Signals and Systems Modelling: SCO-20306*, [12]. In dit hoofdstuk wordt een toestandsmodel met vijf toestanden opgesteld.

3.2 Behoudswetten

3.2.1 Continuïteitsvergelijking

Indien we te maken hebben met een gesloten systeem, zoals een kas, zal de massa erin constant blijven, ongeacht de processen die in het systeem plaats vinden. Anders gezegd, massa wordt niet gecreëerd of geconsumeerd en blijft dus constant. Dit is het principe van de wet van behoud van massa wat wiskundig wordt beschreven met de continuïteitsvergelijking. Het betreffende medium in een kas, lucht, is compressiegevoelig. Dat wil zeggen dat lucht door verandering van temperatuur een andere dichtheid krijgt. De algemene continuïteitsvergelijking houdt naast het behoud van massa ook rekening met variatie in dichtheid. De algemene continuïteitsvergelijking is als volgt:

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \mathbf{v}) = 0 \quad (3.2)$$

Hierin is:

ρ de massadichtheid in $[\text{kg m}^{-3}]$

\mathbf{v} de vectoriële snelheid in $[\text{m s}^{-1}]$

$\frac{D}{Dt}$ de materiële afgeleide $[\text{s}^{-1}]$

∇ de nabla-operator met formeel de eenheid $[\text{m}^{-1}]$

Voor meer informatie over het opstellen van de algemene continuïteitsvergelijking, zie *Modeling in Materials Processing*, [2].

Vergelijking (3.2) kan verder worden uitgewerkt door beide termen in de vergelijking uit te schrijven. De materiële afgeleide term, $\frac{D\rho}{Dt}$, uit vergelijking (3.2) kan als volgt worden uitgewerkt:

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} \Big|_x + \frac{\partial \rho}{\partial x} \frac{\partial x}{\partial t} \Big|_x + \frac{\partial \rho}{\partial y} \frac{\partial y}{\partial t} \Big|_x + \frac{\partial \rho}{\partial z} \frac{\partial z}{\partial t} \Big|_x = \frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} \quad (3.3)$$

Hierin is:

u de snelheid in de x-richting in $[\text{m s}^{-1}]$

v de snelheid in de y-richting in $[\text{m s}^{-1}]$

w de snelheid in de z-richting in $[\text{m s}^{-1}]$

\mathbf{x} de positievector in $[\text{m}]$

t de tijd in $[\text{s}]$

De nabla-operator term uit de continuïteitsvergelijking (3.2) kan als volgt worden uitgewerkt:

$$\rho(\nabla \cdot \mathbf{v}) = \rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad (3.4)$$

Door nu beide uitgewerkte termen, vergelijking (3.3) en (3.4) te substitueren in de algemene continuïteitsvergelijking, vergelijking (3.2), komt deze er als volgt uit te zien:

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} + \rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0 \quad (3.5)$$

3.2.2 Impulsvergelijking

Als op een bepaalde plaats in de kas de temperatuur van de lucht warmer is dan op andere plaatsen zal er een luchtstroming ontstaan wat er toe leidt dat de temperatuur na verloop van tijd weer overal in de kas gelijk wordt. Om een dergelijke stroming te kunnen modelleren zijn vergelijkingen nodig die de snelheden in de tijd beschrijven. Dergelijke vergelijkingen worden verkregen met de impulsvergelijking. De impulsvergelijking is een partiële differentiaal vergelijking die het behoud van impuls beschrijft in de tijd en de ruimte. Voor meer informatie over het opstellen van de impulsvergelijking, zie *Modeling in Materials Processing*, [2].

Omdat er nog geen aannames zijn gemaakt wordt er gestart met de algemene vorm van de impulsvergelijking:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{b} \quad (3.6)$$

Hierin is:

p de hydrostatische druk (*hydrostatic pressure*) in $[\text{N m}^{-2}]$

$\boldsymbol{\tau}$ de schuifspanningstensor (zie vergelijking (3.10))

\mathbf{b} de body kracht vector die aangrijpt op coördinaatpunt \mathbf{x}

Deze impulsvergelijking kan vervolgens worden uitgewerkt in alle drie de richtingen x , y en z . In dit rapport wordt de uitwerking in de z -richting met al zijn aannames in volledigheid weergegeven. De overige uitwerkingen in de richtingen x en y verlopen op dezelfde manier en zullen daarom niet nader worden beschouwd. Wanneer de termen van de impulsvergelijking in de z -richting worden uitgewerkt, zonder verdere aannames, ziet deze er als volgt uit:

$$\rho \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{\partial p}{\partial z} + \nabla \cdot \boldsymbol{\tau} + \rho b_z \quad (3.7)$$

met:

$$\rho b_z = -\rho g \mathbf{e}_z \quad (3.8)$$

met hierin:

g de versnelling door gravitatie in $[m\ s^{-2}]$

\mathbf{e}_z de eenheidsvector in de z -richting

Het minteken in vergelijking (3.8) ontstaat doordat de gravitatie richting tegen de richting van de eenheidsvector \mathbf{e}_z in is gekozen.

Indien een medium stroomt kunnen er schuifspanningen optreden als gevolg van deeltjes binnen het medium die langs elkaar stromen. Vandaar dat in vergelijking (3.7) de schuifspanningstensor, $\boldsymbol{\tau}$, in de impulsvergelijking aanwezig is. De schuifspanningstensor kan als volgt worden uitgewerkt:

$$\boldsymbol{\tau} = 2\mu\mathbf{D} + \lambda(\text{tr}\mathbf{D})\mathbf{I} \quad (3.9)$$

met:

μ de dynamische viscositeit in $[kg\ m^{-1}\ s^{-1}]$

λ de dilatationele viscositeit in $[kg\ m^{-1}\ s^{-1}]$

en:

$$\mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T), \quad \mathbf{L} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} \quad \text{en} \quad \mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Als vergelijking (3.8) wordt uitgewerkt, wordt $\boldsymbol{\tau}$ gelijk aan:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} = \begin{bmatrix} 2\mu \frac{\partial u}{\partial x} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) & \mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & \mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \\ \mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) & 2\mu \frac{\partial v}{\partial y} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) & \mu \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) \\ \mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) & \mu \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right) & 2\mu \frac{\partial w}{\partial z} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \end{bmatrix} \quad (3.10)$$

Nu is echter bekend dat de dilatationele viscositeit, λ , een slecht meetbare eigenschap is. Toch kan deze niet worden verwaarloosd omdat het hier gaat om een stroming met een

variabele dichtheid waar de dilatationele viscositeit invloed op heeft. Een algemene benadering voor λ is daarvoor:

$$\lambda = -\frac{2}{3}\mu \quad (3.11)$$

Dit wordt de Stokes Hypothesis genoemd. Voor meer informatie hierover, zie *Modeling in Materials Processing*, [2].

Wanneer de schuifspanningstensor, vergelijking (3.10) wordt gebruikt samen met de Stokes Hypothesis, vergelijking (3.11), komt vergelijking (3.7) er als volgt uit te zien:

$$\rho \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{4}{3} \frac{\partial^2 w}{\partial z^2} \right) - \rho g \quad (3.12)$$

Als laatste stap worden ook de impulsvergelijking voor de x - en de y -richting opgesteld. Het enige verschil met de z -richting is dat in deze vergelijkingen geen gravitatie term zit. Dit komt omdat de gravitatie werking alleen van invloed is op de hoogte. De impulsvergelijkingen in de x - en de y -richting zijn als volgt:

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{\partial p}{\partial x} + \mu \left(\frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (3.13)$$

$$\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = -\frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{4}{3} \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \quad (3.14)$$

3.2.3 Energievergelijking

Afhankelijk van de begintemperatuur en de stromingen die ontstaan in een kas verandert de temperatuur op elke plaats in de kas op een verschillende manier. Wanneer bijvoorbeeld onderin de kas een iets hogere temperatuur is dan bovenin, zal er een stroming naar boven ontstaan die ervoor zorgt dat eerst aanliggende lagen worden verwarmd en zo geleidelijk aan weer lagen daarboven. De beschrijving van dit temperatuursveld wordt gedaan met de energievergelijking. De energievergelijking is een partiële differentiaal vergelijking die het behoud van energie beschrijft. Voor meer informatie over het opstellen van de energievergelijking, zie *Modeling in Materials Processing*, [2].

Omdat er nog geen aannames zijn gemaakt wordt er gestart met de algemene vorm van de energievergelijking:

$$\begin{aligned} \rho c_p \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) + \rho L_p \frac{Dp}{Dt} = & - \left[\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z} \right] + \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} \\ & + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\} + \rho \dot{R} \end{aligned} \quad (3.15)$$

Hierin is:

c_p de soortelijke warmte in [$\text{J kg}^{-1} \text{K}^{-1}$]

L_p de latent heat of pressure change in [$\text{m}^3 \text{kg}^{-1}$]

q_i de warmte flux in de i -richting in [$\text{N m}^{-1} \text{s}^{-1}$]

\dot{R} de interne warmtebron [$\text{J kg}^{-1} \text{s}^{-1}$]

T de temperatuur in [K]

De τ_{ij} termen kunnen uit de schuifspanningstensor, vergelijking (3.10), worden gehaald. Deze τ_{ij} termen worden hier echter niet in gesubstitueerd omdat de vergelijking dan nog langer wordt en daardoor onoverzichtelijk.

Van vergelijking (3.15) blijven alle termen staan, behalve de interne warmtebron, $\rho\dot{R}$. Een interne warmtebron is bijvoorbeeld een hitteplaat middenin de kas wat een aantal J per kg per seconde afgeeft. Er vanuit gaande dat een dergelijke warmtebron niet aanwezig is kan deze term worden verwaarloosd.

De materiële afgeleide term, $\frac{Dp}{Dt}$, uit vergelijking (3.15) kan als volgt worden uitgewerkt:

$$\frac{Dp}{Dt} = \frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} + w \frac{\partial p}{\partial z} \quad (3.16)$$

Omdat de warmte flux, q_i , evenredig is met de temperatuur, kan de warmte flux worden gesubstitueerd door de warmtegeleidingcoëfficiënt, k , die invloed heeft op de temperatuurgradiënt, ∇T :

$$\mathbf{q} = -\mathbf{k} \cdot \nabla T \quad (3.17)$$

De term L_p in deze vergelijking is om te schrijven tot een functie waarbij L_p uitgedrukt wordt in de compressibiliteit factor Z en de dichtheid ρ :

$$L_p = \frac{1}{Z \cdot \rho} \quad (3.18)$$

Hierin is:

Z de compressibiliteit factor in [-]

De waarde van Z is bij een temperatuur van tussen de 10 °C en de 30 °C gelijk aan $0.99977 \approx 1$. Dus voor L_p kan de volgende vergelijking worden opgesteld:

$$L_p = \frac{1}{\rho} \quad (3.19)$$

Wanneer vervolgens de vergelijkingen (3.16), (3.17) en (3.19) worden gesubstitueerd in vergelijking (3.15) komt deze er als volgt uit te zien:

$$\rho c_p \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) + \left(\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} + w \frac{\partial p}{\partial z} \right) =$$

$$k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] + \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\}$$
(3.20)

3.3 Opstellen van een extra vergelijking

In paragraaf 3.2 zijn de drie verschillende soorten vergelijkingen opgesteld waarmee verschillende soorten processen in de kas beschreven worden. De continuïteitsvergelijking, vergelijking (3.5) beschrijft het behoud van de massa. De impulsvergelijkingen, vergelijkingen (3.12, 3.13 en 3.14), beschrijven het snelheidsveld. De energievergelijking, vergelijking (3.20), beschrijft het temperatuursveld. Al deze vergelijkingen zijn hieronder nogmaals weergegeven:

Continuïteitsvergelijking:

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} + \rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0$$
(3.21a)

Impulsvergelijkingen:

$$\rho \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{4}{3} \frac{\partial^2 w}{\partial z^2} \right) - \rho g$$
(z-richting)

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{\partial p}{\partial x} + \mu \left(\frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right)$$
(x-richting) (3.21b)

$$\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = -\frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{4}{3} \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right)$$
(y-richting)

Energievergelijking:

$$\rho c_p \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) + \left(\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} + v \frac{\partial p}{\partial y} + w \frac{\partial p}{\partial z} \right) =$$

$$k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] + \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\}$$
(3.21c)

In paragraaf 3.1 is uitgelegd dat er wordt gezocht naar een stelsel vergelijkingen die kunnen worden geschreven in toestandsvorm. In de toestandsvorm zijn er verschillende toestanden (*states*) die worden uitgedrukt in evenveel vergelijkingen. Toestanden zijn variabelen waarin afgeleiden naar de tijd voorkomen. Als gekeken wordt naar de vergelijkingen (3.21) is te zien dat er zes toestanden aanwezig zijn, namelijk: ρ , u , v , w , p en T . Ook is te zien in de vergelijkingen (3.21) dat er vijf vergelijkingen zijn. Dit betekent, om tot een toestandsvorm te komen, dat er een extra vergelijking moet worden opgesteld om het aantal vergelijkingen gelijk te maken aan het aantal toestanden.

3.3.1 Dichtheid als functie van de temperatuur

De extra vergelijking die wordt opgesteld is de vergelijking van de dichtheid. In die vergelijking is de dichtheid afhankelijk van de temperatuur. Dit betekent dat de variabelen ρ en T worden gebruikt voor het realiseren van de extra vergelijking en dat er dus geen toestanden bijkomen. De vergelijking is als volgt:

$$\rho(T) = \rho_0 - \rho_0 \beta (T - T_0) \quad (3.22)$$

Hierin is:

ρ_0 de referentiedichtheid in $[\text{kg m}^{-3}]$

T_0 de referentietemperatuur in $[\text{K}]$

$$\beta = - \frac{1}{\rho_0} \left. \frac{\partial \rho}{\partial T} \right|_{\rho_0, T_0} = 0.0034 \quad [\text{K}^{-1}]$$

De referentiedichtheid, ρ_0 , is de gemiddelde dichtheid waarover de lucht varieert. Evenzo is de referentietemperatuur, T_0 , de gemiddelde temperatuur waarover de lucht varieert. Hoe aan deze vergelijking wordt gekomen wordt duidelijk uitgelegd in *Modeling in Materials Processing*, [2]. Daar wordt uitgelegd dat er een Taylor serie wordt opgesteld om de referentie punten ρ_0 en T_0 . Met het argument dat het bij modelleren gaat om kleine veranderingen in de temperatuur en dichtheid, kunnen de hogere orde termen uit de Taylor reeks worden verwaarloosd en blijft daarmee vergelijking (3.22) over. Wanneer $(T - T_0) < 10$ °C is de verandering in de dichtheid klein ten opzichte van de referentiedichtheid ρ_0 .

Vergelijking (3.22) kan worden gesubstitueerd in elk van de vergelijkingen (3.21). Hoe dit wordt gedaan komt in de paragrafen 3.3.2 tot en met 3.3.4 aan bod.

3.3.2 $\rho(T)$ in continuïteitsvergelijking

Om vergelijking (3.22) te substitueren in de continuïteitsvergelijking (3.21a) wordt de continuïteitsvergelijking eerst herschreven in de volgende vorm:

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} + \rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = \frac{D\rho}{Dt} + \rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0 \quad (3.23)$$

Vergelijking (3.23) heeft nu de materiële afgeleide term $\frac{D\rho}{Dt}$ weer in zich zoals ook in vergelijking (3.2). Deze term moet namelijk vervangen worden door de dichtheidsvergelijking (3.22) nadat hiervan de materiële afgeleide is bepaald:

$$\frac{D\rho}{Dt} = -\rho_0 \beta \frac{DT}{Dt} \quad (3.24)$$

Nu blijkt dat in vergelijking (3.24) een materiële temperatuur afgeleide is ontstaan, $\frac{DT}{Dt}$. De term $\frac{DT}{Dt}$ kan als volgt worden uitgewerkt:

$$\frac{DT}{Dt} = \frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \quad (3.25)$$

en wordt vergelijking (3.24) als volgt:

$$\frac{D\rho}{Dt} = -\rho_0 \beta \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) \quad (3.26)$$

Wanneer de vergelijkingen (3.26) en (3.22) worden gesubstitueerd in de continuïteitsvergelijking (3.23) wordt deze als volgt:

$$-\rho_0 \beta \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) + [\rho_0 - \rho_0 \beta (T - T_0)] \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0 \quad (3.27)$$

3.3.3 $\rho(T)$ in impulsvergelijkingen

Wanneer vergelijking (3.22) wordt gesubstitueerd in de impulsvergelijking in de z -richting, vergelijking (3.12), wordt deze als volgt:

$$[\rho_0 - \rho_0 \beta (T - T_0)] \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{4}{3} \frac{\partial^2 w}{\partial z^2} \right) + \rho_0 \beta (T - T_0) g - \rho_0 g \quad (3.28)$$

De laatste twee termen beschrijven de invloed van de dichtheid op de gravitatie en daarbij moet er onderscheid worden gemaakt tussen verschillende gravitatie effecten. Er zijn namelijk twee effecten die optreden:

1. Het effect van de gravitatie op de referentiedichtheid: $-\rho_0 g$
2. Het effect van de gravitatie op dichtheidsvariaties: $\rho_0 g \beta (T - T_0)$

Het is eerste effect levert een term, $-\rho_0 g$, die constant is omdat de referentiedichtheid, ρ_0 , constant is. Om deze reden kan de term, $-\rho_0 g$, bij de drukterm, $\frac{\partial p}{\partial z}$, worden gevoegd door het definiëren van een nieuwe variabele genaamd de gemodificeerde druk, \hat{p} :

$$\hat{p} \equiv p + \rho_0 g h \quad (3.29)$$

met hierin:

h de hoogte, gemeten in de z -richting in [m]

Voor meer informatie over de gemodificeerde druk, zie *Modeling in Materials Processing*, [2]. Wanneer de gemodificeerde druk, \hat{p} , wordt ingevuld in vergelijking (3.28) wordt deze:

$$[\rho_0 - \rho_0\beta(T - T_0)] \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{4}{3} \frac{\partial^2 w}{\partial z^2} \right) + \rho_0\beta(T - T_0)g \quad (3.30)$$

Wanneer vergelijking (3.22) en \hat{p} worden gesubstitueerd in de impulsvergelijkingen in de x- en y-richting, vergelijkingen (3.13 en 3.14) worden deze als volgt:

$$[\rho_0 - \rho_0\beta(T - T_0)] \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial x} + \mu \left(\frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (3.31)$$

$$[\rho_0 - \rho_0\beta(T - T_0)] \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{4}{3} \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \quad (3.32)$$

3.3.4 $\rho(T)$ in energievergelijking

Als laatste kunnen vergelijking (3.22) en \hat{p} worden gesubstitueerd in de energievergelijking (3.21c) en wordt deze als volgt:

$$[\rho_0 - \rho_0\beta(T - T_0)] c_p \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) + \left(\frac{\partial \hat{p}}{\partial t} + u \frac{\partial \hat{p}}{\partial x} + v \frac{\partial \hat{p}}{\partial y} + w \frac{\partial \hat{p}}{\partial z} \right) = k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] + \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\} \quad (3.33)$$

3.3.5 Model met vijf toestandsvariabelen

De drie soorten vergelijkingen, de continuïteitsvergelijking (3.27), de impulsvergelijkingen (3.30, 3.31 en 3.32) en de energievergelijking (3.33) zijn dus:

Continuïteitsvergelijking:

$$-\rho_0\beta \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) + \rho(T) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0 \quad (3.34a)$$

Impulsvergelijkingen:

$$\begin{aligned} z: \rho(T) \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) &= -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{4}{3} \frac{\partial^2 w}{\partial z^2} \right) + \rho_0\beta(T - T_0)g \\ x: \rho(T) \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) &= -\frac{\partial \hat{p}}{\partial x} + \mu \left(\frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \\ y: \rho(T) \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) &= -\frac{\partial \hat{p}}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{4}{3} \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \end{aligned} \quad (3.34b)$$

Energievergelijking:

$$\rho(T)c_p \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) + \left(\frac{\partial \hat{p}}{\partial t} + u \frac{\partial \hat{p}}{\partial x} + v \frac{\partial \hat{p}}{\partial y} + w \frac{\partial \hat{p}}{\partial z} \right) =$$

$$k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] + \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\}$$

(3.34c)

met:

$$\rho(T) = \rho_0 - \rho_0 \beta (T - T_0)$$

Als nu gekeken wordt naar de vergelijkingen (3.34) is te zien dat er geen zes, maar vijf toestanden aanwezig zijn, namelijk: u , v , w , \hat{p} en T . Dit komt omdat de extra vergelijking (3.22) van de dichtheid, $\rho(T)$, is gesubstitueerd in vergelijkingen (3.21) waardoor vergelijkingen (3.34) ontstonden. Hierdoor is de toestand ρ komen te vervallen. Dit betekent dat er evenveel toestanden als vergelijkingen zijn en dat deze set vergelijkingen in toestandsvorm kan worden geschreven. Hoe dit wordt gedaan wordt uitgelegd in hoofdstuk 4.

Hoofdstuk 4

Toestandsvorm

In dit hoofdstuk worden de vergelijkingen die zijn opgesteld in hoofdstuk 4 zodanig verwerkt dat ze gebruikt kunnen worden voor numerieke integratie. Allereerst wordt in 4.1 het stelsel vergelijkingen omgeschreven naar de toestandsvorm. Aan het einde van 4.1 blijkt dat het stelsel vergelijkingen niet geschikt is voor numerieke integratie omdat de oplossing zeer snel divergeert. Om die reden wordt in 4.2 het stelsel vergelijkingen weer aangepast en wordt de Boussinesq benadering geïntroduceerd zodat een nieuwe numerieke oplossingsmethode kan worden gebruikt namelijk het SIMPLE algoritme.

In dit hoofdstuk en de rest van het rapport geldt dat $\rho(T)$ als volgt is geformuleerd:

$$\rho(T) = \rho_0 - \rho_0 \beta (T - T_0)$$

naar vergelijking (3.22) in hoofdstuk 3.

4.1 Omschrijven naar toestandsvorm

In hoofdstuk 4 is een stelsel vergelijkingen opgesteld, vergelijkingen (3.34), die geschreven kan worden in toestandsvorm omdat er evenveel toestanden als vergelijkingen zijn. Zie 3.1 voor meer uitleg over de toestandsvorm. Om tot deze toestandsvorm te komen moet eerst de continuïteitsvergelijking (3.34a) zodanig worden omgeschreven dat deze kan worden gesubstitueerd in de energievergelijking (3.34c). Wanneer vergelijking (3.34a) wordt omgeschreven ziet deze er als volgt uit:

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} = \frac{\rho_0 \beta}{\rho(T)} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad (4.1)$$

Deze vergelijking (4.1) wordt vervolgens gesubstitueerd in de energievergelijking (3.34c):

$$\begin{aligned}
& [\rho_0 - \rho_0 \beta(T - T_0)] c_p \left(\frac{\rho_0 \beta}{\rho(T)} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right) + \left(\frac{\partial \hat{p}}{\partial t} + u \frac{\partial \hat{p}}{\partial x} + v \frac{\partial \hat{p}}{\partial y} + w \frac{\partial \hat{p}}{\partial z} \right) = \\
& k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] + \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\}
\end{aligned} \tag{4.2}$$

Door deze substitutie is de term $\frac{\partial T}{\partial t}$ uit de energievergelijking (3.34c) geëlimineerd en blijft enkel de variabele \hat{p} als tijdsafhankelijke toestand over. Hierdoor kan vergelijking (4.2) als volgt worden geschreven:

$$\begin{aligned}
\frac{\partial \hat{p}}{\partial t} &= \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\} \\
&+ k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] - c_p \frac{(\rho(T))^2}{\rho_0 \beta} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) - \left(u \frac{\partial \hat{p}}{\partial x} + v \frac{\partial \hat{p}}{\partial y} + w \frac{\partial \hat{p}}{\partial z} \right)
\end{aligned} \tag{4.3}$$

Deze vergelijking (4.3) is nu geschreven in de toestandsvorm als uitgelegd in 3.1, vergelijking (3.1). Aan de linkerhand van de vergelijking staat de tijdsafgeleide van \hat{p} . Aan de rechterhand van de vergelijking is geen tijdsafgeleide meer te vinden en zijn alle termen afhankelijk van bekende parameters of de toestanden zelf. Vervolgens kunnen, naast de energievergelijking, ook de continuïteitsvergelijking (3.34a) en de impulsvergelijkingen (3.34b) worden geschreven in toestandsvorm en ontstaat de volgende set vergelijkingen:

Temperatuur:

$$\frac{\partial T}{\partial t} = \left(\frac{1}{\beta} - (T - T_0) \right) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) - \left(u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) \tag{4.4a}$$

Snelheden:

$$\begin{aligned}
\frac{\partial w}{\partial t} &= \frac{\mu}{\rho(T)} \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{4}{3} \frac{\partial^2 w}{\partial z^2} \right) - \left(u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) - \frac{1}{\rho(T)} \frac{\partial \hat{p}}{\partial z} + \frac{\rho_0 \beta (T - T_0)}{\rho(T)} g \\
\frac{\partial u}{\partial t} &= \frac{\mu}{\rho(T)} \left(\frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) - \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) - \frac{1}{\rho(T)} \frac{\partial \hat{p}}{\partial x} \\
\frac{\partial v}{\partial t} &= \frac{\mu}{\rho(T)} \left(\frac{\partial^2 v}{\partial x^2} + \frac{4}{3} \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) - \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) - \frac{1}{\rho(T)} \frac{\partial \hat{p}}{\partial y}
\end{aligned} \tag{4.4b}$$

Druk:

$$\begin{aligned}
\frac{\partial \hat{p}}{\partial t} &= \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\} \\
&+ k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] - c_p \frac{(\rho(T))^2}{\rho_0 \beta} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) - \left(u \frac{\partial \hat{p}}{\partial x} + v \frac{\partial \hat{p}}{\partial y} + w \frac{\partial \hat{p}}{\partial z} \right)
\end{aligned} \tag{4.4c}$$

met:

$$\rho(T) = \rho_0 - \rho_0\beta(T - T_0)$$

Nu de vergelijkingen in de toestandsvorm staan, kunnen de waarden voor de parameters worden ingevuld:

$$c_p = 1012 \text{ [J kg}^{-1} \text{ K}^{-1}\text{]}$$

$$k = 0.0257 \text{ [W m}^{-1} \text{ K}^{-1}\text{]}$$

$$\beta = 0.0034 \text{ [K}^{-1}\text{]}$$

$$\rho(T) = \rho_0 = 1.205 \text{ [kg m}^{-3}\text{]}$$

$$\mu = 1.79 \cdot 10^{-5} \text{ [kg m}^{-1} \text{ s}^{-1}\text{]}$$

$$T_0 = 20 \text{ [}^\circ\text{C]}$$

Deze parameters komen uit *Tables of Thermal Properties of Gases* [3].

Voor $T = 20 \text{ }^\circ\text{C}$ wordt het stelsel (4.4) gelijk aan:

Temperatuur:

$$\frac{\partial T}{\partial t} = 294.12 \cdot \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) - \left(u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} \right) \quad (4.5a)$$

Snelheden:

$$\begin{aligned} \frac{\partial w}{\partial t} &= 1.53 \cdot 10^{-5} \cdot \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{4}{3} \frac{\partial^2 w}{\partial z^2} \right) - \left(u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) - 0.83 \frac{\partial \hat{p}}{\partial z} + \frac{\rho_0 \beta (T - T_0)}{\rho(T)} g \\ \frac{\partial u}{\partial t} &= 1.53 \cdot 10^{-5} \cdot \left(\frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) - \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) - 0.83 \frac{\partial \hat{p}}{\partial x} \\ \frac{\partial v}{\partial t} &= 1.53 \cdot 10^{-5} \cdot \left(\frac{\partial^2 v}{\partial x^2} + \frac{4}{3} \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) - \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) - 0.83 \frac{\partial \hat{p}}{\partial y} \end{aligned} \quad (4.5b)$$

Druk:

$$\begin{aligned} \frac{\partial \hat{p}}{\partial t} &= \left\{ \tau_{xx} \frac{\partial u}{\partial x} + \tau_{yy} \frac{\partial v}{\partial y} + \tau_{zz} \frac{\partial w}{\partial z} \right\} + \left\{ \tau_{xy} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) + \tau_{yz} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) + \tau_{zx} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right\} \\ &+ 0.0257 \cdot \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] - 3.6 \cdot 10^5 \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) - \left(u \frac{\partial \hat{p}}{\partial x} + v \frac{\partial \hat{p}}{\partial y} + w \frac{\partial \hat{p}}{\partial z} \right) \end{aligned} \quad (4.5c)$$

Voor $T = 20 \text{ }^\circ\text{C}$ wordt duidelijk dat het stelsel vergelijkingen (4.5) stijf is vanwege de constante waarde $3.6 \cdot 10^5$ in (4.5c) en de constanten in vergelijkingen (4.5b) van $1.53 \cdot 10^{-5}$. Dit geeft aan dat de druk \hat{p} zeer snel verandert ten opzichte van de andere toestanden. Wanneer de vergelijkingen numeriek worden geïntegreerd zal de oplossing daardoor zeer snel divergeren.

4.2 Numerieke oplossingsmethode

In de vorige paragraaf is gebleken dat het stelsel vergelijkingen (4.5) stijf is. Dit betekent niet dat het stelsel vergelijkingen (4.5) fysisch gezien niet klopt, maar dat de beoogde numerieke oplossingsmethode niet toepasbaar is op (4.5) omdat de oplossing snel divergeert. Om toch tot een geschikt dynamisch model te komen zijn er vanaf hier twee mogelijkheden:

1. Het quasi-statisch maken van het stelsel vergelijkingen (4.5) en vervolgens het numeriek integreren van het stelsel vergelijkingen. (Voor meer informatie over het quasi-statisch maken van vergelijkingen zie bijlage 10.2).
2. Het vinden van een andere oplossingsmethode. Deze oplossingsmethode kan worden gevonden in het SIMPLE algoritme, zie hoofdstuk 6. Het SIMPLE algoritme kan echter niet worden toegepast op dit stelsel vergelijkingen (4.5) en er zal daarvoor worden teruggevallen worden op de basis waarmee begonnen is, vergelijkingen (3.21).

In dit onderzoek is gekozen voor optie twee, het vinden van een andere oplossingsmethode. De reden hiervan is dat het toepassen van het SIMPLE algoritme een methode is die in de literatuur vaak wordt gebruikt, zie [5] en [13]. Wanneer deze methode wordt gebruikt wordt duidelijk wat de structuur is van een dergelijk algoritme wat de toepassing van bijvoorbeeld geavanceerde regeltechniek mogelijk kan maken. Daarnaast is het uiteindelijk ontwikkelde model een open model waardoor er geen beperkingen zijn aan het eventueel combineren van dit model met andere modellen. Optie één is uiteraard nog steeds een mogelijkheid maar valt buiten dit onderzoek.

4.2.1 Boussinesq benadering

Het stelsel vergelijkingen (3.21) is de basis waar mee verder wordt gegaan. Het stelsel vergelijkingen (3.21) geeft weer dat de dichtheid ρ variabel is, wat immers wordt beschreven met de continuïteitsvergelijking (3.21a). Dit betekent dat de dichtheid in elk van de vergelijkingen als variabele wordt beschouwd en niet als constante. Wat hier staat kan echter versimpeld worden omdat de dichtheidsvariatie in elk van de termen verwaarloosbaar klein is behalve bij de term waarmee het met de zwaartekracht g wordt vermenigvuldigd. Dit is de Boussinesq benadering. Deze benadering wordt vaak gebruikt bij het modelleren van luchtcirculaties in gesloten systemen en door het SIMPLE algoritme. Wanneer de benadering wordt toegepast betekent het dat de vergelijkingen (3.21) veranderen op vier punten:

1. Bij alle termen waarin dichtheid ρ voorkomt, wordt deze vervangen door de referentiedichtheid, ρ_0 , behalve voor gravitatie termen.
2. Omdat er nu kan worden uitgegaan van niet alleen een Newtons medium, maar ook een dichtheidsonafhankelijk medium, kunnen de visceuze termen worden vereenvoudigd. De visceuze termen zijn de termen met de dynamische viscositeit, μ , ervoor.
3. Om de dichtheid ρ voor de gravitatie term variabel te laten blijven, wordt hier de vergelijking van de dichtheid, vergelijking (3.22), gesubstitueerd. Uiteraard geldt dan weer dat de druk aangepast moet worden in de gemodificeerde druk, zie 3.3.3.
4. De drukvariatie term in de energievergelijking is verwaarloosbaar omdat deze term alleen geldt wanneer de dichtheid variabel is.
5. De continuïteitsvergelijking (3.21a) verandert omdat de dichtheidsvariatie wordt verwaarloosd. De continuïteitsvergelijking is dan niet meer afhankelijk van de dichtheid en kan vereenvoudigd worden tot:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (4.6)$$

Uiteindelijk komen de vergelijkingen (3.21) er als volgt uit te zien:

Continuïteitsvergelijking:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (4.7a)$$

Impulsvergelijkingen:

$$\begin{aligned} z: \rho_0 \left(\frac{\partial w}{\partial t} + \frac{\partial uw}{\partial x} + \frac{\partial vw}{\partial y} + \frac{\partial ww}{\partial z} \right) &= -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + \rho_0 \beta (T - T_0) g \\ x: \rho_0 \left(\frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial vu}{\partial y} + \frac{\partial wu}{\partial z} \right) &= -\frac{\partial \hat{p}}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \\ y: \rho_0 \left(\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial vv}{\partial y} + \frac{\partial wv}{\partial z} \right) &= -\frac{\partial \hat{p}}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \end{aligned} \quad (4.7b)$$

Energievergelijking:

$$\begin{aligned} \rho_0 c_p \left(\frac{\partial T}{\partial t} + \frac{\partial uT}{\partial x} + \frac{\partial vT}{\partial y} + \frac{\partial wT}{\partial z} \right) &= k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right] + 2\mu \left\{ \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right\} \\ + \mu \left\{ \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right)^2 \right\} \end{aligned} \quad (4.7c)$$

In deze nieuwe set vergelijkingen wordt nog steeds uitgegaan van een compressiegevoelig medium omdat de dichtheid op het belangrijke punt in de beschrijving variabel is, namelijk bij de gravitatieterm g in (4.7b).

Wat verder opvalt in (4.7) is dat de snelheden die eerst voor de afgeleide termen stonden nu binnen de termen staan. Waarom dit kan wordt uitgelegd in bijlage 10.3.

Hoofdstuk 5

Numeriek oplossen van de vergelijkingen

In dit hoofdstuk worden de vergelijkingen die in hoofdstuk 5 zijn opgesteld ruimtelijk en in de tijd gediscretiseerd. Het bijbehorende grid wordt geïntroduceerd in 5.1. In 5.2 wordt uitgelegd naar welke oplossingsmethode wordt toegewerkt wat tot gevolg heeft dat de vergelijkingen nog moeten worden aangepast. In 5.3, worden de vergelijkingen gediscretiseerd. Voor de impulsvergelijkingen en de continuïteitsvergelijking gebeurt dit middels centrale differenties. Voor de temperatuursvergelijking is deze benadering te grof omdat de richtingen van de stroming daar niet mee kunnen worden bepaald. Daarom wordt gekozen voor het ‘*upwind differencing scheme*’.

5.1 Ruimtelijke discretisatie

In hoofdstuk 5 is een stelsel vergelijkingen opgesteld, vergelijkingen (4.7). Op dit stelsel vergelijkingen wordt uiteindelijk in hoofdstuk 7 het SIMPLE algoritme toegepast. Voordat dit algoritme kan worden toegepast moeten de vergelijkingen worden gediscretiseerd. Om de discretisatie methode uit te leggen en toe te passen wordt uitgegaan van een 2D model. Hierbij is gekozen om de x -richting en de z -richting te behouden. De z -richting kan niet worden verworpen omdat in deze vergelijking de dichtheidsvariatie aanwezig is, zie vergelijking (4.7b). Wanneer de set vergelijkingen (4.7) wordt vereenvoudigd naar 2D ziet deze er als volgt uit:

Continuïteitsvergelijking:

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \quad (5.1a)$$

Impulsvergelijkingen:

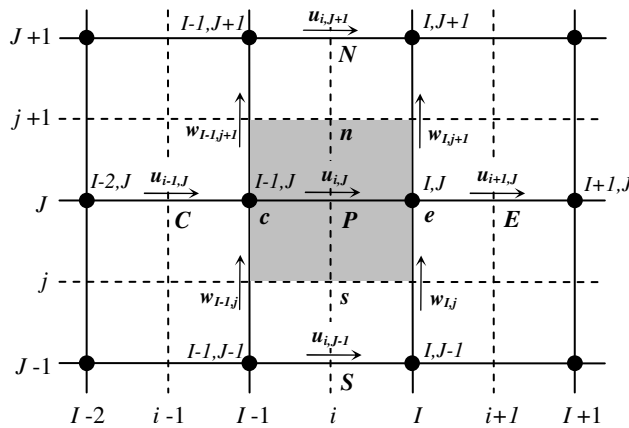
$$x: \rho_0 \left(\frac{\partial u}{\partial t} + \frac{\partial(uu)}{\partial x} + \frac{\partial(wu)}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (5.1b)$$

$$z: \rho_0 \left(\frac{\partial w}{\partial t} + \frac{\partial(uw)}{\partial x} + \frac{\partial(ww)}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial z^2} \right) + \rho_0 \beta (T - T_0) g \quad (5.1c)$$

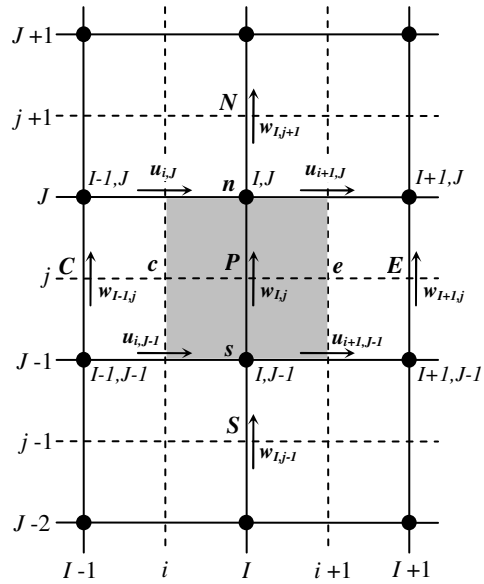
Energievergelijking:

$$\rho_0 c_p \left(\frac{\partial T}{\partial t} + \frac{\partial(uT)}{\partial x} + \frac{\partial(wT)}{\partial z} \right) = k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} \right] + 2\mu \left\{ \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right\} + \mu \left\{ \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right)^2 \right\} \quad (5.1d)$$

De discretisatie methode is de 'staggered grid' centrale differentie methode, zie *An introduction to Computational Fluid Dynamics*, [1]. Het grid wat daarbij wordt gebruikt is te zien in de volgende figuren:



Figuur 5.1a: Staggered grid x-richting



Figuur 5.1b: Staggered grid y-richting

Figuur 5.1a laat het staggered grid zien voor het uitrekenen van het snelheidsveld in de x -richting. Figuur 5.1b laat het grid zien voor het uitrekenen van het snelheidsveld in de y -richting. Uiteraard gaan beide figuren over een en hetzelfde grid en is dit onderscheid enkel gemaakt om duidelijkheid te creëren tussen de te berekenen richtingen x en y . Als goed wordt gekeken naar de figuren, laten ze zien dat de indices I, i de gridverdeling weergeven in de x -richting. De indices J, j geven de gridverdeling weer in de z -richting. De onderverdeling I, i en J, j wordt gemaakt om de volgende reden: vectoren, zoals de snelheid, worden op de plaatsen I, J in het grid uitgerekend en scalair variabelen zoals de druk op i, j . In figuur 5.1a is te zien dat de snelheden, $u_{i,j}$ etc., altijd tussen de zwarte punten in liggen. Deze methode zorgt uiteindelijk voor een hogere stabiliteit bij het numeriek rekenen. Het controle volume dat wordt gebruikt voor de vector variabelen, het vectoriële controlevolume, heeft een grootte die loopt van $I-1$ tot I en $J-1$ tot J en is het donkergekleurde vierkantje in figuur 5.1a. Het controle volume voor de scalar variabelen, het scalaire controlevolume, heeft een grootte die loopt van i tot $i+1$ en j tot $j+1$. Deze is niet te zien in figuur 5.1a. Aangezien het vectoriële controlevolume even groot is als het scalaire controlevolume wordt er voortaan geen onderscheid meer gemaakt tussen de volumes en wordt er steeds uitgegaan van één volume ongeacht waar het zich bevindt. Het grid voor het uitrekenen van de continuïteitsvergelijking en de energievergelijking ziet er als volgt uit:

- Wanneer het algoritme zich bevindt aan het begin van de eerste iteratie, wordt voor deze variabele een beginschatting genomen.
- Wanneer het algoritme zich bevindt midden in een iteratie, wordt voor deze variabele de berekende snelheid van de vorige iteratiestap genomen.
- Wanneer het algoritme zich bevindt aan de start van een nieuwe iteratie, wordt voor deze variabele de laatst berekende snelheid uit de vorige tijd stap.

Wanneer de parameters (5.7a en 5.7b) ingevuld worden in vergelijkingen (5.1b en c), worden deze als volgt:

$$\rho_0 \left(\frac{\partial u}{\partial t} + \frac{\partial(u^{(n-1)}u)}{\partial x} + \frac{\partial(w^{(n-1)}u)}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right) \quad (5.8)$$

$$\rho_0 \left(\frac{\partial w}{\partial t} + \frac{\partial(u^{(n-1)}w)}{\partial x} + \frac{\partial(w^{(n-1)}w)}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial z^2} \right) + \rho_0 \beta (T - T_0) g \quad (5.9)$$

Door de substitutie van (5.7) ontstaat er een stelsel vergelijkingen wat lineair is. Dit stelsel vergelijkingen kan vervolgens met bovenstaande matrixmethode worden opgelost, zie 5.5.1.

5.3 Discretiseren van de vergelijkingen

5.3.1 Discretisatie van de impulsvergelijkingen

In vergelijkingen (5.8) en (5.9) zitten eerste en tweede orde afgeleiden. Deze kunnen worden benaderd met de centrale differentie methode. Als voorbeeld wordt de differentie methode toegepast op de tweede term uit vergelijking (5.8). Deze term is als volgt:

$$f'(u) = \frac{\partial(u^{(n-1)}u)}{\partial x} \quad (5.10)$$

In deze term komt de variabele u en de reeds bekende variabele $u^{(n-1)}$ voor. Hiervan worden de eerste orde afgeleide over x bepaald. Deze zal moeten worden benaderd met de differentie methode. Daarvoor moeten de punten worden bepaald uit het grid waarover wordt gedifferentieerd, zie figuur 5.1a. Punt P wordt beschouwd als het middelste punt en de omliggende punten zijn de punten c en e . De afstand waarover wordt gedifferentieerd is gelijk aan δx wat gelijk is aan de afstand tussen de punten c en e . Wanneer de differentie methode wordt toegepast wordt vergelijking (5.11) als volgt:

$$f'(u_p) = \frac{(u^{(n-1)}u)_e - (u^{(n-1)}u)_c}{\delta x} \quad (5.11)$$

Vervolgens wordt het controlevolume berekend:

$$V = \delta x \cdot \delta y \cdot \delta z \quad (5.12)$$

Waarbij voor δx geldt dat dit de afstand is, zie figuur 5.1a, tussen c en e . Voor δz is het de afstand tussen s en n . Omdat er wordt uitgegaan van een 2D model, wordt δy gelijk gesteld aan 1 en verandert het controlevolume in:

$$V = \delta x \cdot \delta z \quad (5.13)$$

Wanneer vergelijking (5.11) wordt vermenigvuldigd met het controlevolume (5.13) komt deze er als volgt uit te zien:

$$f'(u_p) \cdot V = \frac{(u^{(n-1)}u)_e - (u^{(n-1)}u)_c}{\delta x} \delta x \cdot \delta z = \delta z \cdot (u^{(n-1)}u)_e - \delta z \cdot (u^{(n-1)}u)_c \quad (5.14)$$

De centrale differentie methode wordt op dezelfde manier op alle andere termen uit vergelijking (5.8) toegepast en daarmee komt de vergelijking er als volgt uit te zien:

$$\begin{aligned} & \frac{\rho_0(u_{i,j} - u_{i,j}^0)}{\Delta t} (\delta x \cdot \delta z) + \rho_0 \cdot \delta z \cdot ((u^{(n-1)}u)_e - (u^{(n-1)}u)_c) + \rho_0 \cdot \delta x \cdot ((w^{(n-1)}u)_n - (w^{(n-1)}u)_s) = \\ & \mu \cdot \delta z \cdot \left(\left(\frac{\partial u}{\partial x} \right)_e - \left(\frac{\partial u}{\partial x} \right)_c \right) + \mu \cdot \delta x \cdot \left(\left(\frac{\partial u}{\partial z} \right)_n - \left(\frac{\partial u}{\partial z} \right)_s \right) + (\hat{p}_{I-1,J} - \hat{p}_{I,J}) \cdot \delta z \end{aligned} \quad (5.15)$$

Met hierin:

$u_{i,j}^0$ de snelheid van het vorige tijdstip

Δt de discretisatie tijd

$\hat{p}_{I-1,J}$ de druk op punt c in figuur 5.1a

$\hat{p}_{I,J}$ de druk op punt e in figuur 5.1a

Vervolgens kan vergelijking (5.15) worden herschreven tot een vergelijking waarbij aan de linkerhand onbekende termen staan en aan de rechterhand enkel termen waarvan de waardes bekend zijn.

$$\begin{aligned} & \frac{\rho_0(u_{i,j} - u_{i,j}^0)}{\Delta t} \delta x \cdot \delta z + F_e u_e - F_c u_c + F_n u_n - F_s u_s = \\ & D_e(u_{i+1,j} - u_{i,j}) - D_c(u_{i,j} - u_{i-1,j}) + D_n(u_{i,j+1} - u_{i,j}) - D_s(u_{i,j} - u_{i,j-1}) + (\hat{p}_{I-1,J} - \hat{p}_{I,J}) \cdot \delta z \end{aligned} \quad (5.16)$$

met de constanten:

$$\begin{aligned} F_c &= \rho_0 u_c^{(n-1)} \cdot \delta z & F_e &= \rho_0 u_e^{(n-1)} \cdot \delta z & F_s &= \rho_0 w_s^{(n-1)} \cdot \delta x & F_n &= \rho_0 w_n^{(n-1)} \cdot \delta x \\ D_c &= \frac{\mu}{\delta x_{CP}} \cdot \delta z & D_e &= \frac{\mu}{\delta x_{PE}} \cdot \delta z & D_s &= \frac{\mu}{\delta z_{SP}} \cdot \delta x & D_n &= \frac{\mu}{\delta z_{PN}} \cdot \delta x \end{aligned}$$

In vergelijking (5.16) zijn variabelen u_e , u_c , u_n , en u_s aanwezig op punten waar alleen scalar variabelen zoals de druk worden uitgerekend (zie uitleg over het grid, 5.1). Deze variabelen moeten dus worden benaderd uit omliggende punten wat gebeurt met de centrale differentie benadering. Deze benadering gaat er van uit dat er tussen twee punten een lineair verband ligt. Een tussenliggend punt wordt dan bepaald door het gemiddelde te nemen van de omliggende punten. Met de toepassing van de centrale differentie benadering worden de variabelen als volgt:

$$\begin{aligned}
u_e &= \frac{u_{i,J} + u_{i+1,J}}{2} \quad (a) & u_c &= \frac{u_{i-1,J} + u_{i,J}}{2} \quad (b) \\
u_n &= \frac{u_{i,J} + u_{i,J+1}}{2} \quad (c) & u_s &= \frac{u_{i,J-1} + u_{i,J}}{2} \quad (d)
\end{aligned} \tag{5.17}$$

De vergelijkingen (5.17 a,b,c,d) kunnen worden ingevuld in vergelijking (5.16) waardoor de volgende vergelijking ontstaat:

$$\begin{aligned}
&\frac{\rho_0 (u_{i,J} - u_{i,J}^0)}{\Delta t} \delta x \cdot \delta z + \frac{1}{2} F_e (u_{i,J} + u_{i+1,J}) - \frac{1}{2} F_c (u_{i-1,J} + u_{i,J}) + \frac{1}{2} F_n (u_{i,J} + u_{i,J+1}) - \frac{1}{2} F_s (u_{i,J-1} + u_{i,J}) = \\
&D_e (u_{i+1,J} - u_{i,J}) - D_c (u_{i,J} - u_{i-1,J}) + D_n (u_{i,J+1} - u_{i,J}) - D_s (u_{i,J} - u_{i,J-1}) + (\hat{p}_{I-1,J} - \hat{p}_{I,J}) \cdot \delta z
\end{aligned} \tag{5.18}$$

De oplossingsmethode is de matrixmethode, zie 5.2.1. Daarom moet vergelijking (5.18) worden herschreven zodanig dat deze oplossingmethode kan worden toegepast:

$$\begin{aligned}
&\left(\rho_0 \frac{\delta x \cdot \delta z}{\Delta t} + \frac{1}{2} F_e - \frac{1}{2} F_c + \frac{1}{2} F_n - \frac{1}{2} F_s + D_e + D_c + D_n + D_s \right) u_{i,J} = \\
&\left(\frac{1}{2} F_c + D_c \right) u_{i-1,J} + \left(-\frac{1}{2} F_e + D_e \right) u_{i+1,J} + \left(\frac{1}{2} F_s + D_s \right) u_{i,J-1} + \left(-\frac{1}{2} F_n + D_n \right) u_{i,J+1} \\
&+ (\hat{p}_{I-1,J} - \hat{p}_{I,J}) \cdot \delta z + \rho_0 \frac{\delta x \cdot \delta z}{\Delta t} u_{i,J}^0
\end{aligned} \tag{5.19}$$

Het betreft dus een vergelijking met vijf onbekenden, waarbij de onbekenden de u variabelen zijn (niet $u_{i,J}^0$) en dienen te worden opgelost. Op dit moment lijken de twee druktermen $\hat{p}_{I-1,J}$ en $\hat{p}_{I,J}$ onbekend, maar in hoofdstuk 7 wordt duidelijk, bij het uitleggen van SIMPLE algoritme, hoe dit wordt opgelost.

Vergelijking (5.19) kan vervolgens worden geschreven in de volgende vorm:

$$a_{i,J} u_{i,J} = \sum a_{nb} u_{nb} + (\hat{p}_{I-1,J} - \hat{p}_{I,J}) \cdot \delta z + \rho_0 \frac{\delta x \cdot \delta z}{\Delta t} u_{i,J}^0 \tag{5.20}$$

met:

$$\begin{aligned}
a_{i,J} &= \rho_0 \frac{\delta x \cdot \delta z}{\Delta t} + \frac{1}{2} F_e - \frac{1}{2} F_c + \frac{1}{2} F_n - \frac{1}{2} F_s + D_e + D_c + D_n + D_s \\
\sum a_{nb} u_{nb} &= \left(\frac{1}{2} F_c + D_c \right) u_{i-1,J} + \left(-\frac{1}{2} F_e + D_e \right) u_{i+1,J} + \left(\frac{1}{2} F_s + D_s \right) u_{i,J-1} + \left(-\frac{1}{2} F_n + D_n \right) u_{i,J+1}
\end{aligned}$$

Uiteraard moet dezelfde discretisatie methode worden toegepast op de impuls vergelijking in de z -richting, vergelijking (5.9). Hierbij wordt figuur (5.1b) gehanteerd om de discretisatie uit te voeren en wordt vergelijking (5.9) als volgt:

$$\begin{aligned}
& \left(\rho_0 \frac{\delta x \cdot \delta z}{\Delta t} + \frac{1}{2} F_e - \frac{1}{2} F_c + \frac{1}{2} F_n - \frac{1}{2} F_s + D_e + D_c + D_n + D_s \right) w_{I,j} = \\
& \left(\frac{1}{2} F_c + D_c \right) w_{I-1,j} + \left(-\frac{1}{2} F_e + D_e \right) w_{I+1,j} + \left(\frac{1}{2} F_s + D_s \right) w_{I,j-1} + \left(-\frac{1}{2} F_n + D_n \right) w_{I,j+1} \quad (5.21) \\
& + (\hat{p}_{I,j-1} - \hat{p}_{I,j}) \cdot \delta x + \rho_0 g \beta (T_{I,j}^0 - T_0) \delta x \cdot \delta z + \rho_0 \frac{\delta x \cdot \delta z}{\Delta t} w_{I,j}^0
\end{aligned}$$

met de constanten:

$$\begin{aligned}
F_c &= \rho_0 u_c^{(n-1)} \cdot \delta z & F_e &= \rho_0 u_e^{(n-1)} \cdot \delta z & F_s &= \rho_0 w_s^{(n-1)} \cdot \delta x & F_n &= \rho_0 w_n^{(n-1)} \cdot \delta x \\
D_c &= \frac{\mu}{\delta x_{CP}} \cdot \delta z & D_e &= \frac{\mu}{\delta x_{PE}} \cdot \delta z & D_s &= \frac{\mu}{\delta z_{SP}} \cdot \delta x & D_n &= \frac{\mu}{\delta z_{PN}} \cdot \delta x
\end{aligned}$$

Vergelijking (5.21) kan vervolgens worden geschreven in de volgende vorm:

$$a_{I,j} w_{I,j} = \sum a_{nb} w_{nb} + (\hat{p}_{I,j-1} - \hat{p}_{I,j}) \cdot \delta x + \rho_0 g \beta (T_{I,j}^0 - T_0) \delta x \cdot \delta z + \rho_0 \frac{\delta x \cdot \delta z}{\Delta t} w_{I,j}^0 \quad (5.22)$$

met:

$$\begin{aligned}
a_{I,j} &= \rho_0 \frac{\delta x \cdot \delta y}{\Delta t} + \frac{1}{2} F_e - \frac{1}{2} F_c + \frac{1}{2} F_n - \frac{1}{2} F_s + D_e + D_c + D_n + D_s \\
\sum a_{nb} w_{nb} &= \left(\frac{1}{2} F_c + D_c \right) w_{I-1,j} + \left(-\frac{1}{2} F_e + D_e \right) w_{I+1,j} + \left(\frac{1}{2} F_s + D_s \right) w_{I,j-1} + \left(-\frac{1}{2} F_n + D_n \right) w_{I,j+1}
\end{aligned}$$

5.3.2 Discretisatie van de continuïteitsvergelijking

Vervolgens moet de continuïteitsvergelijking worden gediscetiseerd. De oorspronkelijke vergelijking is vergelijking (5.1a):

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0$$

Wanneer bij het discretiseren van deze vergelijking het staggered grid, figuur 5.1c, wordt gebruikt komt de vergelijking er als volgt uit te zien:

$$(u_{i+1,j} - u_{i,j}) \cdot \delta z + (w_{I,j+1} - w_{I,j}) \cdot \delta x = 0 \quad (5.23)$$

Hierbij is eveneens gebruik gemaakt van de centrale differentie methode. Er wordt dan gebruik gemaakt van omliggende punten ten opzichte van u_p en w_p , zie figuur 5.1c.

5.3.3 Discretisatie van de energievergelijking

Als laatste moet de energievergelijking, vergelijking (5.1d), worden gediscetiseerd. Bij het discretiseren van de energievergelijking wordt de vergelijking eerst vereenvoudigd door het effect van viskeuze dissipatie te verwaarlozen. Visceuze dissipatie is het effect van opwarming door wrijving. Dit effect zit verwerkt in de termen in de rechterhand van de vergelijking met de dynamische viscositeit μ ervoor. Bij het medium lucht is deze parameter erg klein en daardoor mag gesteld worden dat de visceuze dissipatie verwaarloosbaar is.

Tevens vereenvoudigt dit het discretiseren van de vergelijking. Door de verwaarlozing van de visceuze dissipatie wordt vergelijking (5.1d) als volgt:

$$\rho_0 c_p \left(\frac{\partial T}{\partial t} + \frac{\partial(uT)}{\partial x} + \frac{\partial(wT)}{\partial z} \right) = k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} \right] \quad (5.24)$$

Voor het discretiseren moet het grid van figuur 5.1c worden gebruikt omdat bij dit grid punt **P** ligt op een scalair punt en niet een vectorieel punt. Wanneer de centrale differentie methode wordt toegepast om de energievergelijking te discretiseren komt vergelijking (5.24) er als volgt uit te zien:

$$\rho_0 c_p \frac{\delta x \cdot \delta z}{\Delta t} T_P + F_e T_e - F_c T_c + F_n T_n - F_s T_s = D_e (T_E - T_P) - D_c (T_P - T_C) + D_n (T_N - T_P) - D_s (T_P - T_S) + \rho_0 c_p \frac{\delta x \cdot \delta z}{\Delta t} T_P^0 \quad (5.25)$$

met de constanten:

$$\begin{aligned} F_e &= \rho_0 c_p u_e \cdot \delta z & F_c &= \rho_0 c_p u_c \cdot \delta z & F_n &= \rho_0 c_p w_n \cdot \delta x & F_s &= \rho_0 c_p w_s \cdot \delta x \\ D_e &= \frac{k}{\delta x} \cdot \delta z & D_c &= \frac{k}{\delta x} \cdot \delta z & D_n &= \frac{k}{\delta z} \cdot \delta x & D_s &= \frac{k}{\delta z} \cdot \delta x \end{aligned}$$

In deze gediscetiseerde vergelijking komen nog variabelen voor, de variabelen T_e , T_c , T_n en T_s , die met de centrale differentie benadering moeten worden benaderd. De punten liggen namelijk tussen de scalaire hoofdpunten in (voor de energievergelijking zijn dat de punten **P**, **C**, **E**, **S** en **N**, zie figuur 5.1c). Het gebruiken van de centrale differentie benadering geeft bij de energievergelijking echter problemen omdat de centrale differentie benadering niet de mogelijkheid heeft tot het herkennen van de snelheidsrichtingen ten opzichte van diffusieve termen in de energievergelijking. De diffusieve termen zijn de termen waar de warmtegeleidingcoëfficiënt, k , voor staat. Voor meer informatie hierover, zie *An introduction to Computational Fluid Dynamics [1]*. Om toch de richtingen van de snelheden te laten herkennen zal de centrale differentie benadering moeten worden aangevuld zodat de richtingen wel worden herkend. De nieuwe benadering is het ‘*upwind differencing scheme*.’ Met het toepassen van het ‘*upwind differencing scheme*’ komt de vergelijking er als volgt uit te zien:

$$a_{I,J} T_{I,J} = a_{I-1,J} T_{I-1,J} + a_{I+1,J} T_{I+1,J} + a_{I,J-1} T_{I,J-1} + a_{I,J+1} T_{I,J+1} + \rho_0 c_p \frac{\delta x \cdot \delta z}{\Delta t} T_{I,J}^0 \quad (5.26)$$

met:

$$a_{I,J} = \rho_0 c_p \frac{\delta x \cdot \delta z}{\Delta t} + a_{I-1,J} + a_{I+1,J} + a_{I,J-1} + a_{I,J+1} + (F_e - F_c) + (F_n - F_s)$$

en (zie tabel):

$a_{I-1,J} :$ $F_c > 0$ dan $a_{I-1,J} = D_c + F_c$ $F_c \leq 0$ dan $a_{I-1,J} = D_c$	$a_{I+1,J} :$ $F_e \geq 0$ dan $a_{I+1,J} = D_e$ $F_e < 0$ dan $a_{I+1,J} = D_e - F_e$
$a_{I,J-1} :$ $F_s > 0$ dan $a_{I,J-1} = D_s + F_s$ $F_s \leq 0$ dan $a_{I,J-1} = D_s$	$a_{I,J+1} :$ $F_n \geq 0$ dan $a_{I,J+1} = D_n$ $F_n < 0$ dan $a_{I,J+1} = D_n - F_n$

Tabel 5.1: Waarden voor a_{nb} afhankelijk van F_i

De variabelen F_i in de tabel worden bepaald door de snelheden, zie vergelijking (5.25). Dit betekent dat de waarde a_{nb} afhankelijk is van de richting van de snelheid. Hierdoor is de energievergelijking in staat de richtingen van de stroming mee te nemen in het berekenen van een nieuw temperatuurveld.

5.3.4 Set vergelijkingen

De vier gediscretiseerde vergelijkingen, de impulsvergelijkingen (5.20 en 5.22) de continuïteitsvergelijking (5.23), en de energievergelijking (5.26) zijn dus:

Impulsvergelijkingen:

$$a_{i,j} u_{i,j} = \sum a_{nb} u_{nb} + (\hat{p}_{I-1,J} - \hat{p}_{I,J}) \cdot \delta \tilde{x} + \rho_0 \frac{\delta x \cdot \delta \tilde{z}}{\Delta t} u_{i,j}^0 \quad (5.27a)$$

$$a_{i,j} w_{i,j} = \sum a_{nb} w_{nb} + (\hat{p}_{I,J-1} - \hat{p}_{I,J}) \cdot \delta \tilde{x} + \rho_0 g \beta (T_{I,J}^0 - T_0) \delta x \cdot \delta \tilde{z} + \rho_0 \frac{\delta x \cdot \delta \tilde{z}}{\Delta t} w_{i,j}^0 \quad (5.27b)$$

Continuïteitsvergelijking:

$$(u_{i+1,J} - u_{i,J}) \cdot \delta \tilde{x} + (w_{I,j+1} - w_{I,j}) \cdot \delta \tilde{x} = 0 \quad (5.27c)$$

Energievergelijking:

$$a_{I,J} T_{I,J} = a_{I-1,J} T_{I-1,J} + a_{I+1,J} T_{I+1,J} + a_{I,J-1} T_{I,J-1} + a_{I,J+1} T_{I,J+1} + \rho_0 c_p \frac{\delta x \cdot \delta \tilde{z}}{\Delta t} T_{I,J}^0 \quad (5.27d)$$

Hoofdstuk 6

Het SIMPLE algoritme

In dit hoofdstuk wordt het SIMPLE algoritme uitgelegd. In 6.1 zal het algoritme met zijn achterliggende gedachte worden geïntroduceerd zodat duidelijk wordt waar naar toe wordt gewerkt en wat de belangrijkste oplossingsstrategie is. In 6.2 wordt het algoritme stap voor stap verder uitgediept zodat het duidelijk wordt waar welke vergelijking vandaan komt. Uiteraard vormen de vergelijkingen die zijn opgesteld in hoofdstuk 6, vergelijkingen (5.27), de basis van het algoritme.

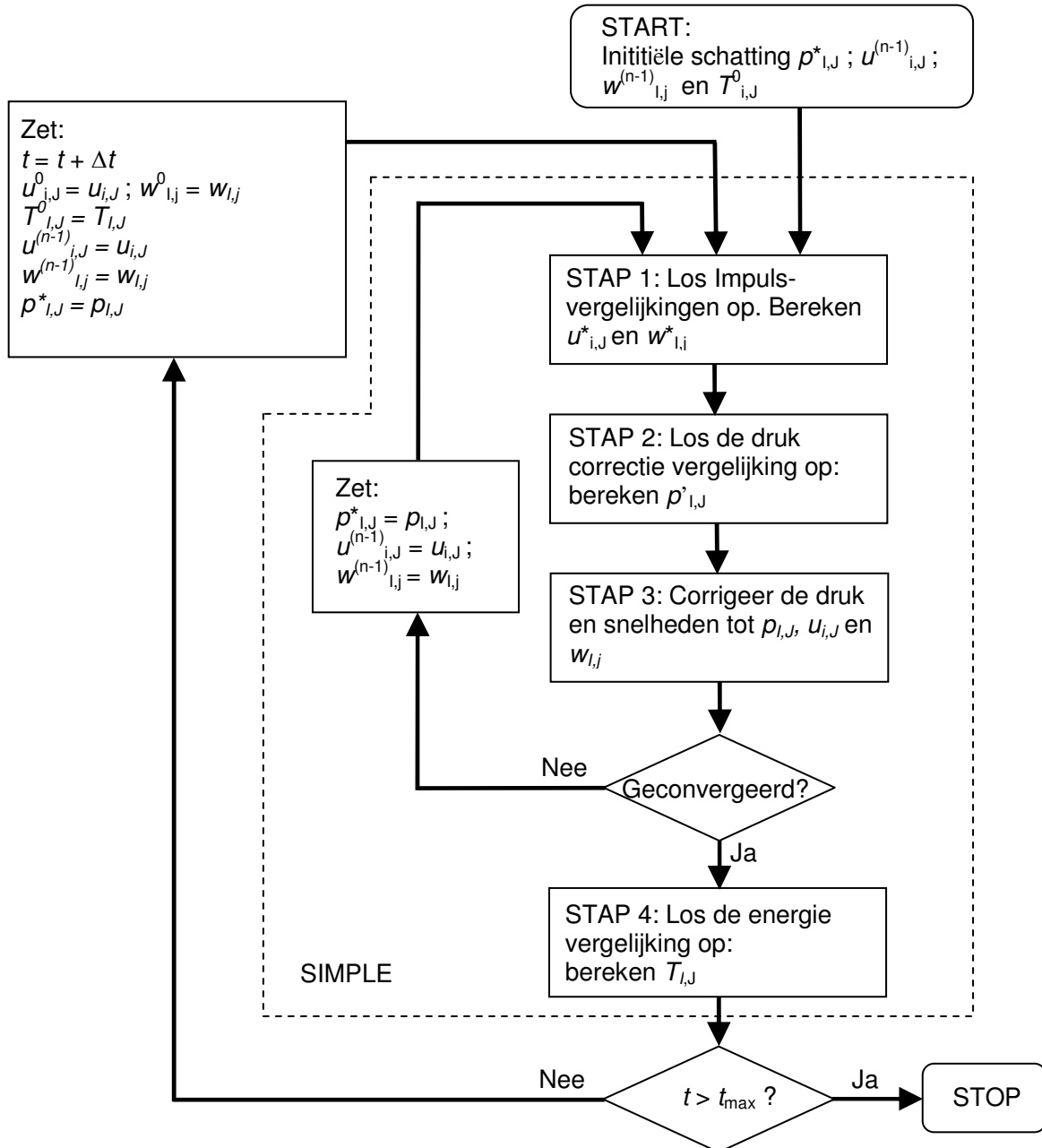
6.1 Introductie SIMPLE algoritme

In hoofdstuk 5 zijn de drie soorten vergelijkingen gediscretiseerd, vergelijkingen (5.27). Het SIMPLE algoritme start met het oplossen van de impulsvergelijkingen (5.27a en b). Zoals gezegd moeten de vergelijkingen worden opgelost met de matrix methode uit 5.2.1. De impulsvergelijkingen bevatten echter twee soorten onbekenden: de snelheid en de druk. Daarom wordt er bij de start van de modellering een drukveld geschat zodat er nog één soort bekende overblijft: de snelheid. Via de matrixmethode, zie 5.2.1, kan het snelheidsveld worden berekend wat eveneens als ‘geschat’ veld wordt beschouwd omdat het drukveld ook is geschat. Vervolgens wordt met de continuïteitsvergelijking (5.27c) gecontroleerd of het snelheidsveld wel voldoet aan de wet van behoud van massa, zie 3.2.1. Als na het invullen van het ‘geschatte’ snelheidsveld de continuïteitsvergelijking niet gelijk wordt aan nul zal het vooraf geschatte drukveld moeten worden aangepast, worden er nieuwe snelheden uitgerekend en wordt deze weer gecontroleerd met de continuïteitsvergelijking net zo lang totdat de continuïteitsvergelijking klopt. Indien de continuïteitsvergelijking klopt is er ook een correcte drukveld en zijn er correcte snelheidsvelden. Met deze correcte velden kan vervolgens het temperatuursveld worden berekend door de snelheidsvariabelen in de energievergelijking (5.27d) in te vullen. Nu alle velden bekend zijn kan worden gegaan naar de volgende tijdstap en zal er weer een drukveld, en daarmee snelheidsvelden, moeten worden berekend door ze te benaderen met behulp van de continuïteitsvergelijking.

De vergelijkingen die moeten worden opgelost, vergelijkingen (5.27), zijn afhankelijk van de tijd. Echter, het basis idee van het SIMPLE algoritme gaat uit van een steady state situatie om op die manier de transportvelden te benaderen. Daarom, zie figuur 6.1, is het SIMPLE algoritme weergegeven binnen de doorbroken lijnen. Buiten dat kader wordt er overgegaan naar een volgend tijdstap door t met Δt te verhogen. Alle berekeningen die met de matrixmethode moeten worden uitgevoerd vragen om het juiste aantal vergelijkingen.

Daarvoor moeten er ook, bij de randen van de ruimte, randvoorwaarden worden opgesteld. Deze randvoorwaarden zijn te vinden in bijlage 10.4.

6.2 Het SIMPLE algoritme



Figuur 6.1: SIMPLE algoritme

In 6.1 is de achterliggende gedachte van het SIMPLE algoritme uitgelegd. In deze paragraaf zullen de stappen verder worden uitgediept zodat alle bewerkingen die nodig zijn binnen de stappen duidelijk zijn.

6.2.1 Stap 1: impulsvergelijkingen

Deze eerste stap betreft het oplossen van de impulsvergelijkingen. De impulsvergelijkingen die moeten worden opgelost zijn vergelijkingen (5.27a en b) met het verschil dat er een

schatting van het drukveld wordt ingevoerd. De geschatte drukvariabele wordt aangegeven met een sterretje: \hat{p}^* . De impulsvergelijkingen die moeten worden opgelost zijn dan als volgt:

$$a_{i,j}u_{i,j}^* = \sum a_{nb}u_{nb}^* + (\hat{p}_{i-1,j}^* - \hat{p}_{i,j}^*) + \rho_0 \frac{\delta x}{\Delta t} u_{i,j}^0 \quad (6.1a)$$

$$a_{i,j}w_{i,j}^* = \sum a_{nb}w_{nb}^* + (\hat{p}_{i,j-1}^* - \hat{p}_{i,j}^*) + \rho_0 g \beta (T_{i,j}^0 - T_0) \delta x + \rho_0 \frac{\delta x}{\Delta t} w_{i,j}^0 \quad (6.1b)$$

Ook voor de snelheden die worden berekend, u_{nb} en $u_{i,j}$ geldt dat deze snelheden als geschat worden beschouwd en worden daarom ook uitgedrukt met een sterretje: u_{nb}^* en $u_{i,j}^*$. Dit geldt uiteraard ook voor w_{nb}^* en $w_{i,j}^*$. Naast het schatten van het drukveld worden ook de parameters $u^{(n-1)}$ en $w^{(n-1)}$ die in te parameters $a_{i,j}$, $a_{i,j}$ en a_{nb} zitten, zie vergelijkingen (6.1), geschat en wordt er uitgegaan van een initieel temperatuursveld T^0 . Pas dan kunnen de vergelijkingen (6.1) worden opgelost en de velden u^* en w^* worden berekend.

6.2.2 Stap 2: druk correctievergelijking

In deze stap worden de drukcorrecties uitgerekend via de druk correctievergelijking. Daarvoor moet eerst de druk correctievergelijking worden ontwikkeld. Om deze vergelijking te krijgen worden de volgende variabelen geïntroduceerd:

$$\begin{aligned} \hat{p} &= \hat{p}^* + \hat{p}' \\ u &= u^* + u' \\ w &= w^* + w' \end{aligned} \quad (6.2)$$

Met hierin:

- \hat{p} : echte druk
- \hat{p}^* : geschatte druk
- \hat{p}' : correctie waarde van de druk

- Wanneer de berekening zich bevindt aan het begin van de eerste iteratie, wordt voor \hat{p}^* een beginschatting genomen.
- Wanneer de berekening zich bevindt midden in de iteratie, is de waarde van \hat{p}^* , de geschatte druk, gelijk aan de berekende waarde van de druk uit de vorige iteratiestap.
- Wanneer de berekening zich bevindt aan de start van een nieuwe iteratie, wordt voor de waarde van \hat{p}^* de laatst berekende waarde uit het vorige tijd stap genomen.

Deze opsomming geldt ook voor $u^{(n-1)}$ en $w^{(n-1)}$, zie 5.2.2.

Wanneer de drukcorrectie \hat{p}' en de snelheidcorrecties u' en w' worden gesubstitueerd in de impulsvergelijkingen (5.27 a en b) in plaats van de echte variabelen, komt de vergelijking er als volgt uit te zien:

$$a_{i,j}u'_{i,j} = \sum a_{nb}u'_{nb} + (\hat{p}'_{I-1,j} - \hat{p}'_{I,j}) \cdot \delta \mathbf{x} + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{\Delta t} u_{i,j}^0 \quad (6.3a)$$

$$a_{I,j}w'_{I,j} = \sum a_{nb}w'_{nb} + (\hat{p}'_{I,j-1} - \hat{p}'_{I,j}) \cdot \delta \mathbf{x} + \rho_0 g \beta (T_{I,j}^0 - T_0) \cdot \delta \mathbf{x} \cdot \delta \mathbf{z} + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{\Delta t} w_{I,j}^0 \quad (6.3b)$$

Nu kan de sommatie term worden weggelaten uit deze vergelijkingen. Dit kan omdat de correctie methode die ontwikkeld wordt een iteratie methode is. Dat wil zeggen, dat via de uiteindelijke ontwikkelde druk correctievergelijking drukcorrecties worden berekend die worden opgeteld bij de berekende druk in de vorige iteratie stap. Er wordt dus geconvergeerd naar een juiste oplossing van de druk. Het blijkt, uit de literatuur, dat de sommatieterm weinig bijdraagt aan het convergeren en daarom kan worden verwaarloosd. Voor meer uitleg hierover zie *An introduction to computational fluid dynamics* [2]. Na het toepassen van deze verwaarlozing worden de vergelijkingen (6.3) als volgt:

$$a_{i,j}u'_{i,j} = (\hat{p}'_{I-1,j} - \hat{p}'_{I,j}) \cdot \delta \mathbf{x} + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{\Delta t} u_{i,j}^0 \quad (6.4a)$$

$$a_{I,j}w'_{I,j} = (\hat{p}'_{I,j-1} - \hat{p}'_{I,j}) \cdot \delta \mathbf{x} + \rho_0 g \beta (T_{I,j}^0 - T_0) \cdot \delta \mathbf{x} \cdot \delta \mathbf{z} + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{\Delta t} w_{I,j}^0 \quad (6.4b)$$

Het omschrijven van de vergelijkingen naar de correctiesnelheden $u'_{i,j}$ en $w'_{I,j}$, geeft:

$$u'_{i,j} = \frac{1}{a_{i,j}} (\hat{p}'_{I-1,j} - \hat{p}'_{I,j}) \cdot \delta \mathbf{x} + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{i,j} \Delta t} u_{i,j}^0 \quad (6.5a)$$

$$w'_{I,j} = \frac{1}{a_{I,j}} (\hat{p}'_{I,j-1} - \hat{p}'_{I,j}) \cdot \delta \mathbf{x} + \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{I,j}} \rho_0 g \beta (T_{I,j}^0 - T_0) + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{I,j} \Delta t} w_{I,j}^0 \quad (6.5b)$$

Vervolgens kunnen de correctie snelheidsvergelijkingen (6.5 a en b) worden gesubstitueerd in de vergelijkingen $u' = u - u^*$ en $w' = w - w^*$ (vergelijkingen (6.2)) en ontstaan de volgende vergelijkingen:

$$u_{i,j} = u_{i,j}^* + \frac{1}{a_{i,j}} (\hat{p}'_{I-1,j} - \hat{p}'_{I,j}) \cdot \delta \mathbf{x} + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{i,j} \Delta t} u_{i,j}^0 \quad (6.6a)$$

$$w_{I,j} = w_{I,j}^* + \frac{1}{a_{I,j}} (\hat{p}'_{I,j-1} - \hat{p}'_{I,j}) \cdot \delta \mathbf{x} + \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{I,j}} \rho_0 g \beta (T_{I,j}^0 - T_0) + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{I,j} \Delta t} w_{I,j}^0 \quad (6.6b)$$

Dezelfde vergelijkingen kunnen worden opgesteld voor $u_{i+1,j}$ en $w_{I,j+1}$:

$$u_{i+1,j} = u_{i+1,j}^* + \frac{1}{a_{i+1,j}} (\hat{p}'_{I,j} - \hat{p}'_{I+1,j}) \cdot \delta \mathbf{x} + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{i+1,j} \Delta t} u_{i+1,j}^0 \quad (6.6c)$$

$$w_{I,j+1} = w_{I,j+1}^* + \frac{1}{a_{I,j+1}} (\hat{p}'_{I,j} - \hat{p}'_{I,j+1}) \cdot \delta \mathbf{x} + \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{I,j+1}} \rho_0 g \beta (T_{I,j+1}^0 - T_0) + \rho_0 \frac{\delta \mathbf{x} \cdot \delta \mathbf{z}}{a_{I,j+1} \Delta t} w_{I,j+1}^0 \quad (6.6d)$$

Dit is het moment dat de continuïteitsvergelijking wordt geïntroduceerd. In de paragrafen 6.1.1 en 6.1.2 staat immers geschreven dat de continuïteitsvergelijking wordt gebruikt om te

controleren of het snelheidsveld aan de massabalans voldoet. Daarom worden de vergelijkingen (6.6) gesubstitueerd in de continuïteitsvergelijking (5.27c):

$$(u_{i+1,J} - u_{i,J}) \cdot \delta \underline{x} + (w_{I,j+1} - w_{I,j}) \cdot \delta \underline{x} = 0$$

Waardoor de volgende vergelijking ontstaat:

$$\begin{aligned} & u_{i+1,J}^* \cdot \delta \underline{x} + \frac{1}{a_{i+1,J}} (\hat{p}'_{I,J} - \hat{p}'_{I+1,J}) \cdot \delta \underline{x} \cdot \delta \underline{x} + \rho_0 \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{i+1,J} \Delta t} u_{i+1,J}^0 - u_{i,J}^* \cdot \delta \underline{x} - \frac{1}{a_{i,J}} (\hat{p}'_{I-1,J} - \hat{p}'_{I,J}) \cdot \delta \underline{x} \cdot \delta \underline{x} \\ & - \rho_0 \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{i,J} \Delta t} u_{i,J}^0 + w_{I,j+1}^* \cdot \delta \underline{x} + \frac{1}{a_{I,j+1}} (\hat{p}'_{I,J} - \hat{p}'_{I,J+1}) \cdot \delta \underline{x} \cdot \delta \underline{x} + \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{I,j+1}} \rho_0 g \beta (T_{I,J+1}^0 - T_0) \\ & + \rho_0 \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{I,j+1} \Delta t} w_{I,j+1}^0 - w_{I,j}^* \cdot \delta \underline{x} - \frac{1}{a_{I,j}} (\hat{p}'_{I,J-1} - \hat{p}'_{I,J}) \cdot \delta \underline{x} \cdot \delta \underline{x} - \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{I,j}} \rho_0 g \beta (T_{I,J}^0 - T_0) \\ & - \rho_0 \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{I,j} \Delta t} w_{I,j}^0 = 0 \end{aligned} \quad (6.7)$$

Na het herschrijven ziet de vergelijking er als volgt uit:

$$\begin{aligned} & \left(\frac{\delta \underline{x} \cdot \delta \underline{x}}{a_{i+1,J}} + \frac{\delta \underline{x} \cdot \delta \underline{x}}{a_{i,J}} + \frac{\delta \underline{x} \cdot \delta \underline{x}}{a_{I,j+1}} + \frac{\delta \underline{x} \cdot \delta \underline{x}}{a_{I,j}} \right) \hat{p}'_{I,J} = \left(\frac{\delta \underline{x} \cdot \delta \underline{x}}{a_{i+1,J}} \right) \hat{p}'_{I+1,J} + \left(\frac{\delta \underline{x} \cdot \delta \underline{x}}{a_{i,J}} \right) \hat{p}'_{I-1,J} + \left(\frac{\delta \underline{x} \cdot \delta \underline{x}}{a_{I,j+1}} \right) \hat{p}'_{I,J+1} + \left(\frac{\delta \underline{x} \cdot \delta \underline{x}}{a_{I,j}} \right) \hat{p}'_{I,J-1} \\ & - \rho_0 \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{i+1,J} \Delta t} u_{i+1,J}^0 + \rho_0 \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{i,J} \Delta t} u_{i,J}^0 - \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{I,j+1}} \rho_0 g \beta (T_{I,J+1}^0 - T_0) - \rho_0 \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{I,j+1} \Delta t} w_{I,j+1}^0 \\ & + \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{I,j}} \rho_0 g \beta (T_{I,J}^0 - T_0) + \rho_0 \frac{\delta \underline{x} \cdot \delta \underline{x} \cdot \delta \underline{x}}{a_{I,j} \Delta t} w_{I,j}^0 - u_{i+1,J}^* \cdot \delta \underline{x} + u_{i,J}^* \cdot \delta \underline{x} - w_{I,j+1}^* \cdot \delta \underline{x} + w_{I,j}^* \cdot \delta \underline{x} \end{aligned} \quad (6.8)$$

Deze vergelijking (6.8) is de druk correctievergelijking die nodig is om correctie waarden voor de druk, $\hat{p}'_{I,J}$, uit te rekenen. De vergelijking heeft dezelfde structuur als vergelijking (5.21) waarbij hier de onbekenden $\hat{p}'_{I,J}$, $\hat{p}'_{I+1,J}$, $\hat{p}'_{I-1,J}$, $\hat{p}'_{I,J+1}$ en $\hat{p}'_{I,J-1}$ zijn en dienen te worden uitgerekend door middel van de matrixmethode uit 5.2.1. De rest, de constanten, zijn allemaal bekend. Indien het hele drukveld in deze vergelijking wordt opgesteld, dus voor elke punt in het veld een vergelijking, zijn er genoeg vergelijkingen om de onbekenden op te lossen. In stap 1 zijn geschatte snelheidsvelden u^* en w^* berekend uit een geschat drukveld \hat{p}^* . Deze geschatte snelheidsvelden worden gebuikt in de druk correctievergelijking (6.8) om het druk correctieveld uit te rekenen.

6.2.3 Stap 3: het corrigeren van de velden

In de vorige stap is het druk correctieveld uitgerekend via de druk correctievergelijking (6.8). Dit druk correctieveld wordt gebruikt in deze stap 3 om het drukveld en vervolgens ook de snelheidsvelden te corrigeren. Daarvoor wordt de volgende vergelijking geïntroduceerd:

$$p = p^* + \alpha_p p' \quad (6.9)$$

Hierin is:

α_p de onderrelaxatiefactor in [-]

Het nieuwe drukveld bestaat dus uit het geschatte drukveld (uit stap 1) met daarbij opgeteld de drukcorrectie (uit stap 2) maal α_p . De waarde van α_p is een waarde tussen de 0 en de 1 en wordt de onderrelaxatie factor genoemd. Dit getal zal proefondervindelijk moeten worden bepaald.

De volgende stap bestaat uit het corrigeren van de snelheidsvelden u en w . Hiervoor worden de volgende vergelijkingen geïntroduceerd:

$$u = \alpha_u u^p + (1 - \alpha_u) u^{(n-1)} \quad (6.10a)$$

$$w = \alpha_w w^p + (1 - \alpha_w) w^{(n-1)} \quad (6.10b)$$

Hierin is:

u de nieuwe snelheid aan het einde van een iteratiestap.

u^p de snelheid die wordt verkregen met het nieuwe gecorrigeerde drukveld.

$u^{(n-1)}$ de snelheid uit de vorige iteratiestap.

Hierbij zijn de waardes α_u en α_w ook onderrelaxatie factoren, met een waarde tussen de 0 en de 1. Om u^p en w^p te krijgen moeten nogmaals de impulsvergelijkingen (5.27a en b) worden opgelost, maar nu met de gecorrigeerde druk uit deze stap in plaats van de geschatte druk uit stap 1. Er kan gekozen worden om eerst u^p en w^p apart uit te rekenen via de impulsvergelijkingen (5.27a en b) en daarna vergelijkingen (6.10) te gebruiken om de nieuwe snelheden u en w uit te rekenen. Hier wordt echter gekozen om beide vergelijkingen te substitueren, vergelijkingen (6.10) met (5.27a en b) waardoor de volgende vergelijkingen ontstaan:

$$\frac{a_{i,j}}{\alpha_u} u_{i,j} = \sum a_{nb} u_{nb} + (\hat{p}_{I-1,j} - \hat{p}_{I,j}) \cdot \delta \bar{x} + \rho_0 \frac{\delta \bar{x} \cdot \delta \bar{x}}{\Delta t} u_{i,j}^0 + (1 - \alpha_u) \frac{a_{i,j}}{\alpha_u} u_{i,j}^{(n-1)} \quad (6.11a)$$

$$\begin{aligned} \frac{a_{I,j}}{\alpha_w} w_{I,j} &= \sum a_{nb} w_{nb} + (\hat{p}_{I,j-1} - \hat{p}_{I,j}) \cdot \delta \bar{x} + \rho_0 g \beta (T_{I,j}^0 - T_0) \cdot \delta \bar{x} \cdot \delta \bar{x} + \rho_0 \frac{\delta \bar{x} \cdot \delta \bar{x}}{\Delta t} w_{I,j}^0 \\ &+ (1 - \alpha_w) \frac{a_{I,j}}{\alpha_w} w_{I,j}^{(n-1)} \end{aligned} \quad (6.11b)$$

Met deze vergelijkingen (6.11) kunnen dus de gecorrigeerde snelheidsvelden u en w worden uitgerekend door het gecorrigeerde drukveld \hat{p} van vergelijking (6.9) te implementeren.

Voordat kan worden overgegaan naar stap 4 moet worden bepaald of er genoeg is geconvergeerd, zie figuur 6.1. Dit kan worden bepaald door bij iedere iteratie stap het maximale verschil tussen het oude en nieuwe drukveld te berekenen (dit verschil is het druk correctieveld). Wanneer dit verschil klein genoeg wordt, de orde grootte daarvan kan zelf

bepaald worden, kan over worden gegaan naar stap 4. Hetzelfde kan worden gedaan voor de snelheidsvelden. Wanneer het er nog niet genoeg is geconvergeerd wordt weer overgegaan naar stap 1. Hierbij wordt het geschatte drukveld p^* gelijk aan p en de snelheden $u^{(n-1)}$ en $w^{(n-1)}$ die in te termen $a_{i,J}$, $a_{i,J}$ en a_{nb} zitten, zie vergelijking (6.1), worden gelijk gesteld aan $u_{i,J}$ en $w_{i,J}$. Zie hiervoor figuur 6.1.

6.2.4 Stap 4: het oplossen van de energievergelijking

Deze stap bestaat uit het berekenen van het temperatuurveld door het oplossen van de gediscretiseerde energie vergelijking, vergelijking (5.27d):

$$a_{i,J}T_{i,J} = a_{i-1,J}T_{i-1,J} + a_{i+1,J}T_{i+1,J} + a_{i,J-1}T_{i,J-1} + a_{i,J+1}T_{i,J+1} + \rho_0 c_p \frac{\delta x \cdot \delta z}{\Delta t} T_{i,J}^0$$

Ook deze vergelijking kan worden opgelost door het creëren van het juiste aantal vergelijkingen, inclusief randvoorwaarden. Met de matrixmethode uit 5.2.1 is dit stelsel vergelijkingen oplosbaar aangezien de snelheden, die verpakt zitten in de a parameters (zie 5.3.3) bekend zijn. Het snelheidsveld is bepaald door de iteratiemethode uit de stappen 1 tot en met 3.

Wanneer het nieuwe temperatuurveld is berekend kan worden overgegaan op de volgende tijdstap door t met Δt te verhogen. Voor de velden die aan de start van de eerste iteratie moeten worden geschat worden de waarden genomen uit de vorige tijdstap, zie figuur 6.1.

Hoofdstuk 7

Model

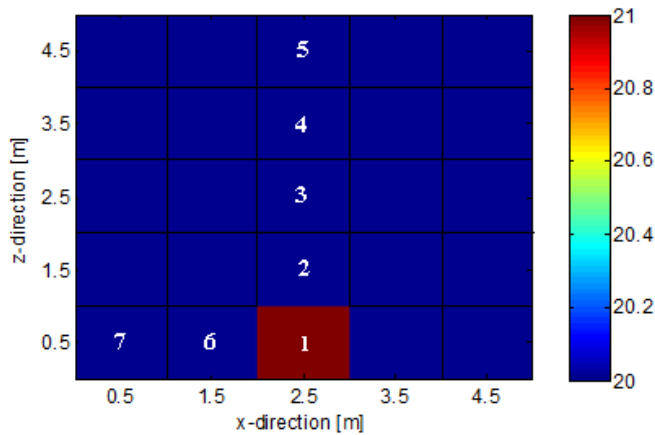
In dit hoofdstuk wordt het model, wat in de vorige hoofdstukken is opgesteld, geprogrammeerd met Matlab en gerund voor verschillende test situaties. Voordat het model gerund kan worden moeten eerst op een correcte wijze parameters worden bepaald wat gebeurt in 7.1. Vervolgens kunnen test situaties worden geschetst, gemodelleerd en gerund wat wordt gedaan in 7.2. Tenslotte, in 7.3, wordt het model gevalideerd met een officieel CFD pakket genaamd FLUENT, versie 5.3. Hieruit blijkt dat het model redelijk overeen komt met dit officiële pakket.

7.1 Bepalen van de parameters

In hoofdstuk 6 staat uitgelegd hoe de vergelijkingen zijn gemodelleerd via het SIMPLE algoritme. In bijlage 10.5 staan de Matlab files waarin het algoritme is geschreven waarbij gebruik gemaakt is van Matlab 6.4.0 (R2007a). Voordat het model kan worden gerund en voorbeelden kunnen worden gegeven zullen de volgende parameters moeten worden bepaald: de onderrelaxatiefactoren α_p , α_w , α_u (zie 6.2.3), de discretisatie tijd Δt , zie vergelijkingen (5.27), en het convergentie criterium (zie 6.2.3).

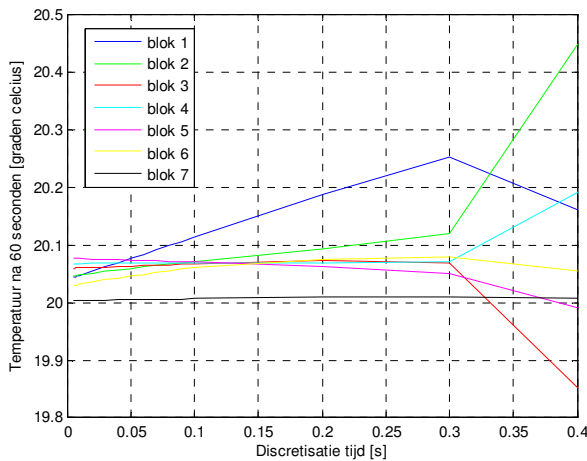
7.1.1 Discretisatie tijd

Voordat het model gerund kan worden moet een juiste discretisatie tijd Δt worden gekozen. Wanneer de discretisatie tijd te groot wordt gekozen is het model te onnauwkeurig. Wanneer de discretisatie tijd te klein wordt gekozen is het model te traag. Voor het testen van deze parameter zal worden begonnen met een hoge discretisatie tijd waarna deze steeds kleiner wordt gemaakt. De discretisatie tijd wordt vervolgens uitgezet tegen de eindtemperatuur op verschillende subruimtes in het temperatuursveld, namelijk de blokken één tot en zeven, zie figuur 7.1. In dit figuur geeft de rechter kleurenbalk de temperatuur behorende bij de kleuren in het veld weer. Het model wordt gerund voor 60 seconden met de verschillende discretisatie tijden gekozen tussen de 0.4 en 0.005 s. De initiële situatie van het model is ook als figuur 7.1. In dit figuur wordt een ruimte weergegeven waar het overal in de ruimte 20 °C is behalve in één subruimte waar het 21 °C is.

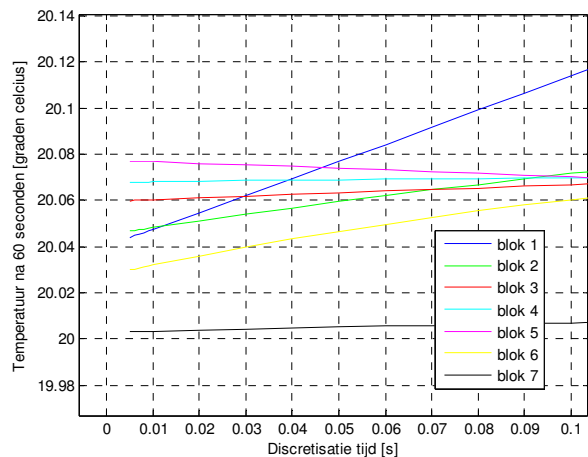


Figuur 7.1: *Initiële conditie van het model*

Het model wordt gerund voor 60 seconden en daarbij wordt per subruimte in het grid de discretisatie tijd uitgezet tegen de eindtemperatuur die op dat moment na 60 seconden is geconstateerd. Het resultaat is weergegeven in de figuren hieronder.



Figuur 7.2a: *Discretisatie tijd uitgezet tegen T*



Figuur 7.2b: *Discretisatie tijd uitgezet tegen T*

Wat blijkt uit figuur 7.2a is dat vanaf $\Delta t = 0.1$ s en lager de eindtemperatuur in de buurt begint te komen van de ideale eindtemperatuur. De ideale eindtemperatuur is de temperatuur die het model theoretisch zou kunnen berekenen wanneer een oneindige kleine discretisatie tijd wordt gekozen. Als er een hogere discretisatie tijd wordt gekozen blijkt dat, vooral voor blok 1, de eindwaarde in het blok nog teveel afwijkt van de ideale waarde die het zal moeten bereiken wanneer Δt richting nul wordt gekozen, zie figuur 7.2b. In de volgende tabel is voor elk blok de ideale eindtemperatuur berekend door elke lijn in de grafiek 7.2a te extrapoleren naar $\Delta t = 0$ s. In de kolommen er naast is het afwijkingspercentage weergegeven. Dit afwijkingspercentage is berekend met het initiële temperatuurverschil wat in dit geval 1 °C is (21 °C ten opzichte van 20 °C, zie figuur 7.1). Een voorbeeld hiervan is gegeven voor blok 1 voor $\Delta t = 0.1$ s.

Voorbeeld 7.1:

In blok 1 is voor $\Delta t = 0.1$ [s] de eindtemperatuur (na het runnen van het model over 60 s) gelijk aan $T_{\Delta t=0.1} = 20.11$ °C. De extrapolatie geeft aan dat de eindtemperatuur in dat blok bij een ideaal model gelijk is aan $T_{\Delta t \rightarrow 0} = 20.04$ °C. Dit betekent dat het verschil tussen deze twee temperaturen gelijk is aan:

$$T_{\Delta t=0.1} - T_{\Delta t \rightarrow 0} = 20.11 - 20.04 = 0.07 \text{ } ^\circ\text{C}$$

0.07 °C afwijking ten opzichte van een initieel temperatuurverschil van 1 °C geeft aan dat het afwijkingspercentage gelijk is aan 7 %.

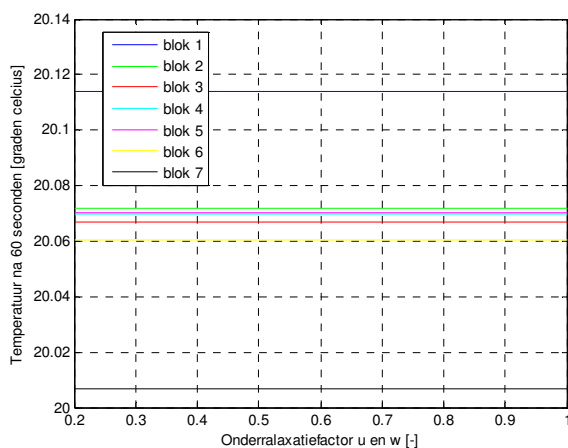
Bloknummer	$T_{\Delta t \rightarrow 0}$	$\Delta t = 0.1 \text{ s}$	$\Delta t = 0.05 \text{ s}$	$\Delta t = 0.02 \text{ s}$	$\Delta t = 0.01 \text{ s}$
1	20.04	7.0 %	4.0 %	1.0 %	1.0 %
2	20.05	2.5 %	1.5 %	0.5 %	0.5 %
3	20.06	1.2 %	0.2 %	0.2 %	0.2 %
4	20.07	0.2 %	0.2 %	0.2 %	0.2 %
5	20.08	0.8 %	0.8 %	0.2 %	0.2 %
6	20.03	3.5 %	2.5 %	1.5 %	0.5 %
7	20.00	0.8 %	0.8 %	0.2 %	0.2 %

Tabel 7.1: Afwijkingspercentage tabel

In tabel 7.1 zijn alle afwijkingspercentages per discretisatie tijd per blok weergegeven. Het blijkt dat blok 1 het grootste afwijkingspercentage heeft bij alle discretisatie tijden. Dit is ook in figuur 7.2b te zien omdat daar blok 1 de grootste helling heeft. Als gekozen wordt voor de norm van een afwijkingspercentage tot 1.0 % zal een discretisatie tijd van 0.01 s nodig zijn om deze norm te halen.

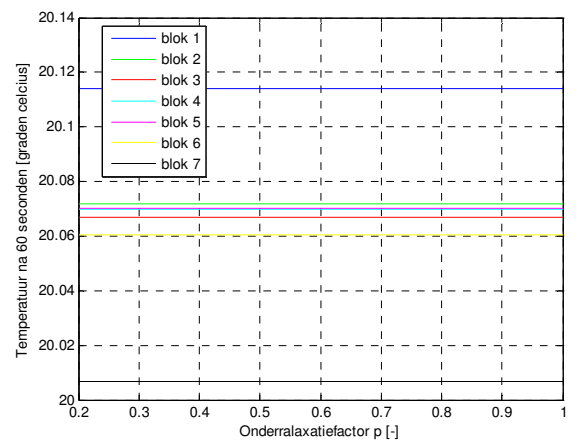
7.1.2 Onderrelaxatie factoren

Om het model goed te laten runnen moeten de juiste onderrelaxatiefactoren α_p , α_w , α_u worden gekozen, zie 6.2.3. Net als bij het bepalen van de discretisatietijd wordt hier per blok in figuur 7.1 bekeken wat de invloed is van de onderrelaxatiefactoren op de eindtemperatuur na 60 seconden. De resultaten voor beide soorten onderrelaxatiefactoren zijn verwerkt in de volgende figuren:



7.3a: Onderrelaxatiefactoren α_w en α_u tegen T

Figuur



Figuur 7.3b: Onderrelaxatiefactor α_p tegen T

Bij deze proef zijn de factoren α_w en α_u altijd aan elkaar gelijk gehouden. Figuur 7.3a geeft de invloed van α_w en α_u op de eindtemperatuur weer en bij α_p is dat figuur 7.3b. Zoals blijkt uit de figuren, heeft de variatie in de waarde van de onderrelaxatiefactoren geen invloed op de eindtemperatuur. In dat geval kan voor alle onderrelaxatiefactoren de waarde één

worden gekozen omdat bij deze waarde er minder iteratiestappen nodig zijn om aan het convergentie criterium te voldoen en is het model daardoor sneller.

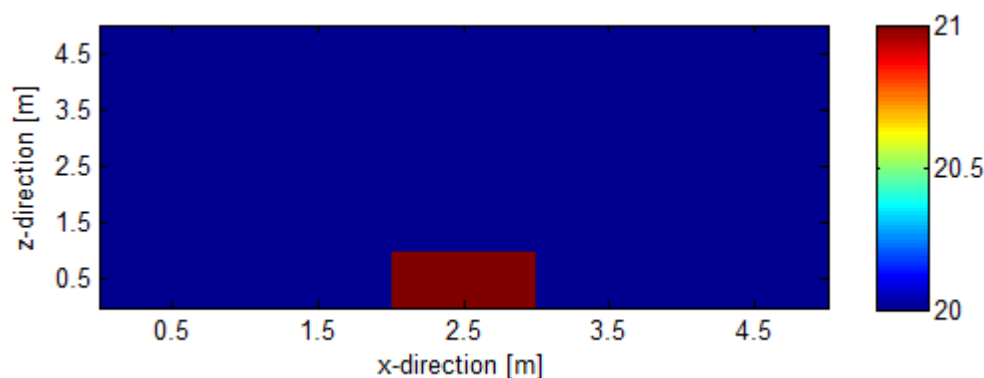
7.1.3 Convergentie criterium

Het convergentie criterium bepaald wanneer er over kan worden gegaan op een volgende tijdstap en er dus geen extra iteratie meer nodig is. In 6.2.3 is hier al het een en ander over gezegd. Bij dit model is gekozen om na een iteratie beide snelheidsvelden, zowel in de x - als de z -richting, te vergelijken met de snelheidsvelden uit de vorige iteratiestap. Wanneer voor beide velden geldt dat het absolute maximale verschil kleiner is dan $1 \cdot 10^{-10}$ m/s kan er worden overgegaan naar een volgende tijdstap.

7.2 Modelling met een grid van 5 bij 5

7.2.1 Afwijkende subruimte onderin

Voordat verderop in dit hoofdstuk, paragraaf 7.3, het model wordt gevalideerd met een officieel CFD pakket zullen eerst voorbeelden worden laten zien van verschillende test situaties. In de voorbeelden wordt uitgegaan van een ruimte van vijf bij vijf meter. De ruimte wordt gemodelleerd met een grid van vijf bij vijf. Initieel is de temperatuur over de hele ruimte $20\text{ }^\circ\text{C}$ behalve in de subruimte onderin de kas waar het $21\text{ }^\circ\text{C}$ is. Aan elke zijde van de ruimte doet zich een wand voor zodat de luchtstroming daar een andere weg moet kiezen. Er wordt vanuit gegaan dat er geen warmte verloren kan gaan door de wanden van de kas en er is geen initiële luchtstroming. Zie figuur 7.4 voor de visualisatie van de kas op $t = 0$ s.



Figuur 7.4: *Intiële conditie van het model*

Voor de parameters in de vergelijkingen (5.27) zijn de volgende waardes gebruikt:

$$\begin{aligned}
 c_p &= 1006 \text{ [J kg}^{-1} \text{ K}^{-1}] & k &= 0.0242 \text{ [W m}^{-1} \text{ K}^{-1}] \\
 \beta &= 0.0034 \text{ [K}^{-1}] & \rho(T) = \rho_0 &= 1.205 \text{ [kg m}^{-3}] \\
 \mu &= 1.846 \cdot 10^{-5} \text{ [kg m}^{-1} \text{ s}^{-1}] & T_0 &= 20 \text{ [}^\circ\text{C]} \\
 g &= 9.81 \text{ [m s}^{-2}]
 \end{aligned}$$

Deze parameters komen uit *Tables of Thermal Properties of Gases* [3].

Voor de discretisatie parameters zijn de volgende waarden gebruikt, zie 7.1:

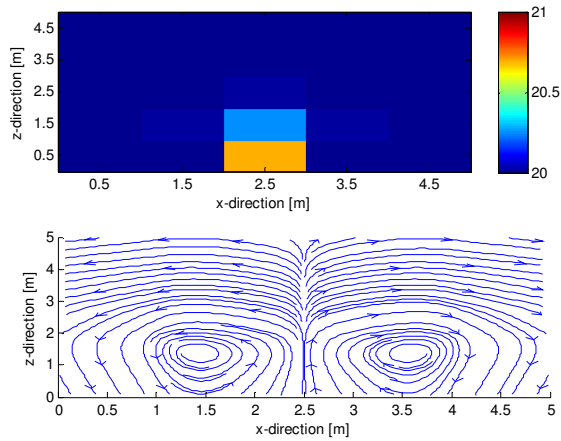
$$\begin{aligned}
 \Delta t &= 0.01 \text{ [s]} \\
 dx &= 1 \text{ [m]} & dz &= 1 \text{ [m]}
 \end{aligned}$$

$$\alpha_p = 1 [-]$$

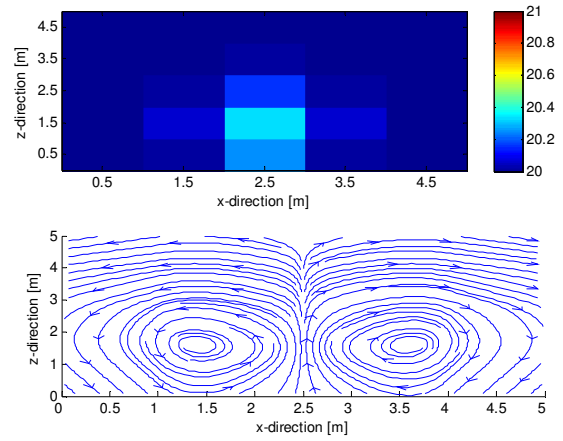
$$\alpha_u = 1 [-]$$

$$\alpha_w = 1 [-]$$

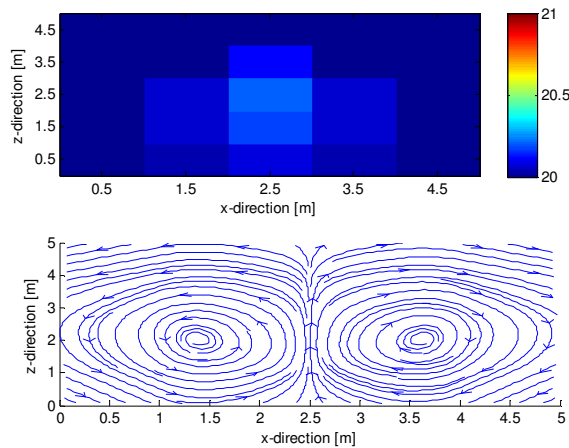
Het model zal worden gerund met een $\Delta t = 0.01$ s wat betekent dat elke 0.01 s er de beschikking is over nieuwe informatie over de velden. Hier is gekozen om de tijdstippen 10, 20, 30, 40, 50 en 60 seconden uit te kiezen om te laten zien in de volgende figuren. In de figuren is telkens een plaatje te zien met daarin het temperatuursveld en daaronder het snelheidsveld. In het snelheidsveld zijn alleen de richtingen te zien. De grootte van de snelheden zijn hierin niet verwerkt.



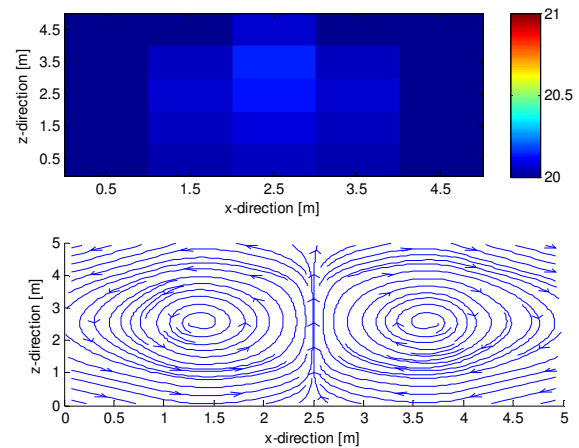
7.5a: $t = 10$ s



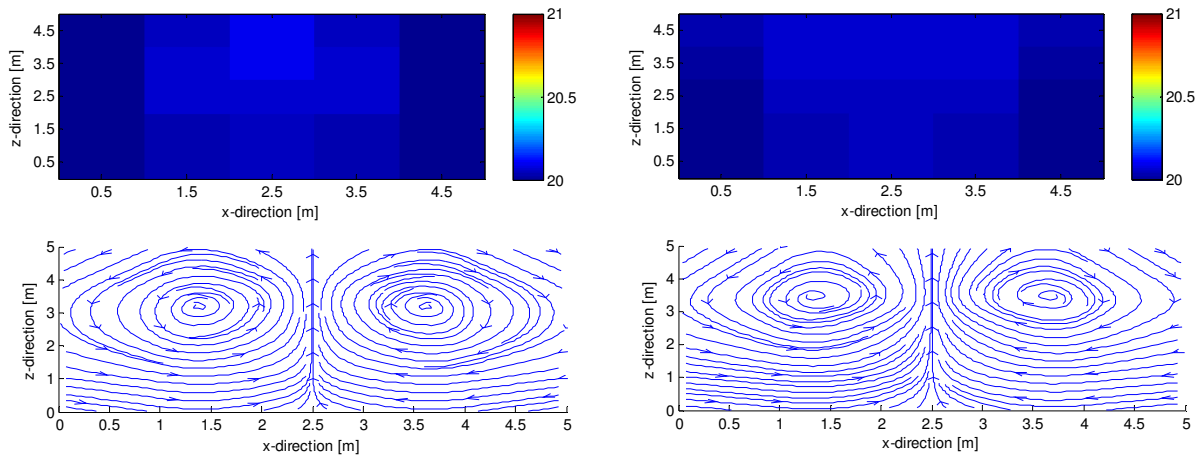
7.5b: $t = 20$ s



7.5c: $t = 30$ s



7.5d: $t = 40$ s



7.5e: $t = 50\text{ s}$

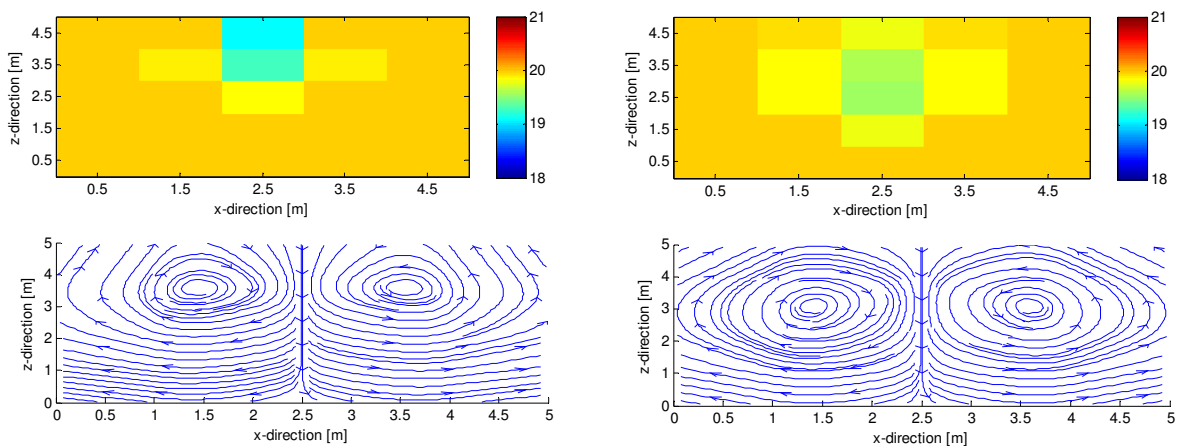
7.5f: $t = 60\text{ s}$

Figuur 7.5: Verloop van de temperatuur op verschillende tijdstippen met afwijkende subruimte van $21\text{ }^\circ\text{C}$

In de figuur 7.5a hierboven is te zien dat na 10 seconden het blok van $21\text{ }^\circ\text{C}$ kouder is geworden en dat bovenliggende en vervolgens weer omliggende blokken zijn opgewarmd. Dit klopt met het snelheidsveld wat is ontstaan. Dit snelheidsveld laat zien dat er een luchtstroming omhoog is ontstaan wat daarna naar opzij gaat. Dit komt omdat er bovenaan een wand is waar de luchtstroming niet verder kan. De figuren erna laten een later tijdstip zien. Het blijkt dat wat eerst het warme blok was steeds meer verdeeld wordt over de bovenliggende blokken en dat de temperatuur steeds meer homogeen verdeeld wordt over de ruimte. Het snelheidsveld verandert hierin mee omdat de warmte nu meer bovenin te vinden is. Het model is dus in staat om aan de hand van een temperatuursveld een snelheidsveld te ontwikkelen en daarmee vervolgens het temperatuursveld aan te passen. Deze wijziging in het temperatuursveld zorgt weer voor een wijziging in het snelheidsveld. Fysisch gezien is het goed verklaarbaar dat de warmte omhoog gaat omdat warmte stijgt.

7.2.2 Afwijkende subruimte bovenin

In 7.2.1 was initieel een ruimte gecreëerd met onderin een afwijkende temperatuur van $21\text{ }^\circ\text{C}$. Bij deze test zal initieel een ruimte worden gecreëerd waarbij bovenin een afwijkende temperatuur is van $18\text{ }^\circ\text{C}$. Voor de rest blijven alle parameters hetzelfde als in 7.2.1. De figuren hieronder laten het verloop zien op de tijdstippen $t = 10\text{ s}$ en $t = 20\text{ s}$.



7.6a: $t = 10\text{ s}$

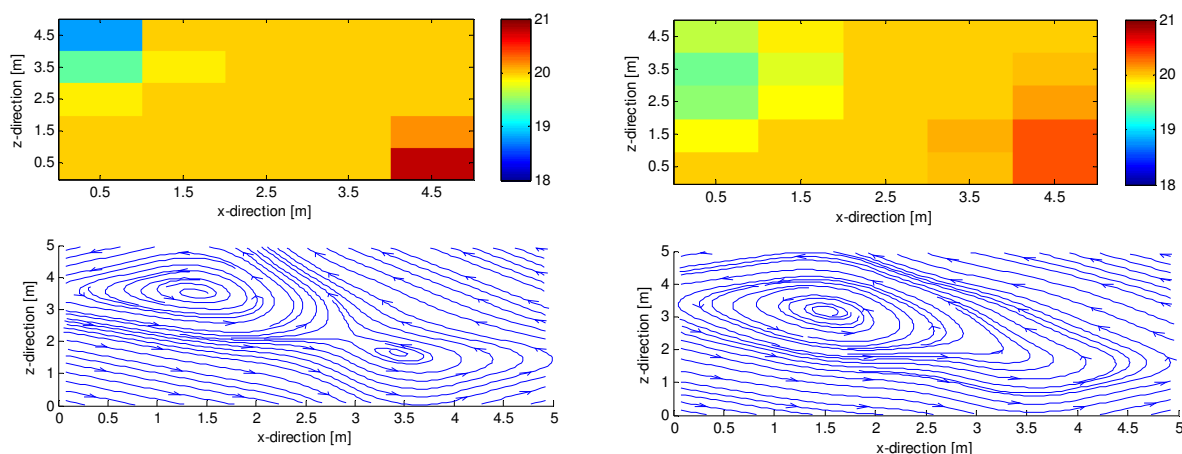
7.6b: $t = 20\text{ s}$

Figuur 7.6: Verloop van de temperatuur op verschillende tijdstippen met afwijkende subruimte van $18\text{ }^\circ\text{C}$

In deze figuren is te zien dat er vanaf de bovenkant een stroming naar beneden ontstaat en vervolgens naar opzij waardoor het koudere blok langzaam wordt opgewarmd en onderliggende blokken worden gekoeld. Fysisch gezien is het goed verklaarbaar dat wanneer bovenin een koeler blok zich bevindt dat deze zich naar beneden verplaatst omdat koele lucht wil zakken.

7.2.3 Afwijkende subruimtes op meerdere plaatsen

In bovenstaande testen zijn koude blokken en warme blokken geplaatst om te testen of het model dit fysisch goed interpreteerde. Omdat alles symmetrisch was opgebouwd ontstonden er ook symmetrie in de temperatuur- en snelheidsvelden. In deze test worden twee verschillende afwijkende subruimtes geplaatst waardoor er geen symmetrie meer is. Er wordt zowel een koude als warme subruimte geplaatst zodat het model moet omgaan met een koel punt en warmt punt. Links bovenin is een blok geplaatst van 18 °C en rechts onderin een blok van 21 °C. De rest van de ruimte heeft een temperatuur van 20 °C. De volgende figuren laten het verloop zien op de tijdstippen $t = 10$ s en $t = 20$ s.



7.7a: $t = 10$ s

7.7b: $t = 20$ s

Figuur 7.7: Verloop van de temperatuur op verschillende tijdstippen met afwijkende subruimtes van 18 °C (links bovenin) en 21 °C (rechts onderin).

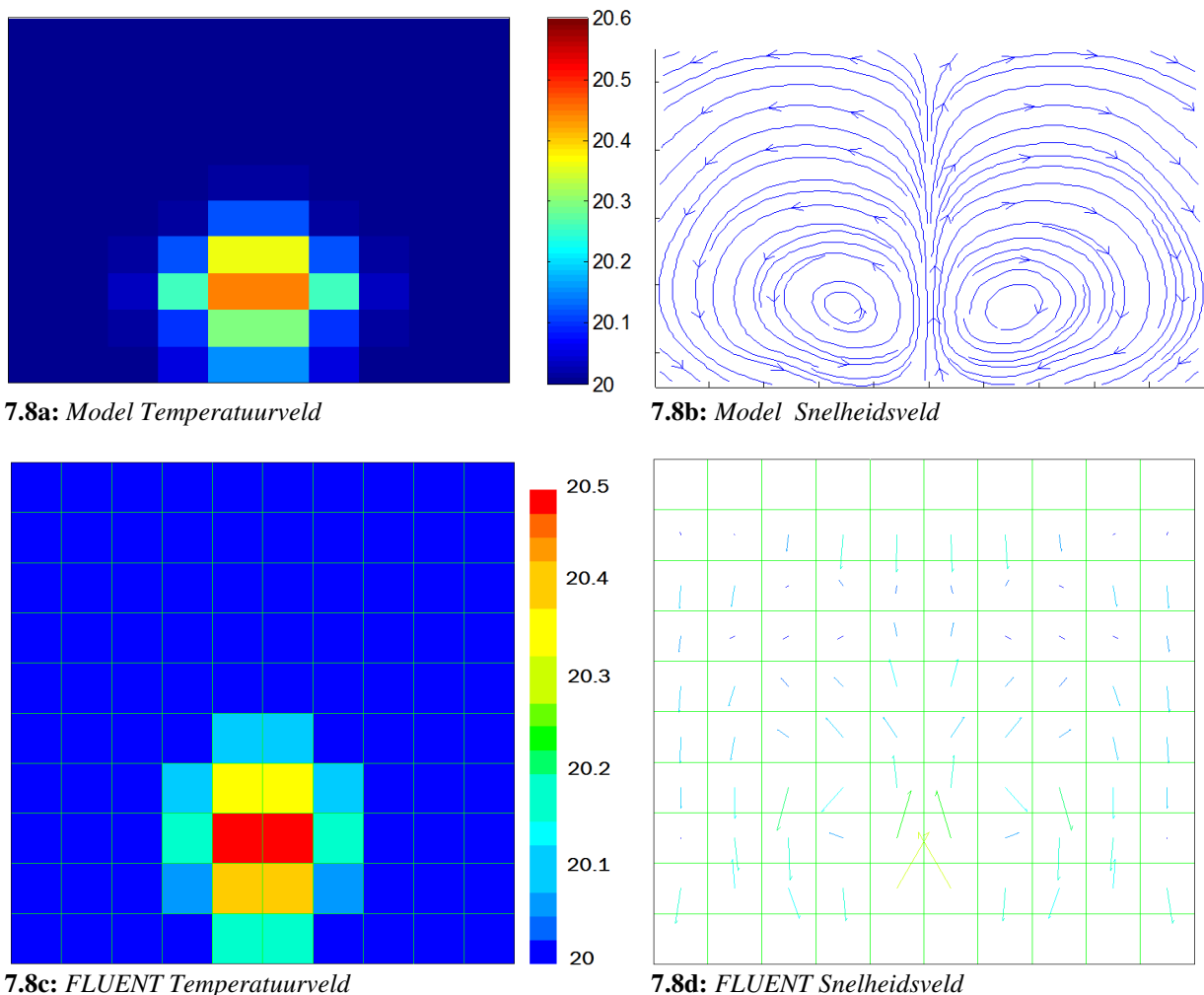
In deze figuur 7.7a is te zien dat er een stroming ontstaat om de afwijkende blokken en om het linker boven blok. 10 seconden later, op $t = 20$ s, is te zien dat de temperatuur al meer homogeen verdeeld is over de ruimte. Het koude blok zakt naar beneden en het warme blok stijgt omhoog.

7.3 Validatie model met CFD pakket

Om het model te valideren worden testen vergeleken met een zelfde soort testen van een officieel CFD pakket. Het CFD pakket wat daarvoor wordt gebruikt heet FLUENT, versie 6.0. Behalve de keuze voor het grid gelden dezelfde voorwaarden (parameter waarden) als zijn uitgelegd in 7.2.1. In FLUENT wordt gekozen voor een laminair model met een Boussinesq benadering omdat deze voorwaarden overeen komen met de vergelijkingen 5.27 die zijn gebruikt bij dit model. Alle condities en parameters zijn in het CFD pakket hetzelfde gekozen behalve de discretisatie tijd Δt . Bij dit model is $\Delta t = 0.01$ s en bij FLUENT is $\Delta t = 0.1$ s klein genoeg om tot goede resultaten te komen.

7.3.1 Grid van 10 bij 10

Bij deze test wordt gekozen voor een grid van 10 bij 10. Dat betekent, gezien dezelfde condities gelden als uitgelegd in 7.2.1, dat $dx = dy = 0.5 \text{ m}$. De lengte van de zowel de ruimte in de x -richting als de y -richting is vijf meter. De starttemperatuur is overal $20 \text{ }^\circ\text{C}$ behalve onderin waar een subruimte $21 \text{ }^\circ\text{C}$ is. Zie figuur 7.4. voor details over de initiële situatie. Beide programma's, het model (met model wordt bedoeld het model wat is ontwikkeld in dit verslag) en FLUENT worden gerund voor 16 seconden. Na 16 seconden worden beide resultaten met elkaar vergeleken. De figuren hieronder geven de resultaten van beide modellen weer. Wanneer beiden figuren van de verschillende modellen met elkaar worden vergeleken (figuren 7.8a en b met c en d) valt op dat zowel het temperatuurveld als snelheidsveld in vorm overeen komen. Wanneer vervolgens wordt gekeken wordt naar de waardes in het temperatuurveld blijken ook deze redelijk overeen te komen. Om in getallen uit te drukken wat het verschil is tussen beide plots wordt weer teruggegrepen op het afwijkingpercentage. Er is namelijk een initieel verschil van $1 \text{ }^\circ\text{C}$. Dit betekent dat wanneer er bijvoorbeeld een verschil is tussen dezelfde punten in het temperatuurveld van $0.03 \text{ }^\circ\text{C}$ er dan een afwijkingpercentage is van 3% , zie voorbeeld 7.1. Hierbij wordt iedere keer de middelste subruimte in een laag genomen om mee te vergelijken. Tabel 7.1 geeft de resultaten weer tussen de figuren 7.8a en c.



Figuur 7.8: Vergelijking van het model met FLUENT (grid van 10 bij 10)

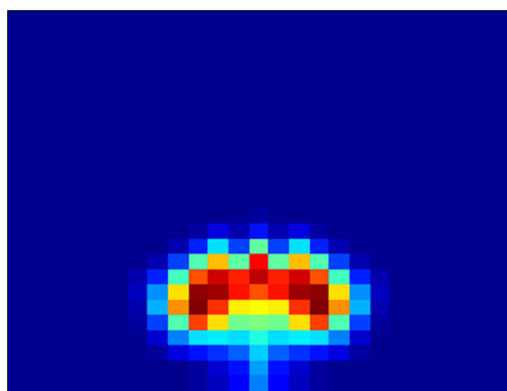
Laag	Temperatuur [°C] FLUENT	Temperatuur [°C] Model	Afwijkingspercentage [%]
1	20.15	20.20	5
2	20.40	20.30	10
3	20.50	20.45	5
4	20.35	20.35	0
Gemiddelde percentage			5

Tabel 7.1: Afwijkingspercentage tabel (grid van 10 bij 10)

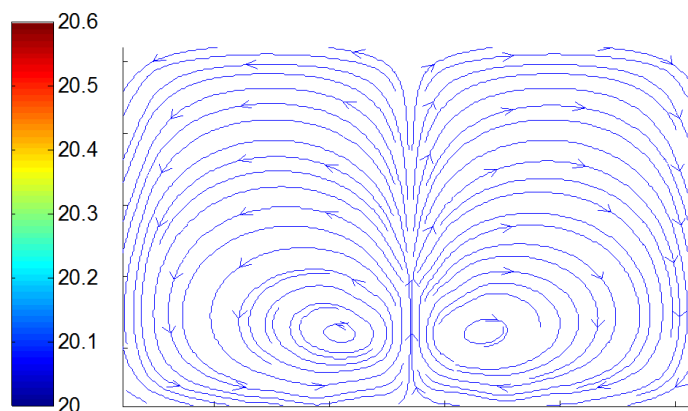
Het blijkt dat er een gemiddeld afwijkingspercentage van 5 % is tussen FLUENT en het model. Aangezien we te maken hebben met het vergelijken van twee modellen, met allebei een zekere benadering van de realiteit, is een afwijkingspercentage van 5 % een acceptabele afwijking.

7.3.2 Grid van 25 bij 25

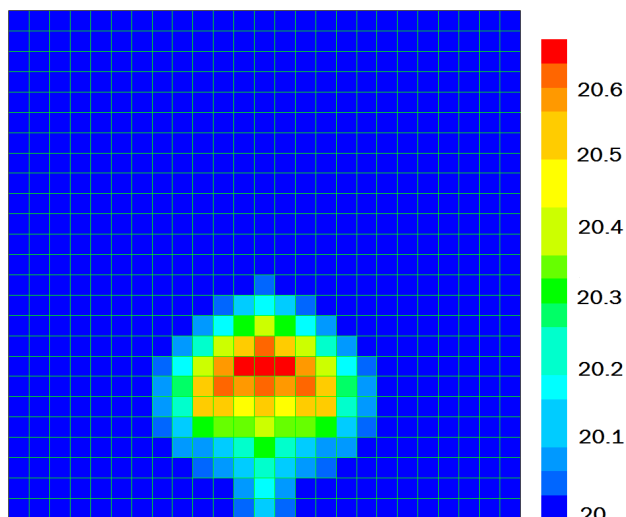
Bij deze test wordt gekozen voor een grid van 25 bij 25. Dat betekent, gezien dezelfde condities gelden als uitgelegd in 7.2.1, dat $dx = dy = 0.2$ m. De lengte van de zowel de ruimte in de x -richting als de y -richting is weer vijf meter. De starttemperatuur is overal 20 °C behalve onderin waar een subruimte 21 °C is. Zie figuur 7.4. voor details over de initiële situatie. Na 16 seconden worden beide resultaten met elkaar vergeleken. De figuren hieronder geven de resultaten van beide modellen weer.



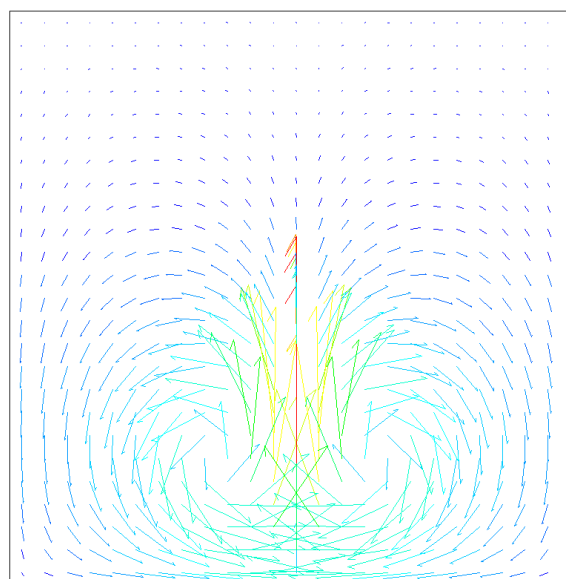
7.9a: Model Temperatuurveld



7.9b: Model Snelheidsveld



7.9c: FLUENT Temperatuurveld



7.9d: FLUENT Snelheidsveld

Figuur 7.9: Vergelijking van het model met FLUENT (grid van 25 bij 25)

Om de figuren 7.9a en c goed met elkaar te kunnen vergelijken wordt weer het afwijkingpercentage uitgerekend voor elke laag. Zie tabel 7.2 voor deze resultaten.

Laag	Temperatuur [°C] FLUENT	Temperatuur [°C] Model	Afwijkingpercentage [%]
1	20.12	20.13	1
2	20.18	20.15	3
3	20.20	20.19	1
4	20.30	20.22	8
5	20.40	20.30	10
6	20.50	20.37	13
7	20.60	20.48	12
8	20.68	20.56	12
9	20.60	20.52	8
Gemiddelde percentage			7.8

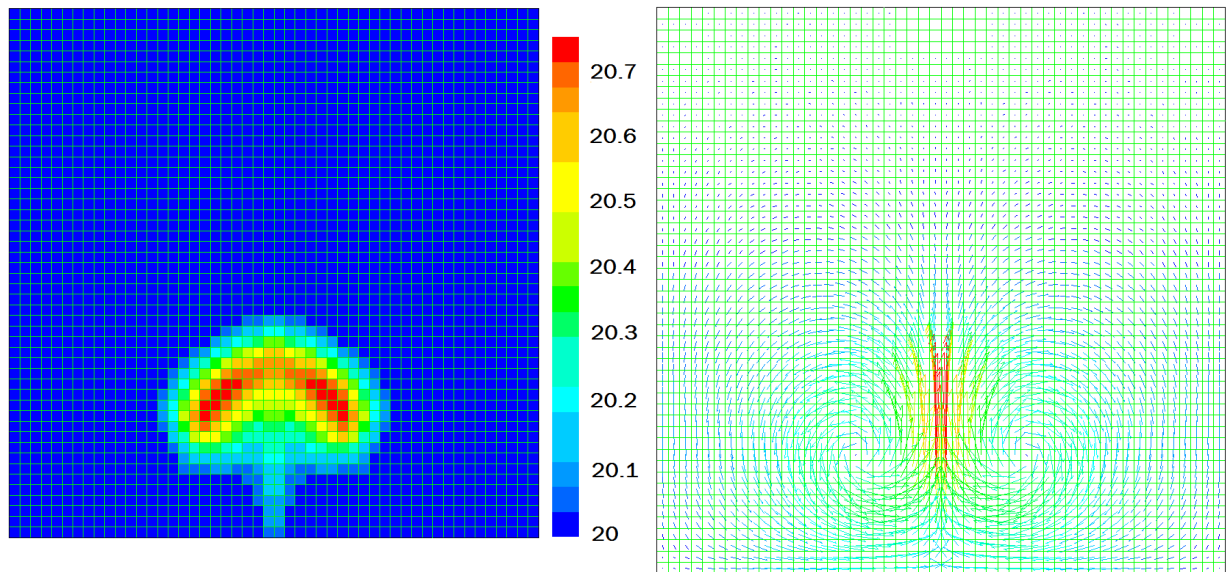
Tabel 7.2: Afwijkingpercentage tabel (grid van 25 bij 25)

Wat blijkt uit de tabel is dat het percentage 7.8 % wat meer is dan de resultaten bij een grid van 10 bij 10 in 7.3.1. Dit is opmerkelijk omdat er een fijner grid is gekozen wat alleen maar betere resultaten zou moeten geven. Wanneer echter figuur 7.9a wordt vergeleken met een nog fijner grid van FLUENT in 7.3.2 zijn er juist weer meer overeenkomsten.

7.3.3 Grid van 50 bij 50

Wanneer er wordt gekozen voor een grid van 50 bij 50 is dit nog wel mogelijk voor FLUENT maar echter niet meer voor het model. Het runnen van een grid van 50 bij 50, wat betekent dat er matrices van 2500 elementen op elkaar moeten worden gedeeld, duurt te lang voor het model. Wat wel kan worden gedaan is dat de resultaten van het model met een grid van 25 bij 25 uit 7.3.2 worden vergeleken met de resultaten van FLUENT met een grid van 50 bij 50. De reden dat dit wordt gedaan is dat de vorm van het temperatuurveld, zie figuur 7.9a, al sterk

begint te lijken op de vorm van het FLUENT temperatuurveld, figuur 7.10a. In tabel 7.3 zijn vervolgens de afwijkingspercentages weergegeven.



7.10a: FLUENT Temperatuurveld

7.10b: FLUENT Snelheidsveld

Figuur 7.10: FLUENT (grid van 50 bij 50)

Laag	Temperatuur [°C] FLUENT (50bij50)	Temperatuur [°C] Model (25bij25)	Afwijkingspercentage [%]
1	20.10	20.13	3
2	20.10	20.15	5
3	20.18	20.19	1
4	20.20	20.22	2
5	20.30	20.30	0
6	20.35	20.37	2
7	20.48	20.48	0
8	20.63	20.56	7
9	20.60	20.52	8
Gemiddelde percentage			3.1

Tabel 7.3: Afwijkingspercentage tabel (grid van 25 bij 25 en grid van 50 bij 50)

Het blijkt dat er tussen de twee temperatuurprofielen een gemiddeld afwijkingspercentage is van 3.1 procent. Dit percentage is veel kleiner dan de het afwijkingspercentage in 7.3.2. Dit komt blijkbaar omdat het temperatuurveld van figuur 7.9a al nauwkeuriger is berekend dan het temperatuurveld van FLUENT van hetzelfde grid, figuur 7.9c. Alle snelheidsvelden komen in vorm prima met elkaar overeen.

Hoofdstuk 8

Conclusie en aanbevelingen

8.1 Conclusie

Doel van dit onderzoek was de ontwikkeling van een dynamisch model dat de temperatuur in een kas beschrijft waarin luchtbehandelingskasten staan. Deze zorgen voor een significante ruimtelijke verdeling van temperatuur die daarom door het model moet worden beschreven. Aan de basis van het model liggen fundamentele fysische behoudswetten zoals het behoud van massa en energie. Op basis van deze wetten is een model in toestandsvorm opgesteld. Een model in deze vorm geeft duidelijk fysisch inzicht alsook inzicht in oorzaak en gevolg. Ook worden in dit model toestanden, parameters, inputs en outputs, welke fundamenteel verschillende modelgrootheden zijn, onderscheiden. Tenslotte is een model in toestandsvorm direct geschikt voor analyse middels standaardtechnieken en gebruik van optimale besturingsalgoritmen.

Het bleek echter dat het model in toestandsvorm een stelsel vergelijkingen bevat dat stijf is. Dit leidt tot divergentie bij numerieke integratie. In de fysica is deze problematiek nauwkeurig bestudeerd en heeft het geleid tot numerieke oplossingen die op een intelligente manier de fysische interpretatie van het model gebruiken bij het numeriek oplossen. In dit onderzoek is uiteindelijk het SIMPLE algoritme, wat in deze categorie valt, nauwkeurig bestudeerd en geprogrammeerd. Dit heeft veel inzicht gegeven in zowel de fysica als de numerieke problematiek.

Wanneer het SIMPLE algoritme wordt toegepast blijkt de oplossing ervan fysisch goed verklaarbaar. Zo blijkt een initieel aangebrachte warme spot in verloop van tijd te stijgen en een koude spot te dalen. Daarnaast blijkt dat luchtstroming circuleert in een gesloten ruimte wat eveneens fysisch verklaarbaar is aangezien de luchtstroming nabij de wanden een andere weg zal moeten vinden.

Vervolgens is het model gevalideerd met het officiële CFD pakket genaamd FLUENT. Het blijkt dat de oplossing van het model redelijk goed overeenkomt met de oplossing van FLUENT. Hiervoor moet wel een kleinere discretisatie tijd worden gekozen ten opzichte van FLUENT. Daarnaast heeft het model langere rekentijden nodig dan FLUENT.

8.2 Aanbevelingen

Het model wat in dit onderzoek is ontwikkeld kan de basis zijn voor een uitgebreider model met daarbij inbegrepen alle transportverschijnselen binnen en buiten een kas. Bijvoorbeeld de invloed van de zon, gewassen, CO₂ gehalte en luchtvochtigheid. Om een dergelijk model te

realiseren zullen er bepaalde aspecten aan het model moeten worden verbeterd, voornamelijk de rekentijd en de nauwkeurigheid.

8.2.1 Rekentijd

Wanneer bij het model een fijn grid wordt gekozen, dat wil zeggen een grid van bijvoorbeeld 25 bij 25, wordt de rekentijd te lang. De rekentijd kan worden ingekort door de programmeerstructuur intelligenter in te richten. Zo kan aan de Matlab code de eigenschap worden toegevoegd dat er alleen gerekend wordt aan plaatsen in het grid waar daadwerkelijk veranderingen optreden.

8.2.2 Nauwkeurigheid

De nauwkeurigheid van het model kan worden verhoogd door te kiezen voor een andere benaderingsmethode dan de differentie benadering in 5.3. De nauwkeurigheid van de oplossing zal stijgen wanneer wordt gekozen voor hogere orde differentie methodes waarbij een oplossing wordt benaderd met meerdere omliggende punten.

8.2.3 Optimal Control

Voor rechtstreekse toepassing van standaard technieken voor analyse alsook optimale besturing is het noodzakelijk dat het model in toestandsvorm wordt gerepresenteerd. In dit verslag is deze representatie daarom afgeleid. Omdat het model stijf is divergeren standaard numerieke integratie technieken. Bij dit onderzoek is besloten om eerst de gebruikelijke oplossingen van dit probleem op basis van fysische interpretaties van het model goed te bestuderen. Dit soort oplossingen staan meestal aan de basis van commercieel verkrijgbare software bekend onder de verzamelnaam CFD (Computational Fluid Dynamics). In dit geval is het SIMPLE algoritme uitgebreid bestudeerd en geprogrammeerd.

De problematiek van stijfheid wordt bij toestandsmodellen ondervangen middels quasi-statische benadering. Hoewel het SIMPLE algoritme een andere modelrepresentatie gebruikt is de oplossing ook hier gebaseerd op een quasi-statische benadering. Deze benadering vindt hier plaats op basis van een intelligente fysische interpretatie die deels gebruik maakt van de specifieke model structuur. De benutting van deze structuur maakt dat men afwijkt van de structuur van het model in toestandsvorm. Niettemin bevat het SIMPLE algoritme vermoedelijk belangrijke aanwijzingen voor het intelligent quasi-statisch benaderen en numeriek oplossen van een model in toestandsvorm. Dit is de volgende stap die in het onderzoek moet worden gezet.

Hoofdstuk 9

Literatuur

- [1] H. K. Versteeg, W. Malalasekera (1995), *An introduction to Computational Fluid Dynamics*.
- [2] Jonathan A. Dantzig, Charles L. Tucker III (2001), *Modeling in Materials Processing*.
- [3] Joseph hilsenrath, Charles W. Beckett, William S. Benedict, Lilla Fano, Harold J. Hoge Joseph F. Masi, Ralph L. Nutall, Yeram S. Touloukian, Harold W. Woolley (1955) *Tables of Thermal Properties of Gases*.
- [4] T. Peekstok, F. Kempkes, J.B. Campen, T. H. Gielink, E. Heuvelink (2008), *Monitor project (invloed van semigesloten systemen op het kas klimaat)*.
- [5] Suhas V. Patankar (1980), *Numerical Heat Transfer and Fluid Flow*.
- [6] R. J. C. van Ooteghem (2007), *Optimal Control Design for a Solar Greenhouse*
- [7] E. J. van Henten (1994), *Greenhouse Climate Management: An Optimal Control Approach*.
- [8] A. A. Sapounas, T. Bartzanas, C. Nikita-Martzopoulou, C. Kittas (2008), *Aspects of CFD Modelling of a Fan and Pad Evaporative Cooling System in Greenhouses*.
- [9] S. L. Sinha, R. C. Arora, Subhransu Roy (1999), *Numerical simulation of two-dimensional room air flow with and without buoyancy*.
- [10] A. W. M. (Jos) van Schijndel (2003), *Advanced HVAC modeling with FemLab/Simulink/MatLab*.
- [11] Nutsbedrijf Westland N.V. (1995), *Handboek verwarming glastuinbouw*.
- [12] L.G. van Willigenburg, *Lecture notes of Signals and systems modelling: SCO-20306*
- [13] K. C. Karki, S. V. Patankar (1988), *Calculation Procedure for Viscous Incompressible Flows in Complex Geometries*.
- [14] Qi-Hong Deng, Jiemin Zhou, Chi Mei, Yong-Ming Shen (2004), *Fluid, heat and contaminant transport structures of laminar double-diffusive mixed convection in a two-dimensional ventilated enclosure*.

Hoofdstuk 10

Bijlagen

10.1 Symbolenlijst

<i>Symbol</i>	<i>Grootheid</i>	<i>Eenheid</i>
x	Afstand in de x -richting	[m]
y	Afstand in de y -richting	[m]
z	Afstand in de z -richting	[m]
\mathbf{e}_x	Eenheidsvector in de x -richting	
\mathbf{e}_y	Eenheidsvector in de y -richting	
\mathbf{e}_z	Eenheidsvector in de z -richting	
\mathbf{x}	Positievector	[m]
\mathbf{v}	Vectoriele snelheid	[m/s]
g	Versnelling door gravitatie	[m s ⁻²]
$\frac{D}{Dt}$	Materiële afgeleide	[s ⁻¹]
∇	Nabla-operator	[m ⁻¹]
u	Snelheid in de x -richting	[m s ⁻¹]
v	Snelheid in de y -richting	[m s ⁻¹]
w	Snelheid in de z -richting	[m s ⁻¹]
t	Tijd	[s]
p	Hydrostatische druk	[N m ⁻²]
$\boldsymbol{\tau}$	Schuifspanningstensor	
\mathbf{b}	'Body' kracht vector die aangrijpt op coördinaatpunt \mathbf{x}	
c_p	Soortelijke warmte	[J kg ⁻¹ K ⁻¹]
L_p	'Latent heat of pressure change'	[m ³ kg ⁻¹]
q_i	Warmte flux in de i -richting	[N m ⁻¹ s ⁻¹]
\dot{R}	Interne warmtebron	[J kg ⁻¹ s ⁻¹]
T	Temperatuur	[K]
ρ_0	Referentiedichtheid	[kg m ⁻³]
T_0	Referentietemperatuur	[K]
β	Volmetrische thermische expansie coefficient	[K ⁻¹]
h	Hoogte, gemeten in de z -richting	[m]
α	Onderrelaxatiefactor	

10.2 Quasi-statisch maken van vergelijkingen

Wanneer een stelsel vergelijkingen stijf is, zoals het stelsel (4.5), kan dit worden verholpen door de snel veranderende toestanden quasi-statisch te veronderstellen. In (4.5) wordt de druk \hat{p} dan quasi statisch verondersteld:

$$\frac{\partial \hat{p}}{\partial t} = f(T(t), u(t), v(t), w(t), \hat{p}) = 0 \quad (10.1)$$

Een vergelijkbare berekening zit verpakt in het SIMPLE algoritme. Dit algoritme bevat een iteratiemethode waarin het drukveld steeds nauwkeuriger wordt benaderd door gebruik te maken van de continuïteitsvergelijking (4.7a). In hoofdstuk 7 is het SIMPLE algoritme tot in detail uitgewerkt.

10.3 Afgeleide termen in impulsvergelijkingen

Wat opvalt in (4.7) is dat de snelheden die eerst voor de afgeleide termen stonden nu binnen de termen staan. In deze paragraaf wordt uitgelegd waarom er wiskundig hetzelfde staat. Als voorbeeld wordt vergelijking (4.7b) in de z -richting genomen:

$$\rho_0 \left(\frac{\partial w}{\partial t} + \frac{\partial uw}{\partial x} + \frac{\partial vw}{\partial y} + \frac{\partial ww}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + \rho_0 \beta (T - T_0) g \quad (10.2)$$

Deze vergelijking kan als volgt worden geschreven:

$$\rho_0 A = -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + \rho_0 \beta (T - T_0) g \quad (10.3)$$

$$\text{met } A = \frac{\partial w}{\partial t} + \frac{\partial uw}{\partial x} + \frac{\partial vw}{\partial y} + \frac{\partial ww}{\partial z} \quad (10.4)$$

Wanneer de productregel wordt toegepast op vergelijking (10.4) kan deze als volgt worden uitgeschreven:

$$A = \frac{\partial w}{\partial t} + \frac{\partial u}{\partial x} w + u \frac{\partial w}{\partial x} + \frac{\partial v}{\partial y} w + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} + \frac{\partial w}{\partial z} w \quad (10.5)$$

en worden herschreven tot:

$$A = \frac{\partial w}{\partial t} + w \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \quad (10.6)$$

Vervolgens wordt de continuïteitsvergelijking (4.7a) ingevoerd:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (10.7)$$

De continuïteitsvergelijking wordt gesubstitueerd in (10.6) waardoor de volgende vergelijking overblijft:

$$A = \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \quad (10.8)$$

Nu kan A weer worden gesubstitueerd in (10.3) waardoor de volgende vergelijking ontstaat:

$$\rho_0 \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = -\frac{\partial \hat{p}}{\partial z} + \mu \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) + \rho_0 \beta (T - T_0) g \quad (10.9)$$

10.4 Randvoorwaarden

10.4.1 Linker wand

a) Impulsvergelijking x-richting (5.27a):

$$u_{i-1,J} = 0 \quad (10.10)$$

b) Impulsvergelijking z-richting (5.27b):

$$u_{i,J}^{(n-1)} = u_{i,J-1}^{(n-1)} = 0 \quad (10.11)$$

$$w_{I-1,j} = 0 \quad (10.12)$$

$$D_c = \frac{\mu}{\frac{1}{2} \delta x} \cdot \delta z = 2 \frac{\mu}{\delta x} \cdot \delta z \quad (10.13)$$

c) Energievergelijking (5.27d):

$$F_c = 0 \quad (10.14)$$

$$T_c = T_w \quad (10.15)$$

$$D_c = \frac{k_w}{\frac{1}{2} \delta x} \cdot \delta z = 2 \frac{k_w}{\delta x} \cdot \delta z \quad (10.16)$$

met k_w de warmtegeleidingcoëfficiënt in de wand.

10.4.2 Rechter wand

a) Impulsvergelijking x-richting (5.27a):

$$u_{i+1,J} = 0 \quad (10.17)$$

b) Impulsvergelijking z-richting (5.27b):

$$u_{i+1,J}^{(n-1)} = u_{i+1,J-1}^{(n-1)} = 0 \quad (10.18)$$

$$w_{I+1,j} = 0 \quad (10.19)$$

$$D_e = \frac{\mu}{\frac{1}{2}\delta x} \cdot \delta z = 2 \frac{\mu}{\delta x} \cdot \delta z \quad (10.20)$$

c) Energievergelijking (5.27d):

$$F_e = 0 \quad (10.21)$$

$$T_e = T_w \quad (10.22)$$

$$D_e = \frac{k_w}{\frac{1}{2}\delta x} \cdot \delta z = 2 \frac{k_w}{\delta x} \cdot \delta z \quad (10.23)$$

10.4.3 Boven wand

a) Impulsvergelijking x-richting (5.27a):

$$w_{I-1,j+1}^{(n-1)} = w_{I,j+1}^{(n-1)} = 0 \quad (10.24)$$

$$u_{i,J+1} = 0 \quad (10.25)$$

$$D_n = \frac{\mu}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{\mu}{\delta z} \cdot \delta x \quad (10.26)$$

b) Impulsvergelijking z-richting (5.27b):

$$w_{I,j+1} = 0 \quad (10.27)$$

c) Energievergelijking (5.27d):

$$F_n = 0 \quad (10.28)$$

$$T_n = T_w \quad (10.29)$$

$$D_n = \frac{k_w}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{k_w}{\delta z} \cdot \delta x \quad (10.30)$$

10.4.4 Onder wand

a) Impulsvergelijking x-richting (5.27a):

$$w_{I-1,j}^{(n-1)} = w_{I,j}^{(n-1)} = 0 \quad (10.31)$$

$$u_{i,J-1} = 0 \quad (10.32)$$

$$D_s = \frac{\mu}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{\mu}{\delta z} \cdot \delta x \quad (10.33)$$

b) Impulsvergelijking z-richting (5.27b):

$$w_{I,j-1} = 0 \quad (10.34)$$

c) Energievergelijking (5.27d):

$$F_s = 0 \quad (10.35)$$

$$T_s = T_w \quad (10.36)$$

$$D_s = \frac{k_w}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{k_w}{\delta z} \cdot \delta x \quad (10.37)$$

10.4.5 Linker bovenhoek

a) Impulsvergelijking x-richting (5.27a):

$$w_{I-1,j+1}^{(n-1)} = w_{I,j+1}^{(n-1)} = 0 \quad (10.38)$$

$$u_{i-1,J} = 0 \quad (10.39)$$

$$u_{i,J+1} = 0 \quad (10.40)$$

$$D_n = \frac{\mu}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{\mu}{\delta z} \cdot \delta x \quad (10.41)$$

b) Impulsvergelijking z-richting (5.27b):

$$u_{i,J}^{(n-1)} = u_{i,J-1}^{(n-1)} = 0 \quad (10.42)$$

$$w_{I-1,j} = 0 \quad (10.43)$$

$$w_{I,j+1} = 0 \quad (10.44)$$

$$D_c = \frac{\mu}{\frac{1}{2}\delta x} \cdot \delta z = 2 \frac{\mu}{\delta x} \cdot \delta z \quad (10.45)$$

c) Energievergelijking (5.27d):

$$F_c = 0 \quad (10.46)$$

$$F_n = 0 \quad (10.47)$$

$$T_c = T_w \quad (10.48)$$

$$T_n = T_w \quad (10.49)$$

$$D_c = \frac{k_w}{\frac{1}{2}\delta x} \cdot \delta z = 2 \frac{k_w}{\delta x} \cdot \delta z \quad (10.50)$$

$$D_n = \frac{k_w}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{k_w}{\delta z} \cdot \delta x \quad (10.51)$$

10.4.6 Rechter bovenhoek

a) Impulsvergelijking x-richting (5.27a):

$$w_{I-1,j+1}^{(n-1)} = w_{I,j+1}^{(n-1)} = 0 \quad (10.52)$$

$$u_{i+1,J} = 0 \quad (10.53)$$

$$u_{i,J+1} = 0 \quad (10.54)$$

$$D_n = \frac{\mu}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{\mu}{\delta z} \cdot \delta x \quad (10.55)$$

b) Impulsvergelijking z-richting (5.27b):

$$u_{i+1,J}^{(n-1)} = u_{i+1,J-1}^{(n-1)} = 0 \quad (10.56)$$

$$w_{I+1,j} = 0 \quad (10.57)$$

$$w_{I,j+1} = 0 \quad (10.58)$$

$$D_e = \frac{\mu}{\frac{1}{2}\delta x} \cdot \delta z = 2 \frac{\mu}{\delta x} \cdot \delta z \quad (10.59)$$

c) Energievergelijking (5.27d):

$$F_e = 0 \quad (10.60)$$

$$F_n = 0 \quad (10.61)$$

$$T_e = T_w \quad (10.62)$$

$$T_n = T_w \quad (10.63)$$

$$D_e = \frac{k_w}{\frac{1}{2}\delta x} \cdot \delta z = 2 \frac{k_w}{\delta x} \cdot \delta z \quad (10.64)$$

$$D_n = \frac{k_w}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{k_w}{\delta z} \cdot \delta x \quad (10.65)$$

10.4.7 Rechter onderhoek

a) Impulsvergelijking x-richting (5.27a):

$$w_{I-1,j}^{(n-1)} = w_{I,j}^{(n-1)} = 0 \quad (10.66)$$

$$u_{i+1,J} = 0 \quad (10.67)$$

$$u_{i,j-1} = 0 \quad (10.68)$$

$$D_s = \frac{\mu}{\frac{1}{2} \delta z} \cdot \delta x = 2 \frac{\mu}{\delta z} \cdot \delta x \quad (10.69)$$

b) Impulsvergelijking z-richting (5.27b):

$$u_{i+1,j}^{(n-1)} = u_{i+1,j-1}^{(n-1)} = 0 \quad (10.70)$$

$$w_{I+1,j} = 0 \quad (10.71)$$

$$w_{I,j-1} = 0 \quad (10.72)$$

$$D_e = \frac{\mu}{\frac{1}{2} \delta x} \cdot \delta z = 2 \frac{\mu}{\delta x} \cdot \delta z \quad (10.73)$$

c) Energievergelijking (5.27d):

$$F_e = 0 \quad (10.74)$$

$$F_s = 0 \quad (10.75)$$

$$T_e = T_w \quad (10.76)$$

$$T_s = T_w \quad (10.77)$$

$$D_e = \frac{k_w}{\frac{1}{2} \delta x} \cdot \delta z = 2 \frac{k_w}{\delta x} \cdot \delta z \quad (10.78)$$

$$D_s = \frac{k_w}{\frac{1}{2} \delta z} \cdot \delta x = 2 \frac{k_w}{\delta z} \cdot \delta x \quad (10.79)$$

10.4.8 Rechter onderhoek

a) Impulsvergelijking x-richting (5.27a):

$$w_{I-1,j}^{(n-1)} = w_{I,j}^{(n-1)} = 0 \quad (10.80)$$

$$u_{i-1,j} = 0 \quad (10.81)$$

$$u_{i,j-1} = 0 \quad (10.82)$$

$$D_s = \frac{\mu}{\frac{1}{2} \delta z} \cdot \delta x = 2 \frac{\mu}{\delta z} \cdot \delta x \quad (10.83)$$

b) Impulsvergelijking z-richting (5.27b):

$$u_{i,j}^{(n-1)} = u_{i,j-1}^{(n-1)} = 0 \quad (10.84)$$

$$w_{I-1,j} = 0 \quad (10.85)$$

$$w_{I,j-1} = 0 \quad (10.86)$$

$$D_c = \frac{\mu}{\frac{1}{2} \delta x} \cdot \delta z = 2 \frac{\mu}{\delta x} \cdot \delta z \quad (10.87)$$

c) Energievergelijking (5.27d):

$$F_c = 0 \quad (10.88)$$

$$F_s = 0 \quad (10.89)$$

$$T_c = T_w \quad (10.90)$$

$$T_s = T_w \quad (10.91)$$

$$D_c = \frac{k_w}{\frac{1}{2}\delta x} \cdot \delta z = 2 \frac{k_w}{\delta x} \cdot \delta z \quad (10.92)$$

$$D_s = \frac{k_w}{\frac{1}{2}\delta z} \cdot \delta x = 2 \frac{k_w}{\delta z} \cdot \delta x \quad (10.93)$$

10.5 Matlab Model

10.5.1 Hoofd programma

```

%% MAIN PROGRAMME
%       This is the Main Programme of the Model. In this programme
%       all parameters are created, the grid, initial values,
%       iteration factors and criterium are defined and the SIMPLE
%       algorithm is executed. Within the SIMPLE algorithm other
%       sub-programmes are requested, like:
%       - 'Usterveld.m'
%       - 'Wsterveld.m'
%       - 'PressureCorrection.m'
%       - 'PressurefieldCorrection.m'
%       - 'UfieldCorrection.m'
%       - 'WfieldCorrection.m'
%       - 'TemperatureField.m'
%       - 'Plotfunction.m'

%% -CLEAR ALL VARIABLES & CLOSE FIGURES-
%       In this part all variables in the memory section will be deleted.

clear all           % Clear all variables
close all          % Close all figures

%% -DEFINE PARAMETERS-
%       In this part the parameters of the model will be defined.

timemax = 16;           % Maximum time [s]
Dt = 0.01;             % Time step [s]
transitionperiod = 0.1; % Transition period: period needed for smaller time step [s]
Dttransition = 0.01;   % Time step within transition period [s]
dt = Dttransition;     % Set dt to Dttransition

dx = 0.5;              % Delta x [m]
dz = 0.5;              % Delta z [m]
Length = 5;           % Length [m] in x-direction of the 2D space
Height = 5;           % Height [m] in z-direction of the 2D space

mu = 1.79e-5; %3.5e-7; % Dynamic viscosity [kg m^-1 s^-1]
g = 9.81;          % Gravity constant [m s^-2]
beta = 0.0034;    % Volumetric Thermal Expansion Coefficient [K^-1]
T0 = 20;          % Reference Temperature [K]
cp = 1006;        % Specific Heat [J kg^-1 K^-1]
kc = 0.0242;      % Thermal Conductivity [W m^-1 K^-1]
kcw = 0;          % Thermal Conductivity at the wall [W m^-1 K^-1]
Tw = 20;          % Wall Temperature
rho0 = 1.225;     % Reference Density [kg m^-3]

Q = 1/4*rho0;     % Often-used term [kg m^-3]

```

```

%% -DEFINE GRID-
%           In this part the grid sizes will be calculated from the
%           'Define Parameters' in the upper part.

Lengthi = Length/dx;           % Grid size in x-direction [m]
LengthJ = Height/dz;          % Grid size in z-direction [m]

%% -DEFINE INITIAL VALUES-
%           In this part the initial values of the model will be
%           determined

t = 0;                          % Initial time [s]

U = zeros(Lengthi+1,LengthJ);   % Estimated U velocity field in the x-direction [m/s]
W = zeros(Lengthi,LengthJ+1);   % Estimated W velocity field in the z-direction [m/s]

u = zeros(Lengthi+1,LengthJ);   % Initial U velocity field in the x-direction [m/s]
w = zeros(Lengthi,LengthJ+1);   % Initial W velocity field in the z-direction [m/s]

T = 20*ones(Lengthi,LengthJ);   % Initial temperature field [K]

Pster = zeros(Lengthi,LengthJ); % Estimated pressure field [N m^-2]

T(5:6,1:2) = 21;                % Deviant initial temperature [K]

%% -DEFINE ITERATION FACTORS & ITERATION CRITERIUM

alphaP = 1;                      % Pressure under-relaxation factor [-]
alphaU = 1;                      % U-velocity under-relaxation factor [-]
alphaW = 1;                      % W-velocity under-relaxation factor [-]

differenceU = 1;                  % Initial differenceU [m/s]; This difference is ...
%                               % the convergence criterium. The initial value must be higher than 1e-10.
differenceW = 1;                  % Initial differenceW [m/s]

iteratie = 1;                    % Start iteration number [-]

%% -SIMPLE ALGORITHM-
%           In this part the SIMPLE-algorithm will be executed. The
%           algorithm exist of an Inner and an Outer Loop.
%
%           Inner Loop:
%           First new estimations for the U-velocity and W-velocity fields, Uster and &
%           Wster, will be calculated from the estimated fields U, W and Pster with the
%           initial fields u, w and T.
%           With Uster and Wster the pressure correction field, Pcorr,
%           will be calculated. This pressure correction field adapts
%           the estimated correction field, Pster, to a new pressure
%           field P. Next, new velocity fields will be calculated, Unew
%           and Wnew, from the new pressure field P. These new velocity fields,
%           Unew and Wnew, will be the new estimated fields U and W in
%           the next iteration step. The same holds for Pster.
%           If differenceU and differenceW is smaller or equal to 1e-10
%           the algorithm goes to the outer loop.
%           Outer Loop:
%           A new temperature field, T, will be calculated from the initial
%           field T, and the new velocity fields U and W.
%           Next the fields u and w will be set to U and W. u and w are
%           the unchanged velocity fields in the inner loop. These
%           fields are only be upgraded in the outer loop when t
%           becomes t + dt.

while t < timemax                % Outer Loop; executed while t is smaller than timemax

    while differenceU > 1e-10 && differenceW > 1e-10
% Inner Loop; exexuted while differenceU and differenceW are greater than 1e-10

        Uster = Usterveld(U,W,u,w,T,Pster,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ);
%                               % Calculation of Uster with 'Usterveld.m'
        Wster = Wsterveld(U,W,u,w,T,Pster,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ);
%                               % Calculation of Wster with 'Wsterveld.m'

        Pcorr =
PressureCorrection(U,W,u,w,T,Pster,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ,Uster,Wster);
%                               % Calculation of Pcorr with 'PressureCorrection.m'

        P = PressurefieldCorrection(Pster,Pcorr,alphaP);

```

```

% Calculation of P with 'PressurefieldCorrection.m'

    Unew =
UfieldCorrection(U,W,u,w,T,P,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ,alphaU);
% Calculation of Unew with 'UfieldCorrection.m'

    Wnew =
WfieldCorrection(U,W,u,w,T,P,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ,alphaW);
% Calculation of Wnew with 'WfieldCorrection.m'

    differenceU = max(max(abs(Unew-U)));
% Calculation of the difference between the new field and the old field (previous iteration
step).
    differenceW = max(max(abs(Wnew-W)));

    U = Unew; % Set U to Unew
    W = Wnew; % Set W to Wnew
    Pster = P; % Set Pster to P

end

T = TemperatureField(U,W,T,Tw,dx,dz,dt,mu,rho0,Lengthi,LengthJ,kc,kcw,cp);
% Calculation of T with 'TemperatureField.m'

u = U; % Set u to U
w = W; % Setu w to W

if t > transitionperiod
    dt = Dt;
end

t = t + dt % Set t to t + dt

Plotfunction(t,U,W,T,Lengthi,LengthJ) % Plot fields with 'Plotfunction.m'

    iteratie = 1; % Set iteratie back to 1
    differenceU = 1; % Set differenceU back to 1
    differenceW = 1; % Set differenceW back to 1
end

%% -TRANSPOSE & PLOT FIELDS-

Horizontalesnelheid = flipud(U'); % Transpose U field to a logical viewable form
Verticalesnelheid = flipud(W'); % Transpose W field to a logical viewable form
Temperatuur = flipud(T'); % Transpose T field to a logical viewable form
Plotfunction(t,U,W,T,Lengthi,LengthJ) % Plot fields with 'Plotfunction.m'

```

10.5.2 Deelprogramma 1: berekenen u *

```

%% CALCULATION OF Uster FIELD
%
% In this pogramme the values of Uster field will be
% calculated. Each velocity point in the grid exists of an
% equation established from arounded velocity points in the grid.
% All equations together form the matrix F en vector C:
% F*Uster = C.
% So Uster = F\C.
% To establish the matrix F, first the matrix B will be
% established. Matrix F is a compact version of matrix B.
% Matrix F does not contain the boundary velocity equations
% from B in which all the elements are equal to zero.
% Each equation in B is build up from the matrix A. The matrix A is
% a matrix which represents all the velocity points of
% the grid. All elements of a new equation will be
% represented in A. After forming the equation, the total
% A-matrix will form a new row in B. So, the program contains
% a for loop in which the A matrix will be reshaped as a row.

%% -CALL PARAMETERS AND CREATE MATRICES-

function [Uster] = Usterveld(U,W,u,w,T,Pster,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ)

A = zeros(Lengthi,LengthJ);

```

```

B = zeros((Lengthi-1)*LengthJ,Lengthi*LengthJ);
F = [];
C = [];
q = 1:(Lengthi*LengthJ);
k = 1;

%% -CENTRE (COMMON EQUATIONS)-

for J = 2:LengthJ-1
    j = J;
    for i = 3:Lengthi-1
        I = i;
        A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)...
+U(i-1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
        A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
        A(i+1,J) = -(-Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
        A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
        A(i,J+1) = -(-Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -LEFT BOUNDARY-

for J = 2:LengthJ-1
    j = J;
    for i = 2
        I = i;
        A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+0)+Q*dx*(W(I,j+1)...
+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
        % --> A(i-1,J) = 0
        A(i+1,J) = -(-Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
        A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
        A(i,J+1) = -(-Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -RIGHT BOUNDARY-

for J = 2:LengthJ-1
    j = J;
    for i = Lengthi
        I = i;
        A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(0+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(W(I,j+1)...
+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
        A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
        % --> A(i+1,J) = 0
        A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
        A(i,J+1) = -(-Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

```



```

        k = k+1;
    end
end

%% -UNDER BOUNDARY-

for J = 1
    j = J;
    for i = 3:Lengthi-1
        I = i;
        A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)...
+U(i-1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(0+0)+2*mu*(dz/dx)+3*mu*(dx/dz);
        A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
        A(i+1,J) = -(-Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
        % --> A(i,J-1) = 0
        A(i,J+1) = -(-Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -UPPER BOUNDARY-

for J = LengthJ
    j = J;
    for i = 3:Lengthi-1
        I = i;
        A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))...
+Q*dx*(0+0)-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+3*mu*(dx/dz);
        A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
        A(i+1,J) = -(-Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
        A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
        % --> A(i,J+1) = 0

        B(k,:) = [A(q)];

        Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -UPPER LEFT CORNER-

J = LengthJ;
j = J;
i = 2;
I = i;

A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+0)+Q*dx*(0+0)-Q*dx*(W(I,j)...
+W(I-1,j))+2*mu*(dz/dx)+3*mu*(dx/dz);
% --> A(i-1,J) = 0
A(i+1,J) = -(-Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
% --> A(i,J+1) = 0

B(k,:) = [A(q)];

Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

```

```

%% -UPPER RIGHT CORNER-

J = LengthJ;
j = J;
i = Lengthi;
I = i;

A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(0+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(0+0)-Q*dx*(W(I,j)...
+W(I-1,j))+2*mu*(dz/dx)+3*mu*(dx/dz);
A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
% --> A(i+1,J) = 0
A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
% --> A(i,J+1) = 0

B(k,:) = [A(q)];

Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UNDER LEFT CORNER-

J = 1;
j = J;
i = 2;
I = i;

A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+0)+Q*dx*(W(I,j+1)...
+W(I-1,j+1))-Q*dx*(0+0)+2*mu*(dz/dx)+3*mu*(dx/dz);
% --> A(i-1,J) = 0
A(i+1,J) = -(Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
% --> A(i,J-1) = 0
A(i,J+1) = -(Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

B(k,:) = [A(q)];

Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UNDER RIGHT CORNER-

J = 1;
j = J;
i = Lengthi;
I = i;

A(i,J) = rho0*((dx*dz)/dt)+Q*dz*(0+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(W(I,j+1)...
+W(I-1,j+1))-Q*dx*(0+0)+2*mu*(dz/dx)+3*mu*(dx/dz);
A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
% --> A(i+1,J) = 0
% --> A(i,J-1) = 0
A(i,J+1) = -(Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

B(k,:) = [A(q)];

Rightpart = dz*(Pster(I-1,J)-Pster(I,J))+rho0*((dx*dz)/dt)*u(i,J);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

clear k

%% -CALCULATION OF Uster FIELD-

for k = 2:Lengthi:Lengthi*LengthJ
    F = [F B(1:((Lengthi-1)*LengthJ),k:k+Lengthi-2)];

```

```

end

Uster = F\C;

k = 1;

for J = 1:LengthJ
    for i = 1:Lengthi-1
        UsterMatrix(i+1,J) = Uster(k);
        k = k+1;
    end
end

Uster = UsterMatrix;

```

10.5.3 Deelprogramma 2: berekenen w^*

```

%% CALCULATION OF Wster FIELD
%       In this pogramme the values of Wster field will be
%       calculated. Each velocity point in the grid exists of an
%       equation established from arounded velocity points in the grid.
%       All equations together form the matrix F en vector C:
%       F*Wster = C.
%       So Uster = F\C.
%       To establish the matrix F, first the matrix B will be
%       established. Matrix F is a compact version of matrix B.
%       Matrix F does not contain the boundary velocity equations
%       from B in which all the elements are equal to zero.
%       Each equation in B is build up from the matrix A. The matrix A is
%       a matrix which represents all the velocity points of
%       the grid. All elements of a new equation will be
%       represented in A. After forming the equation, the total
%       A-matrix will form a new row in B. So, the program contains
%       a for loop in which the A matrix will be reshaped as a row.

%% -CALL PARAMETERS AND CREATE MATRICES-

function [Wster] = Wsterveld(U,W,u,w,T,Pster,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ)

A = zeros(Lengthi,LengthJ);
B = zeros(Lengthi*(LengthJ-1),Lengthi*LengthJ);
F = [];
C = [];
q = 1:(Lengthi*LengthJ);
k = 1;

%% CENTRE (COMMON EQUATIONS)

for j = 3:LengthJ-1
    J = j;
    for I = 2:Lengthi-1
        i = I;
        A(I,j) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-
1))+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);
        A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
        A(I+1,j) = -(-Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
        A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
        A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dx*(Pster(I,J-1)-
Pster(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-T0)*dx*dz;

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -LEFT BOUNDARY-

```

```

for j = 3:LengthJ-1
    J = j;
    for I = 1
        i = I;
        A(I,j) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-
Q*dz*(0+0)+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+W(I,j-1))+3*mu*(dz/dx)+2*mu*(dx/dz);
        % --> A(I-1,j) = 0
        A(I+1,j) = -(-Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
        A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
        A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dx*(Pster(I,J-1)-
Pster(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-T0)*dx*dz;

        C = [C;Rightpart];

        A = zeros(LengthI,LengthJ);

        k = k+1;
    end
end

%% -Right boundary-

for j = 3:LengthJ-1
    J = j;
    for I = LengthI
        i = I;
        A(I,j) = rho0*((dx*dz)/dt)+Q*dz*(0+0)-Q*dz*(U(i,J)+U(i,J-1))+Q*dx*(W(I,j)+W(I,j+1))-
Q*dx*(W(I,j)+W(I,j-1))+3*mu*(dz/dx)+2*mu*(dx/dz);
        A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
        % --> A(I+1,j) = 0
        A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
        A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dx*(Pster(I,J-1)-
Pster(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-T0)*dx*dz;

        C = [C;Rightpart];

        A = zeros(LengthI,LengthJ);

        k = k+1;
    end
end

%% -UNDER BOUNDARY-

for j = 2
    J = j;
    for I = 2:LengthI-1
        i = I;
        A(I,j) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-
1))+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+0)+2*mu*(dz/dx)+2*mu*(dx/dz);
        A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
        A(I+1,j) = -(-Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
        % --> A(I,j-1) = 0
        A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dx*(Pster(I,J-1)-
Pster(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-T0)*dx*dz;

        C = [C;Rightpart];

        A = zeros(LengthI,LengthJ);

        k = k+1;
    end
end

```

```

%% -UPPER BOUNDARY-

for j = LengthJ
    J = j;
    for I = 2:LengthI-1
        i = I;
        A(I, j) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1, J)+U(i+1, J-1))-Q*dz*(U(i, J)+U(i, J-1))+Q*dx*(W(I, j)+0)-Q*dx*(W(I, j)+W(I, j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);
        A(I-1, j) = -(Q*dz*(U(i, J)+U(i, J-1))+mu*(dz/dx));
        A(I+1, j) = -(-Q*dz*(U(i+1, J)+U(i+1, J-1))+mu*(dz/dx));
        A(I, j-1) = -(Q*dx*(W(I, j)+W(I, j-1))+mu*(dx/dz));
        % --> A(I, j+1) = 0

        B(k, :) = [A(q)];

        Rightpart = dx*(Pster(I, J-1)-Pster(I, J))+rho0*((dx*dz)/dt)*w(I, j)+rho0*g*beta*((T(I, J)+T(I, J-1))/2-T0)*dx*dz;

        C = [C;Rightpart];

        A = zeros(LengthI, LengthJ);

        k = k+1;
    end
end

%% -UPPER LEFT CORNER-

j = LengthJ;
J = j;
I = 1;
i = I;

A(I, j) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1, J)+U(i+1, J-1))-Q*dz*(0+0)+Q*dx*(W(I, j)+0)-Q*dx*(W(I, j)+W(I, j-1))+3*mu*(dz/dx)+2*mu*(dx/dz);
% --> A(I-1, j) = 0
A(I+1, j) = -(-Q*dz*(U(i+1, J)+U(i+1, J-1))+mu*(dz/dx));
A(I, j-1) = -(Q*dx*(W(I, j)+W(I, j-1))+mu*(dx/dz));
% --> A(I, j+1) = 0

B(k, :) = [A(q)];

Rightpart = dx*(Pster(I, J-1)-Pster(I, J))+rho0*((dx*dz)/dt)*w(I, j)+rho0*g*beta*((T(I, J)+T(I, J-1))/2-T0)*dx*dz;

C = [C;Rightpart];

A = zeros(LengthI, LengthJ);

k = k+1;

%% -UPPER RIGHT CORNER-

j = LengthJ;
J = j;
I = LengthI;
i = I;

A(I, j) = rho0*((dx*dz)/dt)+Q*dz*(0+0)-Q*dz*(U(i, J)+U(i, J-1))+Q*dx*(W(I, j)+0)-Q*dx*(W(I, j)+W(I, j-1))+3*mu*(dz/dx)+2*mu*(dx/dz);
A(I-1, j) = -(Q*dz*(U(i, J)+U(i, J-1))+mu*(dz/dx));
% --> A(I+1, j) = 0
A(I, j-1) = -(Q*dx*(W(I, j)+W(I, j-1))+mu*(dx/dz));
% --> A(I, j+1) = 0

B(k, :) = [A(q)];

Rightpart = dx*(Pster(I, J-1)-Pster(I, J))+rho0*((dx*dz)/dt)*w(I, j)+rho0*g*beta*((T(I, J)+T(I, J-1))/2-T0)*dx*dz;

C = [C;Rightpart];

A = zeros(LengthI, LengthJ);

k = k+1;

%% -UNDER LEFT CORNER-

```

```

j = 2;
J = j;
I = 1;
i = I;

A(I,j) = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(0+0)+Q*dx*(W(I,j)+W(I,j+1))-
Q*dx*(W(I,j)+0)+3*mu*(dz/dx)+2*mu*(dx/dz);
% --> A(I-1,j) = 0
A(I+1,j) = -(-Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
% --> A(I,j-1) = 0
A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

B(k,:) = [A(q)];

Rightpart = dx*(Pster(I,J-1)-Pster(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-
1))/2-T0)*dx*dz;

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% UNDER RIGHT CORNER

j = 2;
J = j;
I = Lengthi;
i = I;

A(I,j) = rho0*((dx*dz)/dt)+Q*dz*(0+0)-Q*dz*(U(i,J)+U(i,J-1))+Q*dx*(W(I,j)+W(I,j+1))-
Q*dx*(W(I,j)+0)+3*mu*(dz/dx)+2*mu*(dx/dz);
A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
% --> A(I+1,j) = 0
% --> A(I,j-1) = 0
A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

B(k,:) = [A(q)];

Rightpart = dx*(Pster(I,J-1)-Pster(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-
1))/2-T0)*dx*dz;

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

clear k

%% CALCULATION OF Wster FIELD

F = B(1:Lengthi*(LengthJ-1),Lengthi+1:(Lengthi*LengthJ));

Wster = F\C;

k = 1;

for j = 1:LengthJ-1
    for I = 1:Lengthi
        WsterMatrix(I,j+1) = Wster(k);
        k = k+1;
    end
end

Wster = WsterMatrix;

```

10.5.4 Deelprogramma 3: berekenen druk correctie vergelijking

```

%% CALCULATION OF Pcorr FIELD
%
%       In this pogramme the values of Pcorr field will be
%       calculated. Each pressure point in the grid exists of an
%       equation established from arounded pressure points in the grid.
%       All equations together form the matrix B en vector C:
%       B*Pcorr = C.
%       So Pcorr = B\C.
%       Each equation in B is build up from the matrix A. The matrix A is

```

```

%           a matrix which represents all the pressure points of
%           the grid. All elements of a new equation will be
%           represented in A. After forming the equation, the total
%           A-matrix will form a new row in B. So, the program contains
%           a for loop in which the A matrix will be reshaped as a row.

%% -CALL PARAMETERS AND CREATE MATRICES-

function [Pcorr] =
PressureCorrection(U,W,u,w,T,Pster,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ,Uster,Wster)

A = zeros(Lengthi,LengthJ);
B = zeros(Lengthi*LengthJ,Lengthi*LengthJ);
F = [];
C = [];
q = 1:(Lengthi*LengthJ);
k = 1;

%% -CENTRE (COMMON EQUATIONS)-

for J = 2:LengthJ-1
    j = J;
    for I = 2:Lengthi-1
        i = I;
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-
1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
        ailJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+2,J)+U(i+1,J))-
Q*dz*(U(i+1,J)+U(i,J))+Q*dx*(W(I+1,j+1)+W(I,j+1))-
Q*dx*(W(I+1,j)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-
1))+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);
        aiJ1 = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J+1)+U(i+1,J))-
Q*dz*(U(i,J+1)+U(i,J))+Q*dx*(W(I,j+1)+W(I,j+2))-
Q*dx*(W(I,j+1)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

        A(I,J) = (dz*dz)/ailJ+(dz*dz)/aiJ+(dx*dx)/aiJ1+(dx*dx)/aiJ;
        A(I+1,J) = -(dz*dz)/ailJ;
        A(I-1,J) = -(dz*dz)/aiJ;
        A(I,J+1) = -(dx*dx)/aiJ1;
        A(I,J-1) = -(dx*dx)/aiJ;

        B(k,:) = [A(q)];

        Rightpart = -rho0*((dx*dz*dz)/(ailJ*dt))*u(i+1,J)+rho0*((dx*dz*dz)/(aiJ*dt))*u(i,J)-
rho0*((dx*dz*dz)/(aiJ1*dt))*w(I,j+1)+rho0*((dx*dz*dz)/(aiJ*dt))*w(I,j) ...
        +((dx*dz*dz)/aiJ)*rho0*g*beta*(T(I,J)-T0)-((dx*dz*dz)/aiJ1)*rho0*g*beta*(T(I,J+1)-
T0)-dz*Uster(i+1,J)+dz*Uster(i,J)-dx*Wster(I,j+1)+dx*Wster(I,j);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -LEFT BOUNDARY-

for J = 2:LengthJ-1
    j = J;
    for I = 1
        i = I;
        % --> aiJ = 0
        ailJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+2,J)+U(i+1,J))-
Q*dz*(U(i+1,J)+U(i,J))+Q*dx*(W(I+1,j+1)+W(I,j+1))-
Q*dx*(W(I+1,j)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-
1))+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);
        aiJ1 = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J+1)+U(i+1,J))-
Q*dz*(U(i,J+1)+U(i,J))+Q*dx*(W(I,j+1)+W(I,j+2))-
Q*dx*(W(I,j+1)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

        A(I,J) = (dz*dz)/ailJ+(dx*dx)/aiJ1+(dx*dx)/aiJ;
        A(I+1,J) = -(dz*dz)/ailJ;
        % --> A(I-1,J) = 0
        A(I,J+1) = -(dx*dx)/aiJ1;
        A(I,J-1) = -(dx*dx)/aiJ;
    end
end

```

```

B(k,:) = [A(q)];

Rightpart = -rho0*((dx*dz*dz)/(ai1J*dt))*u(i+1,J)-
rho0*((dx*dx*dz)/(aiJ1*dt))*w(I,j+1)+rho0*((dx*dx*dz)/(aiJ*dt))*w(I,j) ...
+((dx*dx*dz)/aiJ)*rho0*g*beta*(T(I,J)-T0)-((dx*dx*dz)/aiJ1)*rho0*g*beta*(T(I,J+1)-
T0)-dz*Uster(i+1,J)+dz*0-dx*Wster(I,j+1)+dx*Wster(I,j);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;
end
end

%% -RIGHT BOUNDARY-

for J = 2:LengthJ-1
j = J;
for I = Lengthi
i = I;
aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-
1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
% --> ai1J = 0
aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-
1))+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);
aiJ1 = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J+1)+U(i+1,J))-
Q*dz*(U(i,J+1)+U(i,J))+Q*dx*(W(I,j+1)+W(I,j+2))-
Q*dx*(W(I,j+1)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

A(I,J) = (dz*dz)/aiJ+(dx*dx)/aiJ1+(dx*dx)/aiJ;
% --> A(I+1,J) = 0
A(I-1,J) = -(dz*dz)/aiJ;
A(I,J+1) = -(dx*dx)/aiJ1;
A(I,J-1) = -(dx*dx)/aiJ;

B(k,:) = [A(q)];

Rightpart = rho0*((dx*dz*dz)/(aiJ*dt))*u(i,J)-
rho0*((dx*dx*dz)/(aiJ1*dt))*w(I,j+1)+rho0*((dx*dx*dz)/(aiJ*dt))*w(I,j) ...
+((dx*dx*dz)/aiJ)*rho0*g*beta*(T(I,J)-T0)-((dx*dx*dz)/aiJ1)*rho0*g*beta*(T(I,J+1)-
T0)-dz*0+dz*Uster(i,J)-dx*Wster(I,j+1)+dx*Wster(I,j);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;
end
end

%% -UNDER BOUNDARY-

for J = 1
j = J;
for I = 2:Lengthi-1
i = I;
aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-
1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
ai1J = rho0*((dx*dz)/dt)+Q*dz*(U(i+2,J)+U(i+1,J))-
Q*dz*(U(i+1,J)+U(i,J))+Q*dx*(W(I+1,j+1)+W(I,j+1))-
Q*dx*(W(I+1,j)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
% --> aiJ = 0
aiJ1 = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J+1)+U(i+1,J))-
Q*dz*(U(i,J+1)+U(i,J))+Q*dx*(W(I,j+1)+W(I,j+2))-
Q*dx*(W(I,j+1)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

A(I,J) = (dz*dz)/ai1J+(dz*dz)/aiJ+(dx*dx)/aiJ1;
A(I+1,J) = -(dz*dz)/ai1J;
A(I-1,J) = -(dz*dz)/aiJ;
A(I,J+1) = -(dx*dx)/aiJ1;
% --> A(I,J-1) = 0

B(k,:) = [A(q)];

```



```

    Rightpart = -rho0*((dx*dz*dz)/(ai1J*dt))*u(i+1,J)+rho0*((dx*dz*dz)/(aiJ*dt))*u(i,J)-
rho0*((dx*dz*dz)/(aiJ1*dt))*w(I,j+1) ...
    -( (dx*dz*dz)/aiJ1)*rho0*g*beta*(T(I,J+1)-T0)-dz*Uster(i+1,J)+dz*Uster(i,J)-
dx*Wster(I,j+1)+dx*0;

    C = [C;Rightpart];

    A = zeros(LengthI,LengthJ);

    k = k+1;
end
end

%% -UPPER BOUNDARY-

for J = LengthJ
    j = J;
    for I = 2:LengthI-1
        i = I;
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-
1,J))+Q*dz*(W(I,j+1)+W(I-1,j+1))-Q*dz*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
        ai1J = rho0*((dx*dz)/dt)+Q*dz*(U(i+2,J)+U(i+1,J))-
Q*dz*(U(i+1,J)+U(i,J))+Q*dz*(W(I+1,j+1)+W(I,j+1))-
Q*dz*(W(I+1,j)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-
1))+Q*dz*(W(I,j)+W(I,j+1))-Q*dz*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);
        % --> aiJ1 = 0

        A(I,J) = (dz*dz)/ai1J+(dz*dz)/aiJ+(dx*dx)/aiJ;
        A(I+1,J) = -(dz*dz)/ai1J;
        A(I-1,J) = -(dz*dz)/aiJ;
        % --> A(I,J+1) = 0
        A(I,J-1) = -(dx*dx)/aiJ;

        B(k,:) = [A(q)];

        Rightpart = -
rho0*((dx*dz*dz)/(ai1J*dt))*u(i+1,J)+rho0*((dx*dz*dz)/(aiJ*dt))*u(i,J)+rho0*((dx*dz*dz)/(aiJ*dt))*w(I,j) ...
        +((dx*dz*dz)/aiJ)*rho0*g*beta*(T(I,J)-T0)-dz*Uster(i+1,J)+dz*Uster(i,J)-
dx*0+dx*Wster(I,j);

        C = [C;Rightpart];

        A = zeros(LengthI,LengthJ);

        k = k+1;
    end
end

%% -UPPER LEFT CORNER-

J = LengthJ;
j = J;
I = 1;
i = I;

% --> aiJ = 0
ai1J = rho0*((dx*dz)/dt)+Q*dz*(U(i+2,J)+U(i+1,J))-
Q*dz*(U(i+1,J)+U(i,J))+Q*dz*(W(I+1,j+1)+W(I,j+1))-
Q*dz*(W(I+1,j)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-
1))+Q*dz*(W(I,j)+W(I,j+1))-Q*dz*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);
% --> aiJ1 = 0

A(I,J) = (dz*dz)/ai1J+(dz*dz)/aiJ+(dx*dx)/aiJ1+(dx*dx)/aiJ;
A(I+1,J) = -(dz*dz)/ai1J;
% --> A(I-1,J) = 0
% --> A(I,J+1) = 0
A(I,J-1) = -(dx*dz)/aiJ;

B(k,:) = [A(q)];

Rightpart = -rho0*((dx*dz*dz)/(ai1J*dt))*u(i+1,J)+rho0*((dx*dz*dz)/(aiJ*dt))*w(I,j) ...
        +((dx*dz*dz)/aiJ)*rho0*g*beta*(T(I,J)-T0)-dz*Uster(i+1,J)+dx*Wster(I,j);

C = [C;Rightpart];

```

```

A = zeros (Lengthi,LengthJ);

k = k+1;

%% -UPPER RIGHT CORNER-

J = LengthJ;
j = J;
I = Lengthi;
i = I;

aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
% --> ai1J = 0
aij = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-1))+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);
% --> ai1j = 0

A(I,J) = (dz*dz)/ai1J+(dz*dz)/aiJ+(dx*dx)/aij1+(dx*dx)/aij;
% --> A(I+1,J) = 0
A(I-1,J) = -(dz*dz)/aiJ;
% --> A(I,J+1) = 0
A(I,J-1) = -(dx*dx)/aij;

B(k,:) = [A(q)];

Rightpart = rho0*((dx*dz*dz)/(aiJ*dt))*u(i,J)+rho0*((dx*dx*dz)/(aij*dt))*w(I,j) ...
            +(dx*dx*dz)/aij*rho0*g*beta*(T(I,J)-T0)+dz*Uster(i,J)+dx*Wster(I,j);

C = [C;Rightpart];

A = zeros (Lengthi,LengthJ);

k = k+1;

%% -UNDER LEFT CORNER-

J = 1;
j = J;
I = 1;
i = I;

% --> aiJ = 0
ai1J = rho0*((dx*dz)/dt)+Q*dz*(U(i+2,J)+U(i+1,J))-Q*dz*(U(i+1,J)+U(i,J))+Q*dx*(W(I+1,j+1)+W(I,j+1))-Q*dx*(W(I+1,j)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
% --> aiJ = 0
aij1 = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J+1)+U(i+1,J))-Q*dz*(U(i,J+1)+U(i,J))+Q*dx*(W(I,j+1)+W(I,j+2))-Q*dx*(W(I,j+1)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

A(I,J) = (dz*dz)/ai1J+(dz*dz)/aiJ+(dx*dx)/aij1+(dx*dx)/aij;
A(I+1,J) = -(dz*dz)/ai1J;
% --> A(I-1,J) = 0
A(I,J+1) = -(dx*dx)/aij1;
% --> A(I,J-1) = 0

B(k,:) = [A(q)];

Rightpart = -rho0*((dx*dz*dz)/(ai1J*dt))*u(i+1,J)-rho0*((dx*dx*dz)/(aij1*dt))*w(I,j+1) ...
            -(dx*dx*dz)/aij1*rho0*g*beta*(T(I,J+1)-T0)-dz*Uster(i+1,J)-dx*Wster(I,j+1);

C = [C;Rightpart];

A = zeros (Lengthi,LengthJ);

k = k+1;

%% -UNDER RIGHT CORNER-

J = 1;
j = J;
I = Lengthi;
i = I;

```

```

aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(W(I,j+1)+W(I-
1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);
% --> ai1J = 0
% --> aiJ = 0
aij1 = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J+1)+U(i+1,J))-
Q*dz*(U(i,J+1)+U(i,J))+Q*dx*(W(I,j+1)+W(I,j+2))-
Q*dx*(W(I,j+1)+W(I,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

A(I,J) = (dz*dz)/ai1J+(dz*dz)/aiJ+(dx*dx)/aij1+(dx*dx)/aiJ;
% --> A(I+1,J) = 0
A(I-1,J) = -(dz*dz)/aiJ;
A(I,J+1) = -(dx*dx)/aij1;
% --> A(I,J-1) = 0

B(k,:) = [A(q)];

Rightpart = rho0*((dx*dz*dz)/(aiJ*dt))*u(i,J)-rho0*((dx*dx*dz)/(aij1*dt))*w(I,j+1) ...
-(dx*dx*dz)/aij1*rho0*g*beta*(T(I,J+1)-T0)+dz*Uster(i,J)-dx*Wster(I,j+1);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

clear k

%% -CALCULATION OF Pcorr FIELD-

Pcorr = B\C;

k = 1;

for J = 1:LengthJ
    for I = 1:Lengthi
        PcorrMatrix(I,J) = Pcorr(k);
        k = k+1;
    end
end

Pcorr = PcorrMatrix;

```

10.5.5 Deelprogramma 4: berekenen nieuwe druk veld

```

%% CALCULATION OF Pcorr FIELD
%           In this programme P will be calculated by adding alphaP*Pcorr
%           to Pster. The parameter alphaP can be chosen between 0 en
%           1. Pcorr is calculated in the programme
%           'PressureCorrection.m'.

%% CALL PARAMETERS & CALCULATE P

function [P] = PressurefieldCorrection(Pster,Pcorr,alphaP)

% Pressure correction

P = Pster + alphaP*Pcorr;

```

10.5.6 Deelprogramma 5: berekenen nieuwe u veld

```

%% CALCULATION OF Unew FIELD
%           In this programme the values of Unew field will be
%           calculated. Each velocity point in the grid exists of an
%           equation established from arounded velocity points in the grid.
%           All equations together form the matrix F en vector C:
%           F*Unew = C.
%           So Unew = F\C.
%           To establish the matrix F, first the matrix B will be
%           established. Matrix F is a compact version of matrix B.
%           Matrix F does not contain the boundary velocity equations
%           from B in which all the elements are equal to zero.
%           Each equation in B is build up from the matrix A. The matrix A is
%           a matrix which represents all the velocity points of
%           the grid. All elements of a new equation will be
%           represented in A. After forming the equation, the total

```

```

%           A-matrix will form a new row in B. So, the program contains
%           a for loop in which the A matrix will be reshaped as a row.

%% -CALL PARAMETERS AND CREATE MATRICES-

function [Unew] =
UfieldCorrection(U,W,u,w,T,P,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ,alphaU);

A = zeros(Lengthi,LengthJ);
B = zeros((Lengthi-1)*LengthJ,Lengthi*LengthJ);
F = [];
C = [];
q = 1:(Lengthi*LengthJ);
k = 1;

%% -CENTRE (COMMON EQUATIONS)-

for J = 2:LengthJ-1
    j = J;
    for i = 3:Lengthi-1
        I = i;
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-
1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

        A(i,J) = aiJ/alphaU;
        A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
        A(i+1,J) = -(Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
        A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
        A(i,J+1) = -(Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-
alphaU)*(aiJ/alphaU)*U(i,J);           % in A(i,J) is already alphaU (line 16)

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -LEFT BOUNDARY-

for J = 2:LengthJ-1
    j = J;
    for i = 2
        I = i;
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+0)+Q*dx*(W(I,j+1)+W(I-
1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

        A(i,J) = aiJ/alphaU;
        % --> A(i-1,J) = 0
        A(i+1,J) = -(Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
        A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
        A(i,J+1) = -(Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-
alphaU)*(aiJ/alphaU)*U(i,J);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -RIGHT BOUNDARY-

for J = 2:LengthJ-1

```

```

j = J;
for i = Lengthi
    I = i;
    aiJ = rho0*((dx*dz)/dt)+Q*dz*(0+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+2*mu*(dx/dz);

    A(i,J) = aiJ/alphaU;
    A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
    % --> A(i+1,J) = 0
    A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
    A(i,J+1) = -(Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

    B(k,:) = [A(q)];

    Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-alphaU)*(aiJ/alphaU)*U(i,J);

    C = [C;Rightpart];

    A = zeros(Lengthi,LengthJ);

    k = k+1;
end
end

%% -UNDER BOUNDARY-

for J = 1
    j = J;
    for i = 3:Lengthi-1
        I = i;
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(0+0)+2*mu*(dz/dx)+3*mu*(dx/dz);

        A(i,J) = aiJ/alphaU;
        A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
        A(i+1,J) = -(Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
        % --> A(i,J-1) = 0
        A(i,J+1) = -(Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-alphaU)*(aiJ/alphaU)*U(i,J);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -UPPER BOUNDARY-

for J = LengthJ
    j = J;
    for i = 3:Lengthi-1
        I = i;
        aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(0+0)-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+3*mu*(dx/dz);

        A(i,J) = aiJ/alphaU;
        A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
        A(i+1,J) = -(Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
        A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
        % --> A(i,J+1) = 0

        B(k,:) = [A(q)];

        Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-alphaU)*(aiJ/alphaU)*U(i,J);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

```

```

        k = k+1;
    end
end

%% -UPPER LEFT CORNER-

J = LengthJ;
j = J;
i = 2;
I = i;

aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+0)+Q*dx*(0+0)-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+3*mu*(dx/dz);

A(i,J) = aiJ/alphaU;
% --> A(i-1,J) = 0
A(i+1,J) = -(-Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));
% --> A(i,J+1) = 0

B(k,:) = [A(q)];

Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-alphaU)*(aiJ/alphaU)*U(i,J);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UPPER RIGHT CORNER-

J = LengthJ;
j = J;
i = Lengthi;
I = i;

aiJ = rho0*((dx*dz)/dt)+Q*dz*(0+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(0+0)-Q*dx*(W(I,j)+W(I-1,j))+2*mu*(dz/dx)+3*mu*(dx/dz);

A(i,J) = aiJ/alphaU;
% --> A(i+1,J) = 0
A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
% --> A(i,J+1) = 0
A(i,J-1) = -(Q*dx*(W(I,j)+W(I-1,j))+mu*(dx/dz));

B(k,:) = [A(q)];

Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-alphaU)*(aiJ/alphaU)*U(i,J);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UNDER LEFT CORNER-

J = 1;
j = J;
i = 2;
I = i;

aiJ = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i,J))-Q*dz*(U(i,J)+0)+Q*dx*(W(I,j+1)+W(I-1,j+1))-Q*dx*(0+0)+2*mu*(dz/dx)+3*mu*(dx/dz);

A(i,J) = aiJ/alphaU;
% --> A(i-1,J) = 0
A(i+1,J) = -(-Q*dz*(U(i+1,J)+U(i,J))+mu*(dz/dx));
% --> A(i,J-1) = 0
A(i,J+1) = -(-Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));

B(k,:) = [A(q)];

Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-alphaU)*(aiJ/alphaU)*U(i,J);

```

```

C = [C;Rightpart];
A = zeros (Lengthi,LengthJ);
k = k+1;
%% -UNDER RIGHT CORNER-
J = 1;
j = J;
i = Lengthi;
I = i;
aiJ = rho0*((dx*dz)/dt)+Q*dz*(0+U(i,J))-Q*dz*(U(i,J)+U(i-1,J))+Q*dx*(W(I,j+1)+W(I-1,j+1))-
Q*dx*(0+0)+2*mu*(dz/dx)+3*mu*(dx/dz);
A(i,J) = aiJ/alphaU;
A(i-1,J) = -(Q*dz*(U(i,J)+U(i-1,J))+mu*(dz/dx));
% --> A(i+1,J) = 0
% --> A(i,J-1) = 0
A(i,J+1) = -(-Q*dx*(W(I,j+1)+W(I-1,j+1))+mu*(dx/dz));
B(k,:) = [A(q)];
Rightpart = dz*(P(I-1,J)-P(I,J))+rho0*((dx*dz)/dt)*u(i,J)+(1-alphaU)*(aiJ/alphaU)*U(i,J);
C = [C;Rightpart];
A = zeros (Lengthi,LengthJ);
clear k
%% -CALCULATION OF Uster FIELD-
for k = 2:Lengthi:Lengthi*LengthJ
    F = [F B(1:((Lengthi-1)*LengthJ),k:k+Lengthi-2)];
end
Unew = F\C;
k = 1;
for J = 1:LengthJ
    for i = 1:Lengthi-1
        UMatrix(i+1,J) = Unew(k);
        k = k+1;
    end
end
Unew = [UMatrix;zeros(1,LengthJ)];

```

10.5.7 Deelprogramma 6: berekenen nieuwe w veld

```

%% CALCULATION OF Wnew FIELD
%           In this pogramme the values of Wnew field will be
%           calculated. Each velocity point in the grid exists of an
%           equation established from arounded velocity points in the grid.
%           All equations together form the matrix F en vector C:
%           F*Wnew = C.
%           So Wnew = F\C.
%           To establish the matrix F, first the matrix B will be
%           established. Matrix F is a compact version of matrix B.
%           Matrix F does not contain the boundary velocity equations
%           from B in which all the elements are equal to zero.
%           Each equation in B is build up from the matrix A. The matrix A is
%           a matrix which represents all the velocity points of
%           the grid. All elements of a new equation will be
%           represented in A. After forming the equation, the total
%           A-matrix will form a new row in B. So, the program contains
%           a for loop in which the A matrix will be reshaped as a row.
%% -CALL PARAMETERS AND CREATE MATRICES-

```

```

function [Wnew] =
WfieldCorrection(U,W,u,w,T,P,dx,dz,dt,mu,g,beta,T0,rho0,Q,Lengthi,LengthJ,alphaW)

A = zeros(Lengthi,LengthJ);
B = zeros((Lengthi-1)*LengthJ,Lengthi*LengthJ);
F = [];
C = [];
q = 1:(Lengthi*LengthJ);
k = 1;

%% -CENTRE (COMMON EQUATIONS)-

for j = 3:LengthJ-1
    J = j;
    for I = 2:Lengthi-1
        i = I;
        aIj = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-
1))+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);

        A(I,j) = aIj/alphaW;
        A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
        A(I+1,j) = -(Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
        A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
        A(I,j+1) = -(Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*(T(I,J)+T(I,J-
1))/2-T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -LEFT BOUNDARY-

for j = 3:LengthJ-1
    J = j;
    for I = 1
        i = I;
        aIj = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(0+0)+Q*dx*(W(I,j)+W(I,j+1))-
Q*dx*(W(I,j)+W(I,j-1))+3*mu*(dz/dx)+2*mu*(dx/dz);

        A(I,j) = aIj/alphaW;
        % --> A(I-1,j) = 0
        A(I+1,j) = -(Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
        A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
        A(I,j+1) = -(Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*(T(I,J)+T(I,J-
1))/2-T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -RIGHT BOUNDARY-

for j = 3:LengthJ-1
    J = j;
    for I = Lengthi
        i = I;
        aIj = rho0*((dx*dz)/dt)+Q*dz*(0+0)-Q*dz*(U(i,J)+U(i,J-1))+Q*dx*(W(I,j)+W(I,j+1))-
Q*dx*(W(I,j)+W(I,j-1))+3*mu*(dz/dx)+2*mu*(dx/dz);

        A(I,j) = aIj/alphaW;
        A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));

```



```

% --> A(I+1,j) = 0
A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

B(k,:) = [A(q)];

Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;
end
end

%% -UNDER BOUNDARY-

for j = 2
    J = j;
    for I = 2:Lengthi-1
        i = I;
        aIj = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-1))+Q*dx*(W(I,j)+W(I,j+1))-Q*dx*(W(I,j)+0)+2*mu*(dz/dx)+2*mu*(dx/dz);

        A(I,j) = aIj/alphaW;
        A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
        A(I+1,j) = -(-Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
        % --> A(I,j-1) = 0
        A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

        B(k,:) = [A(q)];

        Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -UPPER BOUNDARY-

for j = LengthJ
    J = j;
    for I = 2:Lengthi-1
        i = I;
        aIj = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(U(i,J)+U(i,J-1))+Q*dx*(W(I,j)+0)-Q*dx*(W(I,j)+W(I,j-1))+2*mu*(dz/dx)+2*mu*(dx/dz);

        A(I,j) = aIj/alphaW;
        A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
        A(I+1,j) = -(-Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
        A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
        % --> A(I,j+1) = 0

        B(k,:) = [A(q)];

        Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

        C = [C;Rightpart];

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -UPPER LEFT CORNER-

j = LengthJ;
J = j;

```

```

I = 1;
i = I;

aIj = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(0+0)+Q*dx*(W(I,j)+0)-
Q*dx*(W(I,j)+W(I,j-1))+3*mu*(dz/dx)+2*mu*(dx/dz);

A(I,j) = aIj/alphaW;
% --> A(I-1,j) = 0
A(I+1,j) = -(-Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
% --> A(I,j+1) = 0

B(k,:) = [A(q)];

Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-
T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UPPER RIGHT CORNER-

j = LengthJ;
J = j;
I = Lengthi;
i = I;

aIj = rho0*((dx*dz)/dt)+Q*dz*(0+0)-Q*dz*(U(i,J)+U(i,J-1))+Q*dx*(W(I,j)+0)-Q*dx*(W(I,j)+W(I,j-
1))+3*mu*(dz/dx)+2*mu*(dx/dz);

A(I,j) = aIj/alphaW;
A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
% --> A(I+1,j) = 0
A(I,j-1) = -(Q*dx*(W(I,j)+W(I,j-1))+mu*(dx/dz));
% --> A(I,j+1) = 0

B(k,:) = [A(q)];

Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-
T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UNDER LEFT CORNER-

j = 2;
J = j;
I = 1;
i = I;

aIj = rho0*((dx*dz)/dt)+Q*dz*(U(i+1,J)+U(i+1,J-1))-Q*dz*(0+0)+Q*dx*(W(I,j)+W(I,j+1))-
Q*dx*(W(I,j)+0)+3*mu*(dz/dx)+2*mu*(dx/dz);

A(I,j) = aIj/alphaW;
A(I+1,j) = -(-Q*dz*(U(i+1,J)+U(i+1,J-1))+mu*(dz/dx));
% --> A(I-1,j) = 0
A(I,j+1) = -(Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));
% --> A(I,j-1) = 0

B(k,:) = [A(q)];

Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-
T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

```

```

%% -UNDER RIGHT CORNER-

j = 2;
J = j;
I = Lengthi;
i = I;

aIj = rho0*((dx*dz)/dt)+Q*dz*(0+0)-Q*dz*(U(i,J)+U(i,J-1))+Q*dx*(W(I,j)+W(I,j+1))-
Q*dx*(W(I,j)+0)+3*mu*(dz/dx)+2*mu*(dx/dz);

A(I,j) = aIj/alphaW;
A(I-1,j) = -(Q*dz*(U(i,J)+U(i,J-1))+mu*(dz/dx));
% --> A(I+1,j) = 0
% --> A(I,j-1) = 0
A(I,j+1) = -(-Q*dx*(W(I,j)+W(I,j+1))+mu*(dx/dz));

B(k,:) = [A(q)];

Rightpart = dx*(P(I,J-1)-P(I,J))+rho0*((dx*dz)/dt)*w(I,j)+rho0*g*beta*((T(I,J)+T(I,J-1))/2-
T0)*dx*dz+(1-alphaW)*(aIj/alphaW)*W(I,j);

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

clear k

%% -CALCULATION OF Uster FIELD-

F = B(1:Lengthi*(LengthJ-1),Lengthi+1:(Lengthi*LengthJ));

Wnew = F\C;

k = 1;

for j = 1:LengthJ-1
    for I = 1:Lengthi
        WMatrix(I,j+1) = Wnew(k);
        k = k+1;
    end
end

Wnew = [WMatrix zeros(Lengthi,1)];

```

10.5.8 Deelprogramma 7: berekenen temperatuur veld

```

%% CALCULATION OF T FIELD
%           In this pogramme the values of T field will be
%           calculated. Each Temperature point in the grid exists of an
%           equation established from arounded Temperature points in the grid.
%           All equations together form the matrix B en vector C:
%           B*T = C.
%           So T = B\C.
%           Each equation in B is build up from the matrix A. The matrix A is
%           a matrix which represents all the pressure points of
%           the grid. All elements of a new equation will be
%           represented in A. After forming the equation, the total
%           A-matrix will form a new row in B. So, the program contains
%           a for loop in which the A matrix will be reshaped as a row.

%% -CALL PARAMETERS AND CREATE MATRICES-

function [T] = TemperatureField(U,W,T,Tw,dx,dz,dt,mu,rho0,Lengthi,LengthJ,kc,kcw,cp)

A = zeros(Lengthi,LengthJ);
B = zeros((Lengthi-1)*LengthJ,Lengthi*LengthJ);
F = [];
C = [];
q = 1:(Lengthi*LengthJ);
k = 1;

%% -CENTRE (COMMON EQUATIONS)-

for J = 2:LengthJ-1
    j = J;
    for I = 2:Lengthi-1

```

```

    i = I;
    Fe = rho0*cp*U(i+1,J)*dz; Fc = rho0*cp*U(i,J)*dz; Fn = rho0*cp*W(I,j+1)*dx; Fs =
rho0*cp*W(I,j)*dx;
    De = kc*(dz/dx); Dc = kc*(dz/dx); Dn = kc*(dx/dz); Ds = kc*(dx/dz);

    if Fc > 0
        aC = Dc + Fc;
    else
        aC = Dc;
    end
    if Fe > 0
        aE = De;
    else
        aE = De - Fe;
    end
    if Fs > 0
        aS = Ds + Fs;
    else
        aS = Ds;
    end
    if Fn > 0
        aN = Dn;
    else
        aN = Dn - Fn;
    end

    A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
    A(I-1,J) = -aC;
    A(I+1,J) = -aE;
    A(I,J-1) = -aS;
    A(I,J+1) = -aN;

    B(k,:) = [A(q)];

    Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J);

    C = [C;Rightpart];

    A = zeros(Lengthi,LengthJ);

    k = k+1;
end
end

%% -LEFT BOUNDARY-
for J = 2:LengthJ-1
    j = J;
    for I = 1
        i = I;
        Fe = rho0*cp*U(i+1,J)*dz; Fc = 0; Fn = rho0*cp*W(I,j+1)*dx; Fs = rho0*cp*W(I,j)*dx;
        De = kc*(dz/dx); Dc = kc*(dz/dx); Dn = kc*(dx/dz); Ds = kc*(dx/dz);

        if Fc > 0
            aC = 2*Dc + Fc;
        else
            aC = 2*Dc;
        end
        if Fe > 0
            aE = De;
        else
            aE = De - Fe;
        end
        if Fs > 0
            aS = Ds + Fs;
        else
            aS = Ds;
        end
        if Fn > 0
            aN = Dn;
        else
            aN = Dn - Fn;
        end

        A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
        % --> A(I-1,J) = 0
    end
end

```

```

A(I+1,J) = -aE;
A(I,J-1) = -aS;
A(I,J+1) = -aN;

B(k,:) = [A(q)];

Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J)+aC*Tw;

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;
end
end

%% -RIGHT BOUNDARY-

for J = 2:LengthJ-1
    j = J;
    for I = Lengthi
        i = I;
        Fe = 0; Fc = rho0*cp*U(i,J)*dz; Fn = rho0*cp*W(I,j+1)*dx; Fs = rho0*cp*W(I,j)*dx;
        De = kcw*(dz/dx); Dc = kc*(dz/dx); Dn = kc*(dx/dz); Ds = kc*(dx/dz);

        if Fc > 0
            aC = Dc + Fc;
        else
            aC = Dc;
        end
        if Fe > 0
            aE = 2*De;
        else
            aE = 2*De - Fe;
        end
        if Fs > 0
            aS = Ds + Fs;
        else
            aS = Ds;
        end
        if Fn > 0
            aN = Dn;
        else
            aN = Dn - Fn;
        end
        end

A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
A(I-1,J) = -aC;
% --> A(I+1,J) = 0
A(I,J-1) = -aS;
A(I,J+1) = -aN;

B(k,:) = [A(q)];

Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J)+aE*Tw;

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;
end
end

%% -UNDER BOUNDARY-

for J = 1
    j = J;
    for I = 2:Lengthi-1
        i = I;
        Fe = rho0*cp*U(i+1,J)*dz; Fc = rho0*cp*U(i,J)*dz; Fn = rho0*cp*W(I,j+1)*dx; Fs = 0;
        De = kc*(dz/dx); Dc = kc*(dz/dx); Dn = kc*(dx/dz); Ds = kcw*(dx/dz);

        if Fc > 0
            aC = Dc + Fc;
        else

```

```

        aC = Dc;
    end
    if Fe > 0
        aE = De;
    else
        aE = De - Fe;
    end
    if Fs > 0
        aS = 2*Ds + Fs;
    else
        aS = 2*Ds;
    end
    if Fn > 0
        aN = Dn;
    else
        aN = Dn - Fn;
    end

    A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
    A(I-1,J) = -aC;
    A(I+1,J) = -aE;
    % --> A(I,J-1) = 0
    A(I,J+1) = -aN;

    B(k,:) = [A(q)];

    Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J)+aS*Tw;

    C = [C;Rightpart];

    A = zeros(Lengthi,LengthJ);

    k = k+1;
end
end

%% -UPPER BOUNDARY-

for J = LengthJ
    j = J;
    for I = 2:Lengthi-1
        i = I;
        Fe = rho0*cp*U(i+1,J)*dz; Fc = rho0*cp*U(i,J)*dz; Fn = 0; Fs = rho0*cp*W(I,j)*dx;
        De = kc*(dz/dx); Dc = kc*(dz/dx); Dn = kc*w*(dx/dz); Ds = kc*(dx/dz);

        if Fc > 0
            aC = Dc + Fc;
        else
            aC = Dc;
        end
        if Fe > 0
            aE = De;
        else
            aE = De - Fe;
        end
        if Fs > 0
            aS = Ds + Fs;
        else
            aS = Ds;
        end
        if Fn > 0
            aN = 2*Dn;
        else
            aN = 2*Dn - Fn;
        end

        A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
        A(I-1,J) = -aC;
        A(I+1,J) = -aE;
        A(I,J-1) = -aS;
        % --> A(I,J+1) = 0

        B(k,:) = [A(q)];

        Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J)+aN*Tw;

        C = [C;Rightpart];
    end
end

```

```

        A = zeros(Lengthi,LengthJ);

        k = k+1;
    end
end

%% -UPPER LEFT CORNER-

J = LengthJ;
j = J;
I = 1;
i = I;

Fe = rho0*cp*U(i+1,J)*dz; Fc = 0; Fn = 0; Fs = rho0*cp*W(I,j)*dx;
De = kc*(dz/dx); Dc = kcw*(dz/dx); Dn = kcw*(dx/dz); Ds = kc*(dx/dz);

if Fc > 0
    aC = 2*Dc + Fc;
else
    aC = 2*Dc;
end
if Fe > 0
    aE = De;
else
    aE = De - Fe;
end
if Fs > 0
    aS = Ds + Fs;
else
    aS = Ds;
end
if Fn > 0
    aN = 2*Dn;
else
    aN = 2*Dn - Fn;
end

A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
% --> A(I-1,J) = 0
A(I+1,J) = -aE;
A(I,J-1) = -aS;
% --> A(I,J+1) = 0

B(k,:) = [A(q)];

Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J)+aC*Tw+aN*Tw;

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UPPER RIGHT CORNER-

J = LengthJ;
j = J;
I = Lengthi;
i = I;

Fe = 0; Fc = rho0*cp*U(i,J)*dz; Fn = 0; Fs = rho0*cp*W(I,j)*dx;
De = kcw*(dz/dx); Dc = kc*(dz/dx); Dn = kcw*(dx/dz); Ds = kc*(dx/dz);

if Fc > 0
    aC = Dc + Fc;
else
    aC = Dc;
end
if Fe > 0
    aE = 2*De;
else
    aE = 2*De - Fe;
end
if Fs > 0
    aS = Ds + Fs;
else

```

```

        aS = Ds;
end
if Fn > 0
    aN = 2*Dn;
else
    aN = 2*Dn - Fn;
end

A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
A(I-1,J) = -aC;
% --> A(I+1,J) = 0
A(I,J-1) = -aS;
% --> A(I,J+1) = 0

B(k,:) = [A(q)];

Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J)+aE*Tw+aN*Tw;

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UNDER LEFT CORNER-

J = 1;
j = J;
I = 1;
i = I;

Fe = rho0*cp*U(i+1,J)*dz; Fc = 0; Fn = rho0*cp*W(I,j+1)*dx; Fs = 0;
De = kc*(dz/dx); Dc = kcw*(dz/dx); Dn = kc*(dx/dz); Ds = kcw*(dx/dz);

if Fc > 0
    aC = 2*Dc + Fc;
else
    aC = 2*Dc;
end
if Fe > 0
    aE = De;
else
    aE = De - Fe;
end
if Fs > 0
    aS = 2*Ds + Fs;
else
    aS = 2*Ds;
end
if Fn > 0
    aN = Dn;
else
    aN = Dn - Fn;
end

A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
% --> A(I-1,J) = 0
A(I+1,J) = -aE;
% --> A(I,J-1) = 0
A(I,J+1) = -aN;

B(k,:) = [A(q)];

Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J)+aC*Tw+aS*Tw;

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

k = k+1;

%% -UNDER RIGHT CORNER-

J = 1;
j = J;
I = Lengthi;
i = I;

```



```

Fe = 0; Fc = rho0*cp*U(i,J)*dz; Fn = rho0*cp*W(I,j+1)*dx; Fs = 0;
De = kcw*(dz/dx); Dc = kc*(dz/dx); Dn = kc*(dx/dz); Ds = kcw*(dx/dz);

if Fc > 0
    aC = Dc + Fc;
else
    aC = Dc;
end
if Fe > 0
    aE = 2*De;
else
    aE = 2*De - Fe;
end
if Fs > 0
    aS = 2*Ds + Fs;
else
    aS = 2*Ds;
end
if Fn > 0
    aN = Dn;
else
    aN = Dn - Fn;
end

A(I,J) = rho0*cp*((dx*dz)/dt)+aC+aE+aS+aN+(Fe-Fc)+(Fn-Fs);
A(I-1,J) = -aC;
% --> A(I+1,J) = 0
% --> A(I,J-1) = 0
A(I,J+1) = -aN;

B(k,:) = [A(q)];

Rightpart = rho0*cp*((dx*dz)/dt)*T(I,J)+aE*Tw+aS*Tw;

C = [C;Rightpart];

A = zeros(Lengthi,LengthJ);

clear k

%% -CALCULATION OF Uster FIELD-

Tfield = B\C;

k = 1;

for J = 1:LengthJ
    for I = 1:Lengthi
        TMatrix(I,J) = Tfield(k);
        k = k+1;
    end
end

T = TMatrix;

```

10.5.9 Deelprogramma 8: plotten van de velden

```

%% PLOT TEMPERATURE AND VELOCITY FIELDS
%           In this programme the Temperature and velocity fields are
%           plot. To plot the temperature field, the function 'imagesc' is used.
%           To plot the velocity fields, the U and W matrices must be
%           reshaped in such a way they are useful for the function
%           'streamslice'.

%% -CALL PARAMETERS AND CREATE MATRICES-

function [] = Plotfunction(t,U,W,T,Lengthi,LengthJ)

PlotU = zeros(Lengthi+1,LengthJ+1);
PlotW = zeros(Lengthi+1,LengthJ+1);

axis manual

%% -RESHAPE MATRICES U AND W TO PlotU AND PlotW-

```

```

for J = 2:LengthJ
    j = J;
    for i = 2:Lengthi
        I = i;
        PlotU(I,J) = (U(i,J-1)+U(i,J))/2;
    end
end

for J = 2:LengthJ
    j = J;
    for i = 2:Lengthi
        I = i;
        PlotW(I,J) = (W(I-1,j)+W(I,j))/2;
    end
end

%% -PLOT TEMPERATURE AND VELOCITIES-

if t > 0
    clf
end

% -PLOT TEMPERATURE-

subplot(2,1,1)
CLIM = [20 20.6];
imagesc(flipud(T'),CLIM)
set(gca,'XTickLabel',0.5:0.5:5)
set(gca,'YTickLabel',4.5:-1:0.5)
colorbar
xlabel('x-direction [m]')
ylabel('z-direction [m]')

% -PLOT VELOCITIES-

subplot(2,1,2)
streamslice(PlotU',PlotW')
axis([1 Lengthi+1 1 LengthJ+1])
set(gca,'XTickLabel',0:0.25:5)
set(gca,'YTickLabel',0:1:5)
xlabel('x-direction [m]')
ylabel('z-direction [m]')

hold off
drawnow

```