

# Modeling Complex Physical Systems Using 2D Finite Element Cell-DEVS

Hesham Saadawi

Gabriel Wainer

Dept. of Systems and Computer Engineering  
Carleton University  
4456 Mackenzie Building  
1125 Colonel By Drive  
Ottawa, ON. K1S 5B6. Canada.  
[gwainer@sce.carleton.ca](mailto:gwainer@sce.carleton.ca)

**Keywords:** Cell-DEVS, DEVS, Finite Elements Method.

**Abstract:** *We propose a method to analyze complex physical systems (like engineering systems or structures) using two-dimensional Cell-DEVS models. These problems are usually modeled as one or more Partial Differential Equations and solved through numerical methods, and our goal is to improve the definition of such problems by mapping them into the Cell-DEVS formalism. We used two methods for solving PDEs, and deduced the discrete event updating rules for their mapping to Cell-DEVS. In our simulation results, the accuracy of the Cell-DEVS solution was same of these previous methods, showing that we can use Cell-DEVS as a tool to solve systems of equations resulting from approximating the solution of PDEs. Simultaneously, we are provided with a simpler mechanism for model definition, automated parallelism, and faster execution.*

## 1. INTRODUCTION

In recent years, there has been a trend in studying complex natural or man-made physical systems using modeling and simulation techniques. These problems were usually modeled with differential equations, and by traditional analysis, these equations were solved analytically, an approach that works only for very simple systems. Invention of digital computers permitted the use of advanced numerical methods. One of such methods, the **finite element analysis** [1] has been used successfully to analyze complex engineering and physical systems. The Finite Elements Method (FEM) defines a solution satisfying the partial differential equation on average over a finite element, which is connected to neighboring elements, and the field under study is analyzed by propagating the current values from one element to another through connection points.

We have explored the use of the Cell-DEVS formalism [2][3] to model and solve problems usually tackled by FEM. We intend to use FEM as a very precise technique for defining the problem, while having the simplicity of a cellular approach to facilitate model definition. The use of Cell-DEVS also enables integration with other existing DEVS [4] and Cell-DEVS models permitting to define multi-paradigm models. An advantage of this approach is that in real life systems are often composed of continuous and discrete components interacting together. This dictates the need to

integrate both models and simulate their global behavior. DEVS provides means for modeling discrete event systems, while Cell-DEVS enables modeling different continuous systems. Finally, in building these models as Cell-DEVS, we can make use of existing infrastructure, including parallel simulators and distributed environments. We describe a method for mapping problems modeled by partial differential equations and solved by finite differences or FEM, into a Cell-DEVS specification. We first describe the physical system modeled, and show how to obtain the cell update rules from an approximate solution.

## 2. BACKGROUND

FEM was originally created to solve problems of structural mechanics, and it was later applied to many other problems in engineering. FEM provides piecewise approximation of a partial differential equation over a continuum. A finite element is a discrete piece of that continuum. In FEM problems, we usually find two major components that can be identified within each element: *field*, and *potential*. The field is a quantity that varies with position within structure analyzed. The fields are related to the potentials as their derivatives with respect to position. The potential can be thought as the driving force for the spread of the field in the material. For example, temperature difference would cause a heat flux from one point to another through a material. The heat flux direction and quantity is related with the difference in temperature (temperature gradient).

FEM is a mathematical procedure for satisfying a partial differential equation over an element. If the field variable is following a simple function over the finite element, we can approximate the solution of the partial differential equation over that element. Elements in the structure are connected together through the vertices on boundaries of each element, which are called *nodes*. Solving the problem using FEM includes the following:

1. Divide the structure under study into a large number of simple geometry elements.
2. The spatial solution in that element would be represented with an assumed simple interpolation or shape function over the element.
3. The differential equations can be solved for that particular simple element using the assumed shape function of the change of potential in that element.

4. As all the elements in the structure are connected together through their nodes, we obtain a system of equations represented in a form of  $N \times N$  matrices for the whole structure (where  $N$  represents the number of elements in that structure). These values would be used to get the unknown potential inside the structure.
5. The global equations are solved using a suitable mathematical method. The final solution would give the distribution of the potential over the structure, represented by the values obtained at the nodes of each element.

Cell-DEVS is a novel approach to represent models of real systems as cell spaces. Cell-DEVS uses the DEVS (Discrete Events systems Specifications) formalism [4] to define a cell space where each cell is defined as a DEVS model. This technique permits to build discrete-event cell spaces, and it improves their definition by making the timing specification more expressive.

In Cell-DEVS, each cell of a cellular model is defined as an atomic DEVS model. Cell-DEVS atomic models use  $\mathbf{N}$  inputs to compute the future state  $\mathbf{S}$  using the function  $\tau$ . The new value of the cell is transmitted to the neighbors after the consumption of the delay function. **Delay** defines the kind of delay, and **d** its duration.

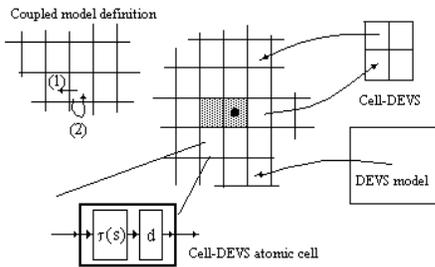


Figure 1: Informal Definition of Cell-DEVS [3].

Once the atomic cell model is defined, a number of cells can be put together to form a coupled model, built as an array of atomic cells. A Cell-DEVS coupled model is defined as an array of atomic cells with size  $\{t_1 \times \dots \times t_n\}$ . Each cell in the space is connected to the cells defined by the neighborhood  $\mathbf{N}$ . The border  $\mathbf{B}$  cells can use a different behavior than the remaining cells. The  $\mathbf{Z}$  function allows one to define the internal and external coupling of cells in the model.  $\mathbf{X}_{list}$  and  $\mathbf{Y}_{list}$  are input/output coupling lists can be used to interchange data with other models.

The **CD++** tool [5] was developed following the formal definitions of the Cell-DEVS formalism. CD++ is a tool to simulate both DEVS and Cell-DEVS models. DEVS models can be written in C++, and Cell-DEVS models are described using a built-in specification language. The language provides a set of primitives to define the size of the cell-space, the type of borders, a cell's interface with other DEVS models and a cell's behavior.

FEM models resembles to a large extent Cell-DEVS models, in which changes of a cell value would trigger its neighboring cells to

change themselves, as though a field is propagating through all of them. The following sections will be devoted to show how Cell-DEVS can be used to describe FEM models. The idea is to describe the model in terms of cell behavior, discrete event interaction and timing delays.

### 3. 2-D CELL-DEVS TRANSFER MODELS

In order to define a Cell-DEVS model, we need to describe every cell's local transition function ( $\tau$ ) as explained in the previous section. Let us consider the heat transfer model originally presented in [1]. This example represents a steady state 2-D heat transfer in a bar of rectangular cross section with thermal conductivity coefficient  $k = 1.5 \text{ W/m}^2 \cdot \text{°C}$ . Two opposite sides are kept at constant temperature of  $180^\circ \text{C}$ ; one side is insulated and the other is exposed to a fluid with temperature of  $25^\circ \text{C}$  and convection heat transfer coefficient  $h = 50 \text{ W/m}^2 \cdot \text{°C}$ . A graphical representation of the problem is depicted in Figure 2.

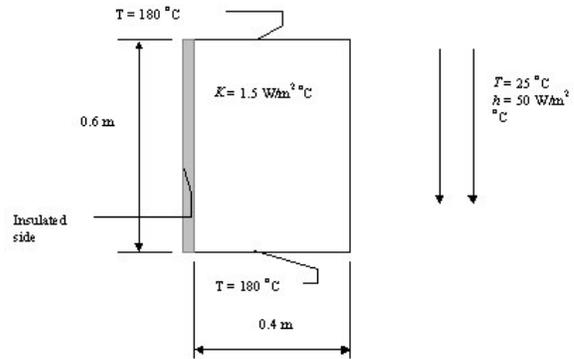


Figure 2: Steady-State heat transfer in a long bar.

In order to define this model using Cell-DEVS, we define a finite grid in the bar, and we map the equations for each node on the grid into a set of rules for updating the value of a corresponding cell at a Cell-DEVS model. We have deduced the Cell-DEVS updating rules by considering two methods for numerical solution: finite differences and finite elements. With the aim of deducing equations from Finite Differences, we wrote the general partial differential equations for a steady-state heat transfer through a material, and then we approximated these equations using finite differences, obtaining the updating rules for our Cell-DEVS model.

Heat transfer occurs when there is a temperature difference within a body or between a body and its surrounding medium. Heat flows from hot spots towards cooler ones. Heat conduction in two dimensions steady state isotropic medium is given by:

$$q_x = -k \frac{\partial T}{\partial x} \quad q_y = -k \frac{\partial T}{\partial y} \quad (1)$$

where  $q$  is the heat flux ( $\text{W/m}^2$ ),  $k$  is the thermal conductivity of the material ( $\text{W/m} \cdot \text{°C}$ ),  $T = T(x,y)$  is the Temperature field in the

medium, and  $\partial T/\partial x$ ,  $\partial T/\partial y$  are the temperature gradients over  $x$  and  $y$  respectively. The minus sign is to indicate that the direction of heat flux is opposite to direction of increasing temperature.

In convection heat transfer, the heat flux is given by:

$$q = h(T_\infty - T_s) \quad (2)$$

where  $h$  ( $W/m^2 \cdot ^\circ C$ ) is the film coefficient is a property of the fluid around the surface,  $T_\infty$  and  $T_s$  are fluid and surface temperature respectively.

Here we apply these equations to a small one-dimensional element, assuming a linear temperature distribution along its length, the resulting heat flux equation would be:

$q = k(T_h - T_l)$ , where  $T_h$  and  $T_l$  are the high and low temperatures of its ends respectively. A steady heat transfer without heat generation in the body in 2D is represented by the following diffusion equation [6]:

$$k \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = 0$$

To solve the above equation, we need to get the second derivative of the temperature gradient. To do so, we study a steady-state heat transfer in a long rod as represented in Figure 3. In this figure, we study a very small section of one-dimensional rod. Points A, B and C along the rod have corresponding temperatures of  $T_1$ ,  $T_2$ , and  $T_3$  respectively. Distances between points in the section are as indicated on the figure. To get the temperature gradient along small section, we assume a linear temperature change in  $x$  direction over the very small finite space  $\Delta x$ .

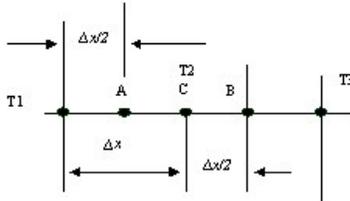


Figure 3: Heat transfer in one dimension.

$\frac{\partial T}{\partial x} = \left( \frac{T_1 - T_2}{\Delta x} \right)$  is the temperature gradient at point A

$\frac{\partial T}{\partial x} = \left( \frac{T_2 - T_3}{\Delta x} \right)$  Is the temperature gradient at point B. Thus, the

temperature gradient at point C is:

$$\frac{\partial^2 T}{\partial x^2} = \frac{\left[ \left( \frac{T_1 - T_2}{\Delta x} \right) - \left( \frac{T_2 - T_3}{\Delta x} \right) \right]}{\Delta x} = \frac{T_1 + T_3 - 2T_2}{\Delta x^2}$$

By applying the previous result in a two-dimensional space, we can approximate the solution of the previous PDE as:

$$k \left[ \left( \frac{T_1 + T_3 - 2T_0}{a^2} \right) + \left( \frac{T_2 + T_4 - 2T_0}{a^2} \right) \right] = 0$$

From which we obtain:  $T_1 + T_2 + T_3 + T_4 - 4T_0 = 0$

This equation relates nodes temperatures of the grid. Thus, it gives us the updating rule for an internal node as node 0 in to be:

$$T_0 = \frac{T_1 + T_2 + T_3 + T_4}{4} \quad \text{Updating Rule 1}$$

These results are for a steady-state heat transfer in a homogeneous medium with constant coefficient of thermal conductivity  $k$  in all directions. We still need the updating rules for a point on the insulated surface, or on the convective side of the rod. Using a similar procedure (described in detail in [7]), we have deduced the rules for a node on the insulated boundary and on the convective boundary as follows:

$$T_0 = \frac{T_1 + T_3 + 2T_2}{4} \quad \text{Updating Rule 2}$$

$$T_0 = \frac{T_1/2 + T_3/2 + T_2 + T_f \cdot h \cdot a / k}{2 + h \cdot a / k} \quad \text{Updating Rule 3}$$

We have defined a method to deduce Cell-DEVS model updating rules from Finite Elements model for 2-Dimensional problems and we applied them to a 2-D triangular finite element with equations of steady state heat transfer. We used a linear change function of the field under study over the triangular element. After getting the matrix equation for one element, we construct a mesh of elements connected together to apply equilibrium equations over those elements, and thus can deduce a node temperature as a function of its surrounding nodes. This function would be the updating rule for cells in our Cell-DEVS model that represents 2-D Finite Elements model of steady-state heat transfer. These updating rules would be used for internal nodes inside the bar. Similarly, we would need to construct another mesh to study boundary nodes on the convective side. We have used the following triangular element.

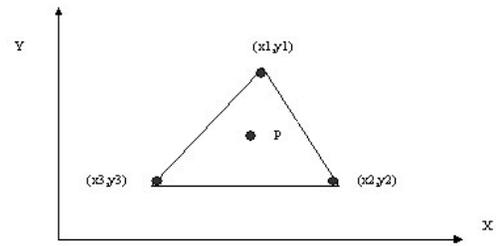


Figure 4. Triangular Element.

The element in Figure 4 has three nodes; each at a vertex of the triangle, and this would contain a value for the field under study, for example temperature values. This type of element (called Constant Strain Triangular) was used historically to analyze body strain problems. Any internal point in the element as P can be evaluated as a function of the three nodes values. We use here a simple triangular element that assumes a linear function over the element for the field under study. That is:  $T_p = N_1 T_1 + N_2 T_2 + N_3 T_3$ , where  $N_1, N_2, N_3$  are linear shape functions.

With further manipulations, we can get temperature gradient inside the element as

$$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} = B^e T^e$$

Where  $B^e = \frac{1}{2A} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{31} & x_{21} \end{bmatrix}$ ,  $T^e = \begin{bmatrix} T1 \\ T2 \\ T3 \end{bmatrix}$ ,  $y_{23} =$

$y_2 - y_3$ ,  $x_{32} = x_2 - x_3, \dots$ , and  $A$  is the area of the triangular element.

The final steady state equations for the element can be represented in a matrix format as:  $\underline{K} \underline{T} = \underline{R}$ , where  $\underline{K}$  is the stiffness matrix that contains the thermal coefficients of the element,  $\underline{T}$  is the nodes temperatures vector, and  $\underline{R}$  is the heat Rate vector at each node (the underline denotes a Matrix). For detailed manipulations, please refer to [1] and [10]. In order to get the updating rules for the Cell-DEVS, we construct a mesh of elements to represent a recurring pattern inside the structure under study. The mesh is shown in Figure 5.

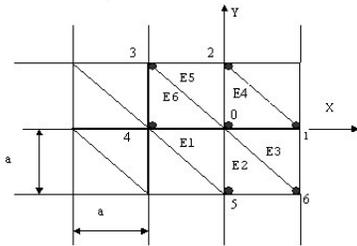


Figure 5. Internal mesh of triangular elements.

The figure shows a mesh composed of six elements marked as E1, E2, ..., E6. Those elements have their nodes numbered as shown from zero to six. The middle node 0 is shared among all the elements, thus its value would be a function in all other elements. By studying this structure, we would be able to deduce the updating rules for node 0, which would then repeat again for all similar internal nodes. Doing a similar mathematical manipulation of these equations (as explained in [7]), we were able to find the equations corresponding to the local rules of our Cell-DEVS models:

$$T_0 = \frac{T5 + T1 + T2 + T4}{4}$$

the finite differences method for an internal node. In order to get the updating rules for boundary conditions, we would need to construct the mesh in the same way at the insulated and convective boundaries. The mesh would then be studied using the same procedure.

#### 4. CELL-DEVS IMPLEMENTATION

In this section, we will present the definition of some of the previous examples as Cell-DEVS models in the CD++ environment. We start by solving the problem introduced in Figure 2. To do so, we divide the bar as shown in Figure 6 into a grid of 6 rows X 4 columns that contains 7 X 5 nodes. Nodes are located at every lines intersection in the grid. We use the updating rules obtained

previously to model this grid with Cell-DEVS in which, each cell in the model would represent a node on the grid

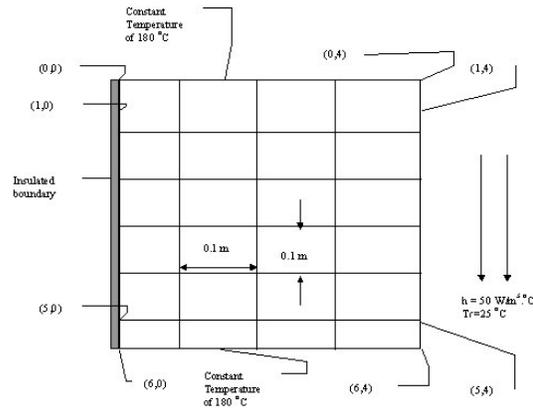


Figure 6: 7 X 5 nodes Grid of Finite Differences.

Figure 7 represents the model definition in CD++, considering that cells on constant temperature boundaries are initialized with value of 180 °C.

```
[heatcond]
type : cell          dim : (7,5)
delay : transport    border : nowraped
neighbors : (-1,0) (0,0) (0,1) (1,0) (0,-1)
localtransition : conduction-rule
zone : Insulated-Boundary { (1,0)..(5,0) }
zone : Constant-Temp { (0,0)..(0,4) }
zone : Constant-Temp { (6,0)..(6,4) }
zone : Convective { (1,4)..(5,4) }

[conduction-rule]
rule : { ((0,1)+(-1,0)+(0,-1)+(1,0)) / 4 } 1 {t}

[Insulated-Boundary]
rule : { ((-1,0)+(1,0)+2*(0,1)) / 4 } 1 {t}

[Constant-Temp]
rule : { (0,0) } 1 {t}

[Convective]
rule : { ( ((1,0)/2)+((-1,0)/2)+(0,-1)+ (25 * (10/3)) ) / (2 + (10/3)) } 1 {t}
% Fluid temperature: 25, and h.a/k is (10/3)
```

Figure 7: Model definition in CD++ for 7 X 5 nodes grid.

In the figure, heatcond is a two-dimensional Cell-DEVS space with 7 rows and 5 columns using transport delays. The border is nowraped, hence, border cells are not connected to the opposite side border cells. Then, we define the neighborhood shape for each cell. Localtransition specifies the name used for the local computing function, and it is defined as a set of updating rules for each cell depending on its location on the grid. By default, each cell would follow the conduction-rule, unless it

is located at one of the defined *zones*, which define the border conditions. Therefore, [conduction-rule] uses updating rule 1 defined above. Likewise, [Insulated-Boundary] uses updating rule 2, and [Convective] uses updating rule 3. CD++ provides many built-in functions and constructs. A more detailed description of these facilities can be found at [3], [7] and [9].

```

Line : 91 - Time: 00:00:00:000
      0   1   2   3   4
+-----+
0| 180.0 180.0 180.0 180.0 180.0 |
1|  20.0  20.0  20.0  20.0  20.0 |
2|  20.0  20.0  20.0  20.0  20.0 |
3|  20.0  20.0  20.0  20.0  20.0 |
4|  20.0  20.0  20.0  20.0  20.0 |
5|  20.0  20.0  20.0  20.0  20.0 |
6| 180.0 180.0 180.0 180.0 180.0 |
+-----+
. . .
Line : 29358 - Time: 00:00:00:204
      0   1   2   3   4
+-----+
0| 180.0 180.0 180.0 180.0 180.0 |
1| 158.4 155.3 144.1 118.7  58.7 |
2| 143.3 138.4 122.6  92.0  42.1 |
3| 137.9 132.6 115.6  84.8  39.4 |
4| 143.3 138.5 122.6  92.0  42.1 |
5| 158.4 155.3 144.1 118.7  58.7 |
6| 180.0 180.0 180.0 180.0 180.0 |
+-----+

```

Figure 8: Results for 7 X 5 nodes grid of finite differences.

When we executed this model in CD++, we have obtained the results shown in Figure 8. In Figure 10, we show CD++ outputs on grid like the one in Figure 6 for easy reading. The results shown in Figure 8 include the simulation time and cell's values at each time step. In this output file, each cell value is printed at its corresponding row and column. Thus, at the start of the simulation, cells (0,0) to (0,4) and (6,0) to (6,4) all had a value of 180, and the rest of the cells have a value of 20. At the final two steps, cell values do not change, causing the simulation to end (there is no cell that would be triggered by a changing neighbor). At this step, cell values would represent the solution of the problem, namely, the temperature distribution through the bar.

The type of diffusion problems as defined with previous PDE is practically solved using numerical methods such as FEM or finite differences for the reason that exact analytical solutions can only be obtained for simple geometries and boundary conditions. Such an analytical exact solution would be an ideal choice to compare precision of results obtained from CD++. For the difficulty of obtaining exact solution for this problem, we have compared results obtained from CD++ solution with those obtained by FEM. We have solved the problem with FEM to show convergence of its results with those of our Cell-DEVS method. In first solution, we divided the bar into six elements (we show the execution results in Figure 9). The heat diffusion problem as defined by the

PDE in section 3.1 has been solved to get closed form solution, only in very simple cases and shapes. Therefore, we used a proven software tool to solve the same problem, and used the results to validate our simulation. To do this, we used the FEM software included in [1]. We divided the bar into six triangular Finite Elements as shown in Figure 9. Execution results are shown besides each node.

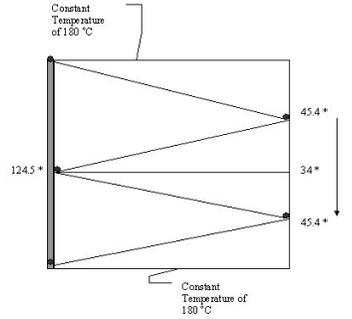


Figure 9: Results for a mesh of 6 Triangular Finite elements.

In order to enhance the precision of our validation results, and to capture the steep temperature gradient over the convective side, we solved the same problem using more Finite elements. Using the sample program provided with [1], the same example was solved by using 192 triangular elements with 117 distinct nodes.

180	180	180	180	180	180	180	180	180	180
169	168.6	167.3	164.8	160.8	154.3	143.7	118.7	68.1	
<b>158.9</b>	158	<b>155.6</b>	151.2	<b>144</b>	132.5	<b>122</b>	94.9	<b>52 *</b>	
150.3	149.3	146	140.3	131.6	119.2	103.2	79	45.1	
<b>143.9</b>	142.6	<b>138.9</b>	132.4	<b>122.7</b>	109.6	<b>92.9</b>	72.5	<b>37.3 *</b>	
139.9	138.5	134.5	127.6	117.4	103.7	86	63.3	42.6	
<b>138.5</b>	137.1	<b>133</b>	126	<b>115.7</b>	101.8	<b>84.1</b>	62.6	<b>39.6 *</b>	

Table 1. Tabular form of simulation results for FEM mesh.

The result of this simulation is shown in Table 1 for half of the bar (as it is symmetric around its middle horizontal line). Every cell in the table contains the corresponding node temperature.

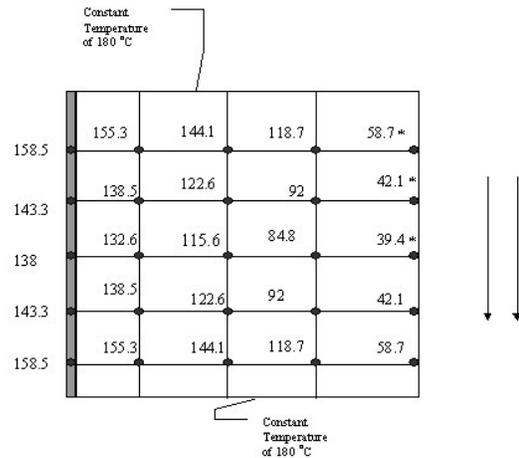


Figure 10: 7 X 5 nodes grid using Finite Differences.

To compare the results shown in Figure 10 and those in Table 1 we marked corresponding rows of nodes with an asterisk (\*). Corresponding cells in the table to those in the figure are further written in bold font. Temperature values from Table 1 and Figure 10 match on most nodes; nevertheless, there is a slight deviation between the two results for nodes on the convective side of the bar. As this side has the steepest temperature gradient, we expect that more elements are needed to capture details of this steep gradient.

To enhance Cell-DEVS Finite Differences results, and to verify the conversion of our Cell-DEVS simulation with that of FEM software, we ran another simulation with Cell-DEVS increasing the number of nodes to 13 X 9. The results of this simulation are shown in Table 2. The corresponding cells (representing node temperatures) are marked with bold font for comparing the results.

180	180	180	180	180	180	180	180	180
169	168.6	167.3	165	161.1	154.9	144.1	123.5	76.9
<b>158.9</b>	158.1	<b>155.7</b>	151.4	<b>144.6</b>	134.1	<b>118.2</b>	93.0	<b>53.4*</b>
150.3	149.3	146.0	140.3	131.6	119	101.4	77.1	45.4
<b>143.8</b>	142.6	<b>138.8</b>	132.3	<b>122.5</b>	109.0	<b>91.2</b>	68.6	<b>41.7*</b>
139.8	138.5	134.4	127.4	117.2	103.4	85.7	64.3	40
<b>138.5</b>	137.1	<b>132.9</b>	125.8	<b>115.4</b>	101.5	<b>84</b>	63	<b>39.5*</b>
139.8	138.5	134.4	127.4	117.2	103.4	85.7	64.3	40
143.8	142.6	138.8	132.3	122.5	109.0	91.2	68.6	41.7
150.3	149.3	146.0	140.3	131.6	119	101.4	77.1	45.4
158.9	158.1	155.7	151.4	144.6	134.1	118.2	93.0	53.4
169	168.6	167.3	165	161.1	154.9	144.1	123.5	76.9
180	180	180	180	180	180	180	180	180

**Table 2. 96 Cell-DEVS finite differences cells with 117 nodes.**

The difference in values between FEM software results, our Cell-DEVS Finite Differences results in Table 1 and those in Table 2 on the convection side comes from the fact that we used constant temperature Finite triangular elements, which are a rough estimate for the varying field. These elements are good to study a slow varying field over the material. Other types of elements can capture the steep variation more precisely.

## 5. CONCLUSION

We showed how to use the Cell-DEVS formalism to model problems that traditionally have been solved with other methods as Finite Elements and Finite Differences. We showed that modeling physical and engineering problems with Cell-DEVS have some advantages over traditional methods.

- Cell-DEVS enables building coupled systems that are composed of many atomic or coupled components. These components can simulate discrete or continuous systems. This property can be very useful when simulating engineering systems as they often compose of many continuous and discrete components.
- Simulation of large and complex models can use the ability

of Cell-DEVS models to execute in parallel. This can be advantageous over other methods of solving continuous systems as FEM, as their parallel execution may not be straightforward. Parallel execution of Cell-DEVS does not need to be specified by the user, but it comes as a result of the asynchronous nature of Cell-DEVS technique.

- Advantages in modeling and improved development facilities. Definition of complex equations can be easily done using the rule specification techniques of CD++, as it could be seen with the examples presented here. This highly reduces the effort spent by the users in developing the applications. Automated verification facilities in the toolkit improve testing and reduce delivery time. Likewise, applying changes to the model only results in slight changes in the model specifications, without any need for code rewriting, which enables easy analysis of more complex system conditions.

Engineering disciplines could benefit from using Cell-DEVS formalism to build reusable components that can be used as building blocks to build models for their products. Those models can then be executed to study system behavior. One or more components in the model can then be replaced with other alternative designs to reach a desired optimum system behavior.

## REFERENCES

- [1] CHANDRUPATLA, T.; BELEGUNDU; A. "Introduction to finite elements in engineering", *Prentice Hall*, 1997.
- [2] WAINER, G.; GIAMBIASI, N. "N-Dimensional Cell-DEVS". In *Discrete Events Systems: Theory and Applications*, Kluwer. Vol. 12, No. 1. January 2002. pp. 135-157.
- [3] WAINER, G.; GIAMBIASI, N. "Application of the Cell-DEVS paradigm for cell spaces modelling and simulation". G. Wainer, N. Giambiasi. *Simulation*, Vol. 71, No. 1. January 2001. pp. 22-39.
- [4] ZEIGLER, B.; KIM, T.; PRAEHOFER, H. "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems". Academic Press. 2000.
- [5] WAINER, G. "CD++: a toolkit to define discrete-event models". G. Wainer. *Software, Practice and Experience*. Wiley. Vol. 32, No.3. pp. 1261-1306. November 2002.
- [6] BACON, D. H. "Basic heat transfer", Butterworth & Co. (Publishers) Ltd. 1989.
- [7] SAADAWI, H.; WAINER, G. "Defining models of complex 2D physical systems using Cell-DEVS". Technical Report SCE-04-04. Carleton University. 2004.
- [8] RODRIGUEZ, D.; AND WAINER, G. 1999. New Extensions to the CD++ tool. In *Proceedings of SCS Summer Computer Simulation Conference*. Chicago, IL. 1-6.
- [9] WAINER, G.; GIAMBIASI, N. "Timed Cell-DEVS: modeling and simulation of cell spaces ". In *Discrete Event Modeling & Simulation: Enabling Future Technologies*. Springer-Verlag. 2001.
- [10] Yijun Liu, "Lecture notes: Introduction to the Finite Element Method", [http://urbana.mie.uc.edu/yliu/FEM-525/FEM\\_Lecture\\_Notes\\_Liu\\_UC.pdf](http://urbana.mie.uc.edu/yliu/FEM-525/FEM_Lecture_Notes_Liu_UC.pdf). Accessed on April 2003.