Society for Science and Education United Kingdom
TMLAI Transactions on Machine Learning and Artificial Intelligence
Volume 3, Issue 3
ISSN 2054-7390

Accuracy of the Java Simulation for the Charge Motion in Electric and Magnetic Fields

Masami Morooka[1] and Midori Morooka[2]
[1]Department of Electrical Engineering, Fukuoka Inst. Tech, higashi-ku, Fukuoka 811-0295, Japan;
[2]Flash Design Center, Micron Japan, Kamata 5-37-1, Ota-ku, Tokyo144-8721, Japan
morooka@fit.ac.jp

ABSTRACT

The accuracy of the Java simulation by the Runge-Kutta method for the charge motion in electric and magnetic fields has been investigated in comparison with the analytical solution. The error of the simulation depends on the time increment, h, used for the numerical calculation. If we use an increment that is larger than the boundary value, the simulation results in a non-accurate image of the charge motion. In this case, the simulation almost results in an underestimation, that is, a motion that is smaller than the real motion. The boundary increment is proportional to the mass of the charge, m, and is inversely proportional to the charge, q, and the magnetic field, B0. The empirical results conclude that the image of the charge motion can be obtained accurately by Java simulation using $h < 0.2m/qB0$.

Keywords: Image learning, charge motion in electric and magnetic fields, Java programming, accuracy of Java simulation by Runge-Kutta method.

1 Introduction

The authors proposed a Java simulation for the rapid and accurate image learning of the charge motion in electric and magnetic fields [1] and for those of the electric characteristics of RCL circuits [2]. In these simulations, the text fields of the selected parameters, such as the electric field and magnetic field for the charge motion or the values of R, C, and L for the electric characteristics, are set on the display, and the calculation by the Runge-Kutta method is initiated by clicking the start button after inputting values into the text fields. Immediately following the completion of the calculation, the results are plotted as a figure on the display, e.g. a charge locus for the charge motion or the change of the current and voltage with time for the electric circuit. By changing the values in the text fields, new results can be represented immediately and a simulation under the new condition can be easily obtained. The value of the time increment, h, used in the numerical calculation by the Runge-Kutta method is limited to obtain an accurate simulation in spite of the useful simulation.

In this paper, the accuracy of the Java simulation for the charge motion in electric and magnetic fields has been investigated in comparison with the analytical solution, and the boundary value of the time increment to obtain an accurate simulation is shown empirically.

DOI: 10.14738/tmlai.33.1180
Publication Date: 04th June, 2015
URL: http://dx.doi.org/10.14738/tmlai.33.1180

2 Numerical Method and Analytical Solutions for the Charge Motion

2.1 Equations for the Charge Motion in Electric and Magnetic Fields

The charge motion in an applied electric field E = (Ex, Ey, Ez) and an applied magnetic field B = (Bx, By, Bz) is given as

$$m\frac{d\boldsymbol{v}}{dt} = -a\boldsymbol{v} + q\boldsymbol{E} + q\boldsymbol{v} \times \boldsymbol{B} - b(\boldsymbol{r} - \boldsymbol{r}_0), \tag{1}$$

$$\frac{d\boldsymbol{r}}{dt} = \boldsymbol{v}. \tag{2}$$

Here, *t* is time and *m*, q, $\boldsymbol{v} = (v_x, v_y, v_z)$, and $\boldsymbol{r} = (x, y, z)$ are the mass, charge, velocity, and displacement of the charge, respectively. *a* and *b* are coefficients of the resistance and restoring forces. $\boldsymbol{r}_0$ is the restoring centre. We consider only the electric and magnetic forces, such as the electron motion in a vacuum, to facilitate an easy comparison between the numerical simulation and the analytical solution. We use $\boldsymbol{E} = (E_x, 0, 0)$, $\boldsymbol{B} = (0, 0, B_z)$, $E_x = E_0\sin(2\pi ft)$, and $B_z = B_0$ to easily obtain the analytical solution. Here, E_0 and B_0 are constants, and *f* is the frequency of the electric field. In this case, we have four ordinary differential equations from Equations (1) and (2) for the charge motion in the *x* – *y* plane.

$$m\frac{dv_x}{dt} = qE_0\sin(2\pi ft) + qv_yB_0, \tag{3}$$

$$m\frac{dv_y}{dt} = -qv_xB_0, \tag{4}$$

$$\frac{dx}{dt} = v_x, \tag{5}$$

$$\frac{dy}{dt} = v_y, \tag{6}$$

2.2 Numerical Method and Analytical Solutions

The numerical method for obtaining the solutions of Equations (3) – (6) by Java programming is described in Reference [1] using the fourth-order Runge-Kutta method.

We have the linear differential equation from Equations (3) and (4), as shown below.

$$\frac{d^2v_x}{dt} = 2\pi fcE_0\cos(2\pi ft) - c^2B_0^2v_x. \tag{7}$$

Here, $c = q/m$. The general solution of Eq. (7) is obtained as the sum of the general solution of the homogeneous equation, $d^2v_x/dt^2 + c^2B_0^2v_x = 0$, and the particular solution.

$$v_x = d_1\exp(jcB_0t) + d_2\exp(-jcB_0t) + \frac{2\pi fcE_0}{(cB_0)^2 - (2\pi f)^2}\cos(2\pi ft). \tag{8}$$

Here, d_1 and d_2 are unfixed constants and the last term is the particular solution. We use the initial conditions $v_x = v_0$ and $v_y = 0$ at $t = 0$, that is, the charge is injected into the x-direction of the fields at the initial velocity v_0. We obtain the other initial condition, $dv_x/dt = 0$ at $t = 0$, from Equation (3) using $v_y = 0$ at $t = 0$. We have the analytical solution, Equation (8), under these initial conditions,

$$v_x = \left[v_0 - \frac{2\pi fcE_0}{(cB_0)^2 - (2\pi f)^2} \right] \cos(cB_0 t) + \frac{2\pi fcE_0}{(cB_0)^2 - (2\pi f)^2} \cos(2\pi ft) \tag{9}$$

We obtain the analytical solution for v_y from Equation (3),

$$v_y = \left[\frac{2\pi fcE_0}{(cB_0)^2 - (2\pi f)^2} - v_0 \right] \sin(cB_0 t) - \frac{cB_0 cE_0}{(cB_0)^2 - (2\pi f)^2} \sin(2\pi ft \tag{10}$$

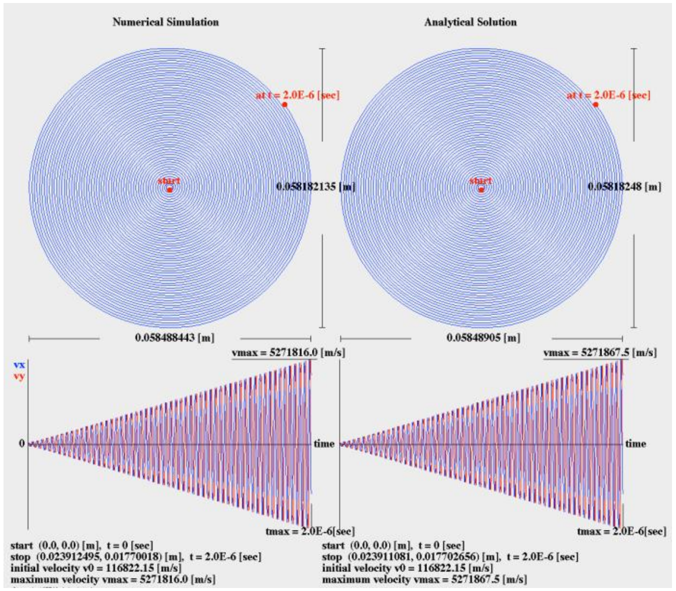
The analytical solutions for x and y under the initial conditions $(x,y) = (0,0)$ at $t = 0$ are obtained from Equations (5) and (6),

$$x = \frac{1}{cB_0} \left[v_0 - \frac{2\pi fcE_0}{(cB_0)^2 - (2\pi f)^2} \right] \sin(cB_0 t) + \frac{cE_0}{(cB_0)^2 - (2\pi f)^2} \sin(2\pi ft), \tag{11}$$

$$\begin{aligned} y = {} & \frac{1}{cB_0} \left[v_0 - \frac{2\pi fcE_0}{(cB_0)^2 - (2\pi f)^2} \right] \cos(cB_0 t) + \frac{1}{2\pi f} \frac{cB_0 cE_0}{(cB_0)^2 - (2\pi f)^2} \cos(2\pi ft) \\ & - \frac{1}{cB_0} \left[v_0 - \frac{2\pi fcE_0}{(cB_0)^2 - (2\pi f)^2} \right] - \frac{1}{2\pi f} \frac{cB_0 cE_0}{(cB_0)^2 - (2\pi f)^2} . \end{aligned} \tag{12}$$

3 Comparison of the Numerical Simulations with the Analytical Solutions

Typical numerical simulations and analytical solutions for an electron motion accelerated by the electric frequency that is synchronized with the Larmor frequency using E_0 = 30 V/m, f = 28.55 MHz, and B_0 = 0.00102 T are shown in Figure 1. The simulations and solutions are shown in the left and right regions, respectively. The loci of the electron are shown in the upper regions and the changes of v_x and v_y with time are shown in the lower regions. The numerical calculation is performed using the time increment h = 0.5 ns by the double precision method. The final position of the calculation at t = 2000 ns, corresponding to 4000 total calculations, is (x,y) = (0.023912495, 0.01770018) in meters for the numerical simulation and (x,y) = (0.023911081, 0.017702656) for the analytical solution. The differences between the two results are $(\Delta x, \Delta y) = (-1.414 \times 10^{-6}, 2.476 \times 10^{-6})$, and their rates are 5.91×10^{-5} for x and 1.40×10^{-4} for y. The differences increase with the increase of the calculation time, such as (x,y)=(-0.6840279, -0.23154235) and (x,y) = (-0.6834052, -0.23318635) after 100,000 calculations in which $(\Delta x, \Delta y)$ = (0.0006227, -0.001644) and their rates are 9.11×10^{-4} for x and 7.05×10^{-3} for y. The global error of this method is $O(h^4)$ [3]. The position at t = 2000 ns by the simulation using h = 0.25 ns is (x,y) = (0.023911174, 0.017702503), and their differences from the analytical results are $(\Delta x, \Delta y) = (-9.3 \times 10^{-8}, 1.53 \times 10^{-7})$, which are approximately 1/16 of the values from the simulation obtained with h = 0.5 ns.

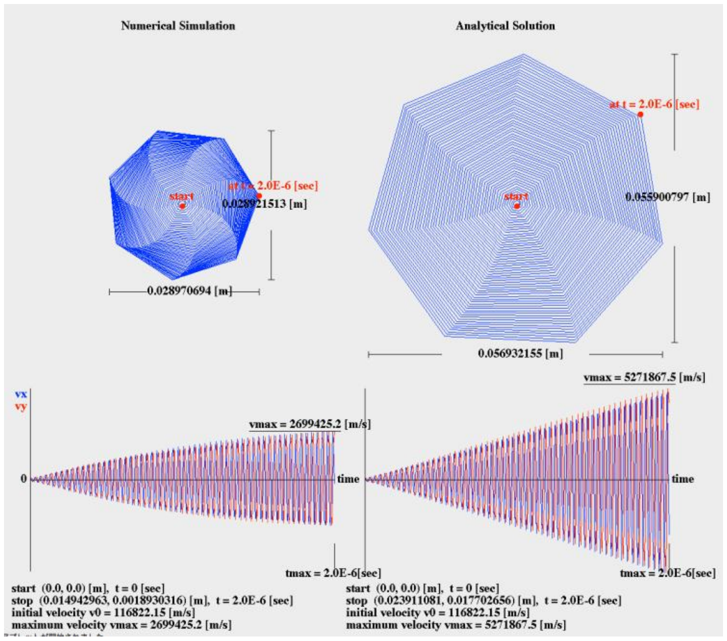


```
start  (0.0, 0.0) [m],  t = 0 [sec]
stop  (0.023912495, 0.01770018) [m],  t = 2.0E-6 [sec]
initial velocity v0 = 116822.15 [m/s]
maximum velocity vmax = 5271816.0 [m/s]
```

```
start  (0.0, 0.0) [m],  t = 0 [sec]
stop  (0.023911081, 0.017702656) [m],  t = 2.0E-6 [sec]
initial velocity v0 = 116822.15 [m/s]
maximum velocity vmax = 5271867.5 [m/s]
```

Figure 1: Comparison of a typical simulation with the analytical solution for an electron motion after 4000 calculations, that is, t = 2000 ns, using h = 0.5 ns at E0 = 30 V/m, f = 28.55 MHz, and B0 = 0.00102 T

The error of the Runge-Kutta simulation increases with the increase of h, and if we use a large h, the Java simulation results in a non-accurate image of the electron motion in comparison with that of the analytical solution, as shown in Figure 2. In this simulation, the range of the electron motion and the velocity at t = 2000 ns are approximately half that of the accurate analytical values.



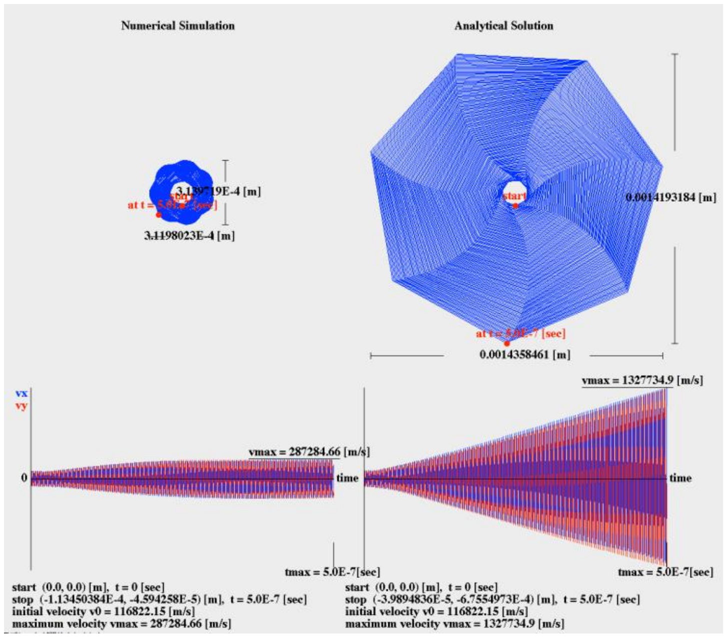
```
start  (0.0, 0.0) [m],  t = 0 [sec]
stop  (0.014942963, 0.0018930316) [m],  t = 2.0E-6 [sec]
initial velocity v0 = 116822.15 [m/s]
maximum velocity vmax = 2699425.2 [m/s]
```

```
start  (0.0, 0.0) [m],  t = 0 [sec]
stop  (0.023911081, 0.017702656) [m],  t = 2.0E-6 [sec]
initial velocity v0 = 116822.15 [m/s]
maximum velocity vmax = 5271867.5 [m/s]
```

Figure 1: A non-accurate Java simulation for the electron motion using h = 5 ns. The values except h are the same as those used for the calculations in Figure1. The calculations are performed 400 times, corresponding to the same amount of time, 2000 ns, as in Figure 1.

The accuracy of the Runge-Kutta simulation depends on hqB_0/m from Equation (3), and the simulation at B_0 = 0.0102 T, which is ten times larger than that used in Figure 1, is not accurate, even with the use of

h = 0.5 ns, similar to the result from using h = 5 ns at B_0 = 0.00102 T, as shown in Figure 3. We use f = 285.5 MHz for the calculation in Figure 3 to synchronize with the Larmor frequency.

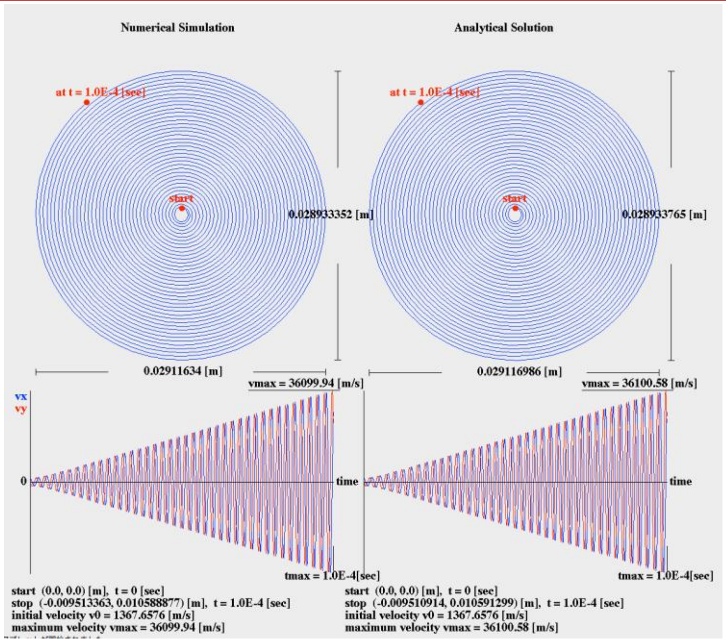


```
start  (0.0, 0.0) [m],  t = 0 [sec]
stop  (-1.13450384E-4, -4.594258E-5) [m],  t = 5.0E-7 [sec]
initial velocity v0 = 116822.15 [m/s]
maximum velocity vmax = 287284.66 [m/s]
```

```
start  (0.0, 0.0) [m],  t = 0 [sec]
stop  (-3.9894836E-5, -6.7554973E-4) [m],  t = 5.0E-7 [sec]
initial velocity v0 = 116822.15 [m/s]
maximum velocity vmax = 1327734.9 [m/s]
```

Figure 2: A non-accurate Java simulation for the electron motion even with the use of h = 0.5 ns due to the use of the ten times larger B0, 0.0102 T, relative to that in Figure 1. f = 285.5 MHz is used to synchronize with the Larmor frequency in this calculation, and the calculations are performed 1000 times until t = 500 ns.

In the case of an ion motion, the value of h used to obtain an accurate image of the motion is not that small, which is in contrast to the electron motion because the accuracy of the Runge-Kutta simulation is proportional to h/m from Equation (3). The Java simulations and analytical solutions for an He^+ motion accelerated by the synchronized electric frequency to the Larmor frequency are shown in Figure 4 using h = 50 ns, E_0 = 30 V/m, f = 391300 Hz, and B_0 = 0.102 T. An accurate image for the ion motion can be obtained by the Java simulation even with the use of h = 50 ns.



```
start  (0.0, 0.0) [m],  t = 0 [sec]
stop  (-0.009513363, 0.010588877) [m],  t = 1.0E-4 [sec]
initial velocity v0 = 1367.6576 [m/s]
maximum velocity vmax = 36099.94 [m/s]
```

```
start  (0.0, 0.0) [m],  t = 0 [sec]
stop  (-0.009510914, 0.010591299) [m],  t = 1.0E-4 [sec]
initial velocity v0 = 1367.6576 [m/s]
maximum velocity vmax = 36100.58 [m/s]
```

Figure 3: A typical He+ motion accelerated by the electric frequency synchronized with the Larmor frequency using h = 50 ns at E0 = 30 V/m, f = 391300 Hz, and B0 = 0.102 T. The calculations are performed 2000 times until t = 0.0001 sec.

4 Discussion

The accuracy of the numerical calculation used for the Java simulation depends on hqB_0/m from Equation (3). If we use an h that is too large, the calculation is not performed accurately and an appropriate image of the motion is not obtained, as shown in Figures 2 and 3. The accuracy of the simulation is better for evaluating the difference of the motion width, which is $\Delta x_{max} = x_{max} - x_{min}$ or $\Delta y_{max} = y_{max} - y_{min}$, compared to that of the analytical solution. Here, the subscripts $_{max}$ and $_{min}$ are used to represent the maximum and minimum values, respectively. The values of Δx_{max} at t = 500, 1000, 2000, 5000, and 10000 ns obtained by the simulations and the solutions for the electron motion using h = 0.5, 1.0, 2.0, 3.0, 4.0, and 5.0 ns at E_0 = 30 V/m, f = 28.55 MHz, and B_0 = 0.00102 T are shown in Table 1. The ratio of Δx_{max} obtained by the simulation to that obtained by the solution and the value of hqB_0/m corresponding to each h are also shown in the table. The ratio represents the accuracy of the Java simulation, that is, accuracy = 1 means that the Java simulation is equal to the analytical solution, and an accuracy > 1 and < 1 indicate the overestimation and underestimation of the simulations, respectively. The dependence of the accuracy for the electron motion on the value of hqB_0/m is shown in Figure 5.

Table 1: The motion width, xmax – xmin, obtained by the Java simulation and the analytical solution for the electron motion at t = 500, 1000, 2000, 5000, and 10000 ns using h = 0.5, 1.0, 2.0, 3.0, 4.0, and 5.0 ns at E0 = 30 V/m, f = 28.55 MHz, and B0 = 0.00102 T

Time (ns)	Methods	*h* = 0.5 ns hqB_0/m = 0.0897	*h* = 1.0 ns hqB_0/m = 0.1794	*h* = 2.0 ns hqB_0/m = 0.3588	*h* = 3.0 ns hqB_0/m = 0.5381	*h* = 4.0 ns hqB_0/m = 0.7175	*h* = 5.0 ns hqB_0/m = 0.8969
500	Java Simulation	0.014223397	0.014209879	0.014060617	0.01385032	0.012969641	0.011567365
	Analytical Solution	0.014223453	0.014211405	0.014097456	0.014027963	0.013868473	0.013841441
	Ratio (Accuracy)	0.99999607	0.9998926	0.9973869	0.9873365	0.9351888	0.8357053
1000	Java Simulation	0.029137922	0.029083265	0.028724693	0.028237505	0.025068847	0.01968627
	Analytical Solution	0.029138096	0.0290887	0.028879715	0.029069254	0.028370567	0.028370567
	Ratio (Accuracy)	0.99999404	0.99981314	0.9946321	0.9713873	0.8836217	0.69389766
2000	Java Simulation	0.058488443	0.058380943	0.05749023	0.054800544	0.045379903	0.028970694
	Analytical Solution	0.05848905	0.05839335	0.057914756	0.057436377	0.057914756	0.056932155
	Ratio (Accuracy)	0.9999896	0.9997875	0.9926698	0.95410866	0.78356373	0.50886345
5000	Java Simulation	0.14653207	0.14628382	0.14346443	0.12560898	0.079387136	0.03248504
	Analytical Solution	0.14653541	0.14635521	0.14584598	0.14584598	0.14476995	0.14280663
	Ratio (Accuracy)	0.99997723	0.9995122	0.9836708	0.861244	0.5483675	0.22747572
10000	Java Simulation	0.2936057	0.29312417	0.282346	0.21564397	0.086871445	0.03248504
	Analytical Solution	0.29361787	0.2933971	0.2933971	0.29165736	0.29239708	0.282726
	Ratio (Accuracy)	0.9999585	0.99906975	0.96233404	0.73937434	0.29710093	0.11489938

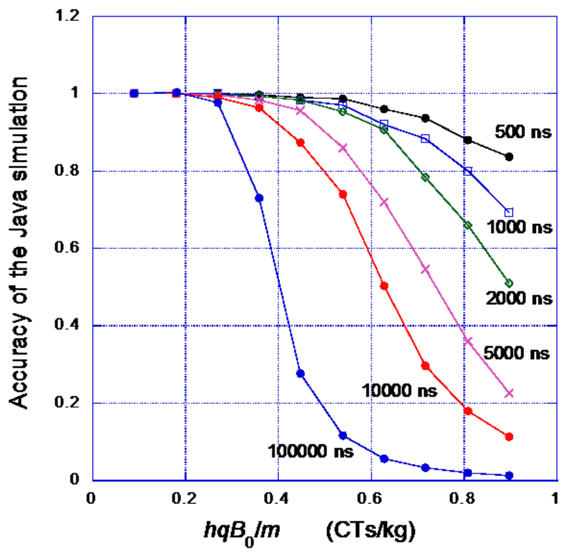


Figure 4: Dependence of the accuracy of the Java simulation on the hqB0/m used for the numerical calculation. Accuracy = 1 means that the Java simulation is equal to the analytical solution. Accuracy > 1 and < 1 indicate the overestimation and underestimation of the simulations, respectively.

The values of Δx_{max} at t = 20000, 50000, 100000, 200000, and 500000 ns obtained by the simulations and the solutions for the He$^+$ motion using h = 25, 50, 100, 200, 300, and 400 ns at E_0 = 30 V/m, f = 391300 Hz, and B_0 = 0.102 T are shown in Table 2. The ratio of Δx_{max} and the value of hqB_0/m corresponding to each h are also shown in the table. The dependence of the accuracy for the Java simulation of the He$^+$ motion on the value of hqB_0/m is shown in Figure 6.

Table 2: The motion width, xmax – xmin, obtained by the Java simulation and the analytical solution for the He+ motion at t = 20000, 50000, 100000, 200000, and 500000 ns using h =25, 50, 100, 200, 300, and 400 ns at E0 = 30 V/m, f = 391300 Hz, and B0 = 0.102 T

Time (ns)	Methods	*h* = 25 ns hqB_0/m = 0.0615	*h* = 50 ns hqB_0/m = 0.1229	*h* = 100 ns hqB_0/m = 0.2459	*h* = 200 ns hqB_0/m = 0.4917	*h* = 300 ns hqB_0/m = 0.7376	*h* = 400 ns hqB_0/m = 0.9834
20000	Java Simulation	0.0055589615	0.0055585667	0.0055583343	0.005472003	0.005226011	0.0047790827
	Analytical Solution	0.0055589606	0.0055585555	0.0055585555	0.005496019	0.005364648	0.0051418506
	Ratio (Accuracy)	1.0000001	1.000002	0.9999602	0.9956304	0.9741573	0.929448
50000	Java Simulation	0.014506064	0.014495008	0.014490361	0.014259509	0.013312547	0.010083964
	Analytical Solution	0.014506063	0.014495134	0.014495134	0.014377031	0.014377031	0.013768999
	Ratio (Accuracy)	1.0000001	0.99999124	0.99967074	0.99182564	0.9259594	0.7323673
100000	Java Simulation	0.029134799	0.02911634	0.028972711	0.02827198	0.024131613	0.014651205
	Analytical Solution	0.029134804	0.029116986	0.028993592	0.028993592	0.028993592	0.028993592
	Ratio (Accuracy)	0.9999998	0.9999778	0.99927986	0.97511137	0.83230853	0.5053256
200000	Java Simulation	0.05841807	0.05838068	0.058114525	0.055021226	0.04045283	0.01634748
	Analytical Solution	0.0584181	0.05838337	0.058229085	0.05778163	0.05778163	0.05778163
	Ratio (Accuracy)	0.99999946	0.99995387	0.99803257	0.952227	0.70009845	0.28291827
500000	Java Simulation	0.14663371	0.1465667	0.14601156	0.12891208	0.056520145	0.01634748
	Analytical Solution	0.14663388	0.14658329	0.14658329	0.14594747	0.14540245	0.1458892
	Ratio (Accuracy)	0.99999887	0.9998869	0.99609965	0.88327724	0.38871524	0.11205408

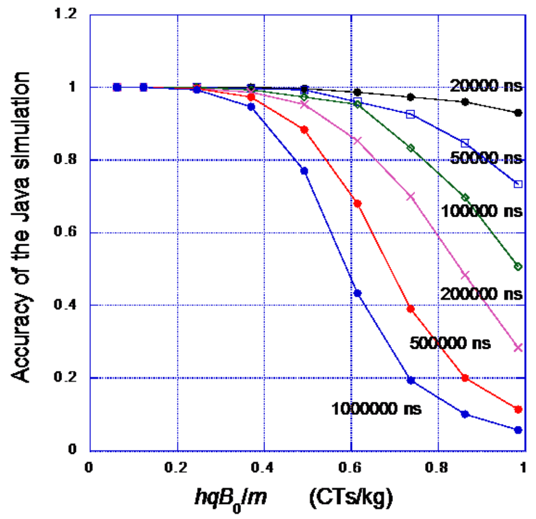


Figure 5: Dependence of the accuracy of the Java simulation for He+ motion on hqB0/m. Accuracy = 1 means that the Java simulation is equal to the analytical solution. Accuracy >1 and <1 indicate the over estimation and under estimation of the simulations, respectively.

5 Conclusion

The accuracy of the Java simulation by the Runge-Kutta method for the charge motion in electric and magnetic fields has been investigated in comparison with the analytical solution. The results are summarized as follows:

1. The accuracy depends on the value of hqB_0/m used for the numerical calculation, which also depends on the numerical calculation time.
2. In the case of non-accurate simulation, the simulation almost results in an underestimation, that is, a motion that is smaller than the real motion.
3. An accurate image for the charge motion in electric and magnetic fields is able to be obtained by the Java simulation using a less value of hqB_0/m than 0.2 sCT/kg, that is, using $h < 0.2m/qB_0$, as shown in Figures 5 and 6.

REFERENCES

[1]. M. Morooka, S. Qian, and M. Morooka, *Image Learnig of Charge Motion in Electric and Magnetic Fields by Java Programming*, Transactions on Machine Learnig and Artificial Intelligence, 2014. **2**(2): p.1- 19.

[2]. M. Morooka, S. Qian, and M. Morooka, *Image Learnig of Electric characteristics of Resistance, Capacitance, Inductance, and their Circuits by Java Programming*, Transactions on Machine Learnig and Artificial Intelligence, 2014. **2**(3): p.1- 19.

[3]. A. Ralston and C. L. Meek, *ENCYCROPEDIA OF COMPUTER SCIENCE*, Van Nostrand Rainhold Company (1976), p. 984.