Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 6 – No.7, February 2017 – www.caeaccess.org

Parallelization with MPI on 1-D Active Thermal Control Problem

Simon Uzezi Ewedafe
Department of Computing
The University of the West Indies,
Mona Kingston 7, Jamaica

Rio Hirowati Shariffudin
Institute of Mathematical Sciences
University Malaya
Kuala Lumpur, Malaysia

ABSTRACT
This paper describes a 1-D Active Thermal Control Problem (1-D ATCP) using parallel domain decomposition with parallel communication approach of Message Passing Interface (MPI) on Stationary Iterative Techniques (Jacobi and Gauss-Seidel. The discretization of the resulted matrices from the 1-D ATCP shows the ease of parallel implementation across the different domain of block sizes. The Single Program Multiple Data (SPMD) techniques was used to describe the parallelization of the method. We implemented the parallelization on Geo Cluster having to exploit inherent parallelism. Hence, the parallelization strategies are discussed, and results of the parallel experiments are presented with the speedup and efficiency on different block sizes agree with the theoretical analysis.

General Terms
Parallel Algorithms, Parallelization

Keywords
1-D ATCP, Stationary Techniques, SPMD, DD, MPI

1. INTRODUCTION
Cluster applications have more processor cores to manage and exploit the computational capacity of high-end machines providing effective and efficient means of parallelism even as the challenges of providing resource management grows. It is a known fact that high capacity computing platform is expensive, and characterized by long-running, high processor jobs. The performance of message-passing programs depends on the parallel target machine, and the parallel programming model to be applied to achieve parallelism. In cluster machines having a large number of processing units, scalability becomes an important issue. Many programs from scientific computing have large potential for parallelism that is exploited best in parallelism in the form of concurrent multi-processor tasks [13]. The focus is on high performance cluster architecture, for which a fast active message-passing layer to a low latency, high bandwidth network is provided [14]. Applications are sensitive to overheads; such that execution time can increase with the increase in overheads [1]. Developments in computer architectures have shown that parallel computer systems are a viable way to overcome hardware design challenges relating to energy consumption, and thermal constraints, while offering high computational peak performance [10]. Therefore, global task with processor data is passed from processor to processor by message passing [12,8]. 1-D ATCP is decomposed into communicating parallel objects, with opportunity for overlap of communication with computation [3]. In the loosely-coupled system, the assembly of parallel nodes only requires interfaces, not the non-overlapped time needed in the SPMD techniques [11]. In case of data for concise expression of the algorithm, the code communication of the data is the message-driven style, and this can dominate the structure of the program, and overwhelm the programmer as in [11]. [17] has discussed the finite difference domain decomposition design and analysis for solving 2-D Heat Equation, and the parallelization for 1-D ATCP on a parallel virtual machine with DD in [11] show the load scheduling of various mesh sizes that produce the expected speedups.

This paper examined the temperature of the semiconductor devices as part of thermal control systems that treats the sequential algorithm of Parabolic Equation in solving thermal control process on printed circuit board [16]. The temperature control of the microprocessor devices is important to determine device performance [6]. The device specifies a temperature and allows some kind of deviations. The test process applies computer-controlled electrical signals to the device and for high-powered devices, this may result in die-average power density in the range of $100KW/m^3$. Microprocessor devices are also subjected to a test to determine the speed of the device. During this test, the goal is to keep the temperature of the die at a temperature where the device power is varied between 0% to 100% [15]. Thermal management testing equipment was proposed by [6]. However, [15] gave the analysis conduction time lag for sinusoidal die power. Minimization of the required laser power, is of importance to limit both the electrical power of the system and the load on the test facility's cooling system. Hence, [16] extended the analysis of [15] to multi-frequency wave form, with the aid of determining optimal control power for multi-frequency test power sequences. They show that the profile control calculation with specified die temperature tolerance in [16] is not suitable for non-sinusoidal die power profiles, and they develop a new approach for the situation. All the high-performance electronic devices are subjected to a 100% functional test before to being shipped by the manufacturers [15]. In this paper, we analyze and solved the 1-D ATCP by using stationary iterative techniques on a Geo cluster with MPI. The SPMD model is employed for the communication and computation that resulted in significantly improved speedup, effectiveness and efficiency across varying mesh sizes as compared to [4]. Our results demonstrated the overlap communication with computation, and the ability to arbitrary use of varying mesh sizes distribution on Geo cluster to reduce memory pressure while preserving parallel efficiency. On the other hand, the advantage of our platform is to have somewhat specification mechanism through a static distribution, and an execution implementation.

The results generated in our parallelization improve the understanding of the stability of ATCP. It also provides useful

stationary techniques for evaluating thermal control problems even to higher dimensional problems.

The paper is organized as follows. Section 2 presents the model problem. Section 3 gives the implementation of the parallel design and stationary schemes. Section 4 presents the results and discussion. Section 5 gives the conclusion.

2. THE MODEL PROBLEM

We reference to [15], the model presented is of the form:

$$\frac{\partial^2 v(x,t)}{\partial x^2} = \frac{1}{(b_t)}\frac{\partial v(x,t)}{\partial t} \qquad (2.1)$$

Where b_t is the thermal diffusivity and $b_t = \alpha = \frac{k}{\rho c_p}$,

and k is the thermal conductivity $(W/(m.k))$, ρ is the density (kg/m^3), and c_ρ is the specific heat capacity $(J/(kgk))$ while ρc_p together can be considered the volumetric heat capacity $(J/(m^3 k))$. Hence,

$$b_t \frac{\partial^2 v(x,t)}{\partial x^2} = \frac{\partial v(x,t)}{\partial t}$$
$$\text{let } v(x,t) = V, \text{then we have} \qquad (2.2)$$
$$b_t \frac{\partial V}{\partial x^2} = \frac{\partial V}{\partial t}$$

we can then solve (2.2) by extending the 1-D explicit finite difference method to the above, Eq. (2.2) becomes:

$$\frac{\partial V}{\partial t} = V_t = \frac{V_{i,j+1} - V_{i,j}}{\Delta t}$$
$$\frac{\partial^2 V}{\partial x^2} = V_{xx} = \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{(\Delta x)^2} \qquad (2.3)$$

applying Eq. (2.3) on Eq. (2.2), hence, the temperature is of the form:

$$\frac{V_{i,j+1} - V_{i,j}}{\Delta t} = b_t\left(\frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{(\Delta x)^2}\right)$$

rearranging we have:

$$V_{i,j+1} = \frac{bt\Delta t}{(\Delta x)^2}\left(V_{i+1,j} - V_{i-1,j}\right) + \left(1 - 2b_t\frac{\Delta t}{(\Delta x)^2}\right)V_{i,j} \qquad (2.4)$$

an implicit scheme of the Crank-Nicolson (C-N) unconditionally stable scheme can be applied to Eq. (2.2) as stated below.

A. Use of C-N Implicit Scheme

Applying the C-N implicit scheme, we have:

$$\frac{V_{i,j+1} - V_{i,j}}{\Delta t} = \frac{b_t}{2}\left(\frac{\begin{matrix}V_{i+1,j+1} - 2V_{i,j+1} + V_{i-1,j+1} \\ + V_{i+1,j} - 2V_{i,j} + V_{i-1,j}\end{matrix}}{(\Delta x)^2}\right) \qquad (2.5)$$

and rearranging to give:

$$-b_t\frac{\Delta t}{(\Delta x)^2}V_{i-1,j+1} + \left(2 + b_t\frac{\Delta t}{(\Delta x)^2}.2\right)V_{i,j+1}$$
$$-b_t\frac{\Delta t}{(\Delta x)^2}V_{i+1,j+1} = b_t\frac{\Delta t}{(\Delta x)^2}V_{i-1,j} + \qquad (2.6)$$
$$\left(2 - b_t\frac{\Delta t}{(\Delta x)^2}.2\right)V_{i,j} + b_t\frac{\Delta t}{(\Delta x)^2}V_{i+1,j}$$

let $b_t\frac{\Delta t}{(\Delta x)^2} = r$, then we have:

$$-rV_{i-1,j+1} + (2+2r)V_{i,j+1} - rV_{i+1,j+1} =$$
$$rV_{i-1,j} + (2-2r)V_{i,j} + rV_{i+1,j} \qquad (2.7)$$

here, we have a system of equations that can be solved.

B. Stationary Techniques on 1-D ATCP

Employing the notation

$$Au = b \qquad (2.8)$$

to represent such systems and the focus of this section is the study of methods for efficiently solving equation (2.8) on parallel computers. We begin with the decomposition

$$A = D - E - F, \qquad (2.9)$$

in which D is the diagonal of A, $-E$ is the strict lower part and $-F$ is the strict upper part. The Jacobi iteration determines the ith component of the next approximation.

Thus,

$$(b - Ax_{k+1}) = 0 \qquad (2.10)$$

However, Eq. (2.8) can be expressed in the form:

$$U^{(n+1)} = GU^{(n)} + K \qquad (2.11)$$

where n is an iterative counter and G is the iteration matrix related to the system matrix A below

$$G = I - Q^{-1}A$$

where I is the identity matrix and Q is generally called the splitting matrix. The Jacobi scheme can be constructed as follows. Firstly, decompose A as in Eq. (2.9), substitute into (2.8) to obtain:

$$(D - L - U)U = b, \text{ or } DU = (L+U)U + b \qquad (2.12)$$

hence, introducing iteration counter, (3.12) becomes

$$U^{(n+1)} = D^{-1}(L+U)U^n + D^{-1}b \qquad (2.13)$$

from Eq. (2.13) $L + U = D - A$, so

$$D^{-1}(L+U) = I - D^{-1}A.$$

Thus, D is the splitting matrix and Eq. (2.13) is in the form (2.11) with

$$G \equiv D^{-1}(L+U) = 1 - D^{-1}A,\ k = D^{-1}b \quad (2.14)$$

hence, in matrix terms Jacobi method can be expressed as

$$X^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b$$

as in Eq. (2.13)
where

$$x_i = \frac{1}{a_{i,j}}(b_i - \sum_{j \neq i} a_{i,j} x_j) \quad (2.15)$$

suggesting an iterative method defined by

$$x_i^{(k+1)} = \frac{1}{a_{i,j}}(b_i - \sum_{j \neq i} a_{i,j} x_j^k) \quad (2.16)$$

consider again the linear equations; proceeding with the Jacobi method one at a time in sequence, and that previously computed results are used as they are available we obtain the Gauss-Seidel (GS):

$$X_i^{(k)} = \frac{1}{a_{i,j}}\left[b_i - \sum_{j<i} a_{i,j} x_j^{(k)} - \sum_{j>i} a_{i,j} x_j^{(k-1)} \right] \quad (2.17)$$

the computation appears to be serial because the new iterate depends upon all previously computed components. Updating cannot be done simultaneously as in the Jacobi scheme. Secondly, the new iterate $x^{(k)}$ depends on the examined equations.

3. PARALLEL IMPLEMENTATION, DESIGN AND ANALYSIS

3.1 The Parallel Design

We implemented the parallelization on Geo Cluster consisting of 16 Intel Celeron CPU J 1900 at 1.99GHz quad core, and 495.7GB of Disk type. MPI [7]. We wrote the program in C and provides access to MPI. The parallelization contains more computations on different mesh sizes, and the environment concerns resource assessment with code placement on the cluster [2].

3.2 Domain Decomposition

Parallelization of the schemes was by means of grid partitioning technique, with computing domain decomposed into blocks with reasonable geometries. We introduced the block interfaces, auxiliary control volumes with boundary values of the neighboring block to overlap neighboring blocks at the boundary. Domain is splinted with each block given an I-D number by a "master" task, which assigns these sub-domains to "slave" tasks running on individual processors. To couple the sub-domains' calculations, boundary data are interchanged after each iteration. The calculation uses old values at the sub-domains' boundaries as boundary conditions. The DD is used to distribute data across the parallel processors while computation is done simultaneously. System partitioning and load balancing is done accordingly when the program is executed. [9] emphasized that data parallelism originated the SPMD with the same computation performed for multiple data sets, and are parts of the overall grid.

3.3 Parallelization with MPI

Domain partitioning used on the 1-D ATCP with stationary techniques on Geo cluster need divisions into sub-domains with no unique way of partitioning the domain of computation. The case of making a balance between the implementation of the techniques and communication is to balance. The processors in parallel work only on a portion of the grid and when the processor needs information from the nearest neighbor a message is passed through the MPI message passing library. For the best parallel performance, one would like to have optimal load balancing and as little communication between processors as possible. Considering load balancing first, one would like each processor to do the same amount of work. Hence, each processor is not idle. For the basic computational element; it makes sense to partition the grid such that each processor gets an equal number of nodes to work on. However, the amount of communication between processors are made as small as possible and to reduce communication. Program is divided into domains in such a way that minimizes the length of the touching faces in the different sub-domains. The number of processors that one processor has to communicate with also contributes to additional communication time, because of the latency penalty for starting the new message. At first step, we divide the spatial computational domain to $P = P_1 \times P_2$.

3.4 Speedup and Efficiency

Speed-up and efficiency are commonly used to measure the performance of parallel code. Runtime of the original serial code is used as a measure of the runtime on one processor. Runtime in this context can be defined as the time that has elapsed from the first moment when the first processor begins execution of the program to the time when the last process executes its statement. Time is measured after the initialization of MPI, and before the domain decomposition. Parallel Speed-up (Sp) is the ratio of the runtime on one processor t1 to the runtime on P processor tp.

$$Sp = \frac{t1}{tp} \quad (3.1)$$

The parallel efficiency (Ep) is the ratio of the parallel Speed-up Sp to the number of processors P.

$$Ep = \frac{Sp}{p}, \quad (3.2)$$

another aspect for the performance is the communication time.

4. RESULTS AND DISCUSSION

Consider following below:

$$\frac{\partial^2 v(x,t)}{\partial x^2} = \frac{1}{(b_t)}\frac{\partial v(x,t)}{\partial t} \quad (4.1)$$

With I.C and B.C as stated below:

$$\begin{aligned} &V(0,t) = h(t), \\ &V(1,t) = k(t), \qquad 0 \le t \le T \\ &\frac{\partial V}{\partial x}(x,0) = f(t), \quad 0 \le t \le 1 \\ &V(x,0) = f(x)\ (0 \le x \le 1) \end{aligned} \quad (4.2)$$

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 6 – No.7, February 2017 – www.caeaccess.org

4.1 Parallel Efficiency

The speedup and efficiency obtained for various sizes, for 100 mesh size to 300 mesh size, are listed on Tables I to III. The Tables presented the recorded parallel timing for the processors. The speedup and efficiency are given as shown in Fig. 1 and Fig. 2, respectively. The results show that the parallel efficiency increases with increasing mesh sizes. As the number of processors increase, with increase execution time, at a time the processors will not have much impact on total execution time. However, when the number of processors increase, balancing the number of computational cells per processors will become a difficult task due to significant load imbalance. The performance begins to degrade with an effect caused by the increase in communication overhead as the mesh increases. The gain in increasing execution time for mess sizes is due to uneven distribution of the computational cell, and the execution time has a small change due to DD influence on performance in parallel computation.

TABLE I. The total time T, the parallel speed-up S_p and the efficiency E_p for a mesh of 100

Scheme	*N*	S_p	E_p
	1	*1.000*	*1.000*
	2	*0.953*	*0.477*
	6	*1.048*	*0.175*
Jacobi	*8*	*1.296*	*0.162*
	10	*1.564*	*0.156*
	12	*1.831*	*0.153*
	14	*2.094*	*0.150*
	16	*2.313*	*0.145*
	1	*1.000*	*1.000*
	2	*1.108*	*0.554*
	6	*1.216*	*0.203*
GS	*8*	*1.397*	*0.175*
	10	*1.645*	*0.165*
	12	*2.109*	*0.176*
	14	*2.271*	*0.162*
	16	*2.526*	*0.158*

Table II. The total time T, the parallel speed-up S_p and the efficiency E_p for a mesh of 200

Scheme	*N*	S_p	E_p
	1	*1.000*	*1.000*
	2	*1.074*	*0.537*
	6	*1.265*	*0.211*
Jacobi	*8*	*1.532*	*0.192*
	10	*1.848*	*0.185*
	12	*2.226*	*0.186*
	14	*2.584*	*0.185*
	16	*2.928*	*0.183*
	1	*1.000*	*1.000*
	2	*1.205*	*0.603*
	6	*1.328*	*0.221*
GS	*8*	*1.612*	*0.202*
	10	*1.924*	*0.192*
	12	*2.497*	*0.208*
	14	*2.703*	*0.193*
	16	*3.034*	*0.190*

TABLE III. The total time T, the parallel speed-up S_p and the efficiency E_p for a mesh of 300

Scheme	*N*	S_p	E_p
	1	*1.000*	*1.000*
	2	*1.348*	*0.674*
	6	*1.665*	*0.278*
Jacobi	*8*	*2.038*	*0.255*
	10	*2.606*	*0.261*
	12	*3.115*	*0.260*
	14	*3.665*	*0.262*
	16	*4.190*	*0.262*
	1	*1.000*	*1.000*
	2	*1.441*	*0.721*
	6	*1.813*	*0.302*
GS	*8*	*2.176*	*0.272*
	10	*2.747*	*0.275*
	12	*3.419*	*0.285*
	14	*3.857*	*0.276*
	16	*4.317*	*0.270*

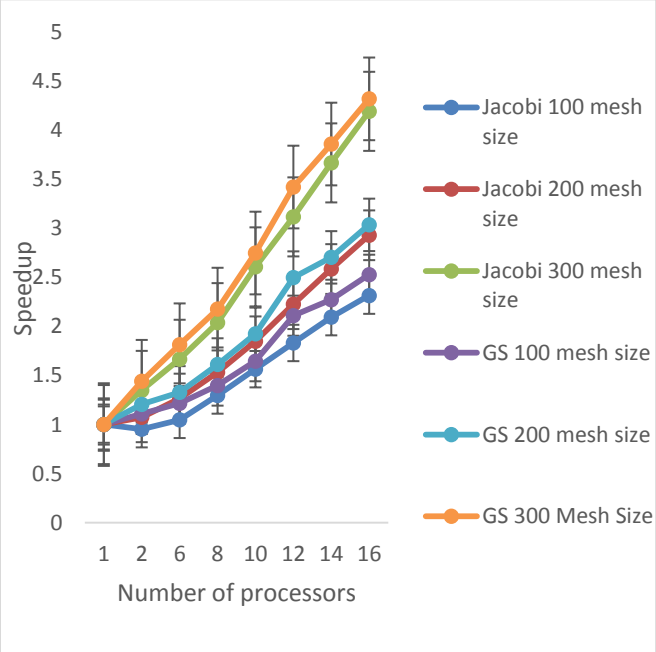

Fig. 1. Sp versus the number of processors for various mesh sizes

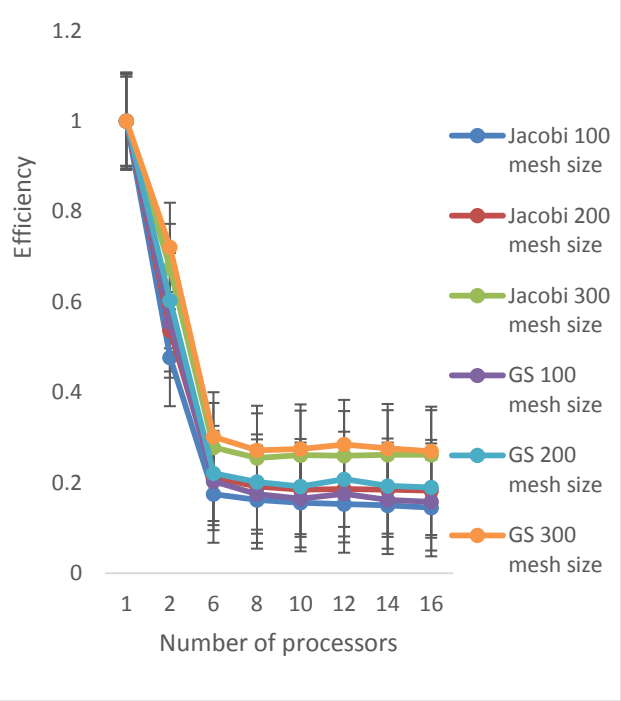

Fig. 2. Ep versus the number of processors for various mesh sizes

The numerical efficiency includes the DD efficiency and convergence rate behavior. System performance is dependent on the load of the nodes and thus allow overlapping of computation with communication. Hence, when the domain is partitioned across the nodes, the execution time and Sp are not noticeable, with larger mesh sizes the parallelization become noticeable. The results presented on the Tables and Figures show that MPI is a good message passing tool.

5. CONCLUSION

Presented results show the parallelization with MPI for 1-D ATCP using the SPMD techniques. The objective is to use parallel domain decomposition method on the stationary techniques on Geo cluster. The system allows for overlapping communication and shows the systematic methodology for investigating parallel performance of the temperature distribution. Our results demonstrated flexibility in parallel execution with ease of implementation. In addition to the use of ease of the parallel platform, there are overheads with load scheduling over various mesh sizes which produce the expected inherent speedups. The implementation confirmed that flexible scheduling for the overlapping communication is useful with the SPMD techniques. Results obtained demonstrated have clearly shown the benefits of parallelization on the Geo cluster with MPI. The DD influences the performance of the 1-D ATCP on the nodes.

6. FUTURE WORK

Parallelization of 1-D ATCP employing the Stationary Iterative Techniques on Geo Cluster with MPI technique has been carried out. We suggest future work be done on the 1-D ATCP model using the Iterative Alternating Direction Implicit (IADE) method. Parallel implementation of the scheme could

7. ACNOWLEDGMENTS

The University of the West Indies Mona Campus awarded the Initiative Grant that motivated this study.

8. REFERENCES

[1] E. Aubanel, 'Scheduling of tasks in the parareal algorithm' Parallel Computing 37 (3), 2011, 172 – 182

[2] Cyril, M. Fabrice, 'Jacobi computation using mobile agent' Int'l Journal of Computer Science & Information Technologies, 1 (5), 2010, 392 – 401

[3] S. U. Ewedafe, H. S. Rio, 'Armadillo Generation Distributed Systems & Geranium CADCAM Cluster for solving 2-D Telegraph Equation' Int'l Jour. of Computer Mathematics, 88, Issue 3, 2011, 589 – 609

[4] S. U. Ewedafe, H. S. Rio, 'Parallelization of 2-D IADE-DY Scheme on Geranium CADCAM Cluster for Heat Equation' Int'l Jour. Of Advanced Research in Artificial Intelligence, 2 (6), 2013, pp. 27 – 33

[5] S. U. Ewedafe, H. S. Rio, 'On the Parallel Design and Analysis for 3-D ADI Telegraph Problem with MPI' Int'l Jour. Of Advanced Computer. Sci. and Applications, Vol. 5, No. 4, 2014

[6] P. Fahnl, A. C., Lienhard V., J. H, A. H. Slocum 'Thermal Management and Control in Testing Packaged Integrated Circuit (IC) Devices' Proc. 34th Inter Society Energy Conversion Conference' 1999

[7] W. Groop, E. Lusk, A. Skjellum, 'Using MPI, portable and parallel programming with the message passing interface,' 1999, 2nd Ed., Cambridge MA, MIT Press

[8] K. Jarvis, D.G. Alan, 'A High-Performance Communication Service for Parallel Computing on Distributed System', Parallel Computing 29, 2003, pp 851 – 878

[9] H. Laurant, 'A method for automatic placement of communications in SPMD parallelization' Parallel Computing 27, 2001, 1655 – 1664

[10] S. Martin, S. Benkner, S. Plans, 'Using Explicit Parallel Description to Support Programming of Heterogeneous Many-Core Systems, Parallel Computing 38 (2012), pp 52 – 65

[11] P. Miller, A. Becker, L. Kale, 'Using Shared Arrays in Message-Driven Parallel Programs' Parallel Computing 38 (2012), pp 66 - 74

[12] Peizong L., Z. Kedem, 'Automatic Data and Computation Decomposition on Distributed Memory Parallel Computers' ACM Transactions on Programming Languages and Systems, vol. 24, number 1, 2002, pp 1 – 50

[13] T. Rauber, G. Runger, 'A Transformation Approach to Derive Efficient Parallel Implementations. IEEE Transactions on Software Engineering, 26 (4), 2000, pp. 315 - 399

[14] P. Richard, M. Amin, E. David, E. Thomas, *'The work of Parallelism*' Computer Science University of California Berkeley CA 94720

[15] M. Sweetland, V. J. H. Lienhard, 'Active Thermal Control of Distributed Parameter System with Application to Testing of Packaged (IC) Devices' ASME Journal of Heat Transfer, 2003, pp 165 – 174

[16] S. Zarith, A. Ghaffar, N. Alias, F. Sham, A. Hassan, H. Hassan, 'Sequential Algorithms of Parabolic Equation in Solving Thermal Control Process on Printed Circuit Board Jour. Fundamental Science Issue 4, 2008, pp 379 – 385.

[17] W. Zheng-Su, Z. Baolin, C. Guang-Nan, 'Design and analysis for finite difference DD for 2-D heat equation', ICA3PP – 02, IEEE Computer Society