Chapter 4

METHODOLOGY

4.1 ANALYTICAL AND NUMERICAL

There are two approaches in dealing with the modeling of lithium-ion concentration in separator and cathode for lithium-ion battery. The past studies were done by (Landfors et al. 1995; Jain et al. 1999; Botte and White 2001; Subramanian and White 2001; Hashim Ali et al. 2002; Hashim Ali et al. 2003; Hashim Ali et al. 2004; Johan and Arof 2007; Smith et al. 2007; Subramanian et al. 2007; Golmon et al. 2009; Subramanian et al. 2009; Norzihani et al. 2010). These two approaches are analytical and numerical. Farlow (1982) stated that analytic solutions are those solutions where the unknown variable u is given as a mathematical expression in terms of the independent variables and parameters of the system which are generally infinite series or integrals. Separation of variables method (Subramanian and White 2001), Green function method (Johan and Arof 2007) and Laplace Transform method (Hashim Ali et al. 2002; Hashim Ali et al. 2003; Hashim Ali et al. 2004) are several methods of analytical approach that have been used by several researches and batteries developers.

Generally, analytic solution is the best way to describe or to illustrate the result of the system but many systems cannot be solved directly using analytic solution because most of the systems are nonlinear in nature. So practically all nonlinear systems must be solved by numerical methods and in fact most realistic models in physics, chemistry, biology and so forth are nonlinear in nature. According Botte et al. (2000), most complicated models must be solved using numerical approach. Numerical approach is very general, and valid for most models.

Doyle and Newman (1997) stated that analytical solution to solve the mathematical model of lithium-ion batteries is available for very few limiting cases. According to him, we must simplify the models under certain operating conditions to present a few limiting cases and analyzed the electrochemical behavior of lithium-ion batteries. Analytical approximation often provide extremely useful information concerning the character of the solution for critical values of the dependant variables but tend to be more difficult to apply and sometimes cannot be used to solved certain type of partial differential equation exactly. According to Doyle et al. (1993) and Subramanian et al. (2004), the performance of batteries can be predicted accurately using electrochemical models but these models are usually complex because of the nonlinear coupling of the dependent variables in the governing equations.

The salt concentration is being uniformed, for example, with a system having a unity transference number for the lithium-ion or at very short times, much less than the diffusion time. If this is the case, the governing equations are much simpler, and several possibilities exist for the examination of approximate analytic solutions. When concentration gradients cannot be neglected, the situation is much more complex due to the coupled nature of the governing equations. For this reason very few analytic solutions can be found in the literature that include concentration variations in the solution phase, and hence this problem is generally relegated to numerical methods (Doyle and Newman 1997). They also added that the numerical solution is needed to solve the complex model of partial differential equations. This complexity of the model arise due to the nonlinear coupling between the electrolyte potential and electrolyte concentration, nonlinear Butler-Volmer kinetics, nonlinear dependence of exchange current on the electrolyte or solid-state concentration, nonlinear dependence of open circuit potential on the solid-state concentration, dependence of electrolyte conductivity or electrolyte diffusion coefficient on the electrolyte concentration, dependence of solid-state diffusion coefficient on the solid-state concentration and dependence of transfer number on the electrolyte concentration. Typically, combination of more than one of the above reasons complicates the lithium-ions batteries.

With the development of fast and efficiently digital computer, the role of numerical method in solving complicated mathematical equations has increased dramatically in recent years. Numerical solutions are referred to finding the solution of the system by replacing the differential equation with an approximate equation and solving the easier one. The result is generally a table of numbers listing the solution for various values of the independent variables (Farlow 1982). According to Botte et al. (2000) and Ceder et al. (2002), the most common numerical method that has been used to solve this set of model equation are the Finite Difference Method (FDM), the Control Volume Formulation (CVF), and the Finite Element Method (FEM). (Landfors et al. 1995; Jain et al. 1999; Botte and White 2001; Smith et al. 2007; Subramanian et al. 2007; Golmon et al. 2009; Subramanian et al. 2009; Norzihani et al. 2010) are some researchers and battery developers that used numerical method in their work. The FDM has been used extensively due to its simplicity and accuracy (Botte et al. 2000).

Botte et al. (2000) has shown that the CVF performs better than the FDM only for boundary value problem (BVP) in which the two boundary conditions are given as fluxes. In all other cases, the FDM is more accurate than the CVF for a small number of nodes especially when interface boundary conditions are presented in the system. They have also shown that the false boundary method (FBM) can be used to obtain mass conservation in boundary value problems in which the two boundary conditions are given as fluxes. However, when only one of the boundary conditions is given as a flux, the FDM is more accurate for one flux boundary condition than the FBM for a small number of nodes. The finite element method (FEM) provides an alternative to model systems with irregular geometry, unusual boundary conditions, or heterogeneous composition. FDM is also easier to program than the FEM, CVF or FBM approach.

Preisig and White (1990) also state that the FDM is a widely used method. FDM does not require any particular mathematical structure of the model equation because a mathematical model that consist a set of governing equations with a set of initial and boundary conditions is readily formulated. Hence, the FDM only need the derivatives in the equations to be replaced by corresponding to Finite difference approximation. The overall procedure is straightforward and easy to implement. Based on the reason above, this study will use the FDM.

4.2 FINITE DIFFERENCE METHOD (FDM)

There are many studies (Noye 1981; Farlow 1982; Banks 2001; Sadiku 2001; Holmes 2007) done in relation to this method in other research areas. In this study, Finite Difference Method (FDM) has been selected to solve this diffusion equation in separator ($u_t = u_{xx}$) and cathode ($u_t = u_{xx} + J$).

The important point in Finite Difference Method is the derivation, which starts with the Taylor series expansion as follows

$$f(y+h) = f(y) + f^{'}(y)h + \frac{f^{''}(y)}{2!}h^2 + \ldots + \frac{f^{(n-1)}(y)}{(n-1)!}h^{(n-1)} + \frac{f^{(n)}(y)}{n!}h^{(n)} \tag{4.1}$$

From Taylor series (4.1) we deduced forward difference approximation for $\vartheta_\tau(y,\tau)$ as follows

$$\vartheta_\tau(y,\tau) \cong \frac{\vartheta(y,\tau+k) - \vartheta(y,\tau)}{k} \tag{4.2}$$

backward difference approximation for $\vartheta_\tau(y,\tau)$ is as follows

$$\vartheta_\tau(y,\tau) \cong \frac{\vartheta(y,\tau) - \vartheta(y,\tau-k)}{k} \tag{4.3}$$

central difference approximation for $\vartheta_\tau(y,\tau)$ is as follows

$$\vartheta_\tau(y,\tau) \cong \frac{\vartheta(y,\tau+k) - \vartheta(y,\tau-k)}{2k} \tag{4.4}$$

We also extended this analysis to get central difference approximation for $\vartheta_{yy}(y,\tau)$ as follows

$$\vartheta_{yy}(y,\tau) \cong \frac{1}{h^2}\left[\vartheta(y+h,\tau) - 2\vartheta(y,\tau) + \vartheta(y-h,\tau)\right] \tag{4.5}$$

Generally for time and space problem are as follows

$h = \Delta y$ where $\Delta y =$ changes in direction

$k = \Delta\tau$ where $\Delta\tau =$ changes in time

In order to use a computer code in solving this problem, the following notation are used as follows

$$\vartheta(y,\tau) = \vartheta_j^i \tag{4.6}$$

$$\vartheta(y,\tau+k) = \vartheta_j^{i+1} \tag{4.7}$$

$$\vartheta(y,\tau-k) = \vartheta_j^{i-1} \tag{4.8}$$

$$\vartheta(y+h,\tau) = \vartheta_{j+1}^i \tag{4.9}$$

$$\vartheta(y-h,\tau) = \vartheta_{j-1}^i \tag{4.10}$$

Then substituted notations (4.6) to (4.10) into equation (4.2) to (4.5) to get forward difference approximation for $\vartheta_\tau(y,\tau)$ as follows

$$\vartheta_\tau(y,\tau) = \frac{1}{k}\left(\vartheta_j^{i+1} - \vartheta_j^i\right) \tag{4.11}$$

backward difference approximation for $\vartheta_\tau(y,\tau)$ is as follows

$$\vartheta_\tau(y,\tau) = \frac{1}{k}\left(\vartheta_j^i - \vartheta_j^{i-1}\right) \tag{4.12}$$

central difference approximation for $\vartheta_\tau(y,\tau)$ is as follows

$$\vartheta_\tau(y,\tau) = \frac{1}{2k}\left(\vartheta_j^{i+1} - \vartheta_j^{i-1}\right) \tag{4.13}$$

and central difference approximation for $u_{yy}(y,\tau)$ is as follows

$$\vartheta_{yy}(y,\tau)=\frac{1}{h^2}\left(\vartheta_{j+1}^{i}-2\vartheta_{j}^{i}+\vartheta_{j-1}^{i}\right) \tag{4.14}$$

Note that: $\vartheta=\vartheta(y,\tau)$; $\frac{\partial\vartheta}{\partial\tau}=\vartheta_{\tau}(y,\tau)$; $\frac{\partial^2\vartheta}{\partial y^2}=\vartheta_{yy}(y,\tau)$

By now the i-terms and j-terms have emerged. These two terms can be related to the grid line illustrated in figure 4.1.

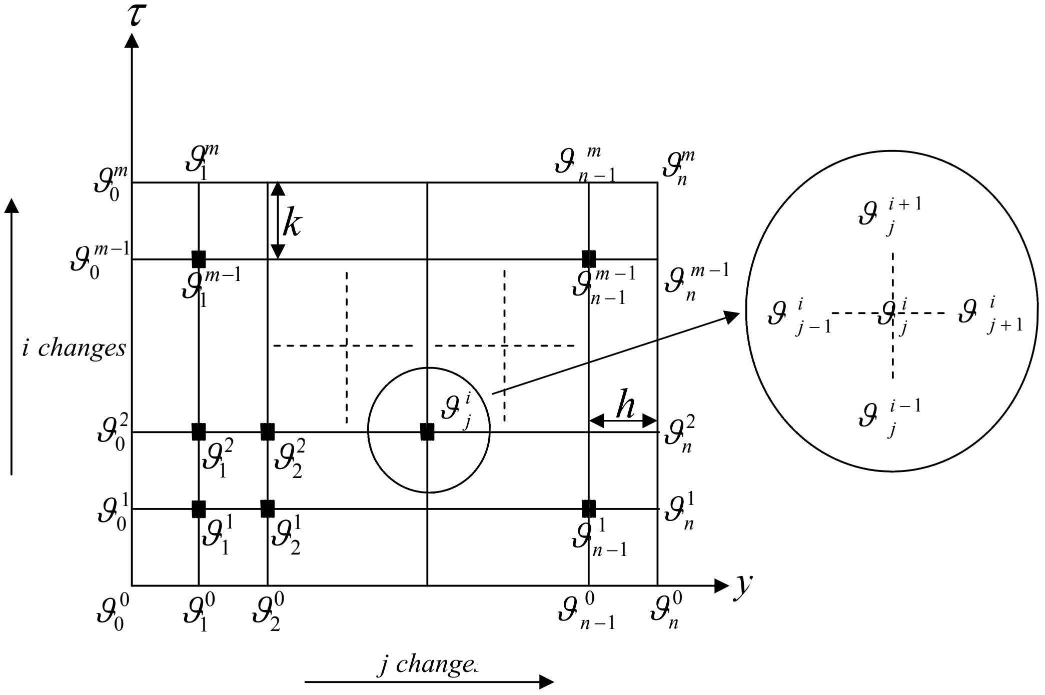


Figure 4.1 : Grid Lines commonly used in Finite Difference Method (FDM).

By substitution of equation (4.11) and (4.14) into equation (3.73) and (3.74) respectively, the diffusion equation in separator became as follows

$$\frac{\vartheta_j^{i+1}-\vartheta_j^i}{k}=\frac{\vartheta_{j+1}^i-2\vartheta_{j+1}^i+\vartheta_{j-1}^i}{h^2} \tag{4.15}$$

or

$$\boxed{\vartheta_j^{i+1}=\frac{k}{h^2}\vartheta_{j+1}^i+\left(1-2\frac{k}{h^2}\right)\vartheta_j^i+\frac{k}{h^2}\vartheta_{j-1}^i} \tag{4.16}$$

and the diffusion equation in cathode became as follows

$$\frac{\vartheta_j^{i+1}-\vartheta_j^i}{k}=\varepsilon^{\frac{1}{2}}\left(\frac{\vartheta_{j+1}^i-2\vartheta_{j+1}^i+\vartheta_{j-1}^i}{h^2}\right)+J \tag{4.17}$$

or

$$\boxed{\vartheta_j^{i+1}=\varepsilon^{\frac{1}{2}}\frac{k}{h^2}\vartheta_{j+1}^i+\left(1-2\varepsilon^{\frac{1}{2}}\frac{k}{h^2}\right)\vartheta_j^i+\varepsilon^{\frac{1}{2}}\frac{k}{h^2}\vartheta_{j-1}^i+kJ} \tag{4.18}$$

Generally Equation (4.16) and (4.18) are known as explicit scheme under FDM. Now to get implicit scheme, the derivation begin with substitution of equation (4.12) and (4.14) into the diffusion equation in separator (3.73) to get the equation as follows

$$\frac{\vartheta_j^i-\vartheta_j^{i-1}}{k}=\frac{\vartheta_{j+1}^i-2\vartheta_j^i+\vartheta_{j-1}^i}{h^2} \tag{4.19}$$

or

$$\boxed{\vartheta_j^{i-1}=-\frac{k}{h^2}\vartheta_{j+1}^i+\left(1+2\frac{k}{h^2}\right)\vartheta_j^i-\frac{k}{h^2}\vartheta_{j-1}^i} \tag{4.20}$$

Similarly, equations (4.12) and (4.14) are substituted into diffusion equation in cathode (3.74), simplified and have obtained the diffusion equation in cathode (3.74) for implicit scheme as follows

$$\frac{\vartheta_j^i - \vartheta_j^{i-1}}{k} = \varepsilon^{1/2}\left(\frac{\vartheta_{j+1}^i - 2\vartheta_j^i + \vartheta_{j-1}^i}{h^2}\right) + J \tag{4.21}$$

or

$$\boxed{\vartheta_j^{i-1} = -\varepsilon^{1/2}\frac{k}{h^2}\vartheta_{j+1}^i + \left(1 + 2\varepsilon^{1/2}\frac{k}{h^2}\right)\vartheta_j^i - \varepsilon^{1/2}\frac{k}{h^2}\vartheta_{j-1}^i + kJ} \tag{4.22}$$

Equation (4.20) and (4.22) has given the implicit scheme under FDM. Next for the Crank Nicolson scheme the *i*-term is rearranged by one step forward in equation (4.19) to get as follows

$$\frac{\vartheta_j^{i+1} - \vartheta_j^i}{k} = \frac{\vartheta_{j+1}^{i+1} - 2\vartheta_j^{i+1} + \vartheta_{j-1}^{i+1}}{h^2} \tag{4.23}$$

Then the left side of the diffusion equation in separator (3.73) by equation (4.11) is replaced and the right side of the diffusion equation in separator (3.73) is replaced by the average of the central difference from equation (4.15) and equation (4.23) to get as follows

$$\frac{\vartheta_j^{i+1} - \vartheta_j^i}{k} = \frac{1}{2}\left(\frac{\vartheta_{j+1}^i - 2\vartheta_{j+1}^i + \vartheta_{j-1}^i}{h^2} + \frac{\vartheta_{j+1}^{i+1} - 2\vartheta_j^{i+1} + \vartheta_{j-1}^{i+1}}{h^2}\right) \tag{4.24}$$

or

$$\boxed{-\frac{k}{2h^2}\vartheta_{j-1}^{i+1} + \left(1 + \frac{k}{h^2}\right)\vartheta_j^{i+1} - \frac{k}{2h^2}\vartheta_{j-1}^{i+1} = -\frac{k}{2h^2}\vartheta_{j-1}^i + \left(1 + \frac{k}{h^2}\right)\vartheta_j^i - \frac{k}{2h^2}\vartheta_{j-1}^i} \tag{4.25}$$

For the diffusion equation in cathode (3.74), the left side of the equation by equation (4.11) is replaced and the right side of the diffusion equation in cathode (3.74) is replaced by the average of the central difference from equation (4.15) and equation (4.23) to get as follows

$$\frac{\vartheta_j^{i+1}-\vartheta_j^i}{k}=\frac{1}{2}\varepsilon^{1/2}\left(\frac{\vartheta_{j+1}^i-2\vartheta_{j+1}^i+\vartheta_{j-1}^i}{h^2}+\frac{\vartheta_{j+1}^{i+1}-2\vartheta_j^{i+1}+\vartheta_{j-1}^{i+1}}{h^2}\right)+J \tag{4.26}$$

or

$$\boxed{-\varepsilon^{1/2}\frac{k}{2h^2}\vartheta_{j-1}^{i+1}+\left(1+\varepsilon^{1/2}\frac{k}{h^2}\right)\vartheta_j^{i+1}-\varepsilon^{1/2}\frac{k}{2h^2}\vartheta_{j-1}^{i+1}=-\varepsilon^{1/2}\frac{k}{2h^2}\vartheta_{j-1}^i+\left(1+\varepsilon^{1/2}\frac{k}{h^2}\right)\vartheta_j^i-\varepsilon^{1/2}\frac{k}{2h^2}\vartheta_{j-1}^i+J} \tag{4.27}$$

Hence, equation (4.26) and equation (4.27) are Crank Nicolson scheme for FDM. FDM also offer one formulation called as Theta formulation (Farlow 1982; Hoffmann and Chiang 2000; Holmes 2007) which is able to present both these schemes in a one general form. To get this general form, a similar step would be used to obtain the Crank Nicolson scheme but for this equation the right side of the diffusion equation in separator (3.73) is replaced and diffusion equation in cathode (3.74) is replaced by convex combination of central difference from equation (4.15) and equation (4.23) to get the new diffusion equation in separator as follows

$$\frac{\vartheta_j^{i+1}-\vartheta_j^i}{k}=(1-\alpha)\frac{\vartheta_{j+1}^i-2\vartheta_{j+1}^i+\vartheta_{j-1}^i}{h^2}+\alpha\frac{\vartheta_{j+1}^{i+1}-2\vartheta_j^{i+1}+\vartheta_{j-1}^{i+1}}{h^2} \tag{4.28}$$

or

$$\boxed{-\alpha\frac{k}{h^2}\vartheta_{j+1}^{i+1}+\left(1+2\alpha\frac{k}{h^2}\right)\vartheta_{j+1}^{i+1}-\alpha\frac{k}{h^2}\vartheta_{j-1}^{i+1}=(1-\alpha)\frac{k}{h^2}\vartheta_{j+1}^i+\left[1-2(1-\alpha)\frac{k}{h^2}\right]\vartheta_{j+1}^i+(1-\alpha)\frac{k}{h^2}\vartheta_{j-1}^i} \tag{4.29}$$

and the new diffusion equation in cathode is as follows

$$\frac{\vartheta_j^{i+1}-\vartheta_j^i}{k}=(1-\alpha)\varepsilon^{1/2}\left(\frac{\vartheta_{j+1}^i-2\vartheta_{j+1}^i+\vartheta_{j-1}^i}{h^2}\right)+\alpha\varepsilon^{1/2}\left(\frac{\vartheta_{j+1}^{i+1}-2\vartheta_j^{i+1}+\vartheta_{j-1}^{i+1}}{h^2}\right)+J \tag{4.30}$$

or

$$\boxed{\begin{aligned}&-\alpha\varepsilon^{1/2}\frac{k}{h^2}\vartheta_{j+1}^{i+1}+\left(1+2\alpha\varepsilon^{1/2}\frac{k}{h^2}\right)\vartheta_{j+1}^{i+1}-\alpha\varepsilon^{1/2}\frac{k}{h^2}\vartheta_{j-1}^{i+1}=\\&\qquad(1-\alpha)\varepsilon^{1/2}\frac{k}{h^2}\vartheta_{j+1}^i+\left[1-2(1-\alpha)\varepsilon^{1/2}\frac{k}{h^2}\right]\vartheta_{j+1}^i+(1-\alpha)\varepsilon^{1/2}\frac{k}{h^2}\vartheta_{j-1}^i+kJ\end{aligned}} \tag{4.31}$$

According to Hoffmann and Chiang (2000), Rezolla (2010) and Recktenwald (2011), for $\frac{1}{2}\le\alpha\le 1$, equation (4.29) and (4.31) is unconditionally stable, means that there is no condition to choose the value of *h* or *k*. Beside that, for $0\le\alpha<\frac{1}{2}$ equation (4.29) and (4.31) is conditionally stable. The conditions that need to be fulfilled is $(1-\alpha)\frac{\varepsilon^{\frac{1}{2}}k}{h^2}<0.5$. From these two equations (4.29) and (4.31) we just need to substitute $\alpha=\frac{1}{2}$ to get the Crank Nicolson scheme, $\alpha=1$ to get implicit scheme and $\alpha=0$ to get explicit scheme. Hence, the general form from Theta formulation gives an easier way to implement the computer coding because it is only required to replace the value of α in the coding in order to get the other three scheme solutions.

Now, initial condition (3.80) is substituted with equation (4.6) and boundary conditions (3.81), (3.94), (3.95) and (3.96) are substituted with the equation (4.11) to equation (4.13) and rearranged to get new initial condition and boundary conditions as follows

$$\vartheta^{0}_{j,separator} = \vartheta^{0}_{j,cathode} = 1 \qquad \text{for} \qquad 0 \le j \le n \tag{4.32}$$

$$\vartheta^{i}_{0} = \vartheta^{i}_{1} - J\varepsilon rh \qquad \text{for} \qquad 1 \le i \le m \tag{4.33}$$

$$\vartheta^{i}_{n} = \vartheta^{i}_{n-1} \qquad \text{for} \qquad 1 \le i \le m \tag{4.34}$$

$$\vartheta^{i}_{v} = \frac{\varepsilon^{3/2}\vartheta^{i}_{v+1} + \vartheta^{i}_{v-1}}{1+\varepsilon^{3/2}} \qquad \text{for} \qquad 1 \le i \le m \tag{4.35}$$

$$\vartheta^{i}_{v,separator} = \vartheta^{i}_{v,cathode} \qquad \text{for} \qquad 1 \le i \le m \tag{4.36}$$

where *v* are the phase between separator and cathode $(x = \delta_s)$ or $(y = 1)$. Then the new diffusion equation in separator (4.29) and new diffusion equation in cathode (4.31) that are called as the Theta formulation are now subject to new initial and boundary condition equation (4.32) to equation (4.36). After the replacement, the diffusion equations in separator and cathode, initial condition and boundary conditions will become a system of algebraic equations. Then, this system of equations can be solved numerically by an iterative process to get its approximations solution towards the diffusion equation in separator and cathode. This iterative process formerly becomes a problem to all scientists and researchers but nowadays there are several computer software available that can be used as a tool to solve the equations.

4.3 WOLFRAM MATHEMATICA 8 SOFTWARE

As we know the major problem or difficulty about the numerical method is its iteration and time. The speed and accuracy of the method depends upon the technique, the system and complexity of boundary conditions. Normally, numerical method need more time in dealing with this kind of algebraic equations system (the diffusion equations in separator and cathode, initial condition and boundary conditions). So to shorten and minimize the time needed to solve the system of algebraic equation in this work, computational software is used as a tool to solve this system. There are many softwares that could be used to solve this problem such as Maple (Subramanian et al. 2007), Fotran (Botte 2000), Mathlab, Comsol (Cai and White 2009), C++, Mathematica and Mathcad. (Botte et al. 2000) has explained quiet well about software that has been used by other researchers. Nowadays, Maple, Mathlab and Mathematica are the most popular mathematical software in order to solve mathematical problems (includes the system of algebraic equations). (Chonacky and Winch 2005; Lipsman et al. 2008; Steinhaus 2008; Stein 2009) have written reviews about several of the mathematical software nowadays.

Steinhaus (2008) has done a survey on comparison of mathematical programs for data analysis and the results are shown in table 4.1. This comparison is between several popular software such as *GAUSS* from Aptech Systems Inc., *Maple* from Waterloo Maple Software Inc., *Mathematica* from Wolfram Research Inc., *Matlab* from The Mathworks Inc., *O-Matrix* from Harmonic Software, *OxMetrics (Ox Prof.)* from Timberlake Consultants Ltd. and *Scilab* from INRIA.

Table 4.1: Comparison of mathematical programs for data analysis in aspect of (a) installation, learnability and usability, (b) mathematical functionality, (c) graphical functionality, (d) data handling, (e) Available operating systems, (f) speed comparison, (g) Functionality of the programming environment and (h) overall result. (Steinhaus 2008)

Function	Software (Version)						
	GAUSS (8.0)	Maple (VII)	Mathematica (6.0)	Mathlab (2008a)	O-Matrix (6.3)	Ox Prof (5.0)	Scilab (4.1.2)
Installation (10%)	66.67%	100.00%	83.33%	100.00%	33.33%	66.67%	80.00%
Learnability (30%)	58.46%	98.46%	98.46%	83.08%	46.15%	52.31%	52.31%
Usability (60%)	18.67%	80.00%	96.27%	76.52%	41.18%	34.36%	39.69%
Overall Result (100% = Best)	35.41%	87.54%	96.27%	76.52%	41.18%	34.36%	39.69%

Table 4.1(a)

Function	Software (Version)						
	GAUSS (8.0)	Maple (VII)	Mathematica (6.0)	Mathlab (2008a)	O-Matrix (6.3)	Ox Prof (5.0)	Scilab (4.1.2)
Standard Mathematics (5%)	77.27%	100.00%	100.00%	98.18%	81.82%	90.91%	81.82%
Algebra (15%)	76.97%	87.88%	84.85%	93.94%	72.73%	66.67%	78.79%
Analysis (10%)	84.62	100.00%	100.00%	100.00%	53.85%	46.15%	84.62%
Numerical Mathematics (10%)	53.33%	75.00%	100.00%	85.00%	41.67%	33.33%	41.67%
Descriptive Statistic, Stochastic and Distribution Function (20%)	63.78%	64.44%	92.00%	46.89%	23.56%	60.00%	35.00%
Statistics (20%)	64.17%	9.57%	34.96%	53.39%	25.04%	41.74%	33.91%
Other Mathematics (20%)	73.85%	23.08%	64.62%	56.15%	11.54%	57.69%	11.54%
Overall Result (100% = Best)	69.56%	55.10%	76.04%	68.79%	36.58%	54.38%	44.63%

Table 4.1(b)

Table 4.1: Comparison of mathematical programs for data analysis in aspect of (a) installation, learnability and usability, (b) mathematical functionality, (c) graphical functionality, (d) data handling, (e) Available operating systems, (f) speed comparison, (g) Functionality of the programming environment and (h) overall result. (Steinhaus 2008)

Function	Software (Version)						
	GAUSS (8.0)	Maple (VII)	Mathematica (6.0)	Mathlab (2008a)	O-Matrix (6.3)	Ox Prof (5.0)	Scilab (4.1.2)
Graphic types (75%)	63.00%	64.50%	79.50%	86.50%	41.00%	47.50%	42.50%
Graphic import/export formats (25%)	54.44%	50.00%	100.00%	94.44%	65.56%	33.33%	72.78%
Overall Result (100% = Best)	60.86%	60.88%	84.63%	88.49%	47.14%	43.96%	51.32%

Table 4.1(c)

Function	Software (Version)						
	GAUSS (8.0)	Maple (VII)	Mathematica (6.0)	Mathlab (2008a)	O-Matrix (6.3)	Ox Prof (5.0)	Scilab (4.1.2)
Data import/export (70%)	50.00%	56.00%	778.00%	66.00%	48.00%	40.00%	40.00%
Data handling and preparation (30%)	91.43%	82.86%	76.03%	88.57%	54.29%	91.43%	85.71%
Overall Result (100% = Best)	62.43%	64.06%	76.03%	72.77%	49.89%	55.43%	53.71%

Table 4.1(d)

Table 4.1: Comparison of mathematical programs for data analysis in aspect of (a) installation, learnability and usability, (b) mathematical functionality, (c) graphical functionality, (d) data handling, (e) Available operating systems, (f) speed comparison, (g) Functionality of the programming environment and (h) overall result. (Steinhaus 2008)

Platform	**Software (Version)**						
	GAUSS (8.0)	**Maple (VII)**	**Mathematica (6.0)**	**Mathlab (2008a)**	**O-Matrix (6.3)**	**Ox Prof (5.0)**	**Scilab (4.1.2)**
HP 9000 (HP-UX)	+	-	+	-	-	+	+
IBM RISC (IBM AIX)	-	-	+	-	-	+	-
Intel / AMD 32 Bit (Windows)	+ / +	+ / +	+ / +	+ / +	+ / +	+ / +	+ / +
Intel / AMD 64 Bit (Windows)	+ / -	- / -	+ / +	+ / +	- / -	+ / +	- / -
Intel / AMD 32 Bit (Linux)	+ / +	+ / +	+ / +	+ / +	- / -	+ / +	- / -
Intel / AMD 64 Bit (Linux)	+ / +	+ / +	+ / +	+ / +	- / -	+ / +	- / -
Intel 32 Bit (MAC OS)	+	+	+	+	-	+	+
Intel 64 Bit (MAC OS)	-	+	+	-	-	-	-
SUN (Solaris)	+	+	+	+	-	+	-
Total amount	76.92% (10/13)	69.23% (9/13)	100.00% (13/13)	76.92% (10/13)	15.38% (2/13)	92.31% (12/13)	46.15% (6/13)

Table 4.1(e)

Table 4.1: Comparison of mathematical programs for data analysis in aspect of (a) installation, learnability and usability, (b) mathematical functionality, (c) graphical functionality, (d) data handling, (e) Available operating systems, (f) speed comparison, (g) Functionality of the programming environment and (h) overall result. (Steinhaus 2008)

Functions	Software (Version)						
	GAUSS (8.0)	Maple (VII)	Mathematica (6.0)	Mathlab (2008a)	O-Matrix (6.3)	Ox Prof (5.0)	Scilab (4.1.2)
IQ test and descriptive statistic	2.479	127.483	6.968	10.375	14.583	5.485	21.036
Loop test 15.000 x 15.000	29.427	259.763	323.828	67.258	58.927	32.229	511.323
2000 x 2000 random matrix^1000	122.829	292.570	31.395	57.393	9.618	15.178	257.784
Sorting of 1000000 random values	67.439	236.543	38.604	14.217	8.985	14.238	36.274
FFT over 1048576 (= 2^20) random values	166.332	47.083	16.864	11.660	7.895	13.386	16.261
Determinant of a 1500 x 1500 random matrix	504.648	43.456	24.304	31.209	19.615	138.781	56.155
Inverse of a 1500 x 1500 random matrix	562.355	150.109	74.452	73.247	54.286	342.961	152.899
Eigenvalues of a 1200 x 1200 random matrix	28.943	27.140	10.562	4.910	4.713	14.484	12.000
Cholesky decomposition of a 1500 x 1500 random matrix	128.197	414.401	19.638	12.980	14.018	43.515	36.566
1500 x 1500 cross product matrix	779.001	100.698	48.264	23.459	35.590	107.149	121.302
Calculation of 10000000 Fibonacci numbers	3.334	821.140	2.172	1.434	1.042	2.261	3.193
Principal component factorization over a 10000 x 1000 matrix	60.578	422.140	37.875	27.107	4.369	-	51.771
Gamma function on a 1500 x 1500 random matrix	53.906	10081.174	252.987	27.473	26.585	6.881	26.104
Gaussian error function on a 1500 x 1500 random matrix	63.755	4815.049	272.294	27.255	1.985	5.327	27.787
Linear regression over a 1000 x 1000 random matrix	154.978	179.907	10.787	13.798	7.005	43.970	20.646
Overall performance	21.848%	11.159%	39.072%	54.676%	83.422%	42.364%	24.510%

Table 4.1(f)

Table 4.1: Comparison of mathematical programs for data analysis in aspect of (a) installation, learnability and usability, (b) mathematical functionality, (c) graphical functionality, (d) data handling, (e) Available operating systems, (f) speed comparison, (g) Functionality of the programming environment and (h) overall result. (Steinhaus 2008)

Programming facillities	Software (Version)						
	GAUSS (8.0)	Maple (VII)	Mathematica (6.0)	Mathlab (2008a)	O-Matrix (6.3)	Ox Prof (5.0)	Scilab (4.1.2)
Editing features							
Built-in editor	+	+	+	+	+	+	+
External editor configurable	+	+	+	+	-	+	-
Source code formatting	+	-	+	+	-	-	+
Syntax highlighting	+	-	+	+	-	+	+
Command completion	-	+	+	+	-	-	+
Debugging							
Breakpoints	+	+	+	+	+	+	+
Function Tracer	+	+	+	-	-	+	-
Line Tracing	+	-	+	+	+	+	-
Profiler	+	+	$	+	+	-	+
Stack inspection	+	-	+	+	+	-	+
Variable inspection	+	-	+	+	+	+	+
Code advisory / best practice report	-	-	-	+	-	-	-
Language features							
API-interface	+	+	+	+	+	+	+
Compiler metacommands	+	-	-	-	-	+	-
Fuzzy conditional functions	+	-	$	$	-	+	-
GUI programming	-	+	+	+	+	+	+
Loops head / foot controlled	+ / +	+ / +	+ / -	+ / -	+ / -	+ / +	- / +

N-dimensional arrays (> 3)	+	+	+	+	**-**	+	+
Object oriented programming	**-**	**-**	+	+	+	+	+
OLE support	**-**	+	+	+	**-**	+	**m**
P code compiling	+	+	+	+	**$**	+	+
Web hosting	**-**	**$**	**$**	**-**	**-**	**-**	**m**
Language interfaces							
C/C++	+	+	+	+	**-**	+	+
GAUSS	+	**-**	**-**	**-**	**-**	+	**-**
Maple	**$**	+	**-**	**$**	**-**	**-**	+
Mathematica	**$**	+	+	**$**	+	**-**	**-**
Matlab	**-**	+	**$**	+	**-**	**-**	+
O-Matrix	**-**	**-**	**$**	**-**	+	**-**	**-**
Ox	**-**	**-**	**-**	**-**	**-**	+	**-**
Scilab	**-**	**-**	**-**	**-**	**-**	**-**	+
DLL-Calls	+	+	+	+	+	+	+
Miscellaneous							
Developer Engine	**$**	**-**	**-**	**$**	**$**	+	+
Redistr. with runtime licenses	**$**	**-**	**-**	**$**	**$**	**$**	+
Standalone application compilation	**-**	**-**	**-**	**$**	**-**	**-**	**-**
Source code optimization	**-**	**-**	**-**	**-**	**-**	**-**	**-**
Interface to source control system (i.e. Visual SourceSafe)	**-**	**-**	**-**	+	**-**	**-**	**-**
Overall Result (100% = Best)	62.70% (23.2/37)	50.81% (18.8/37)	64.86% (24/37)	72.43% (26.8/37)	41.62% (15.4/37)	72.43% (26.8/37)	62.16% (23/37)

Table 4.1(g)

Table 4.1: Comparison of mathematical programs for data analysis in aspect of (a) installation, learnability and usability, (b) mathematical functionality, (c) graphical functionality, (d) data handling, (e) Available operating systems, (f) speed comparison, (g) Functionality of the programming environment and (h) overall result. (Steinhaus 2008)

Test	Software (Version)						
	GAUSS (8.0)	**Maple (VII)**	**Mathematica (6.0)**	**Mathlab (2008a)**	**O-Matrix (6.3)**	**Ox Prof (5.0)**	**Scilab (4.1.2)**
Installation, learnability and usability (15%)	35.41%	87.54%	96.27%	76.52%	41.18%	34.36%	39.69%
Comparison of mathematical functionality (35%)	69.56%	55.10%	76.04%	68.79%	36.58%	54.38%	44.63%
Comparison of graphical functionality (10%)	60.86%	60.88%	84.63%	88.49%	47.14%	43.96%	51.32%
Functionality of the programming environment (11%)	62.70%	50.81%	64.86%	72.43%	41.62%	72.43%	62.16%
Data handling (5%)	62.43%	64.06%	76.03%	72.77%	49.89%	55.43%	53.71%
Available platforms (2%)	76.92%	69.23%	100.00%	76.92%	15.38%	92.31%	46.15%
Speed Comparison (22%)	21.85%	11.16%	39.07%	54.68%	83.42%	42.36%	24.51%
Overall Result	52.11%	51.13%	71.05%	69.58%	49.43%	50.49%	42.54%

Table 4.1(h)

According to him, table 4.1(a), Wolfram Mathematica (96.27%) is the best software that can be classified as user friendly (Installation, Learnability and Usability). His study also dealt into the comparison from the aspect of mathematical functionality. This survey evaluated the competency of the software to solve various mathematical functionalities. The results are in table 4.1(b), where Mathematica leads with 76.04%. The result of comparison in the aspect of the graphical functionality stated that Mathlab is the best software with 88.49%, followed by Mathematica with 84.63% (table 4.1(b)). For data handling aspect comparison (76.03%) and the availability towards various operating systems (100%), Mathematica was again selected as the best software compared to other softwares as shown in table 4.1(d) and table 4.1(e)). On the other hand, Mathematica also has its disadvantages, as refer to table 4.1(f) and table 4.1(g). According to Steinhaus (2008), Mathematica is weak from the aspect of speed comparison (39.072%) and functionality of the programming environment (64.86%). This speed comparison has been performed on Intel Quad Core Q6600 processor with 2.4 GHz and 2 GB RAM running under Windows Vista Home (all timings are displayed in seconds). However the most important result on Steinhaus (2008) study is the overall result where Wolfram Mathematica has been classified as the best mathematical software with 71.05% . The overall result is shown in table 4.1(h).

In another study, Lipsman et al. (2008) found that the differences between these various Mathematical software packages to be relatively minor, compared to the features that they all shared in common. They also found that the differences tended to dissipate with time. In the early 90s, Mathematica and Maple emphasized their “front ends," called notebooks and worksheets, respectively, whereas MATLAB more closely resembled conventional programming languages such as FORTRAN. This feature makes numerical calculation faster in MATLAB, but Mathematica and Maple were more convenient for producing finished documents in a single step (without having to run code, save the output, and paste it into a document). MATLAB was superior for numerical calculation, especially for calculations involving matrices. Beside that MATLAB also was greater for numerical linear algebra but nowadays these differences no longer matter much. While efficiency and speed are still relevant issues for large-scale computations, as desktop and laptop computers have become faster, speed is no longer a consideration. In fact, the “front ends" of all the major problem solving environments (PSE) have improved to the point where they all can be used to produce finished documents incorporating descriptive text, code, numerical or symbolic output, and graphics. There were significant differences between the sorts of problems that could be handled by each of the major PSEs.

From Steinhaus (2008) and Lipsman et al. (2008) survey and review, they showed that the Wolfram Mathematica software is one of the best mathematical software compared to other software. Considering the fact that no other researchers so far used the Wolfram Mathematica software to solve this problem (problem developed in this work), thus the Wolfram Mathematica 8 (WM8) has been chosen as a tool to solve the problem (system of algebraic equation) in this work.

4.4 WOLFRAM MATHEMATICA 8 ALGORITHM

So far we have developed the governing equation, chosen the method of solving (FDM) and choose the tool (WM8) to solve the problem (system of algebraic equation). Now we going to develop the algorithm or programming code in WM8. Figure 4.2 shows the WM8 notebook sheet.

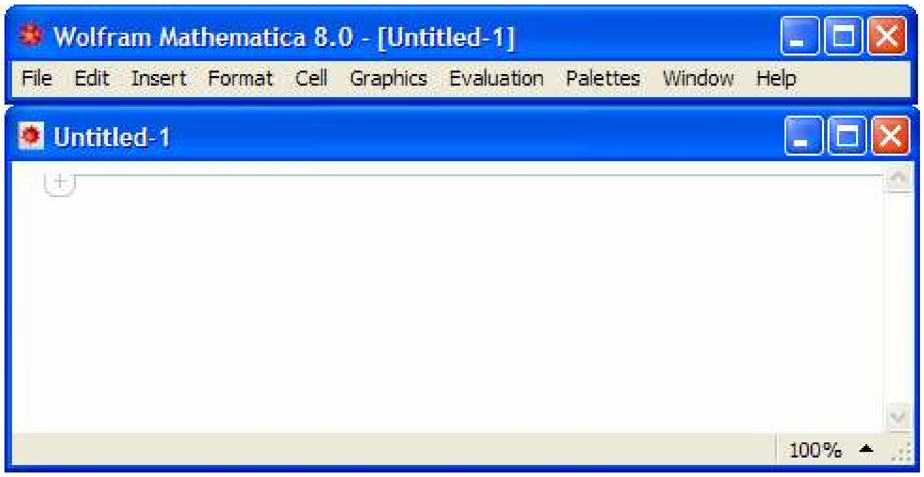


Figure 4.2: WM8 notebook sheet.

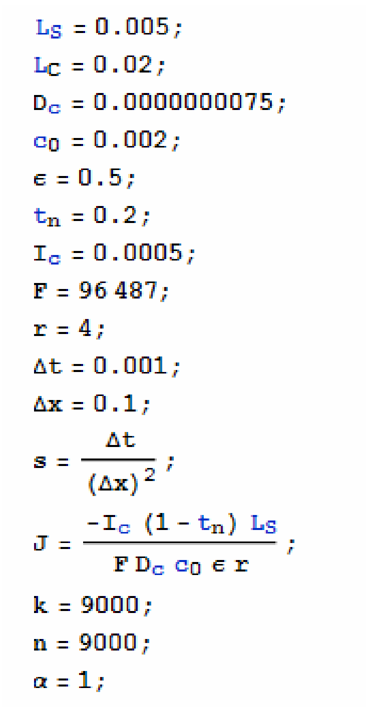
The algorithms begin with defining the parameters involved in the system of algebraic equation. In order to make the comparison with Doyle and Newman (1997) study, the parameter value below are taken to be similar with their findings. The comparison result will be discussed later in chapter 5.

STEP 1: Defining the parameters involved

```
"Ls,Length of seperator=0.005"
"Lc,Length of cathode=0.02"
"Dc,Diffusion coefficient=0.000000075"
"c0,Initial concentration=0.002"
"ϵ,Porosity of the cathode=0.5"
"tn,Tranference number=0.2"
"F,Faraday's constant=96487"
"Ic,Superficial current density"
"r,xyz"
              -Ic (1 - t) Ls
"J=────────────────────────────── "
    F SubscriptBox[D, c] c0 ϵ r
```

For the second step, we key in the input value for the parameter involved as follows

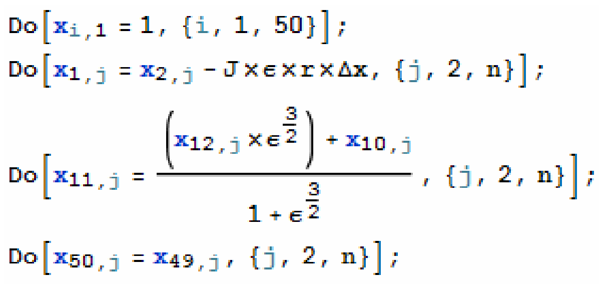
STEP 2: Input value for the parameter involved



$$L_S = 0.005;$$
$$L_C = 0.02;$$
$$D_c = 0.0000000075;$$
$$c_0 = 0.002;$$
$$\epsilon = 0.5;$$
$$t_n = 0.2;$$
$$I_c = 0.0005;$$
$$F = 96487;$$
$$r = 4;$$
$$\Delta t = 0.001;$$
$$\Delta x = 0.1;$$
$$s = \frac{\Delta t}{(\Delta x)^2};$$
$$J = \frac{-I_c\,(1 - t_n)\,L_S}{F\,D_c\,c_0\,\epsilon\,r};$$
$$k = 9000;$$
$$n = 9000;$$
$$\alpha = 1;$$

As for the third step, the algorithm for the initial and boundary conditions are being set up as follows

STEP 3: Setting up the initial and boundary conditions



$$\mathrm{Do}\left[x_{i,1} = 1,\ \{i,\ 1,\ 50\}\right];$$
$$\mathrm{Do}\left[x_{1,j} = x_{2,j} - J \times \epsilon \times r \times \Delta x,\ \{j,\ 2,\ n\}\right];$$
$$\mathrm{Do}\left[x_{11,j} = \frac{\left(x_{12,j} \times \epsilon^{\frac{3}{2}}\right) + x_{10,j}}{1 + \epsilon^{\frac{3}{2}}},\ \{j,\ 2,\ n\}\right];$$
$$\mathrm{Do}\left[x_{50,j} = x_{49,j},\ \{j,\ 2,\ n\}\right];$$

The system of algebraic solution in chapter 3 will be solved using matrix form as follows

$$Ax = b \tag{4.37}$$

where *A* is the matrix with the form of as follows

m row x n column matrix

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \Lambda & a_{1,n} \\ a_{2,1} & a_{2,2} & \Lambda & a_{2,n} \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} \\ a_{m,1} & a_{m,2} & \Lambda & a_{m,n} \end{bmatrix} \tag{4.38}$$

x is the matrix with the form as follows

n row x 1 column matrix

$$x = \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \mathrm{M} \\ x_{n,1} \end{bmatrix} \tag{4.39}$$

and *b* is the matrix with the form as follows

m row x 1 column matrix

$$x = \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \mathrm{M} \\ x_{m,1} \end{bmatrix} \tag{4.40}$$

The inverse matrix for A denoted by A^{-1} is introduced and multiplied into both side in equation (4.37) to get as follows

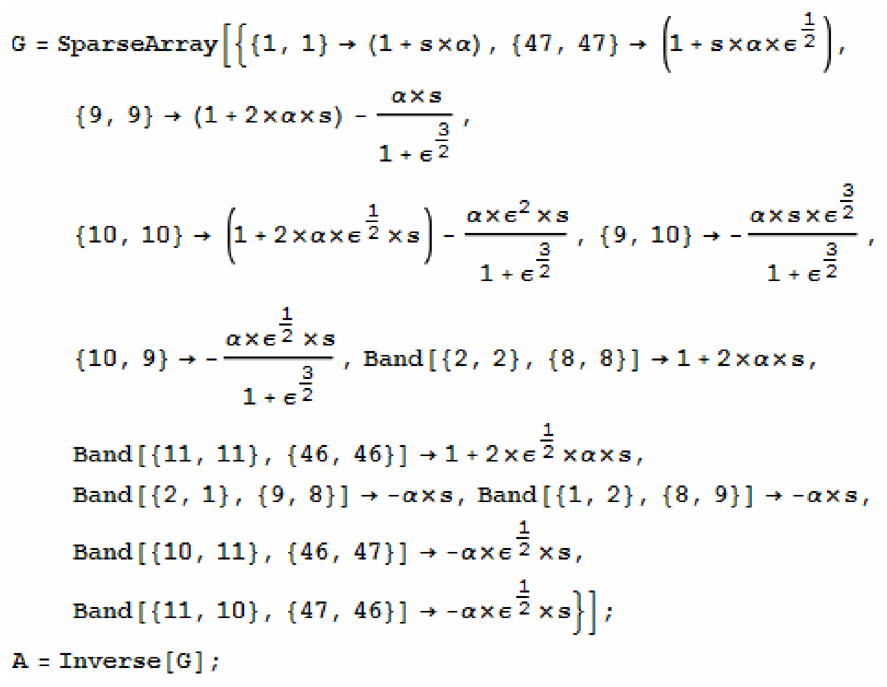
$$A^{-1}Ax = A^{-1}b \tag{4.41}$$

The equation (4.41) is simplified to get as follows

$$x = A^{-1}b \tag{4.42}$$

Next the equation (4.42) is applied into WM8, and followed by the fourth step which is to develop algorithm to matrix A^{-1}, matrix b, matrix x and to compute $x = A^{-1}b$

STEP 4.1: Calculating the inverse matrix, A^{-1}



```
G = SparseArray[{{1, 1} → (1 + s × α), {47, 47} → (1 + s × α × ϵ^(1/2)),
    {9, 9} → (1 + 2 × α × s) - (α × s)/(1 + ϵ^(3/2)),
    {10, 10} → (1 + 2 × α × ϵ^(1/2) × s) - (α × ϵ^2 × s)/(1 + ϵ^(3/2)), {9, 10} → -(α × s × ϵ^(3/2))/(1 + ϵ^(3/2)),
    {10, 9} → -(α × ϵ^(1/2) × s)/(1 + ϵ^(3/2)), Band[{2, 2}, {8, 8}] → 1 + 2 × α × s,
    Band[{11, 11}, {46, 46}] → 1 + 2 × ϵ^(1/2) × α × s,
    Band[{2, 1}, {9, 8}] → -α × s, Band[{1, 2}, {8, 9}] → -α × s,
    Band[{10, 11}, {46, 47}] → -α × ϵ^(1/2) × s,
    Band[{11, 10}, {47, 46}] → -α × ϵ^(1/2) × s}];
A = Inverse[G];
```

STEP 4.2: Defining the matrix *b*

$$\mathrm{For}\Big[m=1,\ m<(n+1),\ m{+}{+},\ \mathrm{Do}\Big[S1_m=$$

$$\left(\begin{array}{c}
(1-\alpha)\ s\ x_{i,m}+(1-2\ s\ (1-\alpha))\ x_{i+1,m}+(1-\alpha)\ s\ x_{i+2,m}-(J\times\epsilon\times r\times\Delta x\times\alpha\times s)\\
(1-\alpha)\ s\ x_{i+1,m}+(1-2\ s\ (1-\alpha))\ x_{i+2,m}+(1-\alpha)\ s\ x_{i+3,m}\\
(1-\alpha)\ s\ x_{i+2,m}+(1-2\ s\ (1-\alpha))\ x_{i+3,m}+(1-\alpha)\ s\ x_{i+4,m}\\
(1-\alpha)\ s\ x_{i+3,m}+(1-2\ s\ (1-\alpha))\ x_{i+4,m}+(1-\alpha)\ s\ x_{i+5,m}\\
(1-\alpha)\ s\ x_{i+4,m}+(1-2\ s\ (1-\alpha))\ x_{i+5,m}+(1-\alpha)\ s\ x_{i+6,m}\\
(1-\alpha)\ s\ x_{i+5,m}+(1-2\ s\ (1-\alpha))\ x_{i+6,m}+(1-\alpha)\ s\ x_{i+7,m}\\
(1-\alpha)\ s\ x_{i+6,m}+(1-2\ s\ (1-\alpha))\ x_{i+7,m}+(1-\alpha)\ s\ x_{i+8,m}\\
(1-\alpha)\ s\ x_{i+7,m}+(1-2\ s\ (1-\alpha))\ x_{i+8,m}+(1-\alpha)\ s\ x_{i+9,m}\\
(1-\alpha)\ s\ x_{i+8,m}+(1-2\ s\ (1-\alpha))\ x_{i+9,m}+(1-\alpha)\ s\ x_{i+10,m}\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+10,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+11,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+12,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+11,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+12,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+13,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+12,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+13,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+14,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+13,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+14,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+15,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+14,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+15,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+16,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+15,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+16,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+17,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+16,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+17,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+18,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+17,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+18,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+19,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+18,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+19,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+20,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+19,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+20,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+21,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+20,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+21,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+22,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+21,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+22,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+23,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+22,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+23,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+24,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+23,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+24,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+25,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+24,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+25,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+26,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+25,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+26,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+27,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+26,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+27,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+28,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+27,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+28,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+29,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+28,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+29,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+30,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+29,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+30,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+31,m}+J\times\Delta t\\
(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+30,m}+\left(1-2\ s\times\left(\epsilon^{\frac{1}{2}}\right)(1-\alpha)\right)x_{i+31,m}+(1-\alpha)\ s\times\left(\epsilon^{\frac{1}{2}}\right)x_{i+32,m}+J\times\Delta t
\end{array}\right.$$

$$\begin{pmatrix}
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+31,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+32,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+33,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+32,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+33,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+34,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+33,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+34,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+35,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+34,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+35,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+36,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+35,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+36,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+37,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+36,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+37,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+38,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+37,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+38,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+39,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+38,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+39,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+40,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+39,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+40,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+41,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+40,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+41,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+42,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+41,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+42,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+43,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+42,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+43,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+44,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+43,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+44,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+45,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+44,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+45,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+46,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+45,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+46,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+47,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+46,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+47,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+48,m} + J \times \Delta t \\
(1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+47,m} + \left(1 - 2\ s \times \left(\epsilon^{\frac{1}{2}}\right) (1-\alpha)\right) x_{i+48,m} + (1-\alpha)\ s \times \left(\epsilon^{\frac{1}{2}}\right) x_{i+49,m} + J \times \Delta t
\end{pmatrix}$$

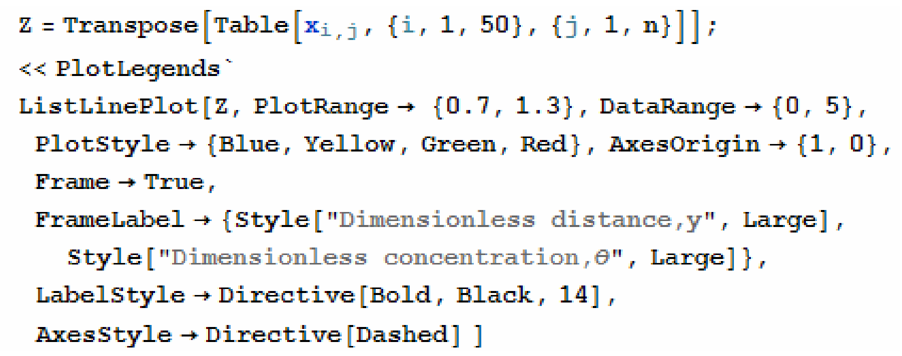
```
, {i = 1}]];
```

STEP 4.3: Computing the matrix $x = A^{-1}b$

$$\text{For}\left[m = 2,\ m < (n + 1),\ m{+}{+},\ \text{Do}\left[\begin{pmatrix} x_{i+1,m}\\ x_{i+2,m}\\ x_{i+3,m}\\ x_{i+4,m}\\ x_{i+5,m}\\ x_{i+6,m}\\ x_{i+7,m}\\ x_{i+8,m}\\ x_{i+9,m}\\ x_{i+11,m}\\ x_{i+12,m}\\ x_{i+13,m}\\ x_{i+14,m}\\ x_{i+15,m}\\ x_{i+16,m}\\ x_{i+17,m}\\ x_{i+18,m}\\ x_{i+19,m}\\ x_{i+20,m}\\ x_{i+21,m}\\ x_{i+22,m}\\ x_{i+23,m}\\ x_{i+24,m}\\ x_{i+25,m}\\ x_{i+26,m}\\ x_{i+27,m}\\ x_{i+28,m}\\ x_{i+29,m}\\ x_{i+30,m}\\ x_{i+31,m}\\ x_{i+32,m}\\ x_{i+33,m}\\ x_{i+34,m}\\ x_{i+35,m}\\ x_{i+36,m}\\ x_{i+37,m}\\ x_{i+38,m}\\ x_{i+39,m}\\ x_{i+40,m}\\ x_{i+41,m}\\ x_{i+42,m}\\ x_{i+43,m}\\ x_{i+44,m}\\ x_{i+45,m}\\ x_{i+46,m}\\ x_{i+47,m}\\ x_{i+48,m} \end{pmatrix} = A.S1_{m-1},\ \{i = 1\}\right]\right];$$

Finally, the fifth step is to develop the algorithm for the simulation result. In the final step, the algorithm for lithium-ion concentration profile will be shown in the solution phase during 1.75 mA/cm^2 discharged current at various value of dimensionless time (simulation result 1)

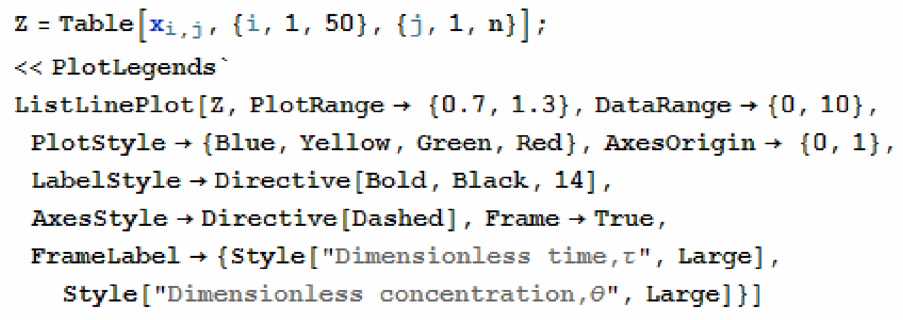
<u>STEP 5.1: Simulation result 1</u>



```
Z = Transpose[Table[xi,j, {i, 1, 50}, {j, 1, n}]];
<< PlotLegends`
ListLinePlot[Z, PlotRange → {0.7, 1.3}, DataRange → {0, 5},
 PlotStyle → {Blue, Yellow, Green, Red}, AxesOrigin → {1, 0},
 Frame → True,
 FrameLabel → {Style["Dimensionless distance,y", Large],
   Style["Dimensionless concentration,θ", Large]},
 LabelStyle → Directive[Bold, Black, 14],
 AxesStyle → Directive[Dashed] ]
```

Then, it can be seen that the algorithm for lithium-ion concentration profile in the solution phase during 1.75 mA/cm^2 discharged current at various value of dimensionless distance (simulation result 2)

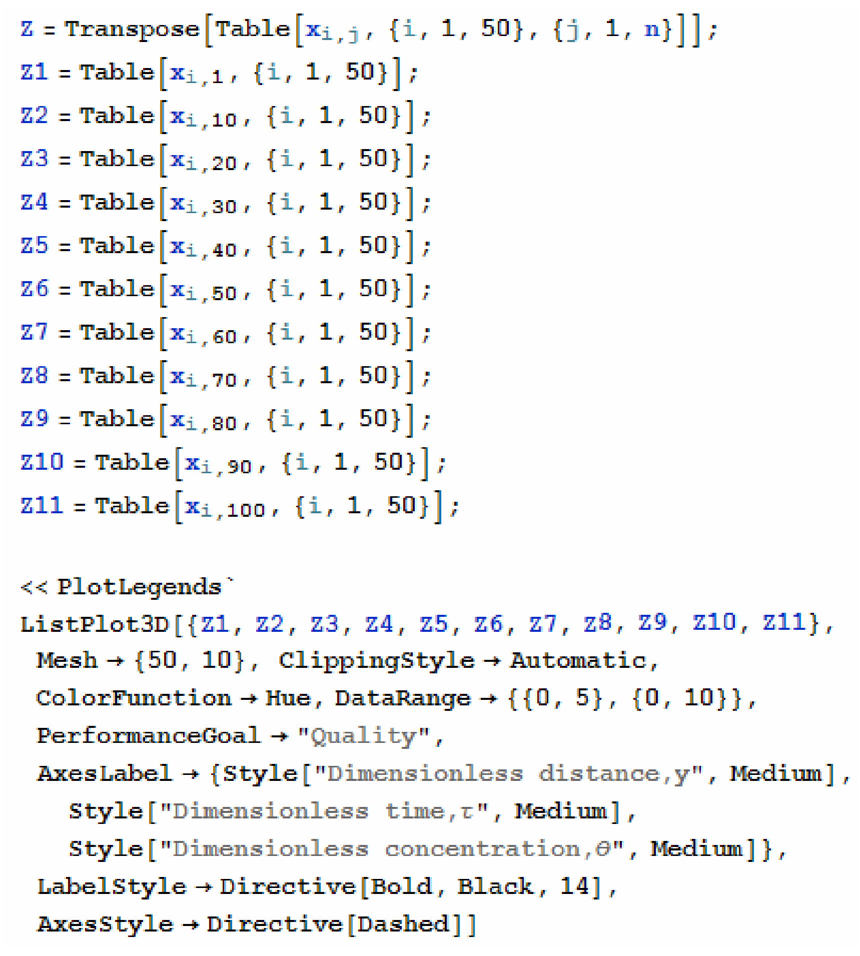
<u>STEP 5.2: Simulation result 2</u>



```
Z = Table[xi,j, {i, 1, 50}, {j, 1, n}];
<< PlotLegends`
ListLinePlot[Z, PlotRange → {0.7, 1.3}, DataRange → {0, 10},
 PlotStyle → {Blue, Yellow, Green, Red}, AxesOrigin → {0, 1},
 LabelStyle → Directive[Bold, Black, 14],
 AxesStyle → Directive[Dashed], Frame → True,
 FrameLabel → {Style["Dimensionless time,τ", Large],
   Style["Dimensionless concentration,θ", Large]}]
```

Furthermore, for lithium-ion concentration profile in the solution phase during 1.75 mA/cm^2 discharged current in 3D simulation (simulation result 3)

STEP 5.3: Simulation result 3



```
Z = Transpose[Table[x_{i,j}, {i, 1, 50}, {j, 1, n}]];
Z1 = Table[x_{i,1}, {i, 1, 50}];
Z2 = Table[x_{i,10}, {i, 1, 50}];
Z3 = Table[x_{i,20}, {i, 1, 50}];
Z4 = Table[x_{i,30}, {i, 1, 50}];
Z5 = Table[x_{i,40}, {i, 1, 50}];
Z6 = Table[x_{i,50}, {i, 1, 50}];
Z7 = Table[x_{i,60}, {i, 1, 50}];
Z8 = Table[x_{i,70}, {i, 1, 50}];
Z9 = Table[x_{i,80}, {i, 1, 50}];
Z10 = Table[x_{i,90}, {i, 1, 50}];
Z11 = Table[x_{i,100}, {i, 1, 50}];

<< PlotLegends`
ListPlot3D[{Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11},
 Mesh → {50, 10}, ClippingStyle → Automatic,
 ColorFunction → Hue, DataRange → {{0, 5}, {0, 10}},
 PerformanceGoal → "Quality",
 AxesLabel → {Style["Dimensionless distance,y", Medium],
   Style["Dimensionless time,τ", Medium],
   Style["Dimensionless concentration,θ", Medium]},
 LabelStyle → Directive[Bold, Black, 14],
 AxesStyle → Directive[Dashed]]
```

The algorithm that has been discussed above is the main algorithm to solve the systems of algebraic equation in chapter 3. By using this main algorithm, we can change the value of parameters, select the certain time level or select the certain position level to get other simulation result under various conditions.

Ali, S. A. H., A. Hussin, et al. (2002). "Short- and Long-time solutions for material balance equation in lithium-ion battery by laplace transform." Journal of Power Sources **112**(2): 435-442.

Ali, S. A. H., A. Hussin, et al. (2003). "An Analytical Expression for the Specific Capacity of Lithium Ion Cells." Ionics **9**: 432-438.

Ali, S. A. H., A. Hussin, et al. (2004). "An Integral Transform for Solving Diffusion Problem of Lithium Cells." MATEMATIKA **20**(2): 93-100.

Banks, B. W. (2001). Differential Equations with Graphical and Numerical Method. New Jersey,US, Prentice Hall.

Botte, G.G. (2000). Thermal Stability And Modeling of Lithium Ion Batteries. Chemical Engineering. South Carolina, University of South Carolina. **Phd thesis:** 324.

Botte, G. G., V. R. Subramanian, et al. (2000). "Mathematical modeling of secondary lithium batteries." Electrochimica Acta **45**: 2595-2609.

Botte, G. G. and R. E. White (2001). "Modeling Lithium Intercalation in a Porous Carbon Electrode " Journal of the Elecrtrochemical Society **148**(1): 54-66.

Cai, L. and R. E. White (2009). Mathematical Modeling of a Lithium Ion Battery. COMSOL Conference, Boston.

Ceder, G., M. Doyle, et al. (2002). Computational Modeling and Simulation for Rechargeable Batteries. www.mrs.org/publications/bulletin. M. R. Society**:** 619-623.

Chonacky, N. and D. Winch (2005). "3M's for Instructions: Review of Maple, Mathematica and Mathlab." Computing in Science and Engineering **7**(3): 7-13.

Doyle, M., T. F. Fuller, et al. (1993). "Modeling of Galvanostatic Charge and Discharge of the Lithium/Polymer/Insertion Cell." Journal of Electrochemistry Society **140**(6): 1526-1533.

Doyle, M. and J. Newman (1997). "Analysis of capacity-rate data for lithium batteries using simplified models of the discharge process " Journal of Electrochemistry **27**: 846-856.

Farlow, S. J. (1982). Partial Differential Equation for Scientists and Engineers. United States, John Wiley & Sons.

Golmon, S., K. Maute, et al. (2009). "Numerical modeling of electrochemical–mechanical interactions in lithium polymer batteries." Computers and Structures **87**: 1567-1579.

Hoffmann, K. A. and S. T. Chiang (2000). Computational Fluid Dynamics Kansas, Engineering Education System.

Holmes, M. H. (2007). Introduction to Numerical Methods in Differential Equations. New York,US, Springer.

Jain, M., G. Nagasubramanian, et al. (1999). "Analysis of a Lithium/Thionyl Chloride Battery under Moderate-Rate Discharge." Journal of The Electrochemical Society **146**(11): 4023-4030.

Johan, M. R. and A. K. Arof (2007). "Modeling of electrochemical intercalation of lithium into a LiMn2O4 electrode using Green function." Journal of Power Sources **170**: 490–494.

Landfors, J., D. Simonsson, et al. (1995). "Mathematical modelling of a lead/acid cell with immobilized electrolyte." Journal of Power Sources **55**: 217-230.

Lipsman, R. L., J. E. Osborn, et al. (2008). "The SCHOL Project at the University of Maryland: Using Mathematical Software in the Teaching of Sophomore Differential Equations." Journal of Numerical Analysis Industrial and Applied Mathematics **3**(1-2): 81-103.

Norzihani, Y., S. A. H. Ali, et al. (2010). Mathematical modeling of lithium-ion concentration in full cell by reinforcing the concept of Finite Difference Method. Applied Mathematics International Conference 2010 (AMIC2010) & The 6th EASIAM Conference, Kuala Lumpur.

Noye, J. (1981). Finite Difference Method for Partial Differential Equations. Numerical Solutions of Partial Differential Equations, Melbourne University,Australia, North Holland.

Sadiku, M. N. O. (2001). Numerical Techniques in Electromagnetics. New York,US, CRC Press LLC.

Smith, K. A., C. D. Rahn, et al. (2007). "Control oriented 1D electrochemical model of lithium ion battery." Energy Conversion and Management **48**: 2565-2578.

Stein, W. (2009). "Mathematical Software and Me: A Very Personal Recollection."

Steinhaus, S. (2008). "Comparison of mathematical programs for data analysis."

Subramanian, V. R., V. Boovaragavan, et al. (2007). "Toward Real-Time Simulation of Physics Based Lithium-Ion Battery Models." Electrochemical and Solid-State Letters **10**(11): 255-260.

Subramanian, V. R., V. Boovaragavan, et al. (2009). "Mathematical Model Reformulation for Lithium-Ion Battery Simulations: Galvanostatic Boundary Conditions." Journal of The Electrochemical Society **156**(4): 260-271.

Subramanian, V. R., D. Tapriyal, et al. (2004). "A Boundary Condition for Porous Electrodes." Electrochemical and Solid-State Letters **7**(9): 259-263.

Subramanian, V. R. and R. E. White (2001). "New separation of variables method for composite electrodes with galvanostatic boundary conditions." Journal of Power Sources **96**: 385-395.