

# 7 Siebte Übungseinheit

Inhalt der siebten Übungseinheit:

- Eigenwertaufgaben
  - Anschauliche Erklärung: `eigshow`
  - Vektoriteration
  - Einige Anwendungen
- Einschrittverfahren für gewöhnliche Differentialgleichungen erster Ordnung
  - MATLAB-Löser
  - Klassisches Euler-Verfahren und andere einfache explizite Verfahren
  - Fehlerordnung, Stabilität
  - Implizite Verfahren

## 7.1 Eigenwerte und Eigenvektoren

Was Sie an theoretischen Grundlagen für diese Einheit brauchen, fasst Abschnitt 8.1 im aktuellen Skriptum zusammen. Siehe dazu auch die Folien der 7. Vorlesung. Informieren Sie sich dort über die Definitionen und grundlegenden Eigenschaften! Die Übungsaufgaben stellen den Zusammenhang zu typischen Anwendungen her.

MATLAB-Befehle:

`d = eig(A)` liefert Vektor von Eigenwerten

`[V,D] = eig(A)` Spalten von  $V$  sind Eigenvektoren, Diagonalelemente von  $D$  sind Eigenwerte.

Ein interaktives, recht lehrreiches Beispiel zu Eigenvektoren liefert das MATLAB-Demoprogramm `eigshow`. Laden Sie `eigshow.m` von der Übungs-Homepage herunter und starten Sie es. Tipp: Unter <http://blogs.mathworks.com/cleve/2013/07/08/eigshow-week-1> erklärt Ihnen Cleve Moler, einer der MathWorks-Gründer, persönlich, warum es in `eigshow` geht.

Bewegen Sie den Vektor  $\mathbf{x}$  und beobachten Sie, wie sich  $A\mathbf{x}$  entsprechend ändert. Drehen Sie den Vektor  $\mathbf{x}$  so, dass er zu  $A\mathbf{x}$  parallel liegt. Jeder solche Vektor  $\mathbf{x}$  ist ein *Eigenvektor* der Matrix  $A$ .

Das Ergebnis  $A\mathbf{x}$  ist in diesem Fall ein Vielfaches der Ausgangsvektors, es gilt also die Gleichung

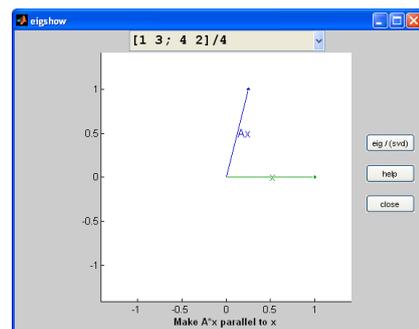
$$A\mathbf{x} = \lambda\mathbf{x} .$$

Der Proportionalitätsfaktor  $\lambda$  ist der zu  $\mathbf{x}$  gehörende *Eigenwert* von  $A$ .

Die Matrix  $A$  können Sie im Fenster oben aus einer Liste wählen. Voreingestellt ist

$$A = \frac{1}{4} \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} .$$

- Finden Sie für die obige Matrix  $A$  Eigenvektoren. Lesen Sie die Komponenten von  $\mathbf{x}$  ab und schätzen Sie  $\lambda$ . Wie viele verschiedene Werte von  $\lambda$  gibt es hier?



- Vergleichen Sie mit dem Resultat des Befehls  $[V,D] = \text{eig}(A)$ .
- Versuchen Sie andere Matrizen aus der Liste. Nicht immer können Sie Eigenvektoren finden. Es gibt Beispiele, wo in *gar keiner* Richtung  $\mathbf{x}$  und  $A\mathbf{x}$  auf einer Linie liegen. Was liefert  $[V,D] = \text{eig}(A)$  in solchen Fällen?
- Im Normalfall (bei regulärer Matrix  $A$ ) beschreibt  $A\mathbf{x}$  eine Ellipse, wenn Sie  $\mathbf{x}$  bewegen. Singuläre Matrizen bilden die Ausnahme. Sie haben  $\lambda = 0$  als Eigenwert. Wie sieht die Bahn von  $A\mathbf{x}$  dann aus? Finden Sie eine Beispielmatrix aus der Liste.

### Aufgabe 61: Spannungstensor, Hauptachsentransformation

Diese Aufgabe testet, ob Sie die MATLAB-Befehle aufrufen und deren Ergebnisse interpretieren können.

Reine Druck- oder Zugkräfte wirken normal auf ein Flächenelement, reine Scherkräfte parallel dazu. Im allgemeinen Fall stellt der Spannungstensor den Zusammenhang zwischen Flächennormalrichtung und Krafrichtung her:

$$\boxed{\text{Kraftvektor} = \text{Spannungstensor} \text{ mal Normalenvektor des Flächenelements}}$$

In einem geeignet gedrehten Koordinatensystem nimmt der Spannungstensor Diagonalgestalt an. Die Achsen dieses Koordinatensystems sind die *Hauptachsen* des Spannungstensors. Sie zeigen in Richtung seiner Eigenvektoren. Der diagonalisierte Spannungstensor enthält die Eigenwerte in der Hauptdiagonalen.

Der Spannungstensor in einem Material sei (in willkürlichen Einheiten)

$$p = \begin{bmatrix} 5 & 10 & -8 \\ 10 & 2 & 2 \\ -8 & 2 & 11 \end{bmatrix}.$$

- Welcher Kraftvektor wirkt auf ein Flächenelement parallel zur  $xy$ -Ebene? — zur Ebene  $x + y + z = 0$ ? (Dazu brauchen Sie noch keine Eigenwerte.)
- In welche Richtungen zeigen die Hauptachsen des Spannungstensors (`format rat` liefert „schöne“ Zahlen)? Wie lautet der auf Diagonalgestalt transformierte Tensor?
- In zwei Richtungen wirkt reine Zug-, in einer reine Druckspannung (Vorzeichenkonvention bei den Diagonaltermen des Spannungstensors: Druck ist negativ<sup>19</sup>. In welcher Richtung wirkt
  - die größere Zugspannung?
  - die kleinere Zugspannung?
  - die Druckspannung?

### Aufgabe 62: Vektoriteration

Was passiert, wenn man einen Vektor wieder und wieder mit derselben Matrix multipliziert? Testen Sie für den Vektor  $x = (1,0,0)^T$  und die Matrix  $A$

$$A = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

<sup>19</sup>wie auch sonst im Leben: Prüfungsdruck, Leistungsdruck, Erfolgsdruck,...

Wenn man den Befehl  $x = A*x$  zehnmal iteriert, werden die Zahlen in  $x$  schon recht groß.

Dividieren Sie nun zusätzlich in jedem Schritt den Vektor  $x$  nach der Multiplikation mit  $A$  durch seine erste Komponente. Das ändert nicht die Richtung von  $x$ , hält aber die Zahlenwerte der Komponenten in übersichtlichen Grenzen. Sie können das in der Form `>> y=A*x; x = y/y(1)` im Workspace durchspielen. Was beobachten Sie? Zu welchem Wert konvergiert  $y_1$ ?

Iterieren Sie, beginnend mit dem Startvektor  $x = (1,0,0)^T$ ,

```
>> y=A*x; x = y/y(1)
```

Zu welchem Wert konvergiert  $y_1$ ?

Vergleichen Sie: Für diese Matrix berechnet das Skriptum auf Seite 67 die Eigenwerte aus dem charakteristischen Polynom.

Im Skriptum auf Seite 69 ist Vektoriteration (in etwas allgemeinerer Fassung) als Pseudocode gegeben. Dabei ist noch berücksichtigt, dass die einfache Skalierung  $x = y/y(1)$  nicht funktioniert, falls  $y(1) = 0$ .

Die Beispielmatrix  $A = \begin{bmatrix} -1 & 0 \\ 1 & 2 \end{bmatrix}$  im Skriptum auf Seite 66 ist so ein Fall, wo Skalierung mit der ersten Komponente nicht zielführend ist. Rechnen Sie wie oben einige Schritte, dann sehen Sie, warum.

Ein Computerprogramm skaliert am besten mit jener Komponente, die größten Absolutbetrag hat. Der MATLAB-Befehl `[m,i]=max(abs(x))` findet im Vektor  $x$  den Index  $i$  dieser Komponente.

Eine simple MATLAB-Implementierung dazu wäre

```
for k=1:100
    x1 = A*x0;
    [m,i]=max(abs(x1));
    lam = x1(i);

    x1 = x1/lam;
    if(norm(lam - lam0)<eps)
        break
    end
    x0=x1;
    lam0=lam;
end
```

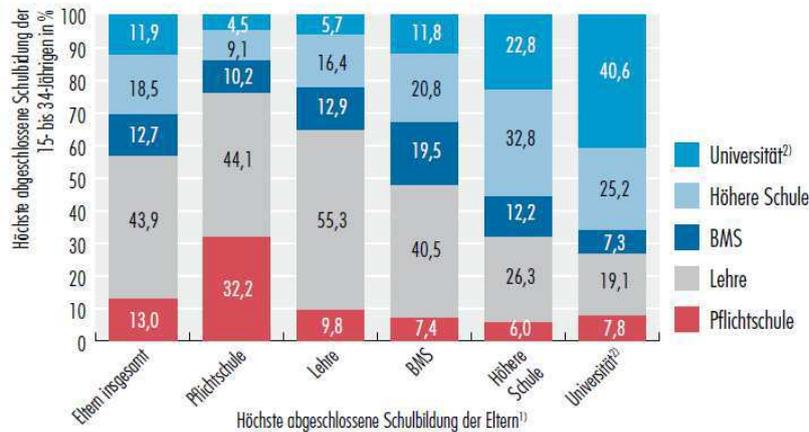
**Konkrete Aufgabe** Schreiben Sie ein Skript, das eine  $10 \times 10$ -Matrix  $A$  mit zufällig gewählten ganzzahligen Elementen  $a_{ij} \in \{0, \dots, 9\}$  erzeugt. Berechnen Sie mit Vektoriteration den betragsgrößten Eigenwert und einen zugehörigen Eigenvektor. Wählen Sie ein Abbruchkriterium mit Genauigkeit  $\epsilon < 10^{-6}$ .

Lassen Sie auch MATLABs `eig`-Befehl Eigenwert und Eigenvektor berechnen und vergleichen Sie die Ergebnisse. Beachten Sie, dass MATLAB Eigenvektoren als Einheitsvektoren in der 2-Norm, während die oben gezeigte Vektoriteration in der Unendlichnorm auf 1 skaliert. Zum Vergleich der Eigenvektoren müssen Sie MATLABs Eigenvektor gleich skalieren wie den aus der Vektoriteration berechneten!

### Aufgabe 63: Bildungsmobilität zwischen den Generationen

Es hängt stark von der sozialen Herkunft ab, welche Ausbildung Kinder und Jugendliche erhalten. Die Abbildung<sup>20</sup> zeigt Daten aus Österreich.

## 9 Intergenerationeller Bildungsvergleich



Q: STATISTIK AUSTRIA, Mikrozensus-Arbeitskräfteerhebung Ad-hoc-Modul „Eintritt junger Menschen in den Arbeitsmarkt“ – 2. Quart. 2009. Bevölkerung in Privathaushalten (15- bis 34-Jährige). – Pers. nicht in Ausbildung. – 1) Höchste abgeschl. Schulbildung d. Eltern: Haben Mutter und Vater nicht denselben Ausbildungsabschluss, wird jeweils die höhere Ausbildung verwendet. – 2) Inkl. hochschulverwandte Lehranstalten.

Erstellen Sie aus diesen Daten eine Matrix  $A = [a_{ij}]$ , in der das Element  $a_{ij}$  angibt:

Von Kindern, deren Eltern Ausbildung  $j$  haben, erreicht der Anteil  $a_{ij}$  die Ausbildung  $i$ .

Beispiel: Kinder aus Akademikerfamilien ( $j = 5$ ) erreichen zu 19.1% eine Lehre ( $i = 2$ ) als höchste Bildungsstufe; daher  $a_{25} = 0.191$ .

Der Balken ganz links in der Graphik gibt den Ist-Zustand (Bildungsniveau der Jugendlichen insgesamt) an. Setzen Sie diese Daten in einen Vektor  $\mathbf{x}^{(1)}$  ein. Angenommen, das Bildungsniveau der Eltern insgesamt ist durch einen Vektor  $\mathbf{x}^{(0)}$  beschrieben (diese Daten lassen sich nicht direkt aus der Grafik ablesen).

- Begründen Sie: Das Bildungsniveau der nächsten Generation insgesamt errechnet sich durch Matrix-Vektor-Multiplikation  $\mathbf{x}^{(1)} = A \cdot \mathbf{x}^{(0)}$ .
- Berechnen Sie  $\mathbf{x}^{(0)}$  aus den gegebenen Daten für  $A$  und  $\mathbf{x}^{(1)}$ .
- Berechnen Sie, ausgehend von  $\mathbf{x}^{(1)}$ , den Zustand nach 1, 2 und 3 weiteren Generationen;
- Ein *stabiler Zustand* besteht, wenn sich von einer Generation zur nächsten nichts ändert. Begründen Sie: dabei handelt es sich um einen Eigenvektor von  $A$ . Berechnen Sie den stabilen Zustand.

Die Berechnung des stabilen Zustandes lässt sich als Vektoriteration (Aufgabe 62) durchführen oder mit MATLABs `eig` Befehl.

<sup>20</sup>Quelle: *Bildung in Zahlen 2010/11, Schlüsselindikatoren und Analysen*. Statistik Austria, Wien, 2012

### Aufgabe 64: AHP – Analytic Hierarchy Process

Seit den 1990er-Jahren wird im *Operations Research* AHP zur Entscheidungsfindung immer populärer; hoffentlich nicht bloß deswegen, weil Anwender sich auf folgende Logik stützen:

- Korrekte Entscheidungen sind schwierig.
- Matrixalgebra ist schwierig.
- Daher kann Matrixalgebra korrekte Entscheidungen treffen.

Angenommen, Sie sollen entscheiden: „*Wohin gehen wir Freitag Abend feiern?*“. Zur Wahl stehen: Arkadenhof, Ristorante Antonio, Gösser Bräu. Zur Entscheidungsfindung mittels *Analytic Hierarchy Process* vergleichen und bewerten Sie die Lokale zuerst paarweise. (Der direkte Vergleich zweier Alternativen fällt den meisten Leuten leichter, als eine Rangordnung mehrerer Möglichkeiten zu erstellen.)

Dabei gilt die nebenstehende Skala:

Skalenwert	Bedeutung
1	gut
3	besser
5	gösser
2, 4	Zwischenwerte

Die Original-Methode verwendet Werte von 1–9; wir haben die Skala hier der Problemstellung angepasst.

Damit erstellen Sie eine Bewertungsmatrix: Der Eintrag in Zeile  $i$ , Spalte  $j$  bewertet, wie stark Alternative  $i$  gegenüber  $j$  bevorzugt wird. Wichtig dabei: der  $ji$ -Eintrag ist der Reziprokwert vom  $ij$ -Eintrag.

Jemand, der 2-mal lieber in den *A-Hof* als ins *Ristorante Antonio* geht und dem das *Gösser Bräu* 3-mal lieber als der *A-Hof* und 5-mal lieber als das *Antonio* ist, würde daher diese Matrix erstellen.

	AH	RA	GB
AH	1	2	1/3
RA	1/2	1	1/5
GB	3	5	1

Der Mathematiker Thomas L. Saaty, Entwickler von AHP, argumentiert, dass der Eigenvektor zum *betragsgrößten Eigenwert* dieser Matrix die beste Bewertung der Prioritäten angibt. Trotzdem sollten Sie dieser Methode keine übersinnliche Prognosequalität zuschreiben, es gilt vielmehr das EMEM-Prinzip: *Eingabe Mist – Ergebnis Mist*<sup>21</sup>

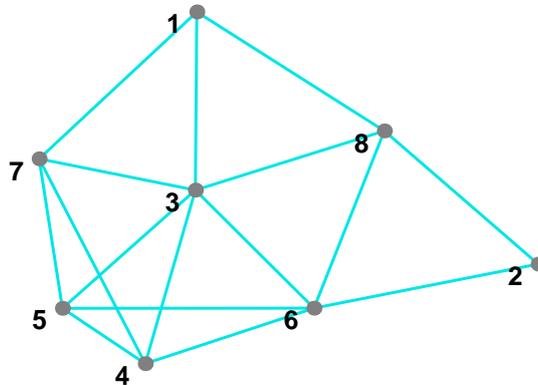
Berechnen Sie zur obigen Bewertungsmatrix den (auf Komponenten-Summe 1 normierten) Eigenvektor zum größten Eigenwert. Interpretieren Sie diesen Vektor als Prioritäten-Gewichtung.

Sie bemerken bei diesem Beispiel: Eigenwerte und -vektoren können auch komplex sein. Bei Matrizen mit ausschließlich positiven Einträgen sind aber garantiert der *betragsgrößte Eigenwert* und alle Komponenten eines zugehörigen Eigenvektors reell und positiv. In unserem Beispiel berechnet MATLAB den betragsgrößten Eigenwert an erster Stelle. Mit  $[V,D]=\text{eig}(A)$  ist automatisch  $V(:,1)$  der gesuchte Eigenvektor - er muss nur noch passend skaliert werden, damit die Summe der Komponenten 1 ergibt.

Die Berechnung dieses Eigenvektors lässt sich aber auch als Vektoriteration (Aufgabe 62) durchführen. Dieser Algorithmus ist im Gegensatz zum *QR*-Verfahren, das MATLAB verwendet, viel einfacher und lässt sich mit geringem Aufwand implementieren.

<sup>21</sup>In der Informatik sagt man: GIGO (*Garbage in – Garbage out*)

### Aufgabe 65: Erreichbarkeit in einem Netzwerk



Ein Netzwerk (Verkehrsverbindungen, verlinkte Seiten im Internet, soziales Netz. . . , mathematisch: ein Graph) lässt sich durch seine Adjazenzmatrix beschreiben<sup>22</sup> .

(a) Stellen Sie zum nebenstehenden Netzwerk die (01)-Adjazenzmatrix  $A$  auf und berechnen Sie den Eigenvektor zum größten Eigenwert. Wie ordnet dieser Vektor die Knoten im Netz nach dem Grad ihrer „Erreichbarkeit“?

(b) Beginnen Sie mit einem Vektor  $\mathbf{x}^{(0)}$  aus lauter Einsen und führen Sie zehn Schritte der Vektoriteration durch. Was erhalten Sie dadurch näherungsweise? Welches Maß für „Vernetzung“ oder „Erreichbarkeit“ ist durch  $A\mathbf{x}^{(0)}$  definiert?

Die Folien zur 7. Vorlesung zeigen ein ähnliches Netzwerk. Für ein praktisch relevantes Beispiel sehen Sie in Wikipedia zum Stichwort Page Rank nach. Im Moodle-Kurs finden Sie im Material zur 7. Vorlesung die Datei `pagerank.m`, sie vollzieht die Rechnungen im Wikipedia-Artikel nach.

<sup>22</sup>Die Idee, Links zwischen Internet-Seiten auf diese Art mathematisch zu untersuchen, untersuchte 1997 ein Forschungsprojekt an der Stanford University. Der daran arbeitende Doktorand (ein gewisser Larry Page), hat bis heute seine Dissertation nicht abgeschlossen. Völlig vergeblich war das Projekt trotzdem nicht; wenn es Sie interessiert, googeln Sie, was aus Larry P. und seinem Eigenvektor-Algorithmus geworden ist.

## 7.2 Gewöhnliche Differentialgleichungen erster Ordnung

---

### Aufgabenstellung:

Explizite gewöhnliche Differentialgleichung 1. Ordnung mit Anfangsbedingung

Gegeben ist eine Funktion  $f(x,y)$ . Gesucht ist eine Funktion  $y(x)$ . Sie soll erfüllen

$$y'(x) = f(x,y(x)) \quad \text{Differentialgleichung}$$

$$y(x_0) = y_0 \quad \text{Anfangsbedingung}$$

Die Folien der 8. Vorlesung zeigen die geometrische Interpretation dieser Aufgabe als Richtungsfeld im  $\mathbb{R}^2$  mit Lösungsfunktionen, die dem Richtungsfeld folgen.

Im Moodle-Kurs finden Sie zur 8. Vorlesung auch die MATLAB-Dateien `BeispieleGDG.m` und `GDGDemo.m`. Laden Sie sich diese Skript-Dateien herunter. Übersichtlich formatiert und gut lesbar werden diese Dateien, wenn Sie MATLABs Publish-Funktion nützen!

Die Datei `BeispieleGDG.m` zeigt zuerst auch die symbolische Lösung von Differentialgleichungen. Wir konzentrieren uns auf die numerische Lösung, denn brauchbare symbolische Lösungen existieren nur in einfachen Fällen.

### 7.2.1 MATLAB-Löser für gewöhnliche Differentialgleichungen

MATLAB bietet fertige Löser für Anfangswertprobleme an.

Es sind sieben Solver verfügbar: `ode23`, `ode23s`, `ode23t`, `ode23tb`, `ode45`, `ode113`, `ode15s`. Eine genauere Beschreibung können Sie in der MATLAB-Hilfe nachlesen.

Die Funktion `ode45` ist das Standardverfahren, welches man für ein „neues“ Anfangswertproblem zuerst probieren wird. Sie ist die Implementierung eines expliziten Einschritt-Runge-Kutta-Verfahrens (Fehlerordnung 5 und Kontrollrechnung mit Fehlerordnung 4).

Es folgt ein kleines Beispiel, das die Verwendung von `ode45` demonstriert. Angenommen, Sie wollen für  $y = y(x)$  das Anfangswertproblem

$$y' = g(x,y) = 2x(1 + y^2), \quad y(0) = 0,$$

im Intervall  $[0,1]$  mit diesem Solver lösen. Sie speichern dafür die Funktion  $g$  in einer Datei `g.m` ab

```
function ydot=g(x,y)
ydot=2*x*(1+y^2);
end
```

und fassen dann die Lösung des Anfangswertproblems in einer Scriptdatei (`test.m`) zusammen:

```
y0=0.0;
xspan=[0, 1];
ode45(@g,xspan,y0)
```

So einfach ist das! Sie erhalten eine graphische Darstellung der Lösung. (Noch kürzer ist die Implementierung in der Beispieldatei `BeispieleGDG.m!`)

Sie erhalten Vektoren von  $x$ - und  $y$ -Werten statt der grafischen Ausgabe, wenn Sie folgende Syntax verwenden:

```
[X, Y] = ode45(@g,xspan,y0);
```

MATLAB wählt sich aber die Lage der  $x$ -Werte selbst aus (um die Fehler unterhalb einer bestimmten Schranke zu halten). Wenn Sie die Lösung numerisch für bestimmte, von Ihnen gewählte  $x$ -Werte berechnen wollen, schreiben Sie beispielsweise

```
sol=ode45(@g,xspan,y0);  
x = [0 0.25 0.5 0.75 1];  
y = deval(sol,x)
```

### Aufgabe 66:

Berechnen Sie für die nachfolgend gegebenen Funktionen  $y = y(x)$  die Lösung des Anfangswert-Problems. Stellen Sie die Lösung graphisch dar.

a)  $y' = y$  mit  $y(0) = 1$ , Lösung gesucht im Bereich  $0 \leq x \leq 2$ .

Das ist die klassische Differentialgleichung der Exponentialfunktion, exakte Lösung ist  $y = e^x$ . Stellen Sie auch (in einem separaten Diagramm) den Fehler zwischen numerischer und exakter Lösung dar.

b)  $y' = xy/4 - 1$  mit drei verschiedenen Anfangsbedingungen:  $y(0) = 2; 2,5; 3$ , Lösungen gesucht im Bereich  $0 \leq x \leq 4$ .

Vergleichen Sie: im Skriptum sind Lösungen dieser Gleichung im Richtungsfeld dargestellt.

c)  $y' = \frac{1}{x^2}$  mit  $y(-1) = 1$ , Lösungen gesucht im Bereich  $-1 \leq x \leq 1$ . Prüfen Sie nach: Die Funktion  $y = -\frac{1}{x}$  erfüllt in ihrem Definitionsbereich Differentialgleichung und Anfangsbedingung. Bei der numerischen Lösung müssen Sie allerdings auf Probleme gefasst sein. Die Funktion  $y'$  hat hier eine spezielle Eigenschaft, die eine Lösung im gesamten Intervall  $-1 \leq x \leq 1$  nicht zulässt – welche Eigenschaft ist das? Vergleichen Sie mit der numerischen Lösung.

### Aufgabe 67: Setzen der Fehlerschranken

Gegeben sei für  $y = y(x)$  das Anfangswertproblem

$$y' = f(x,y)$$

mit

$$f(x,y) = -200xy^2, \quad y(-3) = \frac{1}{901}.$$

Berechnen Sie unter Verwendung des MATLAB-Solvers `ode45` die Lösung für  $x = -2, -1, 0$ .

Fehlerschranken: Verwenden Sie die Befehle

```
options=odeset('reltol',1.e-8,'abstol',1.e-8);  
sol=ode45(@g,xspan,y0,options)
```

um die vorgegebenen Fehlerschranken zu ändern. Versuchen Sie auch Fehlerschranken  $10^{-10}$ ,  $10^{-12}$ ,  $10^{-14}$ . Berechnen Sie jeweils den Wert  $y(0)$ .

Für dieses Problem lässt sich auch die exakte Lösung leicht angeben:

$$y_{\text{exakt}}(x) = \frac{1}{100x^2 + 1}$$

Vergleichen Sie für  $x = 0$  die numerisch berechneten Werte  $y(0)$  mit der exakten Lösung  $y_{\text{exakt}}(0) = 1$ .

## 7.2.2 Explizite Einschrittverfahren

Bei der numerischen Lösung einer DG bestimmt man ausgehend von den Anfangsbedingungen eine Folge von Wertepaaren  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $\dots$ , die den Verlauf der gesuchten Funktion  $y = y(x)$  annähern sollen. Schema:

Wähle Schrittweite  $h$  und maximale Schrittzahl  $N$ ;  
 setze  $x_0$  und  $y_0$  laut Anfangsbedingung;  
 für  $i = 0, 1, \dots, N - 1$   
 $x_{i+1} = x_i + h$ ;  
 $y_{i+1} = y_i + hF(x_i, y_i, h)$ .

Die Funktion  $F(x, y, h)$  heißt die *Verfahrensfunktion* des jeweiligen Verfahrens. Geometrisch interpretiert gibt  $F(x, y, h)$  die *Fortschreit-Richtung*. Die verschiedenen Verfahren unterscheiden sich in der Verfahrensfunktion, also in der Definition der Fortschritt-Richtung. Nur beim expliziten Euler-Verfahren ist die Fortschritt-Richtung im Punkt  $(x; y)$  auch gleich der Steigung  $y'(x, y)$ , also

$$F(x, y, h) = f(x, y),$$

Andere Verfahrensfunktionen versuchen, die Fortschritt-Richtung besser an den Verlauf der Lösung anzupassen. Die Folien der 8. Vorlesung stellen verschiedene Einschrittverfahren und deren Verfahrensfunktionen bildlich dar.

Beim modifizierten Euler-Verfahren ist

$$F(x, y, h) = f\left(x + \frac{h}{2}, y + \frac{h}{2}f(x, y)\right),$$

beim Verfahren von Heun

$$F(x, y, h) = \frac{1}{2}(k_1 + k_2)$$

mit

$$\begin{aligned} k_1 &= f(x, y) \\ k_2 &= f(x + h, y + hf(x, y)). \end{aligned}$$

Dazu gibt es ein Musterprogramm `GDGdemo.m` zum Herunterladen!

Tipp: Lassen Sie sich von Cleve Moler, einer der MathWorks-Gründer, persönlich erklären, wie das Runge-Kutta-Verfahren das Anwachsen einer Flamme berechnet!

## 7.2.3 Handrechnung und einfache Programme fürs Verständnis

„Ich habe ein numerisches Verfahren erst dann verstanden, wenn ich mich selbst dabei verrechnet habe“

Die numerische Lösung einer Differentialgleichung durch händische Rechnung dient heutzutage nur mehr der Illustration der Rechenverfahren. Die praktische Durchführung überlassen Sie dem Computer. Ihr Verständnis für die Rechenverfahren müssen Sie aber zumindest bei der Vorlesungsprüfung dadurch demonstrieren, dass Sie einige Rechenschritte selbst auf dem Papier ausarbeiten.

Die Folien der 8. Vorlesung zeigen sehr ausführlich die einzelnen Rechenschritte der Verfahren, die sie hier durchrechnen sollen. Das Musterprogramm `GDGdemo.m` implementiert die Rechenschritte in MATLAB.

Ein Beispiel sei das Anfangswertproblem für die Funktion  $y(x)$

$$y' = 4xy + 3$$

$$y(0) = 0$$

In diesem Fall ist also  $f(x,y) = 4xy + 3$  und  $x_0 = y_0 = 0$ . Wir wollen mit Schrittweite  $h = 0.2$  den Wert der Funktion  $y(x)$  an den Stellen  $x_1 = 0.2$ ;  $x_2 = 0.4$ ;  $x_3 = 0.6$  näherungsweise bestimmen. Die exakte Lösung ist bekannt, aber nicht durch elementare Funktionen gegeben; auf sechs Nachkommastellen genau beträgt  $y(0.6) = 2.973414$ .

Ein tabellarisches Rechenschema ist hilfreich. Beachten Sie: Die vorletzte Spalte berechnet immer die Verfahrensfunktion, die letzte Spalte entspricht dem allgemeinen Schritt  $y_{i+1} = y_i + hF(x_i, y_i, h)$ , in einfachen Worten: „neuer  $y$ -Wert = alter Wert plus  $h$  mal Fortschrittsrichtung  $F$ “

Für das explizite Euler-Verfahren:

$i$	$x$	$y$	$F = f(x,y) = 4xy + 3$	$y + hF$
0				
1				
2				

Modifiziertes Euler-Verfahren:

$i$	$x$	$y$	$k_1 = f(x,y)$	$y + \frac{h}{2}k_1$	$F = f(x + \frac{h}{2}, y + \frac{h}{2}k_1)$	$y + hF$
0						
1						
2						

Verfahren von Heun:

$x$	$y$	$k_1 = f(x,y)$	$y + hk_1$	$k_2 = f(x + h, y + hk_1)$	$F = (k_1 + k_2)/2$	$y + hF$

### Aufgabe 68: Einfache Einschrittverfahren programmieren, Fehlerordnung erkennen

Verwenden Sie das Musterprogramm `GDGdemo.m` und berechnen Sie für das oben gegebene Beispiel den Wert  $y(0,6)$  mit verschiedenen Schrittweiten und verschiedenen Verfahren:

1. Rechnen Sie mit  $h = 0,2$  und den drei obigen Verfahren. (Vergleichen Sie mit den Werten aus der Handrechnung.) Implementieren Sie auch das dreistufige Verfahren

$$\begin{aligned}k_1 &= f(x,y) \\k_2 &= f\left(x + \frac{h}{2}, y + \frac{h}{2}k_1\right) \\k_3 &= f\left(x + h, y - hk_1 + 2hk_2\right) \\F &= \frac{1}{6}(k_1 + 4k_2 + k_3)\end{aligned}$$

2. Rechnen Sie nun mit feinerer Schrittweite  $h = 0,05$ . Wie groß ist jeweils der Fehler (Differenz zwischen Näherungswert und exakter Lösung  $y(0,6) = 2,973414$ )?
3. Halbieren Sie die Schrittweite, rechnen Sie also mit  $h = 0,025$  und vergleichen Sie die Fehler. Um welchen Faktor hat sich jeweils der Fehler reduziert?

Die verschiedenen MATLAB-Löser für Anfangswertprobleme arbeiten im Prinzip wie das Demo-Programm `GDGdemo.m`, aber

- die Verfahrensfunktion bestimmt die Fortschritt-Richtung aus mehreren, optimal gewählten Zwischenpunkten,
- der (globale) Fehler wird abgeschätzt,
- die Schrittweite wird automatisch angepasst.
- für *steife Probleme* gibt es Methoden (`ode23s`, `ode15s`) mit besserem Stabilitätsverhalten.

## 7.2.4 Stabilität eines Verfahrens

Rechenverfahren sollten den Verlauf der Lösung, wenn schon nicht genau, so doch wenigstens *qualitativ* richtig bestimmen.

Konkret: wenn die exakte Lösung eine exponentiell abklingende Funktion ist, dann soll auch die numerische Lösung abklingen. Leider berechnen bei zu großer Schrittweite die bisher vorgestellten *expliziten* Verfahren exponentiell *anwachsende* Lösungen – ziemlich genau das Gegenteil von dem, was Sache ist.

Ein Verfahren heißt *stabil* bei Schrittweite  $h$ , wenn für das Modellproblem

$$y' = -y \quad y(0) = 1$$

die numerische Lösung mit wachsendem  $x$  nach Null konvergiert.

Sie lernen nun *implizite* Verfahren kennen; diese machen bei jedem einzelnen Rechenschritt ähnlich große Fehler wie explizite Verfahren. Aber bei expliziten Verfahren schaukeln sich Fehler von einem Schritt zum nächsten exponentiell auf; bei impliziten Verfahren klingen die Fehler in Summe ab und zerstören nicht den qualitativen Verlauf der Lösung.

**Aufgabe 69: Stabilität expliziter Einschrittverfahren**

Modifizieren Sie das Musterprogramm `GDGdemo.m` und testen Sie die bisher behandelten Einschritt-Verfahren (Euler explizit, Euler modifiziert, Heun sowie das dreistufigen Verfahren aus Aufgabe 68) für verschiedene Schrittweiten  $h$  beim Anfangswertproblem für  $y(x)$

$$\begin{aligned}y' &= -y \\ y(0) &= 1\end{aligned}$$

Rechnen Sie bis  $x = 10$ , und finden Sie für jedes Verfahren  $h$ -Werte, für die das Verfahren gerade noch abklingende, beziehungsweise bereits leicht anschwellende Lösungen berechnet – das sind also  $h$ -Werte gerade noch innerhalb, beziehungsweise knapp außerhalb der Stabilitätsgrenze.

Vergleichen Sie dazu die Abbildungen auf der vorletzten Vorlesungsfolie, 9. Vorlesung!

**Aufgabe 70: Implizites Eulerverfahren**

Hier ist die Verfahrensfunktion gegeben durch

$$F(x,y,h) = f(x+h,y(x+h))$$

und der neue Näherungswert daher

$$y(x+h) = y(x) + hf(x+h,y(x+h))$$

Anders als beim expliziten Verfahren lässt sich der gesuchte Wert nicht direkt finden: Sie können  $y(x+h)$  auf der linken Seite nur berechnen, wenn Sie es schon kennen, weil Sie es auf der rechten Seite der Gleichung einsetzen müssen. Da beißt sich die Katze in den Schwanz. . .

Mathematisch gesprochen handelt es sich um eine *implizite Gleichung* für  $y(x+h)$ . Sie lässt sich bei unserem Testproblem mit der einfachen Funktion  $f(x,y) = -y$  leicht auflösen.

Orientieren sie sich am Musterprogramm `GDGdemo.m` und implementieren Sie das implizite Euler-Verfahren für das Testproblem

$$\begin{aligned}y' &= -y \\ y(0) &= 1\end{aligned}$$

Untersuchen Sie die Stabilität dieses Verfahrens. Gibt es hier Schrittweiten  $h$ , für die die numerische Lösung anschwillt?