

Accuracy and Stability of Numerical Integration Methods in Penning Trap Simulations

Lars Bogner
(Dated: October 15, 2025)

TODO: ABSTRACT HERE.

I. INTRODUCTION

Penning traps are useful appliances to capture charged particles without contact to any matter surrounding the particles [1, 11]. This property of Penning traps is especially useful in the study of antimatter, as any contact to matter would lead to the direct annihilation of the antimatter. Therefore, Penning traps have become an essential tool in the antimatter research. Experiments like ALPHA-g [2] and AEGIS [3] at CERN have successfully deployed Penning traps for such applications.

To plan future antimatter experiments it is essential to be able to simulate Penning traps. Without the ability of the behavior of particles within such a trap, a precise planning of the Penning trap and antimatter experiment is not possible. Key criterion for a successful simulation in this context is accuracy of the numerical simulation in terms of precise trajectories and key statistical properties of the captured particles, e.g. the kinetic energy. A second metric that is very important to the usefulness of such numerical simulations is the computational performance of the simulation as this decides whether the simulation is even possible or not with finite computing resources. To find the optimal balance between the finite computing power available and an accurate result, different integration algorithms will be compared to solve the ordinary differential equations at play in the Penning Trap.

In section II the methods employed in the simulation will be expanded on. In detail the implementation of the source code will be discussed as well as the specific Penning trap used for the study, introduced. Thirdly the algorithms behind the numerical integration methods will be explained. Section III will show the results obtained from our implementation and discuss their practical implications. This section will be split into four parts expanding on the accuracy of the trajectories, the energy conservation in the system, the computational performance of the different algorithms as well as a discussion on the limit of many particles in the trap. Lastly in section IV our findings will be summarized and consequences of the results derived.

II. METHODS

Penning trap

Penning traps use a combination of a magnetic and electric field to confine the trajectory of a charged particle to a finite volume over long periods of time [1, p. 9-10]. Finite volume in this context refers to volumes in the order of mm³ up to cm³. Long time ranges imply a period that exceeds the period of motion of the particle by multiple orders. For the studied case we will use an ideal Penning trap, i.e. there are no inhomogenities in the fields. The magnetic field will be defined as

$$B(\vec{r}, t) = B_0 \Theta(d - |\vec{r}|) \cdot \vec{e}_z, \quad (1)$$

with the Heaviside function $\Theta(x)$, and we define two electric fields via the electric potentials

$$V_{\text{stat.}}(\vec{r}, t) = \frac{V_0 \cdot (2z^2 - x^2 - y^2) \Theta(d - |\vec{r}|)}{2d^2} \quad (2)$$

and

$$V_{\text{dyn.}}(\vec{r}, t) = (1 + f \cos \omega_V t) \cdot V_{\text{stat.}}(\vec{r}, t). \quad (3)$$

Equations of motion

Using Newton's equations of motion, $\ddot{\vec{r}} = \vec{F}/m$, we can derive ordinary differential equations for the position of a particle which is subject to eqs. (1) and (2). As the magnetic field is parallel to the z -axis, the differential equation for the vertical component is straightforward to derive. Using the electric field resulting from $V_{\text{stat.}}$ (see appendix A for details on all electric field components) we find

$$\ddot{z} + \omega_z^2 z \equiv \ddot{z} + \frac{2qV_0}{md^2} z = 0. \quad (4)$$

In the transversal plane, the equations of motion are coupled due to the Lorentz force $\vec{F} = q\vec{v} \times \vec{B}$. Using the electric field resulting from $V_{\text{stat.}}$ (see appendix A for details on all electric field components) we find

$$\ddot{x} - \omega_0 \dot{y} - \frac{1}{2} \omega_z^2 x \equiv \ddot{x} - \frac{qB_0}{m} \dot{y} - \frac{qV_0}{md^2} x = 0, \quad (5)$$

$$\ddot{y} + \omega_0 \dot{x} - \frac{1}{2} \omega_z^2 y \equiv \ddot{y} + \frac{qB_0}{m} \dot{x} - \frac{qV_0}{md^2} y = 0. \quad (6)$$

Using the sum of equations: (5) + $i(6)$, we can use the general definition $f(t) \equiv x(t) + iy(t)$ to derive a single complex differential equation for the transversal motion

$$\ddot{f} + i\omega_0 \dot{f} - \frac{1}{2}\omega_z^2 f = 0. \quad (7)$$

After solving the differential equation, the real and imaginary parts of $f(t)$ will correspond to the x and y components of the trajectory, respectively. Equation (7) is a damped harmonic oscillator equation with the general solution

$$f(t) = A_+ e^{-i(\omega_+ t + \phi_+)} + A_- e^{-i(\omega_- t + \phi_-)}, \quad (8)$$

with the two characteristic frequencies

$$\omega_{\pm} = \frac{\omega_0 \pm \sqrt{\omega_0^2 - 2\omega_z^2}}{2}. \quad (9)$$

This solution is valid for $\omega_0^2 > 2\omega_z^2$. In the case of $\omega_0^2 \leq \omega_z^2$, the oscillation frequencies become complex, leading to exponential growth of the trajectory in the transversal plane. This condition is known as the stability criterion for a Penning trap [1, p. 67]. In terms of the physical parameters of the trap, the stability criterion can be expressed as

$$\frac{q}{m} B_0^2 > \frac{2V_0}{d^2}. \quad (10)$$

Given this condition, the solution for the transversal motion will be bound, i.e. $|f(t)| < \infty$ for all $t > 0$. Given the stability criterion, the two terms in eq. (8) can be identified as sinusoidal motions with amplitudes $A_{\pm}$. If the two terms are in phase the upper limit of $|f| \equiv R_+$ will be $R_+ = A_+ + A_-$, while the lower limit for the case of antiparallel phases will be the absolute of the difference in the two amplitudes, $R_- = |A_+ - A_-|$. A analytical solution for a special case of initial conditions is derived in appendix B.

Multiple particles

For the case of multiple particles, it is important to include the particle-particle interactions. For this analysis we will only consider the Coulomb interaction between the particles, neglecting magnetic interactions. The force on particle i due to all other particles j is then given by

$$\vec{F}_i = \frac{q_i}{4\pi\epsilon_0} \sum_{j \neq i} \frac{q_j (\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|^3}. \quad (11)$$

Numerical Integration Methods

Forward Euler method

The simplest numerical method to solve the ordinary differential equation system is the Forward Euler method.

It is a first order method, meaning that the local truncation error per step is on the order of $\mathcal{O}(h^2)$, with h being the step size. The global error after N steps is therefore on the order of $\mathcal{O}(h)$. The method is explicit, meaning that the state of the system at the next time step can be calculated directly from the current state. Given a general ordinary differential equation of the form

$$\dot{y}(t) = f(t, y(t)), \quad (12)$$

the Forward Euler method updates the state of the system as follows:

$$y_{n+1} = y_n + hf(t_n, y_n). \quad (13)$$

For our second order differential equations, we first rewrite them as a system of first order equations. The next state of the system is then calculated using the current position and velocity as

$$\vec{r}_{n+1} = \vec{r}_n + h\vec{v}_n, \quad (14)$$

$$\vec{v}_{n+1} = \vec{v}_n + h \frac{\vec{F}(\vec{r}_n, \vec{v}_n, t_n)}{m}. \quad (15)$$

This method is computationally inexpensive, as we only need a single force evaluation per time step. However, it is known to be not very accurate.

Runge-Kutta 4 method

The Runge-Kutta 4 (RK4) method is a popular and more accurate method for solving ordinary differential equations. It is a fourth-order method, meaning that the local truncation error per step is on the order of $\mathcal{O}(h^5)$, and the global error after N steps is on the order of $\mathcal{O}(h^4)$. The RK4 method calculates the next state of the system using a weighted average of four different estimates of the slope (the derivative) at different points within the time step. Given a general ordinary differential equation of the form

$$\dot{y}(t) = f(t, y(t)), \quad (16)$$

the RK4 method updates the state of the system as follows:

$$k_1 = hf(t_n, y_n), \quad (17)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right), \quad (18)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \quad (19)$$

$$k_4 = hf(t_n + h, y_n + k_3), \quad (20)$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \quad (21)$$

As this method is again a generalized solver for ODE of the first order, we again rewrite our second order differential equations as a system of first order equations.

Velocity-Verlet method

In contrast to the previous two methods, the Velocity-Verlet method is a symplectic integrator which is specifically designed for second order differential equations of the form

$$\ddot{\vec{r}}(t) = \frac{\vec{F}(\vec{r}(t), t)}{m}. \quad (22)$$

The method is time-reversible and conserves energy better over long time periods compared to non-symplectic methods like Forward Euler and RK4. The Velocity-Verlet method updates the position and velocity of the system as follows:

$$\vec{r}_{n+1} = \vec{r}_n + h\vec{v}_n + \frac{h^2}{2}\vec{a}_n, \quad (23)$$

$$\vec{a}_{n+1} = \frac{\vec{F}(\vec{r}_{n+1}, t_{n+1})}{m}, \quad (24)$$

$$\vec{v}_{n+1} = \vec{v}_n + \frac{h}{2}(\vec{a}_n + \vec{a}_{n+1}), \quad (25)$$

It is important to note that the Velocity-Verlet integrator requires forces to be independent of velocity. In the case, that the algorithm is applicable, it is a very efficient method, as it only requires a single force evaluation per time step, while still being a second order method with a local truncation error per step on the order of $\mathcal{O}(h^3)$ and a global error after N steps on the order of $\mathcal{O}(h^2)$.

Boris algorithm

TODO

Code Structure

The basic framework for the numerical analysis is based on a `PenningTrap` class, which contains all the particles present in the trap, as well as a parametrization of the electric and magnetic fields. The particles are represented by a `Particle` class, which contains the physical properties of the particle, as well as its current position and velocity. With this information, the `PenningTrap` class can calculate the forces acting on each particle, including the external fields and the particle-particle interactions. The particle-particle interactions can be toggled on and off, allowing for a simulation of both scenarios. The external fields can be modified by supplying a field-method of the form `external_field(const arma::vec& r, double t, const PenningTrap& trap)`. The reference to the `PenningTrap` allows for the parameters of the field to be stored in the trap object. The implementation of the `Particle` and `PenningTrap` classes can be found in `/src/project3/include/classes.hpp` of the project repository, as well as in the corresponding source file `/src/project3/src/classes.cpp`.

Numerical Methods Implementation

All numerical methods are implemented as classes inheriting from a base class `Solver`. The base class contains a reference to the `PenningTrap` object, as well as the time step size. The recording of particle properties over time, like position and velocity is part of the general `Solver` class. Each derived class implements the `step()` method, which updates the state of the system by one time step using the respective numerical method. The implementation of the `Solver` class and its derived classes can be found in `/src/project3/include/solvers.hpp` of the project repository, as well as in the corresponding source file `/src/project3/src/solvers.cpp`. The following solvers are implemented: `ForwardEulerSolver`, `RK4Solver`, `VelocityVerletSolver` and `AnalyticalSolver`, which implements the analytical solution for a special case of initial conditions (see appendix B).

Tools

TODO

III. RESULTS AND DISCUSSION

Numerical Accuracy of Trajectories

Performance and Efficiency

Many-Body Simulations

Symplectic Properties

IV. CONCLUSION

Appendix A: Electric Field Equations

Here we derive the electric field equations used in section II. The electric field following from $V_{\text{stat.}}$ defined in eq. (2) is given by

$$E_{\text{stat.},x} = -\frac{\partial V_{\text{stat.}}}{\partial x} = \frac{V_0}{d^2}x, \quad (A1)$$

$$E_{\text{stat.},y} = -\frac{\partial V_{\text{stat.}}}{\partial y} = \frac{V_0}{d^2}y, \quad (A2)$$

$$E_{\text{stat.},z} = -\frac{\partial V_{\text{stat.}}}{\partial z} = -\frac{2V_0}{d^2}z, \quad (A3)$$

for all $|\vec{r}| < d$ and zero otherwise. The electric field following from $V_{\text{dyn.}}$ is simply

$$E_{\text{dyn.}} = (1 + f \cos \omega_V t) \cdot E_{\text{stat.}} \quad (A4)$$

Appendix B: Special Case Analytical Solution

In the special case of a single particle in the potential $V_{\text{stat.}}$, with the initial conditions $\vec{r}(t_0) = (x_0, 0, z_0)$ and $\vec{v}(t_0) = (0, v_0, 0)$, we can derive an analytical solution for the trajectory. The solution for the z component is straightforward

$$z(t) = z_0 \cos(\omega_z t). \quad (\text{B1})$$

From the equation system

$$f(0) = A_+ e^{-i\phi_+} + A_- e^{-i\phi_-} \equiv x_0, \quad (\text{B2})$$

$$\dot{f}(0) = -i\omega_+ A_+ e^{-i\phi_+} - i\omega_- A_- e^{-i\phi_-} \equiv iv_0, \quad (\text{B3})$$

we derive $\phi_{\pm} = 0$ as $f(0) \in \mathbb{R}$ and therefore $A_+ + A_- = x_0$ and $\omega_+ A_+ + \omega_- A_- = -v_0$. Solving this system of equations for $A_{\pm}$ we find

$$A_{\pm} = \pm \frac{v_0 + \omega_{\mp} x_0}{\omega_- - \omega_+}. \quad (\text{B4})$$

-
- [1] Manuel Vogel. *Particle Confinement in Penning Traps: An Introduction*, volume 126 of *Springer Series on Atomic, Optical, and Plasma Physics*. Springer International Publishing.
 - [2] W. A. Bertsche. Prospects for comparison of matter and antimatter gravitation with ALPHA-g. 376(2116):20170265. Publisher: Royal Society.
 - [3] Paola Scamporrì and James Storey. The AEGIS experiment at CERN for the measurement of antihydrogen gravity acceleration. 29(17):1430017. Publisher: World Scientific Publishing Co.