

**Eine Methode zur Zeitdiskretisierung bei
Discontinuous-Galerkin Verfahren mit lokalen Zeitschritten**

**A local time-stepping method for discontinuous galerkin
schemes**

Seminararbeit

in der Studienrichtung
Computational Engineering Science

an der
Rheinisch-Westfälischen Technischen Hochschule Aachen

von
Melven Zöllner

Betreuer:
Jens Zudrop

Betreuender Professor:
Prof. Manuel Torrilhon

18. November 2013

Inhaltsverzeichnis

Notation und Symbole	1
1 Einleitung	3
2 RK-DG-Verfahren	4
2.1 Discontinuous-Galerkin Diskretisierung im Raum	4
2.2 Runge-Kutta Diskretisierung in der Zeit	5
3 Prädiktor-Korrektor-DG Verfahren	7
3.1 Lokaler CERK Prädiktor	7
3.2 Globaler Korrektor	8
4 Quadratur-freies Discontinuous-Galerkin Verfahren für kubische Gitter	10
4.1 Legendre-Polynome im eindimensionalen Fall	10
4.2 Quadratur-freie Darstellung mittels Projektionsmatrizen	11
4.3 Herleitung eines geeigneten lokalen Prädiktors	12
5 Lokale Zeitschritte	15
5.1 Prädiktor-Korrektor – Ansatz	15
6 Implementierung und Ergebnisse	17
6.1 Skizze des gesamten Algorithmus	17
6.2 Validierung und Performance	18
7 Fazit	20

Notation und Symbole

In dieser Arbeit wird an vielen Stellen die Einsteinsche Summenkonvention verwendet; über doppelt auftretende, freie Indizes wird summiert.

Folgende Schreibweisen sind somit äquivalent: $\partial_{x_\alpha} f_\alpha(w)$ und $\sum_{\alpha=1}^3 \frac{\partial}{\partial x_\alpha} f_\alpha(w)$. Dies ist nichts anderes als $\nabla \cdot \mathbf{f}(w)$.

\mathbf{w}	Vektor von Erhaltungsgrößen in Raum und Zeit: $\mathbf{w} : \mathbb{R}^3 \times \mathbb{R}^+ \rightarrow \mathbb{R}^m, m \in \mathbb{N}$
\mathbf{f}_α	Flussvektor in einer Raumrichtung α für ein System von Erhaltungsgleichungen: $\mathbf{f}_\alpha : \mathbf{w} \rightarrow \mathbf{f}_\alpha(\mathbf{w}) \in \mathbb{R}^m$
\mathbf{s}	Vektor der Quellterme für ein System von Erhaltungsgleichungen: $\mathbf{s} : \mathbf{w} \rightarrow \mathbf{s}(\mathbf{w}) \in \mathbb{R}^m$
∂_t	(Schwache) partielle Ableitung nach t ; Kurzform für $\frac{\partial}{\partial t}$
∂_{x_α}	(Schwache) partielle Ableitung nach x_α ; Kurzform für $\frac{\partial}{\partial x_\alpha}$
α	Index für eine Richtung im Raum: $\alpha = 1, 2, 3$
w	Funktion in Raum und Zeit: $w : \Omega \times \mathbb{R}^+ \rightarrow \mathbb{R}$
$\mathbf{f}(w)$	Physikalischer Fluss: $\mathbf{f} : w \rightarrow \mathbf{f}(w) \in \mathbb{R}^3$
w_0	Anfangsdaten im Raum: $w_0 : \Omega \rightarrow \mathbb{R}$
Ω	Rechengebiet (hier $\Omega = \mathbb{R}^3$), unterteilt in Teilgebiete Ω_k
k, l	Indizes für die Teilgebiete des Rechengebietes Ω : $k, l \in \mathbb{N}$
p	Polynomgrad; kann sich im mehrdimensionalen wahlweise auf den totalen Polynomgrad („P“) oder den maximalen Polynomgrad in eine Richtung („Q“) beziehen.
V_h^p	Raum der stückweisen Polynome (auf Ω_k) vom Grad maximal p ausgestattet mit dem Skalarprodukt $(\cdot, \cdot)_{L^2}$
$(\cdot, \cdot)_{L^2}$	L^2 -Skalarprodukt des Raumes V_h^p : $(v^i, v^j)_{L^2} = \int_\Omega v^i(\mathbf{x})v^j(\mathbf{x}) dx$
u, v	Stückweise Polynome auf Ω : $u, v \in V_h^p$
n_{DOF}	Anzahl der Basisvektoren in jedem Teilgebiet Ω_k
$\varphi_k^i(\mathbf{x})$	Ansatz-Funktionen, Basis des Raumes V_h^p ; dabei ist $\varphi_k^i(\mathbf{x}) = 0$ für $\mathbf{x} \notin \Omega_k$
$\hat{\varphi}_k^j(\mathbf{x})$	Test-Funktionen, analog definiert zu φ_k^i
i, j	Indizes der Basispolynome: $i, j = 1, \dots, n_{DOF}$

$\int_{\Omega_k} \cdot d\mathbf{x}$	Volumenintegral über das Gebiet Ω_k
$\partial\Omega_k$	Rand des Gebietes Ω_k ; d.h. die Seiten von Ω_k zu umliegenden Ω_l
n_α	Äußerer Normaleneinheitsvektor bei der Integration über $\partial\Omega_k$
w^-, w^+	Innerer und äußerer Grenzwert von w am Rand des Gebietes Ω_k
$\int_{\partial\Omega_k} \cdot d\sigma$	Oberflächenintegral über den Rand des Gebietes Ω_k
$g(w^-, w^+, n)$	Numerischer Fluss: $g : (w^-, w^+, n) \rightarrow g(w^-, w^+, n) \in \mathbb{R}^3$
u_k^i	Einzelner Freiheitsgrad in Element Ω_k , Koeffizient des Basispolynoms φ_k^i
\mathbf{u}_k	Vektor aller Freiheitsgrade in Element Ω_k
\mathbf{M}_k	Masse-Matrix des Elements Ω_k : $M_k^{j,i} = (\varphi_k^i, \hat{\varphi}_k^j)_{L^2}$
$U(t)$	Vektor aller zeitabhängigen Koeffizienten u_k^i des semi-diskreten DG-Verfahrens
U_0	Anfangsdaten des semi-diskreten DG-Verfahrens
$R(U)$	Diskreter Operator der DG-Diskretisierung im Raum: $U'(t) = R(U(t))$
U^n	Numerische Lösung aller Koeffizienten u_k^i zum Zeitpunkt t^n
Δt	Größe eines Zeitschrittes: $\Delta t = t^{n+1} - t^n$
$\mathcal{P}_k(\mathbf{u}_k(t^n))$	Operator des lokalen Prädiktors
$\bar{R}_k(\mathbf{v}_k)$	Diskreter Operator des lokalen DG-Prädiktors im Raum: $\mathbf{v}'_k = \bar{R}_k(\mathbf{v}_k)$
$\mathbf{q}_k(t^n + \theta\Delta t)$	Approximative Entwicklung der Koeffizienten \mathbf{u}_k in der Zeit: $\mathbf{q}_k : \theta \rightarrow \mathbf{q}_k(t^n + \theta\Delta t) \in \mathbb{R}^{n_{DOF}}$
Q	Vektor aller lokalen Vorhersagen \mathbf{q}_k
$\mathcal{C}(Q)$	Operator des globalen Korrektors
$\delta_{i,j}$	Kronecker-Delta: $\delta_{i,j} = 1$ für $i = j$ und andernfalls $\delta_{i,j} = 0$
$\hat{\varphi}_E^j$	Duale Legendre-Polynome im eindimensionalen Referenzintervall
$\varphi_E^i(x)$	Legendre-Polynome im eindimensionalen Referenzintervall $\Omega_E = [-1, 1]$
$f_{k,\alpha}^i$	Physikalischer Fluss aus Element Ω_k in Polynomialbasis
$g_{k,l}^i$	Numerische Fluss der Seite $\partial\Omega_k \cap \partial\Omega_l$ in Polynomialbasis
$\mathbf{S}_{k,\alpha}$	Steifigkeitsmatrix zur Projektion des physikalischen Flusses aus Element Ω_k
$\mathbf{F}_{k,l}$	Matrix zur Projektion des numerischen Flusses der Seite $\partial\Omega_k \cap \partial\Omega_l$

1 Einleitung

Die computergestützte Analyse komplexer physikalischer Vorgänge ist wesentlich für einen großen Bereich ingenieurstechnischer Probleme. Die zugrunde liegenden physikalischen Gesetze lassen sich in vielen Fällen mit Hilfe partieller Differentialgleichungen formulieren, die die Erhaltung physikalischer Größen in Raum und Zeit beschreiben:

$$\partial_t \mathbf{w} + \partial_{x_\alpha} \mathbf{f}_\alpha = \mathbf{s}$$

Dabei bezeichnet $\mathbf{w}(\mathbf{x}, t)$ den Vektor der Erhaltungsgrößen und \mathbf{f}_α den Vektor der Flüsse dieser Größen in die Raumrichtungen $\alpha = 1, 2, 3$. Mit \mathbf{s} können Quellen und Senken der Erhaltungsgrößen berücksichtigt werden.

Hier relevante Beispiele für solche Systeme von Erhaltungsgleichungen sind die linearen Maxwell-Gleichungen zur Simulation elektrischer Felder und Ströme, sowie die Euler-Gleichungen zur Simulation kompressibler Strömungen unter Vernachlässigung von Reibungseffekten und Wärmetransport.

In dieser Arbeit werden der Einfachheit halber zumeist skalare Erhaltungsgleichungen betrachtet, die Darstellungen lassen sich jedoch auf Systeme übertragen.

Zur Simulation benötigt man eine Diskretisierung dieser Gleichungen in Raum und Zeit. Dazu kann man die partiellen Differentialgleichung zunächst im Raum diskretisieren und erhält so ein System gewöhnlicher Differentialgleichungen in der Zeit (Method of Lines). Dieses kann dann mit Hilfe von Verfahren für gewöhnliche Differentialgleichungen in der Zeit integriert werden, man muss jedoch das Zusammenspiel der Diskretisierungen in Raum- und Zeit berücksichtigen, um insgesamt ein stabiles Verfahren zu erhalten[1]. Gebräuchliche Ansätze zur Diskretisierung im Raum stellen Finite-Volumen-Methoden (FV) und Finite-Elemente-Methoden (FE) dar. Finite-Volumen-Methoden haben eine diskrete Erhaltungseigenschaft, was für Finite-Elemente-Methoden nicht notwendigerweise gilt. Ideen beider Ansätze vereinen Discontinuous-Galerkin-Methoden (DG); sie weisen ebenfalls eine diskrete Erhaltungseigenschaft auf und erlauben die Verwendung höherer Ordnung bei kleinem Abhängigkeitsbereich, was für die effiziente parallele Berechnung auf aktuellen Supercomputern von Vorteil ist. Des Weiteren ist es gegebenenfalls schwierig, geeignete FV-Stencils höherer Ordnung in komplexen Geometrien zu definieren. DG-Methoden sind jedoch weniger gebräuchlich und benötigen zum Teil zusätzliche Stabilisierung, was einen aktuellen Forschungsgegenstand der Numerik darstellt[7]. Da implizite Diskretisierungen in der Zeit hier in vielen Fällen zu großen, aufwendig zu lösenden Gleichungssystemen führen, werden zumeist explizite Verfahren wie Runge-Kutta-Methoden oder Mehrschrittverfahren (BDF) eingesetzt.

In dieser Arbeit wird ein spezielles Discontinuous-Galerkin Verfahren betrachtet, das am Lehrstuhl für *Angewandtes Supercomputing im Maschinenbau* an der RWTH-Aachen entwickelt wurde[6]. Für dieses wird die Implementierung einer Methode zur Zeitdiskretisierung vorgestellt, die auf einem Prädiktor-Korrektor-Ansatz beruht und lokale Zeitschritte ermöglicht[3]. Der Schwerpunkt liegt dabei auf einer möglichst effizienten Berechnung der benötigten DG-Operationen.

Insgesamt wird verdeutlicht, wie sich eine Prädiktor-Korrektor-Zeitdiskretisierung in ein bestehendes Discontinuous-Galerkin-Framework integrieren lässt.

2 RK-DG-Verfahren

Im folgenden werden kurz klassische RKDG-Methoden skizziert: Eine Discontinuous-Galerkin Diskretisierung im Raum führt zu einem System gewöhnlicher Differentialgleichungen in der Zeit. Dieses wird anschließend mit einem Runge-Kutta Verfahren diskretisiert.

Eine ausführlichere Darstellung von Discontinuous-Galerkin Verfahren findet sich beispielsweise in [7].

2.1 Discontinuous-Galerkin Diskretisierung im Raum

Betrachte die Differentialgleichung:

$$\begin{aligned} \partial_t w + \partial_{x_\alpha} f_\alpha(w) &= 0 & \forall (\mathbf{x}, t) \in \Omega \times \mathbb{R}^+, \\ w(\mathbf{x}, 0) &= w_0(\mathbf{x}) & \forall \mathbf{x} \in \Omega \end{aligned} \quad (1)$$

Mit $\Omega = \mathbb{R}^3$ und $f : w \rightarrow \mathbf{f}(w) \in \mathbb{R}^3$. Des weiteren sei Ω unterteilt in nicht überlappende Gebiete Ω_k und V_h^p der Raum der stückweisen Polynome vom Grad maximal p .

Dieser sei ausgestattet mit einer Basis $\hat{\varphi}_k^i(\mathbf{x})$, die auf jedem Teilgebiet Ω_k aus $i = 1, \dots, n_{DOF}$ transformierten Basispolynomen des Einheitselements besteht. Ein beliebiges $v \in V_h^p$ lässt sich also mit den Koeffizienten $v_k^j \in \mathbb{R}$ und der Basis $\hat{\varphi}_k^j$, $k \in \mathbb{N}$, $j = 1, \dots, n_{DOF}$ darstellen als:

$$v = v_k^j \hat{\varphi}_k^j$$

Man fordert nun, dass das L^2 -Skalarprodukt der Differentialgleichung mit beliebigen stückweisen Polynomen Null ist:

$$\begin{aligned} (w_t + \partial_{x_\alpha} f_\alpha(w), v)_{L^2} &= 0 & \forall v \in V_h^p \\ \Leftrightarrow (w_t + \partial_{x_\alpha} f_\alpha(w), \hat{\varphi}_k^j)_{L^2} &= 0 & \forall k, j \end{aligned}$$

Dies kann man mittels partieller Integration umformen zu:

$$\begin{aligned} 0 &= \int_{\Omega} (w_t + (\partial_{x_\alpha} f_\alpha(w))) \hat{\varphi}_k^j d\mathbf{x} \\ &= \int_{\Omega_k} w_t \hat{\varphi}_k^j d\mathbf{x} + \int_{\Omega_k} (\partial_{x_\alpha} f_\alpha(w)) \hat{\varphi}_k^j d\mathbf{x} \\ &= \int_{\Omega_k} w_t \hat{\varphi}_k^j d\mathbf{x} - \int_{\Omega_k} f_\alpha(w) \partial_{x_\alpha} \hat{\varphi}_k^j d\mathbf{x} + \int_{\partial\Omega_k} f_\alpha(w^-) \hat{\varphi}_k^j n_\alpha d\sigma \\ &= \frac{d}{dt} \int_{\Omega_k} w \hat{\varphi}_k^j d\mathbf{x} - \int_{\Omega_k} f_\alpha(w) \partial_{x_\alpha} \hat{\varphi}_k^j d\mathbf{x} + \int_{\partial\Omega_k} f_\alpha(w^-) \hat{\varphi}_k^j n_\alpha d\sigma \end{aligned}$$

Der letzte Term repräsentiert den Fluss über die Kanten des Gebietes Ω_k mit dem äußeren Normaleneinheitsvektor n . In der obigen Form existiert jedoch keinerlei diskrete Erhaltungseigenschaft, da für jedes Element das Flussintegral über den inneren Grenzwert w^- bestimmt wird.

Setzt man nun hier stattdessen einen numerischen Fluss $g(w^-, w^+, n)$ mit äußerem Grenzwert w^+ (d.h. innere Grenzwerte aus benachbarten Zellen) ein, erhält man:

$$\frac{d}{dt} \int_{\Omega_k} w \hat{\varphi}_k^j d\mathbf{x} = \int_{\Omega_k} f_\alpha(w) \partial_{x_\alpha} \hat{\varphi}_k^j d\mathbf{x} - \int_{\partial\Omega_k} g(w^-, w^+, n) \hat{\varphi}_k^j d\sigma \quad \forall k, j$$

Die Lösung $w(\mathbf{x}, t)$ kann man im Raum mit einem stückweisen Polynom $u(\cdot, t) \in V_h^p$ in einer Basis φ_k^i approximieren:

$$w(\mathbf{x}, t) \approx u(\mathbf{x}, t) = u_k^i(t) \varphi_k^i(\mathbf{x})$$

Damit ergibt sich die folgende semi-diskrete Darstellung eines DG-Verfahrens:

$$\begin{aligned} \left(\int_{\Omega_k} \varphi_k^i \hat{\varphi}_k^j d\mathbf{x} \right) (u_k^i)'(t) &= \int_{\Omega_k} f_\alpha(u_k^i \varphi_k^i) \partial_{x_\alpha} \hat{\varphi}_k^j d\mathbf{x} \\ &\quad - \sum_l \int_{\partial\Omega_k \cap \partial\Omega_l} g(u_k^i \varphi_k^i, u_l^i \varphi_l^i, n) \hat{\varphi}_k^j d\sigma \quad \forall k, j \end{aligned} \quad (2)$$

Hier bezeichnet Ω_l ein zu Gebiet Ω_k benachbartes Gebiet.

Das Integral auf der linken Seite führt zu einer invertierbaren Masse-Matrix \mathbf{M}_k , die nur von der Wahl der Basen der Polynomräume der Ansatz- und Testfunktionen $(\varphi_k^i, \hat{\varphi}_k^j)$ abhängt:

$$\mathbf{M}_k := \left(\int_{\Omega_k} \varphi_k^i \hat{\varphi}_k^j d\mathbf{x} \right)_{j,i} \quad (3)$$

Multiplikation mit der Inversen der Masse-Matrix in jedem Gebiet führt zu einer expliziten Darstellung von \mathbf{u}'_k . Man benötigt weiterhin Quadraturformeln zur numerischen Auswertung der Integrale und erhält so ein System gewöhnlicher Differentialgleichungen der Form:

$$\begin{aligned} U'(t) &= R(U(t)) \\ U(0) &= U_0 \end{aligned} \quad (4)$$

Hier bezeichnet U den Vektor aller Koeffizienten u_k^i , U_0 die Anfangsdaten in der gewählten Basis φ_k^i und $R(\cdot)$ den diskreten Operator, der sich aus der obigen Discontinuous-Galerkin-Diskretisierung im Raum ergibt.

2.2 Runge-Kutta Diskretisierung in der Zeit

Dieses System gewöhnlicher Differentialgleichungen kann mit einem Runge-Kutta-Verfahren diskretisiert werden. Ein Beispiel ist das klassische, explizite vierstufige Runge-Kutta-

Verfahren (siehe beispielsweise [2]):

$$\begin{aligned}U^{n+1} &= U^n + \frac{\Delta t}{6} (k_1 + 2k_2 + 3k_3 + k_4) && \text{mit} \\k_1 &= R(U^n) \\k_2 &= R(U^n + 0.5\Delta t k_1) \\k_3 &= R(U^n + 0.5\Delta t k_2) \\k_4 &= R(U^n + \Delta t k_3)\end{aligned}\tag{5}$$

3 Prädiktor-Korrektor-DG Verfahren

Das Verfahren aus [3] besteht aus zwei Schritten:

1. Einem *lokalen Prädiktor*, der für jedes Element Ω_k eine Vorhersage für die Entwicklung in der Zeit bestimmt:

$$\mathbf{q}_k = \mathcal{P}_k(\mathbf{u}_k(t^n)) \quad \forall k \quad (6)$$

Dabei bezeichnet \mathcal{P}_k einen approximativen Evolutions-Operator und $\mathbf{q}_k(t^n + \theta\Delta t)$, $\theta \in [0, 1)$ sind zeitabhängige Koeffizienten; im Gegensatz dazu gibt $\mathbf{u}_k(t^n)$ nur den Zustand im Element Ω_k zum Zeitpunkt t^n an.

Insbesondere ist der Operator \mathcal{P}_k *lokal* in dem Sinne, dass er nur den Zustandsvektor des Elements Ω_k berücksichtigt und nicht die der umliegenden Elemente.

2. Einem *globalen Korrektor*, der aus den lokalen Vorhersagen aller Elemente den Zustand zum Zeitpunkt t^{n+1} ermittelt:

$$U^{n+1} = U^n + \mathcal{C}(Q) \quad (7)$$

Hier ist Q der Vektor aller lokalen Vorhersagen \mathbf{q}_k und $\mathcal{C}(Q)$ der Korrektur-Operator, der den Zustand zum nächsten Zeitschritt global bestimmt.

3.1 Lokaler CERK Prädiktor

Betrachte hier zunächst erneut die partielle Differentialgleichung zum Zeitpunkt t^n :

$$\begin{aligned} w_t + \partial_{x_\alpha} f_\alpha(w) &= 0, & \forall (\mathbf{x}, t) \in \Omega_k \times [t^n, t^{n+1}) \\ w(\mathbf{x}, t^n) &= u_k^i(t^n) \varphi_k^i & \forall \mathbf{x} \in \Omega_k \end{aligned}$$

Approximiert man $w(\mathbf{x}, t^n + \theta\Delta t)$ für das Gebiet Ω_k wie oben in Polynomialbasis als

$$w_k(\mathbf{x}, t) \approx v_k^i(t^n + \theta\Delta t) \varphi_k^i(\mathbf{x})$$

dann erhält man:

$$\begin{aligned} \partial_t v_k^i \varphi_k^i &= -\partial_{x_\alpha} f_\alpha(v_k^i \varphi_k^i) & \forall k \\ \mathbf{v}_k(t^n) &= \mathbf{u}_k(t^n) \end{aligned}$$

Multiplikation mit den Testfunktionen $\hat{\varphi}_k$ und Integration über Ω_k ergibt:

$$\mathbf{M}_k \mathbf{v}'_k = - \int_{\Omega_k} \partial_{x_\alpha} f_\alpha(v_k^i \varphi_k^i) \hat{\varphi}_k d\mathbf{x} \quad (8)$$

Mit einer Quadraturformel für das Integral (und für nichtlineare f gegebenenfalls einer geeigneten Approximation des Terms $\partial_{x_\alpha} f_\alpha$) erhält man für jedes Gebiet Ω_k ein System gewöhnlicher Differentialgleichungen der Form:

$$\begin{aligned} \mathbf{v}'_k(t^n + \theta\Delta t) &= \bar{R}_k(\mathbf{v}_k(t^n + \theta\Delta t)) \\ \mathbf{v}_k(t^n) &= \mathbf{u}_k(t^n) \end{aligned} \quad (9)$$

Im Unterschied zu (4) hängt in diesem Fall die approximierte Lösung in jedem Gebiet Ω_k nicht von benachbarten Gebieten Ω_l ab.

Mit einem vierstufigen CERK-Verfahren (dritter Ordnung) aus [5] erhält man für den lokalen Prädiktor \mathcal{P}_k :

$$\begin{aligned}
\mathbf{q}_k(t^n + \theta\Delta t) &= \mathbf{u}_k(t^n) + \Delta t (b_1(\theta)k_1 + b_2(\theta)k_2 + b_3(\theta)k_3) && \text{mit} \\
k_1 &= \bar{R}_k(\mathbf{u}_k(t^n)) \\
k_2 &= \bar{R}_k\left(\mathbf{u}_k(t^n) + \frac{12}{23}\Delta t k_1\right) \\
k_3 &= \bar{R}_k\left(\mathbf{u}_k(t^n) - \frac{68}{375}\Delta t k_1 + \frac{368}{375}\Delta t k_2\right) \\
k_4 &= \bar{R}_k\left(\mathbf{u}_k(t^n) + \frac{31}{144}\Delta t k_1 + \frac{529}{1152}\Delta t k_2 + \frac{125}{384}\Delta t k_3\right) && \text{und} \\
b_1(\theta) &= \frac{41}{72}\theta^3 - \frac{65}{48}\theta^2 + \theta \\
b_2(\theta) &= -\frac{529}{576}\theta^3 + \frac{529}{384}\theta^2 \\
b_3(\theta) &= -\frac{125}{192}\theta^3 + \frac{125}{128}\theta^2 \\
b_4(\theta) &= \theta^3 - \theta^2 && (10)
\end{aligned}$$

3.2 Globaler Korrektor

Den globalen Korrektur-Schritt erhält man aus (2) durch Integration über $\int_{t^n}^{t^{n+1}} \cdot dt$:

$$\begin{aligned}
\mathbf{M}_k \mathbf{u}_k(t^{n+1}) &= \mathbf{M}_k \mathbf{u}_k(t^n) + \int_{t^n}^{t^{n+1}} \int_{\Omega_k} f_\alpha(q_k^i \varphi_k^i) \partial_{x_\alpha} \hat{\varphi}_k d\mathbf{x} dt \\
&\quad - \int_{t^n}^{t^{n+1}} \sum_l \int_{\partial\Omega_k \cap \partial\Omega_l} g(q_k^i \varphi_k^i, q_l^i \varphi_l^i, n) \hat{\varphi}_k d\sigma dt \quad \forall k && (11)
\end{aligned}$$

Zur Anwendung des globalen Korrektur-Schrittes benötigt man Quadraturformeln zur numerischen Auswertung der Integrale. Hier kann man für die Integrale im Raum die selbe Diskretisierung wählen wie zuvor bei den RKDG-Methoden.

Für die Integrale in der Zeit bietet sich eine Gauß-Legendre-Quadratur an (siehe beispielsweise [2]); verwendet man den oben vorgestellten Prädiktor, genügt eine Auswertung der Volumen- und Flächen-Integrale an zwei Stützstellen in der Zeit um insgesamt ein Verfahren dritter Ordnung in der Zeit zu erhalten:

$$\begin{aligned}
U^{n+1} &= U^n + \Delta t (b_1 R(Q(t^n + \theta_1 \Delta t)) + b_2 R(Q(t^n + \theta_2 \Delta t))) && \text{mit} \\
b_{1,2} &= \frac{1}{2} \\
\theta_{1,2} &= \frac{\sqrt{3} \pm 1}{2\sqrt{3}} && (12)
\end{aligned}$$

Hierbei bezeichnet $R(\cdot)$ den diskreten Operator aus (4). Insbesondere werden also in der obigen Formulierung des Korrektors nur Operationen benötigt, die auch im RKDG-Verfahren auftreten.

4 Quadratur-freies Discontinuous-Galerkin Verfahren für kubische Gitter

Im folgenden wird kurz ein spezielles Discontinuous-Galerkin Verfahren für kubische Gitter vorgestellt, das im Rahmen des Frameworks APES aus [6] entwickelt wurde. Die Beschreibung hier basiert auf [4], dort wird das Verfahren jedoch noch deutlich ausführlicher dargestellt.

4.1 Legendre-Polynome im eindimensionalen Fall

Die grundlegende Idee besteht darin, spezielle Basen für die Ansatz- und Testfunktionen zu wählen. Im eindimensionalen Fall $\Omega = \mathbb{R}$ bilden die Legendre-Polynome eine orthogonale Basis bezüglich des L^2 -Skalarproduktes. Diese kann man im Referenzintervall $\Omega_E = [-1, 1]$ rekursiv definieren als

$$\begin{aligned}\varphi_E^1(x) &:= 1, \\ \varphi_E^2(x) &:= x, \\ \varphi_E^{i+1}(x) &:= \frac{2i-1}{i}x\varphi_E^i(x) - \frac{i-1}{i}\varphi_E^{i-1}(x).\end{aligned}\tag{13}$$

Für diese gilt insbesondere:

$$(\varphi_E^i, \varphi_E^j)_{L^2} = \frac{2}{2i-1}\delta_{i,j}$$

Des weiteren erhält man für Ableitungen der Legendre-Polynome:

$$\partial_x \varphi_E^i(x) = \frac{1}{2i-1} \partial_x (\varphi_E^{i+1}(x) - \varphi_E^{i-1}(x))$$

Und an den Rändern des Referenzintervalls ergibt sich:

$$\varphi_E^i(-1) = (-1)^{i-1} \quad \wedge \quad \varphi_E^i(1) = 1$$

Zur effizienten Umsetzung des DG-Verfahrens ist es notwendig, die einzelnen in (2) auftretenden Terme sowie die Anwendung der Inversen der Masse-Matrix mit möglichst geringem Aufwand zu bestimmen.

Wählt man als Basis für die Testfunktionen ebenfalls die Legendre-Polynome, ist die Masse-Matrix eine Diagonalmatrix und somit trivial invertierbar. Da jedoch im Flussintegral die Ableitung $\partial_x \hat{\varphi}_k^j(x)$ auftritt, ergibt sich für diesen Term keine ähnlich einfache Darstellung.

Eine Alternative liegt darin, für die Testfunktionen folgende duale Legendre-Basis zu verwenden:

$$\begin{aligned}\hat{\varphi}_E^1(x) &:= 1, \\ \hat{\varphi}_E^2(x) &:= x, \\ \hat{\varphi}_E^j(x) &:= \varphi_E^j(x) - \varphi_E^{j-2}(x)\end{aligned}\tag{14}$$

Damit ergeben sich folgende drei Eigenschaften, die die Grundlage des hier betrachteten DG-Verfahrens bilden:

$$(\varphi_E^i, \hat{\varphi}_E^j)_{L^2} = \frac{2}{2i-1} (\delta_{i,j} - \delta_{i,j-2}) \quad (15)$$

$$(\varphi_E^i, \partial_x \hat{\varphi}_E^j)_{L^2} = 2\delta_{i,j-2} \quad (16)$$

$$\hat{\varphi}_E^j|_{\partial\Omega_E} = 0 \quad \text{für } j > 2 \quad (17)$$

Aus der Eigenschaft (15) folgt, dass die Masse-Matrix im eindimensionalen Fall untere Dreiecksform hat und nur die Diagonale und zweite untere Nebendiagonale besetzt sind. Die Multiplikation der inversen Masse-Matrix mit einem Vektor ist also mit wenig Aufwand mittels Rückwärts-Einsetzen möglich.

Die Eigenschaft (16) führt dazu, dass das Integral des physikalischen Flusses effizient berechnet werden kann. Dies wird im nächsten Abschnitt noch genauer ausgeführt.

Und die Eigenschaft (17) verringert den Aufwand zur Berechnung des Oberflächenintegrals des numerischen Flusses, da dieses offensichtlich nur für die Fälle $j = 1, 2$ ausgewertet werden muss.

4.2 Quadratur-freie Darstellung mittels Projektionsmatrizen

Die Ergebnisse des vorigen Abschnittes zum eindimensionalen Fall lassen sich im Allgemeinen leider nicht direkt auf den dreidimensionalen Fall $\Omega = \mathbb{R}^3$ übertragen. Für kubische Elemente kann man jedoch eine Basis aus den Tensor-Produkten der eindimensionalen Basis-Polynome bilden:

$$\begin{aligned} \vec{\varphi}_k^i(\mathbf{x}) &= \prod_{\alpha=1}^3 \varphi_k^{i_\alpha}(x_\alpha), \\ \vec{\hat{\varphi}}_k^j(\mathbf{x}) &= \prod_{\alpha=1}^3 \hat{\varphi}_k^{j_\alpha}(x_\alpha) \end{aligned} \quad (18)$$

Betrachtet man nun erneut die semi-diskrete Darstellung eines DG-Verfahrens (2) und approximiert die Flüsse ebenfalls in Polynomialbasis, dann erhält man:

$$\begin{aligned} (\varphi_k^i, \hat{\varphi}_k^j)_{L^2} (u_k^i)'(t) &= (\varphi_k^i, \partial_{x_\alpha} \hat{\varphi}_k^j)_{L^2} f_{k,\alpha}^i \\ &\quad - \sum_l \left(\int_{\partial\Omega_k \cap \partial\Omega_l} \varphi_k^i \hat{\varphi}_k^j d\sigma \right) g_{k,l}^i \quad \forall k, j \end{aligned} \quad (19)$$

Dabei seien $f_{k,\alpha}^i$ und $g_{k,l}^i$ geeignete Approximationen:

$$\begin{aligned} f_{k,\alpha}^i \varphi_k^i &\approx f_\alpha(u_k^i \varphi_k^i) && \text{auf } \Omega_k \\ g_{k,l}^i \varphi_k^i &\approx g(u_k^i \varphi_k^i, u_l^i \varphi_l^i, n) && \text{auf } \partial\Omega_k \cap \partial\Omega_l \end{aligned} \quad (20)$$

Diese Darstellung des numerischen Flusses ergibt sich insbesondere aufgrund der kubischen Elemente und der speziellen Wahl der Polynomialbasis. Auf jeder Seite $\partial\Omega_k \cap \partial\Omega_l$

kann man auch eine geeignete zweidimensionale Polynom-Basis über das Tensor-Produkt von zwei eindimensionalen Legendre-Polynomen definieren. Für die folgende Herleitung eines lokalen Prädiktors ist dies aber nicht relevant.

Lineare Flüsse f, g können hier ohne zusätzlichen Aufwand exakt dargestellt werden als

$$\begin{aligned} f_{k,\alpha}^i &= f_\alpha(u_k^i), \\ g_{k,l}^i &= g(u_k^i, -u_l^i, n) \end{aligned} \quad \text{für } \partial\Omega_k \cap \partial\Omega_l. \quad (21)$$

Insgesamt ergibt sich die folgende lineare Darstellung eines Quadratur-freien DG-Verfahrens:

$$\mathbf{M}_k \mathbf{u}'_k = \mathbf{S}_{k,\alpha} \mathbf{f}_{k,\alpha} - \mathbf{F}_{k,l} \mathbf{g}_{k,l} \quad \forall k \quad (22)$$

Ähnlich zu der bereits bekannten Masse-Matrix \mathbf{M}_k werden hier die Steifigkeitsmatrizen $\mathbf{S}_{k,\alpha}$ und Matrizen zur Projektion der numerischen Flüsse $\mathbf{F}_{k,l}$ definiert:

$$\begin{aligned} \mathbf{M}_k &= \left(\int_{\Omega_k} \varphi_k^i \hat{\varphi}_k^j d\mathbf{x} \right)_{j,i} \\ \mathbf{S}_{k,\alpha} &= \left(\int_{\Omega_k} \varphi_k^i \partial_{x_\alpha} \hat{\varphi}_k^j d\mathbf{x} \right)_{j,i} \\ \mathbf{F}_{k,l} &= \left(\int_{\partial\Omega_k \cap \partial\Omega_l} \varphi_k^i \hat{\varphi}_k^j d\sigma \right)_{j,i} \end{aligned} \quad (23)$$

In der speziellen Basis hier ergibt sich insbesondere:

$$\begin{aligned} M_k^{\vec{j},\vec{i}} &= \left(\frac{\Delta x}{2} \right)^3 \prod_{\alpha=1}^3 \frac{2(\delta_{i_\alpha, j_\alpha} - \delta_{i_\alpha, j_\alpha - 2})}{2i_\alpha - 1} \\ S_{k,\alpha}^{\vec{j},\vec{i}} &= \left(\frac{\Delta x}{2} \right)^2 2\delta_{i_\alpha, j_\alpha - 2} \prod_{\beta \neq \alpha} \frac{2(\delta_{i_\beta, j_\beta} - \delta_{i_\beta, j_\beta - 2})}{2i_\beta - 1} \end{aligned} \quad (24)$$

4.3 Herleitung eines geeigneten lokalen Prädiktors

Betrachte nun die rechte Seite des lokalen Prädiktors aus (8) in linearer, Quadratur-freier Darstellung:

$$\mathbf{M}_k \mathbf{v}'_k = \underbrace{\left(\int_{\Omega_k} \partial_{x_\alpha} \varphi_k^i \hat{\varphi}_k^j d\mathbf{x} \right)}_{\hat{\mathbf{S}}_k^{j,i}} \mathbf{f}_{k,\alpha}^i$$

Da hier jedoch ein Term mit einer partielle Ableitung der Ansatz-Polynome $\partial_{x_\alpha} \varphi_k^i$ auftritt, ist die Matrix $\hat{\mathbf{S}}_k$ nicht dünnbesetzt. Zur Auswertung des lokalen Prädiktors ergibt sich so ein erheblicher Aufwand.

Man kann jedoch im lokalen Prädiktor auch andere Polynombasen wählen. Allgemein dargestellt in Polynombasen ψ_k^i und $\hat{\psi}_k^j$ ergibt sich:

$$\begin{aligned}\bar{\mathbf{M}}_k \bar{\mathbf{v}}_k'(t) &= \bar{\mathbf{S}}_{k,\alpha} \bar{\mathbf{f}}_{k,\alpha}(\bar{\mathbf{v}}_k(t)) && \text{mit} \\ \psi_k^i \bar{v}_k^i &= \varphi_k^i v_k^i\end{aligned}\tag{25}$$

Dabei gilt:

$$\begin{aligned}\psi_k^i \bar{f}_{k,\alpha}^i &= \varphi_k^i f_{k,\alpha}^i \\ \bar{\mathbf{M}}_k &:= \left(\psi_k^i, \hat{\psi}_k^j \right)_{L^2} \\ \bar{\mathbf{S}}_{k,\alpha} &:= \left(\partial_{x_\alpha} \psi_k^i, \hat{\psi}_k^j \right)_{L^2}\end{aligned}\tag{26}$$

Als geeignete Wahl der Polynombasen erscheint hier:

$$\begin{aligned}\psi_k^i &:= \hat{\varphi}_k^i, \\ \hat{\psi}_k^j &:= \varphi_k^j\end{aligned}$$

Damit erhält man:

$$\begin{aligned}\bar{\mathbf{M}}_k &= \left(\int_{\Omega_k} \hat{\varphi}_k^i \varphi_k^j d\mathbf{x} \right)_{j,i} = \mathbf{M}_k^T, \\ \bar{\mathbf{S}}_{k,\alpha} &= \left(\int_{\Omega_k} \partial_{x_\alpha} \hat{\varphi}_k^i \varphi_k^j d\mathbf{x} \right)_{j,i} = \mathbf{S}_{k,\alpha}^T\end{aligned}$$

Zur Transformation von der Basis φ_k^i in die duale Basis $\hat{\varphi}_k^i$ kann man folgende Eigenschaft ausnutzen:

$$\begin{aligned}\hat{\varphi}_k^i \bar{v}_k^i &= \varphi_k^i v_k^i && \forall k \\ \Leftrightarrow \left(\hat{\varphi}_k^i, \varphi_k^j \right)_{L^2} \bar{v}_k^i &= \left(\varphi_k^i, \varphi_k^j \right)_{L^2} v_k^i && \forall k, j \\ \Leftrightarrow \bar{\mathbf{M}}_k \bar{\mathbf{v}}_k &= \mathbf{I}_k \mathbf{v}_k && \forall k\end{aligned}$$

Mit der Diagonalmatrix

$$\mathbf{I}_k := \left(\int_{\Omega_k} \varphi_k^i \varphi_k^j \right)_{j,i}\tag{27}$$

Für die hier definierte Basis ergibt sich insbesondere:

$$I_k^{\vec{j},\vec{i}} = \left(\frac{\Delta x}{2} \right)^3 \prod_{\alpha=1}^3 \frac{2\delta_{i_\alpha, j_\alpha}}{2i_\alpha - 1}\tag{28}$$

Skaliert man nun noch die transponierten Masse- und Steifigkeitsmatrizen mit der Inversen von \mathbf{I}_k , so erhält man folgende Darstellung der lokalen Probleme des Prädiktors:

$$\begin{aligned} \mathbf{v}'_k &= \tilde{\mathbf{S}}_{k,\alpha} \bar{\mathbf{f}}_{k,\alpha} && \text{für } t \in [t^n, t^{n+1}), \\ \tilde{\mathbf{M}}_k \bar{\mathbf{f}}_{k,\alpha} &= \mathbf{f}_{k,\alpha}(\mathbf{v}_k) && \text{und} \\ \mathbf{v}_k(t^n) &= \mathbf{u}_k(t^n) \end{aligned} \tag{29}$$

Für die skalierten transponierten Matrizen ergibt sich:

$$\begin{aligned} \tilde{M}_k^{\vec{j},\vec{i}} &= \prod_{\alpha=1}^3 (\delta_{i_\alpha, j_\alpha} - \delta_{i_\alpha-2, j_\alpha}) \\ \tilde{S}_{k,\alpha}^{\vec{j},\vec{i}} &= \frac{2}{\Delta x} (2j_\alpha - 1) \delta_{i_\alpha-1, j_\alpha} \prod_{\beta \neq \alpha} (\delta_{i_\beta, j_\beta} - \delta_{i_\beta-2, j_\beta}) \end{aligned} \tag{30}$$

Im linearen Fall gilt zusätzlich:

$$\begin{aligned} \bar{\mathbf{f}}_{k,\alpha} &= f_\alpha(\bar{\mathbf{v}}_k), && \text{mit} \\ \tilde{\mathbf{M}}_k \bar{\mathbf{v}}_k &= \mathbf{v}_k \end{aligned} \tag{31}$$

Die transponierte und skalierte Masse-Matrix $\tilde{\mathbf{M}}_k$ hat offensichtlich wieder Dreiecksform, sodass sich ihre Inverse auch effizient auf einen Vektor anwenden lässt.

5 Lokale Zeitschritte

Das Ziel des zuvor vorgestellten Verfahrens soll im folgenden Abschnitt veranschaulicht werden: die Verwendung unterschiedlich großer Zeitschritte in verschiedenen Teilen des Rechengebietes.

Hier werden ausschließlich Verfahren mit expliziter Zeitdiskretisierung betrachtet; ein Vergleich mit impliziten Verfahren ist im Rahmen dieser Arbeit nicht möglich. Bei der Wahl des Zeitschrittes muss in diesem Fall die CFL-Bedingung berücksichtigt werden, die das Stabilitätslimit expliziter Zeitdiskretisierungen angibt[1].

Verwendet man globale Zeitschritte wie das zu Beginn vorgestellte RK-DG Verfahren, richtet sich der maximal mögliche Zeitschritt im gesamten Rechengebiet nach dem kleinsten Wert des Quotienten aus charakteristischer Länge und Geschwindigkeit in einem Element.

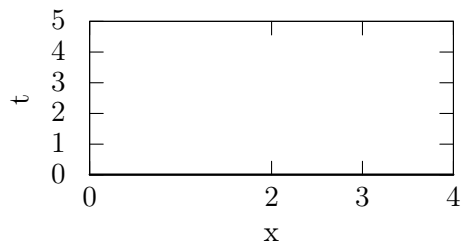
In vielen Fällen unterscheiden sich die Gittergröße und die charakteristische Geschwindigkeiten der Erhaltungsgleichung im Rechengebiet jedoch um mehrere Größenordnungen. Das bedeutet, dass in einem großen Teil des Rechengebietes – lokal gesehen – ein zu kleiner Zeitschritt verwendet wird. Hier bieten Verfahren mit lokalen Zeitschritten die Möglichkeit, den Zeitschritt entsprechend anzupassen und damit bei vergleichbarer Genauigkeit den Rechenaufwand deutlich zu reduzieren.

5.1 Prädiktor-Korrektor – Ansatz

Die Formulierung der Zeitintegration als Prädiktor- und Korrektor-Schritt ermöglicht es, für benachbarte Elemente unterschiedlich große Zeitschritte zu wählen. Dies soll Abbildung 1 verdeutlichen:

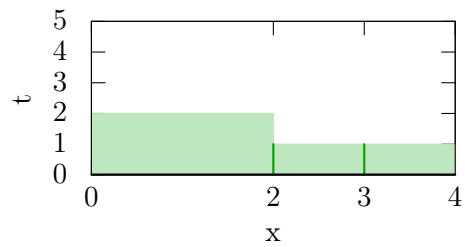
Zu Beginn (oder an ausgewählten Synchronisationspunkten) haben alle Elemente das gleiche Zeitniveau t_k^n (Abbildung 1a). Als erstes wird für alle Elemente Ω_k ein Prädiktor-Teilschritt mit lokal vorgegebenen Zeitschrittweiten Δt_k durchgeführt (Abbildung 1b). Nun kann ein Korrektor-Teilschritt mit den selben Schrittweiten für diejenigen Elemente durchgeführt werden, für die alle benötigten Information in den benachbarten Elementen vorliegen. Das ist genau dann der Fall, wenn der letzte Prädiktor-Schritt in den angrenzenden Elementen mindestens das Zeitniveau t_k^{n+1} des betreffenden Elementes Ω_k erreicht (Abbildung 1c). Anschließend kann man für dieses direkt eine neue Vorhersage über den nächsten Zeitraum $[t_k^{n+1}, t_k^{n+1} + \Delta t_k)$ bestimmen (Prädiktor-Schritt in Abbildung 1d).

Dieses Vorgehen wird nun wiederholt bis die Rechnung ein vorgegebenes Zeitniveau erreicht, wobei man für jedes Element in jeder Iteration eine eigene Zeitschrittweite wählen kann. Im vorliegenden Beispiel sind die Zeitschritte so gewählt, dass im folgenden Korrektor-Schritt wieder ein gemeinsames Zeitniveau erreicht wird (Abbildung 1e). Anschließend kann der ganze Vorgang wiederholt werden, beginnend mit dem nächsten Prädiktor-Schritt (Abbildung 1f).



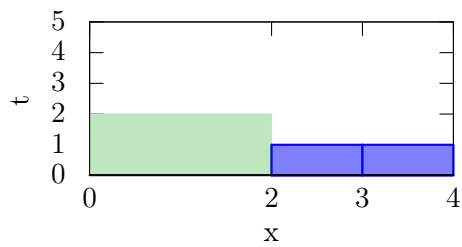
$t = 0$ —

(a)



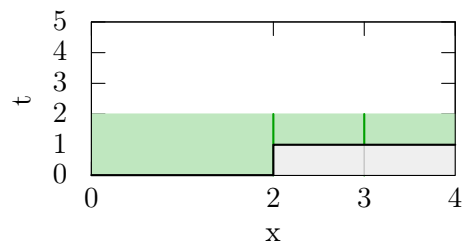
$t = 0$ Prädiktor

(b)



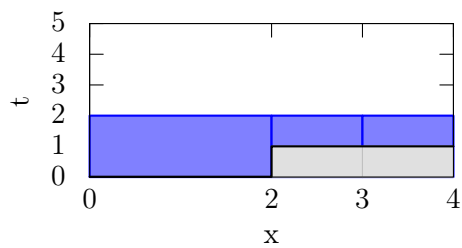
$t = 0$ Korrektor

(c)



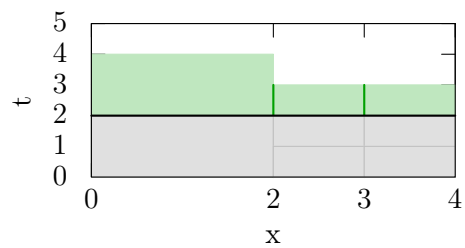
$t = 1$ Prädiktor

(d)



$t = 1$ Korrektor

(e)



$t = 2$ Prädiktor

(f)

Abbildung 1: Schematische Darstellung mehrerer Elemente mit unterschiedlichen lokalen Zeitschritten

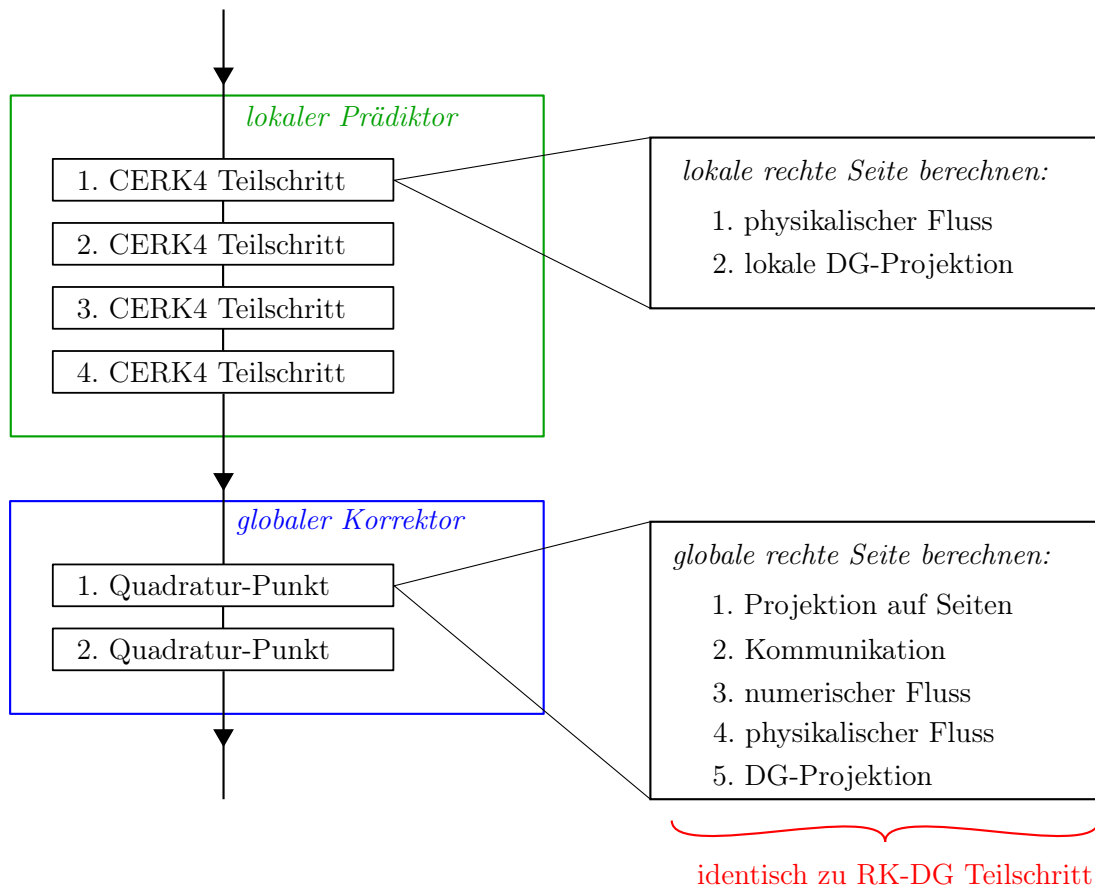


Abbildung 2: Schematische Darstellung des CERK4-Prädiktor-Korrektor-DG-Verfahrens

6 Implementierung und Ergebnisse

Das hier beschriebene Prädiktor-Korrektor-DG Verfahren wurde im Rahmen dieser Arbeit im HPC-Simulationsframework APES[6] implementiert. Bisher fehlt noch die Unterstützung lokaler Zeitschritte, da dafür in APES geeignete Datenstrukturen zur Kommunikation der lokalen Vorhersagen zwischen den Prozessen entworfen und implementiert werden müssen.

Der folgende Abschnitt stellt am Beispiel des APES-Framework dar, wie sich ein Prädiktor-Korrektor-DG Verfahren in ein bestehendes DG-Framework integrieren lässt. Anschließend soll kurz auf die Validierung und Performance der Implementierung eingegangen werden.

6.1 Skizze des gesamten Algorithmus

Abbildung 2 zeigt schematisch den Ablauf des hier vorgestellten CERK4-Prädiktor-Korrektor-DG Verfahrens:

Im ersten Schritt wird der lokale Prädiktor in einem Element mit einem vierstufigen CERK-Verfahren bestimmt. In jedem CERK-Teilschritt muss die rechte Seite der lokalen Diskretisierung berechnet werden. Diese ergibt sich aus dem physikalischen Fluss der Erhaltungsgleichung und der in Abschnitt 4.3 vorgestellten lokalen DG-Projektion.

Im zweiten Schritt wird die Vorhersage eines Elements (und ihrer Nachbarn) an zwei Quadratur-Punkten in der Zeit ausgewertet und an diesen die rechte Seite der DG-Diskretisierung berechnet. Diese Rechnung besteht aus der Projektion des Zustandes in den Elementen (in Polynomdarstellung) auf die Seiten der Elemente, der anschließenden Kommunikation dieser Seiten im parallelen Fall, der Berechnung des numerischen Flusses auf den Seiten und des physikalischen Flusses in den Elementen und der DG-Projektion mit den Testfunktionen. Sie ist identisch zur Bestimmung der globalen rechten Seite in einem Teilschritt eines RK-DG-Verfahrens.

Das Zustand zum nächsten Zeitschritt ergibt sich mit der gewichteten Summe der rechten Seiten an den Quadratur-Punkten.

Insgesamt zeigt sich, dass sich fast alle benötigten Operationen des Prädiktor-Korrektor-Verfahrens aus vorhandenen Bausteinen eines RK-DG-Verfahrens zusammensetzen lassen. Hinzu kommt lediglich die lokale DG-Projektion im Prädiktor-Schritt.

Mit dem Discontinuous-Galerkin-Framework in APES können sowohl lineare Erhaltungsgleichungen wie die Maxwell-Gleichungen simuliert werden, als auch nichtlineare wie die Euler-Gleichungen. Des Weiteren kann man zwischen zwei verschiedenen DG-Polynomräumen wählen (für eine genaue Darstellung siehe [4]).

Im Rahmen dieser Arbeit wurde das vorgestellte CERK4-Prädiktor-Korrektor-DG-Verfahren für alle obigen Kombinationen mit globalen Zeitschritten in APES eingebaut. Ein besonderes Augenmerk liegt dabei in APES auf der effizienten Implementierung der DG-Projektionen, die sich als Matrix-Vektor-Produkte mit speziellen dünnbesetzten Matrizen darstellen lassen. Noch nicht berücksichtigt wurden Quellterme in den Erhaltungsgleichungen, sowie eventuell benötigte Anpassungen an der CFL-Bedingung zur Bestimmung geeigneter Zeitschrittweiten.

6.2 Validierung und Performance

Die Implementierung konnte mit der in APES integrierten Testmatrix überprüft werden. Dabei ergaben sich mit allen Testfällen plausible Abweichungen zu RK4-DG-Rechnungen in Größenordnungen von unter 0,01%. Anhand einzelner Testfälle mit analytischer Lösung konnte überprüft werden, dass der Fehler zwischen RK-DG-Rechnungen 2. und 4. Ordnung in der Zeit liegt. Dies entspricht den Erwartungen, da man für das hier vorgestellte CERK4-Prädiktor-Korrektor-Verfahren einen Fehler 3. Ordnung in der Zeit erwartet.

Zusätzlich wurden die Rechenzeiten und die benötigten Gleitkommaoperationen (FLOPS) pro Freiheitsgrad und Zeitschritt (DOF-Update) bestimmt. Tabelle 1 zeigt die Ergebnisse dieser Messungen. Die benötigten FLOPS belegen, dass die lokale DG-Projektion ähnlich effizient wie die vorhandenen DG-Projektionen implementiert wurde. Anhand der Rechenzeiten lässt sich ablesen, dass der Aufwand des CERK4-Prädiktor-Korrektor-Verfahrens bei der Verwendung globaler Zeitschritte mit dem vorhandenen RK4-Verfahren

Testfall	Polynomraum	RK4	CERK4
Maxwell	Q8	225	194

(a) Flops pro DOF-Update

Testfall	Polynomraum	RK4	CERK4
Maxwell	Q2	1,6s	1,8s
Maxwell	Q4	2,6s	2,9s
Maxwell	Q8	5,4s	6,7s
Maxwell	P2	1,1s	0,9s
Maxwell	P4	0,9s	0,8s
Maxwell	P8	1,4s	1,4s
Maxwell	P16	4,5s	3,5s
Euler	Q4	1,7s	1,7
Euler	Q8	2,9s	3,4

(b) Rechenzeit

Tabelle 1: Gemessene Performance der CERK4-Prädiktor-Korrektor Implementierung im Vergleich zu RK4

vergleichbar ist; je nach Polynomraum- und Ordnung ergeben sich nur leichte Abweichungen. Dies verdeutlicht das Potential der Methode für die Verwendung mit lokalen Zeitschritten.

7 Fazit

In dieser Arbeit wurde veranschaulicht, wie ein Prädiktor-Korrektor-Ansatz zur Zeitdiskretisierung lokale Zeitschritte ermöglicht und wie dieser mit geringem Aufwand in ein bestehendes RKDG-Framework integriert werden kann. Zur effizienten Berechnung des lokalen Prädiktors wurde eine geeignete Formulierung im Rahmen der hier verwendeten speziellen DG-Diskretisierung im Raum hergeleitet.

Die Implementierung der Prädiktor-Korrektor-Zeitdiskretisierung ist in den HPC-Code APES eingeflossen, bisher jedoch unter Verwendung globaler Zeitschritte. Für die Unterstützung lokaler Zeitschritte in APES wird insbesondere noch Kommunikation zum Austausch der lokalen Vorhersagen zwischen den Prozessen der parallelen Rechnung zu unterschiedlichen lokalen Zeitpunkten benötigt.

Literatur

- [1] Courant, Richard, Kurt Friedrichs und Hans Lewy: *Über die partiellen Differenzengleichungen der mathematischen Physik*. Mathematische Annalen, 100:32–74, 1928.
- [2] Dahmen, Wolfgang und Arnold Reusken: *Numerik für Ingenieure und Naturwissenschaftler*. Springer-Lehrbuch. Springer-Verlag Berlin Heidelberg, zweite, korrigierte Auflage, 2008.
- [3] Gassner, Gregor, Michael Dumbser, Florian Hindenlang und Claus-Dieter Munz: *Explicit one-step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors*. Journal of Computational Physics, 230(11):4232–4247, May 2011.
- [4] Hükelheim, Jan: *Highly efficient, high order Discontinuous Galerkin schemes for conservation laws on octree datastructures*. Masterarbeit, RWTH Aachen University, December 2012.
- [5] Owren, Brynjulf und Marino Zennaro: *Derivation of efficient, continuous, explicit Runge-Kutta methods*. SIAM journal on scientific and statistical computing, Seiten 1–17, 1992.
- [6] Roller, Sabine, Jörg Bernsdorf, Harald Klimach, Manuel Hasert, Daniel Harlacher, Metin Cakircali, Simon Zimny, Kannan Masilamani, Laura Didinger und Jens Zudrop: *An adaptable simulation framework based on a linearized octree*. In: Resch, Michael, Xin Wang, Wolfgang Bez, Erich Focht, Hiroaki Kobayashi und Sabine Roller (Herausgeber): *High Performance Computing on Vector Systems 2011*, Seiten 93–105. Springer-Verlag Berlin Heidelberg, 2012.
- [7] Shu, Chi Wang: *Discontinuous Galerkin Methods: General Approach and Stability*. In: *Numerical solution of partial differential equations*, Advanced Courses in Mathematics - CRM Barcelona, Kapitel III, Seiten 149–202. Birkhäuser Basel, Basel, 2009.