Numerical simulation of a Penning trap

Alessio Canclini, Filip von der Lippe
(Dated: October 18, 2022)

We provide an overview of how to structure a scientific report. For concreteness, we consider the example of writing a report about an implementation of the midpoint rule of integration. For each section of the report we briefly discuss what the purpose of the given section is. We also provide examples of how to properly include equations, tables, algorithms, figures and references.

I. INTRODUCTION

The purpose of this report is to study the effects of a Penning trap. This is a device used to store or "trap" charged particles using static electric and magnetic fields as shown in figure 1. These particles can then be used for a variety of experiments. Examples of this are the ALPHA, AEgIS and BASE experiments at CERN, these use Penning traps to control antimatter.

Materials to construct a physical Penning trap are very costly, we will therefore be using a numerical approach to simulate a Penning trap. To implement such a simulation we will be working with a system of coupled non-linear differential equations. These are very difficult and often impossible to solve analytically. An example some readers might be familiar with are the famous Navier-Stokes equations, the solving of which would be rewarded with a million dollar prize. In addition to the material cost, the complexity of the equations also leads us to the use of numerical methods.

Section II will describe the mathematical and physical background as well as concrete algorithms which in this case will be implemented in C++, but can be written in any programming language.

In section III we present...

A detailed discussion of the algorithms' and results in presented in section IV, followed by a summary and potential for further experiments in section V.

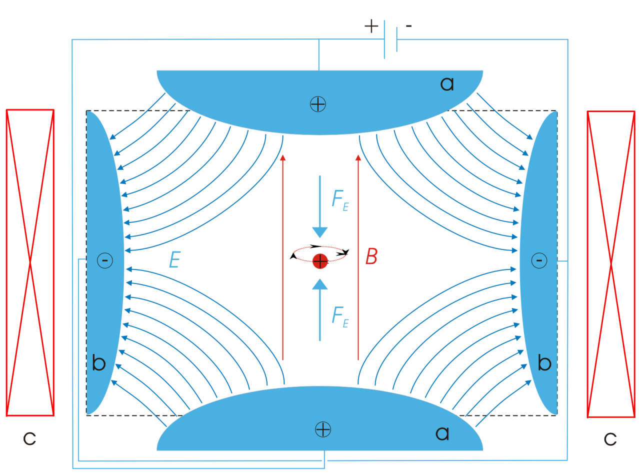

FIG. 1.

II. METHODS

The physical laws used to implement the Penning trap simulation will be from electrodynamics and classical mechanics, we will not take quantum aspects into account. The following equations allow us to express the time evolution of the particles motion:

$$\mathbf{E} = -\nabla V \tag{1}$$

$\mathbf{E}$ is the electric field and V the electric potential.

$$\mathbf{E}(\mathbf{r}) = k_e \sum_{j=1}^{n} q_j \frac{\mathbf{r} - \mathbf{r}_j}{|\mathbf{r} - \mathbf{r}_j|^3} \tag{2}$$

$\mathbf{E}(\mathbf{r})$ is the electric field at a point $\mathbf{r}$. This is set up by point charges $q_1, ..., q_n$ at points $\mathbf{r}_1, ..., \mathbf{r}_n$.

$$\mathbf{F} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B} \tag{3}$$

This is the **Lonretz force**, the force $\mathbf{F}$ on a particle with charge q, an electric field $\mathbf{E}$ and magnetic field $\mathbf{B}$.

$$m\ddot{\mathbf{r}} = \sum_i \mathbf{F}_i \tag{4}$$

This is Newton's second law. Here m is the mass of the particle and $\ddot{\mathbf{r}} \equiv \frac{d^2\mathbf{r}}{dt^2}$. Expressing that the sum of forces equals mass times acceleration.

The algorithm

The algorithm for the midpoint rule is summarized in algorithm 1. The basic idea behind the algorithm is to divide the integration range into to n small subintervals of length h, and on each such subinterval approximate the function $f(x)$ by a constant function. The value for this constant function is taken to be the value of $f(x)$ evaluated at the midpoint of the given subinterval — hence the name of the method.

As demonstrated in algorithm 1, it is conventional to present algorithms in a way that is independent of any specific programming language. This ensures that it is the logic behind the algorithm that remains in focus, rather

Algorithm 1 Midpoint rule for integration

procedure MIDPOINT RULE(f, a, b, n)
  $I \leftarrow 0$ ▷ Initialize the integral variable
  $h \leftarrow (b-a)/n$ ▷ Compute the interval length
  for $i = 1, 2, \ldots, n$ **do**
    $x \leftarrow a + (i - 1/2)h$ ▷ Assign x to the midpoint
    $I \leftarrow I + f(x)$ ▷ Add contribution to integral
  $I \leftarrow Ih$ ▷ Finalize the computation

than the syntax of a particular programming language. In algorithm 1 we have also demonstrated a common notation: The right-to-left arrow ($\leftarrow$) means that we assign the value of everything on the right to the variable on the left. This is nothing but how the "=" symbol functions in most programming languages, but the arrow notation makes it clear that we are in fact assigning a value, rather than stating that two things are equal.

III. RESULTS

To test the midpoint rule algorithm, we perform the integration of $f(x)$ using different choices for the number of subintervals. The results are listed in table I.[1]

TABLE I. Approximate values for the integral of $f(x) = x^3$ on the interval $[0, 1]$, as obtained with the midpoint rule with different numbers of integration subintervals.

Number of subintervals	Integral value
10^1	0.3086
10^2	0.2550
10^3	0.2505
10^4	0.2500

In figure 2 we show the relative error as a function of the number of subintervals n.

FIG. 2. The relative error versus the number of integration subintervals (n) when using the midpoint rule to estimate the integral $\int_0^1 x^3 \mathrm{d}x$. *Given that this plot is based only on a handful of data points, it should also show the data points using some form of point markers, like the dots here. When presented only with a continuous line, it is not necessarily clear for the reader if the result is based on a large or small number of data points.*

Note especially how we reference both the table and the figure with a short explanation of their content. Always do this! In the figure/table captions we can also add additional information, such as information about how the figure/table was produced. You can also do this in the main text if you like. When writing the figure/table captions, keep in mind the general rule of thumb that an expert on the topic should be able to understand the gist of your report simply by reading the abstract and look at the figures/tables and read their corresponding captions.

IV. DISCUSSION

Note that you are free to merge the presentation and discussion of the results into a single section of your report. This can in many cases lead to a more fluid presentation. If you do this, we recommend you use "Results and discussion" or similar for the section title.

From table I, we note that our implementation reproduces the analytical results to four digits precision when the integration range is divided into $n = 10^4$ subintervals. This indicates that that our implementation of the algorithm is correct.

From figure 2, we see that $\log_{10}(\epsilon)$ decreases linearly with $\log_2(n)$. From this, it should be possible to extract the convergence rate of our implementation of the midpoint rule. From a theoretical point of view we know that the midpoint rule should have a convergence rate of $\mathcal{O}(h^2)$. To properly verify our implementation, we should have estimated the convergence rate from our results and compared it to this theoretical rate. Without doing so, we cannot know that the our implementation of the algorithm is correct, even though we have seen that the numerical approximation converges to the correct answer in I.

Although this is a somewhat silly example, please note the following: We are to-the-point in our discussion of the results, and we only make strong claims about what we are actually certain about. In the discussion it is important to try to be as concise as possible — long paragraphs that only make very general points are typically of limited interest. Note that we also highlight aspects of our analysis that could have been improved and that might form a topic for future work.

V. CONCLUSION

In this section we state three things in a concise manner: what we have done, what we have found, and what should or could be done in the future.

We have investigated an implementation of the midpoint rule for numerical integration. As a first validation test we have checked that our implementation of the method reproduces the analytical result for the definite integral of $f(x) = x^3$ on $x \in [0, 1]$, achieving a four-digit precision when the integration range is divided into $n = 10^4$ subintervals. Furthermore, we have presented results for how the relative error of the method varies

[1] A general style recommendation is to avoid having vertical lines in tables. There are of course exceptions, but in most cases vertical lines will make a table less readable.

with the number of subintervals. To use these results to extract a precise estimate for the convergence rate of the method remains a topic for future work. As such, while our implementation of the midpoint rule has passed the initial validation tests, more work is needed to fully assess the validity of the implementation.