Bestimmung quantenmechanischer Energieniveaus durch numerisches Lösen der Schrödinger-Gleichung

Marc Wagner

Goethe-Universität Frankfurt am Main – summer semester 2024

Version: July 2, 2024

- Im Folgenden ausgewählte numerische Grundlagen, die für den erfolgreichen Abschluss des Programmierprojekts im Rahmen der Vorlesung "Einführung in die Programmierung für Studierende der Physik" erforderlich sind.
- Im Wesentlichen ein Auszug aus den Vorlesungsaufzeichnungen zur Vorlesung "Numerical methods in physics" (`https://itp.uni-frankfurt.de/~mwagner/mcwagner.html#nuphe`).
- Das Programmierprojekt gliedert sich in drei aufeinander aufbauende Aufgabenteile, die auf den letzten drei Übungszetteln in diesem Semester gestellt werden (die Übungszettel vom 21.06.2024, 28.06.2024 und 05.07.2024).
- Das Programmierprojekt zählt nicht zu den regulären wöchentlichen Übungsaufgaben. Es wird nicht bepunktet, sondern nur mit "bestanden" oder "nicht bestanden" gewertet.
- Für einen erfolgreichen Abschluss (d.h. "bestanden") sind das vollständig funktionsfähige Programm sowie die Ergebnisse der geforderten numerischen Untersuchungen dem jeweiligen Tutor zu präsentieren. Die Termine für solche Präsentationen sind mit dem Tutor individuell zu vereinbaren. Empfohlen wird dafür die vorletzte Semesterwoche (08.07.2024 bis 12.07.2024). Strikte Deadline ist der letzte Tag der Vorlesungszeit, der 19.07.2024.

******** 1. Aufgabenteil ********

******** (Übungszettel vom 21.06.2024) ********

1 Ordinary differential equations (ODEs), initial value problems

1.1 Physics motivation

- Newton's equations of motion (EOMs), N point masses m_j,

  $$m_j \ddot{\mathbf{r}}_j(t) \quad = \quad \mathbf{F}_j(\mathbf{r}_1(t), \ldots, \mathbf{r}_N(t), \dot{\mathbf{r}}_1(t), \ldots, \dot{\mathbf{r}}_N(t), t) \quad , \quad j = 1, \ldots, N, \tag{1}$$

  initial conditions

  $$\mathbf{r}_j(t=0) \quad = \quad \mathbf{r}_{j,0} \quad , \quad \dot{\mathbf{r}}_j(t=0) \quad = \quad \mathbf{v}_{j,0}. \tag{2}$$

- Calculate trajectories $\mathbf{r}_j(t)$.
- Cannot be done analytically in the majority of cases, e.g. three-body problem "sun and two planets".
- For boundary value problems see section 3 (e.g. quantum mechanics [QM], Schrödinger equation, $\psi(x_1) = 0$, $\psi(x_2) = 0$).

1.2 Euler's method

- Preparatory step: rewrite ODEs as system of first order ODEs.
  - Newton's EOMs equivalent to

    $$\dot{\mathbf{r}}_j(t) \quad = \quad \mathbf{v}_j(t) \quad , \quad \dot{\mathbf{v}}_j(t) \quad = \quad \frac{\mathbf{F}_j(\mathbf{r}_1(t), \ldots, \mathbf{r}_N(t), \dot{\mathbf{r}}_1(t), \ldots, \dot{\mathbf{r}}_N(t), t)}{m_j}. \tag{3}$$

  - Define

    $$\mathbf{y}(t) \quad = \quad (\mathbf{r}_1(t), \ldots, \mathbf{r}_N(t), \mathbf{v}_1(t), \ldots, \mathbf{v}_N(t)) \tag{4}$$

    $$\mathbf{f}(\mathbf{y}(t), t) \quad = \quad \bigg(\underbrace{\mathbf{v}_1(t), \ldots, \mathbf{v}_N(t)}_{\in \mathbf{y}(t)}, \frac{\mathbf{F}_1(\mathbf{y}(t), t)}{m_1}, \ldots, \frac{\mathbf{F}_N(\mathbf{y}(t), t)}{m_N}\bigg). \tag{5}$$

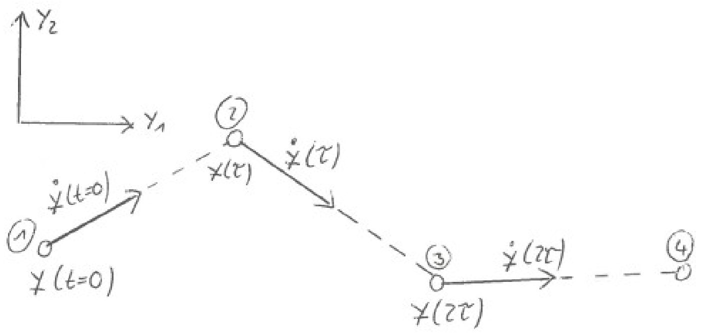
  - Then

    $$\dot{\mathbf{y}}(t) \quad = \quad \mathbf{f}(\mathbf{y}(t), t) \tag{6}$$

    (left hand side (lhs) can be evaluated in a straightforward way for given t and $\mathbf{y}(t)$).
  - Always possible to rewrite a system of ODEs according to (6).
- Solve (6) by iteration, i.e. perform many small steps in time, step size τ:

  $$\mathbf{y}(t+\tau) \quad = \quad \mathbf{y}(t) + \dot{\mathbf{y}}(t)\tau + \mathcal{O}(\tau^2) \quad = \quad \mathbf{y}(t) + \mathbf{f}(\mathbf{y}(t), t)\tau + \mathcal{O}(\tau^2). \tag{7}$$

Figure 3.A



- τ can be positive (→ computation of future) or negative (→ computation of past).
- Problem: method inefficient, because of large discretization errors.
  - $\mathcal{O}(\tau^2)$ error per step.
  - Time evolution from $t = 0$ (initial conditions) to $t = T$
    → T/τ steps
    → $\mathcal{O}((T/\tau)\tau^2) = \mathcal{O}(\tau)$ total error (very inefficient).
  - Total error might be underestimated (e.g. chaotic systems are highly sensitive to initial conditions and, thus, to the error per step).

1.3 Runge-Kutta (RK) method

- Same idea as in section 1.2, but improved discretization (stronger suppression of errors with respect to τ).
- "2nd-order RK":

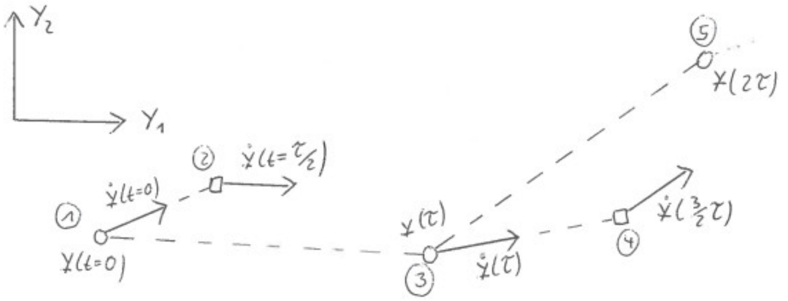
$$\mathbf{k}_1 \;=\; \mathbf{f}(\mathbf{y}(t), t)\tau \quad \rightarrow \text{ "full Euler step"} \tag{8}$$

$$\mathbf{k}_2 \;=\; \mathbf{f}\Big(\underbrace{\mathbf{y}(t) + (1/2)\mathbf{k}_1, t + (1/2)\tau}_{\rightarrow \text{ "half Euler step"}}\Big)\tau \tag{9}$$

$$\mathbf{y}(t+\tau) \;=\; \mathbf{y}(t) + \mathbf{k}_2 + \mathcal{O}(\tau^3). \tag{10}$$

  - $\mathbf{f}(\mathbf{y}(t) + (1/2)\mathbf{k}_1, t + (1/2)\tau)$ in (9): estimated derivative $\dot{\mathbf{y}}(t + \tau/2)$, i.e. after half step.
  - (10): 2nd order RK step (a full "Euler-like" step using the derivative after a half step).

Figure 3.B



- Proof of (10), i.e. that error per step is $\mathcal{O}(\tau^3)$:

$$\begin{aligned}
\mathbf{k}_2 &= \mathbf{f}\Big(\mathbf{y}+(1/2)\mathbf{f}\tau, t+(1/2)\tau\Big)\tau \quad = \\
&= \mathbf{f}\tau + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\frac{1}{2}\mathbf{f}\tau + \frac{\partial \mathbf{f}}{\partial t}\frac{1}{2}\tau\right)\tau + \mathcal{O}(\tau^3) \quad = \quad \mathbf{f}\tau + \frac{1}{2}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\dot{\mathbf{y}} + \frac{\partial \mathbf{f}}{\partial t}\right)\tau^2 + \mathcal{O}(\tau^3) \quad = \\
&= \mathbf{f}\tau + \frac{1}{2}\dot{\mathbf{f}}\tau^2 + \mathcal{O}(\tau^3) && (11) \\
\mathbf{y}(t+\tau) &= \mathbf{y} + \dot{\mathbf{y}}\tau + \frac{1}{2}\ddot{\mathbf{y}}\tau^2 + \mathcal{O}(\tau^3) \quad = \quad \mathbf{y} + \mathbf{f}\tau + \frac{1}{2}\dot{\mathbf{f}}\tau^2 + \mathcal{O}(\tau^3) \quad = \\
&= \mathbf{y} + \mathbf{k}_2 + \mathcal{O}(\tau^3) && (12)
\end{aligned}$$

(no arguments imply time t, e.g. $\mathbf{y} \equiv \mathbf{y}(t)$, $\mathbf{f} \equiv \mathbf{f}(\mathbf{y}(t), t)$).

- Discretization not unique, e.g. for $\mathcal{O}(\tau^3)$ error per step there are many possible RK expressions (an example is discussed in the tutorials).

- Straightforward to derive discretizations with $\mathcal{O}(\tau^4)$, $\mathcal{O}(\tau^5)$, ... error per step:

  - "3rd-order RK":

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}(\mathbf{y}(t), t)\tau && (13) \\
\mathbf{k}_2 &= \mathbf{f}\Big(\mathbf{y}(t) + \mathbf{k}_1, t+\tau\Big)\tau && (14) \\
\mathbf{k}_3 &= \mathbf{f}\Big(\mathbf{y}(t) + (1/4)\Big(\mathbf{k}_1 + \mathbf{k}_2\Big), t+(1/2)\tau\Big)\tau && (15) \\
\mathbf{y}(t+\tau) &= \mathbf{y}(t) + \frac{1}{6}\Big(\mathbf{k}_1 + \mathbf{k}_2 + 4\mathbf{k}_3\Big) + \mathcal{O}(\tau^4). && (16)
\end{aligned}$$

  - "4th-order RK":

$$\begin{aligned}
\mathbf{k}_1 &= \mathbf{f}(\mathbf{y}(t), t)\tau && (17) \\
\mathbf{k}_2 &= \mathbf{f}\Big(\mathbf{y}(t) + (1/2)\mathbf{k}_1, t+(1/2)\tau\Big)\tau && (18) \\
\mathbf{k}_3 &= \mathbf{f}\Big(\mathbf{y}(t) + (1/2)\mathbf{k}_2, t+(1/2)\tau\Big)\tau && (19) \\
\mathbf{k}_4 &= \mathbf{f}\Big(\mathbf{y}(t) + \mathbf{k}_3, t+\tau\Big)\tau && (20) \\
\mathbf{y}(t+\tau) &= \mathbf{y}(t) + \frac{1}{6}\Big(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\Big) + \mathcal{O}(\tau^5). && (21)
\end{aligned}$$

  - ...

- Common choice is 4th-order RK.
- Even better: numerical tests with different order RKs (higher orders allow larger step sizes τ [which is good], require larger numbers of arithmetic operations per step [which is bad]).
- Example: Compute the trajectory of the 1-dimensional harmonic oszillator (HO).
  - Lagrangian:

$$L = \frac{m}{2}\dot{x}^2 - \frac{m\omega^2}{2}x^2. \tag{22}$$

  - EOMs:

$$m\ddot{x}(t) = -m\omega^2 x(t), \tag{23}$$

i.e.

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t), t) \tag{24}$$

with

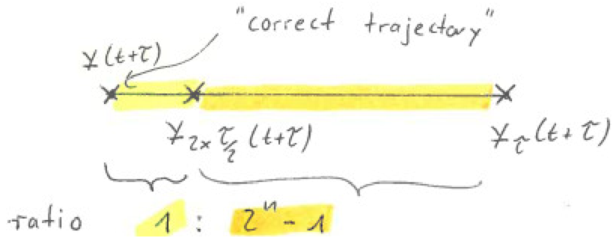
$$\mathbf{y}(t) = (x(t), v(t)) \quad , \quad \mathbf{f}(\mathbf{y}(t), t) = (v(t), -\omega^2 x(t)). \tag{25}$$

  - Initial conditions: $x(t=0) = x_0$, $\dot{x}(t=0) = 0$, i.e. $\mathbf{y}(t=0) = (x_0, 0)$.
  - $\omega = 1.0$, $x_0 = 1.0$, step size $\tau = 0.1$.
  - Resulting trajectories for Euler, 2nd-order RK, 3rd-order RK and 4th-order RK are shown in Figure 1.
  - Errors of the trajectories for Euler, 2nd-order RK, 3rd-order RK and 4th-order RK are shown in Figure 2.

1.3.1 Estimation of errors

- Error per step for n-th order RK can be estimated in the following way:
  - RK step with step size τ
    $\rightarrow \mathbf{y}_\tau(t+\tau)$
    $\rightarrow \vec{\delta}_\tau \approx \mathbf{c}\tau^{n+1}$.
  - 2 RK steps with step size $\tau/2$
    $\rightarrow \mathbf{y}_{2\times\tau/2}(t+\tau)$
    $\rightarrow \vec{\delta}_{2\times\tau/2} \approx 2\mathbf{c}(\tau/2)^{n+1}$.

Figure 3.C



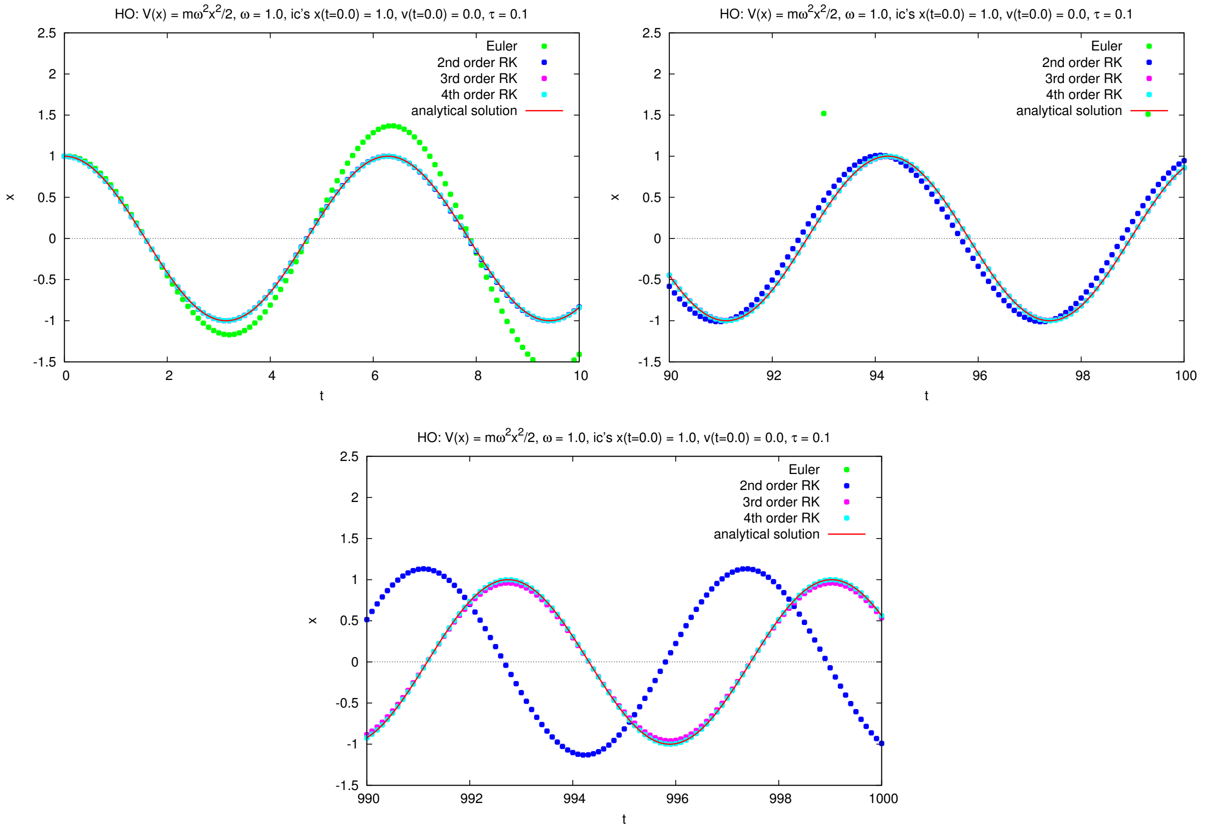


Figure 1: HO, resulting trajectories for Euler, 2nd-order RK, 3rd-order RK and 4th-order RK.

- Estimated absolute error for $\mathbf{y}_{2\times\tau/2}(t+\tau)$:

  $$\delta_{\mathrm{abs}} = \frac{|\mathbf{y}_{2\times\tau/2}(t+\tau) - \mathbf{y}_{\tau}(t+\tau)|}{2^n - 1}, \qquad (26)$$

  where $|\ldots|$ can be e.g. Euclidean norm, maximum norm (might be a better choice for many degrees of freedom [dof's]), ...

- Estimated relative error for $\mathbf{y}_{2\times\tau/2}(t+\tau)$ (might be more relevant than estimated absolute error):

  $$\delta_{\mathrm{rel}} = \frac{\delta_{\mathrm{abs}}}{|\mathbf{y}(t)|}. \qquad (27)$$

- Estimated error allows local extrapolation:
  - Correct by estimated error:

    $$\mathbf{y}_{2\times\tau/2}(t+\tau) \rightarrow \mathbf{y}_{2\times\tau/2}(t+\tau) + \frac{\mathbf{y}_{2\times\tau/2}(t+\tau) - \mathbf{y}_{\tau}(t+\tau)}{2^n - 1}. \qquad (28)$$

  - However, no estimation of errors, when using (28).

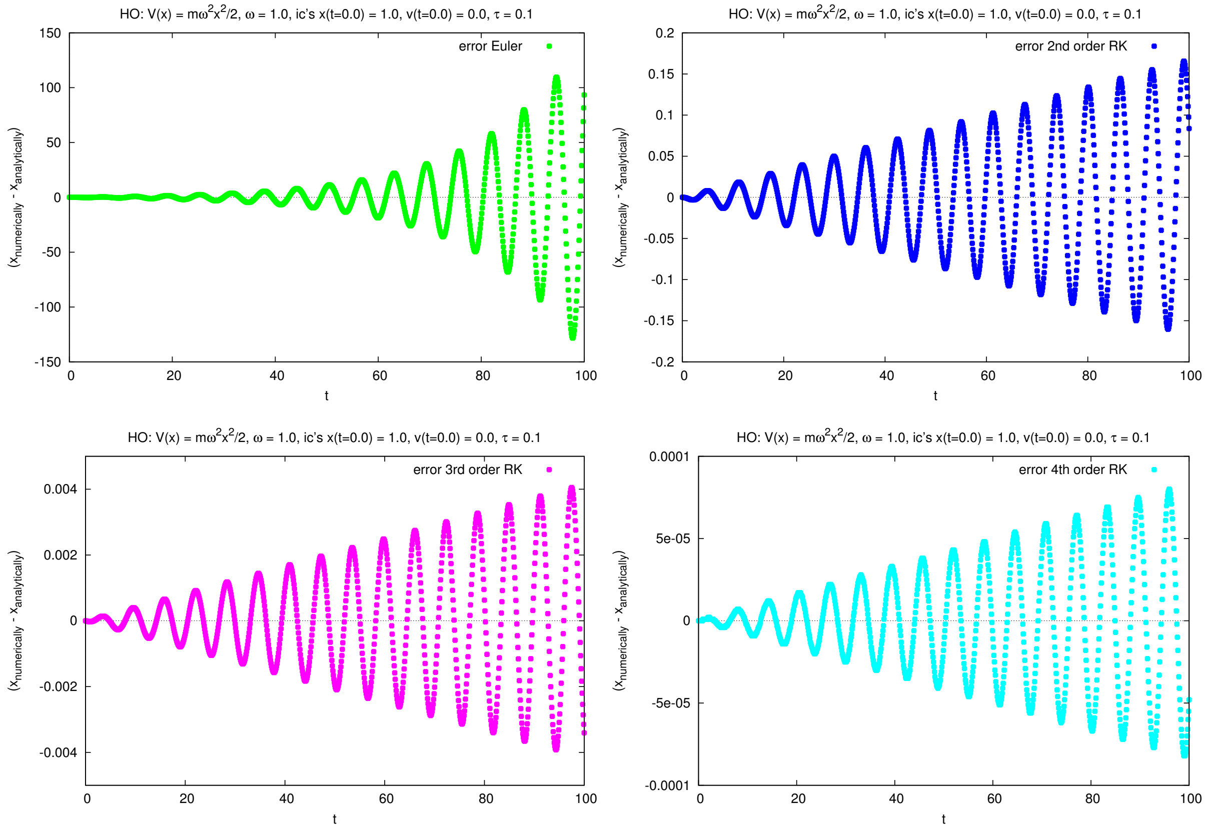


Figure 2: HO, errors of the trajectories for Euler, 2nd-order RK, 3rd-order RK and 4th-order RK.

******** 2. Aufgabenteil ********

******** (Übungszettel vom 28.06.2024) ********

2 Newton-Raphson root finding for univariate functions $f(x)$

- Starting point: arbitrary x_1 (ideally x_1 is close to a root of $f(x)$, i.e. $f(x_1) \approx 0$).
- Newton-Raphson method might find a root of $f(x)$, but convergence is not guaranteed.
- Basic principle:
  - An iterative method, i.e. x_{n+1} is a better estimate of the root than x_n (in that respect similar to bisection; see slides "Die Programmiersprache C – Kontrollstrukturen").
  - Uses derivative $f'(x_n)$:
    - f' can be estimated numerically via a finite difference:

$$f(x+h) \quad = \quad f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \mathcal{O}(h^3) \tag{29}$$

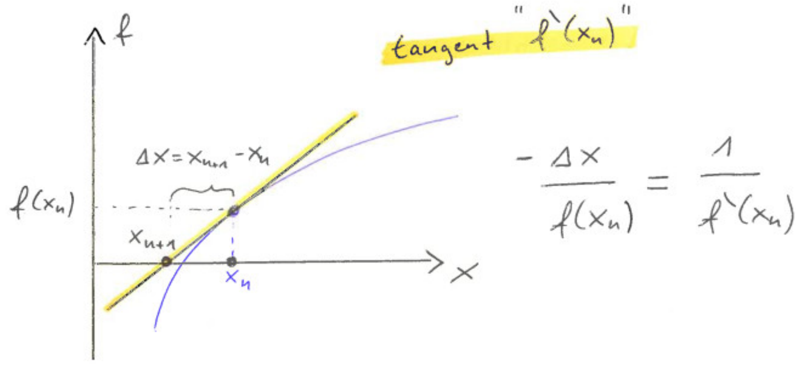
$$\rightarrow \quad f'(x) \quad = \quad \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2) \tag{30}$$

$$\rightarrow \quad f'_{\text{finite difference}}(x) = \frac{f(x+h) - f(x-h)}{2h} \quad \approx \quad f'(x) \tag{31}$$

("symmetric derivative").

    - Optimal choice $h = h_{\text{opt}} \sim \epsilon^{1/3}$, where ϵ denotes the relative precision ($\epsilon \approx 10^{-7}$ for `float` and $\epsilon \approx 10^{-16}$ for `double`). The relative error is then $\delta f'(x)/|f'(x)| \sim \epsilon^{2/3}$, where $\delta f'(x) = |f'(x) - f'_{\text{finite difference}}(x)|$.
  - Each step as sketched below.

Figure 5.C



- Algorithm:
  - $n = 1$.

(1)

$$\Delta x \quad = \quad -\frac{1}{f'(x_n)} f(x_n) \quad , \quad x_{n+1} \quad = \quad x_n + \Delta x. \tag{32}$$

- *If $|\Delta x|$ sufficiently small:*
  - → *x_{n+1} is approximate root.*
    End of algorithm.

  Else:
  - → *$n = n + 1$.*
    Go to (1).

- Advantages and disadvantages:
  - (+) Converges faster than bisection.
  - (−) Does not always find a root (bisection always finds a root).
  - (−) f' has to be known analytically/must be cheap to evaluate numerically (bisection does not need f').
  - (+) Can be generalized to multivariate functions $\mathbf{f}(\mathbf{x})$.

3 Ordinary differential equations (ODEs), boundary value problems

3.1 Physics motivation

- Newton's EOMs, N point masses m_j,

$$m_j \ddot{\mathbf{r}}_j(t) \quad = \quad \mathbf{F}_j(\mathbf{r}_1(t), \ldots, \mathbf{r}_N(t), \dot{\mathbf{r}}_1(t), \ldots, \dot{\mathbf{r}}_N(t), t) \quad , \quad j = 1, \ldots, N, \tag{33}$$

  boundary conditions

$$\mathbf{r}_j(t_1) \quad = \quad \mathbf{r}_{j,1} \quad , \quad \mathbf{r}_j(t_2) \quad = \quad \mathbf{r}_{j,2} \tag{34}$$

  ("Compute trajectory of a particle, which is at $\mathbf{r}_1$ at time t_1 and at $\mathbf{r}_2$ at time t_2.").

- QM, Schrödinger equation in 1 dimension,

$$-\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) \quad = \quad E\psi(x), \tag{35}$$

  boundary conditions

$$\psi(x_1) \quad = \quad \psi(x_2) \quad = \quad 0 \tag{36}$$

  (i.e. "$V(x) = \infty$ at $x = x_1, x_2$", e.g. infinite potential well).

  - Example appropriate? E is unknown, i.e. (35) and (36) is rather an eigenvalue problem, not an ordinary boundary value problem ...?
  - Yes, can be reformulated:
    - ∗ Consider E as a function of x, i.e. $E = E(x)$.
    - ∗ Add another ODE, $E'(x) = 0$, which guarantees $E(x) = \text{const}$.

    → System of ODEs,

$$-\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) \quad = \quad E(x)\psi(x) \quad , \quad E'(x) \quad = \quad 0. \tag{37}$$

3.2 Shooting method

- Preparatory step as in section 1.2: rewrite ODEs as system of first order ODEs,

$$\mathbf{y}'(x) = \mathbf{f}(\mathbf{y}(x), x) \tag{38}$$

  (both $\mathbf{y}$ and $\mathbf{f}$ have N components) and boundary conditions as

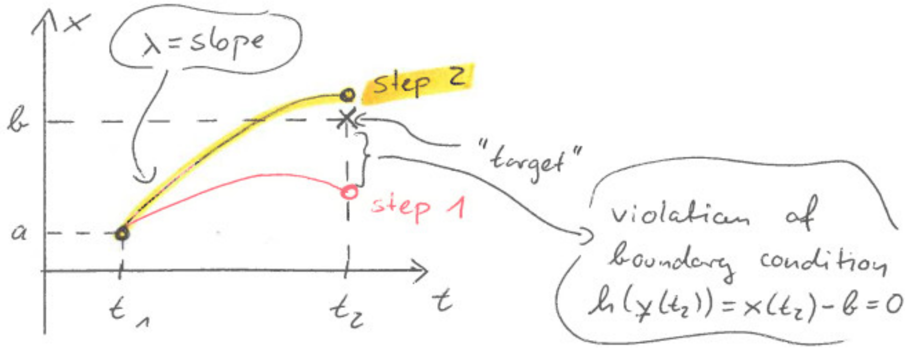
$$g_j(\mathbf{y}(x_1)) = 0 \quad , \quad j = 1, \ldots, n < N \tag{39}$$

$$h_j(\mathbf{y}(x_2)) = 0 \quad , \quad j = 1, \ldots, N - n. \tag{40}$$

- Basic principle:
  - Choose/guess initial conditions $\mathbf{y}(x_1)$ such that
    * boundary conditions $g_j(\mathbf{y}(x_1)) = 0$, $j = 1, \ldots, n < N$ are fulfilled,
    * boundary conditions $h_j(\mathbf{y}(x_2)) = 0$, $j = 1, \ldots, N - n$ are approximately fulfilled ($\mathbf{y}(x_2)$ can be computed using a RK method from section 1.3).
  - Use root finding methods (e.g. bisection [see slides "Die Programmiersprache C – Kontrollstrukturen"] or Newton-Raphson method [see section 2]) to iteratively improve initial conditions $\mathbf{y}(x_1)$, i.e. such that $h_j(\mathbf{y}(x_2)) = 0$.

- Example: mechanics, $m\ddot{x}(t) = F(x(t))$ with $x(t_1) = a$, $x(t_2) = b$.
  - $\mathbf{y}(t) = (x(t), v(t))$, $\mathbf{f}(\mathbf{y}(t), t) = (v(t), F(x(t))/m)$ (as in section 1.2).
  - $g(\mathbf{y}(t_1)) = x(t_1) - a = 0$, $h(\mathbf{y}(t_2)) = x(t_2) - b = 0$.
  - Choose initial conditions $\mathbf{y}(t_1) = (a, \lambda)$.
    * a in 1st component → $g(\mathbf{y}(t_1)) = 0$ fulfilled.
    * λ in 2nd component should lead to $h(\mathbf{y}(t_2)) \approx 0$.
  - RK computation of $\mathbf{y}(t)$ from $t = t_1$ to $t = t_2$.

Figure 6.A



  - Improve initial conditions, i.e. tune λ, using the Newton-Raphson method (see section 2):

  - ∗ Interpret $h(\mathbf{y}(t_2)) = x(t_2) - b$ as function of λ ($x(t_2)$ depends on initial conditions $\mathbf{y}(t_1)$, i.e. on λ).
    - ∗ Compute derivative $dh(\mathbf{y}(t_2))/d\lambda$ (needed by the Newton-Raphson method) numerically (see (31)).
    - ∗ Newton-Raphson step to improve λ:

$$\lambda \quad \rightarrow \quad \lambda - \frac{h(\mathbf{y}(t_2))}{dh(\mathbf{y}(t_2))/d\lambda}. \tag{41}$$

  - Repeat RK computation and Newton-Raphson step, until $h(\mathbf{y}(t_2)) = 0$ (numerically 0, e.g. up to 6 digits).

3.3 Example: QM, 1 dimension, infinite potential well

- Infinite potential well:

$$V(x) \quad = \quad \left\{ \begin{array}{ll} 0 & \text{if } 0 \leq x \leq L \\ \infty & \text{otherwise} \end{array} \right. . \tag{42}$$

- Schrödinger equation and boundary conditions:

$$-\frac{\hbar^2}{2m}\psi''(x) \quad = \quad E\psi(x) \quad , \quad \psi(x=0) \quad = \quad \psi(x=L) \quad = \quad 0. \tag{43}$$

- Reformulate equations using exclusively dimensionless quantities[1]:

$$\hat{x} \quad = \quad \frac{x}{L} \tag{44}$$

$$\rightarrow \quad \frac{d}{d\hat{x}} \quad = \quad L\frac{d}{dx} \tag{45}$$

$$\rightarrow \quad -\frac{d^2}{d\hat{x}^2}\psi(\hat{x}) \quad = \quad \underbrace{\frac{2mEL^2}{\hbar^2}}_{=\hat{E}}\psi(\hat{x}) \tag{46}$$

  ($\hat{E}$ is “dimensionless energy”), i.e.

$$-\frac{d^2}{d\hat{x}^2}\psi(\hat{x}) \quad = \quad \hat{E}\psi(\hat{x}) \quad , \quad \psi(\hat{x}=0) \quad = \quad \psi(\hat{x}=1) \quad = \quad 0. \tag{47}$$

- Analytical solution (to check numerical results):

$$\psi(\hat{x}) \quad = \quad \sqrt{2}\sin(n\pi\hat{x}) \quad , \quad \hat{E} \quad = \quad \pi^2 n^2 \quad , \quad n = 1, 2, \ldots \tag{48}$$

- Numerical solution:

[1] Using exclusively dimensionless quantities is possible for any physics problem and strongly recommended for numerical computations. For details see lecture “Numerical methods in physics” (https://itp.uni-frankfurt.de/∼mwagner/mcwagner.html#nuphe).

- Rewrite Schrödinger equation as system of first order ODEs:

  $$\psi'(\hat{x}) \;=\; \phi(\hat{x}) \quad , \quad \phi'(\hat{x}) \;=\; -\hat{E}(\hat{x})\psi(\hat{x}) \quad , \quad \hat{E}'(\hat{x}) \;=\; 0, \tag{49}$$

  ($'$ denotes $d/d\hat{x}$) i.e.

  $$\mathbf{y}(x) \;=\; \Big(\psi(\hat{x}), \phi(\hat{x}), \hat{E}(\hat{x})\Big) \quad , \quad \mathbf{f}(\mathbf{y}(x), x) \;=\; \Big(\phi(\hat{x}), -\hat{E}(\hat{x})\psi(\hat{x}), 0\Big). \tag{50}$$

- Initial conditions for RK/shooting method:
  - $\psi(\hat{x} = 0.0) = 0.0$
    (boundary condition at $\hat{x} = 0$),
  - $\phi(\hat{x} = 0.0) = 1.0$
    (must be $\neq 0$, apart from that arbitrary; different choices result in differently normalized wavefunctions),
  - $\hat{E}(\hat{x} = 0.0) = \mathcal{E}$
    (will be tuned by Newton-Raphson method such that boundary condition
    $\psi(\hat{x} = 1) = 0$ is fulfilled).
- Crude "graphical determination" of energy eigenvalues (necessary to choose appropriate initial condition for the shooting method):
  - Figure 3 shows $\psi(\hat{x} = 1.0)$ as a function of $\mathcal{E}$ computed with 4th order RK; roots indicate energy eigenvalues.

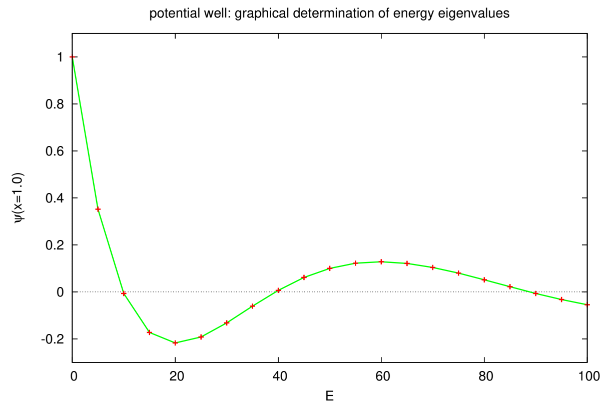


Figure 3: Infinite potential well, crude graphical determination of energy eigenvalues.

  - There are 3 eigenvalues in the range $0.0 < \hat{E} < 100.0$:
    $\hat{E}_0 \approx 10.0$, $\hat{E}_1 \approx 40.0$, $\hat{E}_2 \approx 90.0$.
- Shooting method with $\mathcal{E} \in \{10.0, 40.0, 90.0\}$.
  - Figure 4 (top) illustrates the first Newton-Raphson step for the second excitation (4th order RK).
  - Figure 4 (bottom) shows the resulting non-normalized wave functions of the three lowest states (4th order RK).
  - Convergence after three Newton-Raphson steps (4th order RK, 7 digits of accuracy); see program output below.

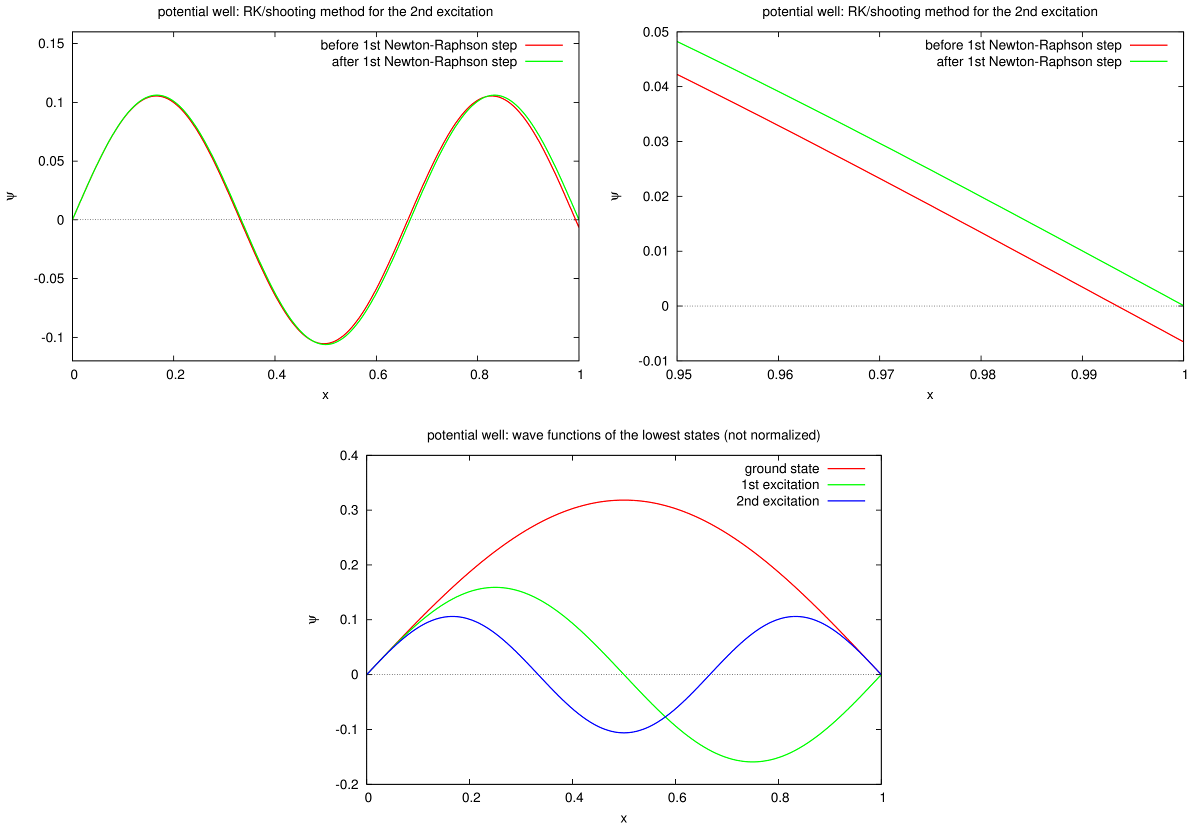


Figure 4: Infinite potential well. **(top)** First Newton-Raphson step for the second excitation. **(bottom)** Non-normalized wave functions of the three lowest states.

```
ground state:
E_num = +10.000000 .
E_num =  +9.868296  ,  E_ana =  +9.869604  ,  \psi(x=1) = -0.006541 .
E_num =  +9.869604  ,  E_ana =  +9.869604  ,  \psi(x=1) = +0.000066 .
E_num =  +9.869604  ,  E_ana =  +9.869604  ,  \psi(x=1) = +0.000000 .

1st excitation:
E_num = +40.000000 .
E_num = +39.472958  ,  E_ana = +39.478418  ,  \psi(x=1) = +0.006539 .
E_num = +39.478417  ,  E_ana = +39.478418  ,  \psi(x=1) = -0.000069 .
E_num = +39.478418  ,  E_ana = +39.478418  ,  \psi(x=1) = -0.000000 .

2nd excitation:
E_num = +90.000000 .
E_num = +88.813303  ,  E_ana = +88.826440  ,  \psi(x=1) = -0.006537 .
E_num = +88.826438  ,  E_ana = +88.826440  ,  \psi(x=1) = +0.000074 .
E_num = +88.826440  ,  E_ana = +88.826440  ,  \psi(x=1) = +0.000000 .
```

******** 3. Aufgabenteil ********

******** (Übungszettel vom 05.07.2024) ********

4 Numerical investigation of the quantum mechanical HO with hard walls

- The third and final part of the programming project does not require any further numerical techniques. The goal is rather to apply the methods discussed in section 1 to section 3, which you have implemented in libraries and tested in part 1 and part 2 of the programming project, to an interesting problem from quantum mechanics, which cannot be solved analytically (or at least seems to be very hard to solve analytically).

- You will consider the potential

$$V(x) \quad = \quad \left\{ \begin{array}{cl} m\omega^2 x^2/2 & \text{if } -L/2 \leq x \leq +L/2 \\ \infty & \text{otherwise} \end{array} \right. , \tag{51}$$

  i.e. a harmonic potential with hard walls at $x = \pm L/2$.

- It is convenient to define the dimensionless quantity

$$\hat{\omega} \quad = \quad \frac{m\omega L^2}{\hbar}. \tag{52}$$

  The Schrödinger equation expressed exclusively in dimensionless quantities together with the hard wall boundary conditions is then

$$-\frac{d^2}{d\hat{x}^2}\psi(\hat{x}) \quad = \quad \Big(\hat{E} - \hat{\omega}^2\hat{x}^2\Big)\psi(\hat{x}) \quad , \quad \psi(\hat{x} = -1/2) \quad = \quad \psi(\hat{x} = +1/2) \quad = \quad 0. \tag{53}$$

- Whenever you carry out numerical investigations, it is essential that you meticulously test your code and results. Without such tests, hardly anybody following your work in detail will trust your numerical results. Moreover, there is a high probability that your code has a bug and your results are wrong. If you publish incorrect numerical results, your reputation as a physicist will suffer and your career may end prematurely.

- Even if your problem cannot be solved analytically, there are often regions in parameter space or specific limits, where it is obvious that your results must approach corresponding results from simpler analytically solvable problems. You should always try to identify such possibilities for testing and carefully and extensively perform the tests. Only then you may trust your numerical results and can confidently proceed by interpreting them in "interesting regions in parameter space", where it is less clear, what to expect, and where are no obvious relations to any existing analytical or numerical results.

  - For $\hat{E} \gg \hat{\omega}^2/4 \geq \hat{\omega}^2\hat{x}^2$, i.e. for large energies, the energy levels should be close those of the infinite well $\hat{E}_n^{\mathrm{IW}} = \pi^2(n+1)^2$, $n = 0, 1, \ldots$ (in this project part we always label the ground state with $n = 0$, the first excitation with $n = 1$, etc.). The reason behind this expectation can be seen by plotting the potential (51) on an appropriate scale ("the bottom of the potential will then look almost flat").

- For $\hat{E} \ll \hat{\omega}^2/4$ the particle should be deep inside the HO contribution to the potential and, thus, mostly localized in the region $-1/2 < \hat{x} < +1/2$. Consequently, the hard walls are essentially irrelevant and the energy levels should be close those of the ordinary HO $\hat{E}_n^{\mathrm{HO}} = \hat{\omega}(2n+1)$, $n = 0, 1, \ldots$

- Your task in part 3 of the programming project is to solve the Schrödinger equation (51) for various values of $\hat{\omega}$ numerically using the shooting method. The goal is to make contact to analytically known results, by studying the two limits discussed above, and then to investigate the effect of $\hat{\omega}$ on the spectrum of energy eigenvalues away from these limits. Keep in mind that it is of utmost importance that you do not just carry out the numerical tasks from the perspective of a programmer, but also to discuss and interpret the results from a physicist's point of view (numerical methods are powerful tools in physics, but just generating and showing plots is not sufficient to understand any physics problem).

References

[1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, "Numerical recipes 3rd edition: the art of scientific computing," Cambridge University Press (2007).