

7. Näherungsmethoden

Systeme wie der harmonische Oszillator oder das Wasserstoffatom, bei denen wir eine exakte Lösung der Schrödinger-Gleichung finden konnten, sind eher Ausnahmefälle. (Solche Systeme werden manchmal „integrierbar“ genannt; sie haben oft eine verborgene zusätzliche Symmetrie, die zur Lösbarkeit führt.)

In anderen Fällen kann eine annähernde analytische Lösung nützlich sein (neben einer „exakten“ numerischen Lösung).

7.1 Rayleigh-Ritz-Variationsprinzip [Fließbach 44]

Sei $\hat{H}$ ein Hamilton-Operator und

$$\hat{H} |n\rangle = E_n |n\rangle, \quad E_0 \leq E_1 \leq E_2 \dots,$$

wobei $\{|n\rangle\}$ und $\{E_n\}$ nicht bekannt sind.

Uns interessieren $|0\rangle$ und E_0 . Sei $|\varphi\rangle$ ein physikalisch sinnvoller Ansatz für $|0\rangle$. Dann ist

$$E_\varphi := \langle \varphi | \hat{H} | \varphi \rangle, \quad \langle \varphi | \varphi \rangle = 1$$

möglicherweise eine gute Näherung für E_0 .

Behauptung: $E_\varphi \geq E_0$.

Beweis:

$\{|n\rangle\}$ ist eine vollständige Menge

$$\Rightarrow |\varphi\rangle = \sum_{n=0}^{\infty} c_n |n\rangle, \quad \sum_{n=0}^{\infty} |c_n|^2 = 1$$

$$\Rightarrow E_\varphi = \sum_{n=0}^{\infty} |c_n|^2 E_n$$

$$= E_0 + \underbrace{\sum_{n=0}^{\infty} |c_n|^2 (E_n - E_0)}$$

alle Terme nicht-negativ $\Rightarrow \square$.

Gleichheit gilt ($E_\varphi = E_0$) genau dann, wenn $|\varphi\rangle = |0\rangle$.

Das „Variationsprinzip“ entsteht, indem wir den Ansatz als Funktion einer Menge von Parametern ausdrücken $(\alpha_1, \alpha_2, \dots, \alpha_N)$, und E_φ in diesem Parameterraum minimieren; das Minimum stellt eine Näherung (eine obere Grenze) für E_0 dar.

Zum Beispiel:

$$R_{00}(r) =: G(\alpha_1, \alpha_2, \dots) e^{-\alpha_1 r - \alpha_2 r^2 \dots},$$

$$\text{mit } G := \left\{ \int_0^\infty dr r^2 [e^{-\alpha_1 r - \alpha_2 r^2 \dots}]^2 \right\}^{-\frac{1}{2}};$$

folglich ist

$$E_\varphi(\alpha_1, \alpha_2, \dots) = \int_0^\infty dr r^2 R_{00}(r) \left[-\frac{\hbar^2}{2m} \left(\frac{d^2}{dr^2} + \frac{2}{r} \frac{d}{dr} \right) + V(r) \right] R_{00}(r).$$

Bemerkungen:

(1)

Die Energie des Grundzustandes kann in der Regel genauer genähert werden als die Wellenfunktion:

$$|\varphi\rangle = |0\rangle + \varepsilon |\Psi\rangle; \quad \langle 0|\Psi\rangle = 0$$

(d.h. Fehler sei der Ordnung ε)

$$\Rightarrow E_\varphi = E_0 + \varepsilon^2 \langle \Psi | \hat{H} | \Psi \rangle$$

(d.h. Fehler ist der Ordnung ε^2)

(2)

Im Prinzip ist es auch möglich, die Energien und Wellenfunktionen der angeregten Zustände zu bestimmen. Seien $|0\rangle, \dots, |N-1\rangle$ bekannt.

Dann verlangt man vom Ansatz $|\varphi\rangle$ neben Normierung ($\langle \varphi | \varphi \rangle = 1$) auch Orthogonalität:

$$\langle \varphi | n \rangle = 0, \quad n = 0, \dots, N-1.$$

So gilt

$$|\varphi\rangle = \sum_{n=N}^{\infty} c_n |n\rangle, \quad \sum_{n=N}^{\infty} |c_n|^2 = 1$$

und

$$E_\varphi = E_N + \sum_{n=N}^{\infty} |c_n|^2 (E_n - E_N) \geq E_N.$$

7.2 Numerische Lösung der Radialgleichung

91

Wie löst man Eigenwertprobleme mit dem Rechner? Nehmen wir als Beispiel das Wasserstoffatom, so dass die richtige Antwort schon bekannt ist, Seiten 85-86:

$$\left\{ \frac{d^2}{ds^2} - \frac{l(l+1)}{s^2} + \frac{s_0}{s} - 1 \right\} u(s) = 0 \quad ; \quad u(0) = u(\infty) = 0$$

$$[\Rightarrow s_0 = 2n \quad ; \quad n = l+1, l+2, \dots]$$

Nochmalige Umschreibung: Dividiere durch s_0^2 ; $x := s_0 s$; $v := \frac{du}{dx}$

$$\Rightarrow \begin{cases} \frac{dv}{dx} = v \\ \frac{dv}{dx} = \left[\frac{l(l+1)}{x^2} - \frac{1}{x} + \frac{1}{s_0^2} \right] u \end{cases}$$

Eine Idee*: Fange bei $x \approx 0$ an, und integriere vorwärts.
 Falls s_0 falsch gewählt wurde, divergiert die Lösung.
 Der richtige Wert kann durch Iteration bestimmt werden.

Ein Problem: Man kann nicht $x \rightarrow 0$ setzen, weil die rechte Seite dann divergiert. Benutze also $x = \varepsilon \ll 1$, und bestimme analytisch die Beziehung zur richtigen Randbedingung.

Asymptotisches Verhalten: Seite 85: $u(s) \sim s^{l+1}$ bei $s \ll 1$.

$$\text{Ansatz: } u = x^{l+1} + \beta x^{l+2} + \dots$$

$$\Rightarrow v = (l+1)x^l + \beta(l+2)x^{l+1} + \dots$$

$$\Rightarrow \frac{dv}{dx} = l(l+1)x^{l-1} + \beta(l+1)(l+2)x^l + \dots$$

$$\text{Vergleichen mit: } \begin{aligned} & l(l+1)x^{l-1} + \beta l(l+1)x^l + \dots & \leftarrow \frac{l(l+1)}{x^2} u \\ & \quad \quad \quad - x^l + \dots & \leftarrow -\frac{1}{x} u \end{aligned}$$

$$\Rightarrow \beta(l+1)(l+2) = \beta l(l+1) - 1$$

$$2\beta(l+1) = -1$$

$$\beta = -\frac{1}{2(l+1)}$$

$$\text{Insgesamt: } \begin{cases} u(\varepsilon) \approx \varepsilon^{l+1} - \frac{\varepsilon^{l+2}}{2(l+1)} \\ v(\varepsilon) \approx (l+1)\varepsilon^l - \frac{(l+2)\varepsilon^{l+1}}{2(l+1)} \end{cases}$$

* Im Prinzip könnte man auch „in der Mitte“, d.h. bei $x=x_0$ anfangen; $u(x_0)$ fixieren; $v(x_0)$ und s_0 variieren; und in beiden Richtungen ($x>0$ sowie $x \rightarrow \infty$) integrieren, um zu checken ob Randbedingungen erfüllt sind. Zweifache Variation ist aber schwierig!

Mit Mathematica:

2 | theoriell_Seq_solution.nb

■ Anfangsbedingungen:

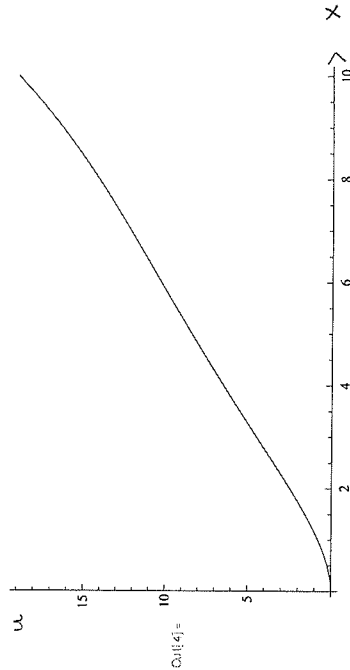
```
In[1]:= eps = 0.0001;
1 = 1;
```

■ Lösung:

```
In[3]:= solution[rho0_, xmax_] := NDSolve[{u'[x] == v[x],
v'[x] == (1/(1+1)/x^2 - 1/x + 1/rho0^2) u[x]}, u[eps] == eps^(1+1) - eps^(1+2)/2/(1+1),
v[eps] == (1+1) eps^1 - (1+2) eps^(1+1)/2/(1+1)}, {u, v}, {x, eps, xmax}]
```

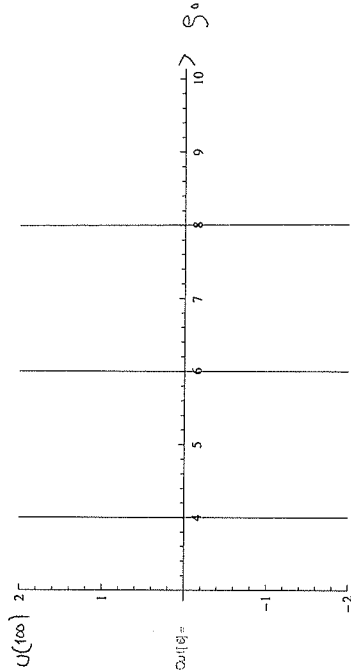
■ Beispiel

```
In[4]:= Plot[Evaluate[u[x] /. solution[3, 10.0]], {x, 0.1, 10.0}]
```



■ Integriere bis xmax=100, und checke Ergebnis:

```
In[6]:= Plot[u[100.] /. solution[rho0, 100.], {rho0, 3, 10}, PlotRange -> {-2, 2}]
```



■ Genauer um rho0 = 4 (15 Iterationen):

```
In[12]:= rhomin = 4 - 0.1;
rhomax = 4 + 0.2;
```

```
In[14]:= Do[If[N[u[100.] /. solution[(rhomin + rhomax)/2, 100.][[1]]] > 0,
{rhomin = (rhomin + rhomax)/2, Print[rhomin]},
{rhomax = (rhomin + rhomax)/2, Print[rhomax]}], {15}]
```

4.05

3.975

4.0125

3.99375

4.00312

3.99844

4.00078

3.99961

4.0002

3.9999

4.00005

3.99998

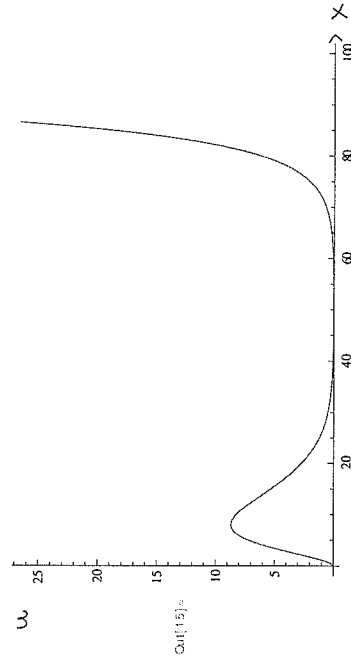
4.00001

3.99999

4.

■ Gegenprobe:

```
In[15]:= Plot[Evaluate[u[x] /. solution[(rhomin + rhomax)/2, 100.0]], {x, 0.1, 100.0}]
```



Zum Spielen:
mit 30 Iterationen
gibt es keine sichtbare
Divergenz mehr, und
so ist richtig
zu 8 Ziffern
(~ 8^8 ?).