Kapitel 9

Numerische Methoden zur Lösung von gewöhnlichen Differentialgleichungen: Randwertprobleme.

9.1 Das lineare Randwertproblem zweiter Ordnung.

Es ist im Rahmen dieser Lehrveranstaltung völlig unmöglich, das sehr umfangreiche Gebiet der numerischen Behandlung von *Randwertproblemen* (RWP) umfassend zu behandeln. Ich habe daher aus der großen Mannigfaltigkeit der Probleme jene Gruppe herausgegriffen, die in der Praxis von Physik und Technik die wichtigste Rolle spielt, nämlich

lineare Randwertprobleme zweiter Ordnung mit entkoppelten Randbedingungen.

Dieses Problem besteht in mathematischer Hinsicht aus der gewöhnlichen linearen Differentialgleichung (Dgl.) zweiter Ordnung

$$r(x)\,y''(x) + s(x)\,y'(x) + q(x)\,y(x) = f(x) \qquad (x \in [a,b]) \tag{9.1}$$

mit den in bezug auf die Intervallgrenzen a und b *entkoppelten* Randbedingungen

$$\alpha_0\,y(a) + \alpha_1\,y'(a) = \gamma \qquad (|\alpha_0| + |\alpha_1| \neq 0) \tag{9.2}$$

und

$$\beta_0\,y(b) + \beta_1\,y'(b) = \delta \qquad (|\beta_0| + |\beta_1| \neq 0) \quad . \tag{9.3}$$

Man nennt ein Problem (9.1–9.3) homogen, wenn gilt:

$$f(x) \equiv 0 \qquad \gamma = 0 \qquad \delta = 0\,.$$

In allen anderen Fällen liegt ein inhomogenes RWP vor.

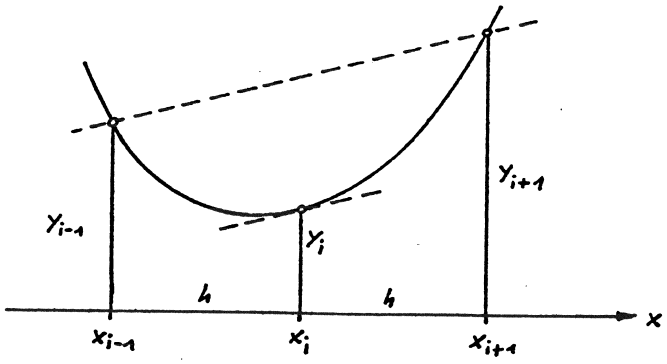


Abbildung 9.1: Grafische Interpretation des finiten Ausdrucks (9.5).

9.2 Numerische Behandlung des inhomogenen RWP mittels des Differenzenverfahrens.

Der wichtigste Schritt beim Differenzenverfahren besteht darin, daß das kontinuierliche Intervall, innerhalb dessen die Lösungsfunktion $y(x)$ gesucht wird, durch *Aufteilung in N (gleich-breite) Subintervalle* diskretisiert wird. Man erhält auf diese Weise die $N+1$ äquidistanten Stützpunkte

$$x_i = a + (i-1)h \qquad \text{mit } h = \frac{b-a}{N} \quad \text{und } (i = 1, \ldots, N+1)\,. \tag{9.4}$$

An diesen Stützpunkten werden nun alle in der Dgl. und in den Randbedingungen auftretenden *Differentialquotienten* durch *geeignete Differenzenquotienten* ersetzt.
Es gibt nun in der Literatur zahlreiche Formeln für solche Differenzenquotienten (s. z. B. [20], S. 267 und 267). In vielen Fällen sind jedoch die einfachsten derartigen Formeln die numerisch praktikabelsten. Aus diesem Grund werden in diesem Kapitel aussschließlich die sogenannten 'symmetrischen Dreipunktsformeln'[1]

$$y_i' \approx \frac{y_{i+1} - y_{i-1}}{2h} \tag{9.5}$$

bzw.

$$y_i'' \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \tag{9.6}$$

verwendet.
Die grafische Interpretation von (9.5) und (9.6) ist besonders einfach: Die Steigung der Kurve im Punkt x_i wird approximiert durch die Steigung der Verbindungsgeraden, die durch den rechten und den linken Nachbarpunkt hindurchgeht (s. Abb. 9.1).
Entsprechend wird die zweite Ableitung der Funktion $y(x)$ im Punkt x_i durch die zweite Ableitung jener Parabel angenähert, die durch den Punkt (x_i, y_i) selbst sowie durch seinen rechten und linken Nachbarn hindurchgeht.

[1] Im folgenden wird stets die Nomenklatur $y(x_i) \equiv y_i$ etc. verwendet.

Ersetzt man nun die erste und zweite Ableitung in (9.1) durch die Ausdrücke (9.5) und (9.6), so erhält man für $i = 1, \ldots, N+1$ die *Differenzengleichungen*

$$r_i \frac{(y_{i+1} - 2y_i + y_{i-1})}{h^2} + s_i \frac{(y_{i+1} - y_{i-1})}{2h} + q_i y_i = f_i$$

bzw.

$$\left(\frac{r_i}{h^2} - \frac{s_i}{2h}\right) y_{i-1} + \left(q_i - \frac{2r_i}{h^2}\right) y_i + \left(\frac{r_i}{h^2} + \frac{s_i}{2h}\right) y_{i+1} = f_i \,. \tag{9.7}$$

Dieses System von Differenzengleichungen bildet ein lineares, inhomogenes Gleichungssystem für die unbekannten Funktionswerte der Lösungsfunktion des RWP. an den $N+1$ Stützpunkten.
Allerdings ist das System (9.7) noch unterbestimmt, denn es enthält (in der ersten Gleichung) die Unbekannte y_0 sowie (in der $(N+1)$ten Gleichung) die Unbekannte y_{N+2}. Diese beiden Größen können, wie gleich gezeigt wird, mittels der beiden Randbedingungen (9.2) und (9.3) aus dem System eliminiert werden.

Einbeziehung der ersten Randbedingung:

Diese Randbedingung beschreibt das Verhalten der Lösungsfunktion an der Stelle $x = a$ d.h. für $i = 1$:

$$y(a) \equiv y_1 \qquad \text{und} \qquad y'(a) \equiv y_1' \approx \frac{y_2 - y_0}{2h}$$

bzw.

$$\alpha_0 y_1 + \alpha_1 \frac{y_2 - y_0}{2h} = \gamma \,.$$

Diese Gleichung wird nach y_0 aufgelöst, und man erhält *unter der Voraussetzung* $\alpha_1 \neq 0$

$$y_0 = \frac{1}{\alpha_1} \left(2h\alpha_0 y_1 + \alpha_1 y_2 - 2h\gamma\right) \,. \tag{9.8}$$

y_0 setzt man in die erste Differenzengleichung von (9.7) ein, also in die Gleichung

$$\left(\frac{r_1}{h^2} - \frac{s_1}{2h}\right) y_0 + \left(q_1 - \frac{2r_1}{h^2}\right) y_1 + \left(\frac{r_1}{h^2} + \frac{s_1}{2h}\right) y_2 = f_1 \,.$$

Dies ergibt die neue erste Differenzengleichung

$$\left[q_1 - \frac{2r_1}{h^2} - \frac{\alpha_0}{\alpha_1}\left(s_1 - \frac{2r_1}{h}\right)\right] y_1 + \frac{2r_1}{h^2} y_2 = f_1 - \frac{\gamma}{\alpha_1}\left(s_1 - \frac{2r_1}{h}\right) \,. \tag{9.9}$$

Dies gilt, wie bereits gesagt, für $\alpha_1 \neq 0$. Im Falle $\alpha_1 = 0$ ergibt sich direkt aus der ersten Randbedingung die Gleichung

$$y_1 = \frac{\gamma}{\alpha_0} \,. \tag{9.10}$$

Einbeziehung der zweiten Randbedingung:

Die Behandlung der Randbedingung (9.3) geschieht äquivalent zu den obigen Überlegungen und führt unter der Bedingung $\beta_1 \neq 0$ zur folgenden neuen *letzten* (d. h. $(N+1)$-ten) Differenzengleichung

$$\frac{2r_{N+1}}{h^2}y_N + \left[q_{N+1} - \frac{2r_{N+1}}{h^2} - \frac{\beta_0}{\beta_1}\left(s_{N+1} + \frac{2r_{N+1}}{h}\right)\right] y_{N+1} = f_{N+1} - \frac{\delta}{\beta_1}\left(s_{N+1} + \frac{2r_{N+1}}{h}\right). \tag{9.11}$$

Im Falle $\beta_1 = 0$ ergibt sich aus (9.3) sofort

$$y_{N+1} = \frac{\delta}{\beta_0}. \tag{9.12}$$

Auf diese Weise hat man das gegebene RWP in ein inhomogenes, lineares Gleichungssystem von $N+1$ Gleichungen für die $N+1$ Unbekannten $y_1, y_2, \ldots, y_{N+1}$.
Die erste Gleichung dieses Systems ist gegeben durch (9.9) bzw. (9.10), die zweite bis N-te Gleichung hat die Form (9.7), und die $(N+1)$-te Gleichung lautet (9.11) bzw. (9.12).

Wie bereits erwähnt, stellen die Lösungen dieses Systems die approximativen Werte der Lösungsfunktion $y(x)$ des RWP an den Stützpunkten $x_1 \ldots x_{N+1}$ dar. Dieses System muß also (i. a. mit numerischen Mitteln) gelöst werden.
Eine Analyse der Differenzengleichungen zeigt Ihnen sofort, daß die Koeffizientenmatrix des linearen Gleichungssystems eine *tridiagonale* Matrix ist. Diese einfache Form ist der speziellen Wahl der finiten Elemente (9.5) und (9.6) zu danken!

Wie Sie sich erinnern, gibt es für die Lösung von linearen Gleichungssystemen mit tridiagonaler Matrix sehr effizente numerische Methoden. Eine solche finden Sie im Abschnitt 2.6.1 dieses Skriptums.

9.2.1 Fehlerdiagnostik und Fehlerkorrektur.

Wiederholt man die Berechnung der y_i mit der *doppelten Zahl von Subintervallen* (d.h.: $N \to 2N$), so entspricht wegen der äquidistanten Stützpunkte jeder zweite Punkt der '$2N$-Rechnung' einem Punkt der 'N-Rechnung'. Für diese Punkte kann man nun eine sehr einfache und äquidistante Reduktion des beim Differenzenverfahren auftretenden Verfahrensfehlers vornehmen; diese Methode beruht auf dem in der Literatur beschriebenen Faktum, daß der Verfahrensfehler für y_i die Form

$$E_V = \frac{C(N)}{N^2} \tag{9.13}$$

hat. Der (unbekannte) exakte Wert von y_i kann also in der Form

$$y_{i,exakt} = \hat{y}_i(N) + \frac{C_i(N)}{N^2}$$

geschrieben werden, wobei $\hat{y}(N)$ den mittels des Differenzenverfahrens (N Subintervalle) erhaltenen Näherungswert bedeutet.
Verdoppelt man die Anzahl der Subintervalle, so ergibt sich

$$y_{i,exakt} = \hat{y}_i(2N) + \frac{C_i(2N)}{(2N)^2} .$$

Wie im ganz ähnlichen Fall der Fehlerabschätzung beim Runga-Kutta-Verfahren (s. Abschnitt 8.4.5), macht man auch hier die Approximation

$$C_i(N) \approx C_i(2N) = C_i$$

und man erhält sofort

$$C_i = \frac{4N^2}{3} [\hat{y}_i(2N) - \hat{y}_i(N)]$$

bzw. die Korrekturformel

$$y_{i,korr} = \frac{4\hat{y}_i(2N) - \hat{y}_i(N)}{3} . \tag{9.14}$$

Die Leistungsfähigkeit dieser Formel wird in den folgenden Testbeispielen demonstriert.

9.2.2 Das Programm DIFF1.

Dieses Programm löst das RWP (9.1)–(9.3) mittels des Differenzenverfahrens. Die Hauptpunkte des Programmes sind:

1. Berechnung der Funktionswerte von $r(x)$, $s(x)$, $q(x)$ und $f(x)$ für die Stützpunkte $x_i = a + (i-1)h$ mit $h = (b-a)/N$ und $i = 1, \ldots, N+1$ mittels der Prozedur **FCT1**.

2. Berechnung der 3 Vektoren $\mathbf{a}, \mathbf{b}, \mathbf{c}$ der tridiagonalen Matrix sowie des inhomogenen Vektors gemäß den Gleichungen (9.9; 9.10), (9.7) und (9.11;9.12).
   Achtung auf die Sonderfälle $\alpha_1 = 0$ und/oder $\beta_1 = 0$.

3. Aufruf des Programmes **TRID** (s. Abschnitt 2.6.1). Der Lösungsvektor $\mathbf{y}$ stellt die Näherungen der Lösungsfunktion $y(x)$ des RWP an den Stützpunkten x_i dar.

4. Die Schritte (1)–(3) werden zweimal durchgeführt, einmal mit der gewählten Intervallanzahl N, und ein zweitesmal mit der Intervallanzahl $2N$.

5. Die entsprechenden $\hat{y}$-Werte werden mittels der Fehlerkorrekturformel (9.14) verbessert.

INPUT-Parameter:

ANF, AEND: a und b in (9.1).

ALP0,ALP1,BET0,BET1: Parameter der entkoppelten Randbedingungen (9.2) und (9.3).

GAMMA, DELTA: inhomogene Parameter in (9.2) und (9.3).

N0: Anzahl der Subintervalle für *den ersten Durchlauf* in DIFF1.

OUTPUT-Parameter:

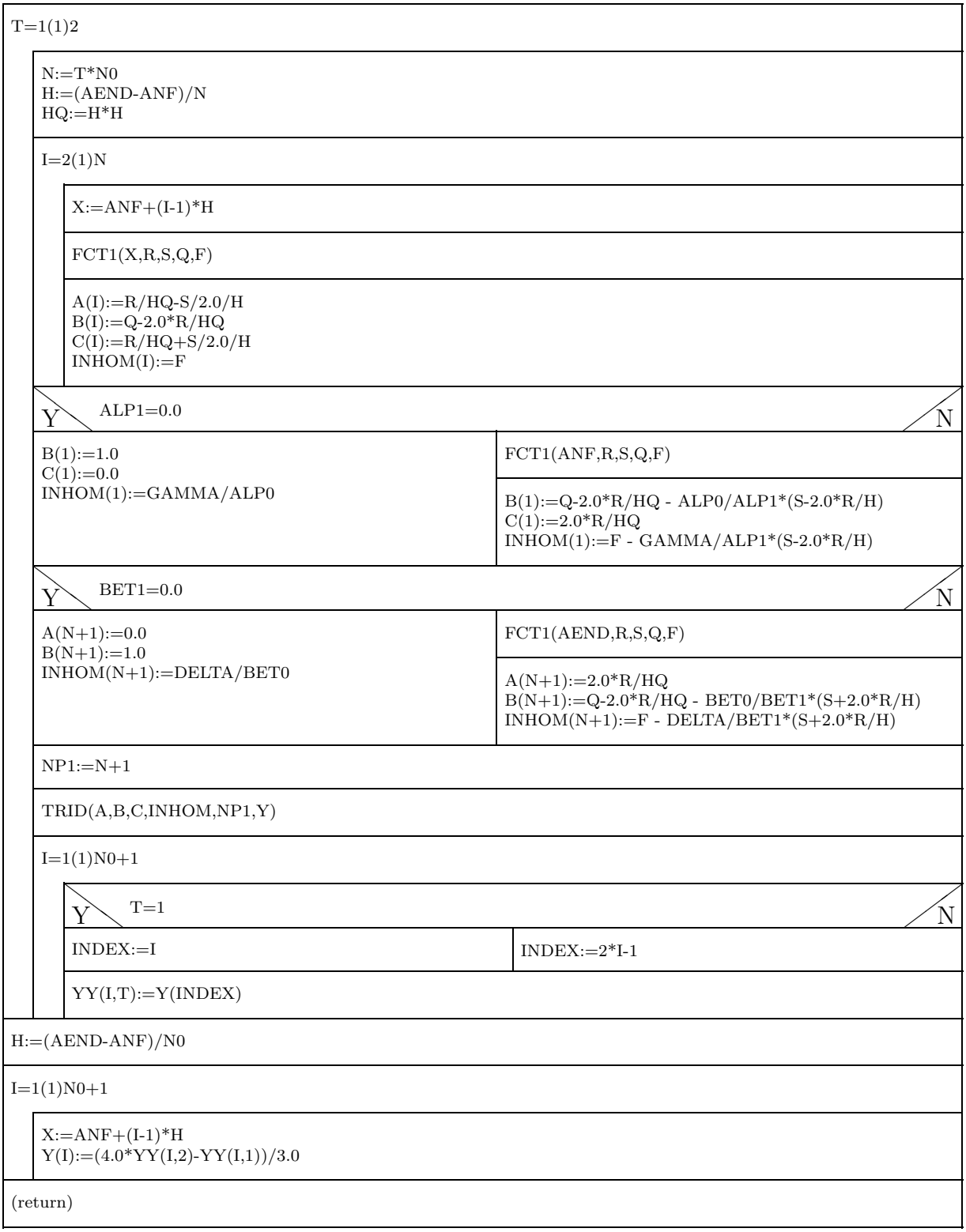
Y(): eindim. Feld mit den Näherungswerten der Lösungsfunktion $y(x)$ an den Stützpunkten im Intervall [a,b].

INTERNE FELDER:

YY(,): zweidim. Feld mit den Näherungswerten der Lösungsfunktion $y(x)$ für N und $2N$ Subintervalle.

Das Programm DIFF1 benötigt die Prozeduren FCT1 und TRID.

Struktogramm 29 — DIFF1(ANF,AEND,ALP0,ALP1,BET0,BET1,GAMMA, DELTA,N0,Y)



9.2.3 Ein Testbeispiel für DIFF1.

Das nun folgende Beispiel soll Ihnen die Leistungsfähigkeit von DIFF1 demonstrieren. Es ist der Referenz [19], S. 252f entnommen und lautet:

$$y''(x) - 2xy'(x) - 2y = -4x \qquad (x \in [0,1])$$

und

$$y(0) - y'(0) = 0 \qquad \text{bzw.} \qquad y(1) = 1 + \mathrm{e}\,.$$

Die exakte Lösung dieses RWP lautet:

$$y(x) = x + \mathrm{e}^{x^2}\,.$$

Der Benutzer hat also eine Prozedur FCT1 zu schreiben, welche z. B. die Form hat:

```
.
PROCEDURE FCT1(x: real; VAR r,s,q,f: real);

BEGIN
  r:=1.0;
  s:=-2.0*x;
  q:=-2.0;
  f:=-4.0*x
END;
```

Die weiteren Parameter lauten:

```
.
  anf=0.0    aend=1.0

  alp0=1.0   alp1=-1.0     bet0=1.0   bet1=0.0
        gamma=0.0              delta=1.0 + e
```

Die Auswertung mittels DIFF1 wurde für N0=10 Subintervalle durchgeführt, d.h. das Programm DIFF1 führte die Rechnung einmal mit 10 Subintervallen und ein zweitesmal mit 20 Subintervallen durch; dann wurde die Korrekturformel (9.14) angewendet.
Die Ergebnisse sehen Sie in den folgenden Tabellen sowie in der Abb. 9.2.

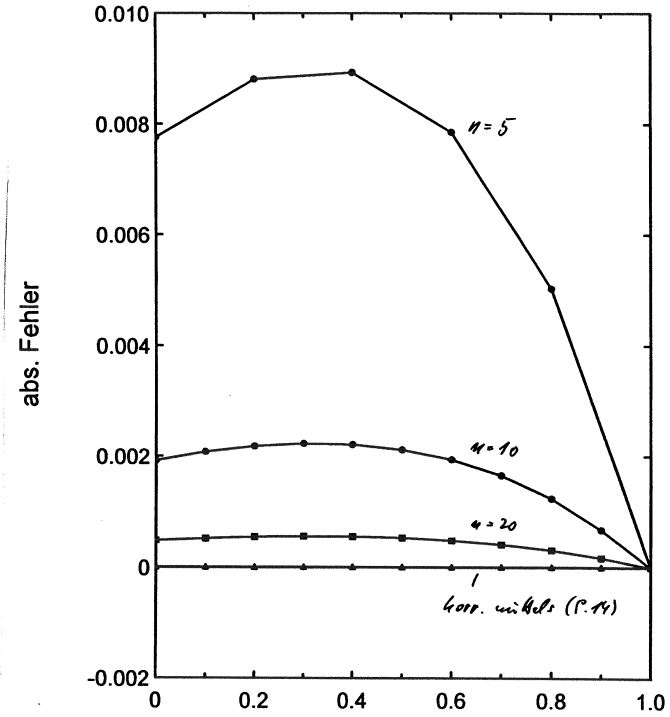


Abbildung 9.2: Absolute Fehler der Näherungslösungen des Testbeispiels für DIFF1.

.

Testergebnisse fuer DIFF1: absolute Fehler zwischen Naeherungswerten und exakten Werten:

x	abs. Fehler n=10	n=20	korr. mittels (9.14)
0.0	1.9162406497E-03	4.7755794913E-04	-2.0029510779E-06
0.1	2.0768600371E-03	5.1777355657E-04	-1.9219369278E-06
0.2	2.1803285636E-03	5.4365274082E-04	-1.9058661564E-06
0.3	2.2261855847E-03	5.5508881815E-04	-1.9434355636E-06
0.4	2.2088322276E-03	5.5069313748E-04	-2.0198913262E-06
0.5	2.1176171067E-03	5.2782016428E-04	-2.1121486498E-06
0.6	1.9371537346E-03	4.8265275109E-04	-2.1809100872E-06
0.7	1.6485163578E-03	4.1051038716E-04	-2.1582709451E-06
0.8	1.2326037795E-03	3.0670399065E-04	-1.9292747311E-06
0.9	6.7823138306E-04	1.6857800802E-04	-1.3064491213E-06
1.0	1.0913936421E-11	1.0913936421E-11	1.0913936421E-11

9.3 Numerische Behandlung des homogenen RWP mittels des Differenzenverfahrens.

Im folgenden soll nun die numerische Auswertung des *homogenen* RWP (9.1)–(9.3) diskutiert werden, und zwar in der Form eines Eigenwertproblems:

$$r(x)\,y''(x) + s(x)\,y'(x) + q(x)\,y(x) = \lambda\,y(x) \qquad (x \in [a,b]) \tag{9.15}$$

mit den Randbedingungen

$$\alpha_0\,y(a) + \alpha_1\,y'(a) = 0 \qquad (|\alpha_0| + |\alpha_1| \neq 0) \tag{9.16}$$

und

$$\beta_0\,y(b) + \beta_1\,y'(b) = 0 \qquad (|\beta_0| + |\beta_1| \neq 0) \quad . \tag{9.17}$$

Für ein solches Problem gibt es immer die triviale Lösung $y(x) \equiv 0$ sowie *für bestimmte Werte von* λ *nicht-triviale Lösungen*. Man spricht von den Eigenwerten und Eigenfunktionen (Eigenlösungen) des homogenen RWP.

Die mathematische Behandlung von (9.15)–(9.17) ist identisch mit der Vorgangsweise in Abschnitt 9.2. Wieder wird das RWP durch Diskretisierung des Intervalles $[a,b]$ in ein lineares, diesmal natürlich *homogenes* Gleichungssystem umgewandelt. Unter der Voraussetzung $\alpha_1 \neq 0$ und $\beta_1 \neq 0$ erhält man

$$\left[\underbrace{q_1 - \frac{2r_1}{h^2} - \frac{\alpha_0}{\alpha_1}\left(s_1 - \frac{2r_1}{h}\right)}_{b_1} - \lambda\right] y_1 + \underbrace{\frac{2r_1}{h^2}}_{c_1}\, y_2 = 0\,, \tag{9.18}$$

für $(i = 2, \ldots, N)$

$$\underbrace{\left(\frac{r_i}{h^2} - \frac{s_i}{2h}\right)}_{a_i}\, y_{i-1} + \left(\underbrace{q_i - \frac{2r_i}{h^2}}_{b_i} - \lambda\right)\, y_i + \underbrace{\left(\frac{r_i}{h^2} + \frac{s_i}{2h}\right)}_{c_i}\, y_{i+1} = 0 \tag{9.19}$$

und

$$\underbrace{\frac{2r_{N+1}}{h^2}}_{a_{N+1}}\, y_N + \left[\underbrace{q_{N+1} - \frac{2r_{N+1}}{h^2} - \frac{\beta_0}{\beta_1}\left(s_{N+1} + \frac{2r_{N+1}}{h}\right)}_{b_{N+1}} - \lambda\right]\, y_{N+1} = 0\,. \tag{9.20}$$

Auf diese Weise hat man das Eigenwertproblem des RWP in das Eigenwertproblem einer tridiagonalen Matrix umgewandelt:

$$\begin{pmatrix} b_1 - \lambda & c_1 & & & & \\ a_2 & b_2 - \lambda & c_2 & & & \\ & & \cdot & & & \\ & & \cdot & & & \\ & & \cdot & & & \\ & & a_N & b_N - \lambda & c_N & \\ & & & a_{N+1} & b_{N+1} - \lambda & \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_N \\ y_{N+1} \end{pmatrix} = 0\,. \tag{9.21}$$

Das Auftreten von $\alpha_1 = 0$ und/oder $\beta_1 = 0$ bedeutet eine harmlose Komplikation: in diesem Fall gehen die Randbedingungen (9.16) und/oder (9.17) in die simple Form

$$y(a) \equiv y_1 = 0 \qquad \text{und/oder} \qquad y(b) \equiv y_{N+1} = 0$$

über, d. h., daß die erste und/oder letzte Komponente des Lösungsvektors bereits feststeht *und daher in (9.21) nicht mehr als Unbekannte aufscheinen muß.*
In einem solchen Fall muß die Koeffizientenmatrix in (9.21) *um die erste und/oder letzte Zeile und Spalte verjüngt werden!* Wie dies im Detail geschieht, können Sie dem Struktogramm Nr. 30 entnehmen.

In jedem Fall hat man nun die Eigenwerte einer tridiagonalen Matrix zu ermitteln. Dies kann z. B. mittels des *Hyman-Verfahrens* in Zusammenarbeit mit einem Nullstellen-Suchprogramm wie INTSCH geschehen. Genaueres zu diesem Thema finden Sie im Abschnitt 7.5.5 dieses Skriptums.

9.3.1 Fehlerkorrektur der Eigenwerte.

Ohne weitere theoretische Ableitungen soll hier eine Formel für die Fehlerkorrektur der so erhaltenen Näherungen der Eigenwerte des RWP angegeben werden. Bezeichnet man mit $\lambda(N)$ bzw. mit $\lambda(2N)$ einen Eigenwert der tridiagonalen Matrix in (9.21), wobei das Intervall $[a, b]$ in N bzw. $2N$ Subintervalle geteilt wurde, so kann der Verfahrensfehler bzgl. dieses Eigenwertes mittels der Formel

$$\lambda_{korr} = \frac{4\lambda(2N) - \lambda(N)}{3} \tag{9.22}$$

signifikant reduziert werden (vgl. Abschnitt 9.2.1).

Es soll hier aber ein ganz wesentlicher Nachteil dieser Korrekturformel nicht verschwiegen werden: man hat keinerlei Information über die Genauigkeit der korrigierten Eigenwerte!

9.3.2 Das Programm DIFF2.

Dieses Programm berechnet einige reelle Eigenwerte des homogenen RWP (9.15)–(9.17) mittels des Differenzenverfahrens sowie des Verfahrens von Hyman (s. Abschnitt 7.5.5).
Die Hauptpunkte des Programmes sind:

1. Zählschleife mit T=1(1)2: Um die Korrekturformel (9.22) anwenden zu können, muß die Eigenwertbestimmung *zweimal* durchgeführt werden, und zwar sowohl mit der gegebenen Subintervallanzahl N0 als auch mit der doppelten Subintervallanzahl.

2. Bestimmung der Funktionswerte der Koeffizientenfunktionen $r(x)$, $s(x)$ und $q(x)$ durch Aufruf der Prozedur **FCT1**.

3. Berechnung der 3 Vektoren der tridiagonalen Matrix in (9.21) *unter Berücksichtigung eines eventuellen Nullwerdens von α_1 oder/und β_1.*

4. Aufruf des Nullstellen-Suchprogrammes **INTSCH** (Abschnitt 5.5). Dieses Programm ruft die HYMAN-Funktion auf, wobei die drei Vektoren der tridiagonalen Matrix [die Felder A(), B() und C()] zusammen mit der aktuellen Ordnung der Matrix an die Funktion HYMAN übergeben werden müssen, gegebenenfalls durch die Definition *globaler Parameter* (Details zum Zusammenwirken von HYMAN und INTSCH siehe Abschnitt 7.5.5).
   INTSCH liefert die 'unkorrigierten' Eigenwerte. Die Anzahl der beim ersten bzw. zweiten Durchlauf gefundenen Eigenwerte ist ANZ(1) bzw. ANZ(2). Abspeicherung der unkorrigierten Eigenwerte auf das Feld EWERTE(...,T).

5. Nach dem zweiten Durchlauf (mit doppelter Subintervallanzahl) werden die entsprechenden Eigenwerte mittels (9.22) verbessert. *Dabei muß sichergestellt sein, daß nur zusammengehörende Eigenwerte in die Formel (9.22) eingesetzt werden. Um dies mit ausreichender Sicherheit zu gewährleisten, wird vor der Fehlerkorrektur abgefragt, ob in beiden Rechendurchläufen dieselbe Zahl von Eigenwerten eruiert wurde.* Ist dies nicht der Fall, Fehleranzeige und Return.

INPUT-Parameter:

ANF,AEND: a und b in (9.15).

ALP0,ALP1,BET0,BET1: Parameter α_0, α_1, β_0 und β_1 in den Randbedingungen (9.16) und (9.17).

Die folgenden 4 Input-Parameter beziehen sich auf die von DIFF2 aufgerufene Prozedur INTSCH:

ANFEIG,ENDEIG: Anfang und Ende des Grobsuchbereiches für die Eigenwerte.

HEIG: Schrittweite für die Eigenwertsuche.

GENEIG: Fehlerschranke für die von INTSCH ermittelten Eigenwerte.

N0: Anzahl der Subintervalle für den ersten Durchlauf von DIFF2.

OUTPUT-Parameter:

ANZEIG: Zahl der ermittelten (korrigierten) Eigenwerte.

EIGW( ): Feld der ermittelten (korrigierten) Eigenwerte.

FEHL1: (logische) Fehlerdiagnostik-Variable:
FEHL1=TRUE: Eigenwert-Korrektur war nicht möglich;
FEHL1=FALSE: Eigenwert-Korrektur ok.

INTERNE FELDER:

EWERTE(,): zweidimensionales Feld mit den unkorrigierten Eigenwerten für N0 bzw. 2 N0 Subintervalle.

Das Programm DIFF2 benötigt die Prozeduren FCT1, INTSCH und HYMAN.

Anmerkung:
Um die Verfahrensfehler von INTSCH möglichst klein zu halten, sollte die Fehlerschranke GENEIG sehr klein gewählt werden,
z.B.: GEN = 0.0000001.
Um dieser Anforderung gerecht zu werden, sollte DIFF2 – wenn möglich – mit *doppelter Genauigkeit* gerechnet werden.

Struktogramm 30 — DIFF2(ANF,AEND,ALP0,ALP1,BET0,BET1,ANFEIG, ENDEIG,HEIG,GENEIG,N0,ANZEIG,EIGW,FEHL1)

T=1(1)2

N:=T*N0
H:=(AEND-ANF)/N
HQ:=H*H

I=2(1)N

X:=ANF+(I-1)*H

FCT1(X,R,S,Q)

A(I):=R/HQ-S/2.0/H
B(I):=Q-2.0*R/HQ
C(I):=R/HQ+S/2.0/H

ALP1=0.0

Y:

IANF:=2

N:

IANF:=1

FCT1(ANF,R,S,Q)

B(1):=Q-2.0*R/HQ - ALP0/ALP1*(S-2.0*R/H)
C(1):=2.0*R/HQ

BET1=0.0

Y:

IEND:=N

N:

IEND:=N+1

FCT1(AEND,R,S,Q)

A(N+1):=2.0*R/HQ
B(N+1):=Q-2.0*R/HQ - BET0/BET1*(S+2.0*R/H)

N:=IEND-IANF+1

ALP1=0.0

Y:

I=1(1)N

A(I):=A(I+1)
B(I):=B(I+1)
C(I):=C(I+1)

N:

.....

INTSCH(ANFEIG,ENDEIG,HEIG,GENEIG,ANZMAX,EIGW,ANZ(T))

ANZ(T)=0

Y:

print: DIFF2 'keine Nullst. in INTSCH'

N:

I=1(1)ANZ(T)

EWERTE(I,T):=EIGW(I)

ANZEIG:=ANZ(2)

ANZ(1)=ANZ(2)

Y:

I=1(1)ANZEIG

EIGW(I):=(4.0*EWERTE(I,2) - EWERTE(I,1))/3.0

FEHL1:=FALSE

N:

print: DIFF2 'Keine Korrektur'

I=1(1)ANZEIG

EIGW(I):=EWERTE(I,2)

FEHL1:=TRUE

(return)

9.3.3 Ein Testbeispiel für DIFF2.

Als Test für die Leistungsfähigkeit des eben besprochenen Algorithmus soll nun das lineare, homogene EWP zweiter Ordnung

$$\begin{aligned} -y''(\vartheta) - \frac{\cos\vartheta}{\sin\vartheta}\, y'(\vartheta) &= \lambda\, y(\vartheta) \\ . & \\ y(\pi/2) = 0 \qquad\qquad & y'(\pi) = 0 \end{aligned} \tag{9.23}$$

mittels DIFF2 ausgewertet werden.

Die Eigenwerte λ dieses Problems gehorchen der Formel

$$\lambda_m = (2m-1)\cdot 2m \qquad m = 1, 2, \ldots \tag{9.24}$$

und die entsprechenden Eigenfunktionen sind die Legendre-Polynome 1. Art mit ungeraden Ordnungen.

Im Bereich $0 \le \lambda \le 100$ gibt es also die 5 Eigenwerte

$$2 \qquad 12 \qquad 30 \qquad 56 \qquad 90\,.$$

Die INPUT-Parameter und die Prozedur FCT1 lauten in diesem Fall:

```
.
anf:=1.57079633    aend:=3.14159265

alp0:=1.0   alp1:=0.0   bet0:=0.0   bet1:=1.0

n0:=50

anfeig:=0.0    endeig:=100.0    heig:=1.0    geneig:=0.0000001



PROCEDURE FCT1(x: double; VAR r,s,q: double);

BEGIN
  r:=-1.0;
  IF(abs(x-PI) < 1.e-15)THEN s:=0.0 ELSE s:=-cos(x)/sin(x);
  q:=0.0;
END;
```

Anmerkung zu FCT1:

Bei der Auswertung der Koeffizientenfunktion $s(x)$ ist darauf zu achten, daß sie für $x = \pi$ eine Singularität hat. In der obigen Routine wird s für diesen Fall einfach Null gesetzt. Dies ist aber nur deshalb möglich, weil der in der Dgl. (9.23) vorkommende Term

$$\frac{\cos\vartheta}{\sin\vartheta}\, y'(\vartheta)$$

für $\vartheta \to \pi$ **nicht** divergiert! Die Folge davon ist nämlich (was hier nicht im Detail bewiesen werden soll), daß der dritte Term des in (9.20) definierten Elementes b_{N+1},

$$-\frac{\beta_0}{\beta_1}\left(s_{N+1}+\frac{2r_{N+1}}{h}\right),$$

für $\beta_0 = 0$ verschwindet.

Löst man das vorliegende Problem mit Hilfe der im vorigen Abschnitt präsentierten Prozedur DIFF2, so erhält man das folgende Ergebnis:

```
.
 Zahl der Subintervalle <=    50
 GENEIG = 0.0000001

          N=50       N=100     korr. nach    exakt
                                 (9.22)

    1    1.99901    1.99975    2.00000         2
    2   11.97850   11.99463   12.00001        12
    3   29.88705   29.97181   30.00006        30
    4   55.64148   55.91053   56.00021        56
    5   89.12707   89.78212   90.00047        90
```

9.4 Die 'shooting method'.

Anschließend soll hier noch die in der Praxis häufig verwendete Methode besprochen werden, das homogene RWP (9.15–9.17) durch Überführung in ein *Anfangswertproblem* zu lösen.
Zu diesem Zweck bringt man (9.15) in die Form

$$y''(x) = -\frac{s(x)}{r(x)}\,y'(x) - \frac{[q(x)-\lambda]}{r(x)}\,y(x) \tag{9.25}$$

für $x \in [a,b]$.

Die Anfangsbedingungen:

Man benötigt für die Dgl. *zweiter Ordnung* natürlich zwei Anfangsbedingungen. Diese folgen sofort aus der ersten Randbedingung (für $x = a$), welche die Form

$$\alpha_0\,y(a) + \alpha_1\,y'(a) = 0$$

hat. Im Fall $\alpha_1 \neq 0$ kann man nun für $y(a)$ *jede beliebige reelle Zahl (außer Null)* setzen. Dies ist möglich, weil die Lösungsfunktion eines homogenen Problems *stets nur bis auf eine multiplikative Konstante* fixiert ist. Man kann also z. B. einfach festsetzen:

$$y(a) = 1.0\,. \tag{9.26}$$

Der Anfangswert von $y'(x)$ ergibt sich dann sofort als

$$y'(a) = -\frac{\alpha_0}{\alpha_1}y(a) = -\frac{\alpha_0}{\alpha_1}\,. \tag{9.27}$$

Ist hingegen $\alpha_1 = 0$, so bleibt für $y(x)$ nur mehr die Möglichkeit

$$y(a) = 0.0\,, \tag{9.28}$$

und die Funktion $y'(x)$ kann am Anfangswert jeden beliebigen Wert außer Null annehmen, also z. B.

$$y'(a) = 1.0\,. \tag{9.29}$$

Die Gleichungen (9.25)–(9.29) definieren ein Anfangswertproblem, das z. B. mittels der im Kapitel 8 besprochenen Verfahren gelöst werden kann.

Bestimmung der Eigenwerte:

Die Eigenwerte λ sind noch unbekannt. Man kann nun wie folgt vorgehen: Man löst das Cauchy-Problem (9.25)–(9.29) im Intervall $[a,b]$ für ein beliebiges 'Probe'–λ.

Nun überprüft man, ob die erhaltene Lösungsfunktion $y(x)$ an der Stelle des rechten Randes, also für $x = b$, die zweite Randbedingung (9.17) erfüllt, d.h. ob gilt:

$$\beta_0\,y(b) + \beta_1\,y'(b) = 0\,. \tag{9.30}$$

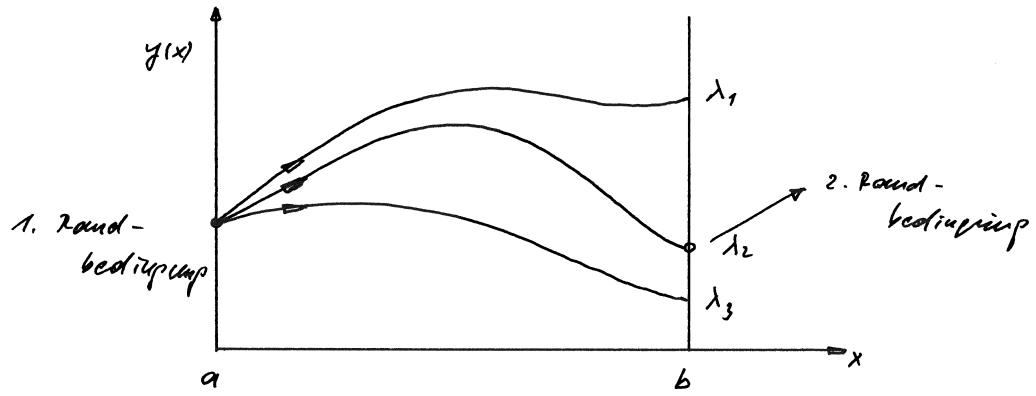


Abbildung 9.3: Das Grundprinzip der *shooting method*.

Wenn dies der Fall ist, ist das gewählte λ einer der Eigenwerte des gegebenen RWP, und $y(x)$ ist (eine Näherung) für die zugehörige Eigenfunktion. Wenn die Bedingung (9.30) nicht erfüllt wird (was natürlich i. a. der Fall sein wird), muß die Auswertung des Anfangswertproblems mit einem neuen Probewert für λ wiederholt werden.
Abb. 9.3 zeigt das Prinzip dieser Methode: λ_1 und λ_3 'treffen daneben', λ_2 'trifft' die Randbedingung bei b und stellt daher einen Eigenwert dar. Die Formulierung 'treffen' bzw. 'daneben treffen' ist bewußt gewählt, um Ihnen den Namen dieser Methode (*shooting method*) klar zu machen.

Die *shooting method* hat jedoch einen gravierenden Nachteil: um das Eigenwertproblem mit genügender Genauigkeit zu lösen, müssen sehr viele 'Probe'–λ verwendet werden d. h. es müssen sehr viele Cauchy-Probleme gelöst werden, was natürlich häufig große Rechenzeiten bedeutet.

Aus diesem Grund wird die *shooting method* in der Praxis vor allem in Kombination mit einer sehr schnellen und dennoch genauen Lösungsmethode für das Cauchy-Problem kombiniert, die auf Numerov zurückgeht.

9.4.1 Die Numerov-Methode.

Diese Methode ist nur anwendbar, wenn es gelingt, die Dgl. des homogenen RWP auf die Form

$$y''(x) + k(x)\, y(x) = 0 \qquad x \in [a, b] \tag{9.31}$$

zu bringen. Dies ist für viele wichtige Anwendungsfälle in Physik und Technik möglich. Ein gutes Beispiel ist etwa die eindimensionale, zeitunabhängige (stationäre) *Schrödingergleichung*, welche bekanntlich die Form

$$\psi''(x) + \frac{2m}{\hbar^2}\,[E - V(x)]\;\psi(x) = 0$$

hat. In diesem Falle wäre die in (9.31) definierte Funktion $k(x)$ also durch

$$k(x) = \frac{2m}{\hbar^2}\,[E - V(x)]$$

bzw., bei Verwendung geeigneter Längen- und Energie-Einheiten, durch

$$k(x) = E - V(x) \tag{9.32}$$

gegeben.

Die Numerov-Methode ist wieder eine Differenzenmethode, d.h. man teilt das gegebene Intervall in N Subintervalle mit den Stützpunkten

$$x_i = a + (i-1)h \qquad (i = 1, \ldots, N+1) \qquad \text{mit} \quad h = \frac{b-a}{N}\,.$$

Der nächste Schritt ist eine Taylorreihenentwicklung der Funktion $y(x)$ an der Stelle x_i für die Werte $x_i \pm h$:

$$y(x_i \pm h) \equiv y_{i\pm1} = y_i \pm h\, y_i' + \frac{h^2}{2}\, y_i'' \pm \frac{h^3}{6}\, y_i^{(3)} + \frac{h^4}{24}\, y_i^{(4)} \pm \cdots$$

Daraus folgt sofort

$$y_{i+1} + y_{i-1} = 2y_i + h^2\, y_i'' + \frac{h^4}{12}\, y_i^{(4)} + \cdots \tag{9.33}$$

Für y_i'' kann man nun gemäß (9.31) setzen:

$$y_i'' = -k_i\, y_i\,. \tag{9.34}$$

Nun kommt der entscheidende Trick: man diskretisiert die vierte Ableitung von $y(x)$ an der Stelle x_i mittels des finiten Elementes (9.6) und erhält mit (9.34)

$$y_i^{(4)} \approx \frac{y_{i+1}'' - 2y_i'' + y_{i-1}''}{h^2} = \frac{2k_iy_i - k_{i+1}y_{i+1} - k_{i-1}y_{i-1}}{h^2}\,. \tag{9.35}$$

Setzt man nun (9.34) und (9.35) in die Glg. (9.33) ein und löst diese Gleichung nach y_{i+1} auf, ergibt sich die folgende *Rekursionsformel von Numerov*:

$$y_i \approx \frac{2\left(1 - \frac{5h^2}{12}k_{i-1}\right) y_{i-1} - y_{i-2} - \frac{h^2}{12}k_{i-2}y_{i-2}}{1 + \frac{h^2}{12}k_i} \tag{9.36}$$

für $i = 3, \ldots, N$, wobei die Werte k_i die Funktionswerte von $k(x)$ [Glg. (9.32)] an den Stützpunkten x_i bedeuten:

$$k_i \equiv k(x_i) = E - V(x_i) \qquad \text{mit} \quad i = 1, \ldots, N+1\,. \tag{9.37}$$

Nun noch einige wichtige Anmerkungen zum shooting-Numerov-Verfahren:

Als ersten Schritt des Numerov-Prozesses muß die Glg. (9.36) für $i = 3$ ausgewertet werden. Dies führt zur Formel

$$y_3 \approx \frac{2\left(1 - \frac{5h^2}{12}k_2\right) y_2 - y_1 - \frac{h^2}{12}k_1y_1}{1 + \frac{h^2}{12}k_3}\,. \tag{9.38}$$

- Um den Numerov-Prozess starten zu können, muß man also die Werte $y_1 \equiv y(a)$ und $y_2 \equiv y(a+h)$ kennen. Diese Größen lassen sich gewöhnlich aus den Nebenbedingungen des Problems eruieren.

- In der obigen Formel für y_3 kommt die Größe $k_1 = E - V(a)$ vor. Man braucht also den Wert der Potentialfunktion am linken Rand a des Integrationsintervalls. In vielen konkreten Fällen - z.B. wenn $V(x)$ ein *Coulomb*-Potential ist - hat diese Funktion für $x = a$ eine Singularität und kann daher dort nicht ausgewertet werden.
  Dann kann man oft wie folgt vorgehen: man schreibt die Glg. (9.38) in der Form

$$y_3 \approx \frac{2\left(1 - \frac{5h^2}{12}k_2\right) y_2 - y_1 - \frac{h^2}{12}\left[Ey_1 - V(x_1)y_1\right]}{1 + \frac{h^2}{12}k_3}$$

  und berechnet analytisch den Grenzwert

$$\mathrm{V1Y1} = \lim_{x \to a} V(x)y(x)\,.$$

  Beispiele für ein solches Vorgehen finden Sie im Abschnitt 9.4.3 sowie in der Übungsbeschreibung.

- Wie aus den obigen Erläuterungen hervorgeht, muß für eine numerische Berechnung der Eigenwerte der Numerov-Prozess in der Regel *sehr oft* durchlaufen werden, wobei sich die verwendeten Werte von k_i nur durch einen neuen *trial value* für E unterscheiden, während die Potentialwerte $V(x_i)$ unverändert bleiben.

  Es wäre nun ausgesprochen ungeschickt, bei jedem Aufruf des *Numerov*-Programms die Potentialwerte $V(x_i)$ aufs Neue zu berechnen, denn wenn die Potentialfunktion kompliziert ist, würde das eine Menge Rechenzeit kosten. Viel besser ist es, das Feld der Werte $V(x_2)$, $V(x_3)$, ..., $V(x_{N+1}$ vor dem ersten *Numerov*-Aufruf ein für alle Mal zu berechnen und dieses Feld über die Parameterliste an *Numerov* zu übergeben. Eine solche Vorgangsweise ist in der folgenden Beschreibung des *Numerov*-Programms berücksichtigt.

9.4.2 Das Programm NUMEROV.

Die Prozedur NUMEROV führt eine numerische Integration der Dgl. (9.31) durch, wobei diese Gleichung den Eigenwert-Parameter λ enthält.

INPUT-Parameter:

ANF,AEND: a und b in (9.31).

Y1,Y2: Werte für $y(a)$ und $y(a+h)$.

V1Y1: Wert für $V(a)y(a)$ (s. dazu die Erläuterungen im vorigen bzw. im Abschnitt 9.4.3).

EIGW: Probewert für den Eigenwert (*shooting method!*).

POT: Vektorfeld mit den Potentialwerten $POT(2) = V(x_2)$, $POT(3) = V(x_3)$, ..., $POT(N+1) = V(x_{N+1})$.

N: Anzahl der Subintervalle in $[a, b]$.

OUTPUT-Parameter:

YF(): Näherungswerte der Lösungsfunktion $y(x)$ an den Stützpunkten $x_1, x_2, \ldots, x_{N+1}$.

Struktogramm 31 — NUMEROV(ANF,AEND,Y1,Y2,V1Y1,EIGW,POTF,N,YF)

```
H:=(AEND-ANF)/N
HH12:=H*H/12.0
YF(1):=Y1
YF(2):=Y2
YLL:=Y1
YL:=Y2
KL:=EIGW-POTF(2)
K:=EIGW-POTF(3)
Y:=(2.0*(1.0-5.0*HH12*KL)*YL – YLL – HH12*(EIGW*YLL-V1Y1)) / (1.0+HH12*K)
YF(3):=Y
I=4(1)N+1
    YLL:=YL
    YL:=Y
    KLL:=KL
    KL:=K
    K:=EIGW-POTF(I)
    Y:=(2.0*(1.0-5.0*HH12*KL)*YL – YLL – HH12*KLL*YLL) / (1.0+HH12*K)
    YF(I):=Y
(return)
```

9.4.3 Testbeispiel für die shooting-Numerov-Methode.

Im folgenden soll nun diese Methode auf ein Problem der Metallphysik angewendet werden. Gesucht ist die Grundzustandsenergie und die entsprechende Wellenfunktion eines Valenzelektrons in metallischem Natrium in der *Wigner-Seitz-Näherung*.

Die Wigner-Seitz-Näherung besteht darin, daß man die i. a. geometrisch recht komplizierte Form der Einheitszelle des Kristalls durch eine volumsgleiche Kugel mit dem Radius r_{WS} ersetzt. Innerhalb der Kugel wird das Kristallpotential durch ein radialsymmetrisches Potential angenähert:

$$V(\mathbf{r}) \approx V(r) \qquad \text{für } r \in [0, r_{WS}]\,.$$

Unter diesen Umständen sind die Energieeigenwerte E und die Eigenfunktionen eines quantenmechanischen Teilchens (Elektron) durch die *stationäre* Schrödingergleichung

$$y''(r) + \left\{ \frac{2m}{\hbar^2} \left[E - V(r) \right] - \frac{\ell(\ell+1)}{r^2} \right\} y(r) = 0$$

gegeben, wobei $y(r)$ der mit r multiplizierte *Radialteil der Wellenfunktion des Elektrons* ist. Der Parameter ℓ stellt die Bahndrehimpuls-Quantenzahl dar; im Falle der Natrium-Valenzelektronen gilt $\ell = 0$.

In atomaren Einheiten (Längen in Bohr, Energien in Rydberg) gilt außerdem $2m/\hbar^2 = 1$, und man erhält

$$y''(r) + k(r)\, y(r) = 0 \qquad \text{mit} \qquad k(r) = E - V(r) \tag{9.39}$$

für $r \in [0, r_{WS}]$.

Der Wigner-Seitz-Radius für das Natrium-Metall beträgt 3.9405 Bohr, und als Potential innerhalb der WS-Zelle verwendeten Wigner und Seitz in ihrer Originalarbeit das im folgenden angegebene Prokofjew-Potential:

$$V(r) = -\frac{2Q(r)}{r^2} \tag{9.40}$$

mit

$$Q(r) = \begin{cases} 11r & 0 \le r \le 0.01 \\ -26.4r^2 + 11.53r - 0.00264 & 0.01 \le r \le 0.15 \\ -2.84r^2 + 4.46r + 0.5275 & 0.15 \le r \le 1.00 \\ +1.508r^2 - 4.236r + 4.876 & 1.00 \le r \le 1.55 \\ 0.1196r^2 + 0.2072r + 1.319 & 1.55 \le r \le 3.30 \\ 0.0005r^2 + 0.9933r + 0.0222 & 3.30 \le r \le 6.74 \\ r & 6.74 \le r < \infty \end{cases} \tag{9.41}$$

.

```
FUNCTION POT(r: double): double;
VAR
  q : double;
BEGIN
  IF(r <= 0.01)THEN q:=11.0*r
  ELSE IF(r <= 0.15)THEN q:=-26.4*r*r+11.53*r-0.00264
  ELSE IF(r <= 1.00)THEN q:=-2.84*r*r+4.46*r+0.5275
  ELSE IF(r <= 1.55)THEN q:=1.508*r*r-4.236*r+4.876
  ELSE IF(r <= 3.30)THEN q:=0.1196*r*r+0.2072*r+1.319
  ELSE IF(r <= 6.74)THEN q:=0.0005*r*r+0.9933*r+0.0222
  ELSE q:=r;
  pot:=-2.0*q/r/r
END;
```

Die Randbedingungen:

Die Bedingungen, die eine Eigenfunktion $y(r)$ von (9.39) erfüllen muß, sind die folgenden:

- In der Nähe des linken Randpunktes (d. h. für $r << 1$) ist das Kristallpotential $V(r)$ praktisch identisch mit den Coulomb-Potential des Atomkerns mit der Ladungszahl Z (für unser Beispiel Natrium ist $Z = 11$). Das heißt, für den Bereich $r << 1$ kann die Dgl. (9.39) als

$$y''(r) + \left(E + \frac{2Z}{r}\right) y(r) = 0$$

geschrieben werden. Die Lösung dieses Problems hat für kleine r die einfache Form

$$y(r) \propto r\,.$$

Aus diesem Verhalten ergeben sich sofort die Startwerte für die Numerov-Rekursion, nämlich

$$y_1 = 0 \qquad \text{und} \qquad y_2 = h\,.$$

Nun noch ein wichtiges Detail:
Die Funktion $V(r)$ in (9.39) divergiert offenbar für $r = 0$ wegen

$$V(r) = -\frac{2Z}{r} \qquad \text{für} \quad r \to 0\,.$$

Das Produkt aus den Funktionen $V(r)$ und $y(r)$ bleibt jedoch auch für $r = 0$ beschränkt und ergibt

$$\lim_{r\to 0} V(r)y(r) = -2Z\,.$$

Dieser Wert ist unter dem Namen V1Y1 an die Prozedur NUMEROV zu übergeben (s. Abschnitt 9.4.2).

- Nun wird die *shooting method* angewendet, d. h. es wird die Variable λ solange variiert, bis die mittels NUMEROV erhaltene Funktion $y(x)$ auch die zweite Randbedingung an der rechten Grenze b (im konkreten Fall also für $r = r_{WS}$) erfüllt.
Die Randbedingung, welcher die Eigenfunktion des Valenzelektrons an der Oberfläche der Wigner-Seitz-Kugel genügen muß, ist die in der Theoretischen Festkörperphysik wohlbekannte *Bloch-Bedingung*

$$y(r_{WS}) - r_{WS}\, y'(r_{WS}) \stackrel{!}{=} 0\,.$$

Überträgt man diese Bedingung in die entsprechende Differenzengleichung, so ergibt sich die Bedingung

$$y_{N+1} - b\frac{(y_{N+2} - y_N)}{2h} \stackrel{!}{=} 0.0\,. \tag{9.42}$$

Die Inputgröße EIGW ist also solange zu variieren, bis (9.42) erfüllt wird.

Anmerkung: wie Sie sehen, braucht man zur Auswertung der Glg. (9.42) noch den Wert y_{N+2} rechts neben dem Ende des eigentlichen Integrationsintervalls. Um diesen Wert zu erhalten, müssen Sie im Programm *Numerov* die Zählschleife von I=4(1)N+1 auf I=4(1)N+2 erweitern, und auch die Felder POTF und YF müssen dementsprechend um 1 Speicherplatz erweitert werden.

Die folgende Tabelle zeigt die 'Grobsuche' für das konkrete Problem:

```
.
TESTRECHNUNG   SHOOTING-NUMEROV  1999

Wigner-Seitz-Radius [Bohr] = 3.9405
Zahl der Subintervalle =  400

  eigw        y(b)       b*y'(b)      diff (soll Null sein!)

-0.7500      0.3844      0.6288     -0.2444
-0.6500      0.3259      0.3886     -0.0627
-0.6400      0.3203      0.3664     -0.0461
-0.6300      0.3147      0.3445     -0.0298
-0.6200      0.3092      0.3229     -0.0138
-0.6100      0.3037      0.3017      0.0020
-0.6000      0.2982      0.2808      0.0175
-0.5900      0.2928      0.2601      0.0327
-0.5800      0.2875      0.2398      0.0476
-0.5700      0.2821      0.2198      0.0623
-0.5600      0.2769      0.2002      0.0767
-0.5500      0.2716      0.1808      0.0908
-0.4500      0.2215      0.0036      0.2179
```

Der gesuchte Energie-Eigenwert E für das Natrium-Valenzelektron liegt also im Bereich zwischen –0.62 und –0.61 Rydberg. Eine genauere Rechnung führt zum Ergebnis

$$E = -0.611277 \quad \text{Rydberg}\,.$$

Die Abb. 9.4 zeigt die entsprechende (unnormierte) Eigenfunktion y(x).

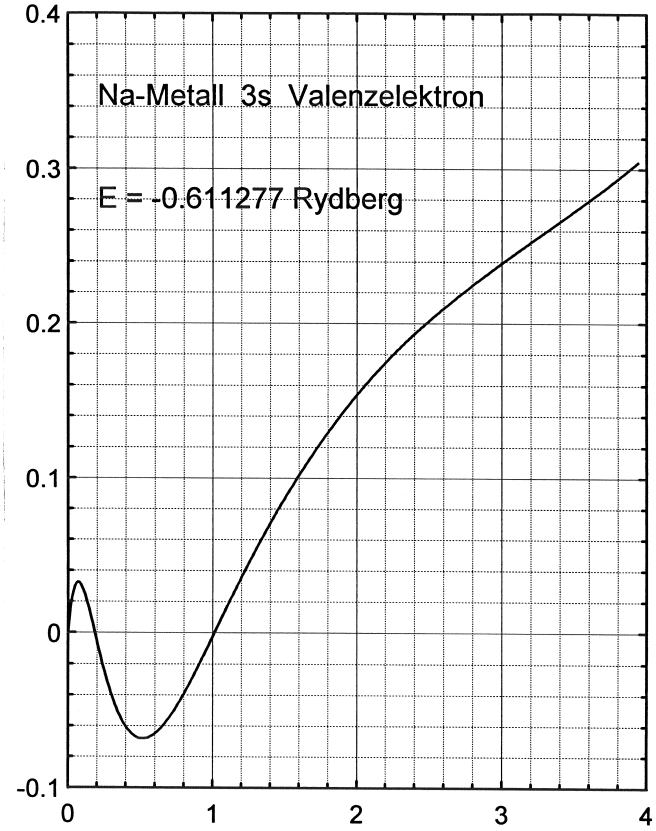


Abbildung 9.4: Ein Ergebnis der *shooting*-Numerov-Methode: die Eigenfunktion eines Valenzelektrons im Natrium-Kristall (400 Subintervalle im Intervall $[0, r_{WS}]$ mit r_{ws}=3.9405 Bohr).
Die Eigenenergie beträgt -0.611277 Rydberg.

9.5 Software-Angebot

In bezug auf Randwertprobleme gewöhnlicher Differentialgleichungen hat die NAG-Bibliothek einiges zu bieten, wie die folgende Tabelle zeigt:

Boundary-value Problems Shooting Method	
simple parameters	**D02HAF**
generalised parameters	**D02HBF, D02AGF**
additional facilities	**D02SAF**
Boundary-value Problems Finite-difference Method	
simple parameters	**D02GAF**
linear problem	**D02GBF**
full nonlinear problem	**D02RAF**
Chebyshev Collocation	
single equation	**D02JAF**
first-order system	**D02JBF**
general system	**D02TGF**
Sturm-Liouville Eigenvalue Problems	
regular problems	**D02KAF**
general problems	**D02KDF**
eigenfunction calculation	**D02KEF**

Sie finden in dieser Tabelle bekannte Begriffe wie *shooting method* und *finite-difference method*, wobei die Differenzenmethode nicht nur auf lineare, sondern auch auch nicht-lineare RWP angewendet werden kann. Die *Kollokationsmethode* nach Chebyshev ist für lineare Probleme ebenfalls sehr leistumgsfähig; es handelt sich dabei um eine Entwicklung der Lösungsfunktion nach Chebyshev-Polynomen und um eine numerische Berechnung der Polynom-Koeffizienten.

Obwohl in dieser LV. die Besprechung der numerischen Behandlung von *partiellen Differentialgleichungen* (leider) keinen Platz hat, möchte ich abschließend zumindest einige Hinweise auf das sehr umfangreiche Software-Angebot in bezug auf diese Probleme geben:
Einen sehr guten Überblick über p.d. und kommerizielle Software finden Sie in den schon häufig zitierten Büchern von Ch. Überhuber ([21],[22]). So ist z. B. in [21], S. 331ff eine Fallstudie 'Software für elliptische partielle Differentialgleichungen' enthalten, die einschlägige Programme aus den (p.d.) Bibliotheken ELLPACK und TOMS sowie aus NAG und IMSL vorstellt.

Literaturverzeichnis

[1] H. Ernst, *Numerische Methoden für Kleincomputer*, Luther-Verlag 1984.

[2] G. Engeln-Müllges und F. Reutter, *Formelsammlung zur Numerischen Mathematik mit Standard-FORTRAN-Programmen*, BI Mannheim 1984.

[3] G. Engeln-Müllges und F. Reutter, *Formelsammlung zur Numerischen Mathematik mit C-Programmen*, BI Mannheim 1990.

[4] G. Engeln-Müllges und F. Reutter, *Formelsammlung zur Numerischen Mathematik mit PASCAL-Programmen*, BI Mannheim 1985.

[5] G. Engeln-Müllges und F. Reutter, *Numerik-Algorithmen mit Programmen in FORTRAN, C und Turbo Pascal*, VDI-Verlag, 1996.

[6] G. Engeln-Müllges and F. Uhlig, *Numerical Algorithms with C or FORTRAN*, Springer-Verlag, 1996.

[7] W.S. Dorn and D.D. McCracken, *Numerical Methods with Fortran IV Case Studies*, Wiley 1972.

[8] A. Björck and G. Dahlquist, *Numerische Methoden*, Oldenburg Verlag 1972.

[9] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes*, 2nd ed., Cambridge Uni Press 1992.

[10] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C*, 2nd ed., Cambridge Uni Press 1992.

[11] M. Abramowitz and I.A. Segun, *Handbook of Mathematical Functions*, Dover Publications 1968.

[12] G.E. Forsythe und C.B. Moler, *Computer-Verfahren für lineare algebraische Systeme*, Oldenbourg Verlag 1971.

[13] G. Paulin und E. Griepentrog, *Numerische Verfahren der Programmiertechnik*, VEB Verlag Berlin 1975.

[14] P.K. MacKeown and D.J. Newman, *Computational Techniques in Physics*, Hilger 1987.

[15] I.S. Beresin und N.P. Shidkow, *Numerische Methoden 2*, VEB Verlag Berlin 1971.

[16] I.S. Beresin und N.P. Shidkow, *Numerische Methoden 1*, VEB Verlag Berlin 1970.

[17] Y.A. Shreider, *The Monte Carlo Method*, Pergamon Press Oxford 1967.

[18] R. Frühwirt, M. Regler, *Monte-Carlo-Methoden*, BI Mannheim 1983.

[19] B. P. Demidowitsch, L. A. Maron und E. S. Schuwalowa, *Numerische Methoden der Analysis*, VEB Verlag Berlin 1968.

[20] G. N. Poloshi, *Mathematisches Praktikum*, Teubner Verlag Leipzig 1963.

[21] Ch. Überhuber, *Computer-Numerik 1*, Springer-Verlag Berlin 1995.

[22] Ch. Überhuber, *Computer-Numerik 2*, Springer-Verlag Berlin 1995.

[23] P. L. DeVries, *Computerphysik*, Spektrum Akademischer Verlag Heidelberg, 1995.

Inhaltsverzeichnis