Fundamental locally one-dimensional method for 3-D thermal simulation

Tay, Wei Choon; Tan, Eng Leong; Heh, Ding Yu

2014

Tay, W. C., Tan, E. L., & Heh, D. Y. (2014). Fundamental locally one-dimensional method for 3-D thermal simulation. IEICE transactions on electronics, E97.C(7), 636-644.

<https://hdl.handle.net/10356/105063>

<https://doi.org/10.1587/transele.E97.C.636>

© 2014 The Institute of Electronics, Information and Communication Engineers. This paper was published in IEICE Transactions on Electronics and is made available as an electronic reprint (preprint) with permission of The Institute of Electronics, Information and Communication Engineers. The paper can be found at the following official DOI: <http://dx.doi.org/10.1587/transele.E97.C.636>. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper is prohibited and is subject to penalties under law.

Downloaded on 26 Apr 2023 19:35:19 SGT

Fundamental Locally One-Dimensional Method for 3-D Thermal Simulation

Wei Choon TAY^{†a)}, Eng Leong TAN[†], and Ding Yu HEH[†], *Nonmembers*

SUMMARY This paper presents a fundamental locally one-dimensional (FLOD) method for 3-D thermal simulation. We first propose a locally one-dimensional (LOD) method for heat transfer equation within general inhomogeneous media. The proposed LOD method is then cast into compact form and formulated into the FLOD method with operator-free right-hand-side (RHS), which leads to computationally efficient update equations. Memory storage requirements and boundary conditions for both FLOD and LOD methods are detailed and compared. Stability analysis by means of analyzing the eigenvalues of amplification matrix substantiates the stability of the FLOD method. Additionally, the potential instability of the Douglas Gunn (DG) alternating-direction-implicit (ADI) method for inhomogeneous media is demonstrated. Numerical experiments justify the gain achieved in the overall efficiency for FLOD over LOD, DG-ADI and explicit methods. Furthermore, the relative maximum error of the FLOD method illustrates good trade-off between accuracy and efficiency.

key words: Alternating-direction-implicit (ADI), finite-difference method, heat transfer, locally one-dimensional (LOD), stability, temperature

1. Introduction

To keep up to the pace of growth of Systems on Chips (SoC), the 3-D chip multi-processors, which is the vertical stacking of multiple silicon layers, referred to as 3-D stacking, is becoming the latest trend in processor development. The shorter interconnect length that is implemented using Through Silicon Vias (TSVs) across the layers reduces the resistivity, which results in shorter RC delays. This translates to greater performance, higher speed and a reduction in power consumption.

In spite of these benefits, a higher power density in the chip leads to higher temperature, which affects the reliability. As the number of stacked layers increases, the performance of the chip will also be affected. Therefore, it is important for chip-level thermal simulation to efficiently analyze the thermal distribution and locate the hot spots [1]–[12], not only for steady state but also for transient state.

The finite-difference method is known to be more flexible than other approaches such as finite element method for handling complex structures. However, the conventional explicit finite-difference method has its time step size restricted by the minimum cell size in the computation domain [13]. To overcome this stability constraint, an implicit algorithm based on the Douglas Gunn (DG) alternating-direction-implicit (ADI) method [14] has been introduced

in [2] for transient thermal simulation.

Unfortunately, it is found in this paper that for inhomogeneous media with large time step size, potential instability will still occur for the DG-ADI method in [2]. To mitigate this potential instability, we shall propose a locally one-dimensional (LOD) method for 3-D thermal simulation within general inhomogeneous media. The 3-D LOD method can be further simplified into efficient fundamental LOD (FLOD) method with operator-free right-hand-side (RHS). This results in computationally efficient update equations. Such algorithm is similar to a family of fundamental implicit schemes in electromagnetics [15]–[17], which feature similar fundamental updating structures. Nonetheless, the LOD method in heat transfer and LOD method in electromagnetics differ in terms of their updating procedures and operators.

The organization of this paper is as follows. Section 2 describes the formulation of the heat transfer equation using finite-difference method. The 3-D DG-ADI method for heat transfer equation will be provided first. However, this DG-ADI method is potentially unstable within inhomogeneous media. To overcome this instability, a stable 3-D LOD method will be presented in Section 3. The LOD method is further formulated into the FLOD method for computationally efficient update equations. The memory storage requirements and boundary conditions for both LOD and FLOD methods are detailed and compared. Section 4 provides the stability analysis by means of analyzing the eigenvalues of amplification matrix for DG-ADI and FLOD methods. Computational efficiency of FLOD over LOD, DG-ADI and explicit methods as well as the relative maximum error of FLOD method will be demonstrated.

2. Potentially Unstable DG-ADI method

To simulate the temperature of a system, the partial differential heat transfer equation [18], [19] is written as

$$\rho(\vec{r})C_p(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial t} = \nabla \cdot [\kappa(\vec{r})\nabla T(\vec{r}, t)] + g(\vec{r}, t) \quad (1)$$

subject to the thermal boundary condition which will be discussed in Section 3.4. T is the time-dependent temperature at any point, κ is the thermal conductivity (W/m·K), ρ is the density of the material (kg/m³), C_p is the specific heat capacity (J/kg·K) and g is the heat energy generation rate (W/m³).

To simulate temperature distribution using finite-difference methods [20], discretization is performed in both

Manuscript received September 27, 2013.

Manuscript revised January 28, 2014.

[†]The authors are with Nanyang Technological University, Singapore.

a) E-mail: tayw0034@e.ntu.edu.sg

DOI: 10.1587/transele.E97.C.636

space and time domains. This is done by fixing a grid spacing of Δx , Δy and Δz in space and Δt in time. The points that lie on the domain are then defined as

$$(i, j, k, n) \equiv (i\Delta x, j\Delta y, k\Delta z, n\Delta t)$$

where i, j, k and n are integers. For convenience, $T(x, y, z, t)$ is denoted as $T_{i,j,k}^n$ for this paper. For general inhomogeneous material, the heat transfer equation can now be expanded into

$$\begin{aligned} \rho(\vec{r})C_p(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial t} = & \frac{\partial}{\partial x} \left(\kappa(\vec{r}) \frac{\partial T(\vec{r}, t)}{\partial x} \right) + \frac{\partial}{\partial y} \left(\kappa(\vec{r}) \frac{\partial T(\vec{r}, t)}{\partial y} \right) \\ & + \frac{\partial}{\partial z} \left(\kappa(\vec{r}) \frac{\partial T(\vec{r}, t)}{\partial z} \right) + g(\vec{r}, t) \end{aligned} \quad (2)$$

We first discretize (2) into

$$\begin{aligned} T_{\text{DG}|i,j,k}^{n+1} = & r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+1} + T_{\text{DG}|i,j,k}^n)}{2\partial x} \right) \\ & + r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+1} + T_{\text{DG}|i,j,k}^n)}{2\partial y} \right) + T_{\text{DG}|i,j,k}^n \\ & + r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+1} + T_{\text{DG}|i,j,k}^n)}{2\partial z} \right) + G_{i,j,k}^n \end{aligned} \quad (3)$$

where

$$r_{i,j,k} = \frac{\Delta t}{\rho_{i,j,k} C_{p,i,j,k}}, \quad G_{i,j,k}^n = \frac{\Delta t}{\rho_{i,j,k} C_{p,i,j,k}} g_{i,j,k}^n.$$

According to DG-ADI method, the update equation in (3) is solved in three procedures as
For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} T_{\text{DG}|i,j,k}^{n+\frac{1}{3}} = & r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}|i,j,k}^n)}{2\partial x} \right) \\ & + r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial T_{\text{DG}|i,j,k}^n}{\partial y} \right) \\ & + r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial T_{\text{DG}|i,j,k}^n}{\partial z} \right) + T_{\text{DG}|i,j,k}^n + G_{i,j,k}^n \end{aligned} \quad (4a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$\begin{aligned} T_{\text{DG}|i,j,k}^{n+\frac{2}{3}} = & r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}|i,j,k}^n)}{2\partial x} \right) \\ & + r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}|i,j,k}^n)}{2\partial y} \right) \\ & + r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial T_{\text{DG}|i,j,k}^n}{\partial z} \right) + T_{\text{DG}|i,j,k}^n + G_{i,j,k}^n \end{aligned} \quad (4b)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$\begin{aligned} T_{\text{DG}|i,j,k}^{n+1} = & r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}|i,j,k}^n)}{2\partial x} \right) \\ & + r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}|i,j,k}^n)}{2\partial y} \right) \\ & + r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}|i,j,k}^{n+1} + T_{\text{DG}|i,j,k}^n)}{2\partial z} \right) + T_{\text{DG}|i,j,k}^n + G_{i,j,k}^n \end{aligned} \quad (4c)$$

Note that all spatial derivatives are approximated using central differencing. The 3-D DG-ADI method for heat transfer equation within inhomogeneous media is still potentially unstable (to be demonstrated in Section 4.1). In order to achieve an unconditionally stable implicit method for heat transfer equation, we now propose the 3-D LOD method which will be described in the next section.

3. Proposed Stable LOD method

3.1 Formulation of LOD method

To formulate the 3-D LOD method, we rearrange (3) into compact form as

$$\left(1 - \frac{1}{2}A\right) T_{\text{LOD}|i,j,k}^{n+\frac{1}{3}} = \left(1 + \frac{1}{2}A\right) T_{\text{LOD}|i,j,k}^n + \frac{1}{3}G_{i,j,k}^n \quad (5a)$$

$$\left(1 - \frac{1}{2}B\right) T_{\text{LOD}|i,j,k}^{n+\frac{2}{3}} = \left(1 + \frac{1}{2}B\right) T_{\text{LOD}|i,j,k}^{n+\frac{1}{3}} + \frac{1}{3}G_{i,j,k}^n \quad (5b)$$

$$\left(1 - \frac{1}{2}C\right) T_{\text{LOD}|i,j,k}^{n+1} = \left(1 + \frac{1}{2}C\right) T_{\text{LOD}|i,j,k}^{n+\frac{2}{3}} + \frac{1}{3}G_{i,j,k}^n \quad (5c)$$

where

$$\begin{aligned} AT_{i,j,k} &= r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial T_{i,j,k}}{\partial x} \right), BT_{i,j,k} = r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial T_{i,j,k}}{\partial y} \right), \\ CT_{i,j,k} &= r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial T_{i,j,k}}{\partial z} \right). \end{aligned}$$

By applying central difference approximation for spatial operators in (5) and upon some arithmetic manipulations, we have

For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} -\frac{1}{2}\alpha_{x_{i,j,k}} T_{\text{LOD}|i-1,j,k}^{n+\frac{1}{3}} + (1 + \alpha_{x_{i,j,k}}) T_{\text{LOD}|i,j,k}^{n+\frac{1}{3}} - \frac{1}{2}\beta_{x_{i,j,k}} T_{\text{LOD}|i+1,j,k}^{n+\frac{1}{3}} \\ = \frac{1}{2}\alpha_{x_{i,j,k}} T_{\text{LOD}|i-1,j,k}^n + \frac{1}{2}\beta_{x_{i,j,k}} T_{\text{LOD}|i+1,j,k}^n \\ + (1 - \alpha_{x_{i,j,k}}) T_{\text{LOD}|i,j,k}^n + \frac{1}{3}G_{i,j,k}^n \end{aligned} \quad (6a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$\begin{aligned} -\frac{1}{2}\alpha_{y_{i,j,k}} T_{\text{LOD}|i,j-1,k}^{n+\frac{2}{3}} + (1 + \alpha_{y_{i,j,k}}) T_{\text{LOD}|i,j,k}^{n+\frac{2}{3}} - \frac{1}{2}\beta_{y_{i,j,k}} T_{\text{LOD}|i,j+1,k}^{n+\frac{2}{3}} \\ = \frac{1}{2}\alpha_{y_{i,j,k}} T_{\text{LOD}|i,j-1,k}^{n+\frac{1}{3}} + \frac{1}{2}\beta_{y_{i,j,k}} T_{\text{LOD}|i,j+1,k}^{n+\frac{1}{3}} \\ + (1 - \alpha_{y_{i,j,k}}) T_{\text{LOD}|i,j,k}^{n+\frac{1}{3}} + \frac{1}{3}G_{i,j,k}^n \end{aligned} \quad (6b)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$\begin{aligned} & -\frac{1}{2}\alpha_{z_{i,j,k}}T_{\text{LOD}}|_{i,j,k-1}^{n+1} + \left(1 + a_{z_{i,j,k}}\right)T_{\text{LOD}}|_{i,j,k}^{n+1} - \frac{1}{2}\beta_{z_{i,j,k}}T_{\text{LOD}}|_{i,j,k+1}^{n+1} \\ & = \frac{1}{2}\alpha_{z_{i,j,k}}T_{\text{LOD}}|_{i,j,k-1}^{n+\frac{2}{3}} + \frac{1}{2}\beta_{z_{i,j,k}}T_{\text{LOD}}|_{i,j,k+1}^{n+\frac{2}{3}} \\ & + \left(1 - a_{z_{i,j,k}}\right)T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{3}G_{i,j,k}^n \end{aligned} \quad (6c)$$

where

$$\begin{aligned} \alpha_{x_{i,j,k}} &= r_{i,j,k} \frac{K_{i,j,k} + K_{i-1,j,k}}{2\Delta x^2}, \quad \beta_{x_{i,j,k}} = r_{i,j,k} \frac{K_{i+1,j,k} + K_{i,j,k}}{2\Delta x^2}, \\ \alpha_{y_{i,j,k}} &= r_{i,j,k} \frac{K_{i,j,k} + K_{i,j-1,k}}{2\Delta y^2}, \quad \beta_{y_{i,j,k}} = r_{i,j,k} \frac{K_{i,j+1,k} + K_{i,j,k}}{2\Delta y^2}, \\ \alpha_{z_{i,j,k}} &= r_{i,j,k} \frac{K_{i,j,k} + K_{i,j,k-1}}{2\Delta z^2}, \quad \beta_{z_{i,j,k}} = r_{i,j,k} \frac{K_{i,j,k+1} + K_{i,j,k}}{2\Delta z^2}, \\ a_{x_{i,j,k}} &= \frac{\alpha_{x_{i,j,k}} + \beta_{x_{i,j,k}}}{2}, \quad a_{y_{i,j,k}} = \frac{\alpha_{y_{i,j,k}} + \beta_{y_{i,j,k}}}{2}, \\ a_{z_{i,j,k}} &= \frac{\alpha_{z_{i,j,k}} + \beta_{z_{i,j,k}}}{2}. \end{aligned}$$

3.2 Formulation of FLOD method

Despite having a single operator on the RHS of the updating equations, the LOD method in Section 3.1 can be further reduced into the efficient fundamental scheme similar to electromagnetics [16]. From (5a), we have

$$\begin{aligned} \left(1 - \frac{1}{2}A\right)T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} &= \left(1 + \frac{1}{2}A\right)T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3}G_{i,j,k}^n \\ &= 2T_{\text{LOD}}|_{i,j,k}^n - \left(1 - \frac{1}{2}A\right)T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3}G_{i,j,k}^n. \end{aligned} \quad (7)$$

This can be manipulated readily to give

$$\left(1 - \frac{1}{2}A\right)\left(T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{LOD}}|_{i,j,k}^n\right) = 2T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3}G_{i,j,k}^n \quad (8)$$

where the scalar terms in bracket may be denoted by auxiliary variable

$$V|_{i,j,k}^{n+\frac{1}{3}} = T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{LOD}}|_{i,j,k}^n. \quad (9)$$

Similar manipulation applies to (5b) and (5c) which leads to auxiliary variables $V|_{i,j,k}^{n+\frac{2}{3}}$ and $V|_{i,j,k}^{n+1}$ respectively. Combining all auxiliary and field variables, we can obtain the 3-D FLOD method written as

$$\left(\frac{1}{2} - \frac{1}{4}A\right)V|_{i,j,k}^{n+\frac{1}{3}} = T_{\text{F}}|_{i,j,k}^n + \frac{1}{6}G_{i,j,k}^n \quad (10a)$$

$$T_{\text{F}}|_{i,j,k}^{n+\frac{1}{3}} = V|_{i,j,k}^{n+\frac{1}{3}} - T_{\text{F}}|_{i,j,k}^n \quad (10b)$$

$$\left(\frac{1}{2} - \frac{1}{4}B\right)V|_{i,j,k}^{n+\frac{2}{3}} = T_{\text{F}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{6}G_{i,j,k}^n \quad (10c)$$

$$T_{\text{F}}|_{i,j,k}^{n+\frac{2}{3}} = V|_{i,j,k}^{n+\frac{2}{3}} - T_{\text{F}}|_{i,j,k}^{n+\frac{1}{3}} \quad (10d)$$

$$\left(\frac{1}{2} - \frac{1}{4}C\right)V|_{i,j,k}^{n+1} = T_{\text{F}}|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{6}G_{i,j,k}^n \quad (10e)$$

$$T_{\text{F}}|_{i,j,k}^{n+1} = V|_{i,j,k}^{n+1} - T_{\text{F}}|_{i,j,k}^{n+\frac{2}{3}} \quad (10f)$$

This algorithm has no (spatial) operators on the RHS, hence it leads to a decrease in the overall flops count as well as an increase in the overall efficiency. Note that FLOD method for electromagnetics [16] is not directly applicable here because it involves two procedures with its RHS free of matrix operators **A** and **B**. The FLOD method for heat transfer in (10) here involves three procedures with its RHS free from scalar operators *A*, *B* and *C*.

Both FLOD and LOD methods are one in the same. Therefore the temperature of FLOD method for each time step, i.e. T_{F}^n , $T_{\text{F}}^{n+\frac{1}{3}}$, $T_{\text{F}}^{n+\frac{2}{3}}$ and T_{F}^{n+1} directly corresponds to those of LOD method. The equivalence of both FLOD and LOD methods becomes evident here.

By applying central approximation and arithmetic manipulation for (10), we have

For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} & -\frac{1}{4}\alpha_{x_{i,j,k}}V|_{i-1,j,k}^{n+\frac{1}{3}} + \frac{1}{2}\left(1 + a_{x_{i,j,k}}\right)V|_{i,j,k}^{n+\frac{1}{3}} - \frac{1}{4}\beta_{x_{i,j,k}}V|_{i+1,j,k}^{n+\frac{1}{3}} \\ & = T_{\text{F}}|_{i,j,k}^n + \frac{1}{6}G_{i,j,k}^n \end{aligned} \quad (11a)$$

$$T_{\text{F}}|_{i,j,k}^{n+\frac{1}{3}} = V|_{i,j,k}^{n+\frac{1}{3}} - T_{\text{F}}|_{i,j,k}^n \quad (11b)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$\begin{aligned} & -\frac{1}{4}\alpha_{y_{i,j,k}}V|_{i,j-1,k}^{n+\frac{2}{3}} + \frac{1}{2}\left(1 + a_{y_{i,j,k}}\right)V|_{i,j,k}^{n+\frac{2}{3}} - \frac{1}{4}\beta_{y_{i,j,k}}V|_{i,j+1,k}^{n+\frac{2}{3}} \\ & = T_{\text{F}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{6}G_{i,j,k}^n \end{aligned} \quad (11c)$$

$$T_{\text{F}}|_{i,j,k}^{n+\frac{2}{3}} = V|_{i,j,k}^{n+\frac{2}{3}} - T_{\text{F}}|_{i,j,k}^{n+\frac{1}{3}} \quad (11d)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$\begin{aligned} & -\frac{1}{4}\alpha_{z_{i,j,k}}V|_{i,j,k-1}^{n+1} + \frac{1}{2}\left(1 + a_{z_{i,j,k}}\right)V|_{i,j,k}^{n+1} - \frac{1}{4}\beta_{z_{i,j,k}}V|_{i,j,k+1}^{n+1} \\ & = T_{\text{F}}|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{6}G_{i,j,k}^n \end{aligned} \quad (11e)$$

$$T_{\text{F}}|_{i,j,k}^{n+1} = V|_{i,j,k}^{n+1} - T_{\text{F}}|_{i,j,k}^{n+\frac{2}{3}} \quad (11f)$$

By comparing (6) and (11), we find that the update equations of (11) are obviously the most computationally efficient. This is due to the operator-free RHS of the FLOD algorithm.

3.3 Memory Allocation

At first sight, one might think that (11) incurs more memory resources due to the presence of auxiliary variables *V*'s. To clarify the memory allocation for various variables, the pseudocode of the implicit methods across iterations is shown in Figures. 1 and 2.

Figure 1 presents the pseudocode of 3-D FLOD method across iterations. It can be seen that two variables, i.e. **t** and **v** are required. That is, we are able to reuse the memory spaces such that **t** is multi-purpose and may represent T_{F}^n , $T_{\text{F}}^{n+\frac{1}{3}}$, $T_{\text{F}}^{n+\frac{2}{3}}$ and T_{F}^{n+1} . Likewise, **v** is multi-purpose and may

```

for n = 0, 1, 2, ... //n ← time step in the main iteration
v = inv(1/2 - 1/4 A)t //t ← T_F^n, v ← V^{n+1/3}
t = v - t //v ← V^{n+1/3}, t at RHS ← T_F^n, t at LHS ← T_F^{n+1/3}
v = inv(1/2 - 1/4 B)t //t ← T_F^{n+1/3}, v ← V^{n+2/3}
t = v - t //v ← V^{n+2/3}, t at RHS ← T_F^{n+1/3}, t at LHS ← T_F^{n+2/3}
v = inv(1/2 - 1/4 C)t //t ← T_F^{n+2/3}, v ← V^{n+1}
t = v - t //v ← V^{n+1}, t at RHS ← T_F^{n+2/3}, t at LHS ← T_F^{n+1}
end

```

Fig. 1 Pseudocode of 3-D FLOD method across iterations.

```

for n = 0, 1, 2, ... //n ← time step in the main iteration
v = (1 + 1/2 A)t //t ← T_LOD^n, v ← RHS of (5a)
t = inv(1 - 1/2 A)v //t ← T_LOD^{n+1/3}, v ← RHS of (5a)
v = (1 + 1/2 B)t //t ← T_LOD^{n+1/3}, v ← RHS of (5b)
t = inv(1 - 1/2 B)v //t ← T_LOD^{n+2/3}, v ← RHS of (5b)
v = (1 + 1/2 C)t //t ← T_LOD^{n+2/3}, v ← RHS of (5c)
t = inv(1 - 1/2 C)v //t ← T_LOD^{n+1}, v ← RHS of (5c)
end

```

Fig. 2 Pseudocode of 3-D LOD method across iterations.

represent $V^{n+1/3}$, $V^{n+2/3}$ and V^{n+1} .

The pseudocode of 3-D LOD method in Fig. 2 also requires two variables, i.e. t and v for computation in the simulation. Therefore, it is evident that both LOD and FLOD methods require the same amount of memory. Due to the simplicity and conciseness of the update equations for FLOD method, it is more appealing than LOD method of its implementation for solving heat transfer equation.

3.4 Boundary Conditions

The pseudocodes in Figs. 1 and 2 present the memory allocation and the general flow of the simulated program. However, another taxing issue with the implicit methods is at the boundary of the computation domain.

For example, consider a simulation domain of $ie \times je \times ke$ grids, where the grids are indexed from 0 to ie in the x -direction, 0 to je in the y -direction and 0 to ke in the z -direction. Temperature variable T at $T_{-1,j,k}$, $T_{ie+1,j,k}$, $T_{i,-1,k}$, $T_{i,je+1,k}$, $T_{i,j,-1}$ or $T_{i,j,ke+1}$ (virtual points) will be out of the simulation domain. This will result in extra efforts for the implementation of the tridiagonal system of equations.

In order to cater for general thermal simulation, we shall treat the boundary by introducing

$$\kappa(\vec{r}) \frac{\partial T(\vec{r}, t)}{\partial n_\xi} = h_\xi T_\infty - h_\xi T(\vec{r}, t) \quad (12)$$

where $\partial/\partial n_\xi$ ($\xi = x, y, z$) is the differentiation along the outward direction normal to the boundary, T_∞ is a pre-specified temperature and h_ξ is the equivalent heat transfer coefficients on the boundary.

By discretizing (12) at various boundaries ($i=0$, $i=ie$, $j=0$, $j=je$, $k=0$, $k=ke$), the virtual points at various time steps can be expressed as

$$T_{-1,j,k}^n = T_{1,j,k}^n + \frac{2\Delta x}{\kappa_{0,j,k}} (h_x^- T_\infty - h_x^- T_{0,j,k}^n) \quad (13a)$$

```

First Procedure
// Main grid
for j=0:je
for k=0:ke
i=0;
    auxiliary Eq. (11a) with BC for  $V_{-1,j,k}^{n+1/3}$ 
    Eq. (11b)
for i=1:ie-1
    auxiliary Eq. (11a)
    Eq. (11b)
end
i=ie;
    auxiliary Eq. (11a) with BC for  $V_{ie+1,j,k}^{n+1/3}$ 
    Eq. (11b)
end
end

```

Fig. 3 Pseudocode of 3-D FLOD method for first procedure of iteration.

```

First Procedure
// Main grid
for j=0:je
for k=0:ke
i=0;
    Eq. (6a) with BCs for  $T_{LOD,-1,j,k}^{n+1/3}$  &  $T_{LOD,-1,j,k}^n$ 
for i=1:ie-1
    Eq. (6a)
end
i=ie;
    Eq. (6a) with BCs for  $T_{LOD,ie+1,j,k}^{n+1/3}$  &  $T_{LOD,ie+1,j,k}^n$ 
end
end

```

Fig. 4 Pseudocode of 3-D LOD method for first procedure of iteration.

$$T_{ie+1,j,k}^n = T_{ie-1,j,k}^n + \frac{2\Delta x}{\kappa_{ie,j,k}} (h_x^+ T_\infty - h_x^+ T_{ie,j,k}^n) \quad (13b)$$

$$T_{i,-1,k}^n = T_{i,1,k}^n + \frac{2\Delta y}{\kappa_{i,0,k}} (h_y^- T_\infty - h_y^- T_{i,0,k}^n) \quad (13c)$$

$$T_{i,je+1,k}^n = T_{i,je-1,k}^n + \frac{2\Delta y}{\kappa_{i,je,k}} (h_y^+ T_\infty - h_y^+ T_{i,je,k}^n) \quad (13d)$$

$$T_{i,j,-1}^n = T_{i,j,1}^n + \frac{2\Delta z}{\kappa_{i,j,0}} (h_z^- T_\infty - h_z^- T_{i,j,0}^n) \quad (13e)$$

$$T_{i,j,ke+1}^n = T_{i,j,ke-1}^n + \frac{2\Delta z}{\kappa_{i,j,ke}} (h_z^+ T_\infty - h_z^+ T_{i,j,ke}^n) \quad (13f)$$

where h_x^- , h_x^+ , h_y^- , h_y^+ , h_z^- , h_z^+ are the effective heat transfer coefficients calculated from the equivalent thermal resistance on the boundary $x = 0$, $x = ie$, $y = 0$, $y = je$, $z = 0$ and $z = ke$, respectively.

Figures 3 and 4 show the pseudocode of 3-D FLOD and LOD methods for first procedure of iteration respectively. For every procedure, the boundary conditions can be implemented together with the main grids within a single for-loop. Using FLOD method [c.f. Fig. 3], only two virtual point treatments, $V_{-1,j,k}^{n+1/3}$ and $V_{ie+1,j,k}^{n+1/3}$ (one on each boundary) are required for the update equation on the boundaries. For LOD method [c.f. Fig. 4], four virtual point treatments $T_{LOD,-1,j,k}^{n+1/3}$, $T_{LOD,-1,j,k}^n$, $T_{LOD,ie+1,j,k}^{n+1/3}$ and $T_{LOD,ie+1,j,k}^n$ (two on each boundary) are required. Moreover, with a reduction in the

number of arithmetic operations for FLOD method, the implementation of FLOD method is more desirable as compared to LOD method.

4. Numerical Experiments

4.1 Stability Analysis

To analyze the stability of 3-D DG-ADI and FLOD methods in inhomogeneous media, we consider a simulation layout of a computation domain with inhomogeneous medium shown in Fig. 5. The computation domain has dimension of $12 \times 12 \times 12$ grids with spatial step $\Delta x = \Delta y = \Delta z = 20$ nm and time step Δt specified in terms of γ where

$$\gamma = \frac{\kappa_{i,j,k} \Delta t}{\rho_{i,j,k} C_{p,i,j,k}} \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right) \leq \frac{1}{2} \quad (14)$$

The materials and associated parameters used in the inhomogeneous medium are as follows: silicon (thermal conductivity $\kappa = 131$ W/m·K, density $\rho = 2500$ kg/m³, specific heat $C_p = 700$ J/kg·K), silver (thermal conductivity $\kappa = 315.82$ W/m·K, density $\rho = 8954.5$ kg/m³, specific heat $C_p = 383.6$ J/kg·K) and alumina (thermal conductivity $\kappa = 20$ W/m·K, density $\rho = 2699$ kg/m³, specific heat $C_p = 901$ J/kg·K). The effective heat transfer coefficients for all convection boundary conditions are set as 2×10^4 W/m²·K. Ambient temperature is assumed to be 26.85°C. The heat generation rate (W/m³) in silver and alumina is given by

$$g_{i,j,k}^m = \begin{cases} (1 + \cos(2\pi f_0 t + \pi)) \times 0.5 \times 10^{14} & \text{if } 0 \leq t \leq \frac{1}{2f_0} \\ 1.0 \times 10^{14} & \text{otherwise} \end{cases} \quad (15)$$

where $f_0 = 1.38 \times 10^5$ Hz. Note that the maximum value of the heat generation rate is set at 1.0×10^{14} W/m³ [8].

We now investigate the eigenvalues of amplification matrix of the system for DG-ADI and FLOD methods. Note that the von-Neumann Fourier method is not applicable here due to the inhomogeneity of the computation domain. Instead, we need to consider the amplification matrix of the

Fig. 5 Layout of a computation domain with inhomogeneous medium.

whole computation domain. Fig. 6 shows the scatter plot of eigenvalues of the amplification matrix for conventional 3-D DG-ADI method [c.f. (4)] for various γ . By gradually increasing γ , it is found that some eigenvalues start to deviate outside the unit semi circle around $\gamma = 30$. This further substantiates the potential instability for the conventional DG-ADI method. On the other hand, Fig. 7 shows that all the eigenvalues for FLOD method [c.f. (11)] are located inside the unit semi circle. This verifies the stability of our FLOD method.

Figure 8 shows the transient temperature at observation point ($i = 4, j = 4, k = 4$) computed using the conventional 3-D DG-ADI method [c.f. (4)] at $\gamma = 30$. It can be seen that the transient temperature goes unbounded over time, which confirms the potential instability of the conventional DG-ADI method. However, there is no instability in the computed transient temperatures for FLOD method [c.f. (11)] for $\gamma \geq \frac{1}{2}$, as shown in Fig. 9. In addition, the explicit method is included for comparison. The results computed by FLOD method are similar with that of explicit method, which validates our proposed method.

Fig. 6 Scatter plot of eigenvalues of the amplification matrix for 3-D DG-ADI method [c.f. (4)] for various γ .

Fig. 7 Scatter plot of eigenvalues of the amplification matrix for FLOD method [c.f. (11)] for various γ .

Fig. 8 Transient temperature at observation point ($i = 4, j = 4, k = 4$) computed using the conventional 3-D DG-ADI method [c.f. (4)], $\gamma = 30$.

Fig. 9 Transient temperature at observation point ($i = 4, j = 4, k = 4$) computed using the 3-D FLOD method [c.f. (11)], for various γ .

4.2 Efficiency

Next, we conduct numerical experiments for a range of computation domains from $50 \times 50 \times 50$ to $250 \times 250 \times 250$ grids with $\gamma = 50$. The programs have been compiled using Microsoft Visual C++ under Microsoft Windows 7 operating system (OS) running on Intel Dual Core 2.66 GHz processor platform.

Table 1 shows the CPU efficiency gains of FLOD method over LOD, DG-ADI and explicit methods for various computation domains. It can be seen that the efficiency gain of FLOD method over LOD method ranges from 1.3 to 1.6. This is due to the operator-free RHS for FLOD method, [c.f. (11)], which reduces the number of arithmetic operations compared to that of LOD method, [c.f. (6)]. If FLOD method is compared to DG-ADI method, the efficiency gain increases to 3 to 4 times as the DG-ADI method involves even larger amount of RHS operators, [c.f. (4)]. If

Table 1 Efficiency gains of FLOD method over LOD, DG-ADI and explicit methods for various computation domains ($\gamma=50$).

Domain Size	Efficiency Gain		
	FLOD vs LOD	FLOD vs DG-ADI	FLOD vs Explicit
$50 \times 50 \times 50$	1.43	4.10	91.93
$100 \times 100 \times 100$	1.58	3.65	104.14
$150 \times 150 \times 150$	1.35	3.30	93.85
$200 \times 200 \times 200$	1.35	3.29	100.96
$250 \times 250 \times 250$	1.34	3.31	100.60

Fig. 10 A chip layout with the power density in each hierarchical function blocks.

Fig. 11 Temperature profile of the chip at 0.1 s.

the FLOD method is further compared to explicit method, the efficiency gain improves up to 100 times. This is because the explicit method has its time step size restricted by the minimum cell size in the computation domain, while the FLOD method has its time step size 100 times ($\gamma = 50$) larger.

Fig. 13 Relative maximum error for 3-D FLOD method with various γ .

Fig. 14 (a) Floorplan of Alpha 21364 processor (b) Power density in each hierarchical function blocks (c) Temperature profile of the chip in steady state.

For the numerical simulations, we consider a 3-D chip with layout of its hierarchy function blocks and power density shown in Fig. 10. The chip size is $2.4 \text{ mm} \times 2.8 \text{ mm} \times 0.4 \text{ mm}$ and the spatial steps are $\Delta x = \Delta y = \Delta z = 20 \text{ }\mu\text{m}$. The effective heat transfer coefficients h_x^- , h_x^+ , h_y^- and h_y^+ for the convection boundary conditions on the sides of the chip are set as $2.5 \times 10^3 \text{ W/m}^2 \cdot \text{K}$. The primary heat transfer path h_z^+ and the secondary heat transfer path h_z^- are set as $3 \times 10^3 \text{ W/m}^2 \cdot \text{K}$ and $4 \times 10^4 \text{ W/m}^2 \cdot \text{K}$, respectively.

Next, we investigate the error of the FLOD method by computing the relative maximum error defined as

$$\text{Error} = \frac{\left\| T_{\text{F}}|i,j,k\rangle - T_{\text{explicit}}|i,j,k\rangle \right\|_{\infty}}{\left\| T_{\text{explicit}}|i,j,k\rangle \right\|_{\infty}}. \quad (16)$$

Figure 13 shows the relative maximum error for 3-D FLOD method with various γ . At $\gamma = 5$, FLOD method allows us to choose Δt to be 10 times larger than that of the explicit method, at the expense of 9.7×10^{-4} relative maximum error. Even at $\gamma = 50$, FLOD method allows us to choose Δt to be 100 times larger than that of the explicit method, at the expense of 1.1×10^{-2} relative maximum error. This shows that FLOD method exhibits good trade-off between accuracy and efficiency.

We further simulate another example which floorplan closely resembles the Alpha 21364 processor [3]–[5] shown in Fig. 14a. The processor chip follows the dimension of $3.3\text{ mm} \times 3.3\text{ mm} \times 0.5\text{ mm}$ with a 20 nm power source layer included on top of the processor chip [7]–[9]. The power density in each hierarchical function blocks of the Alpha 21364 processor is provided in Fig. 14b. Figure 14c

shows the temperature profile at the centre of the chip in steady state. It can be seen that the core function blocks are the region with the higher temperature where the arithmetics operations and instructions are executed.

5. Conclusion

This paper has presented a FLOD method for 3-D thermal simulation. The LOD method is first proposed for heat transfer equation within general inhomogeneous media. The proposed LOD method is then cast into matrix form and formulated into the FLOD method with operator-free RHS, which leads to computationally efficient update equations. Memory storage requirements and boundary conditions for both FLOD and LOD methods have been detailed and compared. Stability analysis by means of analyzing the eigenvalues of amplification matrix has substantiated the stability of the FLOD method. Additionally, the potential instability of the DG-ADI method for inhomogeneous media has been demonstrated. Numerical experiments have justified the gain achieved in the overall efficiency for FLOD over LOD, DG-ADI and explicit methods. Furthermore, the relative maximum error of the FLOD method has also been illustrated, which exhibits good trade-off between accuracy and efficiency.

References

- [1] D. Chen, E. Li, E. Rosenbaum, and S. M. Kang, "Interconnect thermal modeling for accurate simulation of circuit timing and reliability," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 2, pp. 197–205, Feb. 2000.
- [2] T.-Y. Wang and C. C.-P. Chen, "3-D Thermal-ADI: A linear-time chip level transient thermal simulator," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 12, pp. 1434–1445, Dec. 2002.
- [3] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," *Proc. of the 30th Annual Int. Symp. on Computer Architecture*, (ISCA), vol. 31, no. 2, pp. 2–13, May 2003.
- [4] K. Sankaranarayanan, S. Velusamy, M. R. Stan, and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level," *J. Instruction-Level Parallelism*, vol. 8, no. 1, pp. 1–16, Oct. 2005.
- [5] J. Long, S. O. Memik, G. Memik, and R. Mukherjee, "Thermal monitoring mechanisms for chip multiprocessors," *Trans. on Architecture and Code Optimization*, vol. 5, no. 2, pp. 1–33, Aug. 2008.
- [6] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "IC thermal simulation and modelling via efficient multigrid-based approaches," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, no. 9, pp. 1763–1776, Sept. 2006.
- [7] Y. Zhan and S. S. Sapatnekar, "High-efficiency green function-based thermal simulation algorithms," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1661–1675, Sept. 2007.
- [8] P.-Y. Huang, C.-K. Lin and Y.-M. Lee, "Full-chip thermal analysis for the early design stage via generalized integral transforms," *IEEE Trans. VLSI Syst.*, vol. 17, no. 5, pp. 613–626, May 2009.
- [9] H.-C. Chang, P.-Y. Huang, T.-J. Li, and Y.-M. Lee, "Statistical electro-thermal analysis with high compatibility of leakage power models," *Proc. IEEE Int. Syst. on Chip Conf. (SOCC)*, pp. 139–144, Sept. 2010.
- [10] J. L. Ayala, A. Sridhar, and D. Cuesta, "Thermal modeling and analysis of 3D multi-processor chip," *Integration, the VLSI Journal*, vol. 43, no. 4, pp. 327–341, Sept. 2010.
- [11] Y. Liu and H. Yang, "Temperature-aware leakage estimation using piecewise linear power models," *IEICE Trans. Electron.*, vol. E93-C, no. 12, pp. 1679–1691, Dec. 2010.
- [12] L. Zhou, N. Wu, and X. Chen, "A design methodology for three-dimensional hybrid NoC-Bus architecture," *IEICE Trans. Electron.*, vol. E96-C, no. 4, pp. 492–500, Apr. 2013.
- [13] R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow*. Springer-Verlag, Berlin, 1984.
- [14] J. Douglas, Jr. and J. E. Gunn, "A general formulation of alternating direction methods—Part I: Parabolic and hyperbolic problems," *Numer. Math.*, vol. 6, no. 1, pp. 428–453, 1964.
- [15] E. L. Tan, "Unconditionally stable LOD-FDTD method for 3-D Maxwell's equations," *IEEE Microw. Wireless Comp. Lett.*, vol. 17, no. 2, pp. 85–87, Feb. 2007.
- [16] E. L. Tan, "Fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain methods," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 1, pp. 170–177, Jan. 2008.
- [17] J. Shibayama, N. Sasaki, Y. Wakabayashi, J. Yamauchi, and H. Nakano, "Frequency-dependent fundamental LOD-FDTD formulation for a multipole debye model," *IEEE Photonics Tech. Lett.*, vol. 25, no. 10, pp. 965–968, May 2013.
- [18] M. N. Ozisik, *Boundary Value Problems of Heat Conduction*, New York: Dover, 1968.
- [19] G. E. Myers, *Analytical Methods in Conduction Heat Transfer*. 2nd ed. Schenectady, NY: Genium, 1987.
- [20] M. N. Ozisik, *Finite Difference Methods in Heat Transfer*, New York: CRC, 1994.

Wei Choon Tay received the B.Eng. degree with first class honors from Nanyang Technological University, Singapore in 2008. He is currently working toward the Ph.D. degree in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include finite-difference time-domain method in areas such as electromagnetics and thermal simulations.

Eng Leong Tan received the B.Eng. (Electrical) degree with first class honors from the University of Malaya, Malaysia, and the Ph.D. degree in Electrical Engineering from Nanyang Technological University, Singapore. From 1991 to 1992, he was a Research Assistant at the University of Malaya. From 1991 to 1994, he worked part time at Commercial Network Corporations Sdn. Bhd., Malaysia. From 1999 to 2002, he was a Member of Technical Staff at the Institute for Infocomm Research, Singapore.

Since 2002, he is with the School of Electrical and Electronic Engineering, Nanyang Technological University, where he is currently an Associate Professor. His research interests include electromagnetic and acoustic simulations, RF and microwave circuit design.

Ding Yu Heh received the B.Eng. degree (first class honors) from Multimedia University, Cyberjaya, Malaysia, the M.Sc. and Ph.D. degrees from Nanyang Technological University, Singapore in 2006, 2007 and 2012, respectively. He is currently a postdoctoral fellow in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research areas are focused mainly on computational electromagnetics, especially the finite-difference time-domain method and its ap-

plications in bioelectromagnetics and photonics.