

EDUCATIONAL SOFTWARE SIMULATION OF CHARGED PARTICLE TRAJECTORIES IN PRESENCE OF MAGNETIC FIELDS

F. Calvino and C. Martinez, Eng. Nuclear ETSEIB-UPC, 08028-Barcelona, Spain

Abstract

A MS-Windows based program for learning about charged particle trajectories in presence of magnetic fields is presented. After a brief introduction, a program overview is given and the numerical resolution of the differential equations of motion is discussed. Following the program options are described. Finally a set of hardcopies showing some examples of particle trajectories, both in real and phase space, are presented.

1 INTRODUCTION

An example of a non-linear system are the equations of motion of a charged particle in a multipolar magnetic field. It is possible to find an approximate solution for the first order in the multipole expansion, but numerical methods are necessary to find a solution for higher order expansions.

The first order approximation can be solved by means of a matrix formalism or alternatively by transforming the equations of motion to look like an harmonic oscillator problem.

2 PROGRAM OVERVIEW

The program computes and graphically displays the trajectories (both in phase and real spaces) of charged particles when passing through a cell, i.e. a sequence of drift spaces and magnets (from dipoles up to octupoles, in each case upright or skew). This is done by solving the equations of motion by means of numerical methods. There is the possibility also of using transport matrix to obtain the different parameters of a cell (beta function, dispersion, emittance, ...).

The purpose of the program is mainly didactic. It gives the possibility of experimenting with the parameters on which the trajectory of the particle depends. Any change of these parameters will be reflected, after a short computing time, in the graphical display of the trajectories.

The user starts by defining the basic cell. At this point, the position, length and field strength of each element have to be provided. Then, the particle type and initial conditions need to be given. Using this

information the particle's trajectories are computed and the results displayed on the screen.

3 EQUATIONS OF MOTION

The differential equations of motion are derived using a cartesian co-ordinate system. A left handed reference frame is used, where the symbols x and y stay for the transverse axis and s for the longitudinal one.

In this co-ordinate system we can write the third order multipole field expansion (1) (including only upright elements) as [1,2]

$$\begin{aligned} B_x &= g(s)y + s(s)xy + \frac{1}{6}o(s)(3x^2y - y^3) \\ B_y &= B_{y0} + g(s)x + \frac{1}{2}s(s)(x^2 - y^2) + \frac{1}{6}o(s)(x^3 - 3xy^2) \quad (1) \\ B_s &= 0 \end{aligned}$$

Using (1) one immediately finds out

$$\begin{aligned} \ddot{x} &= s \left[G(s)x + \frac{1}{2}S(s)(x^2 - y^2) + \frac{1}{6}O(s)(y^3 - 3xy^2) \right] \\ \ddot{y} &= s \left[G(s)y + S(s)xy + \frac{1}{6}O(s)(3x^2y - y^3) \right] \quad (3) \\ \ddot{s} &= x \left[G(s)x + \frac{1}{2}S(s)(x^2 - y^2) + \frac{1}{6}O(s)(y^3 - 3xy^2) \right] \\ &\quad + y \left[G(s)y + S(s)xy + \frac{1}{6}O(s)(3x^2y - y^3) \right] \end{aligned}$$

where

$$G(s) = \frac{q}{m} g(s)$$

$$S(s) = \frac{q}{m} s(s)$$

$$O(s) = \frac{q}{m} o(s)$$

and m is the mass of the particle, $m = m_0(1 - \beta^2)^{-\frac{1}{2}}$

Equation (3) allows the computation of the particle trajectory at any time, provided that the initial position and velocity and the gradient are known.

From (3) can also be obtained, in linear approximation, two second order differential equations that by a proper transformation of variables can be express in the form of the equations for a harmonic oscillator.

$$\begin{aligned} x'' + \left(\frac{1}{\rho^2(s)} + k_0(s) \right) x &= \frac{1}{\rho} \frac{\Delta p}{p_0} \\ y'' - k_0(s) y &= 0 \end{aligned} \quad (4)$$

$$\text{where } s \equiv \frac{d^2}{ds^2}$$

A matrix formalism can be used to solve the linear approximation.

4 INTEGRATION METHOD

4.1 Description

The equations of motion are solved by means of the embedded fifth order Runge-Kutta method with adaptive stepsize [3,4].

The first step is to transform the second order differential equations of motion in a set of equivalent first order differential equations $\dot{y}_i = f_i(x_i)$. This is achieved using an appropriate change of variables, in this case. In this way a system of 6 first order differential equations is obtained

For a first order differential equation of the type $\frac{dy}{dx} = f(x, y)$ the general form of a fifth-order Runge-Kutta formula is

$$\begin{aligned} k_1 &= h f(x_n, y_n) \\ k_2 &= h f(x_n + a_2 h, y_n + b_{2,1} k_1) \\ &\dots \\ k_6 &= h f(x_n + a_6 h, y_n + b_{6,1} k_1 + \dots + b_{6,5} k_5) \\ y_{n+1} &= y_n + \sum_{i=1}^6 c_i k_i + O(h^6) \end{aligned} \quad (5)$$

Using a different set of parameters and embedded fourth-order formula can be obtained

$$y_{n+1}^* = y_n^* + \sum_{i=1}^6 c_i^* k_i + O(h^5) \quad (6)$$

and so the error estimate is

$$\Delta \equiv y_{n+1} - y_{n+1}^* = \sum_{i=1}^6 (c_i - c_i^*) k_i \quad (7)$$

The values of the various constants that are used ($a_i, b_{i,j}, c_i$) are those found by Cash and Karp.

4.2 Mechanism of control

Now that it is known, approximately, the error made in the calculation a relation should be found between h and Δ . Taking into account that the error scales as h^5 , an estimation can be made to obtain the step h_0 that would have given an error Δ_0 knowing that a step h_1 produced an error Δ_1

$$h_0 = h_1 \left| \frac{\Delta_0}{\Delta_1} \right|^{0.2} \quad (8)$$

If Δ_0 is the desired accuracy the equation tells in case the error is larger than the desired accuracy, how much to decrease (an repeat the failed step) or in case it is smaller how much to increase the next step.

5 PROGRAM DESCRIPTION

The program has been divided into a dynamic link library written in C, where all the numerical computations are carried on, and a graphical user interface written in Visual Basic.

The program is menu based, so that the user should not need to learn any special command.

In the first menu the user is prompted for the initial conditions of the particle. The sequence of magnets the particle will go through, the cell, has to be given too. The user can build cells and save them in a file disk for later use.

If the user has to build a cell, a set of windows will help him with the task.

Once the computation is finished a menu will be showed with different option for plotting the trajectories in real and phase space, either for the x or for the y plane.

6 RESULTS

We present here some results obtained from the program previously described.

As an example we have use the following cell definition:

DS1:	$L_{d,1} = 1.335 \text{ m}$	[drift section]
DS2:	$L_{d,2} = 45.535833632 \text{ m}$	[drift section]
Q1 :	$L_{q,f} = 3.05 \text{ m}$	[quadrupole]
	$G_{q,f} = 8.402938 \cdot 10^{-3} \text{ m}^{-2}$	
Q2 :	$L_{q,d} = 3.05 \text{ m}$	[quadrupole]
	$G_{q,d} = -1.05804 \cdot 10^{-2} \text{ m}^{-2}$	
S1 :	$L_{s,1} = 1.1 \text{ m}$	[sextupole]
	$G'_{s,1} = 9.1 \cdot 10^{-3} \text{ m}^{-3}$	

S2 : $L_{s,2} = 1.1 \text{ m}$ [sextupole]
 $G'_{s,2} = 2.1 \cdot 10^{-2} \text{ m}^{-3}$

The cell geometry is as follows:

Q1-DS1-S1-DS2-Q2-DS1-S2-DS2

In fig.1 we present the real space trajectory of a single proton, passing 30 times through the cell. The initial conditions were:

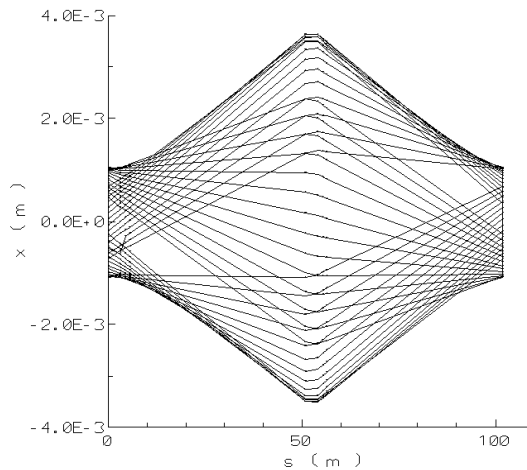
Energy : $E = 7.5 \text{ TeV}$
 Position: $x = 10^{-3} \text{ m}$
 $y = 10^{-3} \text{ m}$
 Angular dispersion: $\theta = 0^\circ$
 $\varphi = 0^\circ$

In fig.2 we have included the phase space trajectories for 6 protons at the end of the first quadrupoles. All the particles had the same energy and initial position.

Energy : $E = 7.5 \text{ TeV}$
 Position : $x = 10^{-3} \text{ m}$
 $y = 10^{-3} \text{ m}$

The different initial angles are included in the figure.

Fig. 1



7 CONCLUSIONS

We have presented a MS-Windows based software that allows the user to experiment with the parameters related with the particle's trajectories in an accelerator. The program has been developed to offer user-friendly menus and graphical display. Further development will be needed to offer the user an on-line help.

REFERENCES

- [1] H. Wiedemann. Particle Accelerators Physics, Basic Principles and Linear Beam Dynamics (Springer-Verlag, 1993).
- [2] J. Rossbach and P. Schmuser. Basic Course On Accelerator Optics (Proceedings of the 1992 CERN Accelerator School, CERN 94-01, 1994).
- [3] H. Gould and J. Tobochnik. Computer Simulation Methods (Clarendon Press, Oxford, 1987).
- [4] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Brian. Numerical Recipes in C (Cambridge Univ. Press, Cambridge 1992).

Fig. 2

