

Jürgen Geiser

Computational Engineering

Theorie und Praxis der Transportmodelle:
Lösungen der Übungsaufgaben

25. Oktober 2017

Springer

Kapitel 1

Lösungen zu den Übungsaufgaben im Buch: *Computational Engineering: Theorie und Praxis der Transportmodelle*

Zusammenfassung Im Folgenden geben wir die Lösungen der Übungsaufgaben zum vorliegenden Buch *Computational Engineering: Theorie und Praxis der Transportmodelle* an. Dabei wollen wir die einzelnen Ideen motivieren und die Übungsaufgaben und Fragen sollen zum Nachdenken anregen. Wichtig ist auch das selbstständige Arbeiten im Bereich des Computational Engineering, da es sehr praxisorientiert ist und eine Programmiersprache, wie hier z. B. MATLAB[®], voraussetzt. Damit lassen sich die theoretischen Resultate aus den Modellproblemen und deren zugehörigen Methoden mittels numerischer Simulationen praktisch austesten und überprüfen. Die Theorie und Praxis muss daher miteinander geübt und verstanden werden. Deshalb ist es so wichtig, so dass nach der Theorie auch die praktischen Programmieraufgaben gelöst werden sollten. Ich wünsche Ihnen viel Erfolg bei der Durcharbeitung der Übungsaufgaben.

1.1 Fragen und Übungen zum Kapitel 3: Transport- und Strömungsmodelle im Bereich der Multiskalenmodelle

1.1.1 Fragen zum Verständnis des Kapitels (Kapitel 3)

Motivation und Definition von Modellierung und Multiskalenmodellen:

- 1) Was wird in der Hydrodynamik beschrieben?

Lösung:

Man beschäftigt sich mit der Strömungslehre von bewegten Fluiden (insbesondere bei der Hydrodynamik mit Flüssigkeiten). Man untersucht laminare und turbulente Strömungen, sowie Bewegungen und Kraftverhältnisse.

- 2) Was wird bei der Elektrodynamik beschrieben?

Lösung:

Die klassische Elektrodynamik beschäftigt sich mit bewegten elektrischen Ladungen und mit zeitlich veränderlichen elektrischen und magnetischen Feldern. Die zugrundeliegende Grundkraft ist dabei die elektromagnetische Wechselwirkungskraft, vgl. [3].

Eine Erweiterung der klassischen Elektrodynamik ist die Quantenelektrodynamik (QED). Für sehr kleine Skalen ergeben sich Probleme bei Punktladungen in der klassischen Elektrodynamik und man muss auf quantendynamische Prozesse zurückgreifen. Bei der Quantenelektrodynamik (QED) hat man die Vereinigung von Elektrodynamik mit quantenmechanischen Konzepten, vgl. [4].

- 3) Kennen Sie die einzelnen Ebenen zur Herleitung der hydrodynamischen Gleichungen?

Lösung:

Wir haben insgesamt drei Ebenen besprochen:

1. Vielteilchen Beschreibung (Liouville-Gleichung),
 2. Statistische Beschreibung (kinetische Gleichung),
 3. Kontinuierliche Beschreibung (Erhaltungsgleichungen).
- 4) Kennen Sie die einzelnen Ebenen zur Herleitung der elektrodynamischen Gleichungen?

Lösung:

Wir haben insgesamt drei Ebenen besprochen:

1. Quantenelektrodynamische Beschreibung (Dirac-Gleichung),
2. Mikromagnetische Beschreibung (Landau-Lifshitz-Gilbert-Gleichung),
3. Klassische elektrodynamische Beschreibung (Maxwell-Gleichungen).

- 5) Wie kann man Multiskalenmodelle definieren?

Lösung:

Durch die Multiskalenmodellierung wird eine Verbindung zwischen den unterschiedlichen skalenabhängigen Modellen hergestellt, z. B. hat man ein mikroskopisches und ein makroskopisches Modell. Die Parameter und Lösungen der einzelnen Gleichungen werden über sogenannte Kopplungsoperatoren miteinander verbunden.

- 6) Was ist das Problem bei den Multiskalenmodellen?

Lösung:

- Es werden unterschiedliche Zeit- und Raumskalen gekoppelt. Man hat ein Problem bei der Auflösung von sehr feinen Strukturen (Steifheit von Gleichungen).
- Kopplung von unterschiedlichen Modellen, d. h. hier werden diskrete oder kontinuierliche Modelle miteinander gekoppelt.
- Asymptotische Probleme, d. h. bestimmte Gleichungsoperatoren können in einer ε -Umgebungen zu Null werden. Damit ändert man den Gleichungscharakter, z. B. von einer gewöhnlichen Differentialgleichung zu einer algebraischen Gleichung.

- 7) Können Sie etwas über die Stabilität der expliziten und impliziten Verfahren sagen, wann sollte man explizite, wann implizite Verfahren verwenden?

Lösung:

Bei konvektiven Problemen, d. h. bei Problemen mit rauen Lösungen (kaum differenzierbar), sollte man explizite Verfahren verwenden. Dabei muss man die CLF-Bedingung beachten, d. h. Beschränkung der Zeitschrittweite. Bei diffusiven Problemen, d. h. bei Problemen mit glätt Lösungen (gut differenzierbar), sollte man implizite Verfahren verwenden. Dabei muss man keine CLF-Bedingung beachten, d. h. keine Beschränkung der Zeitschrittweite.

- 8) Wie sind die Ansätze von Multiskalenverfahren?

Lösung:

Man hat jeweils 3 Komponenten bei den Multiskalenverfahren:

1. Mikroskopische Gleichung,
2. Makroskopische Gleichung,
3. Kopplungsoperator.

Dabei muss das Interface, d. h. der Kopplungsoperator, auf die jeweiligen Skalenbedingungen zwischen Mikro- und Makro-Ebene (d. h. man hat einen Restriktionsoperator) bzw. Makro- und Mikroebene (d. h. man hat einen Interpolationsoperator) angepaßt werden.

- 9) Was können Sie über die Genauigkeit von Multiskalenverfahren aussagen?

Lösung:

Die Genauigkeit der Multiskalenverfahren hängt von den einzelnen Komponenten ab, d. h. Genauigkeit des mikroskopischen Löser, des makroskopischen Löser und des Kopplungsverfahrens. Die niedrigste Genauigkeit der Komponenten bestimmt dann die Ordnung des Gesamtverfahrens. Zum Beispiel sind die Kopplungsoperatoren oft von niedriger Ordnung (z. B. erster Ordnung), dann hat auch das Gesamtverfahren eine erste Ordnung, obwohl z. B. die Löser eine

höhere Ordnung haben.

- 10) Können Sie ein Beispiel für ein modulares Programmieren für Multiskalenprobleme beschreiben?

Lösung:

Ein klassisches Beispiel ist die Kopplung von zwei unterschiedlichen physikalischen Prozessen, z. B. Transport und Reaktion. Wir nehmen an, dass wir jeweils einen Softwarecode für die Transportgleichung und einen Softwarecode für die Reaktionsgleichung haben. Sprich wir haben zwei Module, diese werden dann mit einem dritten Modul, d. h. dem Kopplungsoperator, gekoppelt. Diese Programme sind dann modular aufgebaut in einem Hauptprogramm, welches auf die einzelnen Komponenten zugreifen kann.

- 11) Was für Programmpakete gibt es?

Lösung:

- Kommerziell: COMSOL, FEMLAB, ADINA, usw. .
- Akademisch: OperaSplitt (programmiert in MATLAB[®]), akademische Softwarepakete mit Fortran, akademische Programme mit MATLAB[®], MAPLE[®], MATHEMATICA[®].

- 12) Was ist das Problem der kommerziellen Softwarepakete im Bezug auf Multiskalenmodelle?

Lösung:

In den kommerziellen Programmpaketen sind die neuen Multiskalenmethoden nicht implementiert. Oft kann man auch die alten Strukturen der Programme, die auf einkalige (monoskalige) Probleme festgelegt sind, nicht mehr verändern. Man rechnet damit mit einem schlechteren Verfahren und erzielt ungenaue Ergebnisse.

- 13) Warum ist es notwendig neue Programme zu entwickeln?

Lösung:

Man muss oft neue Verfahren programmieren, die man so nicht mehr in Standardprogrammen findet. Oft haben die neuen Verfahren andere Datenstrukturen und man muss neue Konzepte entwickeln. Neue Programme helfen auch die aktuelle Forschung auf dem Gebiet der Multiskalenmodelle weiterzuentwickeln.

1.1.2 Übungsaufgabe: Steife Verfahren und deren Probleme (Kapitel 3)

Vergleichen Sie die Verfahren:

- Eulerverfahren (*euler.m*) und
- Diffusionsgleichung (*euler_implicit.m*),

für eine einfache Reaktionsgleichung:

$$y' = -\lambda y, y(0) = 1.0, \quad (1.1)$$

wobei $\lambda = 1.0$. Was passiert bei zu grossen Zeitschritten?

Lösung:

- Explizite Verfahren (explizites Eulerverfahren):
Hier haben wir eine Zeitschrittweitenbeschränkung von:

$$\Delta t \leq \frac{1}{\lambda}. \quad (1.2)$$

Falls wir zu große Zeitschritte nehmen haben wir Oszillationen (numerische Fehler).

- Implizite Verfahren (implizites Eulerverfahren):

Hier haben wir keine Zeitschrittweitenbeschränkung.

Aber falls wir zu große Zeitschritte wählen, z. B. $\Delta t \gg \frac{1}{\lambda}$, haben wir eine sehr starke numerische Diffusion und die Lösung verschmiert.

Ein Auszug aus einem MATLAB[®]-Code:

```
% Simulation

c1= 1- lambda* dt;
c2= 1/ (1+ lambda* dt);
for k=2:length(t)
    nForw(k)= c1* nForw(k-1);
    nBackw(k)= c2* nBackw(k-1);
end
```

Die Abbildung 1.1 zeigt einen zu großen Zeitschritt. Man sieht, dass das explizite Verfahren oszilliert, d. h. negative und positive Resultate. Das implizite Verfahren ist zwar stabil, aber hat dennoch starke Unterschiede zur exakten Lösung (Ausschmieren der Lösung).

Die Abbildung 1.2 zeigt angepasste kleine Zeitschritt. Man sieht, dass nun beide Verfahren gute Ergebnisse haben.

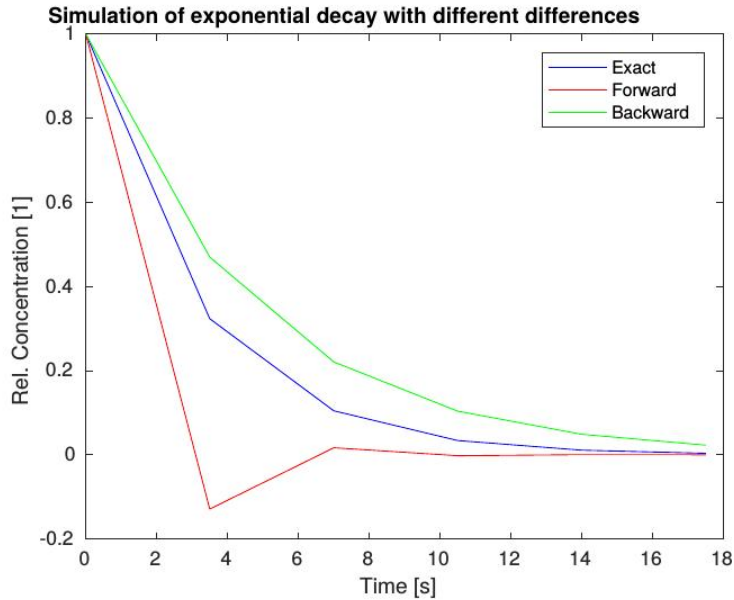


Abbildung 1.1: Probleme bei zu großen Zeitschritten.

1.2 Fragen und Übungen zum Kapitel 4: Theoretischer Überblick zu den numerischen Verfahren

1.2.1 Fragen und Programmieraufgaben zum Verständnis des Kapitels (Kapitel 4)

In Kapitel 4 haben wir die Modellierung von Transport- und Strömungsmodelle besprochen. Nachfolgend haben wir nun einige Programmieraufgaben in MATLAB®, sowie anschließend die Fragestellungen bereitgestellt.

1.2.1.1 Programmieraufgaben zur den Grundlagenaufgaben (Transportgleichung und Strömungsgleichungen)

Reaktionsgleichung

Nehmen Sie das MATLAB®-Programm aus Kapitel 4 her und modifizieren Sie den Code entsprechend der folgenden Reaktionsgleichungen:

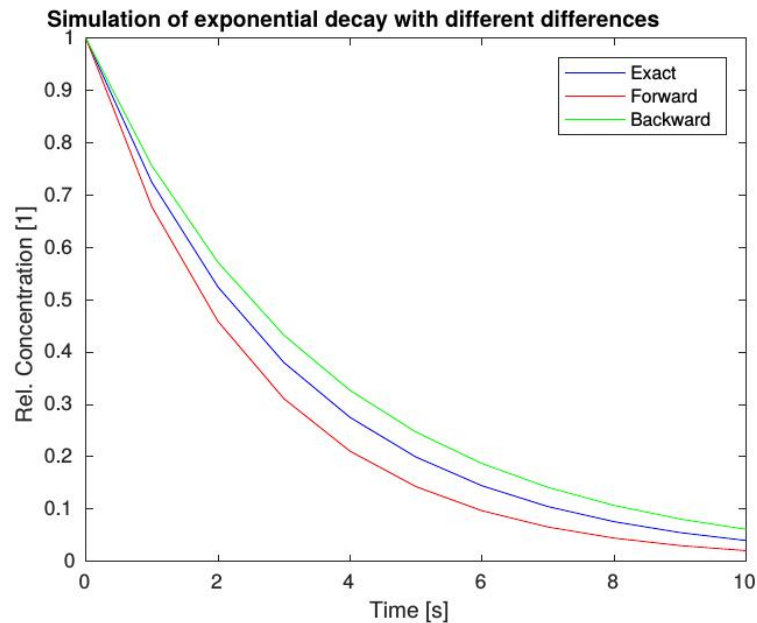


Abbildung 1.2: Verbesserte numerische Lösungen mit kleinen Zeitschritten.

$$\frac{dy_1}{dt} = y_2 y_3, \quad (1.3)$$

$$\frac{dy_2}{dt} = -y_1 y_3, \quad (1.4)$$

$$\frac{dy_3}{dt} = -0.51 y_1 y_2. \quad (1.5)$$

Testen Sie verschiedene Zeitschrittweiten aus.

Lösung:

```
%% Reaction equation written J.Geiser
syms t x l1 l2 l3
g = @(t,x,l1,l2,l3)...
[l1*x(2)*x(3);l2*x(1)*x(3);l3*x(1)*x(2)]
l1= 1.0;
l2= -1.0;
l3= -0.51;

% Kleiner Zeitschritt mit der internen
% Zeitschrittweitenoptimierung
```

```

%[t,x1] = ode45(@(t,x) g(t,x,l1,l2,l3), ...
%               [0 10.0], [1 1 1]);

% Grosser Zeitschritt von aussen gewaehlt,
% der zu gross ist
[t,x1] = ode45(@(t,x) g(t,x,l1,l2,l3), ...
               [0:5:10.0], [1 1 1]);
figure
plot(t,x1(:,1),t,x1(:,2),t,x1(:,3))
title(['Concentrations: x1 (blue), x2 (red), ...
       x3 (yellow)'])

```

Das Ergebnis ist in der folgenden Abbildung 1.3 gegeben. Dabei hat man bei einem zu großen Zeitschritt eine schlechte Auflösung mit dem impliziten GDGL Löser von MATLAB®. Erst mit einem genügend kleinen Zeitschritt hat man eine gute Auflösung.

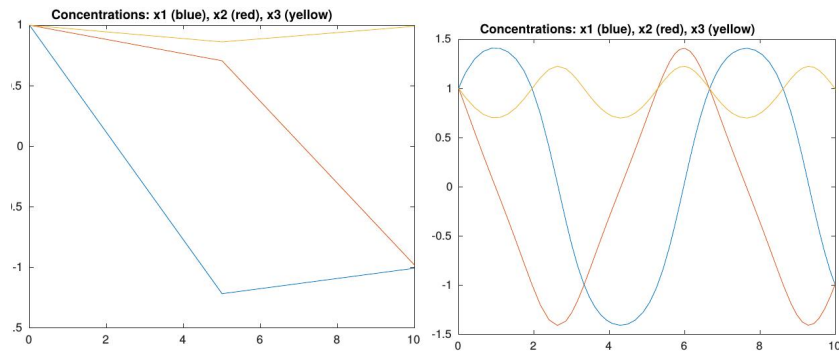


Abbildung 1.3: Das linke Bild zeigt eine schlechte Auflösung mit zu großem Zeitschritt, das rechte Bild hingegen hat eine bessere Auflösung mit einem kleinen Zeitschritt.

Konvektions-Diffusionsgleichung

Nehmen Sie die MATLAB®-Programme aus Kapitel 4 her.

Mittles das AB-Splitting können Sie die Programme koppeln. Dabei sei die nachfolgende Gleichung (1.6) mit dem Operator A die Konvektionsgleichung und die Gleichung (1.7) mit dem Operator B die Diffusionsgleichung.

$$\frac{dy_1(t)}{dt} = Ay_1(t), \quad y_1(t^n) = y(t^n), \quad (1.6)$$

$$\frac{dy_2(t)}{dt} = By_2(t), \quad y_2(t^n) = y_1(t^{n+1}), \quad (1.7)$$

für $n = 0, \dots, N$. Die Startbedingung $y(t^0)$ ist als Anfangsbedingung gegeben. Weiter sind die Ergebnisse des AB-Splitting gegeben als $y(t^{n+1}) = y_2(t^{n+1})$ und diese sind wiederum die Anfangsbedingungen für den nächsten Zeitschritt.

Lösung:

Ein möglicher MATLAB[®]-Code:

```
% Ein modulares Programm
% Teil A: Konvektionsgleichung mit expliziter
%         Zeitdiskretisierung und upwinding im Raum
% Teil B: Diffusionsgleichung mit impliziter
%         Zeitdiskretisierung und 2ter Ordnungs-
%         differenz im Raum
% Kopplung: Die Gleichungen werden mittels
%           des Ergebnisse aus dem vorhergehenden
%           Schritt gekoppelt (AB-splitting)

% Konstante

T= 1;
dx= 0.01;
x= 0: dx: 10;
u0= 0* x';
u0((x <= 1))= 1; % step 0

v= 1;
D= 0.1;

% CFL Bedingung
dt= dx/v;

t= 0: dt: T;

u= zeros(length(x), length(t));

% Operator A: Konvektionsgleichung

A= eye(length(x));
A(:, 1:end-1)= A(:, 2:end);
A(:, end)= 0;
A= A* (v*dt/dx);
A= A + eye(length(x))* (1 - v * dt/dx);
```

```

% Diffusion part

% Operator B: Diffusionsgleichung

C= eye(length(x));
B= [C(2:end, :); zeros(1, length(x))];
B= B + [zeros(1, length(x)); C(1:end-1, :)];
B= -B + 2* C;
B= B * D * dt/ dx/ dx;
B= B + C;
Binv= inv(B);

% Initialisierung und Randbedingungen

u(:,1)= u0; % initian condition
u(1,1)= 0; % boundary condition

% AB-Splitting Algorithmus

for k=2:length(t)

    u1= (A)*u(:,k-1); % A-step with the convection
    u(:,k)= Binv*u1; % B-step with the diffusion
    plot(x, u(:,k));
    title([num2str(t(k)), ' s']);
    ylim([0 1]);
    pause(0.1);
end

```

Das Ergebnis des Splittingverfahrens ist in der folgenden Abbildung 1.4 gegeben.

1.2.1.2 Fragen zum Verständnis des Kapitels (Kapitel 4)

Diskretisierungs- und Lösungsverfahren für Transportmodelle und Strömungsmodelle:

- 1) Welche Komponenten hat ein Transportmodell?

Lösung:

Wir haben folgende Komponenten:

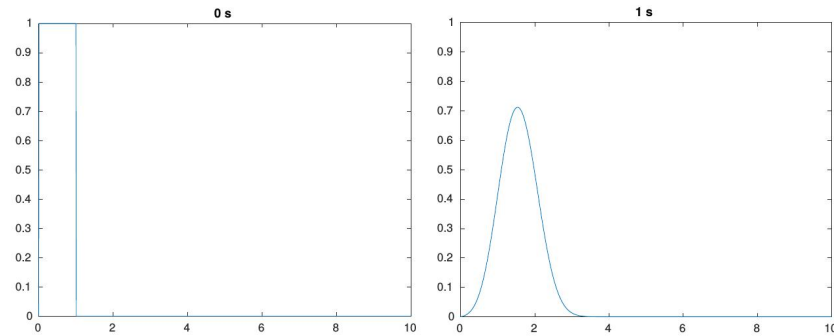


Abbildung 1.4: Das linke Bild zeigt die Initialisierung bei $t = 0$. Das rechte Bild zeigt die Konzentration der Konvektions-Diffusions-Gleichung bei $t = 1.0$.

- Konvektionsterm: Er gibt den konvektiven Anteil an, d. h. den gerichteten Transport (Bewegungsgleichung).
- Diffusionsterm: Er gibt den diffusiven Anteil an, d. h. die ungerichteten Ausbreitungen, z. B. eine symmetrische Ausbreitung in alle Richtungen (Interaktion der Fluidteilchen untereinander).
- Reaktionsterm: Er gibt den reaktiven Anteil an, d. h. den Zerfall- oder das Entstehen von Konzentrationen.

2) Was hat man für numerische Herausforderungen bzw. Probleme um eine Konvektions- oder Diffusionsgleichung zu lösen?

Lösung:

- Konvektionsgleichung: Hier hat man eine ungerade Ordnung des Differentialquotienten, d. h. man hat ein aufsteilendes Verhalten der Lösung und eine Abnahme der Differenzierbarkeit, vgl. [7]. Man muss numerische Verfahren verwenden, die dieses Verhalten auslösen, z. B. upwind-Verfahren, Charakteristiken-Verfahren usw..
- Diffusionsgleichung: Hier hat man eine gerade Ordnung des Differentialquotienten, d. h. man hat ein glättendes Verhalten der Lösung und eine Zunahme der Differenzierbarkeit, vgl. [7]. Man muss numerische Verfahren verwenden, die dieses Verhalten auslösen, z. B. symmetrische Diskretisierungsverfahren von höherer Ordnung, höhere Ansatzfunktionen für finite Elemente-Verfahren usw..

3) Können Sie einen einfachen expliziten Algorithmus für die Konvektionsgleichung aufschreiben?

Lösung:

Wir haben die Konvektionsgleichung:

$$c_t + ac_x = 0. \quad (1.8)$$

Wir diskretisieren explizit in der Zeit und nehmen ein upwind-Verfahren im Raum, dann haben wir die diskrete Gleichung (Differenzengleichung):

$$C_j^{n+1} = C_j^n - a \frac{\Delta t}{\Delta x} (C_j^n - C_{j-1}^n), \text{ für } n \geq 0, j = 1, 2, \dots, M, \quad (1.9)$$

wobei M die Anzahl der Raumpunkte und a die Geschwindigkeit ist. Weiter sind noch die Rand- und Anfangsbedingungen zu beachten, vgl. Kapitel 4.

- 4) Können Sie einen einfachen impliziten Algorithmus für die Diffusionsgleichung aufschreiben?

Lösung:

Wir haben die Diffusionsgleichung:

$$c_t - Dc_{xx} = 0. \quad (1.10)$$

Wir diskretisieren explizit in der Zeit und nehmen ein zweite Ordnungsverfahren im Raum, dann haben wir die diskrete Gleichung (Differenzengleichung):

$$\frac{1}{\Delta t} (C_j^{n+1} - C_j^n) = \frac{D}{\Delta x^2} (C_{j-1}^n - 2C_j^n + C_{j+1}^n), \quad (1.11)$$

$$C_j^{n+1} = (1 - 2\lambda)C_j^n + \lambda C_{j-1}^n + \lambda C_{j+1}^n, \quad (1.12)$$

für $n \geq 0, j = 1, 2, \dots, M$, wobei $\lambda = \frac{\Delta t D}{\Delta x^2}$ und D der Diffusionsparameter ist. M ist die Anzahl der Raumpunkte. Weiter sind noch die Rand- und Anfangsbedingungen zu beachten, vgl. Kapitel 4.

- 5) Wie löst man die Gleichungen zusammen?

Lösung:

Die Kombination zwischen der Konvektions- und der Diffusionsgleichung ist im Bereich der Splittingverfahren zu finden, z. B. AB-Splittingverfahren. Dabei löst man in einem Schritt die Konvektionsgleichung und im nächsten Schritt die Diffusionsgleichung. Die Gleichungen werden jeweils durch die letzten Resultate der anderen Gleichung gekoppelt (AB-Splitting).

- 6) Was für reale Anwendungen bei der Transportgleichung kennen Sie?

Lösung:

Transportgleichungen kommen bei der Modellierung von Transportproblemen

vor, z. B. Transport in Flüssigkeiten, Wärmetransport, Partikeltransport usw. .

7) Welche Komponenten hat eine Navier-Stokes-Gleichung?

Lösung:

Man hat einen nichtlinearen Konvektionsterm, einen linearen Diffusionsterm und einen linearen Druckterm. Die Erweiterung zur Burgers-Gleichung ist der zusätzliche Druckterm.

8) Wie sieht eine Zerlegung in Komponenten bei der NS-Gleichung aus?

Lösung:

- Nichtlineare Terme, d. h. der nichtlineare Konvektionsterm. Dieser Term kann z. B. mit expliziter Euler-Methode diskretisiert und mit der Lösung vom letzten Zeitschritt linearisiert werden.
- Linear Terme, d. h. der lineare Diffusionsterm. Dieser Term kann z. B. mit impliziter Euler Methode diskretisiert werden.
- Druck Terme, d. h. der lineare Druckterm. Dieser Term kann z. B. mit der Druckkorrekturmethode gelöst. Dabei löst man eine Poissongleichung und korrigiert die Lösung.

9) Was ist die Herausforderung bei den Burgers- und Navier-Stokes-Gleichungen?

Lösung:

Man hat zusätzlich nichtlineare Terme, die man mit einem weiteren Löser, dem sogenannten nichtlinearen Löser behandeln muss.

10) Welche nichtlinearen Löser kennen Sie?

Lösung:

Wir haben Fixpunktiterationsverfahren und Newtonverfahren zum Lösen von nichtlinearen Gleichungen.

11) Was für Vorteile haben Newtonverfahren?

Lösung:

Sie sind von höherer Ordnung. Ein Newtonverfahren hat die Ordnung 2, während das einfache Fixpunktverfahren von Ordnung 1 ist.

12) Wie sehen die Diskretisierungsverfahren für die Wellengleichungen aus?

Lösung:

Man verwendet explizite oder implizite Zeitdiskretisierungen und eine zweite Ordnungsdifferenz im Raum.

Hier ein Beispiel einer 1D-Wellengleichung:

$$u_{tt} = c_0^2 \frac{\partial}{\partial x} \frac{\partial}{\partial x} u. \quad (1.13)$$

Sie wird mit expliziten Differenzen, d. h. hier mit 2ter Ordnungsdifferenz in der Zeit und ebenso mit 2ter Ordnungsdifferenz im Raum, wie folgt diskretisiert:

$$U_j^{n+1} - 2U_j^n + U_j^{n-1} = \mu^2 (U_{j+1}^n - 2U_j^n + U_{j-1}^n), \quad j = 1, 2, \dots, M-1, \quad (1.14)$$

$$\mu^2 = \frac{\Delta t^2}{\Delta x^2} c_0^2, \quad (1.15)$$

wobei M die Anzahl der Raumschritte ist und c_0 ist die Geschwindigkeit. Weiter sind noch die Rand- und Anfangsbedingungen zu beachten, vgl. Kapitel 4.

- 13) Was ist ein FDTD-Verfahren, wie sieht das Staggered-Time-Space-Gitter aus?

Lösung:

Es ist ein explizites Diskretisierungsverfahren in Zeit und Raum (FDTD: Finite Difference Time Domain). Durch die Kopplung von Zeit und Raum und die verschobenen halben Schrittweiten in Zeit und Raum wird es besonders effektiv zur Lösung von hyperbolischen Differentialgleichungen, wie z. B. der Maxwell-Gleichung, eingesetzt werden. Dabei erhält man im Raum sowie in der Zeit ein Verfahren von 2ter Ordnung.

Ein typisches Staggered Grid (versetztes Gitter) ist in Abbildung 1.5 dargestellt.

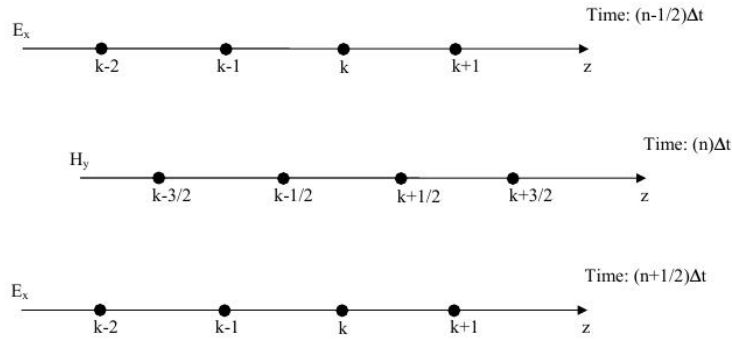


Abbildung 1.5: Das 1d-Staggered-Grid zeigt die Verschiebung zwischen Zeit und Raum um einen halben Diskretisierungsschritt.

- 14) Welche Programme kennen Sie, was muss neu entwickelt werden?

Lösung:

Es gibt die Programmpakete MATLAB®, COMSOL, MAPLE®, MATHEMATICA®. Standard-Codes haben oft nicht die Flexibilität um neue Modelle und Herausforderungen zu lösen. Man muss deshalb oft die neuen Verfahren in eigene Programm-Codes implementieren. Oft kann man mit den Standard-Codes nur spezielle Modelle lösen und zu starke Modifikationen bei neuen Modellen führen dann zu anderen Problemstellung. Deshalb ist es wichtig neue Programm-Codes zu entwickeln um neue Modelle und Methoden einzubinden.

- 15) Was können Sie über die Wirtschaftlichkeit der neuen Verfahren im Vergleich zu den alten Verfahren sagen?

Lösung:

Die neuen Verfahren sind effektiver und haben weniger Fehler.

1.3 Fragen und Programmieraufgaben zum Kapitel 5: Multiskalenverfahren zur effektiven Simulation von Transport- und Strömungsmodellen

In Kapitel 5 haben wir Multiskalenverfahren behandelt und an verschiedenen Transport- und Strömungsmodellen angewandt.

Nachfolgend nun eine Programmieraufgabe in MATLAB®, sowie anschließende Fragen zum Verständnis des Kapitels.

1.3.1 Programmieraufgabe zum HMM-Verfahren

Wir haben folgende Konvektionsgleichung gegeben:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left((f(x, \varepsilon) \frac{\partial}{\partial x} u) \right), \quad (1.16)$$

wobei die Multiskalenfunktion gegeben ist mit $f(x, \varepsilon) = 2 + \sin(2\pi x/\varepsilon)$.

Die Anfangs- und Randbedingungen sind gegeben mit:

$$\Omega = [a, b] = [0, 1], \quad (1.17)$$

$$\text{Grobes Gitter (macro-cells)} N = 100, \quad (1.18)$$

$$\text{Fines Gitter (micro-cells)} n = 10N, \quad (1.19)$$

$$T_{\text{end}} = 0.2, \quad (1.20)$$

$$\text{Mikroskopischer Faktor : } \varepsilon, \quad (1.21)$$

$$\text{Periodische Randbedingungen in der Mikrozelle.} \quad (1.22)$$

Wir wenden die HMM (*Heterogeneous Multiscale Method*) an

- Makroskopisches Problem:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta X} (J_i^n - J_{i-1}^n), \quad (1.23)$$

$$\text{mit } J_i^n = \frac{1}{M} \sum_{k=1}^M j_{i,k}^n, \quad (1.24)$$

und j_k^n sind die mikroskopischen Flüsse.

- Mikroskopisches Problem am Punkt x_j (gleiches auch bei x_{j-1}):

$$\begin{aligned} u_{i,k}^{n+1} = & u_{i,k}^n + \frac{\delta t}{(\delta x)^2} (f(x_{i,k+1}, \varepsilon) u^n(x_{i,k+1}) \\ & - (f(x_{i,k+1}, \varepsilon) + f(x_{i,k}, \varepsilon)) u^n(x_{i,k}) + f(x_{i,k}, \varepsilon) u^n(x_{i,k-1})), \end{aligned} \quad (1.25)$$

wobei $k = 1, 2, \dots, M-1$ mit $U_j^n = u_{j,0}^n$ und der Fluss $j_{i,k}^n = f(x_{i,k}, \varepsilon) \frac{u_{i,k}^n - u_{i,k-1}^n}{\delta x}$ ist.

Skizzieren Sie ein solches Programm in MATLAB[®], es können auch modulare Anteile sein.

Lösung:

Wir haben folgende modularen Anteile:

- Programm 1: Diskretisierung und Lösung der makroskopische Gleichung,
- Programm 2: Diskretisierung und Lösung der mikroskopische Gleichung,
- Kopplung der Gleichungen mittels Restriktion und Interpolation.

Die Idee ist im Folgenden skizziert, o.B.d.A. nehmen wir an, dass wir doppelt so viele feine Gitterpunkte haben.

- Teil: Diskretisierung der makroskopischen Gleichung:

...

% Macroscopic equation

```

for i=2:nx-1

    u(i)=un(i)+ dt*(j(i)-j(i-1))/dx;
end

...

```

- Teil: Diskretisierung der mikroskopischen Gleichung:

```

....

% Microscopic equation
for i=2:(2*(nx-1))

    umic(i)=unmic(i)+(vis*dt*(unmic(i+1)-2*unmic(i) ...
        +unmic(i-1))/(dxmic*dxmic));

end

...

```

- Teil: Kopplungsoperatoren zwischen den mikroskopischen und makroskopischen Variablen

```

...

% Initialisation of the microscopic values
% (restriction)

for i=2:nx-1
    unmic(2*(i-1))=un(i) ;
end

...

% Initialisation of the macroscopic values
% (Interpolation)

for i=2:nx-1

    visfact = vis * (2 + sin(2*pi*i*epsilon));
    j(i)= visfact*(umic(2*(i-1)+1) ...

```

```

- umic(2*(i-1))/dxmic ;

end

...

```

1.3.2 Fragen zum Verständnis des Kapitels (Kapitel 5)

Der Schwerpunkt ist dabei das Verständnis von den Multiskalenmodellen und Multiskalenlösern.

- 1) Können Sie etwas über einfache und komplexe Flüssigkeiten?

Einfache Flüssigkeiten benötigen nur eine Zeit- oder Raumskala zur Beschreibung. Es reicht die makroskopische Beschreibung mittels der Navier-Stokes-Gleichung aus. Komplexe Flüssigkeiten haben unterschiedliche Zeit- und Raumskalen und benötigen mehrere Modelle zur Beschreibung der komplexen Prozesse. So muss man z. B. die Polymere mit Modellen aus binären Mischungen von festen und flüssigen Phasen beschreiben. Man hat wiederum mikroskopische und makroskopische Modelle, die miteinander gekoppelt werden.

- 2) Was sind die Modelle für die einfachen Flüssigkeiten?

Lösung:

Bei den einfachen Flüssigkeiten, oder auch Monofluiden genannt, reichen makroskopische Modellgleichungen (Kontinuumsmodelle), die ein newtonsches Fluid beschreiben. Man verwendet dabei die Navier-Stokes-Gleichung. Bei idealen Fluiden kann man auch die Euler-Gleichung verwenden.

- 3) Was sind die Modelle für die komplexen Flüssigkeiten?

Lösung:

Hier hat man unterschiedliche Modellebenen und man braucht mikroskopische und makroskopische Modelle. Dabei kann man auf der makroskopischen Ebene eine Navier-Stokes-Gleichung hernehmen, muss aber unbekannte Parameter, wie z. B. die Viskosität oder den Spannungstensor mittels eines mikroskopischen Modells ergänzen. Das mikroskopische Modell ist dabei ein diskretes Modell auf der Ebene von Partikeln oder Atomen. Mittels mikroskopischer Gleichungen, z. B. Bewegungsgleichung, lassen sich dann die mikroskopischen Viskositäten oder Spannungstensoren berechnen. Diese werden dann mittels Kopplungsoperatoren in die makroskopische Gleichung übergeben.

4) Was versteht man unter *hybriden* numerische Verfahren?

Lösung:

Es sind Verfahren die unterschiedliche numerische Verfahren mit einander koppeln, z. B. Löser im Bereich von Gitterverfahren (finite Differenzen- oder finite Elemente-Verfahren) und Löser im Bereich von Partikelverfahren (Monte-Carlo-Verfahren, Partikellöser). Die Verfahren werden dann mittels Kopplungsoperatoren miteinander gekoppelt, z. B. Interpolation des Phasenraums in den Gitterraum und umgekehrt. Ein typisches Beispiel sind die PIC (Particle in Cell)-Verfahren.

5) Können Sie ein einfaches hybrides Verfahren nennen?

PIC (Particle in Cell)-Verfahren zum Lösen von Bewegungsgleichungen von Partikeln im Phasenraum und der elektromagnetischen Felder im Gitterraum. Dabei werden die Lösungen über Interpolationsfunktionen (Splines) miteinander gekoppelt.

6) Welche Komponenten haben die Multiskalenlöser?

Lösung:

- Löser für die makroskopische Gleichung,
- Löser für die mikroskopische Gleichung,
- Kopplungsoperatoren, z. B. Kompressionsoperatoren (Makro $\Rightarrow$ Mikro) und Rekonstruktionsoperatoren (Mikro $\Rightarrow$ Makro).

7) Was hat man für numerische Herausforderungen bzw. was für Probleme muss man bei Multiskalenmodellen überwinden?

Lösung:

Man muss jeweils die passenden Löser für die mikroskopische und makroskopische Gleichung finden. Weiter muss man die dazugehörigen Kopplungsoperatoren entwickeln, die das mikroskopische und makroskopische Modell koppeln.

8) Können Sie den HMM (Heterogeneous Multiscale Method)-Algorithmus skizzieren und erläutern (Ideen, Konzepte usw.)?

Lösung:

Man hat ein Top-Down-Verfahren, d. h. man geht von dem bekannten makroskopischen Modell aus und ergänzt noch fehlende Parameter, z. B. man ergänzt die makroskopische Viskosität mit der Berechnung der mikroskopischen Viskosität. Wir haben folgende Gleichung:

$$\frac{dx}{dt} = -x + y = f(x, y), \quad (1.26)$$

$$\frac{dy}{dt} = \frac{1}{\varepsilon}(-y + x), \quad (1.27)$$

wobei x die makroskopische Variable und y die mikroskopische Variable ist. Das HMM-Verfahren zur Lösung der Gleichungen (1.26) und (1.27) ist in Algorithmus 1.1 beschrieben.

Algorithmus 1.1 *Man initialisiert die Gleichungen mit den entsprechenden Anfangswerten.*

- *Schritt 1: Lösen der Mikrogleichung (hier werden die fehlenden Werte in der mikroskopischen Gleichung berechnet):*

$$y^{n,m+1} = y^{n,m} - \frac{\delta t}{\varepsilon}(y^{n,m} - \phi(x^n)), \quad (1.28)$$

mit $m = 0, 1, \dots, M-1$, z. B. $\delta t \leq \Delta t/M$ als Mikrozeitschritt. Weiter haben wir Δt als makroskopischen Zeitschritt.

- *Schritt 2: Equilibrieren der Mikrooperatoren, d. h. Rekonstruktion (hier wird die mikroskopische Variable durch Mittelung auf eine makroskopische Variable approximiert):*

$$F^n = \frac{1}{M} \sum_{m=1}^M f(x^n, y^{n,m}). \quad (1.29)$$

- *Schritt 3: Lösen der Makrogleichung (hier wird der verbesserte makroskopische Wert in der makroskopischen Gleichung verwendet):*

$$x^{n+1} = x^n - \Delta t F^n, \quad (1.30)$$

mit Δt als Makrozeitschritt.

- 9) Können Sie den EFM (Equation Free Method)-Algorithmus skizzieren und erläutern (Ideen, Konzepte usw.)?

Lösung:

Es ist ein Bottom-Up-Verfahren, d. h. das mikroskopische Modell ist bekannt und man konstruiert das makroskopische Modell. Es ist auch als ab-initio-Verfahren bekannt.

Es hat die folgenden Komponenten:

- Initialisieren der mikroskopischen Gleichung (Lifting). Hier werden bekannte makroskopische Werte in die mikroskopische Gleichung interpoliert.
- Berechnung der mikroskopischen Gleichung (Evolving). Hier werden sehr kleine Zeit- und Raumschritte verwendet.

- Extrapolieren der Makrogleichung (Restriction). Hier werden die makroskopischen Werte aus den mikroskopischen Werten extrapoliert.
- Rekonstruieren der Makrogleichung: Es werden die einzelnen Komponenten, d. h. die Initialisierungs-, Lösungs- und Extrapolationsoperatoren zusammengesetzt und man erhält die makroskopische Gleichung.

10) Können Sie den MSM (Multiscale Iterative Splitting Method)-Algorithmus skizzieren und erläutern (Ideen, Konzepte usw.)?

Lösung:

Es ist ein Top-Down-Verfahren, d. h. man beginnt wieder bei dem makroskopischen Modell und ergänzt noch fehlende Werte, z. B. Spannungstensoren aus dem mikroskopischen Modell. Der Vorteil zum HMM-Verfahren ist hierbei, dass man zusätzliche Iterationszyklen in einem makroskopischen Schritt hat. Damit hat man einen weiteren Löser, der die Ergebnisse verbessert und ausrelaxiert.

Es hat die folgenden Komponenten:

- Initialisierung von Iterationsschritten und Zeitintervallen.
- Evaluation der makroskopischen Gleichung: Die makroskopische Gleichung wird in der groben Skala mit einem groben Zeitschritt gelöst.
- Interpolation: Die makroskopischen Operatoren werden in die mikroskopische Skala transformiert (Kopplung von Makro $\Rightarrow$ Mikro).
- Evaluation der mikroskopischen Gleichung: Die mikroskopische Gleichung wird in der kleinen Skala mit vielen kleinen Zeitschritten gelöst.
- Restriktion: Die mikroskopischen Operatoren werden in die makroskopische Skala transformiert (Kopplung von Mikro $\Rightarrow$ Makro).

11) Was für Programmpakete gibt es?

Lösung:

- Kommerzielle Pakete: COMSOL, MATLAB[®], MAPLE[®], MATHEMATICA[®], ADINA usw.
- Akademische Pakete: universitäre und forschungsspezifische Codes in Fortran, MATLAB[®], C usw., z. B. OperaSplitt (Softwarepaket im Bereich der Splittingverfahren).

12) Was ist das Problem der kommerziellen Softwarepakete im Bezug auf Multiskalenmodelle?

Lösung:

In diesen Programmpaketen hat man sehr oft noch keine Multiskalenmodelle implementiert oder es gibt nur sehr einfache Kopplungsalgorithmen. Deshalb sollte man solche kommerziellen Softwarepakete immer sorgfältig prüfen, ob sie die

Modellanforderungen erfüllen, und ob hinreichend gute Löser vorhanden sind.

- 13) Warum ist es notwendig neue Programme zu entwickeln?

Lösung:

Die neuen Programme bilden die aktuelle Forschung im Bereich der Multiskalenmodelle und Multiskalenlöser ab. Es ist notwendig neue Programme zu entwickeln, da oft alte Datenstrukturen vorhanden sind, die man nicht mehr erweitern kann. Die neuen Methoden mit den unterschiedlichen Lösern brauchen wiederum neue Programmstrukturen, die man neu implementieren muss.

- 14) Was können Sie über die Wirtschaftlichkeit der neuen Verfahren im Vergleich zu den alten Verfahren sagen?

Lösung:

Die neuen Verfahren sind genauer und lösen das Problem besser auf. Da man weitere Modellgleichungen, wie z. B. das mikroskopische Modell lösen muss, sind sie zeitaufwendiger und teurer. Die alten Verfahren liefern nicht die Genauigkeit und können das Modellproblem nicht mehr beschreiben, sie sind deshalb nicht mehr zeitgemäss.

1.4 Fragen zum Kapitel 6: Ergänzung zu den Multiskalenmodellen, den Multiskalenmethoden, den Anwendungsbeispielen und der programmtechnischen Umsetzung

Schwerpunkt: Ergänzung zu den Multiskalenmodellen, den Multiskalenmethoden, den Anwendungsbeispielen und der programmtechnischen Umsetzung:

- 1) Kennen Sie praxisnahe Kopplungsverfahren, die man als Multiskalenverfahren einsetzen kann?

Lösung:

Man kann Operator-Splitting-Verfahren als Kopplungsverfahren verwenden:

- Entkoppeln der Gesamtgleichung in Teilgleichungen, die man jeweils für sich selbstständig lösen kann.
- Koppeln der Ergebnisse mit:
 - serieller Kopplung, d. h. die nächste Teilgleichung nimmt als Initialisierung die Lösung der vorhergehenden Teilgleichung,
 - parallele Kopplung, d. h. alle Ergebnisse der Teilgleichungen werden addiert, minus die Anzahl der letzten Lösungen.

2) Was ist ein modulares Splittingverfahren?

Lösung:

Man verwendet ein multiplikatives oder additives Splittingverfahren und kann dann nacheinander weitere Module hinzufügen. Die Idee kommt dem modularen Programmieren sehr nahe, da man nun ein Programm-Code erweitern kann, z. B. Transport-Code kann um einen Reaktions-Code erweitert werden. Die Kopplung wird dann mit einem Operator-Splitting-Verfahren durchgeführt.

3) Können Sie grob die Fokker-Planck-Gleichung skizzieren?

Lösung:

Die Fokker-Planck-Gleichung wird im Bereich der Modellierung von Partikeldichten verwendet, z. B. bei Plasmasimulationen oder Fluidsimulationen.

Sie ist im wesentlichen eine mehrdimensionale Konvektions-Diffusions-Gleichung, die aber nichtkonstante Konvektionsterme hat, die man über Charakteristikenverfahren lösen muss, vgl. [6].

Im Folgenden ist die 2D-FP-Gleichung gegeben.

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - E(x) \frac{\partial f}{\partial v} = \frac{\partial}{\partial v} (\gamma v f + \beta^{-1} \gamma \frac{\partial f}{\partial v}). \quad (1.31)$$

Sie hat nichtlineare Konvektionsterme in x und v und eine linear Diffusion in v .

4) Wo wird die FP-Gleichung angewendet?

Lösung:

Die FP-Gleichung wird im Bereich der Plasmaphysik verwendet, um statistische Verteilungsdichten vorherzusagen. Man hat hier eine mesoskopische Modellierung, die eine statistische Aussage über die Verteilung der Partikel gibt. Weiter wird sie auch im Bereich der Fluidmodellierung bei komplexen Flüssigkeiten verwendet, vgl. [1].

5) Was ist ein PIC-Verfahren?

Lösung:

Ein PIC-Verfahren ist ein Verfahren, das Partikel- und Gitterverfahren mischt (hybrides Verfahren). Es bietet sich bei der Simulation von dünnen Plasmen (geringe Plasmadichten) an, bei denen man große Clusters von Partikeln sogenannte Superpartikel bilden kann. Dabei hat man den Vorteil, dass man schnelle Löser für die Bewegungsgleichung (Partikellöser) und schnelle Löser für die Maxwell-Gleichungen (Gitterlöser) verwenden kann. Gekoppelt werden die Phasen- und Gitterräume über Interpolationsfunktionen.

6) Können Sie die einzelnen Komponenten des PIC-Verfahrens skizzieren?

Lösung:

Die Abbildung 1.6 zeigt die einzelnen Komponenten.

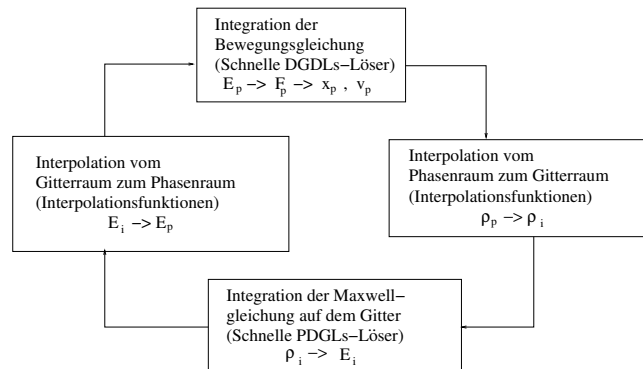


Abbildung 1.6: Einzelne Komponenten eines PIC-Verfahrens, sog. PIC-Zyklus.

- 7) Wie kann man eine Fokker-Planck-Gleichung mit einem PIC-Verfahren lösen?

Lösung:

Die FP-Gleichung kann man über die Trajektorien lösen. Dabei hat man eine Bewegungsgleichungen (newtonsche Gleichungen) mit der elektromagnetischen Kraft, die die Teilchen beeinflussen. Diese Kraft kann über die elektromagnetischen Felder berechnet werden, welche wiederum zu einer Maxwell-Gleichung führen. Damit hat man wiederum die Kombination von Partikel- und Gitterlöser. Dies lässt sich zyklisch mit einem PIC-Verfahren lösen.

- 8) Was sind komplexe Fluide und welche Anwendungen kennen Sie?

Lösung:

Komplexe Fluide sind Fluide mit komplexen Eigenschaften, z. B. viskoelastisches Kriechen, gebrochene Symmetrien (Flüssigkristalle), heterogene Systeme. Modelle für komplexe Fluide werden im Bereich der Simulation von Polymeren, von Flüssigkristallen und von Kolloiden angewendet.

- 9) Welche Probleme gibt es bei der Lösung von komplexen Fluiden im Bereich der Polymere?

Lösung:

Man muss zusätzlich den Spannungsterm in einem mikroskopischen Modell berechnen. Dabei hat man einen erheblichen Rechenaufwand. Weiter kann man auch ein makroskopisches Modell entwickeln unter Einbindung eines zeitabhängi-

gen Spannungstensor, der ein exponentielles Wachstum hat. Diese Systeme von partiellen DGL sind wiederum sehr aufwendig zu lösen, oder müssen entsprechend transformiert werden, z. B. logarithmische Formulierungen, vgl. [5].

- 10) Was sind Anwendungen der Magnetohydrodynamik und was ist die Magnetohydrodynamik?

Lösung:

Die Magnetohydrodynamik (MHD) beschreibt das Verhalten von elektrisch leitenden Fluiden. Dabei sind wiederum die magnetischen und elektrischen Felder wichtig und die Strömung von Flüssigkeiten. Man modelliert ein kontinuierliches Fluid und kann deshalb kontinuierlichen Modelle verwenden. Die Gleichungen sind eine Kombination aus der Navier-Stokes-Gleichungen (Hydrodynamik) und der Maxwell-Gleichungen (Elektrodynamik).

Man beschreibt die Strömungsbeeinflussung von Halbleitereinkristallzüchtung oder die Beschreibung von Plasmen in stellaren Atmosphären oder in Fusionsreaktoren.

- 11) Können Sie die Grundgleichung der MHD skizzieren?

Lösung:

Ein Idealfall ist eine Kombination aus Erhaltungsgleichungen aus der Hydrodynamik und der Magnetodynamik:

- Euler-Gleichungen (Hydrodynamik):
 - Massenerhaltung

$$\frac{\partial \rho}{\partial t} - \nabla \cdot (\rho \mathbf{u}) = 0, \quad (1.32)$$

- Impulserhaltung

$$\frac{\partial}{\partial t} (\rho \mathbf{u}) + \nabla \cdot \left(\rho \mathbf{u} \mathbf{u} + \left(p + \frac{B^2}{2} \right) \mathbf{I} - \mathbf{B} \mathbf{B} \right) = 0, \quad (1.33)$$

- Energieerhaltung

$$\begin{aligned} & \frac{\partial}{\partial t} \left(\frac{\rho u}{2} + \frac{p}{\gamma - 1} + \frac{B^2}{2} \right) + \\ & + \nabla \cdot \left(\left(\frac{\rho u}{2} + \frac{p}{\gamma - 1} + p \right) \mathbf{u} - (\mathbf{u} \times \mathbf{B}) \times \mathbf{B} \right) = 0. \end{aligned} \quad (1.34)$$

- Maxwell-Gleichung (Magnetodynamik):
 - Faraday'sches Gesetz

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{B} - \mathbf{B} \mathbf{u}) = 0. \quad (1.35)$$

- 12) Aus welchen Gleichungen besteht die MHD-Gleichung?

Lösung:

Im Idealfall hat man eine Euler-Gleichung und eine Maxwell-Gleichung.

- 13) Auf welche Eigenschaften muss man bei der Löserauswahl für die MHD-Gleichung achten?

Lösung:

Die MHD-Gleichungen sind hyperbolische partielle Differentialgleichungen und benötigen Diskretisierungsmethoden im Bereich von Erhaltungsgleichungen, z. B. mit Finite-Volumen-Methoden. Oft werden solche Gleichungen explizit in der Zeit diskretisiert und man hat zusätzlich noch Zeitschrittweitenbeschränkungen.

- 14) Was ist modulares Programmieren?

Lösung:

Die Idee ist die Verwendung von Modulen, die man dann in einem Hauptprogramm koordiniert. Damit lassen sich umfangreiche Programmpakete entwickeln und man kann solche Pakete später noch erweitern, z. B. durch die Hinzunahme von neuen Modulen.

- 15) Können Sie das modulare Programmieren für eine Konvektions-Diffusions-Reaktions-Gleichung skizzieren (Tipp: Wir nehmen an, dass wir ein Splittingverfahren verwenden)?

Lösung:

- Programm für die Lösung der Konvektionsgleichung.
- Programm für die Lösung der Diffusionsgleichung.
- Programm für die Lösung der Reaktionsgleichung.
- Programm für die Kopplung (z. B. Operator-Splitting-Verfahren) der Lösungen aus den einzelnen Teilprogrammen (Konvektion, Diffusion, Reaktion).
- Hauptprogramm zur Ablaufsteuerung des Splitting-Verfahrens (Kopplungsverfahren).

1.5 Fragen zum Kapitel 7: Numerische Umsetzung und Softwarepakete

Der Schwerpunkt ist hier im Verständnis der Umsetzung der numerischen Verfahren in Softwarepakete.

- 1) Welche kommerziellen Softwarepakete im Bereich der Transport- und Strömungsmodelle kennen Sie?

Lösung:

Adina, COMSOL, FEMLAB, usw..

- 2) Was ist das Problem kommerzieller Softwarepakete?

Lösung:

Kommerzielle Softwarepakete sind speziell für bestimmte Anwendungen programmiert und können vom Benutzer nicht mehr verändert werden. Damit lassen sich nicht die neuesten numerischen Verfahren implementieren. Oft muss dann der Benutzer einfachere Modelle für sein Problem finden, dass er noch mit der kommerziellen Software rechnen kann. Damit erhält er schlechtere Ergebnisse und qualitativ andere Resultate.

- 3) Sind schon neue Verfahren, z. B. Multiskalenlöser, in kommerziellen Softwarepaketen enthalten?

Lösung:

Nein, oft sind nur sehr einfache Kopplungsverfahren implementiert, die aber keine Multiskalenmethoden sind. Damit hat man nicht die Möglichkeit die aktuellsten und neuesten Entwicklungen aus der Forschung zu verwenden. Man rechnet oft mit alten und überholten Standardverfahren.

- 4) Was sind *Opensource* Softwarepakete und welche kennen Sie?

Lösung:

Im Bereich der Programmiersprache MATLAB[®] gibt es viele *offene* Softwarepakete, die man selbst anwenden und modifizieren kann. Weiter gibt es auch die Programmiersprache Python, die ebenfalls objektorientiert arbeitet und viele *offene* Software-Codes zur Verfügung stellt.

- 5) Was versteht man unter der Assemblierung einer Gleichung?

Lösung:

Die Assemblierung ist die Umsetzung einer Gleichung in einen Programmcode. Beispiel ist die Diskretisierung einer partiellen Differentialgleichung, diese kann dann in einer Matrix-Vektor Schreibweise in ein Programmcode umgesetzt werden. Man löst dann nur noch lineare oder nichtlineare Gleichungssysteme mit einer Matrix-Vektor-Notation.

- 6) Was ist der Vorteil der eigenen Entwicklung von Programmpaketen?

Lösung:

Man hat die Möglichkeit selbstständig die Programme zu modifizieren und zu erweitern.

- 7) Wie geht man bei der Umsetzung einer Modellgleichung bis hin zur Programmierung in MATLAB[®] vor?

Lösung:

Die Programmiersprache MATLAB[®] braucht eine Matrix-Vektor Notation, d. h. die Modellgleichung muss als lineares Gleichungssystem vorliegen. Man hat folgende Schritte:

- Diskretisierung der Modellgleichungen: Die Modellgleichungen werden diskretisiert und in lineare oder nichtlineare Gleichungssysteme geschrieben.
- Assemblieren der Diskretisierung: Die diskreten Gleichungen werden als lineare Gleichungssysteme programmiert.
- Simulation der Modellgleichung: Die lineare Gleichungssysteme werden gelöst und die Lösungsvektoren werden visualisiert, z. B. die Lösungen werden als Konturlinien auf einem Raumgitter dargestellt.

- 8) Erläutern Sie die Schritte für die Umsetzung vom Modell bis zum Simulationsprogramm?

Lösung:

- Mathematisches Modell: Das Modell wird in eine Modellgleichung geschrieben.
- Diskretisierung der Modellgleichungen: Die Modellgleichungen werden diskretisiert und in lineare oder nichtlineare Gleichungssysteme geschrieben.
- Programmierung der Diskretisierung: Die diskreten Gleichungen werden programmiert.
- Programmieren der Löser: Die Gleichungslöser, z. B. lineare oder nichtlineare Gleichungslöser, werden programmiert oder eingebunden.
- Simulation des Modells: Die Gleichungen werden gelöst und visualisiert, z. B. als eine Graphik der Lösungen auf einem Rechengitter.

- 9) Wie bewerten Sie die Multiskalenmodelle (Vorteil/Nachteil)?

Lösung:

Der Vorteil der Multiskalenmodelle liegt im Bereich der exakteren Auflösung des Modellproblems und dem besseren Verständnis der unterschiedlichen skalenabhängigen Prozesse. Der Nachteil ist ein erheblicher Programmieraufwand und eine neue Einarbeitung in unterschiedliche skalenabhängige Modelle und deren Methoden.

Insgesamt lohnt sich aber der Aufwand, da man verbesserte Ergebnisse erhält. Oft kann man nur mit den Multiskalenmodellen ein Verständnis, z. B. bei der Materialmodellierung oder der Modellierung von komplexen Flüssigkeiten, bekommen. Das Verständnis der makroskopischen Prozesse fängt schon im Bereich der mikroskopischen Prozesse an und deshalb ist der Zusammenhang und der Austausch der Parameter zwischen den unterschiedlichen Modellen sehr wichtig, vgl. [2].

Literatur

1. M. Deville and T.B. Gatski. *Mathematical Modeling for Complex Fluids and Flows*. Springer-Verlag, Berlin, Heidelberg, 2012.
2. J. Geiser. *Multicomponent and Multiscale Systems: Theory, Methods, and Applications in Engineering*. Springer, Cham, Heidelberg, New York, Dordrecht, London, 2016.
3. J.D. Jackson. *Classical Electrodynamics*. 3rd edition, John Wiley & Sons, Inc. Hoboken, New Jersey, USA, 1998.
4. C. Kleppel. *Von der Dirac-Gleichung zur Quantenelektrodynamik*. Springer Spektrum, Wiesbaden, 2015.
5. C. Le Bris and T. Lelièvre. *Multiscale modelling of complex fluids : A mathematical initiation*. in Multiscale Modeling and Simulation in Science Series, B. Engquist, P. Lötstedt, O. Runborg, eds., Lecture Notes in Computational Science and Engineering 66, Springer, 49-138, 2009.
6. H. Risken. *The Fokker-Planck Equation: Methods of Solutions and Applications* 2nd edition, Springer, Berlin, 1996.
7. A.E.P. Veldman. *Computational Fluid Dynamics*. Lecture Notes in Applied Mathematics, Academic year 2012-2013, 2013, www.math.rug.nl/~veldman/Colleges/CFD/CFDdictaat1314.pdf.