FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

Physikalisch-Astronomische Fakultät
Theoretisch-physikalisches Institut

Solving the Initial Boundary Value Problem for the Wave Equation

---

Report on Research Labworks
submitted on November 20, 2020

Supervisor: Vsevolod Nedora
Students: Johannes Nicklaus (170099)
Anne Weber (162587)

Contents

1. Theoretical Background

1.1. General Analytical Solution

The wave equation (WE) for the scalar field $\phi(t, x)$ in one spatial and one time dimension is

$$0 = \frac{\partial^2 \phi}{\partial t^2} - c^2 \frac{\partial^2 \phi}{\partial x^2} = \partial_{tt}\phi - c^2 \partial_{xx}\phi \tag{1}$$

with $c \in \mathbb{R}$. According to the symmetry of second derivatives this can be divided into two factors

$$0 = (\partial_t - c\partial_x)(\partial_t + c\partial_x)\,\phi\ . \tag{2}$$

With introduction of two variables, $\xi = x - ct$ and $\nu = x + ct$, now we can write:

$$0 = \partial_\xi \partial_\nu \phi\ . \tag{3}$$

Therefor the wave equation has the general solution

$$\phi(t, x) = f(\xi) + g(\nu) = f(x - ct) + g(x + ct) \tag{4}$$

which is a composition of a forward and a backward propagating elementary wave. As boundary condition we assume the initial time value and its derivative:

$$\phi_0(x) = \phi(0, x) = f(x) + g(x) =: A(x) \tag{5}$$

$$\Big(\partial_t \phi(0, x)\Big) = cf'(x) - cg'(x) =: B(x)\ . \tag{6}$$

Integrating eq. 6 results in

$$f(x) - g(x) = \int_{x_0}^{x} B(s)\,\mathrm{d}s \tag{7}$$

so that the separated solution is given by (here $x_0 = x(t = 0)$)

$$f(x) = \frac{1}{2}\left(A(x) + \frac{1}{c}\int_{x_0}^{x} B(s)\,\mathrm{d}s\right) \tag{8}$$

$$g(x) = \frac{1}{2}\left(A(x) - \frac{1}{c}\int_{x_0}^{x} B(s)\,\mathrm{d}s\right)\ . \tag{9}$$

Putting these together yields the general solution in dependence of the initial values, the d'Alembert's formula:

$$\phi(t, x) = \frac{1}{2}\left(A(x - ct) + A(x + ct) + \frac{1}{c}\int_{x-ct}^{x+ct} B(s)\,\mathrm{d}s\right) \tag{10}$$

This means, for given initial parameters the wave's evolution in time is clearly defined.

1.2. Rewriting the WE as two differential equations out of one

Now we introduce the time derivative of the field as a new variable: $\Pi(t, x) = \partial_t \phi$. Thereby the initial wave equation eq. 1 can be rewritten as a wave equation including the first-order derivative in time and second-order derivative in space:

$$\partial_t \Pi = \partial_{tt}\phi = c^2 \partial_{xx}\phi\ . \tag{11}$$

Now one can, in accordance to the notation we introduced above, regard the field's derivative at the initial time $\Pi(0, x)$ as $B(x)$. Typical values for this initial velocity $\Pi(0, x)$ are either 0 or a constant.

Example 1: $\Pi(0,x) = 0$ If we consider the field's initial velocity to be zero, $\Pi(0,x) = 0$ almost everywhere, this means the wave roughly is a pulse. Furthermore, if we assume no spatial boundaries the pulse will dominate the shape of the wave as time moves forward.

Example 2: $\Pi(0,x) =$ **const.** If we consider the initial velocity to be a constant, the wave function is at rest in a suitable inertial frame of reference. Hence, there will be no wave propagation.

1.3. Rewriting the WE as a fully first order system / eigenvalue problem / Matrix form of the wave equation

1.4. Rewriting the WE as a fully first order system / eigenvalue problem / Matrix form of the wave equation

Similar to the definition of the velocity variable $\Pi(t,x)$ we now introduce the slope variable as the spatial derivative of the field: $\chi = \partial_x \phi$. Thereby we can write the waveequation 1 as a first order system

$$\partial_t \vec{u} + \hat{M} \partial_x \vec{u} = \vec{S} \ . \tag{12}$$

Herein u is the state vector $\vec{u} = \{\phi, \Pi, \chi\}$, S is a source term and the matrix $\hat{M}$ can be found by simple coefficient comparision to be

$$\hat{M} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & c^2 \\ 0 & 1 & 0 \end{pmatrix} \ .$$

The source the then is of type $S = \{\Pi, 0, 0\}$.

1.4.1. Detailed matrix derivation

1.4.2. Choices for Initial Values

???

1.4.3. Eigenvalue Problem

The matrix formulation 12 in fact is an eigenvalue problem, where the eigenvalues can be easily found: $\lambda_1 = 0$ with eigenvector $v_1 = \{1, 0, 0\}$. For $\lambda_{2/3} = +c$, eigenvectors are $v_2 = \{0, c, 1\}$ and $v_3 = \{0, -c, 1\}$. Horizontally concatenated these yield the transformation matrix

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -c \\ 0 & 1 & 1 \end{pmatrix}$$

by which we can diagonalize matrix $\hat{M}$:

$$\Lambda = \mathrm{diag}\hat{M} = R^{-1}\hat{M}R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2c & 1/2 \\ 0 & -1/2c & 1/2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & c^2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c & -c \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & -c \end{pmatrix}$$

Here, one can see that the diagonalized matrix simply has the eigenvalues on the diagonal. For the eigenvalue problem $\hat{M}R = \Lambda R$ holds, so that $\hat{M} = R\Lambda R^{-1}$, which we can plug in 12:

$$\partial_t \vec{u} + R\Lambda R^{-1}\partial_x \vec{u} = \vec{S} \; . \tag{13}$$

Multiplying both sides by R^{-1} yields

$$R^{-1}\partial_t \vec{u} + \Lambda R^{-1}\partial_x \vec{u} = R^{-1}\vec{S} \; . \tag{14}$$

By introducing the new variable $w := R^{-1}\vec{u}$, this becomes

$$\partial_t w + \Lambda \partial_x w = R^{-1}\vec{S} \tag{15}$$

and we arrive at a decoupled system.

2. Numerical Differentiation

2.1. Forward, Backward and Central Difference Quotients

In order to numerically solve a wave equation one typically approximates all the derivatives by finite differences with regard to discrete grid points. Hence, the first step is to choose a discretization. For brevity we consider a uniform partition in the spatial dimension, so that

$$x_i = ih \quad \text{with } i = 0, ..., n-1$$

and stepsize $h = 1/(n-1)$ specifies the position on the domain $\Omega = [0, 1]$.

Now we want to find a formula for the approximation of the derivative of the field function $u(x)$ (we neglect the temporal dependency for a moment and will tackle that later). We consider the Taylor expansion of the field at "the next position", i.e. $u(x+h)$:

$$u(x+h) = \sum_{j=0}^{\infty} \frac{h^j}{j!} \left(\frac{\mathrm{d}^n u}{\mathrm{d}x^n} \right)_{x+h} = u(x) + h\frac{\mathrm{d}u}{\mathrm{d}x} + \frac{h^2}{2!}\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + \frac{h^3}{3!}\frac{\mathrm{d}^3 u}{\mathrm{d}x^3} + ... \tag{16}$$

Solving this for the first derivative yields

$$\frac{\mathrm{d}u}{\mathrm{d}x} = \frac{u(x+h) - u(x)}{h} - \frac{h}{2!}\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} - \frac{h^2}{3!}\frac{\mathrm{d}^3 u}{\mathrm{d}x^3} - ... \tag{17}$$

Taking into account only the first term of the right-hand side, this is called the *forward difference quotient*:

$$\mathrm{D}_h^+ u(x) = \frac{u(x+h) - u(x)}{h} \tag{18}$$

Similarly, one can approximate the derivative by expanding the value $u(x-h)$ in a Taylor series. This results in the *backward difference quotient*

$$\mathrm{D}_h^- u(x) = \frac{u(x) - u(x-h)}{h} \ . \tag{19}$$

In order two minimize errors we can combine the forward and backward difference to a *central difference quotient*

$$\mathrm{D}_h u(x) = \frac{1}{2}[\mathrm{D}_h^+ + \mathrm{D}_h^-]u(x) = \frac{u(x+h) - u(x-h)}{2h} \ . \tag{20}$$

2.2. Error Estimations

The discretization error is the deviation of the numerical approximation compared to the analytical value

$$\mathrm{E}_h = |\frac{\mathrm{d}u}{\mathrm{d}x} - \mathrm{D}_h u(x)| \ . \tag{21}$$

It stems from the finiteness of step size h, as well as from rounding errors arising from the computer accuracy when conducting numerical calculations. In the following we want to find expressions for the former error to compare them for the various difference quotients. In general both, forward and backward difference quotient, are consistent, i.e. for $h \to 0$ the

approximated derivative approaches the analytical derivative. From the comparison of 16 with 18 we find that the discretization error for the forward differentiation method is

$$\mathrm{E}_h^+ = \frac{h}{2!}\frac{\mathrm{d}^2u}{\mathrm{d}x^2} + \frac{h^2}{3!}\frac{\mathrm{d}^3u}{\mathrm{d}x^3} + \ldots \tag{22}$$

$$= \mathcal{O}(h) \ . \tag{23}$$

The same holds for the backward differentiation method. However, for the central difference quotient we get

$$\mathrm{E}_h = \frac{h^2}{3!}\frac{\mathrm{d}^3u}{\mathrm{d}x^3} + \frac{h^3}{4!}\frac{\mathrm{d}^4u}{\mathrm{d}x^4}\ldots \tag{24}$$

$$= \mathcal{O}(h^2) \ . \tag{25}$$

2.3. Approximating the Second Derivative

To approximate the second derivative of function $u(x)$, we recall the Taylor expansion of $u(x+h)$ and $u(x-h)$:

$$u(x+h) = u(x) + h\frac{\mathrm{d}u}{\mathrm{d}x} + \frac{h^2}{2!}\frac{\mathrm{d}^2u}{\mathrm{d}x^2} + \frac{h^3}{3!}\frac{\mathrm{d}^3u}{\mathrm{d}x^3} + \ldots \tag{26}$$

$$u(x-h) = u(x) - h\frac{\mathrm{d}u}{\mathrm{d}x} + \frac{h^2}{2!}\frac{\mathrm{d}^2u}{\mathrm{d}x^2} - \frac{h^3}{3!}\frac{\mathrm{d}^3u}{\mathrm{d}x^3} + \ldots \tag{27}$$

Adding up both sides respectively and solving for the second derivative yields

$$\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} - \frac{h^2}{3!}\frac{\mathrm{d}^3u}{\mathrm{d}x^3} - \ldots \tag{28}$$

which results in the differential operator with the error of $\mathcal{O}(h^2)$:

$$\mathrm{D}_h^2u(x) = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} \tag{29}$$

2.4. Approximating to a higher order

As we plan to use a Runge-Kutta time-integration with a error term that accumulates to $\mathcal{O}(h^4)$. Therefor it is useful to provide the derivatives with the same fourth order error term. We conduct this in the similar manner as the first and second order differential operators while simply including more terms of higher order. This leads to the following central derivative stencils:

$$\mathrm{D}_h^1u(x) = \frac{-u(x+2h) + 8u(x+h) - 8u(x-h) + u(x-2h)}{12h} \tag{30}$$

$$\mathrm{D}_h^2u(x) = \frac{-u(x+2h) + 16u(x+h) - 30u(x) + 16u(x-h) - u(x-2h)}{12h^2} \tag{31}$$

2.5. Convergence

In the following we show the results of the finite difference approximation in 1st and 2nd order for a set of example functions. These are:

$f_1 = (x-\frac{1}{2})^2 + x$	$f_1' = 2x$	$f_1'' = 2$
$f_2 = (x-\frac{1}{2})^3 + (x-\frac{1}{2})^2 + x$	$f_2' = 3(x-\frac{1}{2})^2 + 2(x-\frac{1}{2}) + 1$	$f_2'' = 6x - 1$
$f_3 = \sqrt{x}$	$f_3' = \frac{1}{2\sqrt{x}}$	$f_3'' = -\frac{1}{4}x^{-\frac{3}{2}}$
$f_4 = \sin(12\pi x)$	$f_4' = 12\pi\cos(12\pi x)$	$f_4'' = -(12\pi)^2\sin(12\pi x)$
$f_5 = \sin^4(12\pi x)$	$f_5' = 4\cdot 12\pi\sin^3(12\pi x)\cos(12\pi x)$	$f_5'' = 4(12\pi\sin(12\pi x))^2$ $\cdot\left(3\cos^2(12\pi x) - \sin^2(12\pi x)\right)$
$f_6 = g_a = \exp(-ax^2)$	$g_a' = -2ax\exp(-ax^2)$	$g_a'' = \exp(-ax^2)((2ax)^2 - 2a)$

We call a differential Operator D consistent of order k if,

$$something \tag{32}$$

If a sequence $x_1, x_2, ..., x_n$ converges to a value r and if there exist real numbers $\lambda > 0$ and $\alpha \geq 1$ such that

$$\lim_{n\to\infty} \frac{|x_{n+1} - r|}{|x_n - r|^\alpha} = \lambda \tag{33}$$

then we say that α is the **order** and λ the **rate** of convergence of the sequence.

3. Runge-Kutta time-integration

After the former chapters treated numerical method to calculate the (spatial) derivative of a given function, we now want to devote to the temporal integration in order to deal with the initial value problem (IVP) of an ordinary differential equation (ODE). I.e. we want to find the function $u(t)$, and all we know is its derivative

$$\frac{\mathrm{d}u}{\mathrm{d}t} = F(u) \tag{34}$$

and the initial value $u(t=0) = u_0$. We assume $F(u)$ to be continuous, especially Lipschitz continuous, to make sure that there exists exactly one solution for the IVP.

3.1. General Principle of Numerical Integration Methods

Similarly to the spatial differential methods, the first step is to discretize the respective interval. Generally, this discretization does not have to be equidistant, so we define

$$\Delta t_i = t_{i+1} - t_i > 0 \tag{35}$$

to be the local step sizes for $i = 0, ..., n-1$. Now we search for approximate values $U_i = U(t_i) \approx u(t_i) = u_i$. By integrating both sides of 34 over t we find

$$u(t_{i+1}) = u(t_i) + \int_{t_i}^{t_{i+1}} F(t, u(t))\mathrm{d}t \qquad \text{for } i = 0, ..., n-1 \tag{36}$$

Different methods are distinguished, depending on how the integral is solved. *One-step methods* take into account only the previous value, hence calculating U_{i+1} from U_i. *Multi-step methods* use more than one previous values and *extrapolation algorithms* use the Richardson approximation to improve results from integration by trapezoidal or rectangular integration methods. In the following we want to focus on the one-step methods, especially the Runge-Kutta methods, as this is the most common one in usage.

One very simple approach is the Euler-Cauchy method, where U_{i+1} is approximated just as a linear continuation from U_i. As this method fails even for very simple functions (e.g. think of a quadratic function which very fast will diverge from that tangential approximation), this method was improved by taking into account the function's further derivatives (tangents).

3.2. Runge-Kutta Methods

A Runge-Kutta method (RK method) in general can be written as

$$U_{n+1} = U_n + \Delta t_n \sum_{i=1}^{s} b_i k_i(t_n, U_n, \Delta t_n) \tag{37}$$

$$\text{where} \quad k_i(t, U, \Delta t) = F(t + c_i \Delta t, U + \Delta t \sum_{j=0}^{i} a_{ij} k_j(t, U, \Delta t)) \tag{38}$$

with the set of characteristic coefficients (a_{ij}, b_i, c_i) for $i = 1, ..., s$ where s denotes the so-called *stage* of the algorithm. It estimates a function's next point by evaluating the weighted

average of various derivates at pieces of the next time step. The most common variant of doing so is using half time steps and therefor getting four helping tangents, the average value of which then acts as the linear continuations increase. Hence, this version is referred to as the *Fourth Order RK method, RK4* or simply *Classical RK method* and will be treated in more detail below.

The Butcher Table The characterisistic coefficients c_i, b_i and a_{ij} are ususally written in the *Butcher table*, i.e.

$$\begin{pmatrix} c_i & a_{ij} \\ & b_i^T \end{pmatrix} = \begin{pmatrix} 0 & & & & & \\ c_2 & a_{21} & a_{22} & \dots & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & & \ddots & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} & \\ & b_1 & b_2 & \dots & b_{s-1} & b_s \end{pmatrix} \tag{39}$$

Explicit and Implicit RK methods A RK method can be either explicit or implicit, depending on how many coefficients a_{ij} are taken into account for the calculation of k_i. If a_{ij} are set to zero for $i \leq j$, the internal stages $k_{n1}, ..., k_{ns}$ can be be computed directly, i.e. explicitly. With respect to the Butcher table, this means that explicit methods only have the lower diagonal elements of the the a_{ij} nonzero, i.e. a triangular matrix. On the other hand, if all a_{ij} are used and hence the matrix is fully occupied, then the calculation of k_i ends up as a nonlinear system and is therefor called implicit. Implicit methods tend to have a higher stability, as more terms are used. Their application is required for some partial differential equations, especially for stiff problems.

Estimated Error From the stage number s one can estimate the magnitude of error, e.g. for $s \leq 4$ the local error is of order $s + 1$. For larger stage numbers this linear relation will flatten so that the order of error grows slower than the stage number.

The classical RK method: RK4 .... (Derivation of how RK4 has error of order t5) .....
The RK4 method usually comes with the Butcher table

$$\begin{pmatrix} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ & 1/6 & 1/3 & 1/3 & 1/6 \end{pmatrix} \tag{40}$$

and hence the various tangents are

$$k_1 = F(t, U) \tag{41}$$

$$k_2 = F(t + \frac{1}{2}\Delta t, U + \frac{1}{2}\Delta t k_1) \tag{42}$$

$$k_3 = F(t + \frac{1}{2}\Delta t, U + \frac{1}{2}\Delta t k_2) \tag{43}$$

$$k_4 = F(t + \Delta t, U + \Delta t k_3), \tag{44}$$

which by being weighted yield the estimated function value

$$U_{n+1} = U_n + \Delta t \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \right) . \tag{45}$$

Example: Solving the Harmonic Oscillator Equation using RK4 Given is the Hamiltonian for the harmonic oscillator

$$H(q,p) = \frac{1}{2}\left(\frac{p^2}{m} + q^2 m \right) \tag{46}$$

where $q = \omega x$ and $p = m\dot{x}$, as well as the initial values q_0 and p_0. The Hamiltonian yields the equations of movement, i.e. the time-derivatives of the generalized coordinates q and p

$$\dot{q} = \frac{\partial H}{\partial q} = \frac{p}{m} \qquad\qquad \dot{p} = \frac{\partial H}{\partial p} = -qm \tag{47}$$

so that we can write our system of ODEs in simple vector form, introducing $u = \{q, p\}$:

$$\tfrac{\mathrm{d}}{\mathrm{d}t}u = \begin{bmatrix} \frac{p}{m} \\ -qm \end{bmatrix} . \tag{48}$$

In the following we present a possible implementation to solve this initial value problem, given by 48 and $u_0 = q_0, p_0$, using the RK4 method. The implementation is divided in two functions: `F(u,t)` and `rk4(u0, deltat, T, F)`. The former returns the differentiation of u, i.e. 48. The latter is the actual approximation of function values. Here, above's algorithm is straight-forward implemented.

```
def F(u,t):
  dqdt = u[1]/m
  dpdt = -u[0]*m
  return np.array([dqdt,dpdt])

def rk4(u0,deltat,T,F):
  u = np.zeros([n,2])
  u[0,0] = q0
  u[0,1] = p0
  t=1
  for i in range(0,n-1):
    k1 = F(u[i,:], t)
    k2 = F(u[i,:] + 0.5*deltat* k1,t + 0.5*deltat)
    k3 = F(u[i,:] + 0.5*deltat* k2,t + 0.5*deltat)
    k4 = F(u[i,:] + deltat* k3,t + deltat)
    u[i+1,:] = u[i,:] + deltat*(1/6*k1 + 1/3*k2 +1/3*k3 + 1/6*k4)
  return u

u_rk4 = rk4(u0,deltat,T,F)
```

It shall be noted, first, that `u` is an array of 2 columns (one for values of q and p respectively) and as long as the number of time steps. Second, there occurs a time variable `t` in the implemented function `F(u,t)` which is actually unnecessary for the harmonic oscillator. Nevertheless it is written there, just to show, that there could possibly be a time dependence (e.g. for the heat equation this is the case).

A. Derivatives for example functions

B. Plots of Runge Kutta Method

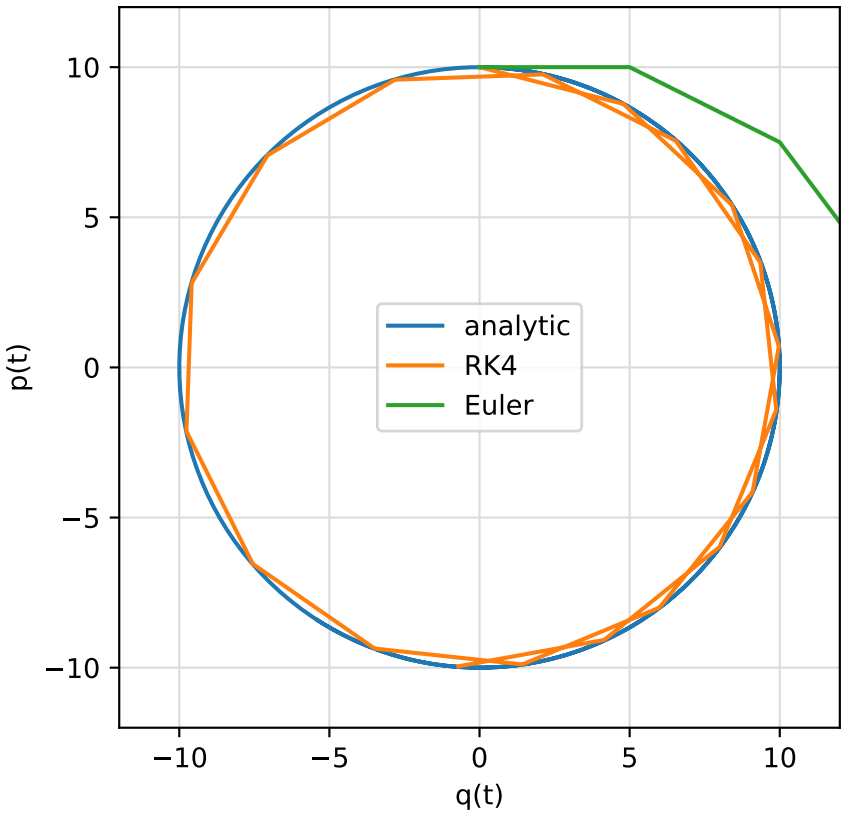


Figure 11: This shows the evolution of the harmonic oscillator IVP given in the main section in phase space, solved by different mthods. Initial values are $u_0 = \{q_0, p_0\} = \{5, 0\}$. Stepsize `deltat` $= 0.4$. One can see that the phase path calculated using Euler's method very quickly deviates from the analytical solution given by $q(t) = \frac{p_0}{m}\sin(t) + q_0\cos(t)$ and $\frac{\mathrm{d}q}{\mathrm{d}t}(t) = \frac{p_0}{m}\cos(t) - q0\sin(t)$. The RK4 solution instead remains one the circular path, hence is energy conserving.

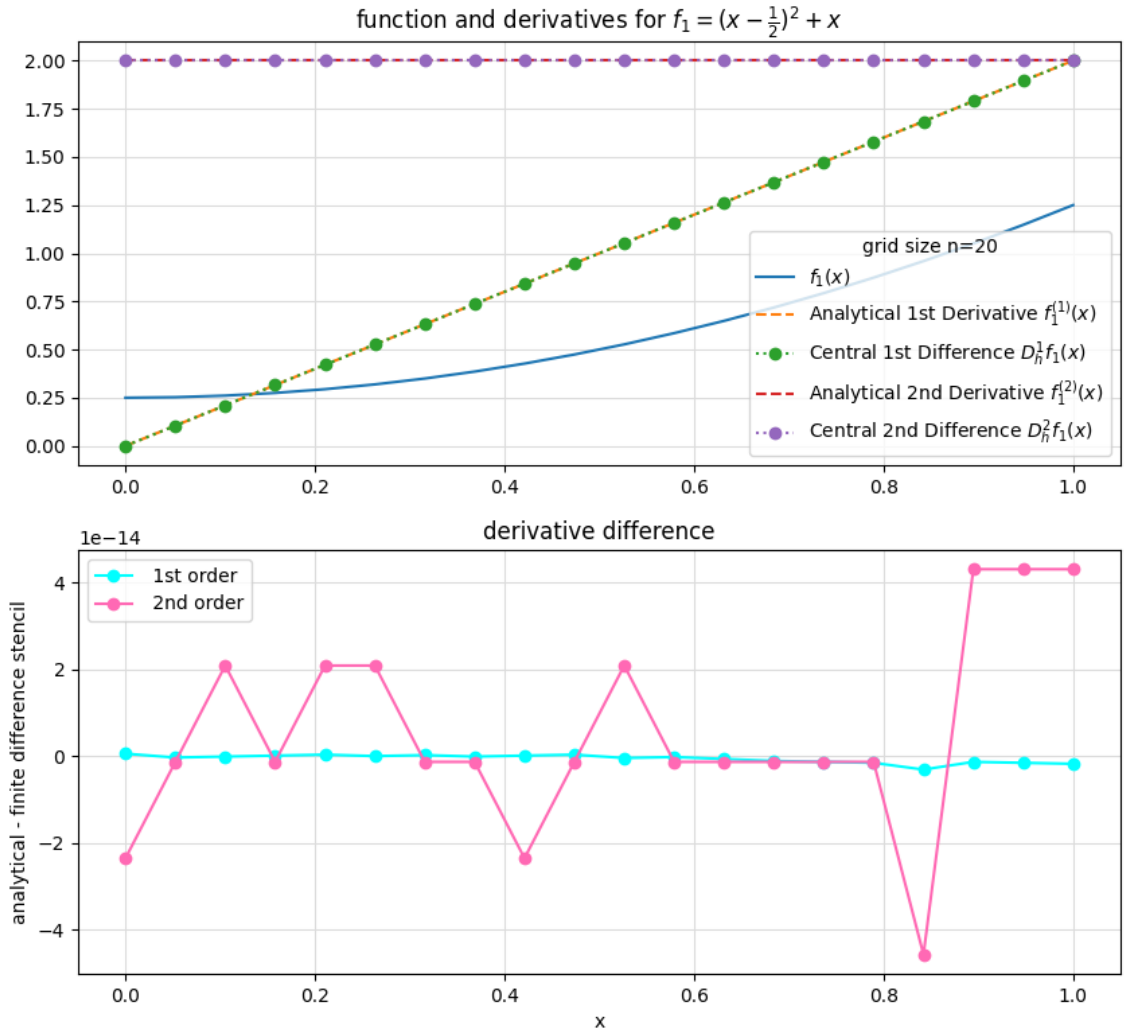


Figure 1: Upper panel: Function f_1 in blue, dashed lines symbolize its analytical derivatives, dotted lines the finite difference approximation to the derivative. $n = 20$ Lower panel: The cyan and pink lines show the difference between analytical and numerically generated derivatives. One can see here, that for this quadratic function the error only consists of the roundoff error which stems from the finite computer storage

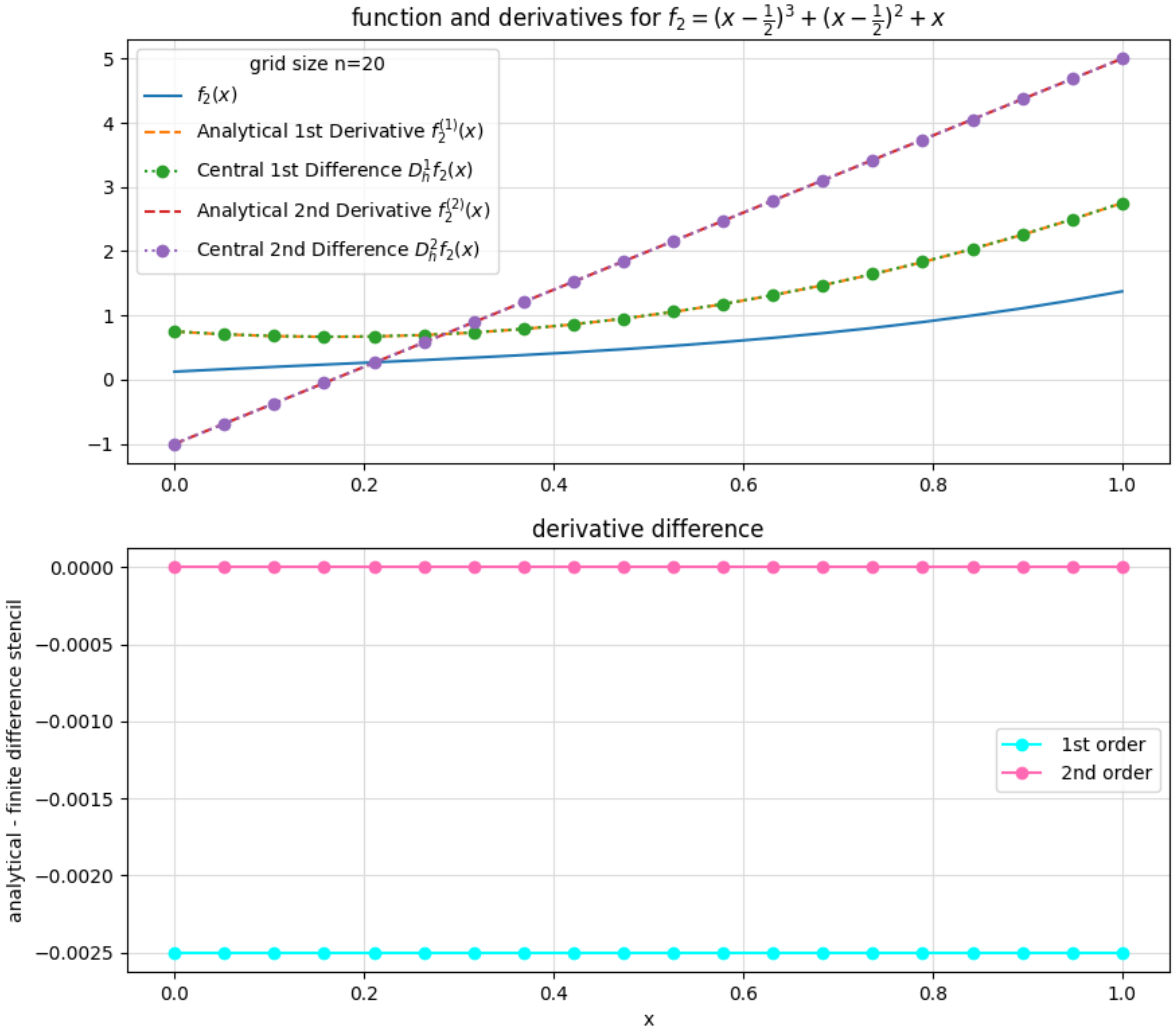


Figure 2: Here the error term for both differences is constant. As the function itself is of order 3, one now truely has a truncation error.

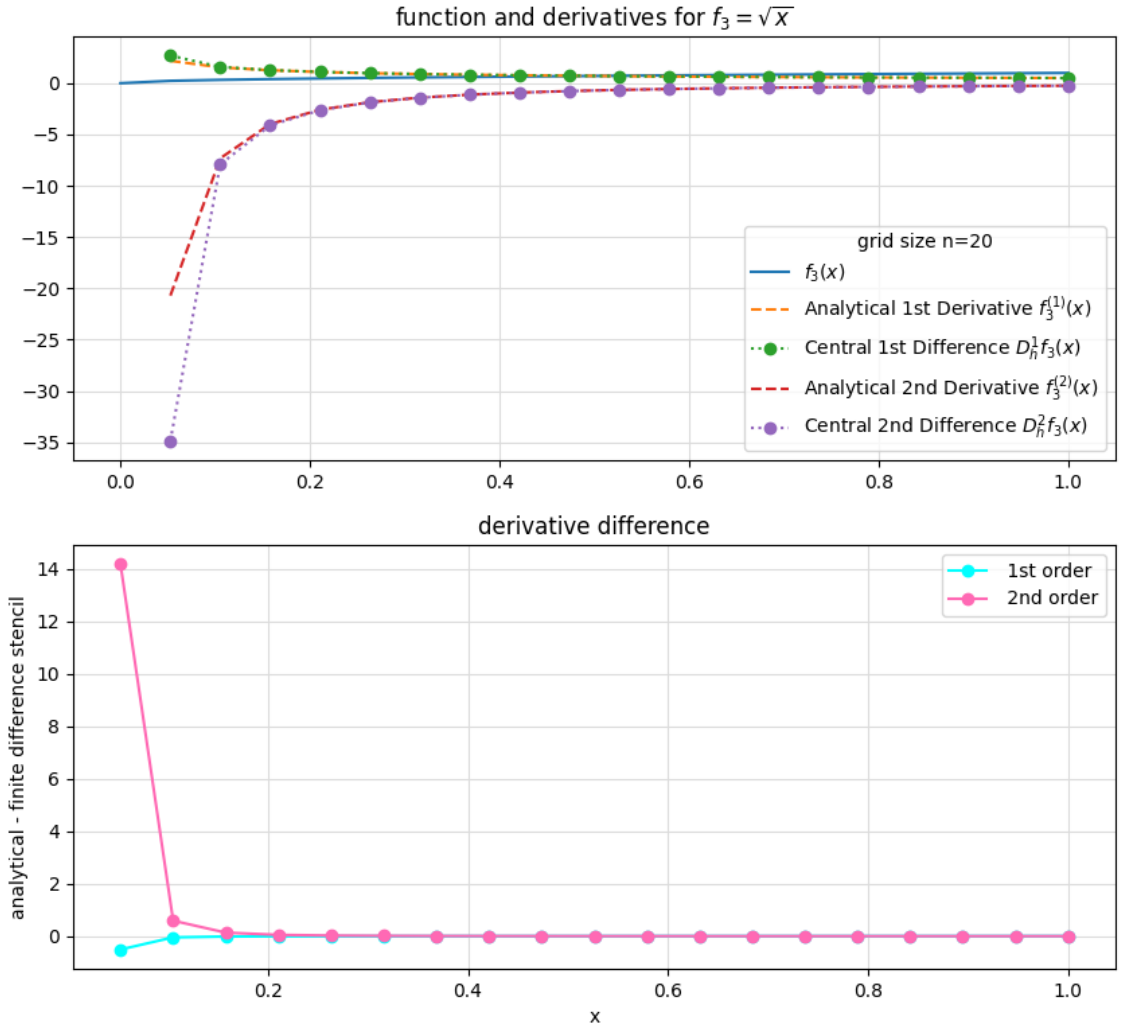


Figure 3: Here the error term for both derivatives vanishes quickly. It stems from the fact, that one uses central difference stencils even though one doesn't have all the function values necessary (since one cannot take the squareroot of negative numbers

.

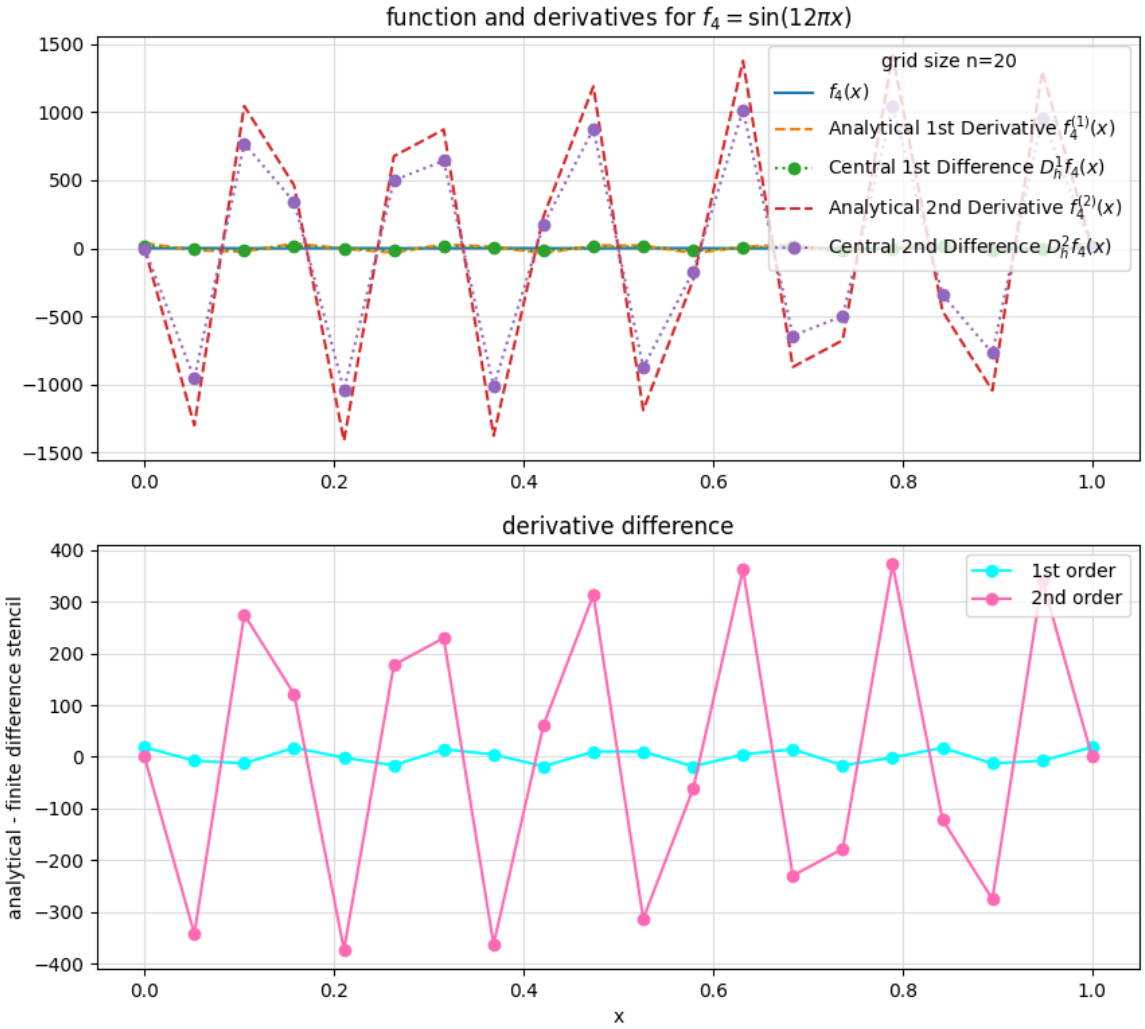

function and derivatives for $f_4 = \sin(12\pi x)$
grid size n=20
$f_4(x)$
Analytical 1st Derivative $f_4^{(1)}(x)$
Central 1st Difference $D_h^1 f_4(x)$
Analytical 2nd Derivative $f_4^{(2)}(x)$
Central 2nd Difference $D_h^2 f_4(x)$
derivative difference
1st order
2nd order
analytical - finite difference stencil
x

Figure 4

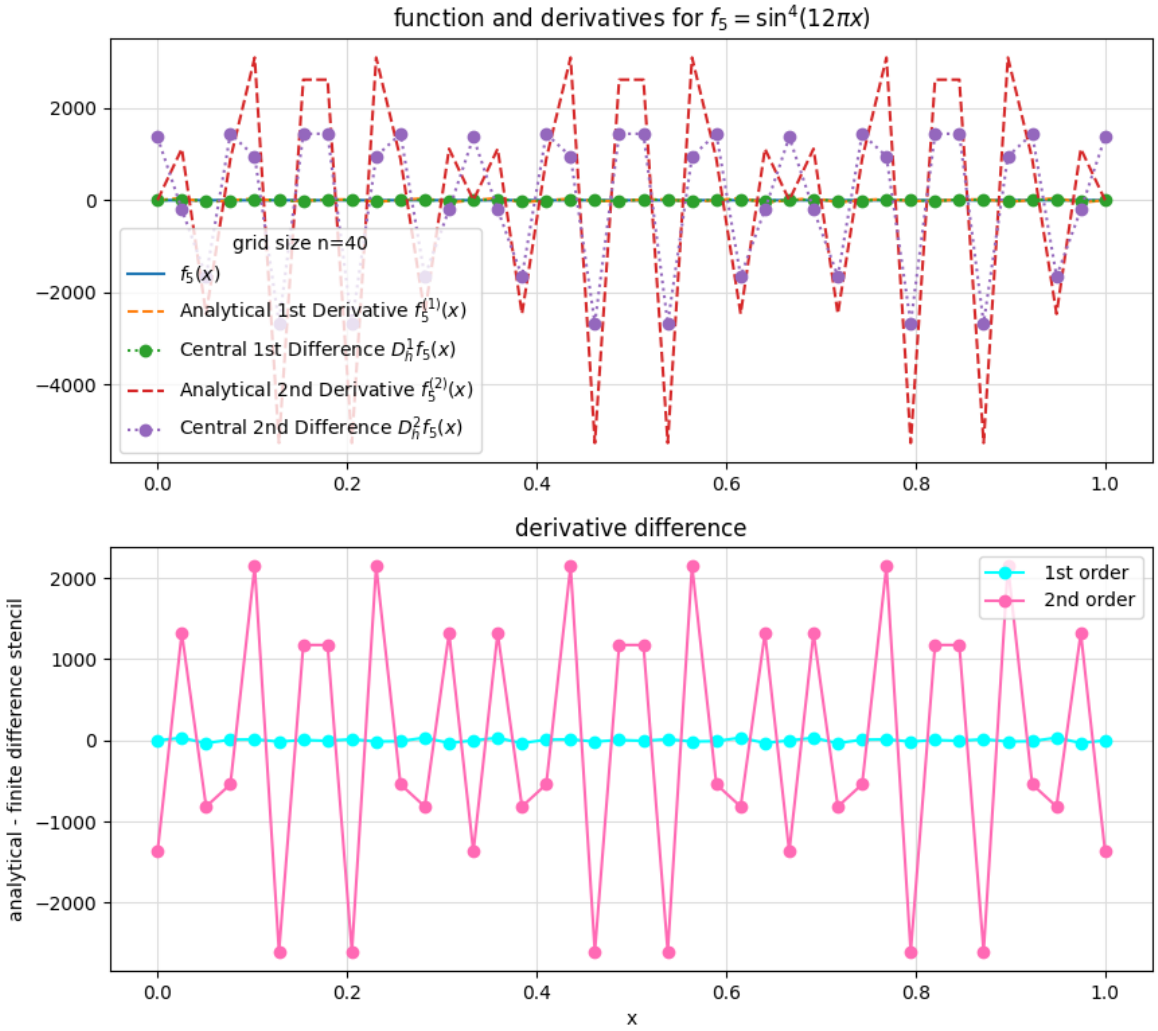

function and derivatives for $f_5 = \sin^4(12\pi x)$
grid size n=40
$f_5(x)$
Analytical 1st Derivative $f_5^{(1)}(x)$
Central 1st Difference $D_h^1 f_5(x)$
Analytical 2nd Derivative $f_5^{(2)}(x)$
Central 2nd Difference $D_h^2 f_5(x)$
derivative difference
1st order
2nd order
analytical - finite difference stencil
x

Figure 5

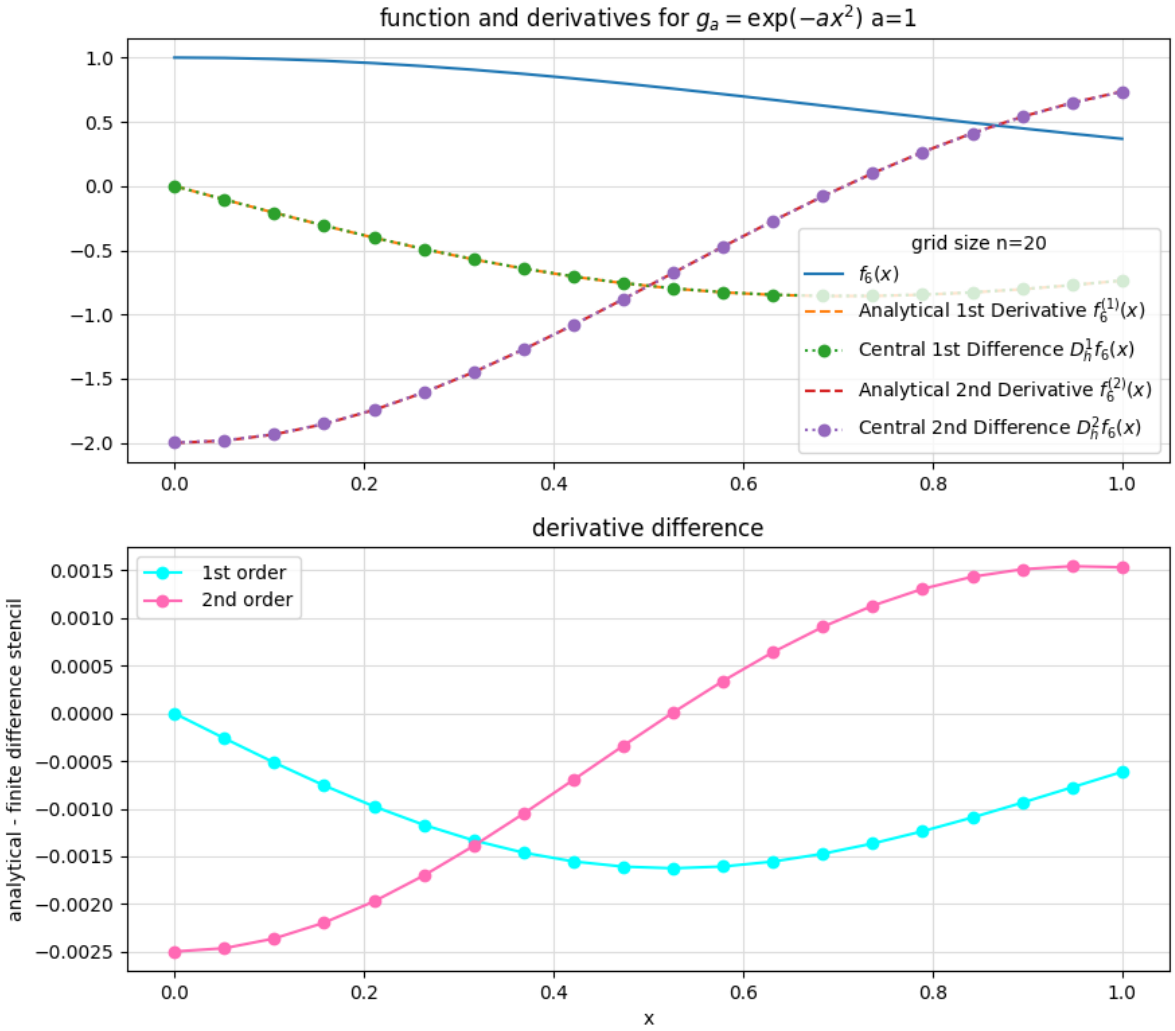
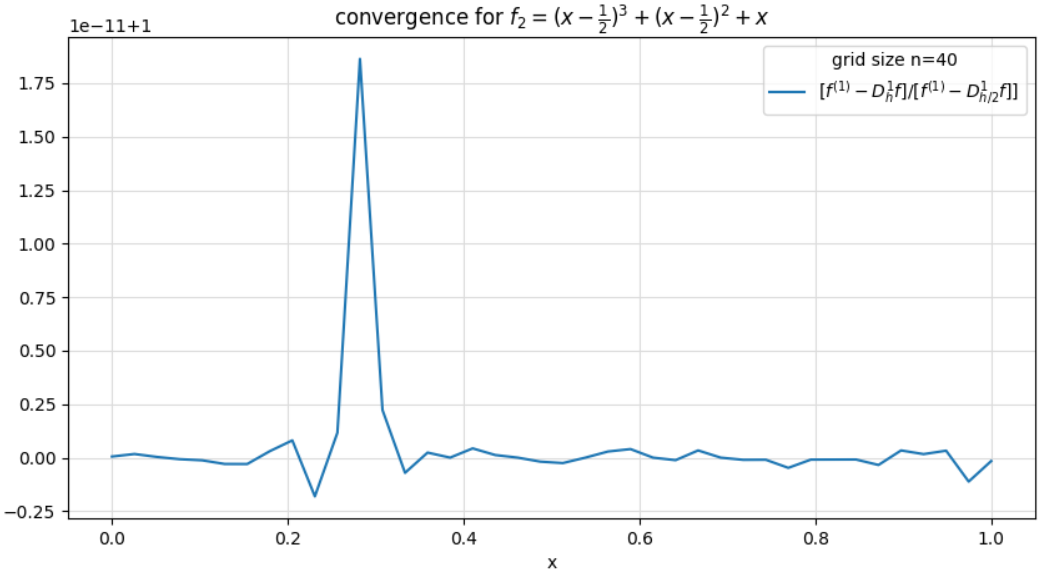

function and derivatives for $g_a = \exp(-ax^2)$ a=1
grid size n=20
$f_6(x)$
Analytical 1st Derivative $f_6^{(1)}(x)$
Central 1st Difference $D_h^1 f_6(x)$
Analytical 2nd Derivative $f_6^{(2)}(x)$
Central 2nd Difference $D_h^2 f_6(x)$
derivative difference
1st order
2nd order
analytical - finite difference stencil
x

Figure 6



(a) convergence

(b) self-convergence

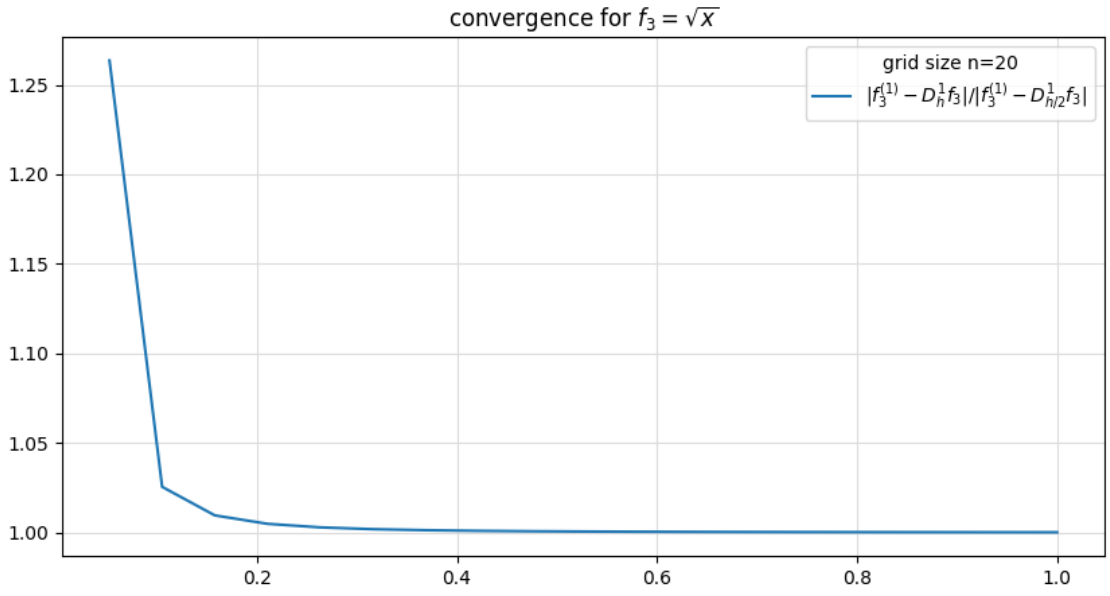


Figure 8

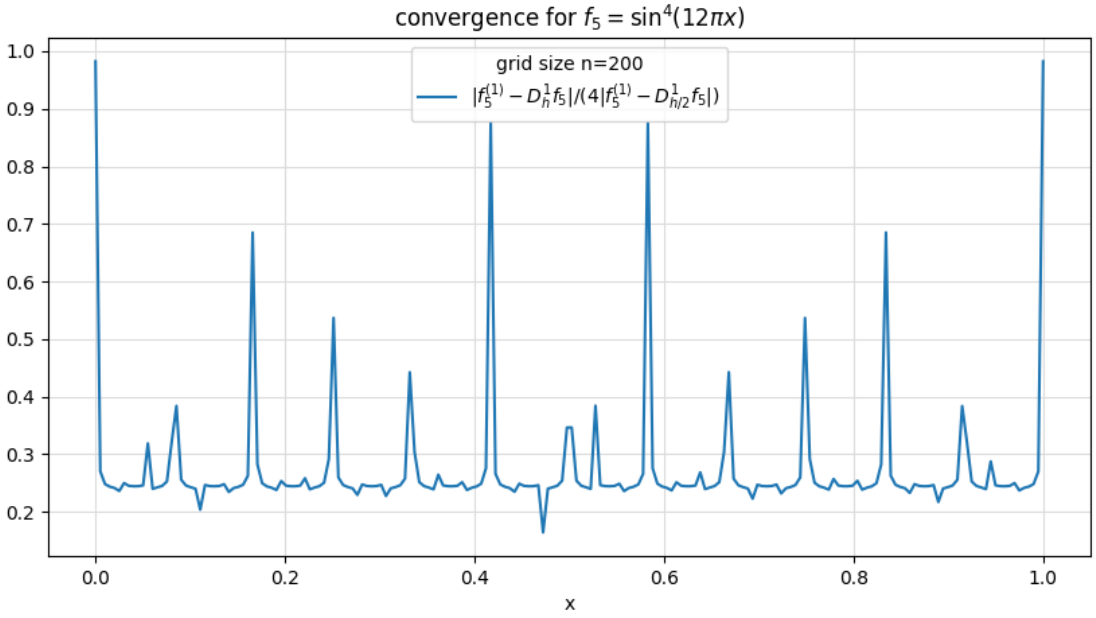


Figure 9: convergence for function

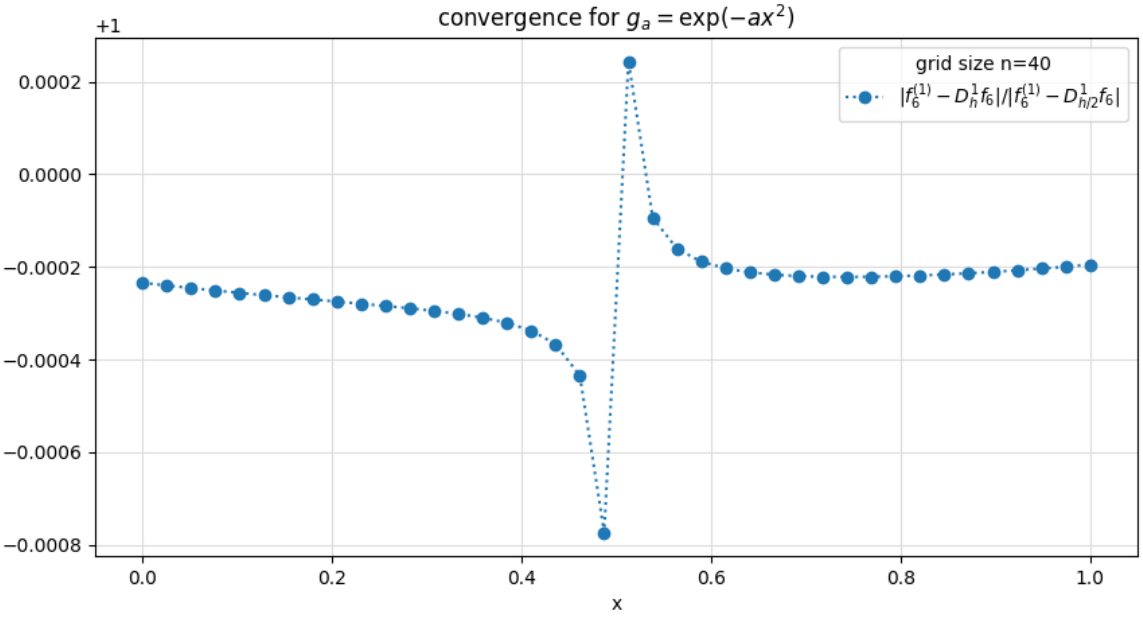


Figure 10: convergence for function