Numerische Methoden der Umweltphysik

Aufgabe A

Das notwendige Courant-Friedrichs-Levi (CFL) Kriterium für numerische Stabilität lautet im allgemeinen Fall

$$\left| c \cdot \frac{\Delta t}{\Delta x} \right| < 1$$

Dabei bezeichnet Δx die räumliche und $\Delta \tau$ die zeitliche Diskretisierung. Beantworten Sie die folgenden Fragen:

1. Was ist für c ganz allgemein einzusetzen?

2. Was würden Sie für c in der Atmosphäre einsetzen? Warum?

3. Kennen Sie Verfahren, deren Zeitschritt nicht durch das CFL-Kriterium begrenzt wird?

4. Heisst das, dass Sie bei solchen Verfahren einen beliebig grossen Zeitschritt wählen können?

Aufgabe B

Betrachten Sie folgende Teile eines Fortran-Programmes. Beantworten Sie folgende Fragen:

1. Welche Gleichung wird integriert? Geben Sie die Gleichung, die Grösse des Rechengebietes und die Art der Randbedingungen an!

2.Welches numerische Verfahren (im Orts und Zeitraum) wird verwendet?

3. Wie lautet die diskretisierte formale Form der Integration?

```
do i=2,nx-1
      new(i)=old(i)-(u*dt/dx)*(now(i+1)-now(i-1))
enddo
new(1) =new(nx-1)
new(nx)=new(2)
do i=1,nx
      old(i)=now(i)
      now(i)=new(i)
enddo
```

Aufgabe C

Betrachten Sie die null-dimensionale Reibungsgleichung

$$\frac{\partial u}{\partial t} = -\gamma u$$

und beantworten Sie dazu die folgenden Fragen:

1. Wie lautet die Diskretisierung dieser Gleichung mit Finiten Differenzen und einem Backward-Zeitschritt?

2. Welches ist der Stabilitätsbereich der Methode. Benutzen Sie zur Herleitung desselben die Von Neumann Methode.

3. Würden Sie empfehlen, den theoretischen Stabilitätsbereich voll auszuschöpfen? Begründen Sie Ihre Empfehlung!

Aufgabe D

Betrachten Sie folgende Teile eines Fortran-Programmes. Beantworten Sie folgende Fragen:

1. Welche Gleichung wird integriert? (Gleichung angeben)

2. Welches numerische Verfahren (im Orts und Zeitraum) wird verwendet?

3. Wie lautet die diskretisierte formale Form der Integration?

4. Was ist der Vorteil/Besonderheit der Methode?

```
do i=2,nx-1
      f(i)=0.5*(u(i)+u(i+1))*0.5*(now(i)+now(i+1))
enddo
do i=2,nx-1
      new(i)=old(i)-(dt/dx)*(f(i)-f(i-1))
enddo
new(1) =new(nx-1)
new(nx)=new(2)
do i=1,nx
      old(i)=now(i)
      now(i)=new(i)
enddo
```