

Moving-Grid-Verfahren zur rechnerunterstützten Vorverarbeitung von partiellen Differentialgleichungen

Jochen Rieber

(kyb)

Studienarbeit Simulationstechnik

Prof. Dr.-Ing. M. Zeitz

Betreuer: Dipl.-Ing. R. Köhler

Institut für Systemdynamik und Regelungstechnik

Prof. Dr.-Ing. Dr. h.c. mult. E.D. Gilles

Universität Stuttgart

2000

Kurzfassung

Ein Standardverfahren zur numerischen Lösung von partiellen Differentialgleichungen ist die Method-of-Lines. Die dafür notwendige Ortsdiskretisierung erfolgt oftmals mit einem festen äquidistanten Gitter. Um wandernde Fronten oder, allgemeiner gesagt, große, zeitlich veränderliche Ortsgradienten genügend genau auflösen zu können, ist die Verwendung sehr feiner Gitter notwendig, was eine ineffiziente Berechnung zur Folge hat. In dieser Arbeit wird ein Verfahren mit adaptiver Gittersteuerung hinsichtlich Anwendbarkeit, Effizienz und Robustheit untersucht und in das symbolische Vorverarbeitungswerkzeug SYPPROT der Simulationsumgebung DIVA implementiert. Das auf der Gleichverteilung eines Bogenlängen-Kriteriums basierende Verfahren ist in der vorliegenden Implementierung für verkoppelte quasi-lineare partielle Differentialgleichungen maximal zweiter Ordnung einsetzbar. Dabei ist eine Kombination sowohl mit Finite-Differenzen- als auch mit Finite-Volumen-Methoden möglich. Um die Leistungsfähigkeit des Verfahrens zu überprüfen, wird es auf einfache Benchmark-Modelle wie auch auf realistische Modelle aus der Verfahrenstechnik angewandt.

Abstract

An established procedure for the numerical solution of partial differential equations is the Method-of-Lines. Therefore required spatial discretization is often performed by a fixed uniform grid. In order to resolve travelling wave fronts or, more generally speaking, large variable spatial gradients, it is necessary to use very fine grids, which leads to inefficient computation. In this work a method with adaptive grid movement is examined with respect to applicability, efficiency, and robustness, and it is implemented into the symbolic preprocessing tool SYPPROT of the simulation environment DIVA. The implemented method is based on equidistribution of an arclength monitor and can be used for coupled quasi-linear partial differential equations with maximum order of two. Its combination is possible with both finite-difference and finite-volume methods. In order to test the capacity of the method, it is applied to benchmark models as well as real models from chemical engineering.

Inhaltsverzeichnis

1	Einführung	5
2	Simulationsumgebung DIVA	8
3	Diskretisierung mit der Method-of-Lines	11
3.1	Finite-Differenzen-Methode	13
3.2	Finite-Volumen-Methode	15
3.3	Adaptive Verfahren	18
3.3.1	Gleichverteilung	20
3.3.2	Fehlerschätzung	21
3.3.3	Weitere Verfahren zur Stützstellensteuerung	23
4	Darstellung des Moving-Grid-Verfahrens	26
4.1	Gleichungs-Transformation	27
4.2	Moving-Grid-Gleichungen	28
4.2.1	Gleichverteilung	28
4.2.2	Die Monitorfunktion	29
4.2.3	Gitterglättung	29
4.3	Fehler und Stabilität der numerischen Lösung	33
4.4	Zusammenfassung	35
5	Implementierung	37
5.1	Die MATHEMATICA-Daten-Struktur	37
5.1.1	Erweiterung der MATHEMATICA-Daten-Struktur	40
5.2	Erweiterung der Implementierung	43
5.2.1	Das Modul <code>MovGrid.m</code>	44
5.2.2	Anpassung des Finite-Differenzen-Verfahrens	49
5.2.3	Anpassung des Finite-Volumen-Verfahrens	50
5.2.4	Weitere Änderungen	51
6	Simulationsstudien	52
6.1	Transportmodell	53
6.2	Dwyer-Sanders-Flammenmodell	56
6.3	Gegenstrom-Adsorber	62
6.4	Zirkulations-Festbett-Reaktor	66

6.5	Einstellregeln und Benutzungshinweise	71
7	Zusammenfassung	74
7.1	Ausblick	75
A	MDS-Code Flammenmodell	76
B	Funktionen und globale Variablen von MovGrid.m	79
	Literaturverzeichnis	82

Häufig verwendete Formelzeichen

Symbol	Bedeutung
D	Diffusionskonstante eines Diffusionsterms
L	Länge des Ortsbereichs
P	Anzahl Gitterpunkte
N	Dimension eines Gleichungssystems
u	Eingangsgröße
t	unabhängige Zeitvariable
v	Geschwindigkeitskonstante eines Konvektionsterms
x	Zustandsgröße
z	unabhängige Ortskoordinate
κ	Moving-Grid-Parameter der örtlichen Gitterglättung
τ	Moving-Grid-Parameter der zeitlichen Gitterglättung

Kursiv gedruckte Formelzeichen entsprechen Skalaren,
fett kursiv gedruckte Formelzeichen stehen für Vektoren bzw. Matrizen.

Allgemeine Abkürzungen

Abkürzung	Bedeutung
AB	Anfangsbedingungen
CG	CODE GENERATOR
DA	Differential-Algebra-
Dgl.	Differentialgleichung
FD	Finite-Differenzen-Methode
FV	Finite-Volumen-Methode
MDS	MATHEMATICA-Daten-Struktur
MOL	Method-of-Lines
PDgl.	Partielle Differentialgleichung
RB	Randbedingungen

Kapitel 1

Einführung

Die mathematische Modellierung verfahrenstechnischer Systeme führt meist auf gewöhnliche bzw. partielle Differentialgleichungen mit algebraischen Gleichungen. Außer in einigen einfachen Fällen besitzen diese Modellgleichungen nichtlinearen Charakter und sind einer analytischen Lösung sehr oft nicht zugänglich. Deshalb muss auf eine numerische Lösung mittels Rechnerunterstützung zurückgegriffen werden.

Eine partielle Differentialgleichung (PDgl.) für ein örtlich verteiltes System wird üblicherweise mit Hilfe der *Method-of-Lines* (MOL) [20] auf gewöhnliche Zeit-Differentialgleichungen (Dgl.) zurückgeführt. Dies wird erreicht, indem durch eine geeignete Diskretisierung der Ortskoordinate eine endliche Anzahl von Stützpunkten anstatt eines kontinuierlichen Ortsbereiches eingeführt wird. Mittels der Zeit-Dgl. können Approximationen der Zustandsgrößen und damit eine numerische Lösung der PDgl. berechnet werden.

Schwierigkeiten können auftreten, wenn sich im Ortsprofil der Zustandsgrößen steile wandernde Fronten ausbilden. Das heißt, es treten große, zeitlich veränderliche Ortsgradienten auf, wie dies zum Beispiel bei wandernden Temperaturfronten in einem Reaktor der Fall ist (Abbildung 1.1). Benutzt man dabei im einfachsten Fall ein festes äquidistantes Ortsgitter, muss für hinreichende Lösungsgenauigkeit mit einer sehr hohen Stützstellenanzahl gearbeitet werden. Dadurch hält man zwar den Diskretisierungsfehler klein, verschwendet aber auch zu viele Stützstellen auf Bereiche mit kleinen Gradienten. Daraus resultieren hohe Rechenzeiten und Ineffizienz.

Ein Ansatz zur Lösung dieses Problems ist die Verwendung eines adaptiven Gitters. In den sogenannten *Moving-Grid-* bzw. *Moving-Mesh-Verfahren* werden die Ortsstützstellen weder zeitlich noch örtlich fest gewählt, sondern mittels einer Stützstellensteuerung dem Lösungsverlauf angepaßt. Die Gitterpunkte werden so verschoben oder hinzugefügt, dass in Lösungsbereichen mit großen Ortsgradienten eine Verbesserung der Lösungsgenauigkeit erzielt wird. Deshalb kann mit einer vergleichsweise geringen Anzahl von Stützstellen gearbeitet werden, was die Effizienz der Berechnung der numerischen Lösung steigert.

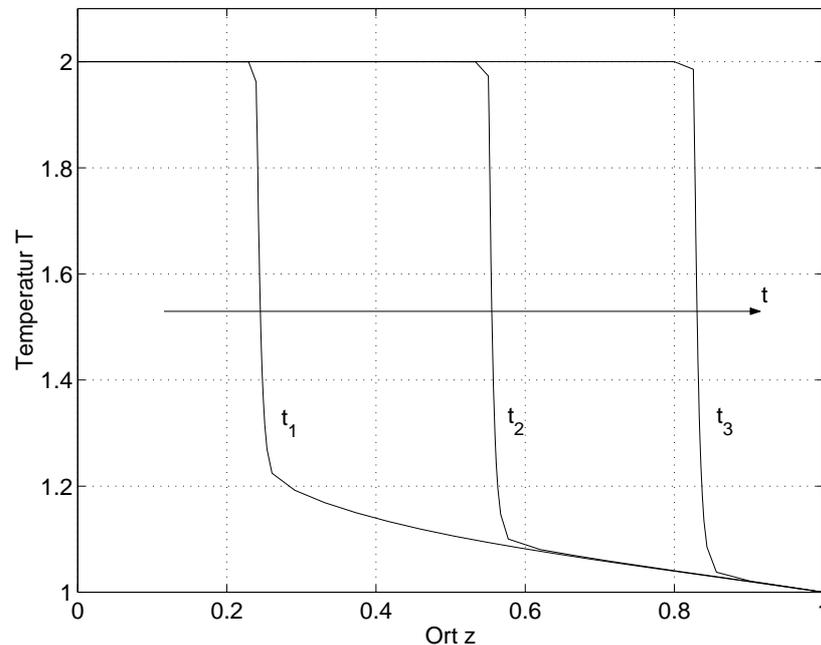


Abb. 1.1: Wandernde Temperaturfront (normiert) zu drei verschiedenen Zeitpunkten t_i , $i = 1, 2, 3$.

Aufgabenstellung

Für die am Institut für Systemdynamik und Regelungstechnik entstandene Simulationsumgebung DIVA [8, 18] wird das symbolische Vorverarbeitungswerkzeug SYPPROT zur automatischen Diskretisierung örtlich verteilter Modelle entwickelt. Die Ortsdiskretisierung basiert auf der MOL und stellt Finite-Differenzen- und Finite-Volumen-Schemata zur Verfügung. Dabei können beliebige unbewegte Ortsgitter verwendet werden.

Ein Moving-Grid-Verfahren, welches die Gleichverteilung eines Bogenlängen-Kriteriums benutzt [24], wurde ausgewählt, um in das bestehende Vorverarbeitungswerkzeug integriert zu werden. Das Verfahren soll mathematisch dargestellt und hinsichtlich einer Implementierung in das Vorverarbeitungswerkzeug untersucht und aufgearbeitet werden. Danach ist eine geeignete Erweiterung der vorhandenen Datenstruktur, welche verteilte Modelle und Diskretisierungsparameter repräsentiert, für die Anwendung bewegter Gitter vorgesehen. Darüber hinaus sollen das Moving-Grid-Verfahren in SYPPROT implementiert und die vorhandenen Diskretisierungsmethoden entsprechend angepasst werden. Eine Reihe von Simulationsexperimenten mit dem Moving-Grid-Verfahren und geeigneten Testmodellen soll schließlich Aufschluss über Anwendbarkeit und Grenzen des Verfahrens sowie über Einstellregeln der Parameter geben.

Gliederung der Arbeit

Die vorliegende Studienarbeit beschreibt in Kapitel 2 die Simulationsumgebung `DiVA` und das symbolische Vorverarbeitungswerkzeug `SYPPROT`. Nachdem in Kapitel 3 die MOL mit Hilfe von Finite-Differenzen- (FD) und Finite-Volumen- (FV) sowie mit adaptiven Methoden dargestellt wird, erfolgt eine detaillierte Erläuterung des ausgewählten Moving-Grid-Verfahrens in Kapitel 4 im Detail. Daran schließt sich in Kapitel 5 eine Beschreibung der Implementierung in das Vorverarbeitungswerkzeug sowie der neuen Funktionalität von `SYPPROT` an. In Kapitel 6 werden vier Testbeispiele aus der Verfahrenstechnik untersucht. Ein Transportmodell, ein Reaktions-Diffusions-Modell sowie die Modelle eines Gegenstrom-Adsorbers und eines Zirkulations-Festbett-Reaktors geben Aufschluss über die Leistungsfähigkeit des Moving-Grid-Verfahrens und Hinweise für seine Benutzung. In Kapitel 7 werden schließlich eine Zusammenfassung der Ergebnisse und ein Ausblick auf Erweiterungen des Verfahrens gegeben.

Kapitel 2

Simulationsumgebung DIVA

Die Simulationsumgebung DIVA (**D**ynamischer **S**imulator für verfahrenstechnische **A**nlagen) [8, 18] ist ein Software-Paket, das die numerische Simulation, Analyse und Optimierung von Differential-Algebra-Gleichungen ermöglicht. DIVA wurde vorwiegend für verfahrenstechnische Prozessmodelle konzipiert.

Die Architektur von DIVA (Abbildung 2.1) läßt sich in vier Ebenen einteilen. Die erste Ebene besteht aus den numerischen Methoden und einer Modell-Bibliothek. Damit ist es möglich, über vordefinierte Prozessmodelle komplexe Anlagenmodelle mittels einer Fließbildverschaltung zu erstellen. Diese Prozessmodelle oder Grundelemente werden als FORTRAN-Dateien in einer bestimmten Datenstruktur repräsentiert und zusammen mit den Numerik-Routinen von DIVA in ein ausführbares FORTRAN-Programm überführt. Um die Eingabe der Modelle zu vereinfachen, wurde der CODE GENERATOR (CG) [19, 10] entwickelt, welcher die zweite Ebene darstellt. Aus einer sogenannten CG-Eingabedatei erzeugt der CG die benötigten FORTRAN-Dateien. In der CG-Eingabedatei werden das mathematische Modell und die zugehörigen Parameter in einer an die Programmiersprache LISP angelehnten Syntax definiert.

Sowohl die DIVA-Numerik als auch der CODE GENERATOR benutzen zur Modellbeschreibung die linear-implizite Differential-Algebra-Form (DA-Form)

$$\begin{aligned} \mathbf{B}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) \frac{d\mathbf{x}}{dt} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) & t > t_0, & \quad \mathbf{x}(t_0) = \mathbf{x}_0, & \quad (2.1) \\ \mathbf{y} &= \mathbf{H}\mathbf{x} \end{aligned}$$

mit einem differentiellen Index von kleiner oder gleich eins. Dabei bedeuten die Bezeichner $\mathbf{x} \in \mathbb{R}^N$ Zustandsvektor, \mathbf{B} linke-Seite-Matrix (normalerweise singular), \mathbf{f} Funktionsvektor, \mathbf{u} Eingangsvektor, \mathbf{p} Parametervektor, t Zeitvariable, \mathbf{x}_0 Vektor der Anfangsbedingungen, \mathbf{y} Ausgangsvektor.

Die CG-Eingabe-Datei wird vom Benutzer per Hand eingegeben oder durch das Vorverarbeitungswerkzeug SYPPROT [12, 19, 2] bzw. das Modellierungswerkzeug PROMOT [23] erzeugt. Diese beiden Werkzeuge stellen die dritte bzw. vierte Ebene dar.

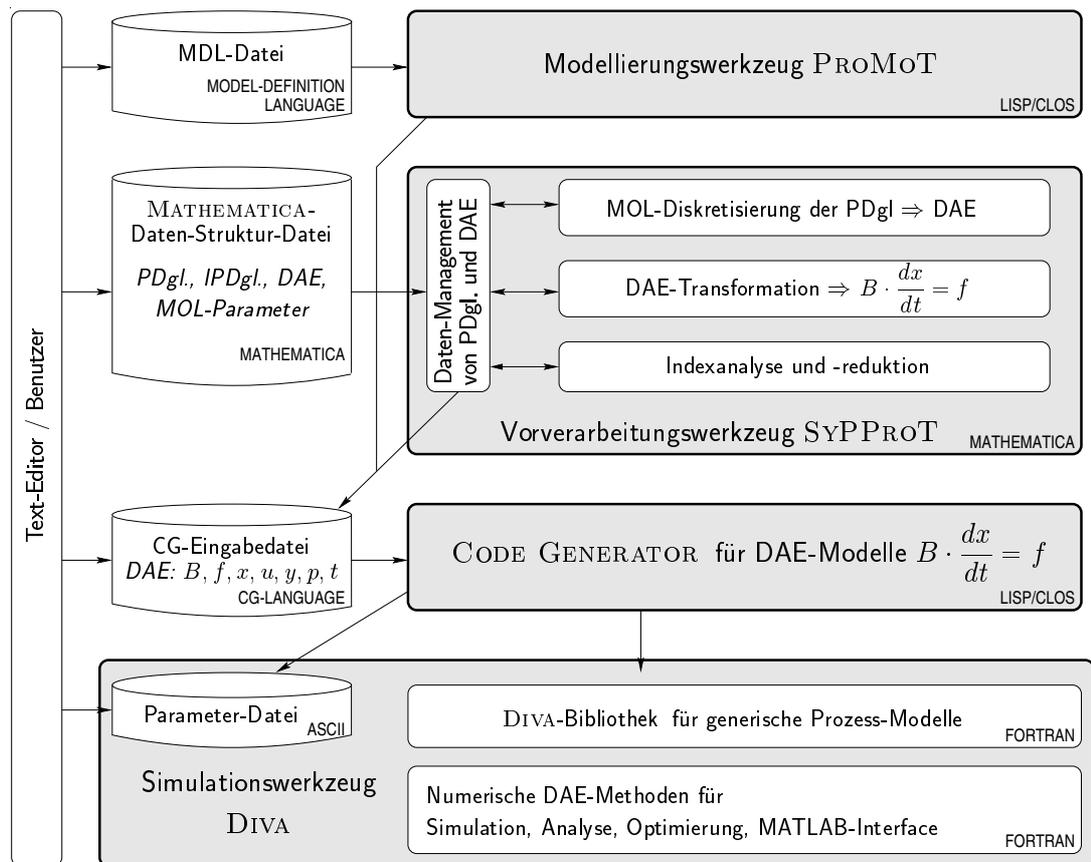


Abb. 2.1: Architektur der Simulationsumgebung DIVA [18].

Das Modellierungswerkzeug PROMOT ist ein wissensbasiertes Rechnerwerkzeug zur gleichungsbasierten, objektorientierten Modellierung von verfahrenstechnischen Apparaten auf der Phasenebene. Die mit PROMOT erstellten Apparatemodelle werden in Form von CG-Eingabedateien ausgegeben.

Die Aufgabe des Vorverarbeitungswerkzeugs SYPPROT (**S**ymbolic **P**re**P**rocessing **T**ool) ist die Transformation von Prozessmodellen, hergeleitet aus Massen-, Energie- und Impulsbilanzen, in die linear-implizite DA-Form. Die dafür bereitgestellten Funktionalitäten umfassen die MOL-Diskretisierung von verteilten Modellen, die tatsächliche Transformation in die DA-Form sowie eine Indexanalyse und -reduktion der DAGleichungen. Zur MOL-Diskretisierung stellt das Programmpaket die Methoden der Finite-Differenzen- und der Finite-Volumen-Approximation zur Verfügung, sowohl für äquidistante als auch für nicht äquidistante Gitter.

SYPPROT ermöglicht die Eingabe und Weiterverarbeitung von gemischten Gleichungssystemen von PDgln., Integro-PDgln. und DA-Gleichungen. Die Klasse der quasilinearen PDgl.-Systeme maximal 2. Ordnung nach folgender Form wird dabei

unterstützt:

$$\mathbf{A}(\mathbf{x}, \mathbf{u}, \mathbf{p}, z, t) \frac{\partial \mathbf{x}}{\partial t} = \mathbf{C}(\mathbf{x}, \mathbf{u}, \mathbf{p}, z, t) \frac{\partial^2 \mathbf{x}}{\partial z^2} + \frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u}, \mathbf{p}, z, t)}{\partial z} + \mathbf{S}(\mathbf{x}, \mathbf{u}, \mathbf{p}, z, t) \quad (2.2)$$

$$t > t_0, \quad 0 < z < L$$

$$\mathbf{C}_{0,L}(\mathbf{x}, \mathbf{u}, \mathbf{p}, z, t) \frac{\partial \mathbf{x}}{\partial z} + \mathbf{F}_{0,L}(\mathbf{x}, \mathbf{v}_{0,L}, \mathbf{p}, z, t) = 0 \quad t > t_0, \quad z \in \{0, L\}. \quad (2.3)$$

$$\mathbf{x}(z, t_0) = \mathbf{x}_0(z) \quad 0 \leq z \leq L \quad (2.4)$$

Die Gleichungen beschreiben das Verhalten von N Zuständen $\mathbf{x} \in \mathbb{R}^N$. Dabei bedeuten die Bezeichner \mathbf{A} , \mathbf{C} , $\mathbf{C}_{0,L}$ ($N \times N$)-Koeffizientenmatrizen, \mathbf{F} , $\mathbf{F}_{0,L}$ Funktionsvektoren, \mathbf{S} vektorieller Quellterm des Modells, \mathbf{u} , $\mathbf{v}_{0,L}$ Eingangsvektoren, \mathbf{p} Parametervektor, z Ortskoordinate, t Zeitvariable und \mathbf{x}_0 Vektor der Anfangsbedingungen. Diese Modellformulierung schließt sowohl hyperbolische als auch parabolische PDgln. mit Nichtlinearitäten in den Termen \mathbf{A} , \mathbf{C} , \mathbf{F} , \mathbf{S} , $\mathbf{C}_{0,L}$ und $\mathbf{F}_{0,L}$, verbunden mit verschiedenen Arten von Randbedingungen, ein. Außerdem können durch Formulierung des Quellterms \mathbf{S} mit Integralen auch Integro-PDgln. angegeben werden, wie sie typischerweise für Populationsbilanzen auftreten.

Als Eingabesprache wird dazu die sogenannte MATHEMATICA-Daten-Struktur (MDS) verwendet, siehe Abbildung 2.1. Das Vorverarbeitungswerkzeug SYPPROT ist mit Hilfe des Compute-Algebra-Systems MATHEMATICA implementiert.

Zusammengefasst läuft die Simulation eines verteilten Modelles in der Simulationsumgebung DIVA wie folgt ab:

1. Erstellung des Modells als MDS-Datei;
2. MOL-Diskretisierung des verteilten Modells und Erzeugung der CG-Eingabedatei mit SYPPROT;
3. Erzeugung der Prozessmodelle mit dem CODE GENERATOR;
4. Erstellung des DIVA-FORTRAN-Programms zur Simulation.

Kapitel 3

Diskretisierung mit der Method-of-Lines

Der Ansatz der Method-of-Lines (MOL) zur numerischen Lösung von PDgln. basiert auf einer Approximation der Gleichungen auf einem Ortsgitter [20]. Dazu wird die ursprünglich kontinuierliche Ortskoordinate diskretisiert, also durch eine endliche Anzahl von unterscheidbaren Stützstellen ersetzt. Grundsätzlich kann dies sowohl durch ein äquidistantes als auch durch ein nicht-äquidistantes Gitter erfolgen. Damit wird die PDgl. in ein System von gewöhnlichen Differentialgleichungen mit der Zeit als unabhängiger Variablen umgeformt. Zusammen mit den Randbedingungen erhält man ein DA-Gleichungssystem aus Zeit-Dgln. und gegebenenfalls algebraischen Gleichungen (Abbildung 3.1). Dieses Gleichungssystem kann mit numerischen Verfahren zur Zeitintegration, möglichst mit Integratoren für steife Dgln., gelöst werden. Da in den heute zur Verfügung stehenden Integratoren eine Zeitschrittweiten-Steuerung eingebaut ist, kann der Fehler der Zeitintegration klein gehalten werden. Als Hauptfehlerquelle muss also der Fehler bei der Ortsdiskretisierung angesehen werden.

Die MOL wird im Folgenden auf ein verkoppeltes Gleichungssystem von N PDgln. angewandt. Die hier verwendete kompakte Form schließt dabei auch die von SYP-PROT unterstützten Gleichungen (2.2) bis (2.4) mit ein.

$$\mathbf{A}(\mathbf{x}, \mathbf{u}, \mathbf{p}, z, t) \frac{\partial \mathbf{x}}{\partial t} = \mathbf{f} \left(\mathbf{x}, \frac{\partial \mathbf{x}}{\partial z}, \frac{\partial^2 \mathbf{x}}{\partial z^2}, \mathbf{u}, \mathbf{p}, z, t \right) \quad t > t_0, \quad 0 < z < L \quad (3.1)$$

mit den Randbedingungen (RB)

$$\mathbf{g}_0 \left(\mathbf{x}, \frac{\partial \mathbf{x}}{\partial z}, \mathbf{v}_0, z = 0, t \right) = 0, \quad \mathbf{g}_L \left(\mathbf{x}, \frac{\partial \mathbf{x}}{\partial z}, \mathbf{v}_L, z = L, t \right) = 0 \quad t > t_0 \quad (3.2)$$

und den Anfangsbedingungen (AB)

$$\mathbf{x}(z, t_0) = \mathbf{x}_0(z) \quad 0 \leq z \leq L. \quad (3.3)$$

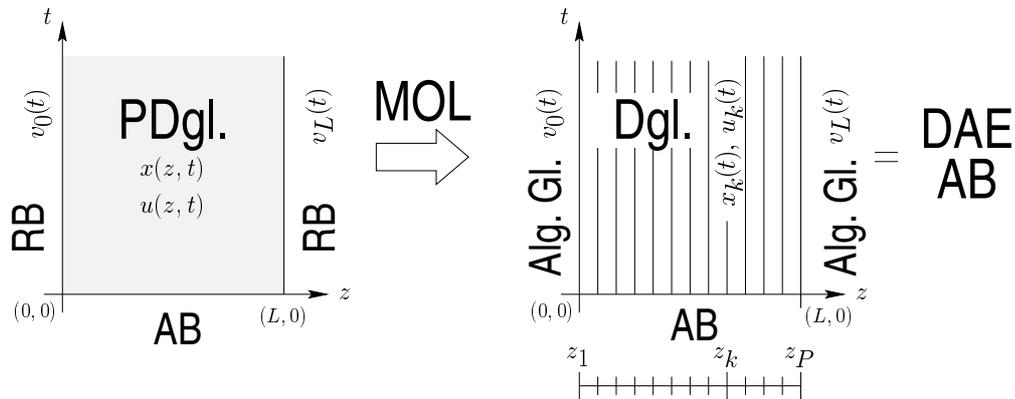


Abb. 3.1: Diskretisierung mit der Method-of-Lines (MOL). Umwandlung von partiellen Differentialgleichungen (PDgln.), Randbedingungen (RB) und Anfangsbedingungen (AB) in Differential-Algebra-Gleichungen (DAE) mit Anfangsbedingungen [20].

Es liegen N Zustände $\mathbf{x}(z, t) = [x_1(z, t), \dots, x_N(z, t)]^T$ vor. Die Matrix \mathbf{A} stellt die $(N \times N)$ -Koeffizientenmatrix der Zeitableitungen dar. Die Operatoren \mathbf{f} bzw. \mathbf{g}_0 und \mathbf{g}_L sind Differentialoperatoren bezüglich des Ortes und enthalten Ortsableitungen höchstens vom Grad 2 bzw. vom Grad 1. Als Systemeingänge dienen die Vektoren \mathbf{u} und $\mathbf{v}_{0,L}$.

Die Ortsdiskretisierung erfolgt üblicherweise auf einem zeitlich und örtlich konstanten, äquidistanten Gitter, womit die kontinuierliche Ortsvariable z durch diskrete Stützpunkte z_k , $k = 1(1)P$, ersetzt wird. Aus den zeit- und ortsabhängigen Zuständen $\mathbf{x}(z, t)$ werden Zeittrajektorien $\mathbf{x}_k(t) \equiv \mathbf{x}(z_k, t)$ bezüglich des Stützpunktes z_k . Die partiellen Ortsableitungen $\partial^i \mathbf{x} / \partial z^i$, $i = 1, 2, \dots$, werden durch Zustandsgrößen an mehreren Stützpunkten approximiert (siehe Kapitel 3.1 und 3.2). Einige oft verwendete Begriffe betreffend der Approximation der partiellen Ortsableitungen sollen hier noch erklärt werden. Bei einer *Zentraldifferenz* wird eine symmetrische Punktverteilung um das Approximationszentrum z_k zur Diskretisierung benutzt, also beispielsweise die Punkte z_{k-1} , z_k und z_{k+1} . Die *Rückwärtsdifferenz* benutzt zusätzlich zu z_k nur Punkte mit kleinerem Index, also z.B. die Punkte z_{k-1} und z_k . Demgegenüber benutzt eine *Vorwärtsdifferenz* zusätzlich zu z_k nur Punkte mit größerem Index, also z.B. die Punkte z_{k+1} und z_k . In Abhängigkeit von der Strömungsrichtung in einer Anlage werden noch die Begriffe *Upwind*-Diskretisierung und *Downwind*-Diskretisierung verwendet. Die erstgenannte Diskretisierung verwendet nur Punkte entgegen der Wanderungsrichtung der Front, entspricht also bei Frontwanderung in positive z -Richtung der Rückwärtsdifferenz. Downwind benutzt zur Approximation dagegen nur Punkte in Richtung der Strömung.

Das Ergebnis der MOL-Diskretisierung sind gewöhnliche Zeit-Dgl. für die Stützstellen z_k , $k = 2(1)P - 1$, resultierend aus den PDgl. Dazu kommen für die Randpunkte $k = 1$ und $k = P$ algebraische Gleichungen bei Anwendung Finiter Differenzen (Kapitel 3.1) oder Dgl. bei Anwendung Finiter Volumen (Kapitel 3.2), resultierend aus

den Randbedingungen. Die diskretisierten Anfangswerte $\mathbf{x}_k(t_0)$, $k = 1(1)P$, sind mit den Randbedingungen konsistent, falls Anfangsbedingungen und Randbedingungen in den Gleichungen (3.2) und (3.3) konsistent formuliert wurden.

Die folgenden Unterkapitel beschreiben Vorgehensweisen zur Ausführung der MOL und zur Approximation von Ortsableitungen. Die Finite-Differenzen-Methode (FD) sowie die Finite-Volumen-Methode (FV) sind zwei einfache und häufig angewandte Verfahren zur Umwandlung von PDgln. mit RB in DA-Gleichungen. Zusätzlich werden adaptive Verfahren vorgestellt und die wichtigsten Grundlagen zu Gittersteuerung und Fehlerschätzung bei bewegten Gittern beschrieben.

3.1 Finite-Differenzen-Methode

Die Approximation mittels Finiten-Differenzen-Methode ergibt sich ursprünglich aus einer Taylorreihen-Entwicklung der diskretisierten Zustandsvariablen $x_k(t)$ an verschiedenen Stützstellen.

Als Beispiel sollen mögliche Approximationen der ersten und zweiten Ortsableitungen aufgezeigt werden. Die Taylorentwicklungen für die Gitterpunkte mit Index $k+1$ bzw. $k-1$ um den Punkt mit Index k liefern mit der Abkürzung $\Delta z_k := z_{k+1} - z_k$ (Abhängigkeiten von t werden im Folgenden der Übersichtlichkeit halber weggelassen)

$$x_{k+1} = x_k + \left. \frac{\partial x}{\partial z} \right|_{z_k} \Delta z_k + \frac{1}{2} \left. \frac{\partial^2 x}{\partial z^2} \right|_{z_k} \Delta z_k^2 + \frac{1}{6} \left. \frac{\partial^3 x}{\partial z^3} \right|_{z_k} \Delta z_k^3 + \mathcal{O}(\Delta z_k^4) \quad (3.4)$$

$$x_{k-1} = x_k - \left. \frac{\partial x}{\partial z} \right|_{z_k} \Delta z_{k-1} + \frac{1}{2} \left. \frac{\partial^2 x}{\partial z^2} \right|_{z_k} \Delta z_{k-1}^2 - \frac{1}{6} \left. \frac{\partial^3 x}{\partial z^3} \right|_{z_k} \Delta z_{k-1}^3 + \mathcal{O}(\Delta z_{k-1}^4) . \quad (3.5)$$

Aus der Subtraktion der beiden Gleichungen folgt die Näherung der ersten Ortsableitung am Ort z_k als Zentralfdifferenz:

$$\left. \frac{\partial x}{\partial z} \right|_{z_k} = \frac{x_{k+1} - x_{k-1}}{z_{k+1} - z_{k-1}} + \mathcal{O}(\Delta z) \approx \frac{x_{k+1} - x_{k-1}}{z_{k+1} - z_{k-1}} . \quad (3.6)$$

Eine Rückwärtsdifferenz ergibt sich beispielsweise aus (3.5) zu

$$\left. \frac{\partial x}{\partial z} \right|_{z_k} = \frac{x_k - x_{k-1}}{z_k - z_{k-1}} + \mathcal{O}(\Delta z) \approx \frac{x_k - x_{k-1}}{z_k - z_{k-1}} . \quad (3.7)$$

Nach Addition von $\Delta z_{k-1} \cdot (3.4)$ und $\Delta z_k \cdot (3.5)$ erhält man die Näherung der zweiten Ortsableitung am Ort z_k ebenfalls als Zentralfdifferenz:

$$\left. \frac{\partial^2 x}{\partial z^2} \right|_{z_k} = \frac{2}{\Delta z_k + \Delta z_{k-1}} \left(\frac{x_{k+1} - x_k}{\Delta z_k} - \frac{x_k - x_{k-1}}{\Delta z_{k-1}} \right) + \mathcal{O}(\Delta z) . \quad (3.8)$$

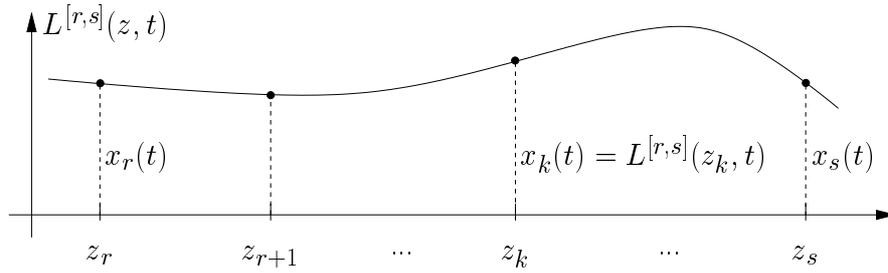


Abb. 3.2: Approximation eines Ortsprofils mit Lagrange-Polynomen an den Stützpunkten z_r bis z_s .

Da für äquidistante Gitter der Abstand zweier Stützpunkte Δz für alle k gleich groß ist, erhält man in diesem Fall für (3.7) und (3.8)

$$\left. \frac{\partial x}{\partial z} \right|_{z_k} = \frac{x_k - x_{k-1}}{\Delta z} + \mathcal{O}(\Delta z), \quad \left. \frac{\partial^2 x}{\partial z^2} \right|_{z_k} = \frac{x_{k+1} - 2x_k + x_{k-1}}{\Delta z^2} + \mathcal{O}(\Delta z^2). \quad (3.9)$$

Die Fehlerordnung der Näherung der zweiten Ortsableitung erhöht sich dabei um eins.

Das Vorgehen soll anhand einer skalaren Reaktions-Konvektions-Diffusions-PDgl. veranschaulicht werden. Die Anfangs- und Randbedingungen werden dabei nicht betrachtet.

$$\frac{\partial x}{\partial t} = -v \frac{\partial x}{\partial z} + D \frac{\partial^2 x}{\partial z^2} + s(x) \quad t > 0, \quad 0 < z < L.$$

Mit Rückwärtsdifferenz für die erste und Zentraldifferenz für die zweite Ortsableitung wird im äquidistanten Fall entsprechend (3.9) aus der PDgl. nach der Diskretisierung

$$\frac{dx_k}{dt} = -v \frac{x_k - x_{k-1}}{\Delta z} + D \frac{x_{k+1} - 2x_k + x_{k-1}}{\Delta z^2} + s(x_k) \quad k = 2(1)P - 1. \quad (3.10)$$

Approximation mit Lagrange-Polynomen

Zur Generierung der Finite-Differenzen-Approximationen werden im Vorverarbeitungswerkzeug SYPPROT Lagrange-Polynome verwendet. Die Approximation der Zustandsvariablen

$$x(z, t) \approx L^{[r,s]}(z, t) = \sum_{j=r}^s l_j(z) x_j(t) \quad (3.11)$$

am Approximationszentrum z_k wird durch die diskretisierten Zustände $x_j(t)$, $j = r(1)s$, an den Gitterpunkten z_r bis z_s vorgenommen, siehe Abbildung 3.2. Die Be-

rechnung der Koeffizienten $l_j(z)$ erfolgt hierbei nach der Vorschrift

$$l_j(z) = \frac{\prod_{i=r, i \neq j}^s (z - z_i)}{\prod_{i=r, i \neq j}^s (z_j - z_i)} \quad j = r(1)s .$$

Damit können die Zustände und Ortsableitungen von $x(z, t)$ an der Stelle z_k durch Differentiation nach z angenähert werden:

$$\left. \frac{\partial^i x(z, t)}{\partial z^i} \right|_{z=z_k} \approx \left. \frac{\partial^i L^{[r_i, s_i]}(z, t)}{\partial z^i} \right|_{z=z_k} . \quad (3.12)$$

Für jede Ordnung $i = 0, 1, 2$ der Ortsableitungen müssen die Parameter r_i und s_i der Näherung gesondert definiert werden. An den Rändern des Ortsbereichs werden meist Ordnungsreduktionen der Lagrange-Polynome notwendig, um die Stabilität der resultierenden DA-Gleichungen zu gewährleisten. Außerdem werden sogenannte „sliding differences“ angewandt, um Gleichungen bezüglich undefinierter Gitterpunkte wie zum Beispiel z_{-1} oder z_0 zu vermeiden [11, 19].

Die resultierenden Näherungen aus den Lagrange-Polynomen entsprechen im äquidistanten Fall denjenigen aus Taylorreihen, im nicht äquidistanten Fall nicht immer. Beispielsweise kann die Zentraldifferenz der ersten Ortsableitung (3.6) für nicht äquidistante Gitter mit Lagrange-Polynomen nicht erzeugt werden. Stattdessen ergibt sich bei Einbeziehung der Orte z_{k-1} , z_k und z_{k+1} mit Lagrange-Polynomen

$$\begin{aligned} \left. \frac{\partial x}{\partial z} \right|_{z_k} \approx & \frac{z_k - z_{k+1}}{(z_{k-1} - z_k)(z_{k-1} - z_{k+1})} x_{k-1} + \frac{2z_k - z_{k-1} - z_{k+1}}{(z_k - z_{k-1})(z_k - z_{k+1})} x_k + \\ & + \frac{z_k - z_{k-1}}{(z_{k+1} - z_{k-1})(z_{k+1} - z_k)} x_{k+1} , \end{aligned} \quad (3.13)$$

was im äquidistanten Fall wieder in (3.6) übergeht.

3.2 Finite-Volumen-Methode

Die Finite-Volumen-Methode (FV) unterteilt den Ortsbereich in eine endliche Anzahl von diskreten Kontrollvolumen. Der Grundgedanke der FV ist die Berechnung von Strömen an den Rändern und Zustandsgrößen in der Mitte der Volumenelemente. Diese physikalisch motivierte Herangehensweise erfordert im Gegensatz zur FD immer physikalisch sinnvoll formulierte Probleme. Die Randbedingungen müssen also mindestens einen Strom enthalten und zu den Pdgl. „passen“. Der Hauptvorteil der FV ist die Einhaltung von integralen Bilanzgleichungen, was bei der FD nicht der Fall ist.

Die Gitterpunkte werden als Zentrumspunkte der Kontrollvolumen definiert. Wie in Abbildung 3.3 zu sehen ist, sind zwei Vorgehensweisen denkbar, die im Vorverarbeitungswerkzeug SYPPROT den Gitterverfahren mit Namen „Grid Points“ bzw. „Volume Bounds“ entsprechen. Beim Verfahren „Grid Points“ werden die Gitterpunkte vorgegeben und die Volumengrenzen mittig dazwischen platziert. Man erhält also Stützstellen am Rand. Beim Verfahren „Volume Bounds“ definieren die Volumengrenzen die Aufteilung des Ortsbereichs und die Gitterpunkte werden mittig in die Kontrollvolumen gelegt. Dies entspricht einer intuitiven Vorgehensweise, allerdings liegen am Rand des Ortsbereichs keine Stützstellen.

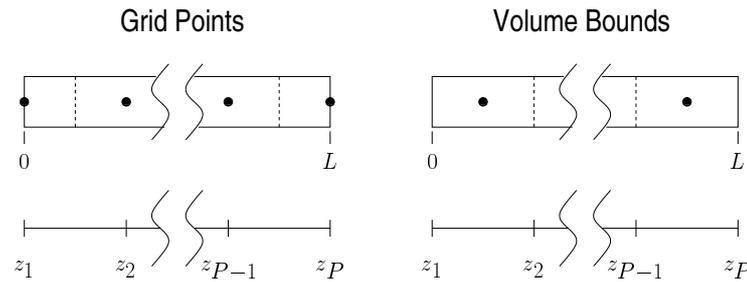


Abb. 3.3: Verfahren zur Gittereinteilung bei der FV

Die Diskretisierung der PDgl. nach der FV erfolgt in zwei Schritten: die Integration über jedes Kontrollvolumen und anschließende Approximation der Randwerte für jedes Volumenelement mittels Profilannahmen.

Dies soll anhand einer skalaren Reaktions-Konvektions-Diffusions-PDgl. für $x(z, t)$ mit Randbedingungen veranschaulicht werden. Die Anfangsbedingungen werden dabei nicht betrachtet und wir beschränken uns auf die Formalismen des Gitterverfahrens „Grid Points“.

$$\frac{\partial x}{\partial t} = -v \frac{\partial x}{\partial z} + D \frac{\partial^2 x}{\partial z^2} + s(x) \quad t > 0, \quad 0 < z < L \quad (3.14)$$

$$0 = -v \cdot (x_{zu} - x|_{z=0}) - D \left. \frac{\partial x}{\partial z} \right|_{z=0}, \quad 0 = \left. \frac{\partial x}{\partial z} \right|_{z=L} \quad t > 0$$

Nach Integration der PDgl. über das k -te Kontrollvolumen mit den Randpunkten z_k^- und z_k^+ (zu den Bezeichnungen siehe Abbildung 3.4) erhält man

$$(z_k^+ - z_k^-) \frac{dx_k}{dt} = -v (x_k^+ - x_k^-) + D \left(\left. \frac{\partial x}{\partial z} \right|_{z_k^+} - \left. \frac{\partial x}{\partial z} \right|_{z_k^-} \right) + s(x_k) \quad k = 2(1)P - 1. \quad (3.15)$$

Nun müssen noch die unbekanntenen Zwischenwerte x_k^\pm und $\partial x / \partial z|_{z_k^\pm}$ approximiert werden. Dazu werden sogenannte Profilannahmen getroffen, worin festgelegt wird, wie die verbliebenen Größen auf den Kontrollvolumengrenzen durch Werte bezüglich der Gitterpunkte beschrieben werden. Die einfachsten Profilannahmen sind

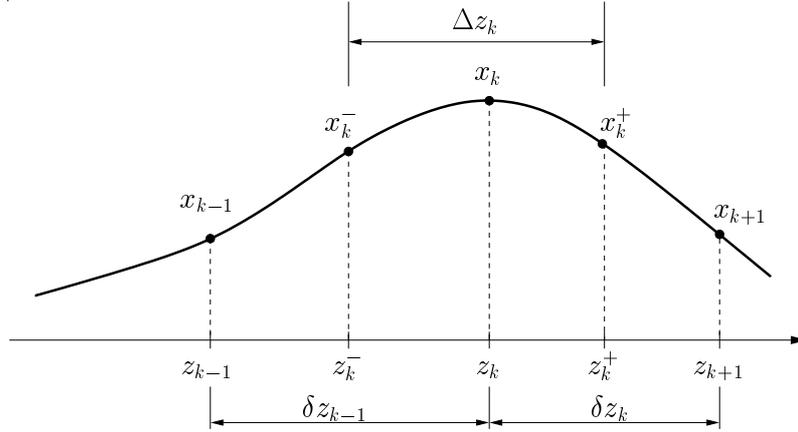


Abb. 3.4: Bezeichnungen der FV

- piecewise constant upwind:

$$x_k^+ = x_k, \quad x_k^- = x_{k-1}, \quad \left. \frac{\partial x}{\partial z} \right|_{z_k^+} = 0, \quad \left. \frac{\partial x}{\partial z} \right|_{z_k^-} = 0$$

- piecewise constant downwind:

$$x_k^+ = x_{k+1}, \quad x_k^- = x_k, \quad \left. \frac{\partial x}{\partial z} \right|_{z_k^+} = 0, \quad \left. \frac{\partial x}{\partial z} \right|_{z_k^-} = 0$$

- piecewise-linear:

$$x_k^+ = \frac{1}{2}(x_{k+1} + x_k), \quad x_k^- = \frac{1}{2}(x_k + x_{k-1})$$

$$\left. \frac{\partial x}{\partial z} \right|_{z_k^+} = \frac{x_{k+1} - x_k}{\delta z_k}, \quad \left. \frac{\partial x}{\partial z} \right|_{z_k^-} = \frac{x_k - x_{k-1}}{\delta z_{k-1}}.$$

Verwendung von „piecewise constant upwind“ für x_k^\pm und „piecewise linear“ für $\partial x / \partial z|_{z_k^\pm}$ führt für Gleichung (3.15) zu

$$\Delta z_k \frac{dx_k}{dt} = -v(x_k - x_{k-1}) + D \left(\frac{x_{k+1} - x_k}{\delta z_k} - \frac{x_k - x_{k-1}}{\delta z_{k-1}} \right) + s(x_k), \quad (3.16)$$

worin mit $\Delta z_k = 0.5(z_{k+1} - z_{k-1})$ und $\delta z_k = z_{k+1} - z_k$ nur noch Größen bezüglich der Gitterpunkte vorkommen.

An den Rändern des Ortsbereichs erfolgt eine gesonderte Behandlung der Volumengrenzen. Hier werden vor den Profilannahmen die jeweiligen Randbedingungen eingesetzt. Deshalb muss ein mit der FV zu behandelndes Problem physikalisch sinnvoll

formuliert werden, da sich ansonsten in diesem Schritt Schwierigkeiten bei der Elimination der Randwerte ergeben. Für den linken Rand der PDgl. (3.14) wird dies für das Gitterverfahren „Grid Points“ gezeigt. Aus (3.15) folgt mit $k = 1$, $z_1^- = z_1$ und $x_1^- = x_1$ (man beachte, dass am Rand nur ein halbes Kontrollvolumen vorliegt)

$$(z_1^+ - z_1) \frac{dx_1}{dt} = -v (x_1^+ - x_1) + D \left(\left. \frac{\partial x}{\partial z} \right|_{z_1^+} - \left. \frac{\partial x}{\partial z} \right|_{z_1} \right) + s(x_1). \quad (3.17)$$

Eliminiert man die Größen x_1 und $\partial x / \partial z|_{z_1}$ auf der rechten Seite mit der linken Randbedingung am Punkt $z_1 = 0$

$$0 = -v \cdot (x_{zu} - x_1) - D \left. \frac{\partial x}{\partial z} \right|_{z_1},$$

erhält man aus (3.17)

$$(z_1^+ - z_1) \frac{dx_1}{dt} = -v (x_1^+ - x_{zu}) + D \left. \frac{\partial x}{\partial z} \right|_{z_1^+} + s(x_1),$$

woraus sich nach Einsetzen der Profilannahmen

$$\frac{1}{2}(z_2 - z_1) \frac{dx_1}{dt} = -v (x_1 - x_{zu}) + D \frac{x_2 - x_1}{\delta z_1} + s(x_1) \quad (3.18)$$

ergibt. Die diskretisierte Randgleichung für das Randkontrollvolumen ist also ebenfalls eine Dgl. Die Diskretisierung des rechten Randes erfolgt analog. Für nähere Ausführungen sei auf [22] verwiesen.

3.3 Adaptive Verfahren

Wie man anhand der Gleichungen (3.6) bis (3.9) sehen kann, ist der Diskretisierungsfehler oft proportional zu oder zumindest quadratisch in Δz . Um eine ausreichende Genauigkeit der errechneten Lösung zu erzielen, legt dieser Zusammenhang eine größtmögliche Reduzierung der Abstände von Gitterpunkten nahe. Dies ist weniger wichtig in Bereichen der Lösung, wo ein glatter Verlauf mit geringen Ortsgradienten und geringen Krümmungen vorliegt, da die Ortsableitungen und damit die Fehler dann sehr klein sind. Treten in einem Problem allerdings große, zeitlich veränderliche Ortsgradienten oder starke Krümmungen auf, wird man eine hohe Anzahl von Ortsstützstellen verwenden müssen, um den Diskretisierungsfehler klein zu halten. Da sich im äquidistanten Fall dann auch in unkritischen Bereichen viele Stützstellen befinden, wird die Folge dieses Vorgehens eine unnötig lange Rechenzeit und damit eine ineffiziente Berechnung sein.

Der Übergang zu zeitlich festen, nicht äquidistanten Gittern bringt nur dann eine Verbesserung, wenn die auftretenden Fronten im Ortsprofil örtlich fixiert sind. Dies

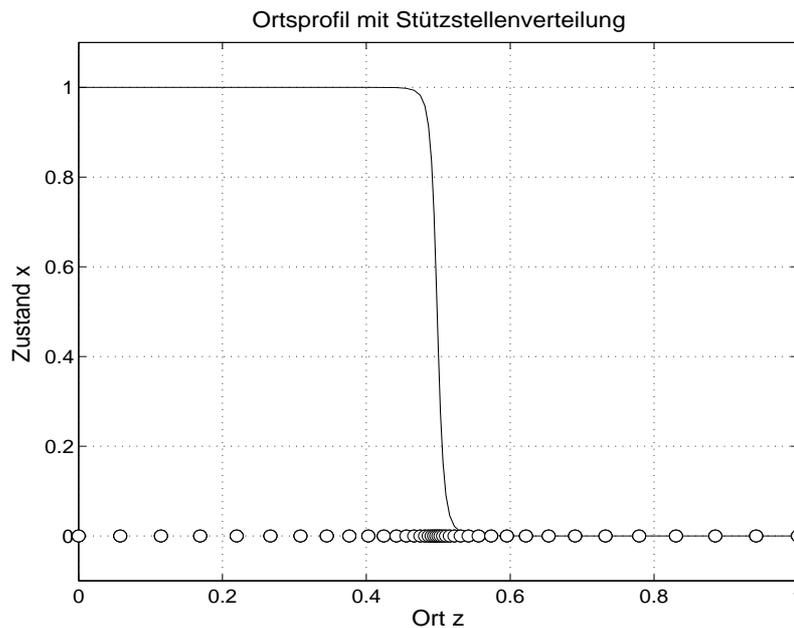


Abb. 3.5: Ortsprofil (—) mit an die Front angepasster Stützstellenverteilung (◦◦◦).

ist bei stehenden Wellen im Ortsprofil der Fall. Meist werden die Fronten in verteilten dynamischen Systemen aber zeitlich und örtlich veränderlich sein.

Ein Behebung der Nachteile erfolgt, wenn die Gitterpunkte nicht von vorneherein fest vorgegeben sind, sondern während der Berechnung der Lösung angepasst werden, siehe Abbildung 3.5. Dadurch ist es möglich, mit einer geringen Anzahl von Stützpunkten auszukommen, die dann bei Bedarf verstärkt in Bereiche großer Ortsableitung oder Krümmung gelegt werden. Die Effizienz der Berechnung wird damit gesteigert, ohne dass der Diskretisierungsfehler zu groß wird und die Genauigkeit leidet.

Die verschiedenen Algorithmen zur Benutzung eines adaptiven Gitters führen stets die beiden Schritte *Schätzung des Fehlers der Ortsdiskretisierung* und *Neuverteilung der Stützstellen* aus.

Man kann die Moving-Grid-Verfahren grob in Static-Regridding- und Dynamic-Regridding-Verfahren einteilen. Bei den *Static-Regridding-Verfahren* wird nach jedem oder nach einer bestimmten Anzahl von Zeitschritten die Integration angehalten, und es wird aufgrund der Fehlerschätzung eine Neuverteilung der Stützstellen vorgenommen, die besser an den aktuellen Lösungsverlauf angepasst ist. Dazu ist eine Interpolation der Lösung auf dem alten Gitter auf eine Lösung auf dem neuen Gitter notwendig. Die Trennung von Zeitschrittberechnung und Gitterbewegung führt im Allgemeinen zu Geschwindigkeitsvorteilen. Die Interpolation benötigt allerdings wieder Rechenzeit und bedingt weitere Fehler. Auf diese Verfahrensgruppe soll im Weiteren nicht eingegangen werden.

Bei den *Dynamic-Regridding-Verfahren* werden die Fehlerschätzung sowie die Gitter-

bewegung simultan zur Zeitintegration vorgenommen, damit entfällt die Interpolation. Dazu werden die Stützstellen des Ortsbereichs als Zustandsvariablen mit in das DA-Gleichungssystem aufgenommen. Die Gleichungen für die Stützstellensteuerung sind dann die Zustandsgleichungen dieser neuen Variablen. Liegen N PDgln. und P Ortsstützstellen vor, erweitert sich das Gleichungssystem von $N \cdot P$ Gleichungen im Fall ohne Stützstellensteuerung auf $(N + 1) \cdot P$ Gleichungen. Die Trajektorien $z_k(t)$ des Ortsgitters weisen außerdem keine Sprünge wie bei Static-Regridding-Verfahren auf. Kombinationen aus Static- und Dynamic-Regridding-Verfahren sind ebenfalls möglich.

Der Aufwand für die Berücksichtigung der zusätzlichen Gitterbewegungs-Gleichungen muss sich natürlich auszahlen. So sollte die Simulation mittels Moving-Grid-Verfahren bei gleichem Rechenaufwand eine bessere oder bei geringerem Rechenaufwand eine gleich gute Lösungsgenauigkeit erreichen.

In den folgenden Unterkapiteln wird das in dieser Arbeit benutzte Prinzip der Gleichverteilung eines Fehlermaßes zur Gittersteuerung eingeführt. Daneben werden verschiedene Vorgehensweisen zur Fehlerabschätzung beschrieben. Schließlich gibt ein Überblick zu weiteren Moving-Grid-Verfahren Aufschluss über alternative Strategien zur Stützstellensteuerung.

3.3.1 Gleichverteilung

Eine weit verbreitetes Prinzip der Stützstellensteuerung ist die Gleichverteilung, engl. equidistribution. Dieser Ansatz ist als Static- wie auch als Dynamic-Regridding-Verfahren realisierbar. Bei der Gleichverteilung soll das Verhältnis zwischen Fehlermaß und Stützstellenkonzentration lokal konstant gehalten werden. Anders gesagt: ein hohes Fehlermaß bedingt eine hohe Dichte der Stützstellen, ein niedriges Fehlermaß bedingt weiter auseinanderliegende Gitterpunkte. Damit erreicht man, dass der Diskretisierungsfehler über den gesamten Ortsbereich in etwa konstant ist, dass also eine Gleichverteilung dieses Fehlers vorliegt.

Eine Monitorfunktion $M(x, z)$ liefert eine Abschätzung für den lokalen Diskretisierungsfehler. Damit lässt sich das Gleichverteilungsprinzip folgendermaßen beschreiben:

$$\int_{z_k}^{z_{k+1}} M(x, z) dz = \int_{z_{k-1}}^{z_k} M(x, z) dz = \text{const.} \quad k = 2(1)P - 1 \quad (3.19)$$

oder diskretisiert

$$M(x, z)|_{z_k^+} \cdot (z_{k+1} - z_k) = M(x, z)|_{z_k^-} \cdot (z_k - z_{k-1}) \quad k = 2(1)P - 1 \quad (3.20)$$

Zur Notation: $M(x, z)|_{z_k^+}$ bezeichnet die Approximation von $M(x, z)$ im Bereich zwischen z_k und z_{k+1} , da dieser Wert mit der Differenz $z_{k+1} - z_k$ ins Verhältnis gesetzt wird. Im Folgenden wird dies mit $M_k^+ \equiv M(x, z)|_{z_k^+}$ abgekürzt.

Auftretende Probleme der Stützstellensteuerung sind vor allem das Überkreuzen von Gitterpunkten, starke Konzentration von Stützstellen oder Schwingungen in den Zeitverläufen der Gitterpunkte. Dazu kommen Konvergenzprobleme und zu kleine Zeitschrittweiten der numerischen Lösungsalgorithmen. Dies versucht man zu verhindern, indem die Monitorwerte M_k^+ oder die Punktdichten $n_k^+ = (z_{k+1} - z_k)^{-1}$ einer örtlichen Glättung unterzogen werden. Dadurch halten die Stützpunkte maximale und minimale Abstände voneinander ein. Außerdem kann durch Ableiten der Beziehung (3.19) nach z_k oder durch Einbau von Verzögerungstermen eine zeitliche Glättung der Gittertrajektorien erreicht werden. Für Details sei auf das Kapitel 4.2.3 sowie auf [14] oder [9] verwiesen.

3.3.2 Fehlerschätzung

Die Schätzung des Fehlers wird in den meisten Fällen mit sogenannten *Monitor-Funktionen* M_k^+ geschehen. Allgemein ist M_k^+ eine diskrete Ersetzung für ein von der Lösung abhängiges Funktional $M(x, z)$ und enthält eine oder mehrere Ortsableitungen. Hier werden vor allem heuristische Kriterien wie lokale Steigung oder lokale Krümmung verwendet. Diese besitzen allerdings keinen direkten Zusammenhang zum lokalen Diskretisierungsfehler, können aber bei vergleichsweise geringem Aufwand eine explizite Berechnung dieses Fehlers erfolgreich ersetzen. Einige der gebräuchlichsten Fehler-Kriterien werden im Folgenden vorgestellt.

Bogenlängenmonitor

Der Bogenlängenmonitor berechnet sich nach dem Zusammenhang für die Bogenlänge einer Funktion. Da er die erste Ortsableitung enthält, ist er ein Maß für die lokale Steigung bezüglich des Ortes. In Verbindung mit dem Ansatz der Gleichverteilung bewirkt der Bogenlängenmonitor eine Platzierung der Gitterpunkte in lokal gesehen gleichförmigen Abständen entlang des Ortsprofils.

$$M_k^+ = \sqrt{1 + \left(\frac{\partial x}{\partial z} \Big|_{z_k^+} \right)^2} \quad (3.21)$$

Krümmungsmonitor

Durch das Bogenlängen-Kriterium werden diejenigen Bereiche der Lösung, wo ein annähernd linearer Verlauf mit großer örtlicher Steigung auftritt, als wichtig betrachtet, während den Kurvenkrümmungen eventuell zuwenig Bedeutung zugemessen wird. Dem kann durch einen Krümmungsmonitor abgeholfen werden, welcher hohe Werte

für Bereiche großer Krümmung des Lösungsverlaufs liefert.

$$M_k^+ = \sqrt{1 + \left| \frac{\partial^2 x}{\partial z^2} \Big|_{z_k^+} \right|} \quad (3.22)$$

Kombination

Die letzte vorgestellte heuristische Monitorfunktion ist eine Kombination aus erster und zweiter Ortsableitung und kann mit den Parametern a_1 und a_2 gewichtet werden.

$$M_k^+ = 1 + a_1 \left| \frac{\partial x}{\partial z} \Big|_{z_k^+} \right| + a_2 \left| \frac{\partial^2 x}{\partial z^2} \Big|_{z_k^+} \right| \quad (3.23)$$

Diskretisierungsfehler

Es ist auch möglich, während der Simulation den lokalen Diskretisierungsfehler direkt zu berechnen. Liegt zum Beispiel eine PDgl. der Form

$$\frac{\partial x}{\partial t} = -v \frac{\partial x}{\partial z} + D \frac{\partial^2 x}{\partial z^2} + s(x)$$

vor, so errechnet sich das Fehlermaß nach

$$M_k^+ = -ve_1 + De_2 ,$$

wobei die Fehlerterme e_1 und e_2 aus den Termen höherer Ordnung der Näherungen (3.6) bis (3.8) abgeschätzt werden können. Allerdings müssen hierbei Approximationen höherer Ortsableitungen verwendet werden, was wiederum Fehler, numerische Störungen und Instabilitäten einschleust. Außerdem erfordern die Approximationen höherer Ortsableitungen einen erhöhten Rechenaufwand. Verschiedene Autoren raten deshalb von einer direkten Betrachtung dieses Fehlers ab (z.B. [14]). Durch dieses Vorgehen ist jedoch die Einhaltung einer vorgegebenen Fehlerschranke für den lokalen Diskretisierungsfehler möglich [21].

Der Diskretisierungsfehler kann auch durch Vergleich zweier Lösungen berechnet werden [1]. Dazu werden zwei unterschiedliche Gitter mit verschiedener Stützstellenanzahl verwendet, und in jedem Zeitschritt die Lösung auf beiden Gittern separat berechnet. Aus den Unterschieden der Lösungen kann der Diskretisierungsfehler abgeschätzt werden. Durch die auf verschiedenen Gittern berechnete numerische Lösung ist wieder ein Anstieg des Rechenaufwands festzustellen.

Fehlerschätzung bei PDgl.-Systemen

Eine Simulation von PDgln.-Systemen macht eine Anpassung der Monitorfunktion für die Überwachung mehrerer Zustände notwendig. Dabei ist zusätzlich eine Normierung der Zustände sowie des Ortsbereiches sinnvoll, um die Vergleichbarkeit der geschätzten Ortsableitungen zu gewährleisten. Auf die genauen Formalismen wird hier nicht näher eingegangen, sondern auf Kapitel 4.2.2 verwiesen.

3.3.3 Weitere Verfahren zur Stützstellensteuerung

In diesem Unterkapitel werden einige weitere Ansätze zur Adaption eines bewegten Gitters an die numerische Lösung von PDgln. beschrieben. Dabei wird kurz auf Vor- und Nachteile der Verfahren, auch im Hinblick auf eine Implementierung in SYP-PROT, eingegangen.

Minimierung von Funktionalen

Eine weitere Möglichkeit, die näherungsweise Gleichverteilung des Diskretisierungsfehlers zu erreichen, ist die Minimierung von bestimmten Zielfunktionalen in Abhängigkeit des Lösungsverlaufs. Dabei wird beispielsweise die Bewegung der Gitterpunkte, dz/dt , so gewählt, dass

$$J = \left(\frac{dx}{dt}\right)^2 + \alpha \left(\frac{dz}{dt}\right)^2 = \left(\frac{\partial x}{\partial t} + \frac{\partial x}{\partial z} \frac{dz}{dt}\right)^2 + \alpha \left(\frac{dz}{dt}\right)^2, \quad \alpha \geq 0,$$

minimiert wird. Differentiation von J nach dz/dt und Nullsetzen ergibt die Dgl.

$$\frac{dz}{dt} = -\frac{\partial x}{\partial t} \frac{\partial x}{\partial z} \bigg/ \left(\alpha + \left(\frac{\partial x}{\partial z}\right)^2 \right),$$

wodurch die Trajektorien der Gitterpunkte $z_k(t)$ so berechnet werden, dass die totale Ableitung dx/dt minimal im Sinne des obigen Funktionalen wird. Allerdings werden die Stützstellen durch dieses Verfahren nicht immer in die gewünschte Richtung, das heißt in Bereiche hoher Ortsgradienten, verschoben [5].

Moving-Point-Verfahren

Im Moving-Point-Verfahren werden einzelne Gitterpunkte mittels A-priori-Wissen aus analytischen Untersuchungen des Modells an die bewegte Front angekoppelt. Dadurch ist die ausreichende örtliche Auflösung der wandernden Front gewährleistet und der Diskretisierungsfehler wird klein gehalten, ohne zu viele Gitterpunkte in unkritischen Bereichen zu verschwenden. Dieses Verfahren ist deshalb vor allem zur

Behandlung von wandernden Diskontinuitäten geeignet, bei denen die Frontgeschwindigkeit analytisch im Voraus berechnet werden kann. Wird dieses A-priori-Wissen dem Vorverarbeitungswerkzeug mitgeteilt, wäre eine einfache und effiziente Art der Gittersteuerung möglich. Soll die Vorverarbeitung der PDgl. aber unabhängig von vorher durchgeführten analytischen Berechnungen sein, ist eine Einbindung dieses Verfahrens nicht ohne die Entwicklung von automatischen Analyse-Werkzeugen zur Charakterisierung der Frontwanderung möglich.

Multigrid-Verfahren

Bei Multigrid-Verfahren werden mehrere Gitter unterschiedlicher Maschenweite verwendet. Ausgehend vom größten Gitter werden sukzessive die Lösungen auf allen Gittern berechnet und dabei die Informationen der gröberen Gitter verwendet. Bei sachgemäßer Anwendung dieser Verfahren kann die Konvergenzgeschwindigkeit vor allem für dünn besetzte Gleichungssysteme im Vergleich zu Verfahren mit nur einem feinen Gitter beträchtlich erhöht werden, da sowohl die groben als auch die feinen Anteile der Dynamik erfasst werden. Mögliche Probleme betreffen die Konvergenz, variierende Stützstellenanzahl und Ineffizienz in Bereichen mit hoher Genauigkeit schon bei groben Gittern. Nachteilig wirkt sich auf jeden Fall der hohe Rechenaufwand für die verwendeten Gitter aus.

Moving-Finite-Elemente-Verfahren (MFE)

Moving-Finite-Elemente-Verfahren [17, 7] benutzen zur Näherung der Lösung der PDgl. bestimmte Basisfunktionen, die jeweils zwischen zwei Stützstellen gültig sind:

$$x(z, t) \approx x^*(z, t) = \sum_{i=1}^P a_i(t) p_i(z, z_i(t))$$

mit den Amplituden der Lösungsfunktion $a_i(t)$ und den polynomialen Basisfunktionen $p_i(z, z_i(t))$. Diese Basisfunktionen p_i werden vorab als einfache, stückweise definierte Funktionen wie Geraden- oder Parabelstücke festgelegt. Die beweglichen Stützstellen werden so platziert, dass das Quadrat der L_2 -Norm des Residuums der PDgl. $\partial x / \partial t = f(x, \partial x / \partial z, \partial^2 x / \partial z^2)$ über den gesamten Ortsbereich bezüglich da_i/dt und dz_i/dt minimiert wird:

$$\left(\left\| \frac{\partial x^*}{\partial t} - f \left(x^*, \frac{\partial x^*}{\partial z}, \frac{\partial^2 x^*}{\partial z^2} \right) \right\|_2 \right)^2 \stackrel{!}{=} \min.$$

mit

$$\frac{\partial x^*}{\partial t} = \sum_{i=1}^P \left(p_i(z, z_i(t)) \frac{da_i}{dt} + \frac{\partial x^*}{\partial z_i} \frac{dz_i}{dt} \right). \quad (3.24)$$

Diese Optimierungsaufgabe führt auf $2P$ gewöhnliche Differentialgleichungen für die Amplituden a_i sowie die Gitterpunkte z_i . Durch Einführung sogenannter Penalty-Funktionen kann zusätzlich die Knotenbewegung reglementiert und die Lösbarkeit der Optimierungsaufgabe gewährleistet werden. Als Nachteil der MFE ist vor allem die Anzahl der Parameter für Korrekturterme und Knotenverteilung zu nennen, was die Einbindung in ein automatisch ablaufendes Programm erschwert.

Kollokationsverfahren

Ähnlich wie beim Finite-Elemente-Verfahren werden durch das Kollokationsverfahren angenäherte Lösungen aus gewissen Basisfunktionen berechnet. Die Amplituden $a_i(t)$ werden hier so bestimmt, dass die Dgl. an einer definierten Anzahl von Ortsstützstellen, den sogenannten Kollokationspunkten, möglichst gut erfüllt ist.

Kapitel 4

Darstellung des Moving-Grid-Verfahrens

Das hier dargestellte Moving-Grid-Verfahren basiert auf einer Arbeit von *Verwer et al.* [24] und geht ursprünglich auf *Dorfi/Drury* [3] zurück. Es verwendet eine konstante Anzahl von Gitterpunkten und bedient sich zur Stützstellensteuerung des Gleichverteilungsprinzips mit Fehlerschätzung durch einen Bogenlängen-Monitor. Zur Reglementierung der Gitterbewegung können sowohl eine örtliche als auch eine zeitliche Glättung der Punktdichten angewandt werden.

Ausschlaggebend für die Auswahl dieses Verfahrens waren zum einen die Einfachheit und die geringe Anzahl der einzustellenden Parameter, zum anderen die Zugehörigkeit zu den Dynamic-Regridding-Verfahren, was eine reibungslose Integration in SYP-PROT ohne Eingriffe in die DAE-Numerik von DIVA erlaubt.

Zu lösen sei eine skalare partielle Differentialgleichung vom parabolischen Typ für den Zustand $x(z, t)$ mit einer Ortskoordinate z und der Zeitvariablen t :

$$\frac{\partial x}{\partial t} = f \left(x, \frac{\partial x}{\partial z}, \frac{\partial^2 x}{\partial z^2}, u, z, t \right) \quad t > t_0, \quad 0 < z < L \quad (4.1)$$

$$g_0 \left(x, \frac{\partial x}{\partial z}, z = 0, t \right) = 0, \quad g_L \left(x, \frac{\partial x}{\partial z}, z = L, t \right) = 0 \quad t > t_0 \quad (4.2)$$

$$x(z, t_0) = x_0(z) \quad 0 \leq z \leq L. \quad (4.3)$$

Dies stellt eine Reduzierung des allgemeinen PDgl.-Systems (3.1) bis (3.3) auf den Fall $N = 1$ dar. Eine Übertragung auf den mehrdimensionalen Fall $N > 1$ ist leicht möglich. Es wird von der Existenz und Eindeutigkeit der Lösung $x(z, t)$ des Problems ausgegangen.

Das Ziel ist eine möglichst effiziente Berechnung des Verlaufs von $x(z, t)$, wobei die Abweichungen von der exakten Lösung so klein wie möglich bleiben sollen.

4.1 Gleichungs-Transformation

Die Diskretisierung nach der MOL geschieht auf einem veränderlichen Ortsgitter mit P Gitterpunkten der Form

$$0 = z_1 < z_2(t) < \dots < z_{P-1}(t) < z_P = L \quad t \geq 0. \quad (4.4)$$

Der erste und letzte Punkt des Ortsgitters sind Fixpunkte auf den Rändern. Die Größen $z_k(t)$ sind glatte Zeittrajektorien und müssen als zusätzliche Zustandsgrößen des Problems aufgefaßt werden.

Die diskretisierten, nur noch zeitabhängigen Zustandsgrößen werden in der Form

$$x_k(t) \equiv x(z_k(t), t) \quad k = 1(1)P \quad (4.5)$$

notiert. Sie stellen eine Näherung der exakten Lösung $x(z, t)$ am Punkt $(z_k(t), t)$ dar. Zur Herleitung der transformierten PDgl. wird die totale Ableitung von $x(z(t), t)$ gebildet:

$$\frac{dx}{dt} = \frac{\partial x}{\partial z} \frac{dz}{dt} + \frac{\partial x}{\partial t}.$$

Daraus ergibt sich die Lagrange-Form der partiellen Zeitableitung [4]

$$\frac{\partial x}{\partial t} = \frac{dx}{dt} - \frac{\partial x}{\partial z} \frac{dz}{dt}$$

oder in diskretisierter Form

$$\frac{\partial x_k}{\partial t} = \frac{dx_k}{dt} - \frac{\partial x}{\partial z} \Big|_{z_k(t)} \cdot \frac{dz_k}{dt}. \quad (4.6)$$

Einsetzen in die Diskretisierung der PDgl. (4.1) ergibt

$$\frac{dx_k}{dt} - \frac{\partial x}{\partial z} \Big|_{z_k(t)} \cdot \frac{dz_k}{dt} = \tilde{f} \left(x_k, \frac{\partial x}{\partial z} \Big|_{z_k(t)}, \frac{\partial^2 x}{\partial z^2} \Big|_{z_k(t)}, u_k, t \right) \quad k = 2(1)P - 1. \quad (4.7)$$

$\tilde{f}(\dots)$ bzw. $\partial x / \partial z|_{z_k(t)}$ und $\partial^2 x / \partial z^2|_{z_k(t)}$ bezeichnen die Approximationen des Operators f bzw. der Ortsableitungen am Ort $z_k(t)$. Für diese Terme können wie gewöhnlich Diskretisierungsverfahren wie Finite Differenzen oder Finite Volumen verwendet werden.

Gleichung (4.7) stellt die transformierte und diskretisierte PDgl. dar, die anstatt der ursprünglichen Gleichung (4.1) zur Simulation verwendet wird muss. Der hinzugefügte Transformationsterm $\partial x / \partial z|_{z_k(t)} \cdot dz_k / dt$ berücksichtigt die Mitbewegung des Koordinatensystems.

Die Diskretisierung der Randbedingungen (4.2) erfolgt in gewohnter Weise und kann ohne Transformation durchgeführt werden, da hier keine Zeitableitungen auftreten.

Sie führt auf folgende Gleichungen, wobei $\tilde{g}_0(\dots)$ bzw. $\tilde{g}_L(\dots)$ die diskreten Approximationen der Operatoren g_0 bzw. g_L sind:

$$\tilde{g}_0 \left(x_1, \frac{\partial x}{\partial z} \Big|_{z_1}, t \right) = 0, \quad \tilde{g}_L \left(x_P, \frac{\partial x}{\partial z} \Big|_{z_P}, t \right) = 0. \quad (4.8)$$

Damit wurde das ursprüngliche Problem bezüglich des bewegten Gitters transformiert und die MOL-Diskretisierung der PDgln. durchgeführt.

4.2 Moving-Grid-Gleichungen

Da mit den Größen $z_k(t)$ neue Zustände eingeführt wurden, muss auch die Zahl der Modellgleichungen erhöht werden. Dabei sollen Gleichungen angesetzt werden, welche die Bewegung der Gitterpunkte $z_k(t)$ in Abhängigkeit der Lösungsverläufe $x_k(t)$ steuern. Diese Gleichungen werden als Moving-Grid-Gleichungen oder als Regularisierungsfunktionen bezeichnet. Mit Hilfe der Gleichverteilung eines Fehlermaßes für den lokalen Diskretisierungsfehler werden die beweglichen Gitterpunkte in Bereiche großer Ortsgradienten verschoben. Um Instabilitäten der Gitterbewegung zu vermeiden, wird diese mittels örtlichen und gegebenenfalls zeitlichen Glättungen gedämpft. Daraus resultieren ebenfalls Zwangsbedingungen für eingeschränkte Beweglichkeit, um Überkreuzungen von Gitterpunkten zu verhindern.

4.2.1 Gleichverteilung

Als Abkürzungen werden die Differenz zweier Gitterpunkte

$$\Delta z_k(t) := z_{k+1}(t) - z_k(t) \quad k = 1(1)P - 1 \quad (4.9)$$

und die lokale Punktdichte im Ortsbereich zwischen $z_k(t)$ und $z_{k+1}(t)$

$$n_k^+(t) := \frac{1}{\Delta z_k(t)} = \frac{1}{z_{k+1}(t) - z_k(t)} \quad k = 1(1)P - 1 \quad (4.10)$$

eingeführt.

Als erste Grundlage für die Gittergleichungen wird die Beziehung für Gleichverteilung (3.20) herangezogen, welche besagt, dass das Verhältnis zwischen Punktdichte und Fehlermaß lokal gesehen konstant sein soll. Der Übersichtlichkeit halber wird im Folgenden die Abhängigkeit (t) meist weggelassen.

$$\frac{n_{k-1}^+}{M_{k-1}^+} = \frac{n_k^+}{M_k^+} \quad k = 2(1)P - 1. \quad (4.11)$$

M_k^+ ist eine diskrete Monitorfunktion und damit ein Fehlermaß für den Ortsbereich zwischen $z_k(t)$ und $z_{k+1}(t)$. Sie wird im nächsten Abschnitt näher spezifiziert. Diese algebraischen Beziehungen können in der Zeitintegration simultan zu den übrigen Differential-Algebra-Gleichungen gelöst werden.

4.2.2 Die Monitorfunktion

Als Maß für die Steigung bezüglich des Ortes, d.h. für die Größe des Ortsgradienten, wird der Bogenlängen-Monitor (3.21) in diskretisierter Form verwendet:

$$M_k^+(t) := \sqrt{1 + \left(\frac{L}{X} \cdot \frac{x_{k+1} - x_k}{z_{k+1} - z_k} \right)^2} \quad k = 1(1)P - 1 . \quad (4.12)$$

Dabei wird durch die beiden Faktoren L und X eine Normierung des Ortsbereiches und der Zustandsgrößen erreicht. Der Ortsbereich wird auf die Länge 1 normiert, die Werte der Zustandsgrößen x_k werden zu einem typischen oder maximalen Wertebereich X in Beziehung gesetzt. Liegen mehrere PDgln. auf demselben Gitter vor, muss eine Modifikation für mehrere Zustände verwendet werden. Hierbei bezeichnet N die Anzahl der betrachteten PDgln. und X_j den Normierungsfaktor des j -ten Zustands.

$$M_k^+(t) := \sqrt{1 + \frac{1}{N} \sum_{j=1}^N \left(\frac{L}{X_j} \cdot \frac{x_{k+1,j} - x_{k,j}}{z_{k+1} - z_k} \right)^2} \quad k = 1(1)P - 1 . \quad (4.13)$$

Es werden alle Zustandsgrößen zur Fehlerabschätzung verwendet. Als Alternative kann man aber auch einen oder mehrere Zustände als sogenannte Leitzustände wählen und die Berechnung der Monitorwerte nur von diesen Größen abhängig machen. Dann läuft die Summe in Gleichung (4.13) nicht über alle N Zustände, sondern beinhaltet nur die ausgewählten Leitzustände.

In der vorliegenden Fassung sind die Monitorwerte auf streng positive Werte festgelegt, also $M_k^+ \geq 1 > 0$.

4.2.3 Gitterglättung

Die bisher abgeleitete Form der Moving-Grid-Gleichungen hat den Nachteil, dass in der Gitterbewegung Überkreuzen und starke Konzentrationen der Stützpunkte sowie Schwingungen in den Gittertrajektorien auftreten können. Außerdem kann durch numerische Experimente nachgewiesen werden, dass die zum Lösen von Gleichungen oder im Integrator angewandte Newton-Iteration oft nicht konvergiert oder die Simulation wegen zu kleinen Zeitschrittweiten des Integrators abgebrochen wird [24, 14]. Deshalb werden im Folgenden zwei Modifikationen der Gleichverteilungsbeziehung (4.11) zur Glättung der Gitterbewegung bezüglich des Ortes und der Zeit eingeführt.

Örtliche Gitterglättung

Um die örtliche Gitterbewegung einzuschränken, werden die Punktdichten $n_k^+(t)$ durch ihre sogenannten „anti-diffusen“ Entsprechungen $\tilde{n}_k^+(t)$ ersetzt, was eine numerische Diffusion der Punktdichten $n_k^+(t)$ bewirkt. Dabei wird der Parameter κ der

örtlichen Gitterglättung eingeführt. Eine anschauliche Interpretation von κ folgt am Ende des Kapitels.

$$\tilde{n}_1^+ := n_1^+ - \kappa(\kappa + 1)(n_2^+ - n_1^+) \quad (4.14)$$

$$\tilde{n}_k^+ := n_k^+ - \kappa(\kappa + 1)(n_{k+1}^+ - 2n_k^+ + n_{k-1}^+) \quad k = 2(1)P - 1 \quad (4.15)$$

$$\tilde{n}_{P-1}^+ := n_{P-1}^+ - \kappa(\kappa + 1)(n_{P-2}^+ - n_{P-1}^+) \quad (4.16)$$

Die Gleichungen (4.14) und (4.16) ergeben sich aus (4.15) durch die „virtuellen“ Randbedingungen

$$n_0^+ := n_1^+, \quad n_P^+ := n_{P-1}^+.$$

Damit lautet die neue Gleichverteilungsbeziehung

$$\frac{\tilde{n}_{k-1}^+}{M_{k-1}^+} = \frac{\tilde{n}_k^+}{M_k^+} \quad k = 2(1)P - 1 \quad (4.17)$$

oder ausgeschrieben mit der Abkürzung $\mu := \kappa(\kappa + 1)$ für

$k = 2$:

$$\frac{-(1 + \mu)n_1^+ + \mu n_2^+}{M_1^+} + \frac{-\mu n_1^+ + (1 + 2\mu)n_2^+ - \mu n_3^+}{M_2^+} = 0 \quad (4.18)$$

$k = 3(1)P - 2$:

$$\frac{\mu n_{k-2}^+ - (1 + 2\mu)n_{k-1}^+ + \mu n_k^+}{M_{k-1}^+} + \frac{-\mu n_{k-1}^+ + (1 + 2\mu)n_k^+ - \mu n_{k+1}^+}{M_k^+} = 0 \quad (4.19)$$

$k = P - 1$:

$$\frac{\mu n_{P-3}^+ - (1 + 2\mu)n_{P-2}^+ + \mu n_{P-1}^+}{M_{P-2}^+} + \frac{-\mu n_{P-2}^+ + (1 + \mu)n_{P-1}^+}{M_{P-1}^+} = 0. \quad (4.20)$$

(4.18) bis (4.20) stellen ein algebraisches Gleichungssystem dar, das bezüglich der Stützstellen z_k 5-Punkt-verkoppelt ist, da Gleichung k von n_{k-2}^+ bis n_{k+1}^+ und damit von z_{k-2} bis z_{k+2} abhängt.

In [24] wird nachgewiesen, dass durch diese Modifikation benachbarte Punktdichten und damit auch benachbarte Intervallabstände stets folgender Beschränkung unterliegen:

$$\frac{\kappa}{\kappa + 1} \leq \frac{n_{k-1}^+}{n_k^+} \leq \frac{\kappa + 1}{\kappa} \quad k = 2(1)P - 1. \quad (4.21)$$

Damit ist über den Parameter $\kappa > 0$ ein einfacher und anschaulicher Weg gegeben, wie man Punkthäufungen und das Überkreuzen von Stützstellen vermeiden kann. Während eine Wahl von $\kappa \lesssim 1$ eine großzügige Verschiebung der Stützstellen erlaubt, wird mit $\kappa \gg 1$ die Bewegung der Gitterpunkte sehr gering sein.

Zeitliche Gitterglättung

Die Zeitverläufe der Gitterpunkte $z_k(t)$ können durch eine weitere Modifikation geglättet werden. Dazu wird die Gleichverteilungsbeziehung (4.17) nach örtlicher Glättung zu einem System von Differentialgleichungen erweitert:

$$\frac{\tilde{n}_{k-1}^+ + \tau \frac{d}{dt} \tilde{n}_{k-1}^+}{M_{k-1}^+} = \frac{\tilde{n}_k^+ + \tau \frac{d}{dt} \tilde{n}_k^+}{M_k^+} \quad k = 2(1)P - 1. \quad (4.22)$$

Der Parameter $\tau > 0$ wirkt als Verzögerung und zwingt das Gitter, sich nicht sofort den neuen Monitorwerten anzupassen, sondern sich über das Zeitintervall τ einzustellen. Das Resultat dieser zeitlichen Glättung ist die Vermeidung von zeitlichen Oszillationen und ein glatter Verlauf von $z_k(t)$. Wird für τ ein sehr kleiner Wert oder Null gewählt, tritt die Verzögerung kaum bzw. gar nicht auf. Bei sehr hohen Werten für τ ergibt sich ein praktisch nichtbewegtes Gitter. In den meisten Fällen ist die zeitliche Glättung jedoch überflüssig, da die rein örtliche Gitterglättung schon gute Ergebnisse liefert.

Es sei noch angemerkt, dass durch die zeitliche Glättung die Beschränkung der Verhältnisse benachbarter Punktdichten (4.21) weiterhin ihre Gültigkeit behält [24].

Die Gleichungen (4.22) sind in der bisherigen Form noch nicht direkt implementierbar, da Ableitungen von \tilde{n}_k^+ vorkommen. Im Kontext von SYPROT und DIVA ist dies eine sogenannte Hilfsvariable und hat die Funktion einer Abkürzung. Ableitungen bezüglich der Zeit dürfen in der linear-impliziten DA-Form nach Gleichung (2.1) aber nur von Zustandsvariablen gebildet werden. Deshalb müssen noch Umformungen vorgenommen werden, so dass in den Gleichungen nur Ableitungen von Zustandsvariablen auftreten. Aus

$$n_k^+ = \frac{1}{\Delta z_k}$$

folgt

$$\frac{d}{dt} n_k^+ = -\frac{1}{(\Delta z_k)^2} \frac{d}{dt} \Delta z_k = -(n_k^+)^2 \left(\frac{dz_{k+1}}{dt} - \frac{dz_k}{dt} \right)$$

Einsetzen in die Zeitableitung der Gleichungen (4.14) bis (4.16) ermöglicht eine Umformung von (4.22) und liefert die endgültigen Gleichverteilungsbeziehungen für

$k = 2 :$

$$\begin{aligned}
& \tau \left[\left(\frac{(1+\mu)(n_1^+)^2}{M_1^+} + \frac{\mu(n_1^+)^2}{M_2^+} \right) \frac{dz_1}{dt} \right. \\
& \quad - \left(\frac{(1+\mu)(n_1^+)^2 + \mu(n_2^+)^2}{M_1^+} + \frac{\mu(n_1^+)^2 + (1+2\mu)(n_2^+)^2}{M_2^+} \right) \frac{dz_2}{dt} \\
& \quad + \left(\frac{\mu(n_2^+)^2}{M_1^+} + \frac{(1+2\mu)(n_2^+)^2 + \mu(n_3^+)^2}{M_2^+} \right) \frac{dz_3}{dt} \\
& \quad \left. - \left(\frac{\mu(n_3^+)^2}{M_2^+} \right) \frac{dz_4}{dt} \right] \\
& = \frac{-(1+\mu)n_1^+ + \mu n_2^+}{M_1^+} + \frac{-\mu n_1^+ + (1+2\mu)n_2^+ - \mu n_3^+}{M_2^+} \tag{4.23}
\end{aligned}$$

$k = 3(1)P - 2 :$

$$\begin{aligned}
& \tau \left[- \left(\frac{\mu(n_{k-2}^+)^2}{M_{k-1}^+} \right) \frac{dz_{k-2}}{dt} \right. \\
& \quad + \left(\frac{\mu(n_{k-2}^+)^2 + (1+2\mu)(n_{k-1}^+)^2}{M_{k-1}^+} + \frac{\mu(n_{k-1}^+)^2}{M_k^+} \right) \frac{dz_{k-1}}{dt} \\
& \quad - \left(\frac{(1+2\mu)(n_{k-1}^+)^2 + \mu(n_k^+)^2}{M_{k-1}^+} + \frac{\mu(n_{k-1}^+)^2 + (1+2\mu)(n_k^+)^2}{M_k^+} \right) \frac{dz_k}{dt} \\
& \quad + \left(\frac{\mu(n_k^+)^2}{M_{k-1}^+} + \frac{(1+2\mu)(n_k^+)^2 + \mu(n_{k+1}^+)^2}{M_k^+} \right) \frac{dz_{k+1}}{dt} \\
& \quad \left. - \left(\frac{\mu(n_{k+1}^+)^2}{M_k^+} \right) \frac{dz_{k+2}}{dt} \right] \\
& = \frac{\mu n_{k-2}^+ - (1+2\mu)n_{k-1}^+ + \mu n_k^+}{M_{k-1}^+} + \frac{-\mu n_{k-1}^+ + (1+2\mu)n_k^+ - \mu n_{k+1}^+}{M_k^+} \tag{4.24}
\end{aligned}$$

$k = P - 1$:

$$\begin{aligned}
& \tau \left[- \left(\frac{\mu(n_{P-3}^+)^2}{M_{P-2}^+} \right) \frac{dz_{P-3}}{dt} \right. \\
& \quad + \left(\frac{\mu(n_{P-3}^+)^2 + (1 + 2\mu)(n_{P-2}^+)^2}{M_{P-2}^+} + \frac{\mu(n_{P-2}^+)^2}{M_{P-1}^+} \right) \frac{dz_{P-2}}{dt} \\
& \quad - \left(\frac{(1 + 2\mu)(n_{P-2}^+)^2 + \mu(n_{P-1}^+)^2}{M_{P-2}^+} + \frac{\mu(n_{P-2}^+)^2 + (1 + \mu)(n_{P-1}^+)^2}{M_{P-1}^+} \right) \frac{dz_{P-1}}{dt} \\
& \quad \left. + \left(\frac{\mu(n_{P-1}^+)^2}{M_{P-2}^+} + \frac{(1 + \mu)(n_{P-1}^+)^2}{M_{P-1}^+} \right) \frac{dz_P}{dt} \right] \\
& = \frac{\mu n_{P-3}^+ - (1 + 2\mu)n_{P-2}^+ + \mu n_{P-1}^+}{M_{P-2}^+} + \frac{-\mu n_{P-2}^+ + (1 + \mu)n_{P-1}^+}{M_{P-1}^+}. \tag{4.25}
\end{aligned}$$

Setzt man in diesen Beziehungen $\tau = 0$, erhält man wieder die Gleichungen (4.18) bis (4.20).

4.3 Fehler und Stabilität der numerischen Lösung

Die in der Literatur meist angewandte Diskretisierung der ersten Ortsableitung im Transformationsterm $\partial x / \partial z|_{z_k(t)} \cdot dz_k / dt$ in Gleichung (4.7) ist die Zentralfdifferenz

$$\left. \frac{\partial x}{\partial z} \right|_{z_k(t)} \approx \frac{x_{k+1} - x_{k-1}}{z_{k+1} - z_{k-1}}. \tag{4.26}$$

Sie stellt üblicherweise die Approximation mit dem kleinsten Fehler dar, da sie die wahre Tangente an das Ortsprofil besser annähert als die Rückwärtsdifferenz

$$\left. \frac{\partial x}{\partial z} \right|_{z_k(t)} \approx \frac{x_k - x_{k-1}}{z_k - z_{k-1}} \tag{4.27}$$

oder die Vorwärtsdifferenz

$$\left. \frac{\partial x}{\partial z} \right|_{z_k(t)} \approx \frac{x_{k+1} - x_k}{z_{k+1} - z_k}, \tag{4.28}$$

siehe Abbildung 4.1.

Treten allerdings dominante Konvektionsterme in den Modellgleichungen auf, können sich durch Anwendung der Zentralfdifferenz numerisch bedingte Oszillationen im Lösungsverlauf ergeben [14]. In diesem Fall muss auf eine Upwind-Diskretisierung zurückgegriffen werden, womit die DA-Gleichungen aus diskretisierter PDgl., Randbedingungen und Gittersteuerung örtlich stabilisiert werden.

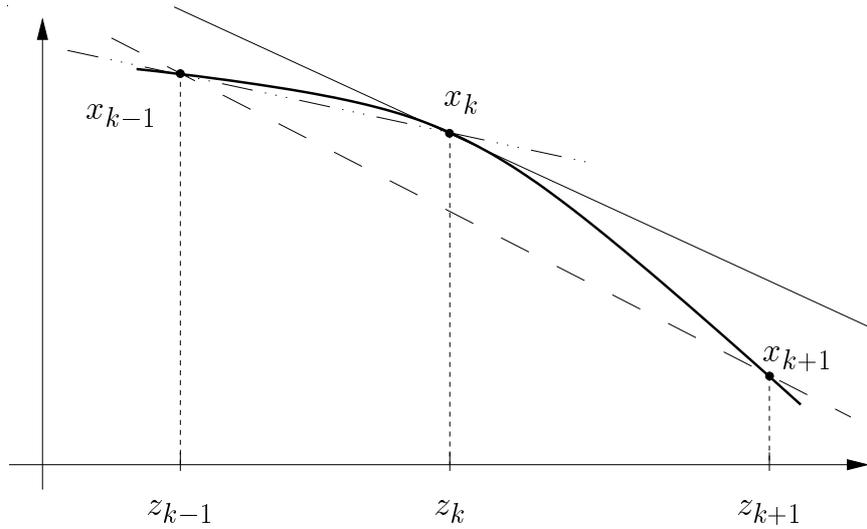


Abb. 4.1: Approximation der ersten Ortsableitung anhand des Ortsprofils. Die Zentraldifferenz (4.26) (- -) nähert die Steigung der wahren Tangente $\partial x/\partial z$ (—) im Allgemeinen besser an als die Rückwärtsdifferenz (4.27) (- · · ·).

Sollten trotz aller sinnvollen Diskretisierungs- und Glättungsmaßnahmen Instabilitäten auftreten, muss die Zahl der Gitterpunkte erhöht und gleichzeitig eine Anpassung der Verfahrensparameter κ und τ vorgenommen werden. Jedoch kann die für Stabilität benötigte Anzahl der Gitterpunkte viel höher sein als die Anzahl der Gitterpunkte, die für genügend genaue örtliche Auflösung und damit zur Erfüllung der eigentlichen Funktion des Verfahrens notwendig wären. Damit wird die Rechenzeit zur Integration der DA-Gleichungen steigen.

Eine Betrachtung der Stabilität der DA-Gleichungen inklusive PDgl. und Gittergleichungen wird beispielsweise in [15] vorgenommen. Die Ergebnisse sind allerdings problemabhängig und können schwerlich innerhalb einer automatisierten Verarbeitung eingesetzt werden. Die DA-Gleichungen nach der Diskretisierung sind nach [15] in der Form

$$\begin{aligned} \frac{dx_k}{dt} &= \left. \frac{\partial x}{\partial z} \right|_{z_k(t)} \cdot \frac{dz_k}{dt} + \mu \left. \frac{\partial^2 x}{\partial z^2} \right|_{z_k(t)} + s(x, z)|_{z_k(t)} \quad k = 2(1)P \\ g(x_1, \dots, x_P, z_1, \dots, z_P) &= 0 \end{aligned} \quad (4.29)$$

gegeben, wobei $s(x, z)|_{z_k(t)}$ den Quellterm der PDgl. und $g(\dots) = 0$ die Gittergleichungen mit Randbedingungen darstellen. Damit können Reaktions-Diffusions-Systeme beschrieben werden.

Der Diskretisierungsfehler ergibt sich aus den Termen höherer Ordnung der Gleichun-

gen (3.6) und (3.8), die ausführlich geschrieben lauten ($\Delta z_k = z_{k+1} - z_k$):

$$\begin{aligned} \left. \frac{\partial x}{\partial z} \right|_{z_k(t)} &= \frac{x_{k+1} - x_{k-1}}{z_{k+1} - z_{k-1}} - \frac{1}{2} \left. \frac{\partial^2 x}{\partial z^2} \right|_{z_k(t)} (\Delta z_k - \Delta z_{k-1}) \\ &\quad - \frac{1}{6} \left. \frac{\partial^3 x}{\partial z^3} \right|_{z_k(t)} \frac{\Delta z_k^3 + \Delta z_{k-1}^3}{\Delta z_k + \Delta z_{k-1}} + \dots \\ \left. \frac{\partial^2 x}{\partial z^2} \right|_{z_k(t)} &= \frac{2}{\Delta z_k + \Delta z_{k-1}} \left(\frac{x_{k+1} - x_k}{\Delta z_k} - \frac{x_k - x_{k-1}}{\Delta z_{k-1}} \right) \\ &\quad - \frac{1}{3} \left. \frac{\partial^3 x}{\partial z^3} \right|_{z_k(t)} (\Delta z_k - \Delta z_{k-1}) - \frac{1}{12} \left. \frac{\partial^4 x}{\partial z^4} \right|_{z_k(t)} \frac{\Delta z_k^3 + \Delta z_{k-1}^3}{\Delta z_k + \Delta z_{k-1}} + \dots \end{aligned}$$

Der Diskretisierungsfehler für Gleichung (4.29) hat also die Form

$$\begin{aligned} v \approx & -\frac{dz_k}{dt} \left[\left. \frac{1}{2} \frac{\partial^2 x}{\partial z^2} \right|_{z_k(t)} (\Delta z_k - \Delta z_{k-1}) + \left. \frac{1}{6} \frac{\partial^3 x}{\partial z^3} \right|_{z_k(t)} \frac{\Delta z_k^3 + \Delta z_{k-1}^3}{\Delta z_k + \Delta z_{k-1}} + \dots \right] \\ & - \mu \left[\left. \frac{1}{3} \frac{\partial^3 x}{\partial z^3} \right|_{z_k(t)} (\Delta z_k - \Delta z_{k-1}) + \left. \frac{1}{12} \frac{\partial^4 x}{\partial z^4} \right|_{z_k(t)} \frac{\Delta z_k^3 + \Delta z_{k-1}^3}{\Delta z_k + \Delta z_{k-1}} + \dots \right] \quad (4.30) \end{aligned}$$

Die Terme $dz_k/dt[\dots]$ in der ersten Zeile des Ausdrucks kommen erst aufgrund der Einführung des bewegten Gitters ins Spiel und bedingen weitere Diskretisierungsfehler, die bei Benutzung eines festen Gitters nicht auftreten. Diese zusätzlichen Fehler überwiegen die Ungenauigkeiten aufgrund des Diffusionsterms $\mu \cdot \partial^2 x / \partial z^2$, wenn die Frontgeschwindigkeiten hoch sind (dz_k/dt groß) oder der Diffusionskoeffizient μ sehr klein ist. Dieser Einfluss kann verringert werden, wenn die Gittertrajektorien einen glatten und gleichmäßigen Verlauf haben und im Kleinen gesehen keine großen Änderungen vollführen. Dies wird vor allem durch eine zeitliche Gitterglättung mit $\tau > 0$ erreicht.

Eine weiteres wichtiges Ergebnis betrifft die numerische Stabilität der Lösung. In [15] wird festgestellt, dass der Fehler sehr schnell anwächst, wenn $\partial s / \partial x$ große positive Werte annimmt. Dies entspricht exothermen Reaktionen. In diesen Fällen wird eine Verwendung von Moving-Grid-Methoden für diese Reaktions-Diffusions-Systeme nicht empfohlen.

4.4 Zusammenfassung

Insgesamt benötigt man zur numerischen Lösung von (4.1) bis (4.3) folgende Rechen- bzw. Simulationsgleichungen:

- Die auf das mitbewegte Gitter transformierten und diskretisierten Dgln. (4.7) jeder Pdgl. mit Randbedingungen (4.8) und Anfangsbedingungen

$$\frac{dx_k}{dt} - \left. \frac{\partial x}{\partial z} \right|_{z_k(t)} \cdot \frac{dz_k}{dt} = \tilde{f}(\dots) \quad k = 2(1)P - 1, \quad t > t_0$$

$$\tilde{g}_0(\dots) = 0, \quad \tilde{g}_L(\dots) = 0 \quad t > t_0$$

$$x_k(t_0) = x_0(z_k) \quad k = 1(1)P.$$

- Die Gittergleichungen (4.18) bis (4.20) nur mit örtlicher Gitterglättung bzw. (4.23) bis (4.25) mit örtlicher und zeitlicher Gitterglättung. Dazu die Randwerte des Gitters

$$z_1 = 0, \quad z_P = L \quad t > 0$$

und eine Anfangsverteilung der Gitterpunkte

$$z_k(t_0) = z_{k0} \quad k = 1(1)P \quad \text{mit} \quad z_{10} = 0, \quad z_{P0} = L.$$

- Die Monitorfunktion (4.13), wobei $N \hat{=}$ Anzahl PDgln., L bzw. $X_j \hat{=}$ Normierungsfaktoren,

$$M_k^+(t) := \sqrt{1 + \frac{1}{N} \sum_{j=1}^N \left(\frac{L}{X_j} \frac{x_{k+1,j} - x_{k,j}}{z_{k+1} - z_k} \right)^2} \quad k = 1(1)P - 1.$$

- Die Hilfsvariablen (Zuweisungen)

$$n_k^+(t) := \frac{1}{z_{k+1}(t) - z_k(t)} \quad k = 1(1)P - 1$$

und die Abkürzung

$$\mu := \kappa(\kappa + 1).$$

- Die Verfahrensparameter für örtliche Gitterglättung κ , für zeitliche Gitterglättung τ und die Anzahl der Gitterpunkte P .

Die Anzahl der Gleichungen ist $(N + 1) \cdot P$ und damit identisch mit der Zahl der Unbekannten $x_k(t)$ und $z_k(t)$, $k = 1(1)P$. Im Vergleich mit der Anzahl der Gleichungen $N \cdot P$ bei nichtbewegtem Gitter wird deutlich, dass die zusätzliche Rechenzeit bedingt durch die Vergrößerung der Gleichungsanzahl mit wachsendem N an Bedeutung verliert.

Bei Anwendung von Dirichlet-Randbedingungen (Wert des Zustands wird vorgegeben) liegt für den Fall $\tau > 0$ bei angenommener Nicht-Singularität der Matrix aus den linken Gleichungsseiten ein echtes ODE-Modell vor, im Fall $\tau = 0$ ein DA-Modell mit Index eins [24].

Kapitel 5

Implementierung

Zur Einbindung des in Kapitel 4 dargestellten Moving-Grid-Verfahrens in das Vorverarbeitungswerkzeug SYPPROT muss zum einen die Eingabesprache der MATHEMATICA-Daten-Struktur (MDS) angepasst und zum anderen der Programm-Code von SYPPROT erweitert werden. Die Erweiterungen der MDS betreffen die MOL-Parameter zur Definition bewegter Gitter. Anhand dieser Parameter aus der MDS-Datei muss das Diskretisierungs-Modul des Programm-Codes eine automatische Gleichungstransformation und Generierung der Moving-Grid-Gleichungen, abgestimmt auf die bereits vorhandenen Diskretisierungsschemata der FD und der FV, vornehmen.

5.1 Die MATHEMATICA-Daten-Struktur

Die Definition verteilter Modelle erfolgt mittels der MATHEMATICA-Daten-Struktur (MDS). Sie bedient sich einer Anzahl von symbolischen Programmiermethoden zur Spezifizierung von Informationen:

- Ausdruck: `kopf[argument1, ..., argumentn]`
Der *kopf* des Ausdrucks dient zur Charakterisierung der repräsentierten Informationen, die *argumente* stellen die Informationen dar.
- Gleichung: `lhs == rhs`
- Zuweisungen: `lhs := rhs`
- Regeln: `lhs -> rhs`
Regeln stellen ebenfalls eine Zuweisung von `rhs` an `lhs` dar. Im Gegensatz zur eigentlichen Zuweisung `lhs := rhs` erlauben mehrere Regeln innerhalb eines Ausdrucks eine flexible Spezifikation von Informationen in weitgehend beliebiger Reihenfolge.

- Listen: $\{ element1, \dots, elementn \}$
Listen dienen der geordneten Repräsentation von Einzelinformationen.

Die MDS wird anhand der folgenden Definitionsfunktionen in einzelne Abschnitte unterteilt. Dazu gehören zum einen Ausdrücke zur Definition des Modelles aus PDgln., Integro-PDgln., Dgln. und algebraischen Gleichungen:

- `GeneralDescription[...]` beinhaltet allgemeine Angaben wie Dateiname, Autor, Datum usw.;
- `IndependentVariables[...]` definiert Zeit-, Orts- und Eigenschaftsvariablen;
- `StructureParameters[...]` definiert strukturelle Modellparameter wie Komponenten- und Stützstellenzahl;
- `SystemParameters[...]` definiert Modellparameter wie Masse, Diffusionskoeffizient usw.;
- `InputVariables[...]` benennt die Eingangsgrößen;
- `OutputEquations[...]` definiert die Ausgangsgrößen;
- `StateVariables[...]` definiert die Zustandsgrößen;
- `AuxiliaryVariables[...]` definiert Hilfsvariablen (Abkürzungen);
- `SystemEquations[...]` definiert die Modellgleichungen;
- `ProcessValues[...]` definiert Ausgabegrößen für Offline-Grafik.

Die Gitterdefinitionen und MOL-Parameter werden durch zwei weitere Ausdrücke festgelegt:

- `Domains[...]` zur Definition der Ortsgitter;
- `Discretizations[...]` zur Definition der MOL-Diskretisierungsparameter.

Für die genaue Formulierung eines Modells und der einzelnen Ausdrücke sei auf [11] oder auf die MDS-Datei im Anhang verwiesen.

Gitterdefinition und MOL-Parameter

Da im Folgenden auf die Definition von Gittern und MOL-Parametern und ihre Erweiterung eingegangen wird, werden die beiden Definitionsfunktionen `Domains` und `Discretizations` näher vorgestellt.

Eine Gitterdefinition in der MDS zeigt das folgende Beispiel:

```
Domains[
  Grid[ Grid[z],
        Computation -> Function[{k}, (k-1)*L/(kmax-1)],
        Granularity  -> "kmax",
        GridPractice -> "GridPoints" ] ];
```

Der Ausdruck `Domains` besitzt hier als einziges Argument wiederum einen Ausdruck, nämlich `Grid`. Die Argumente von `Grid` definieren das Gitter mittels eines Namens `Grid[z]`, worin auch die unabhängige Ortsvariable z genannt wird, und drei Regeln. Diese Regeln geben (in beliebiger Reihenfolge) die Berechnungsvorschrift für die Lage der Gitterpunkte (Regel `Computation`), die Anzahl der verwendeten Gitterpunkte (`Granularity`) sowie das Gitterverfahren (`GridPractice`) für die mögliche Benutzung der FV an.

Die Definition der MOL-Diskretisierungs-Parameter erfolgt beispielsweise mit

```
Discretizations[
  FDMMethod[ FDUpwind,
             PolynomPoints -> {1,2,3},
             Eccentricity  -> {0,1,0},
             OrderReduction -> {0,0,1}],
  FVMMethod[ FVUpwind,
             Profile -> {"upwind", "piecewise-linear"} ]];
```

Innerhalb der Ausdrucks `Discretizations` definieren die Argumente `FDMMethod` bzw. `FVMMethod` die Diskretisierungsparameter für Finite Differenzen bzw. Finite Volumen. In `FDMMethod` folgen nach einem Namen (hier `FDUpwind`) drei Regeln. `PolynomPoints` gibt die Anzahl der einbezogenen Stützpunkte der örtlichen Ableitungen mit Ordnungen 0, 1, 2 für die Lagrange-Polynome an. `Eccentricity` legt die Verschiebung des mittleren Stützpunktes gegenüber dem Approximationszentrum fest. `OrderReduction` schließlich definiert die vorzunehmenden Ordnungsreduktionen an den Rändern des Ortsbereiches. In `FVMMethod` folgt nach dem Namen (hier `FVUpwind`) nur die Regel `Profile` zur Definition der zu verwendenden Profilannahmen für die örtlichen Ableitungen der Ordnungen 0 und 1.

Vorgaben zur Erweiterung der MDS

Die Ergänzung der MATHEMATICA-Daten-Struktur sollte unter folgenden Gesichtspunkten geschehen:

- Der Benutzer soll das Moving-Grid-Verfahren einfach anwenden können, d.h. im Idealfall ohne tiefgehende Kenntnisse über das Verfahren und seine Parameter.
- Es sollen nur wenige Ergänzungen, aber keine Umstrukturierungen an der MDS vorgenommen werden.

5.1.1 Erweiterung der MATHEMATICA-Daten-Struktur

Die MATHEMATICA-Daten-Struktur musste zur Ankopplung des Moving-Grid-Verfahrens nur an einer Stelle geändert werden. Die Wahl eines bewegten Gitters erfolgt im Argument `Grid` der `Domain`-Definition als eigene Regel. Damit ist es jetzt möglich, jedes Ortsgitter einzeln auf ein bewegtes Gitter umzustellen, und gegebenenfalls die Verfahrensparameter zu definieren.

Will man eine Umstellung auf ein bewegtes Gitter vornehmen, ist es nur notwendig, eine neue Regel der Form

```
MovingGrid -> { SpaceSmoothing -> Real-Zahl,  
                 TimeSmoothing -> Real-Zahl,  
                 Monitor -> Real-Zahl,  
                 MonitorStates -> Liste,  
                 CentralDiff -> True | False  
               }
```

an die bestehenden Regeln innerhalb der Funktion `Grid` anzufügen. Dabei können die Regeln `SpaceSmoothing`, `TimeSmoothing` usw. in beliebiger Reihenfolge auftreten, oder es können beliebige oder alle Regeln weggelassen werden. Im Einzelnen werden folgende Funktionen unterstützt:

- `SpaceSmoothing` definiert den Parameter κ der örtlichen Gitterglättung.
- `TimeSmoothing` definiert den Parameter τ der zeitlichen Gitterglättung.
- Mit `Monitor` kann die Monitorfunktion über eine zugeordnete Nummer ausgewählt werden. Bisher kann auf die Implementierung des Bogenlängenmonitors entsprechend Gleichung (4.13) zurückgegriffen werden. Dem Bogenlängenmonitor entspricht die Nummer 1.
- Die Liste in der Regel `MonitorStates` gibt die vom Monitor zu überwachenden verteilten Zustände an.

- Die Regel `CentralDiff` erlaubt eine Diskretisierung des Transformationsterms als Zentralkdifferenz nach Gleichung (3.6), da dies für nicht äquidistante Gitter mittels der Lagrange-Polynome nicht möglich ist.

Hierzu eine *Warnung*: Da in der aktuellen Version von SYPPROT keine Diskretisierung von einzelnen Termen, sondern nur von kompletten Gleichungen vorgesehen ist, kann ohne eine tiefgreifende Änderung am Vorverarbeitungswerkzeug keine gesonderte Diskretisierung des Transformationsterms vorgenommen werden. Deshalb wurde dies durch die Regel `CentralDiff` für diesen Term provisorisch ermöglicht. Allerdings wird die automatische Indizierung der diskretisierten PDgln. umgangen, so dass mehrdimensionale PDgln. in Zukunft Probleme verursachen können. Außerdem wird die Zentralkdifferenz eingesetzt, ohne auf Probleme mit Indexlaufbereichen zu achten. Bei unsachgemäßer Anwendung kann also eine Überschreitung der maximalen Indizes auftreten. Diese Nachteile ergeben sich, da die Transformation auf das bewegte Gitter vor den Diskretisierungen stattfindet. Grundsätzlich soll diese Regel nur angewandt werden, wenn zweite Ortsableitungen in der betreffenden PDgl. vorkommen und der Anwender sich über die Struktur der diskretisierten Gleichungen im Klaren ist.

Einzelne Regeln können weggelassen werden, wenn keine Abweichung von den Standardeinstellungen beabsichtigt ist. Im einfachsten Fall bleibt die Liste auf der rechten Seite leer:

```
MovingGrid -> {} .
```

Damit werden folgende Standardeinstellungen der Verfahrensparameter vorgenommen:

- Parameter der Gitterglättung $\kappa = 2$;
- Parameter der Zeitglättung $\tau = 0$;
- Wahl des Bogenlängenmonitors (Nr. 1);
- Überwachung aller verteilten Zustandsgrößen aus einem zusammengehörenden Gleichungsabschnitt;
- Automatische Diskretisierung des Transformationsterms anhand einer benutzerdefinierten FD oder FV. Keine Anwendung der Zentralkdifferenz nach Gleichung (3.6).

Zusätzlich muss vom Benutzer eine Anfangsverteilung der Gitterpunkte vorgegeben werden. Dies geschieht mit einer Berechnungsvorschrift über die schon vorhandene Regel `Computation` der Funktion `Grid`. Die Verteilung der Stützstellen ist prinzipiell beliebig. Es muss nur darauf geachtet werden, dass für den Index 1 der z -Wert des

```

Domains[
  Grid[ Gitter1[z],
        Computation -> Function[{k},(k-1)*L/(kmax-1)],
        Granularity -> "kmax",
        MovingGrid -> {}
      ],
  Grid[ Gitter2[z],
        Computation -> Function[{k},(k-1)*L/(kmax-1)],
        Granularity -> "kmax",
        MovingGrid -> { SpaceSmoothing -> 3 }
      ],
  Grid[ Gitter3[z],
        Computation -> Function[{k},(k-1)*L/(kmax-1)],
        Granularity -> "kmax",
        MovingGrid -> { SpaceSmoothing -> 0.5,
                        TimeSmoothing -> 0.001,
                        Monitor -> 1,
                        MonitorStates -> {x[z,t],y[z,t]},
                        CentralDiff -> True }
      ]
];

```

Abb. 5.1: Beispiele für Moving-Grid-Definitionen

linken Randes und für den Index aus der *Granularity*-Regel der z -Wert des rechten Randes erzeugt wird.

Um die Normierung anzuwenden, muss der Benutzer eine Definition der typischen oder maximalen Wertebereiche X_j für Zustandsgrößen vornehmen. Dies geschieht über die Regeln *MaxValue* -> *MaxWert* und *MinValue* -> *MinWert* bei der Definition der Zustandsvariablen im Ausdruck *StateVariables*. Der Normierungsfaktor berechnet sich daraus mit $X_j = \text{MaxWert} - \text{MinWert}$. Eine Normierung der Länge des Ortsbereichs erfolgt automatisch.

Einige Beispielformen sind in Abbildung 5.1 gezeigt. Die erste Definition erzeugt ein bewegtes Gitter mit Namen *Gitter1[z]* und den Standard-Verfahrensparametern. Das zweite Beispiel erzeugt ein bewegtes Gitter mit Namen *Gitter2[z]* und den Standardeinstellungen, aber $\kappa = 3$. Die letzte Definition erzeugt ein bewegtes Gitter mit Namen *Gitter3[z]*, $\kappa = 0.5$, $\tau = 0.001$, Bogenlängenmonitor und Zentrodifferenz für den Transformationsterm. Zur Berechnung der Monitorwerte werden dabei nur die beiden Zustände $x(z, t)$ und $y(z, t)$ herangezogen, obwohl gegebenenfalls noch mehr Zustände des verkoppelten Modells überwacht werden könnten. In allen drei Beispielen wird ein äquidistantes Anfangsgitter erzeugt. Der Wert des linken Randes ist dabei 0 (Index $k = 1$), der Wert des rechten Randes ist L (Index $k = kmax$).

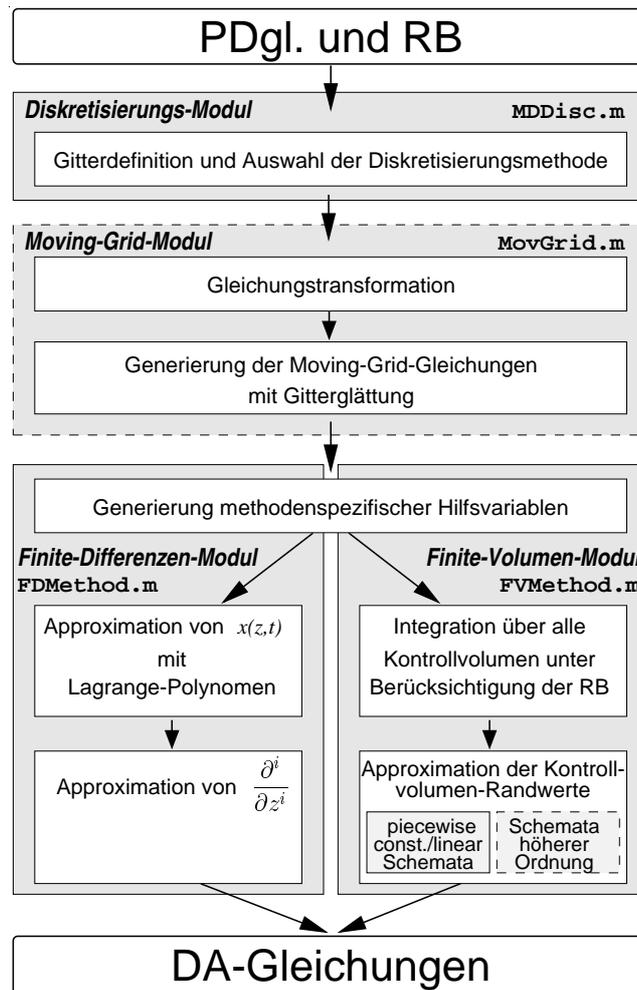


Abb. 5.2: Ablaufdiagramm für die Vorverarbeitung einer PDgl. mit SYPPROT.

5.2 Erweiterung der Implementierung

Um die Ansatzpunkte der Erweiterung des Programm-Codes ersehen zu können, wird hier kurz auf die vorhandene Struktur der Diskretisierungs-Module des Vorverarbeitungswerkzeugs SYPPROT eingegangen. Ein Ablaufdiagramm für die Diskretisierung einer PDgl. mit Darstellung der einzelnen Module zeigt Abbildung 5.2.

Zunächst sammelt das übergeordnete Diskretisierungs-Modul `MDDisc.m` die zur korrekten Verarbeitung benötigten Informationen aus der MDS-Eingabedatei. Das Moving-Grid-Modul wird vorerst nicht betrachtet. Auf die PDgl. wird dann die gewählte Diskretisierungsmethode angewandt. Das FD-Modul `FDMMethod.m` berechnet mittels Lagrange-Polynomen die Approximationen für die Zustandsvariablen und deren Ortsableitungen, wohingegen das FV-Modul `FVMMethod.m` die PDgl. über jedes Kontrollvolumen integriert und die unbekanntenen Werte an den Rändern der Kontrollvolumina entsprechend der getroffenen Profilannahmen approximiert.

Für das neue Modul sind zwei Varianten denkbar:

- Als erste Möglichkeit kommt eine Anwendung des Moduls *nach* den Diskretisierungsmethoden `FDMethod.m` und `FVMethod.m` in Frage. Das Modul müsste dann die schon diskretisierten PDgln. transformieren, was eine Anpassung der Diskretisierung des Transformationsterms für bewegtes Gitter an die Diskretisierung der PDgl. erfordert. Die implementierten FD und FV könnten also nicht ausgenutzt werden oder müssten ein zweites Mal aufgerufen werden.
- Tritt das neue Modul für die Bearbeitung bewegter Gitterpunkte *vor* den Diskretisierungsmethoden in Aktion, wie in Abbildung 5.2 dargestellt, kann die automatische Diskretisierung der Transformationsterme und Monitorfunktionen gemeinsam mit den ursprünglichen Modellgleichungen in einem Aufruf erfolgen. Ein weiterer Vorteil ergibt sich bei zukünftigen Erweiterungen von SYPPROT um neue Diskretisierungs-Methoden: diese Methoden können schon bei ihrer Implementierung auf die Anforderungen des Moving-Grid-Verfahrens ausgerichtet werden.

Deshalb wurde auf eine Anordnung des neuen Moduls `MovGrid.m` wie in Abbildung 5.2 entschieden.

Das Modul `MovGrid.m` muss die Transformation der Zeitableitungen in PDgln. vornehmen und zusätzliche Gleichungen mit Gitterglättung für das bewegte Gitter hinzufügen. Darüber hinaus ist es erforderlich, am Ablauf der vorhandenen Diskretisierungsmethoden einige Modifikationen vorzunehmen.

In den weiteren Unterkapiteln soll in aller Kürze auf die Grundzüge der Implementierung und auf einige Details eingegangen werden. Dies kann bei zukünftigen Erweiterungen von Nutzen sein.

5.2.1 Das Modul `MovGrid.m`

Das neue Modul `MovGrid.m` der Vorverarbeitung führt alle Aktionen des hier behandelten Moving-Grid-Verfahrens aus, welche unabhängig von den Diskretisierungsmethoden vorgenommen werden können. Im Hauptprogramm der MOL-Diskretisierung, `MDDisc.m`, werden diese Aktionen mit dem Aufruf `MovingGridProcedure[]` gestartet, bevor alle sonstigen Vorverarbeitungsschritte getätigt werden. Ein Struktogramm der wichtigsten Funktionalitäten der `MovingGridProcedure` zeigt Abbildung 5.3.

Die Prozedur `MovingGridProcedure` erzeugt zuerst eine Anzahl von Hilfslisten, um oft benötigte Informationen auf einfache Weise abfragen zu können. Es sei vor allem auf die Liste `MGGridParameters` hingewiesen, welche die Parameter der verschiedenen verwendeten Gitter enthält. Diese sind in Unterlisten in einer bestimmten Reihenfolge angeordnet, um einen einfachen und schnellen Zugriff zu gewährleisten:

{ *Name, Computation, Granularity, GridPractice, SpaceSmoothing, TimeSmoothing, Monitor, MonitorStates, CentralDiff* } .

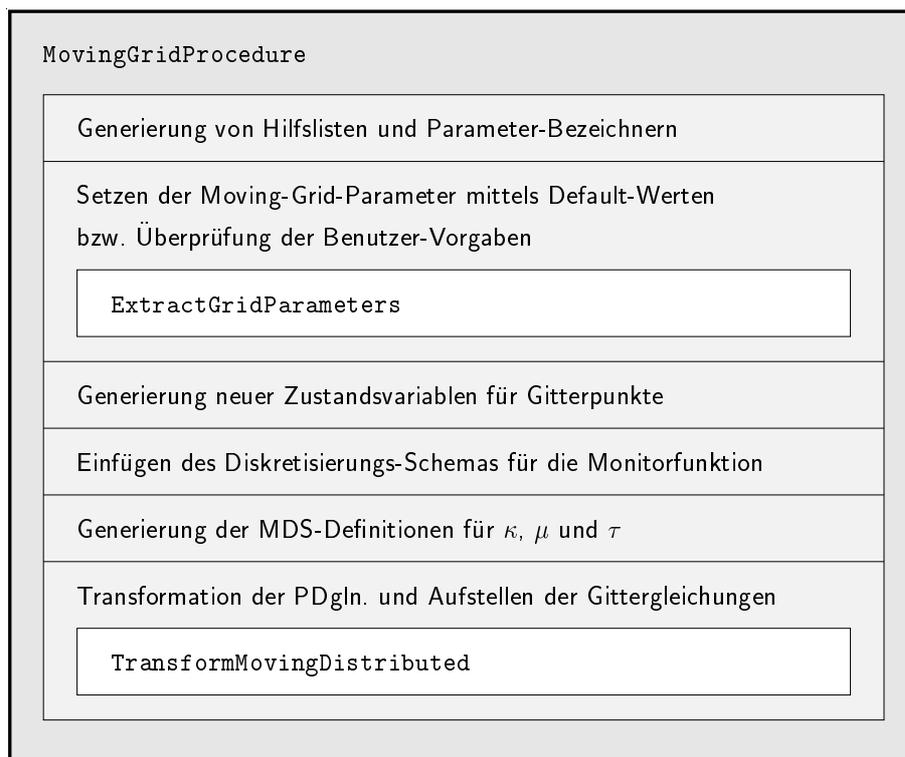


Abb. 5.3: Struktogramm der Hauptfunktion `MovingGridProcedure`

Bei der Erzeugung dieser Unterlisten werden gegebenenfalls die Default-Werte des Moving-Grid-Verfahrens eingesetzt und eventuell auftretende Fehleingaben abgefangen. Diese Default-Werte sind als interne Variablen mit Namen `MGDefaultKappa`, `MGDefaultTau`, `MGDefaultMonitor`, `MGDefaultMonitorStates` und `MGDefaultCentralDiff` definiert.

Im Unterschied zu nichtbewegten Gittern müssen hier natürlich zusätzliche Zustandsvariablen für die zeitlich veränderlichen Orte der Gitterpunkte eingeführt werden. Diese werden in bereits diskretisierter Form der `StateVariablesList` zugefügt, da eine Anwendung verschiedener Diskretisierungsmethoden für Zustandsgrößen nicht sinnvoll ist.

Da die Berechnungsvorschrift der Monitorfunktion später in nicht-diskretisierter Form in das Modell geschrieben wird, muss hierfür eine bestimmte Diskretisierung verwendet werden, welche die richtigen Indizes erzeugt. Deshalb wird an die Liste der Diskretisierungsschemata ein weiteres Schema zur Verarbeitung des Monitors angehängt:

```

FDMMethod[ MGMonitorScheme,
            PolynomPoints -> {1,2,3},
            Eccentricity  -> {0,0,0},
            OrderReduction-> {0,0,1}
];
  
```

Diese Diskretisierung mit Finiten Differenzen erzeugt aus einer kontinuierlichen Monitor-Funktion genau die diskretisierten Monitor-Funktionen (4.13). Diese Form ist sowohl für PDgln. geeignet, welche mit der FD diskretisiert werden, als auch für solche, die mit der FV vorverarbeitet werden.

Danach werden die MDS-Definitionen der Verfahrensparameter der `SystemParametersList` zugeschlagen. Sie lauten zum Beispiel `ppMGkappa1` für κ , `ppMGmu1` für $\mu := \kappa(\kappa + 1)$ und `ppMGtau1` für τ . Die Namen setzen sich zusammen aus `pp` für „Pre-Processing“, `MG` für „Moving Grid“, einem Bezeichner wie `kappa` und einer Zahl zur Unterscheidung der verschiedenen Parameter zugehörig zu verschiedenen bewegten Gittern.

Schließlich werden diejenigen Modellgleichungen, welche auf bewegten Ortsgittern definiert sind, mittels der Prozedur `TransformMovingDistributed` einer gegebenenfalls erforderlichen Transformation unterzogen. Außerdem werden die Gleichungen zur Stützstellensteuerung erzeugt.

Transformation und Erzeugung der Gittergleichungen

Die Gleichungs-Transformation auf das bewegte Gitter und die Generierung der Gittergleichungen und Monitorfunktionen wird in der Prozedur `TransformMovingDistributed` vorgenommen. Abbildung 5.4 zeigt die wichtigsten Schritte dieser Prozedur.

Wenn ein Gleichungsabschnitt mit verteilten Gleichungen vorliegt, werden im Einzelnen die folgenden Aktionen ausgeführt (Zahlen in Klammern beziehen sich auf Abbildung 5.4):

- (1) Über den aktuellen Gleichungsabschnitt werden Informationen wie Gleichungen, Gitterparameter und Ortsvariable beschafft.
- (2) Um partielle Ableitungen von Zustandsgrößen entsprechend Gleichung (4.6) transformieren zu können, werden auf beliebige Zustandsgrößen zugeschnittene Regeln gebildet.
- (3) Diese Regeln wirken sich wie folgt aus. Auf dem Gitter mit Namen `Grid[z]` wird beispielsweise der Differentialoperator

$$D[x[z,t], t] \quad \text{ersetzt durch} \quad (5.1)$$

$$D[x[z,t], t] - D[x[z,t], z] * D[zGrid[z,t], t], \quad (5.2)$$

woraus die später angewandte Diskretisierung die korrekten Gleichungen gewinnt. Dies ist mathematisch gesehen nicht ganz richtig, da erstens die Ortsvariable `zGrid` nicht von z und t , sondern nur von t abhängt, und zweitens im Ausdruck (5.2) nur partielle Ableitungen und nicht wie hergeleitet partielle und totale Ableitungen vorkommen. Durch die Angabe der Abhängigkeit von z wird

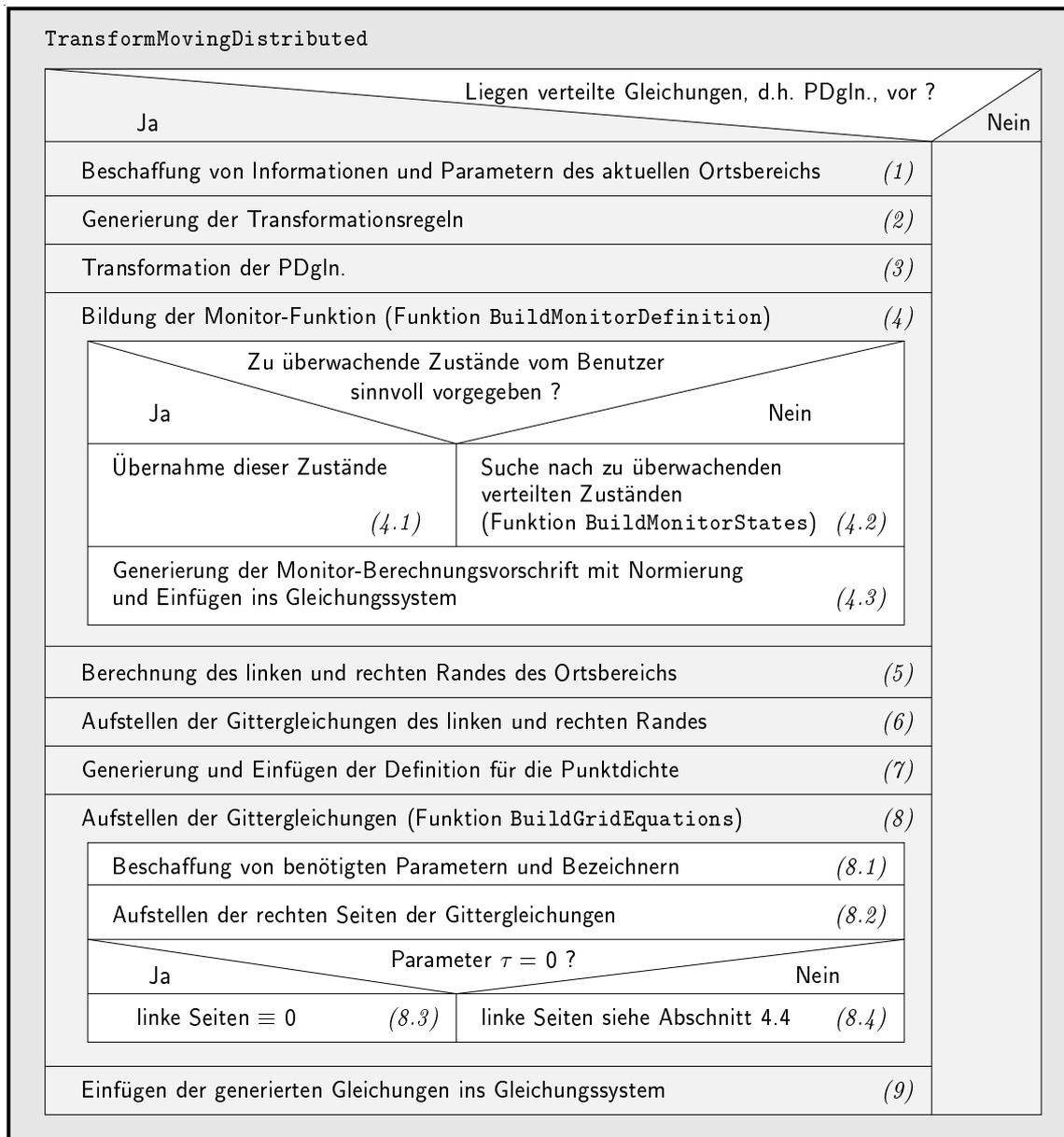


Abb. 5.4: Struktogramm zur Gleichungs-Transformation, Monitor- und Gittergleichungs-Generierung mit Hilfe der Funktionen TransformMovingDistributed, BuildMonitorDefinition und BuildGridEquations.

jedoch die Generierung eines Vektors diskreter Ortsstützstellen veranlasst, so wie jede kontinuierlich verteilte Zustandsgröße $x(z, t)$ in einen Vektor $x_k(t)$ an diskreten Ortsstützstellen transformiert wird. Im Falle der Zentraldifferenz mit `Centraldiff->True` wird der obige Term (5.1) ersetzt durch

$$D[x[z,t],t] - (x[ppI+1][t]-x[ppI-1][t]) / (zGrid[ppI+1][t]-zGrid[ppI-1][t]) * D[zGrid[z,t],t] . \quad (5.3)$$

- (4) Mittels der Funktion `BuildMonitorDefinition` wird die Berechnungsvorschrift der zum jeweiligen Gitter gehörenden Monitorfunktion gebildet und den Hilfsvariablen sowie der entsprechenden Liste von Modellgleichungen unter der Bezeichnung `ppMGmon1[z,t]` usw. angefügt. Falls der Benutzer über die Regel `MonitorStates` eine Liste von zu überwachenden Zuständen angegeben hat, wird untersucht, ob diese sinnvoll definiert sind. Sinnvoll bedeutet, dass diese Zustände im aktuellen Abschnitt mit verteilten Gleichungen überhaupt auftreten und auch wirklich verteilte Zustände repräsentieren. Bei Fehleingaben wird eine Fehlermeldung ausgegeben und zu (4.2) verzweigt.
- (4.1) Sind die vom Benutzer vorgegebenen, zu überwachenden Zustände sinnvoll definiert, werden diese übernommen.
- (4.2) Sind die vom Benutzer vorgegebenen, zu überwachenden Zustände nicht sinnvoll definiert oder wurde keine Vorgabe gemacht, wird nach allen verteilten Zustandsgrößen im aktuellen Gleichungsabschnitt gesucht (Funktion `BuildMonitorStates`). Diese werden damit verwendet.
- (4.3) Anhand der zur Verfügung stehenden Zustände aus (4.1) bzw. (4.2) wird die Berechnungsvorschrift für die Monitor-Funktion gebildet. Die tatsächlichen Realisierungen der Berechnungsvorschriften einschließlich der Normierung sind in Funktionen mit Namen `BuildMonitor` programmiert. Als Argumente werden zuerst eine Zahl als Kennzeichnung des gewünschten Monitors und ausserdem eine Liste der zu überwachenden Zustände übergeben. Das Ergebnis ist die nicht diskretisierte Berechnungsvorschrift. Für künftige Erweiterungen muss einfach eine neue `BuildMonitor`-Funktion in fortlaufender Nummerierung erstellt werden, welche in ähnlicher Weise wie die vorhandenen Funktionen eine Berechnungsvorschrift generiert. Um Benutzerfehler abzufangen, wurde zusätzlich eine interne Variable mit Namen `MGNumberOfMon` erstellt, welche die höchste Nummer der verfügbaren Monitorrealisierungen enthält. Bei `MGNumberOfMon = 3` dürfen vom Benutzer also die Monitorrealisierungen 1 bis 3 verwendet werden. Eine Fehleingabe bewirkt automatisch die Ausgabe einer Fehlermeldung sowie die Verwendung des Defaultmonitors. `MGNumberOfMon` hat in der aktuellen Version den Wert 1.
- (5) Aus der `Computation`-Regel der Gitterdefinition werden zur Erzeugung der Gittergleichungen für die Randpunkte die zugeordneten Werte der Ortskoordinate

- berechnet, also zum Beispiel $z_1 = 0$ und $z_P = L$. Das Verfahren ist so ausgelegt, dass der linke bzw. rechte Rand des Ortsbereichs als erster bzw. P -ter Gitterpunkt festliegt.
- (6) Mit dem Ergebnis aus (5) können die beiden Gleichungen für die Randpunkte generiert und in einen Gleichungsabschnitt geschrieben werden.
 - (7) Die Punktdichten $n_k^+(t)$ werden mit den Bezeichnern `ppMGn1[ppI][t]` usw. versehen und die Beziehungen entsprechend Gleichung (4.10) in denselben Gleichungsabschnitt wie unter Schritt (6) geschrieben.
 - (8) Mit der Funktion `BuildGridEquations` werden die eigentlichen Gleichungen für die Gittersteuerung erzeugt.
 - (8.1) Zunächst werden benötigte Informationen wie die Parameter μ und τ und die Bezeichnungen der aktuellen Punktdichten, der Monitorfunktionen und Gitter-Zustandsgrößen beschafft.
 - (8.2) Dann werden die rechten Seiten der Gittergleichungen entsprechend (4.18) bis (4.20) generiert. Sie sind sowohl für $\tau = 0$ als auch für $\tau > 0$ identisch.
 - (8.3) Gilt $\tau = 0$, sind die linken Gleichungsseiten identisch Null.
 - (8.4) Gilt $\tau > 0$, werden die linken Seiten entsprechend (4.23) bis (4.25) gebildet.
 - (9) Nach Einfügen der Gittergleichungen in den Gleichungsabschnitt aus Schritt (6) wird die Prozedur `TransformMovingDistributed` abgeschlossen.

5.2.2 Anpassung des Finite-Differenzen-Verfahrens

Eine Kombination des Finite-Differenzen-Moduls `FDMMethod.m` ist nach einigen wenigen Änderungen problemlos möglich.

Die PDgln. müssen dazu nach der Transformation weiterhin in nicht diskretisierter (kontinuierlicher) Form vorliegen, wie dies durch (5.2) realisiert wird. Das Finite-Differenzen-Verfahren kann dann wie gewohnt abgearbeitet werden, da es rein mathematisch arbeitet und unempfindlich gegen physikalisch nicht sinnvolle Modell-Formulierungen ist.

Änderungen betreffen die neue Zeitabhängigkeit der Abstände von Gitterpunkten. Diese sind jetzt nicht mehr konstant über die ganze Simulationszeit, sondern ändern sich laufend aufgrund der Verschiebung der Stützstellen nach der Beziehung

$$\Delta z_k(t) = z_{k+1}(t) - z_k(t) .$$

Deshalb werden sie als Hilfsvariablen eingeführt und bei jedem Zeitschritt neu berechnet. Dabei musste ebenfalls die Behandlung von Hilfsvariablen auf nicht äquidistanten Gittern vereinheitlicht werden, um die Wartbarkeit des Programmcodes zu erhöhen und den Code zu verkürzen.

5.2.3 Anpassung des Finite-Volumen-Verfahrens

Die Anpassung des Finite-Volumen-Moduls `FVMethod.m` konnte nicht so einfach erfolgen, da sich einige prinzipielle Hürden ergaben.

Da das implementierte Moving-Grid-Verfahren mit zwei unbeweglichen Stützstellen auf dem linken bzw. rechten Rand arbeitet, konnte eine unmittelbare Übertragung auf das Gitterverfahren „Volume Bounds“, welches ohne Stützstellen auf den Rändern auskommt, nicht vorgenommen werden. Erste Ansätze hatten zur Grundlage, dass die Berechnung mittels bewegter Volumengrenzen durchgeführt wird und die Gitterpunkte mittig dazwischen platziert werden. Das Moving-Grid-Verfahren ist aber in der vorgestellten Form auf feste Gitterpunkte am Rand angewiesen, um keine unkontrollierbar bewegten Ränder zu erzeugen. Da im Gitterverfahren „Volume Bounds“ aber keine Gitterpunkte am Rand vorliegen, müssten deren Werte auf den äußersten Volumengrenzen extrapoliert werden (siehe Abbildung 3.3). Diese Vorgehensweise wurde vorerst nicht umgesetzt und wird in dieser Arbeit nicht weiter betrachtet.

Eine Kombination mit dem Gitterverfahren „Grid Points“ ist grundsätzlich möglich, allerdings musste hier in der Behandlung der Randbedingungen eine gesonderte Betrachtung des Transformationsterms aus Gleichung (4.7) vorgenommen werden. Ist beispielsweise das skalare Reaktions-Konvektions-Diffusions-System aus Kapitel 3.2 gegeben, lautet die integrierte PDgl. vor Einsetzen der Randbedingungen im Fall mit bewegtem Gitter anstatt (3.17) jetzt

$$(z_1^+ - z_1) \frac{dx_1}{dt} - (x_1^+ - x_1) \frac{dz_k}{dt} = -v(x_1^+ - x_1) + D \left(\left. \frac{\partial x}{\partial z} \right|_{z_1^+} - \left. \frac{\partial x}{\partial z} \right|_{z_1} \right) + s(x_1).$$

Die FV versucht nun, alle auftretenden Terme x_1 und $\partial x / \partial z|_{z_1}$ (aber nicht dx_1/dt) aus dieser Gleichung mittels der Randbedingung

$$0 = -v(x_{zu} - x_1) - D \left. \frac{\partial x}{\partial z} \right|_{z_1}$$

zu eliminieren. Aufgrund des neu hinzugetretenen Transformationsterms $(x_1^+ - x_1) \cdot dz_k/dt$ in der PDgl. wird der Eliminationsalgorithmus von MATHEMATICA in diesem Beispiel wie in den meisten anderen Fällen keine korrekte Einbindung der Randbedingung liefern. Deshalb wird im Falle des Auftretens dieses Transformationsterms eine interne Substitution durchgeführt, welche diesen Term von der Elimination ausschließt. Je nach Art der Profilannahmen wird dieser Term dann herausfallen oder entsprechend weiterverarbeitet. Nach Einsetzen der Profilannahmen ergibt sich im betrachteten Beispiel die Rand-Dgl. (3.18)

$$\frac{1}{2}(z_2 - z_1) \frac{dx_1}{dt} = -v(x_1 - x_{zu}) + D \frac{x_2 - x_1}{\delta z_1} + s(x_1).$$

Weitere Anpassungen waren auch hier wie bei der FD im Hinblick auf die neue Zeitabhängigkeit der Abstände von Gitterpunkten und Volumengrenzen notwendig,

genauso wurde die Behandlung der Hilfsvariablen auf nicht äquidistanten Gittern vereinheitlicht. Die Beziehungen für Abstände von Gitterpunkten

$$\delta z_k(t) = z_{k+1}(t) - z_k(t) \quad k = 1(1)P - 1$$

und Abstände von Kontrollvolumengrenzen

$$\begin{aligned} \Delta z_k(t) &= 0.5(z_{k+1}(t) - z_{k-1}(t)) \quad k = 2(1)P - 1 \\ \Delta z_1(t) &= 0.5(z_2(t) - z_1) , \quad \Delta z_P(t) = 0.5(z_P - z_{P-1}(t)) \end{aligned}$$

mussten wie bei der FD als zeitabhängige Hilfsgrößen eingeführt werden.

5.2.4 Weitere Änderungen

Im Zuge der Vereinheitlichung der Behandlung der Hilfsvariablen wurde im Modul `MDDisc.m` eine neue Hilfsliste mit Namen `GeneratedAuxElementList` geschaffen, welche während der gesamten Vorverarbeitung die Definitionen neu erzeugter Hilfsvariablen sammelt, um sie nach erfolgter Diskretisierung in einen eigenen Gleichungsabschnitt zu schreiben. Dazu wurden im Modul `DiscUtils.m` drei neue Funktionen geschrieben, nämlich `ProbablyAppendToAuxVarsList2`, `AddGeneratedAuxiliaryElements` und `TreatAuxiliaryVariables`. Außerdem wurde die neue globale Liste `TimeDependentListNew` erzeugt.

Desweiteren wird an einigen Stellen abgefragt, ob ein bewegtes Gitter vorliegt und demzufolge bestimmte Operationen getätigt oder unterlassen werden müssen, so beispielsweise bei der Generierung der Gitter-Hilfsvariablen (Funktion `BuildAuxVariables`) sowie beim Test auf äquidistantes Gitter (Funktion `Destination`) und bei der Generierung der Prozessgrößen (Funktion `TransformAndCompleteComponents`).

Kapitel 6

Simulationsstudien

In den folgenden Abschnitten werden einige örtlich verteilte Modelle mit Hilfe des in SYPPROT neu implementierten Moving-Grid-Verfahrens diskretisiert und anschließend mit DIVA simuliert. Dadurch sollen Kenntnisse über die Einsatzgrenzen der Implementierung und über Einstellregeln für die Parameter κ , τ und P gesammelt werden. Außerdem werden die Ergebnisse jeweils mit Simulationsversuchen mittels der schon vorhandenen Verfahren bezüglich Genauigkeit und Effizienz verglichen.

Die Simulationen der diskretisierten DA-Modelle wurden auf einer SUN-Ultra60-Workstation mit der Simulationsumgebung DIVA vorgenommen. Das Integrationsverfahren DASSL wurde mit einer absoluten und relativen Genauigkeit von 10^{-6} , der Gleichungslöser NLEQ1S mit einer Genauigkeit von 10^{-5} betrieben (Standardeinstellungen). Soweit nicht anders angegeben, wurde ein äquidistantes Anfangsgitter verwendet, der Moving-Grid-Parameter der zeitlichen Gitterglättung $\tau = 0$ gewählt und der Monitor zur Überwachung aller Zustände eingesetzt.

Einige Bemerkungen zu den Grafiken, Tabellen und Bezeichnungen:

- Die Grafiken der Lösungsverläufe mit bewegtem Gitter enthalten immer ein Ortsprofil mit Kreisen. Diese geben die Lage der Stützstellen zum jeweiligen Zeitpunkt an, um einen Eindruck von der Anpassung des Gitters zu vermitteln.
- Sind Einheiten nicht explizit angegeben, handelt es sich um normierte Größen.
- Unter der *Simulationszeit* ist im Folgenden immer der Definitionsbereich der Zeitvariablen t im Modell gemeint, unter *(CPU-)Rechenzeit* wird die Dauer der Prozessorbenutzung bei der tatsächlichen Berechnung verstanden. Damit diese Rechenzeiten für ein Modell vergleichbar sind, wurde mit identischen Simulationsexperimenten, Ausgabezeiten und Prozessoren für festes bzw. bewegtes Gitter gearbeitet.

6.1 Transportmodell

Ein anspruchsvolles Problem für die numerische Integration stellen reine Transportmodelle, also lineare hyperbolische PDgln. erster Ordnung, dar.

Gegeben sei ein normiertes Konvektionsmodell für die Konzentration $c(z, t)$ eines Stoffes mit den Gleichungen

$$\frac{\partial c}{\partial t} = -v \frac{\partial c}{\partial z} \quad t > 0, \quad 0 < z < L \quad (6.1)$$

$$c(z, 0) = 0 \quad 0 < z < L, \quad c(0, t) = h(t) \quad t \geq 0 \quad (6.2)$$

und mit der konstanten Transportgeschwindigkeit $v = 1$.

Für dieses Modell lässt sich nach Laplace-Transformation auf die Form

$$\frac{dC}{dz} + \frac{s}{v} C(z, s) = 0, \quad C(0, s) = \frac{1}{s}$$

mit Laplace-Variable s die Lösung im Laplace-Bereich

$$C(z, s) = \frac{1}{s} e^{-\frac{s}{v}z}$$

beziehungsweise die Lösung im Zeitbereich nach Rücktransformation leicht angeben:

$$c(z, t) = h\left(t - \frac{z}{v}\right). \quad (6.3)$$

Die analytische Lösung besteht also aus einer Front mit unendlich großer Steigung, die mit der Geschwindigkeit v von links nach rechts durch die Anlage wandert. Ab dem Zeitpunkt $t = z/v$ liegt ein konstantes Profil $c(z, t) \equiv 1$ vor.

Festes Gitter

Die übliche Diskretisierungsmethode für Konvektionsterme ist eine stabile Upwind-Approximation erster Ordnung. Dabei werden zur Berechnung der ersten Ortsableitung an einem bestimmten Punkt dieser Punkt selbst und der benachbarte Punkt entgegen der Strömungsrichtung verwendet. Liegt eine Strömung in Richtung der positiven z -Achse vor, kommt also Gleichung (3.7) zum Einsatz. So werden zwar unerwünschte Oszillationen im Ortsprofil vermieden, aber aufgrund der niedrigen Fehlerordnung führt dieses Vorgehen zur sogenannten numerischen Dispersion, einem starken Verschleifen der Front.

Dies sieht man deutlich in Abbildung 6.1 links. Hier wurde ein festes äquidistantes Gitter mit $P = 200$ Punkten und der Upwind-Diskretisierung verwendet. Die CPU-Rechenzeit lag bei 1.1 Sekunden. Man erkennt, dass die Front zwar im Rahmen der

Zeichengenauigkeit für die Höhe $c(z, t) = 0.5$ an der richtigen Stelle liegt, durch die numerische Dispersion ergeben sich aber stark verschliffene Fronten und das Resultat ist insgesamt unbefriedigend. Als Referenz für die Lösung mit Moving Grid wurde deshalb eine Simulation mit 2000 Punkten durchgeführt, welche ein deutlich besseres Ergebnis lieferte, allerdings mit einer CPU-Rechenzeit von 6.6 Sekunden. Für das feste Gitter lag dabei die Integrator-Zeitschrittweite bei $\Delta t \approx 10^{-4}$ bis $7 \cdot 10^{-3}$.

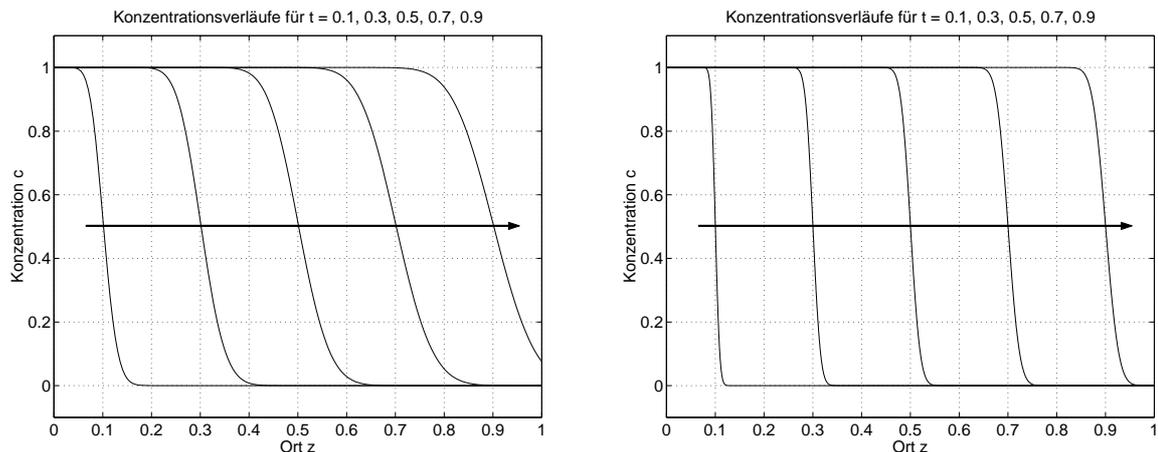


Abb. 6.1: Transportmodell, Ortsprofile der Konzentration für äquidistantes Gitter mit $P = 200$ (links) bzw. $P = 2000$ (rechts).

Bewegtes Gitter

Wird anstatt des festen Gitters die Stützstellensteuerung zugeschaltet, können sehr gute Lösungsgenauigkeiten mit vergleichsweise geringem Rechenaufwand erreicht werden. In Abbildung 6.2 sieht man das Ergebnis für $P = 40$ Stützstellen und einem Gitterglättungsparameter von $\kappa = 3$. Für kleinere Werte von κ konnten keine Simulationen in DIVA durchgeführt werden, was auf Konvergenz-Probleme der Gleichungslöser zurückzuführen ist. Die Genauigkeit der Lösung ist vergleichbar mit derjenigen der Lösung auf äquidistantem Gitter mit $P = 2000$, siehe Abbildung 6.1 rechts. Für $t < 0.5$ ist sie sogar deutlich besser, für $t > 0.7$ leicht schlechter. Da die Rechenzeit hier 1.3 Sekunden betrug, im Vergleich zu 6.6 Sekunden beim festen Gitter, ist bei etwa gleicher Ergebnisqualität die bessere Effizienz des Verfahrens für dieses Modell ersichtlich.

Abbildung 6.2 rechts zeigt die zeitliche Bewegung der Stützstellen. Man erkennt die schnelle Verschiebung der Gitterpunkte zum linken Rand des Ortsbereiches, wo die Front bei $t = 0$ durch einen Einheitssprung $h(t)$ ausgelöst wird. Mit fortschreitender Zeit wandert die Mehrzahl der Gitterpunkte zusammen mit der Front nach rechts, um sich dann bei $t \approx 1$ wieder zu verteilen.

Bei Erhöhung der Stützstellenanzahl kann die Qualität der numerischen Lösung stark verbessert werden. Es muss aber festgestellt werden, dass die Wahl des Parameters

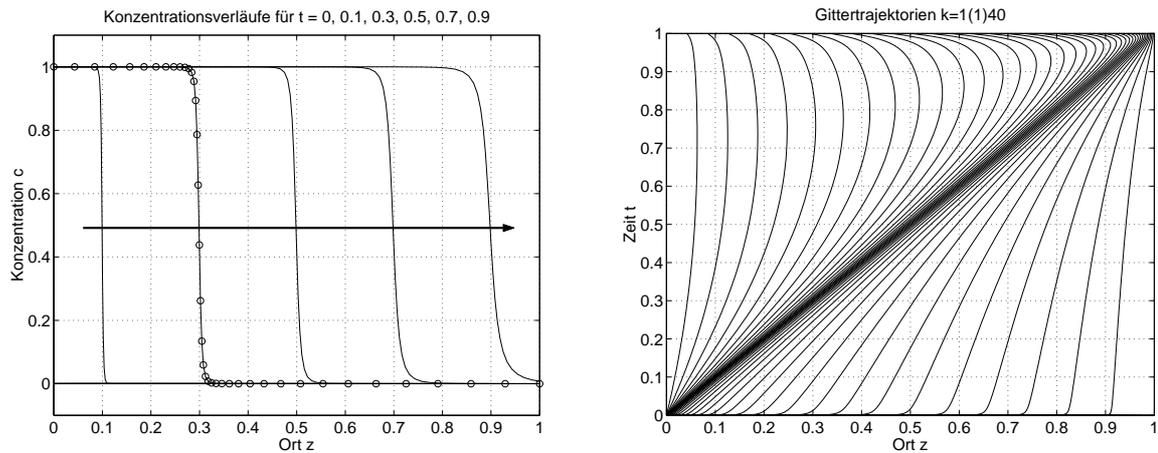


Abb. 6.2: Transportmodell, links Ortsprofile der Konzentration für bewegtes Gitter mit $P = 40$, $\kappa = 3$, $\tau = 0$, rechts zugehörige Zeitverläufe der Gittertrajektorien.

κ in diesem Fall nicht von der Anzahl P der Stützstellen unabhängig ist und hier bei $P = 80$ Gitterpunkten auf $\kappa = 4.9$ erhöht werden musste. Der Lösungsverlauf in Abbildung 6.3 wurde nach einer Rechenzeit von 2.2 Sekunden erreicht, was angesichts der Verdopplung der Stützstellenzahl recht kurz ist. Dieses Ergebnis zeigt also die sehr gute Leistungsfähigkeit des Verfahrens in Bezug auf Genauigkeit und Rechenzeit bei diesem Modell. Dies wird durch die Integrator-Zeitschrittweite bestätigt, die sich von $\Delta t \approx 10^{-6}$ zum Start der Simulation schnell bei $\Delta t \approx 2 \cdot 10^{-2}$ einpendelte. Es sei betont, dass der minimale Gitterpunkt Abstand hier etwa $\frac{1}{6} \cdot 10^{-6}$ betrug, was einem festen Gitter mit 6 Millionen äquidistanten Stützstellen entspricht. Im Mittel war der Gitterpunkt Abstand an der Front bei etwa $2 \cdot 10^{-4}$, was einem äquidistanten Gitter mit 5000 Stützstellen entspricht.

Simulationsexperimente mit Zeitglättung $\tau > 0$ zeigen, dass der Parameter κ damit auf dem Standardwert 2 belassen werden kann, allerdings wird für dieses Modell dann keine ausreichende Genauigkeit erzielt. Die Gitterpunkte müssen sich nämlich schnell zum linken Rand hin verschieben, was mit $\tau > 0$ aber verzögert wird. Die Folge ist eine sehr starke numerische Dispersion, die aufgrund der kleinen Stützstellenanzahl von $P = 40$ diejenige aus Abbildung 6.1 weit übertrifft. Mit einem selbstgewählten nicht äquidistanten Anfangsgitter kann dieser Fehler wieder ausgeglichen werden. Die Anfangslage der Gitterpunkte sollte dabei eine starke Konzentration am linken Rand aufweisen.

Die Ergebnisse für dieses einfache Modell, das numerische Standard-Verfahren meist vor große Probleme hinsichtlich einer ausreichenden Lösungsgenauigkeit stellt, sind mit bewegtem Gitter außerordentlich gut und werden in Tabelle 6.1 nochmals zusammengestellt.

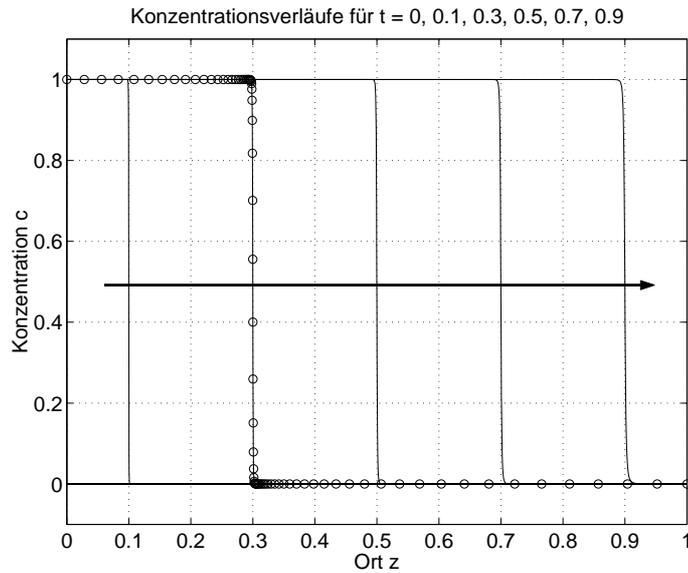


Abb. 6.3: Transportmodell, Ortsprofile der Konzentration für bewegtes Gitter mit $P = 80$, $\kappa = 4.9$, $\tau = 0$.

Diskretisierung	P	äqui-distant	κ	τ	CPU /sec.	Bemerkungen
FD, U	200	×			1.1	starke num. Dispersion
FD, U	1000	×			2.0	num. Dispersion
FD, U	2000	×			6.6	geringe num. Dispersion
FD, U	4000	×			17.3	geringe num. Dispersion
FD, U	40		3	0	1.3	geringe num. Dispersion
FD, U	80		4.9	0	2.2	kaum num. Dispersion

Tab. 6.1: Übersicht zu Simulationsexperimenten mit dem Transportmodell (6.1) - (6.2).
Abkürzung in der ersten Spalte: U - Upwind.

6.2 Dwyer-Sanders-Flammenmodell

Das Flammenausbreitungsmodell von Dwyer und Sanders umfasst ein normiertes, zweidimensionales, verkoppeltes Reaktions-Diffusions-System mit PDgln. für die Temperatur $T(z, t)$ und den Massenanteil $g(z, t)$

$$\begin{aligned} \frac{\partial T}{\partial t} &= \frac{\partial^2 T}{\partial z^2} + g \cdot 3.52 \cdot 10^6 \cdot e^{-4/T} \\ \frac{\partial g}{\partial t} &= \frac{\partial^2 g}{\partial z^2} - g \cdot 3.52 \cdot 10^6 \cdot e^{-4/T} \quad 0 < z < 1, \quad 0 < t \leq 0.006 \end{aligned} \quad (6.4)$$

mit den Randbedingungen

$$\frac{\partial T}{\partial t}(0, t) = \frac{\partial g}{\partial t}(0, t) = \frac{\partial g}{\partial t}(1, t) = 0 \quad t > 0$$

$$\begin{cases} T(1, t) = 0.2 + t/0.0002, & 0 < t \leq 0.0002 \\ T(1, t) = 1.2, & t > 0.0002 \end{cases} \quad (6.5)$$

und den Anfangsbedingungen

$$T(z, 0) = 0.2, \quad g(z, 0) = 1 \quad 0 \leq z \leq 1. \quad (6.6)$$

Die Randbedingung für T am rechten Rand stellt eine Wärmequelle dar. Wenn T seinen maximalen Wert 1.2 erreicht, wird eine steile Front ausgelöst, welche mit in etwa konstanter Geschwindigkeit von rechts nach links wandert.

Festes Gitter

Als Referenzlösung wurde das Modell mit einem festen äquidistanten Gitter und Finiten Differenzen simuliert. Bei der Verwendung von 400 oder mehr Stützstellen kann keine signifikante Veränderung des Lösungsverlaufs mehr festgestellt werden. Abbildung 6.4 zeigt die Ortsprofile der Temperatur zu verschiedenen Zeitpunkten. Die CPU-Rechenzeit für dieses Experiment betrug 6.1 Sekunden, die Zeitschrittweiten des Integrators lagen in der Größenordnung von 10^{-6} .

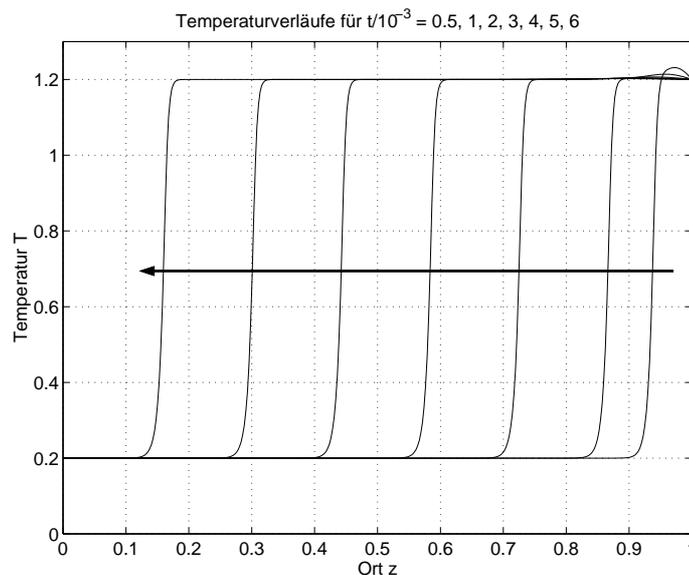


Abb. 6.4: Flammenmodell, Ortsprofile der Temperatur für äquidistantes Gitter mit $P = 400$.

Bewegtes Gitter mit Finiten Differenzen

Das Moving-Grid-Verfahren wird mit $P = 40$ Gitterpunkten, den Standardeinstellungen und FD mit Zentraldifferenz des Transformationsterms nach Gleichung (3.6) zugeschaltet. Das Ergebnis ist in Abbildung 6.5 links dargestellt und weist im Rahmen der Zeichengenauigkeit keine Unterschiede zur Referenzlösung aus Abbildung 6.4

auf. Die CPU-Rechenzeit betrug 1.2 Sekunden, stellt also eine deutliche Effizienzsteigerung bei etwa gleicher Genauigkeit dar. Vor allem die deutlich höhere Integrator-Zeitschrittweite von $\Delta t \approx 10^{-4}$ kann zur Erklärung herangezogen werden. Die zu dieser Simulation gehörenden Trajektorien der Gitterpunkte sind in Abbildung 6.6 links dargestellt.

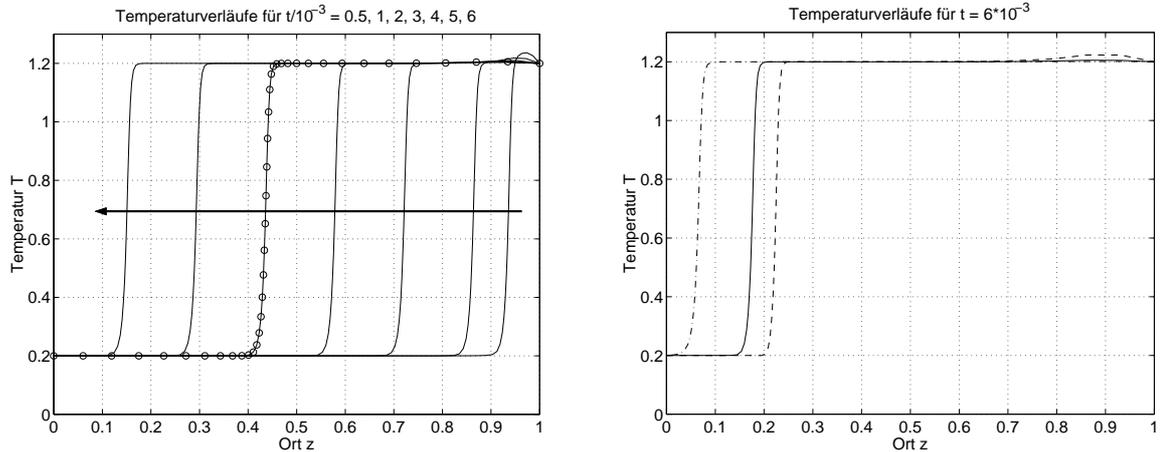


Abb. 6.5: Flammenmodell, Ortsprofile der Temperatur für bewegtes Gitter mit $P = 40$, $\kappa = 2$, $\tau = 0$. Links Diskretisierung der ersten Ortsableitung im Transformationsterm mit Zentralfferenz nach (4.26), rechts mittels Downwind (4.27) (-.-), Upwind (4.28) (- -) und Zentralfferenz mit Lagrange-Polynomen (3.13) (—).

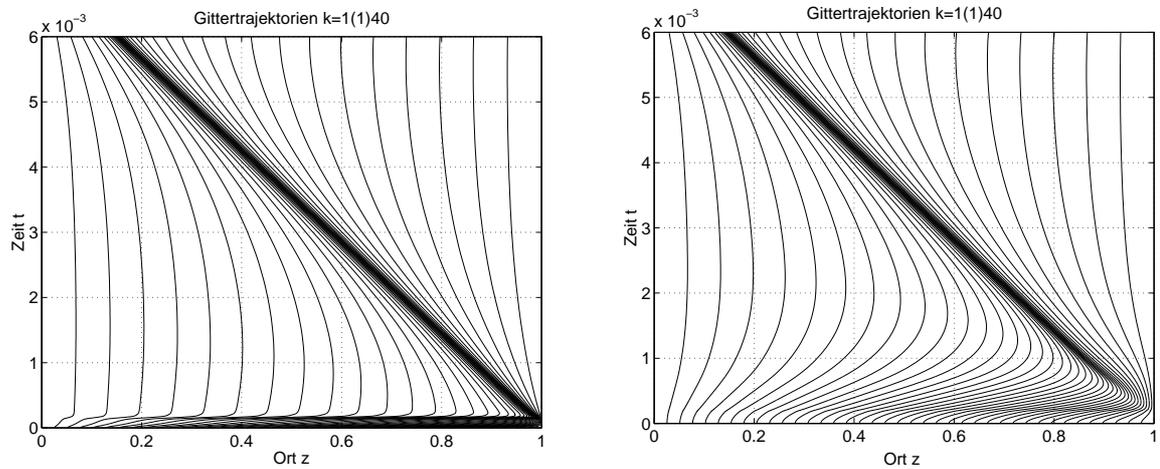


Abb. 6.6: Flammenmodell, Zeitverläufe der Gitterpunkte für $P = 40$, $\kappa = 2$ und $\tau = 0$ (links) bzw. $\tau = 0.0003$ (rechts).

Als Vergleich dazu zeigt Abbildung 6.6 rechts die Gitterbewegung für $\tau = 3 \cdot 10^{-4}$ und ansonsten unveränderten Parametern. Die Stützstellen verschieben sich weniger schnell, was einen deutlich gleichmäßigeren Verlauf der Trajektorien ergibt. Die Lösungsgenauigkeit wurde dadurch nicht verringert, die CPU-Rechenzeit erhöhte sich leicht auf 1.5 Sekunden. Ebensovienig hatte die exakte Wahl von τ einen wesentlichen Einfluss auf die Lösung, solange die Größenordnung 10^{-4} in etwa eingehalten

wurde. Zu diesem sinnvollen Wert für τ kommt man, indem man von der Simulationszeit 0.006 ausgeht und diesen Wert um etwa eine oder zwei Zehnerpotenzen verkleinert. Noch kleinere Werte sind etwa äquivalent zu $\tau = 0$, da diese dann unter der Integrator-Zeitschrittweite liegen.

Allerdings ist dieses Modell sehr empfindlich im Bezug auf die Diskretisierung des Transformationsterms. Abbildung 6.5 rechts zeigt das Ortsprofil der Temperatur zur Zeit $t = 0.006$ mit den Diskretisierungen Upwind, Downwind und Zentralfdifferenz, jeweils aus Lagrange-Polynomen. Im Vergleich zu den Referenzlösungen ist die Upwind-Front stark zurückhängend, die Front aus der Zentralfdifferenz leicht zurückhängend, und die Downwind-Front stark voreilend. Eine Erklärung für dieses Ergebnis lässt sich aus Abbildung 6.7 ablesen. Hier sind die partiellen Ortsableitungen für drei verschiedene Diskretisierungen dargestellt. Zum Zeitpunkt $t = 0.001$ sind sie an der selben Stelle $z \approx 0.85$, allerdings weist die Upwind-Diskretisierung den größten Wert, die Downwind-Diskretisierung den kleinsten Wert auf, und die Zentralfdifferenz liegt dazwischen. Zum Zeitpunkt $t = 0.004$ sind die Peaks der Ortsableitung schon eindeutig örtlich trennbar, da die Fronten mit unterschiedlichen Diskretisierungen auch verschieden schnell gewandert sind. Die Größenverhältnisse sind immer noch die selben, d.h.

$$\left. \frac{\partial x}{\partial z} \right|_{k,Downwind} < \left. \frac{\partial x}{\partial z} \right|_{k,Zentralfdiff.} < \left. \frac{\partial x}{\partial z} \right|_{k,Upwind} \quad (6.7)$$

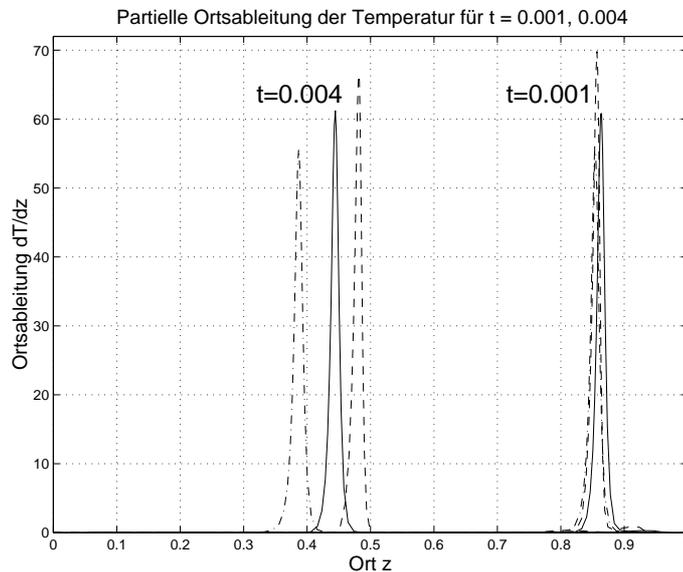


Abb. 6.7: Ortsableitungen $\partial T / \partial z$ für das Flammenmodell (6.4) - (6.6) bei bewegten Gitter für die Diskretisierungen Downwind (4.27) (-.-), Upwind (4.28) (- -) und Zentralfdifferenz nach (4.26) (—).

Dies hat Auswirkungen auf die für mitbewegte Gitterpunkte transformierten PDgln.

der Form (4.7)

$$\frac{dx_k}{dt} - \frac{\partial x}{\partial z} \Big|_{z_k(t)} \cdot \frac{dz_k}{dt} = \tilde{f} \left(x_k, \frac{\partial x}{\partial z} \Big|_{z_k(t)}, \frac{\partial^2 x}{\partial z^2} \Big|_{z_k(t)}, t \right), \quad k = 2(1)P - 1.$$

Bleibt der Diskretisierungsfehler der ersten Ortsableitung im Term $\partial x / \partial z|_{z_k(t)} \cdot dz_k / dt$ genügend klein, wird die Bewegung der Gitterpunkte und der Front korrekt berechnet. Treten hier aber große Fehler auf, werden die Gitterpunkte und damit die Front zu schnell oder zu langsam bewegt. Diese erste Ortsableitung kann deshalb als Gewichtung der Dynamik zwischen dx_k / dt und dz_k / dt interpretiert werden: ein zu kleiner Wert von $\partial x / \partial z|_{z_k(t)}$ bedingt zu große Werte von dz_k / dt und damit eine zu große Frontgeschwindigkeit. Entsprechend haben zu große Werte von $\partial x / \partial z|_{z_k(t)}$ zu kleine Frontgeschwindigkeiten zur Folge.

Weitere Simulationsexperimente mit höherer Anzahl der Stützpunkte oder anderen Werten für κ oder τ ergaben keine prinzipiellen Veränderungen oder Verbesserungen der Genauigkeit. Für dieses Modell ist also eine Diskretisierung mit Zentralfdifferenz ohne Lagrange-Polynome wie in Gleichung (3.6) angebracht, da die Approximationen mit Lagrange-Polynomen unzureichende Ergebnisse liefern. Außerdem ist die Wahl der Verfahrensparameter κ und τ nicht entscheidend und kann bei den Standardeinstellungen belassen werden.

Bewegtes Gitter mit Finiten Volumen

Wie in [22] gezeigt wurde, können Stützstellenanzahl abhängige Frontverschiebungen in FD-diskretisierten Simulationsergebnissen vorkommen. Dies ist bei der FV (normalerweise) nicht der Fall. Deshalb wurden weitere Simulationsexperimente mit der FV vorgenommen. Da die Dirichlet-Randbedingung der Temperatur-PDgl. am rechten Rand den Wert der Temperatur explizit vorgibt, ist sie im Kontext der FV nicht passend, das heißt physikalisch nicht sinnvoll formuliert. Diese Randgleichung muss deshalb von Hand eingefügt werden. Nach dieser Korrektur können die Simulationen mit den Standardparametern $\kappa = 2$ und $\tau = 0$ erfolgen, als Diskretisierungen werden „downwind“ für erste und „piecewise linear“ für zweite Ortsableitungen gewählt. Die Wahl „downwind“ entspricht in diesem Modell einer Upwind-Diskretisierung, da die Strömungsrichtung entgegen der positiven z -Achse verläuft. Die Anzahl der Gitterpunkte wird zuerst zu $P = 40$ gesetzt, wodurch sich das Ergebnis in Abbildung 6.8 links ergibt. Die CPU-Rechenzeit beträgt 1.1 Sekunden, die Integrator-Zeitschrittweite liegt bei $\Delta t \approx 7 \cdot 10^{-5}$. Man sieht im Vergleich zu Abbildung 6.4, dass die Fronten einerseits zu weit zurückliegen. Außerdem sind Ansätze von Oszillationen im Ortsprofil zu erkennen. Dieses Ergebnis ist bemerkenswert, da bei Anwendung der FD mit der gleichen Stützpunktanzahl offensichtlich keine numerisch bedingten Oszillationen auftreten.

Diese Instabilitäten kommen aber nicht aus einer falschen Diskretisierungsmethode, sondern beruhen auf einer zu geringen Anzahl von Gitterpunkten. Dies zeigt sich bei

einer Erhöhung der Stützstellenzahl auf $P = 60$ bzw. $P = 80$. Dadurch verschwinden zum einen die Oszillationen, zum anderen nähert sich die Front ihrer korrekten Lage an. Das Resultat für $P = 80$ in Abbildung 6.8 rechts erhält man nach einer Rechenzeit von 1.7 Sekunden. Es zeigt keine Abweichungen zu den Referenz-Lösungen aus den Abbildungen 6.4 und 6.5. Bemerkenswert sind noch die zwei Tatsachen, dass eine Verdopplung der Stützstellenanzahl einerseits eine Erhöhung der Rechenzeit nur um die Hälfte bedingt und andererseits im Gegensatz zum Transportmodell keine Erhöhung der κ -Werte erfordert.

Insgesamt ist klar ersichtlich, dass mit dem Moving-Grid-Verfahren eine genaue und doch effizientere Lösung als auf dem festen Gitter gefunden werden kann. Die wichtigsten Simulationsergebnisse werden nochmals in Tabelle 6.2 zusammengefasst.

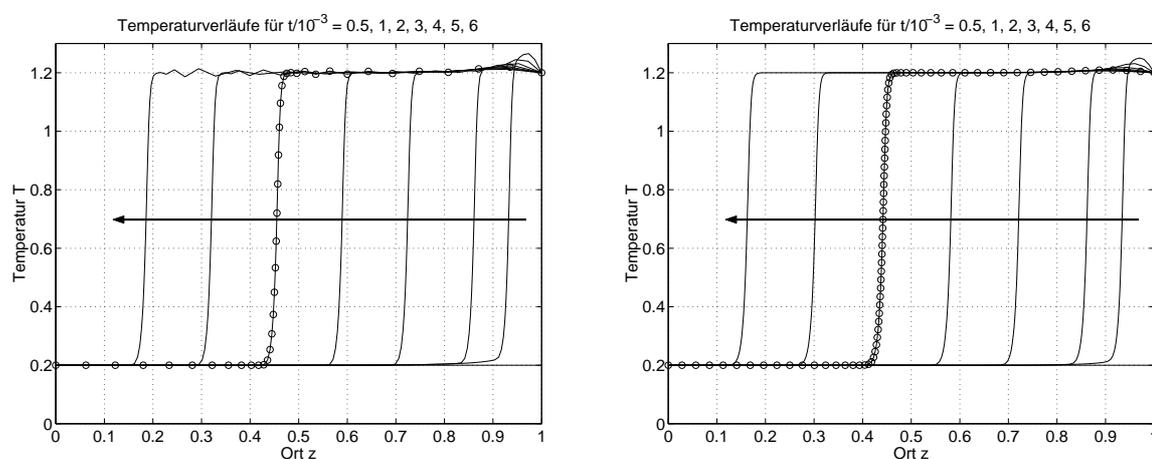


Abb. 6.8: Flammenmodell, Ortsprofile der Temperatur für bewegtes Gitter mit FV, $\kappa = 2$, $\tau = 0$ und $P = 40$ (links) bzw. $P = 80$ (rechts).

Diskretisierung	P	äquidistant	κ	τ	CPU /sec.	Bemerkungen
FD	200	×			3.6	kaum Nacheilen der Front
FD	400	×			6.1	
FD, U	40		2	0	1.2	starkes Nacheilen der Front
FD, D	40		2	0	1.3	starkes Voreilen der Front
FD, ZL	40		2	0	1.4	geringes Nacheilen der Front
FD, ZL	80		2	0	2.3	geringes Nacheilen der Front
FD, ZL	40		2	10^{-4}	1.5	geringes Nacheilen der Front
FD, Z	40		2	0	1.2	
FD, Z	40		2	$3 \cdot 10^{-4}$	1.5	
FV, U	40		2	0	1.1	numerische Oszillationen
FV, U	80		2	0	1.7	

Tab. 6.2: Übersicht zu Simulationen mit dem Flammenmodell (6.4) - (6.6). Abkürzungen bedeuten: Z - Zentraldifferenz, ZL - Zentraldiff. mit Lagrange-Polynomen, U - Upwind, D- Downwind, jeweils für erste Ortsableitung im Transformationsterm.

6.3 Gegenstrom-Adsorber

Der Gegenstrom-Adsorber (GSA) [6] besteht aus einer mobilen ersten Phase (') und einer zweiten Phase ("). Wenn Phase (") unbewegt ist, liegt ein Festbettprozess vor, strömt sie der ersten Phase entgegen, handelt es sich um einen Gegenstromprozess. Dem hier untersuchten Modell (Schema siehe Abbildung 6.9) liegen einige vereinfachende Annahmen zugrunde: es liegen isotherme Bedingungen vor und axiale Dispersion sowie der Stoffaustauschwiderstand der Phase (") werden vernachlässigt. Es wird ein binäres Gemisch aus einer Komponente mit Molenbruch $x(z, t)$ und einem inerten Träger mit Molenbruch $y(z, t)$ betrachtet.

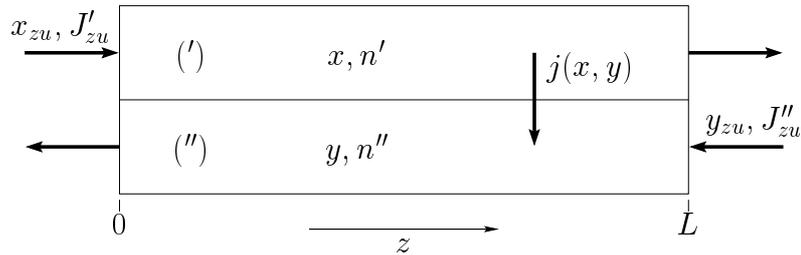


Abb. 6.9: Schema des Gegenstrom-Adsorbers

Die Modellgleichungen ergeben sich zu

$$\begin{aligned} \frac{\partial x}{\partial t} &= -\frac{\partial x}{\partial z} - j(x, y) \\ \nu \frac{\partial y}{\partial t} &= \frac{1}{\mu} \frac{\partial y}{\partial z} + j(x, y) \quad t > 0, \quad 0 < z < 1 \end{aligned} \quad (6.8)$$

mit den Rand- und Anfangsbedingungen

$$\begin{aligned} x(0, t) &= x_{zu}(t), \quad y(1, t) = y_{zu}(t) \quad t > 0 \\ x(z, 0) &= x_0(z), \quad y(z, 0) = y_0(z) \quad 0 < z < 1. \end{aligned} \quad (6.9)$$

Die Ortskoordinate z und die Zeitvariable t wurden durch folgende Normierungen dimensionslos gemacht:

$$z = \frac{\bar{z}}{L}, \quad t = \frac{\bar{t} J'_{zu}}{L n'}.$$

Die noch auftretenden Abkürzungen sind das Verhältnis der Stoffmengen ν , das Verhältnis der molaren konvektiven Ströme μ sowie der Austauschstrom j und werden mit der Gleichgewichtskonstanten α und dem Massenübertragungskoeffizienten β wie folgt definiert :

$$\nu = \frac{n'}{n''}, \quad \mu = \frac{J'_{zu}}{J''_{zu}}, \quad j(x, y) = \beta \left(\frac{\alpha x}{1 + (\alpha - 1)x} - y \right).$$

Dieses Modell soll mit der Finite-Volumen-Methode diskretisiert werden, um die Kombination mit dem Moving-Grid-Verfahren zu überprüfen. In [22] wird aufgezeigt, dass die FD demgegenüber einen lokalen Fehler durch die algebraischen Gleichungen der Randbedingungen zur Folge hat. Durch diese algebraischen Gleichungen besitzt der Ortsbereich zwischen dem Rand und der benachbarten Stützstelle keine Speicherfähigkeit, wohingegen bei der FV auch am Rand Differentialgleichungen und damit Speicherterme vorliegen.

Der GSA wird mit der Vereinfachung $1/\mu = 0$, den Eingangsgrößen $x_{zu}(t) = 0.6$ und $y_{zu} \equiv 0$ sowie den Anfangsbedingungen $x_0(z) = y_0(z) = 0$ betrieben. Dadurch liegt ein Festbettprozess mit unbewegter Komponente y vor.

Festes Gitter

Die Simulation für das feste äquidistante Gitter mit $P = 200$ Punkten führt auf das Ergebnis in Abbildung 6.10 links. Als Referenzlösung dienen die Ortsprofile aus Abbildung 6.10 rechts mit $P = 1000$ Punkten und deutlich geringerer numerischer Dispersion. Man erkennt die Dominanz der Konvektionsterme, wobei die Abweichung von einer idealen Schockfront aus dem nichtlinearen Austauschterm $j(x, y)$ kommt. Die CPU-Rechenzeit betrug 1.7 Sekunden für $P = 200$ bzw. 10.0 Sekunden für $P = 1000$, wobei die Integrator-Zeitschrittweite in der Größenordnung $\Delta t \approx 10^{-3}$ bis 10^{-2} lag. Bei Erhöhung der Anzahl der Gitterpunkte wird zwar die Genauigkeit verbessert, die Rechenzeit steigt aber stark an.

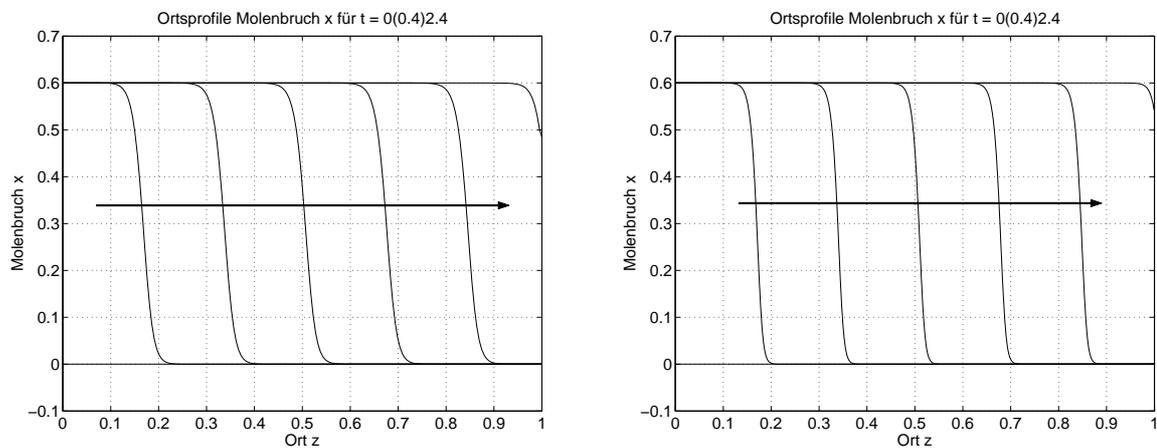


Abb. 6.10: Gegenstrom-Adsorber, Ortsprofile des Molenbruchs x für äquidistantes Gitter. Links $P = 200$, rechts $P = 1000$.

Bewegtes Gitter

Bei Anwendung des bewegten Gitters wird aufgrund der Konvektionsdominanz der Wert $\kappa = 3$ für ein Ortsgitter von $P = 40$ Punkten gewählt. Dabei werden vorerst

beide Zustände, $x(z, t)$ und $y(z, t)$, von der Monitorfunktion überwacht. Die Simulationsergebnisse sind Abbildung 6.11 links zu entnehmen und weisen leichte Abweichungen zum Resultat mit festem Gitter auf. Allerdings bildet sich im Ortsprofil bei $t = 2.4$ eine Instabilität aus, die für höhere Zeiten noch anwächst. Die Rechenzeit betrug in diesem Fall 1.2 Sekunden, die Zeitschrittweite war dementsprechend höher als im Fall mit festem Gitter und lag bei $\Delta t \approx 0.03$.

Eine Erhöhung der Stützstellenanzahl bis $P = 80$ brachte keine sichtbaren Veränderungen des Lösungsverlaufs. Um diese Simulationen durchführen zu können, müssen die κ -Werte übermäßig erhöht werden, damit Konvergenzprobleme vermieden werden. So erforderten 60 Gitterpunkte $\kappa = 6$, für 80 Gitterpunkte war sogar $\kappa = 11$ Voraussetzung für eine Lösung. Der Vorteil, der durch die hohe Zahl der verwendeten Stützpunkte erreicht würde, wird durch diese κ -Werte wieder zunichte gemacht, da dadurch die Gitterbewegung stark eingeschränkt ist.

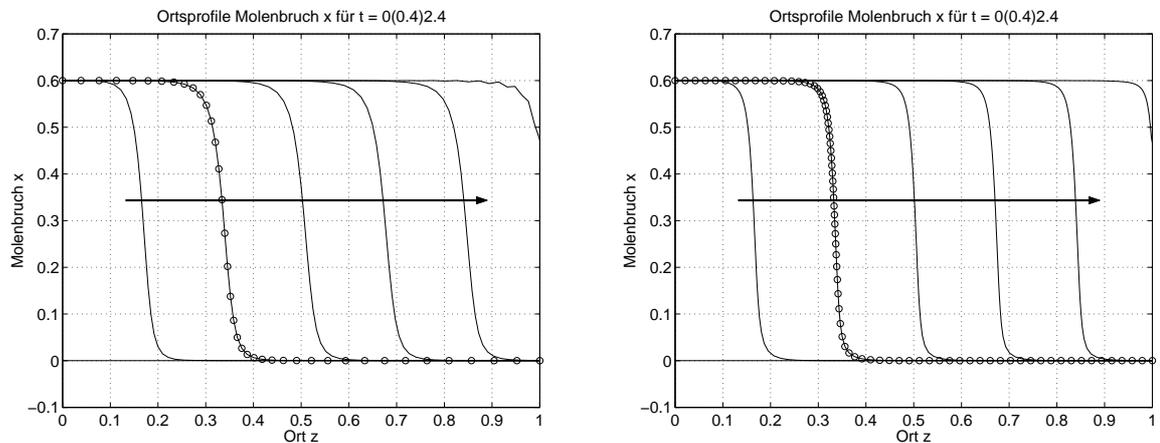


Abb. 6.11: Gegenstrom-Adsorber, Ortsprofile des Molenbruchs x für bewegtes Gitter. Links $P = 40$, $\kappa = 3$, $\tau = 0$, rechts $P = 80$, $\kappa = 1$, $\tau = 0.1$; jeweils Überwachung von x und y .

Die Resultate verbessern sich beträchtlich, wenn die Zeitglättung mit $\tau = 0.1$ aktiviert wird oder wenn nur der Zustand x überwacht wird. Wie in den Abbildungen 6.11 rechts und 6.12 links zu sehen ist, treten keine Oszillationen im Ortsprofil mehr auf, die Frontverläufe werden steiler und die Krümmungen nähern sich denen aus Abbildung 6.10 an. Dabei betragen die CPU-Rechenzeiten etwa ein Viertel bis ein Drittel verglichen mit dem festen Gitter.

Vergleichende Simulationsexperimente mit der FD brachten ähnliche Ergebnisse hinsichtlich Stabilität und Rechenzeit, die allerdings bezüglich der Lösungsgenauigkeit meist leicht schlechter ausfielen.

Dieses Modell bereitet dem Moving-Grid-Verfahren in den Standardeinstellungen und mit einer geringen Anzahl von Stützstellen also beträchtliche Schwierigkeiten, wie aus der Zusammenfassung in Tabelle 6.3 zu sehen ist. Es können aber auch hier leicht Parameterwerte gefunden werden, welche eine genaue Lösung zum Resultat haben, obwohl eine deutlich schnellere Berechnung als auf dem festen Gitter erfolgt.

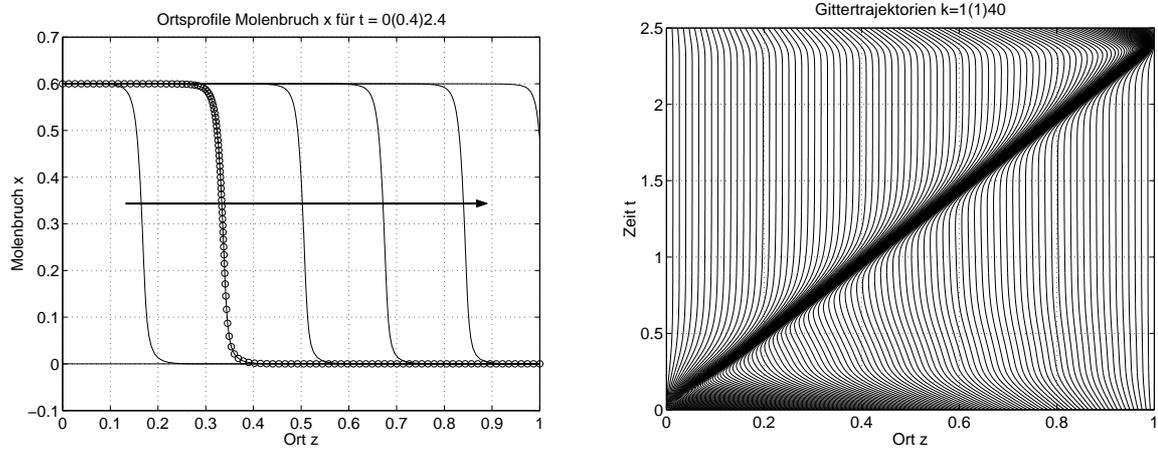


Abb. 6.12: Gegenstrom-Adsorber, links Ortsprofile des Molenbruchs x für bewegtes Gitter mit $P = 120$, $\kappa = 1$, $\tau = 0.1$, Überwachung von x . Rechts zugehörige Gittertrajektorien.

Diskretisierung	P	äqui-dist.	κ	τ	überwachte Zust.	CPU /sec.	Bemerkungen
FV	200	×				1.7	leichte num. Dispersion
FV	1000	×				10.0	
FV	40		3	0	x, y	1.1	num. Oszillationen
FV	80		11	0	x, y	1.7	num. Oszillationen
FV	40		1	0.001	x, y	1.5	num. Oszillationen
FV	40		1	0.1	x, y	1.3	leichte num. Dispersion
FV	80		1	0.1	x, y	2.3	
FV	40		3	0	x	1.1	leichte num. Dispersion
FV	80		16	0	x	1.5	leichte num. Dispersion
FV	80		1	0.1	x	2.0	
FV	120		1	0.1	x	3.4	

Tab. 6.3: Übersicht zu Simulationsexperimenten mit dem Gegenstrom-Adsorber.

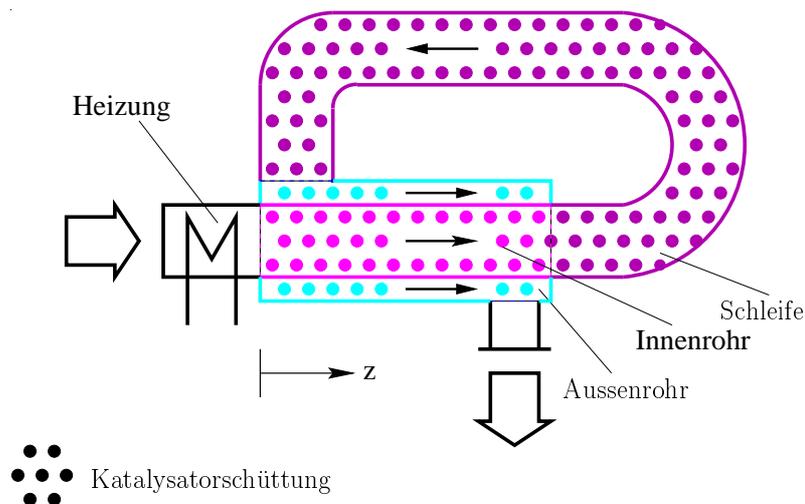


Abb. 6.13: Schema des Zirkulations-Festbett-Reaktors [16].

6.4 Zirkulations-Festbett-Reaktor

Um das Moving-Grid-Verfahren für ein komplexe reale Anlage anzuwenden und zu untersuchen, wurde das Modell eines Zirkulations-Festbett-Reaktors (ZFBR) ausgewählt. Eine schematische Übersicht der Anlage gibt Abbildung 6.13. Es handelt sich um einen Rohrreaktor mit Katalysatorschüttung, der von einem Gasgemisch mit zwei Komponenten durchströmt wird. Das Gemisch wird zuerst das Innenrohr durchqueren, bevor es über eine Schleife in das Außenrohr geleitet wird, wo ein Wärmeaustausch mit dem Innenrohr stattfindet. Danach verlässt das Gemisch die Anlage. Am Eingang in den Reaktor kann dem Gasgemisch durch eine Heizung Wärme zugeführt werden und damit eine bestimmte Eingangstemperatur eingestellt werden.

Jeder der drei Ortsbereiche Innenrohr, Schleife und Außenrohr wird durch eine dynamische PDgl. für die Energiebilanz und zwei PDgln. für die Materialbilanzen mit Quasistationaritäts-Annahmen beschrieben. Neben den Randbedingungen am Eingang und Ausgang des Reaktors müssen Innenrandbedingungen an den Übergängen der drei Ortsbereiche formuliert werden. Dadurch liegen sowohl sequentiell als auch parallel verschaltete Ortsbereiche vor. Die Rohre haben die Länge $L_{Rohr} = 0.45 \text{ m}$, die Schleife hat die Länge $L_{Schleife} = 0.584 \text{ m}$. Um eine sequentielle Darstellung der drei Bereiche zu erreichen, wird eine Ortskoordinate z eingeführt, die am Eingang mit $z = 0$ festgelegt wird, und über alle drei Ortsbereiche zum Ausgang verläuft. Am Übergang von Innenrohr zur Schleife ist also $z = L_{Rohr}$, am Übergang von Schleife zu Außenrohr $z = L_{Rohr} + L_{Schleife} = 1.034 \text{ m}$ und am Ausgang $z = 2L_{Rohr} + L_{Schleife} = L = 1.484 \text{ m}$.

Die Modellgleichungen haben für jeden der drei Ortsbereiche $j \in \{I - \text{Innenrohr}, S - \text{Schleife}, A - \text{Außenrohr}\}$ die gleiche Struktur und lauten für die Temperaturen $T^j(z, t)$

und die Molenbrüche $x_{1,2}^j(z, t)$

$$(\rho c_p)_s \frac{\partial T^j}{\partial t} = -(\rho c_p)_g v^j \frac{\partial T^j}{\partial z} + \lambda \frac{\partial^2 T^j}{\partial z^2} + \dot{Q}_{ex}^j + \sum_{i=1}^2 (-\Delta h_{R,i}) r_i(x_i^j, T^j) \quad t > 0$$

$$0 = -v^j \frac{\partial x_i^j}{\partial z} - \frac{M_g}{\rho_g} r_i(x_i^j, T^j) \quad i = 1, 2, \quad t > 0 .$$

Die Ausdrücke für die Wärmeströme $\dot{Q}_{ex}^j(z, t)$, die Geschwindigkeit v^j und die Reaktionsrate $r_i(x_i^j, T^j)$ können aus [16] entnommen werden. Mit den Randbedingungen am Eingang und Ausgang

$$(\rho c_p)_g v^I (T_{in}(t) - T^I(0, t)) + \lambda \frac{\partial T^I}{\partial z}(0, t) = 0, \quad x_{i,in}(t) - x_i^I(0, t) = 0$$

$$\frac{\partial T^A}{\partial z}(L, t) = 0 \quad t > 0 ,$$

den Innenrandbedingungen für $z = L_{Rohr}$

$$(\rho c_p)_g v^I T^I - \lambda \frac{\partial T^I}{\partial z} = (\rho c_p)_g v^S T^S - \lambda \frac{\partial T^S}{\partial z}, \quad T^I = T^S, \quad x_i^I = x_i^S \quad t > 0$$

(analog für $z = L_{Rohr} + L_{Schleife}$) sowie den Anfangsbedingungen

$$T^j(z, 0) = T_0^j(z), \quad x_i^j(z, 0) = x_{i0}^j(z)$$

wird das Modell komplettiert.

Simulationsexperiment I

Die diskretisierten Gleichungen werden auf einem Ortsgitter von 40 Punkten für das Innen- und Außenrohr und auf einem Ortsgitter mit 50 Punkten für die Schleife aufgestellt. Die Zustandsgrößen des Außenrohrs sind dabei ebenfalls auf dem Gitter des Innenrohrs definiert, so dass also kein eigenes Gitter für das Außenrohr benutzt wird. Die Parameter des Moving-Grid-Verfahrens werden für das Rohrgitter auf $\kappa_1 = 6$ und $\tau_1 = 0$ sowie für das Schleifengitter auf $\kappa_2 = 5$ und $\tau_2 = 0$ eingestellt, was sich aus verschiedenen Versuchen als zweckmäßig ergeben hat. Die Parameter $\kappa_{1,2}$ liegen so hoch, weil Sprünge der Eingangsgrößen auftreten. Außerdem wurden die Monitorfunktionen auf die Überwachung der Temperatur im Innenrohr bzw. in der Schleife eingeschränkt. Dabei wurden die Temperaturen auf den Wert $X_1 = X_2 = 1000$ K sowie die Werte der bewegten Ortskoordinaten auf $L_1 = 0.45$ m bzw. $L_2 = 0.584$ m normiert.

Nach Aufheizen des Reaktors auf ein bestimmtes Anfangsprofil mit den Eingangsgrößen

$$\begin{aligned} T_{zu} &= 500.0 \text{ K} \\ x_{1,zu} &= 0.0046 \\ x_{2,zu} &= 0.0092 \end{aligned} \tag{6.10}$$

wird die Eingangs-Temperatur schrittweise auf den Wert

$$T_{zu} = 275.0 \text{ K} \tag{6.11}$$

verringert, und zwar über eine Simulationszeit von 60 Sekunden. Die Molenbrüche im Zufluss bleiben gleich. Danach wird bis $t = 10000$ Sekunden simuliert. Dabei bilden sich besonders in der Schleife zyklische Wellen mit ansteigenden und absinkenden Temperaturfronten. Die Konzentrationen gehen innerhalb der Schleife schnell auf 0 zurück.

In Abbildung 6.14 sind einige Ortsprofile der Temperatur T und des Molenbruchs x_2 mit wandernden Fronten dargestellt. In allen Simulationsergebnissen sind im Ortsprofil gepunktete Trennlinien eingezeichnet, welche die drei Ortsbereiche trennen. Links oben sieht man, wie sich aus einem in der Schleife in etwa konstanten Profil durch Temperaturerhöhung eine Front ausbildet. Rechts oben ist dargestellt, wie diese Front dann nach rechts wandert und dadurch wieder ein konstantes Profil in der Schleife hinterlässt. In den beiden unteren Darstellungen erkennt man, dass die Konzentration von Komponente 2 ebenfalls eine Front ausbildet und dabei an denjenigen Orten stark abnimmt, wo ein starker Anstieg der Temperatur vorliegt. Dieser Vorgang, nämlich die Reaktion bis zu einem annähernd konstanten Verlauf und anschließende Bildung einer neuen Temperaturfront, tritt im ZFBR zyklisch.

Besonders deutlich kann man die Fronten und zyklischen Wellen aus dem Zeitverlauf der einzelnen Gitterpunkte in Abbildung 6.15 ablesen. Die Regionen mit größerer Schwärzung kennzeichnen eng beieinanderliegende Gitterpunkte und damit die Lage der Fronten. In der Schleife zwischen $z = 0.45$ m und $z = 1.034$ m sieht man die sich wiederholende Bewegung der Gitterpunkte. Die Zykluszeit lässt sich an der Z-förmigen Struktur leicht zu $t_z \approx 2300$ Sekunden ablesen.

Als Vergleich wurde eine Simulation mit festem äquidistanten Gitter mit 213 Stützstellen für das Innen- und Außenrohr und 232 Stützstellen für die Schleife herangezogen. Bis zum Zeitpunkt $t \approx 2200$ Sekunden stimmen die Ergebnisse sehr gut überein, danach ergeben sich zum Teil starke Abweichungen in der Lage der Fronten, wobei das qualitative Verhalten und die Zykluszeit aber übereinstimmen.

Die Resultate für das bewegte Gitter wurden nach einer Rechenzeit von 42.2 Sekunden erhalten, für das feste Gitter dauerte die Berechnung 105 Sekunden, also etwa die 2.5-fache Zeitspanne.

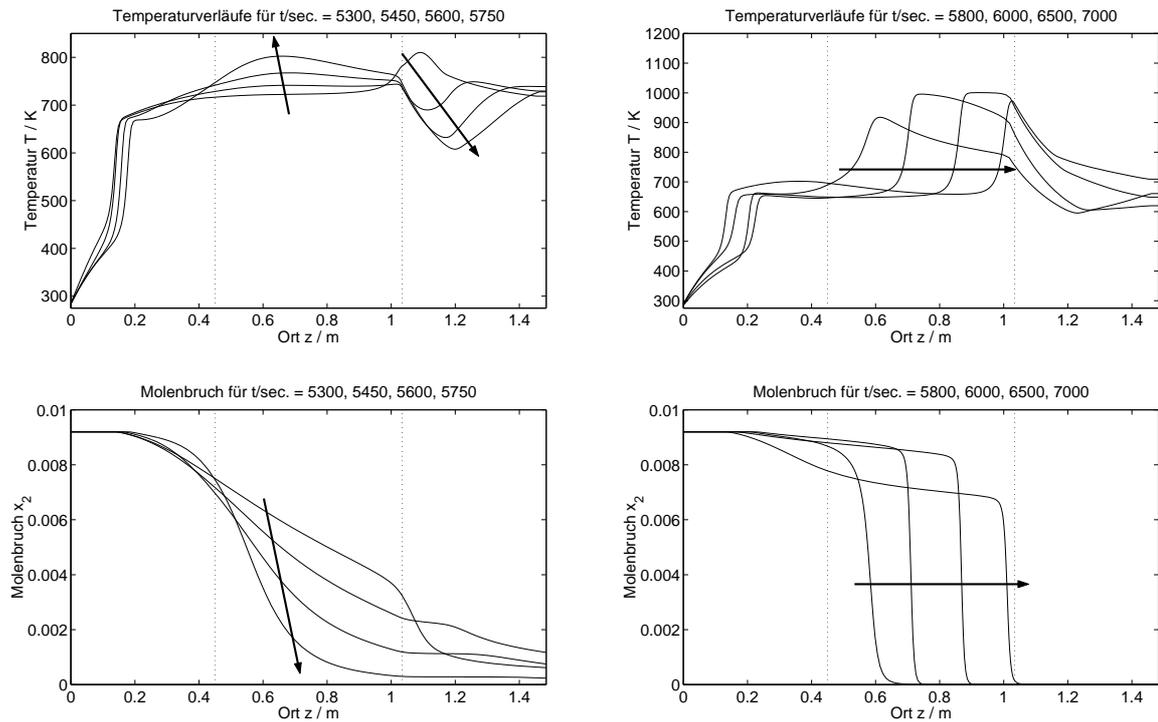


Abb. 6.14: Zirkulations-Festbett-Reaktor, Ortsprofile der Temperatur und des Molenbruchs nach Experiment I mit $P_1 = 40$, $\kappa_1 = 6$, $\tau_1 = 0$ für das bewegte Rohrgitter und $P_2 = 50$, $\kappa_2 = 5$, $\tau_2 = 0$ für das bewegte Schleifengitter.

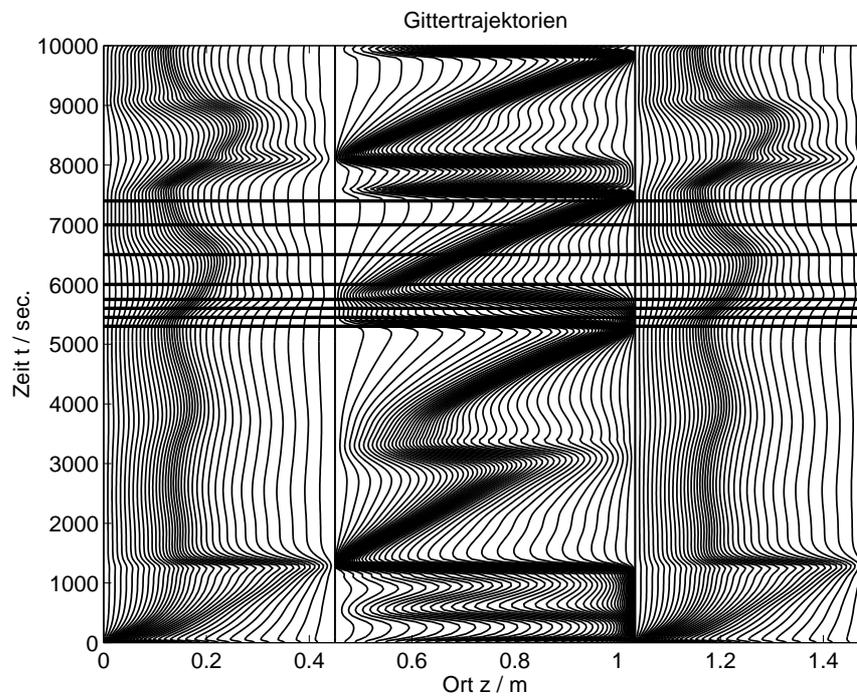


Abb. 6.15: Zirkulations-Festbett-Reaktor, Gittertrajektorien für Experiment I. Horizontale Linien kennzeichnen die Zeitpunkte aus Abb. 6.14.

Simulationsexperiment II

Ausgehend vom oben erwähnten Anfangsprofil werden die Eingangsgrößen (6.10) beibehalten und die stationäre Lösung berechnet, indem bis zur Zeit $t = 50000$ Sekunden simuliert wird. Abbildung 6.16 zeigt die Resultate für das feste Gitter mit 213 bzw. 232 Punkten für das Rohr bzw. die Schleife und für das bewegte Gitter mit 40 bzw. 50 Punkten. Allerdings muss ein qualitativer Unterschied festgestellt werden, da das stationäre Profil für das bewegte Gitter merklich höher liegt als das Profil für das feste Gitter. Dies wird auch an den Ausgangstemperaturen bei $z = L = 1.484$ m von 908.5 K bzw. 900.4 K deutlich. Hier kann ein relativer Fehler von 1% festgestellt werden. Wird mit veränderten Parametern von $\kappa_{1,2} = 1$ und $\tau_{1,2} = 0.1$ gerechnet, ist die Ausgangstemperatur sogar bei 918.8 K.

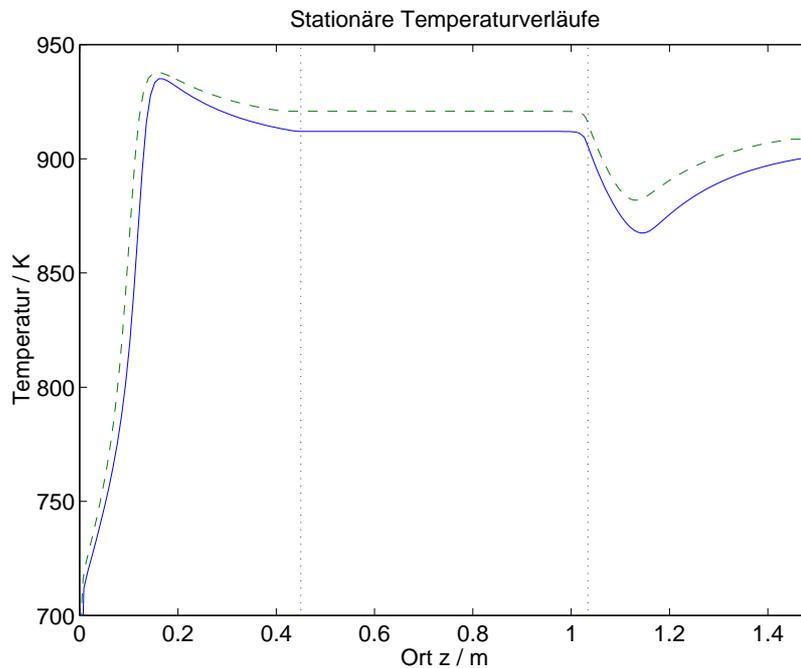


Abb. 6.16: Zirkulations-Festbett-Reaktor, stationäres Ortsprofil der Temperatur für Eingangsgrößen (6.10); festes Gitter (—), bewegtes Gitter (- -)

Eine mögliche Erklärung für diese Abweichungen ist in der FD begründet. Die in diesem Modell vorkommenden Randbedingungen mit Ortsableitungen erster Ordnung werden durch die Approximation in algebraische Gleichungen umgeformt, beispielsweise

$$\frac{\partial T^A}{\partial z}(L, t) = 0 \quad \Rightarrow \quad \frac{T_P^A - T_{P-1}^A}{z_P - z_{P-1}} = 0 .$$

Ähnliches geschieht an allen Rändern der Ortsbereiche, wo eine Ortsableitung erster Ordnung auftritt. Dadurch entsteht über die algebraische Gleichung ein direkter Durchgriff vom Randwert z_1 bzw. z_P ins Innere des Ortsbereichs, obwohl physikalisch

eine dynamische Gleichung mit Speicherterm, also eine Dgl., wirken müsste. Liegt nun ein festes äquidistantes Gitter mit genügender Anzahl an Stützstellen vor, ist dieser Fehler immer gleich groß und verringert sich mit steigender Anzahl der Gitterpunkte. Wird aber ein bewegtes Gitter benutzt, kann dieser Fehler durch die Verschiebung der Stützstellen einmal sehr klein sein (z.B. in Abbildung 6.15 im Bereich $t \approx 0$ und $z \approx 0$), in anderen Regionen aber auch sehr groß werden (z.B. in Abbildung 6.15 im Bereich $t \approx 5000$ sec. und $z \approx 0.45$ m). Es treten also zusätzliche Fehler auf, die um so gravierender sind, je stärker Gitterpunkte vom Rand entfernt sind.

Eine mögliche Abhilfe wäre das „Festklemmen“ von zwei oder mehr Stützpunkten am Rand mit sehr kleinen Abständen zu z_1 bzw. z_P , um den Durchgriff ohne Speicherterm ins Innere des Ortsbereiches so klein wie möglich zu halten. Eine Diskretisierung des ZFBR mit Finiten Volumen könnte diese Fehlerquellen ebenfalls ausschließen. Eine Verkopplung sequentiell angrenzender Ortsbereiche mit Hilfe von Innenrandbedingungen ist aber derzeit mit dem symbolischen Vorverarbeitungswerkzeug für Finite Volumen noch nicht möglich.

6.5 Einstellregeln und Benutzungshinweise

Um die Erkenntnisse und Erfahrungen aus den verschiedenen Simulationsstudien zusammenzufassen, werden in den folgenden Abschnitten einige Faustregeln und Hinweise zur Handhabung des Moving-Grid-Verfahrens zusammengestellt. Zum Teil widersprechen sich einzelne Anforderungen an die Verfahrens-Parameter. Dafür müssen eventuell fallabhängige Kompromisse geschlossen werden.

Örtliche Gitterglättung κ

Der Parameter κ der örtlichen Gitterglättung bewirkt eine lokale Begrenzung der Abstände von Gitterpunkten. Die Intervalle zwischen Stützstellen werden also nicht beliebig vergrößert oder verkleinert, sondern unterliegen der Beschränkung (4.21).

Die Standardeinstellung von $\kappa = 2$ kann erfahrungsgemäß immer dann sehr gute Simulationsergebnisse liefern, wenn in den Modellgleichungen Ortsableitungen zweiter Ordnung vorkommen, und keine Sprünge der Randbedingungen oder Eingangsgrößen stattfinden.

Treten stark konvektionsdominante Modelle oder Sprünge in Systemeingängen oder Randbedingungen auf, muss die örtliche Glättung nach den Erfahrungen aus Simulationsexperimenten auf κ -Werte von 3 oder höher eingestellt werden.

Außerdem wurde festgestellt, dass mit zunehmender Anzahl der Gitterpunkte auch der Wert von κ meist erhöht werden muss, wobei allerdings kein allgemeiner Zusammenhang gefunden werden konnte.

Zeitliche Gitterglättung τ

Die Glättung der Zeittrajektorien der Gitterpunkte wird mit dem Parameter τ gesteuert und wirkt sich in einer verzögerten Anpassung der Stützstellen auf die Fronten aus. Darüberhinaus ist eine Wahl von $\tau > 0$ oft für die Stabilisierung der numerischen Lösung förderlich [14]. Wie bereits erwähnt, ist die zeitliche Gitterglättung in vielen Anwendungsfällen aber überflüssig. Wenn starke Ortsgradienten im Anfangsprofil der Lösung vorkommen und ein äquidistantes Anfangsgitter gewählt wird, ist ein Wert $\tau > 0$ sogar für einen verzerrten und falschen Lösungsverlauf verantwortlich, da sich das bewegte Gitter zu langsam anpasst.

In einigen Fällen ist eine zufriedenstellende Lösung aber durch die Wahl von $\tau > 0$ überhaupt erst möglich. Dies ist beispielsweise für nichtkonstante Anfangsprofile der Fall. Wird eine zeitliche Gitterglättung gewählt, so sollte der Wert von τ viel kleiner als die Simulationszeit gewählt werden. Eine Wahl von τ kleiner als die Größenordnung der Integratorschrittweite hat auf die Zeittrajektorien kaum einen glättenden Einfluss, da die Anpassung der Gitterpunkte nicht verzögert wird. Dies kann manchmal gewünscht sein (siehe oben), dadurch wird aber auf den stabilisierenden Einfluss der Zeitglättung verzichtet.

Mit $\tau > 0$ kann der Parameter κ der örtlichen Gitterglättung meist auf dem Standardwert 2 belassen oder auf den Wert 1 gesetzt werden (siehe Beispielmodell GSA). Damit wird zu hohen κ -Werten, z.B. für eine hohe Anzahl an Stützstellen, entgegengewirkt. Allgemein hat die zeitliche Gitterglättung keine übermäßig verlängerte CPU-Rechenzeit zur Folge. Gelegentlich ist der Rechenaufwand durch Werte von $\tau > 0$ sogar geringer.

Anzahl der Gitterpunkte P

Die Anzahl der Gitterpunkte ist wie auch im Fall eines festen Gitters immer problemabhängig zu wählen. Dabei muss ein Kompromiss zwischen Genauigkeit und Rechenzeit gefunden werden. Es können deshalb nur grobe Faustregeln angegeben werden, welche aber auch sehr unbefriedigende Lösungen liefern können.

Aus verschiedenen Simulationsexperimenten und Vergleichen mit der Literatur kann für den Fall des bewegten Gitters ein Richtwert von 30 bis 80 Stützstellen erhalten werden, wobei niedrige Werte aus diesem Bereich oft gute Resultate ergaben.

In [24] wird eine weitere Abschätzung zur Anzahl der Gitterpunkte angegeben. Da der Wert von κ bei gegebenem P die minimalen und maximalen Veränderungen der Stützstellenabstände bestimmt, kann daraus abgeleitet werden, wieviele Stützstellen lokal in einer Region mit Übergang von kleinen zu großen Ortsgradienten notwendig sind. Es seien in einer Region Q Stützstellen, und die Lösung erfordere eine Verfeinerung des Gitters um den Faktor 10^m . Da die Änderung der Punktdichten durch

Beziehung (4.21) von dem Wert $1 + 1/\kappa$ beschränkt wird, folgt aus

$$\left(1 + \frac{1}{\kappa}\right)^{0.5Q} = 10^m,$$

dass Q mindestens den Wert

$$Q = 2m \frac{\ln(10)}{\ln(1 + 1/\kappa)} \approx \frac{4.6m}{\ln(1 + 1/\kappa)} \quad (6.12)$$

haben soll. Der Faktor 2 kommt aus der Annahme, dass nach einer Gitterverfeinerung auch wieder eine Gitterexpansion folgt. Durch Aufsummieren dieser lokalen Abschätzung kann eine grobe Angabe über die benötigten Stützstellen gemacht werden. Als Beispiel wird das Flammenmodell aus Kapitel 6.2 betrachtet. Es wird eine Verfeinerung um den Faktor 1000 angestrebt, was auf $m = 3$ führt. Mit dem Parameter $\kappa = 2$ ergibt die Abschätzung $Q \approx 34$ lokale Gitterpunkte zusätzlich zu den Gitterpunkten für Bereiche ohne Gradienten. Dadurch folgert man eine ungefähre Stützstellenzahl zwischen 40 und 50 für das komplette Gitter.

Anfangsgitter

In den durchgeführten Simulationsexperimenten wurde meist mit einem äquidistanten Anfangsgitter gerechnet. Dies ergibt zumindest für $\tau = 0$ keine Nachteile, da das Gitter sehr schnell auf die Lösung eingestellt wird. Liegt aber ein Anfangsprofil der Zustandsgrößen mit steilen Fronten vor, ist die Aktivierung der zeitlichen Gitterglättung oft notwendig, um die numerische Lösung zu ermöglichen. Dann sollte ein selbstdefiniertes Anfangsgitter gewählt werden, welches schon eine Konzentration der Gitterpunkte am Ort der Fronten im Anfangsprofil besitzt. Andernfalls kann es zu starken Verfälschungen aufgrund der verzögerten Verschiebung der Stützstellen kommen. Die Beschränkung (4.21) soll dabei möglichst eingehalten werden.

Kapitel 7

Zusammenfassung

In der vorliegenden Arbeit wurde auf der Grundlage der Method-of-Lines ein Verfahren zur dynamischen Anpassung der Ortsstützstellen an die numerische Lösung von PDgln. implementiert und untersucht. Der Moving-Grid-Algorithmus basiert auf dem Prinzip der Gleichverteilung und benutzt zur Fehlerschätzung den Bogenlängen-Monitor. Anhand einiger Testmodelle wurde das Verfahren überprüft und mit Diskretisierungen auf äquidistantem Gitter verglichen. Dabei wurde auch auf Probleme wie Wahl der Parameter, Diskretisierung des Transformationsterms, numerische Stabilität der Lösung, Konvergenzprobleme und Schwierigkeiten bei Verwendung der FD eingegangen.

Es konnte für das Transportmodell, das Flammenmodell und den Gegenstrom-Adsorber gezeigt werden, dass mittels des Moving-Grid-Verfahrens jeweils die Genauigkeit der Simulationsexperimente mit festem Gitter erreicht oder übertroffen wird, wobei deutlich niedrigere Rechenzeiten benötigt wurden. Die Rechenzeiten betragen etwa ein 1/6 bis 1/2 der Rechenzeiten bei Lösung auf äquidistantem Gitter, die Anzahl der benötigten Punkte lag bei etwa 1/50 bis 1/5. Auch beim Zirkulations-Festbett-Reaktor wurde eine niedrigere Rechenzeit erreicht, allerdings ergeben sich aufgrund der Verwendung der FD Fehler in der Lage der Fronten. Damit kann eindeutig Effizienzsteigerung durch dieses Verfahren bestätigt werden, wenn auch zum Teil Parameteranpassungen notwendig waren.

Vor- und Nachteile des Verfahrens

Für die Wahl dieses Moving-Grid-Verfahrens waren folgende Gründe ausschlaggebend. Der Algorithmus erlaubt die simultane Lösung der PDgln. und der Gleichungen zur Gittersteuerung und ermöglicht damit eine einfache Implementierung in SYPPROT. Außerdem ist die Kombination mit den Finite-Differenzen- und Finite-Volumen-Methoden weitgehend problemlos möglich. Im Verfahren selbst ist die Berechnung von Fehlermaß und Stützstellensteuerung anschaulich und einfach gehalten und besitzt nur zwei Parameter κ und τ für örtliche bzw. zeitliche Gitterglättung,

wobei τ meist zu Null gesetzt werden kann. Als wichtige Eigenschaften erlaubt das Verfahren damit größere Zeitschrittweiten als auf äquidistanten Gittern und verhindert Stützstellenüberkreuzungen.

Es gibt jedoch auch Nachteile, die einer grundsätzlichen Anwendung aber nicht im Wege stehen. So wird der Diskretisierungsfehler über den Bogenlängen-Monitor nur heuristisch abgeschätzt und nicht exakt berechnet. Wie in den Simulationsstudien gezeigt, ist die Wahl der Verfahrensparameter nicht eindeutig und generell vorgebar. Deshalb sind oft Abweichungen von den Standardeinstellungen notwendig. Außerdem kann in Kombination mit Finiten Differenzen eine Verstärkung der Diskretisierungsfehler am Rand auftreten, da hier nur algebraische Gleichungen vorliegen und damit die Dynamik in Randgebieten vernachlässigt wird. Im Vergleich zur Verwendung eines festen äquidistanten Gitters wird das Differential-Algebra-Modell bei bewegtem Gitter um P Gleichungen erweitert. Dieser zusätzliche Aufwand zahlt sich aber durch die geringe Stützstellenanzahl meist in weit niedrigeren Rechenzeiten aus.

7.1 Ausblick

Mit der Implementierung des hier untersuchten Moving-Grid-Verfahrens wurde das symbolische Vorverarbeitungswerkzeug SYPPROT um eine dynamische Gittersteuerung erweitert. Sie beinhaltet das Verfahren mit den wichtigsten Eigenschaften und wurde an die bisher vorhandenen Diskretisierungsmethoden angepasst, wobei die Möglichkeit von Erweiterungen in Betracht gezogen wurde.

Als direkte Ergänzung des vorliegenden Programmcodes kommt zum einen die Implementierung von weiteren Monitor-Funktionen wie dem Krümmungsmonitor oder der Kombination aus erster und zweiter Ortsableitung in Frage. Mit Alternativen zum Bogenlängen-Monitor kann die Flexibilität und die Genauigkeit für Spezialfälle erhöht werden. Desweiteren ist eine automatische Berechnung der Anfangsverteilung von Gitterpunkten für beliebige Anfangsprofile der Zustandsgrößen denkbar. Damit wird der generelle Einsatz der zeitlichen Gitterglättung sowie die Erzeugung konsistenter Anfangsbedingungen des gesamten DA-Gleichungssystems besser unterstützt. Nicht zu realisieren ist dagegen eine Anfangsverteilung der Gitterpunkte, wenn eine Anpassung an Sprünge in Eingangsgrößen erfolgen soll, da Informationen über Eingänge im Modell nicht vorliegen.

Zuletzt seien noch Erweiterungen des Moving-Grid-Verfahrens im Hinblick auf die Kombination mit weiteren Diskretisierungsverfahren, den sogenannten High-Resolution-Schemes, angesprochen. Dabei kommen beispielsweise ENO-Schemata (Essentially-Non-Oscillatory) [2] und das Robust-Upwind-Verfahren [13, 2] in Betracht. Eine Kopplung dieser Diskretisierungsverfahren höherer Ordnung mit dem vorgestellten Moving-Grid-Verfahren verspricht eine weitere Steigerung hinsichtlich Robustheit und Effizienz [14].

Anhang A

MDS-Code Flammenmodell

```
1  (* ***** *)
   (* General Description *)
   (* ***** *)

5  GeneralDescription[
    ProgramName ->"dsflame",
    FileName    ->"dsflame.mds",
    Date        ->"13.04.2000",
10   Author     ->"Jochen Rieber",
    Comment     ->"Dwyer-Sanders flame propagation model"
   ];

   (* ***** *)
   (* Independent Variables *)
   (* ***** *)
15

   IndependentVariables[
    TimeVariable[ t ],
    SpaceVariable[ z ]
20  ];

   (* ***** *)
   (* Structure Parameters *)
   (* ***** *)
25

   StructureParameters[
    Scalar[ kmax,
30     Default-> 40,
        Comment-> "number of grid points" ]
   ];

   (* ***** *)
   (* System Parameters *)
   (* Integer- and Real-Parameters *)
   (* ***** *)
35

   SystemParameters[
```

```

RealParameters->{
  Scalar[ L,
40      Default-> 1,
      Comment-> "length of space scope" ]
  }
];

45 (* ***** *)
(* Input Variables *)
(* ***** *)

InputVariables[
50 ];

(* ***** *)
(* State Variables *)
(* ***** *)

55 StateVariables[
  Scalar[ md[z,t],
      InitialValue-> 1,
      Comment-> "mass density",
60      Domain-> "Grid[z]" ],
  Scalar[ temp[z,t],
      InitialValue-> 0.2,
      Comment-> "temperature",
65      Domain-> "Grid[z]" ]
];

(* ***** *)
(* System Equations *)
(* Distributed-Sections for partial differential equations *)
70 (* ***** *)

SystemEquations[
  Distributed[
75      Domain-> "Grid[z]",
      Comment-> "Equations of Dwyer-Sanders flame propagation model ",
      Scope-> {
          Scalar[ D[md[z,t],t] == D[md[z,t],{z,2}]
                  - md[z,t]*3.52*10^6*Exp[-4/temp[z,t]],
80          LowerBound -> D[md[z,t],z] == 0,
          UpperBound -> D[md[z,t],z] == 0,
          Name-> "Pdgl(1)",
          Comment-> "",
          Discretization -> FiniteDiff ],
          Scalar[ D[temp[z,t],t] == D[temp[z,t],{z,2}]
                  + md[z,t]*3.52*10^6*Exp[-4/temp[z,t]],
85          LowerBound -> D[temp[z,t],z] == 0,
          UpperBound -> temp[z,t] == Which[t <= 0.0002, 0.2 + t/0.0002,
                                          True, 1.2],

          Name-> "Pdgl(2)",
90          Comment-> "",

```

```

        Discretization -> FiniteDiff ]
    }
]
];
95
(* ***** *)
(* Spatial domains or grid definition *)
(* ***** *)
100 Domains[
    Grid[ Grid[z],
        Computation-> Function[{k},(k-1)*L/(kmax-1)],
        Granularity-> kmax,
        MovingGrid-> { CentralDiff->True }
105 ]
];

(* ***** *)
(* Discretization-Methods *)
110 (* ***** *)

Discretizations[
    FDMMethod[ FiniteDiff,
115         PolynomPoints -> {1,2,3},
         Eccentricity -> {0,0,0},
         OrderReduction-> {0,0,1} ],
    FVMMethod[ FiniteVol,
         Profile -> { "downwind", "piecewise-linear" } ]
120 ];

(* ***** *)
(* Process values for offline graphics *)
(* ***** *)
125 ProcessValues[
    ProcessGroup[
        DistribScope[
            Distributed[
130             Components[
                 Scalar[ md[z,t] ],
                 Scalar[ temp[z,t] ]
            ],
            Domain -> "Grid[z]",
            Range -> {0,L,1}
135         ]
    ],
    Name -> "results",
    Comment -> "state variables"
140 ];

```

Anhang B

Funktionen und globale Variablen von MovGrid.m

Funktion	Aufgabe
BuildScalarEquObject	Definition einer skalaren Gleichung
BuildScalarStateObject	Definition einer skalaren Zustandsvariablen
BuildScalarParObject	Definition eines skalaren Parameters
BuildTensorEquObject	Definition von indizierten Gleichungen
BuildTensorStateObject	Definition von indizierten Zustandsvariablen
BuildTensorTimeDepAuxObject	Definition einer indizierten zeitvarianten Hilfsvariablen
BuildTensorTimeIndepAuxObject	Definition einer indizierten zeitinvarianten Hilfsvariablen
BuildGridVariable	Generierung der Gittervariablen als Zustände oder als Hilfsgrößen
TestDistributedStateVar	Test auf verteilten Zustand
GetGridIndex	Berechnung des Indexes der Informationen zu bestimmtem Ortsbereich in MGGridParameters
BuildSpatialState	Generierung der Gitterzustände, z.B. zGrid[t] aus Ortsbereich Grid[z]
DiscretizedSpatialState	Generierung der Gitterzustände, z.B. zGrid[ppI][t] aus zGrid[t]
ContinuousSpatialState	Generierung der Gitterzustände, z.B. zGrid[z,t] aus zGrid[t]

Tab. B.1: Hilfsfunktionen des Moduls MovGrid.m

Name	Aufgabe
MGDistributedStateVars	Liste der verteilten Zustandsvariablen
MGGridList	Liste mit Namen von Gitter-Definitionen mit bewegten Gitterpunkten, in der Form { Grid[z] }
MGGridList2	wie MGGridList, aber in der Form { Grid }
MGGridParameters	Liste aller Gitterdefinitionen mit Parametern in der Form wie oben beschrieben
MGStateVarList	Liste der Zustandsvariablen der Gitterpunkte, in der Form { zGrid[t] }
MGMethodParameterList	Liste der Namen der Moving-Grid-Parameter, in der Form { ppMGkappa1,ppMGmu1 }
MGMonitorNameList	Liste der Namen der Monitorfunktionen, in der Form { ppMGmon1 }
MGPointConcList	Liste der Namen der Punktdichten, in der Form { ppMGn1 }
MGA11	allgemeine Symboldefinition
MGMonitorScheme	allgemeine Symboldefinition

Tab. B.2: Globale Variable des Moduls MovGrid.m

Literaturverzeichnis

- [1] M. Bauer. Moving Grid Methods for 1D and 2D PDEs - Implementation and simulation results. Diplomarbeit, Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart, 1999.
- [2] M. Brahmadata, R. Köhler, A. Mitrović, E.D. Gilles, and M. Zeitz. Symbolic discretization of population models for process simulation. In S. Pierucci, editor, *European Symposium on Computer Aided Process Engineering - 10*, Computer-Aided Chemical Engineering 8, pages 547–552. Elsevier, 2000.
- [3] E.A. Dorfi and L.O'C. Drury. Simple adaptive grids for 1-D initial value problems. *J. Comp. Phys.*, 69:175–195, 1987.
- [4] B. A. Finlayson. Numerical methods for problems with moving fronts. *Ravenna Park Publishing, Inc.*, 1992.
- [5] R.M. Furzeland, J.G. Verwer, and P. A. Zegeling. A numerical study of three moving grid methods for one dimensional partial differential equations which are based on the method of lines. *J. Comp. Phys.*, 89:349, 1990.
- [6] S. Grüner. Entwicklung reduzierter Modelle für Adsorptionsprozesse auf der Basis nichtlinearer Wellenphänomene. Diplomarbeit, Universität Stuttgart, 1999.
- [7] A. Holl. Numerische Lösung von partiellen Differentialgleichungen mit Moving Finite Elemente Verfahren. Studienarbeit, Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart, 1995.
- [8] P. Holl, W. Marquardt, and E.D. Gilles. DIVA - A powerful tool for dynamic process simulation. *Computers Chem. Engng.*, 12:421–425, 1988.
- [9] W. Huang, Y. Ren, and R.D. Russel. Moving mesh method based on moving mesh partial differential equations. *J. Comp. Phys.*, 113:279–290, 1994.
- [10] R. Köhler. Beschreibung Code-Generator Version 1.3.1 zur Erzeugung von DIVA-Grundelementen. Interner Bericht 99-2, Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart, April 1999.
- [11] R. Köhler. Manual SYPPROT Version Beta 1. Interner Bericht, Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart, April 2000.

-
- [12] R. Köhler, A. Gerstlauer, and M. Zeitz. Symbolic preprocessing for simulation of PDE models of chemical processes. *Journal of Math. and Comp. in Sim., special issue "Method of Lines"*, 2000 (submitted).
- [13] B. Koren. A robust upwind discretization method for advection, diffusion and source terms. In C.B. Vreugdenhil and B. Koren, editor, *Numerical Methods for Advection-Diffusion Problems*, pages 117–138. Vieweg, 1993.
- [14] S. Li and L. Petzold. Moving mesh method with upwinding schemes for time-dependent PDEs. *J. Comp. Phys.*, 131:368–377, 1997.
- [15] S. Li, L. Petzold, and Y. Ren. Stability of moving mesh systems of partial differential equations. Technical Report TR96-027, Department of Computer Science, University of Minnesota, 1996.
- [16] M. Mangold, A. Kienle, E.D. Gilles, M. Richter, and E. Roschka. Coupled reaction zones in a circulation loop reactor. *Chem. Eng. Sci.*, 54:2597–2607, 1999.
- [17] K. Miller and R.N. Miller. Moving finite elements I. *SIAM J. Numer. Anal.*, 18:1019–1032, 1981.
- [18] K.D. Mohl, A. Spieker, E. Stein, and E.D. Gilles. DIVA - Eine Umgebung zur Simulation, Analyse und Optimierung verfahrenstechnischer Prozesse. In A. Kuhn and S. Wenzel, editors, *Simulationstechnik, 11. ASIM-Symposium in Dortmund*, pages 278–283. Vieweg Verlag, Braunschweig/Wiesbaden, 1997.
- [19] S. Räumschüssel. Rechnerunterstützte Vorverarbeitung und Codierung verfahrenstechnischer Modelle für die Simulationsumgebung DIVA. *VDI Fortschritt-Berichte Nr. 20/270*, VDI-Verlag, 1998.
- [20] W.E. Schiesser. The numerical method of lines: Integration of partial differential equations. *Academic Press*, 1991.
- [21] S. Schmalzriedt. Stützstellensteuerung bei parabolischen Differentialgleichungen. Diplomarbeit, Institut für Chemische Verfahrenstechnik, Universität Stuttgart, 1993.
- [22] S. Schwarzkopf. Rechnerunterstützte Ortsdiskretisierung von partiellen Differentialgleichungen mit Hilfe von Finite-Volumen-Methoden. Studienarbeit, Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart, 1999.
- [23] Frank Tränkle. Rechnerunterstützte Modellierung verfahrenstechnischer Prozesse für die Simulationsumgebung DIVA. *VDI Fortschritt-Berichte Nr. 20/309*, VDI-Verlag, 2000.
- [24] L.G. Verwer, J.G. Blom, R.M. Furzeland, and P.A. Zegeling. A moving grid method for one-dimensional PDEs based on the method of lines. In J.E. Flaherty, P.J. Paslow, M.S. Shepard, and J.D. Vasilakis, editors, *Adaptive Methods for Partial Differential Equations*, pages 160–175. SIAM, Philadelphia, 1989.