

POLITECNICO DI MILANO

Corso di Laurea Magistrale in
Ingegneria dell' Automazione



Object-oriented thermal modelling with non-uniform grids: a metaprogramming approach

Relatore: Prof. Leva Alberto

Co-relatore: Prof. Federico Terraneo

Tesi di Laurea di:
Palermo Fabio matr. 900190

Anno Accademico 2018-2019

Scrivo queste poche righe a ridosso del termine ultimo per la consegna della tesi. Arriva anche per me un momento molto importante, quello della laurea, che segnerà un passaggio fondamentale della mia vita.

Ripensando ai cinque anni appena trascorsi non posso che rendermi conto di quanto sia cambiato e di quanto questo tempo mi abbia formato.

Tutto questo non sarebbe stato possibile senza l'aiuto ed il sostegno di alcune persone che è per me impossibile non menzionare e ringraziare.

Non posso che iniziare dalla mia famiglia, senza la quale non sarebbe stato possibile intraprendere questa strada, e non parlo solo del sostegno economico ma anche e soprattutto di quello morale; spesso è stato silenzioso o è passato inosservato, ma so che loro hanno sempre fatto il massimo per incoraggiarmi e starmi vicino ed è per questo che dedico a loro in primis questa gioia così grande.

Un ringraziamento particolare ai miei nonni e soprattutto a mia nonna Anna che fin da piccolo mi preparò a ciò che mi aspettava.

Ci sono poi gli amici, quelli vicini e quelli lontani, quelli che hanno sempre creduto in me, senza i quali non sarei quel che sono adesso e con cui ho condiviso tutte le vittorie e le gioie ma anche le sconfitte e le delusioni di questo percorso, e con i quali mi accingo a festeggiare un traguardo così tanto importante.

Ringrazio il professor Leva e il professor Terraneo per tutta la disponibilità e l'aiuto fornitomi dall'inizio alla fine di questo lavoro, risultando sempre capaci di chiarire ogni mio dubbio con gentilezza, ammirevole comprensione e simpatia.

Probabilmente ci sono tante altre persone che dovrei nominare, ma non voglio rubare altro tempo a chi si accinge nella lettura di questa tesi.

Milano 25 luglio 2019

Contents

Introduction	7
<i>Description of the complete system</i>	9
<i>Part of the system considered</i>	10
<i>Technological Motivation</i>	12
Mathematical Model	13
<i>Starting consideration</i>	15
<i>Finite element methods</i>	15
<i>Study of the approximations</i>	17
Ziggurat-based partition	18
Pyramid based partition	18
Decreased area partition	18
<i>Electrical model result</i>	19
Interfaces	21
<i>Introduction</i>	23
<i>Causal interface</i>	23
Point-point and inspection approach	24
Line-Line inspection approach	24
Coefficient approach	24
<i>Acausal interface</i>	25
Least Common Multiple approach	25
Area coefficient approach	27
Metaprogramming	29
<i>Introduction</i>	31
Coefficient approach with metaprogramming	31
Area coefficient approach with metaprogramming	32
<i>Block summary</i>	35
Base Block	35
Causal Block	36
Acausal Block	37
Tests and Conclusions	39
<i>Test summary</i>	41
<i>3x3x3 Test with sinusoidal (0 offset)</i>	41
Results	42
<i>Test 3x3x3 (with offset)</i>	42
Result	43
<i>Test 3x3x3 and 2x3x3</i>	43
Result	44
<i>Conclusions</i>	44
Bibliography	45

Figure Index

1: GENERAL SYSTEM'S SCHEME.....	9
2: V-F OPERATIONAL ZONE.....	10
3: HEAT SINK IN 3D.....	11
5: HEATSINK WITH SEPARATED BASE.....	11
4: HEATING CELL.....	11
6: EQUIPMENT PHOTO 2.....	12
7: EQUIPMENT PHOTO 1.....	12
8: HEATSINK WITH HIGHLIGHTED BASE.....	15
9: MAP OF HEATING AREAS OF THE CHIP.....	15
10: SCHEME OF AXIS AND SURFACES' NAMES.....	16
11: ELECTRICAL SCHEME OF A SINGLE FINITE ELEMENT.....	16
12: THREE SUBDIVISION OF A FINITE ELEMENT FOR THE COMPUTATION OF THE CONDUCTANCE AMONG X, Y AND Z.....	17
15: VOLUME CONSIDERED FOR THE DECREASED AREA PARTITION.....	17
14: 3D SECTION FOR THE BASE CONDUCTANCE COMPUTATION WITH PYRAMID PARTITION.....	17
13: 3D SECTION FOR THE BASE CONDUCTANCE COMPUTATION WITH ZIGGURAT PARTITION.....	17
16: SIDE ELECTRICAL SCHEME WITH ZIGGURAT PARTITION.....	18
17: SINGLE VOLUME CONSIDERED WITH DECREASED AREA PARTITION.....	18
18: BASE'S TYPES OF SECTIONS EACH ONE DIVIDED IN FINITE ELEMENTS (ARRAY FOR SIDES, 2D MATRICES FOR FACES...).....	19
19: BASE'S SECTION PARTITION WITH AXES AND NAME'S EXAMPLES.....	19
20: HEATING AREAS.....	23
21: MODELICA CAUSAL BLOCK.....	23
22: POINT-POINT INSPECTION PROCEDURE'S STEP (SETTING ADIABATIC CELL).....	24
23: LINE-LINE INSPECTION PROCEDURE'S STEP (SETTING ADIABATIC LINE).....	24
24: EXAMPLE'S SCHEME WITH POSSIBLE CASES OF COVERING AREAS.....	25
25: ACAUSAL LCM PARTIAL CONNECTION SCHEME.....	26
26: SIDE SECTION OF CONNECTIONS BETWEEN DIFFERENT GRIDS.....	27
27: DIFFERENT FACING GRIDS (RED 3x3 WITH BLUE 2x2).....	27
28: SCHEME OF A FILE .FLP.....	31
29: BASE ICON IN MODELICA.....	35
30: ACTUAL CODE CONTAINED IN THE BASE MODEL (SAME FORMULATION OF PAGE X).....	35
31: CAUSAL BLOCK IN MODELICA.....	36
33: CODE EXAMPLE AFTER META-PROGRAM ACTION IN CAUSAL INTERFACE.....	36
32: CODE EXAMPLE BEFORE META-PROGRAM ACTION IN CAUSAL INTERFACE.....	36
34: ACAUSAL MODELICA BLOCK.....	37
35: CODE EXAMPLE BEFORE META-PROGRAM ACTION IN ACAUSAL INTERFACE.....	37
36: CODE EXAMPLE AFTER META-PROGRAM ACTION IN ACAUSAL INTERFACE.....	37
37: SCHEME OF HEATING SOURCES MAP DURING TESTS.....	41
38: MODELICA BLOCKS SETTING FOR 3x3x3 TESTS.....	41
39: TEST 3x3x3 (NO OFFSET) RESULTS.....	42
40: MODELICA POWER SINK BLOCK.....	42
41: TEST 3x3x3 (WITH OFFSET) RESULTS.....	43
42: COLOR SCHEME OF TEMPERATURES.....	44
43: TEST 3x3x3 AND 2x3x2 (WITH OFFSET) RESULTS.....	44

Abstract

The purpose of this thesis is the modeling of a thermal system for the simulation of a copper heat sink used as a cooling system for electrical components. We will focus on the dynamics that occur from the surface of the processor at the base of the heatsink fins.

The peculiarity of this work lies in the application of meta-programming methods for the formulation, and therefore, the writing in Modelica code, of equations for the interfaces linking models with different characteristics.

Will be presented the mathematical formulations of models, their implementation and their use in conclusive verification simulations.

All this with the aim of keeping each result adaptable to related thermal cases like the one considered.

Riassunto della Tesina

Lo scopo di questa tesina è la modellazione di un sistema termico per la simulazione di un dissipatore in rame usato come sistema di raffreddamento per componenti elettrici. In particolare, ci si focalizzerà sulle dinamiche che avvengono dalla superficie del processore alla base delle alette del dissipatore.

La peculiarità di questo lavoro risiede nell'applicazione di metodi di meta-programmazione per la formulazione, e quindi, la scrittura in codice Modelica, di equazioni per le interfacce di collegamento tra modelli con caratteristiche diverse.

Verranno presentati quindi le formulazioni matematiche di modelli, la loro implementazione e il loro utilizzo in simulazioni di verifica conclusive.

Tutto questo con l'obiettivo di mantenere adattabile ogni risultato a casi simili a quello considerato.

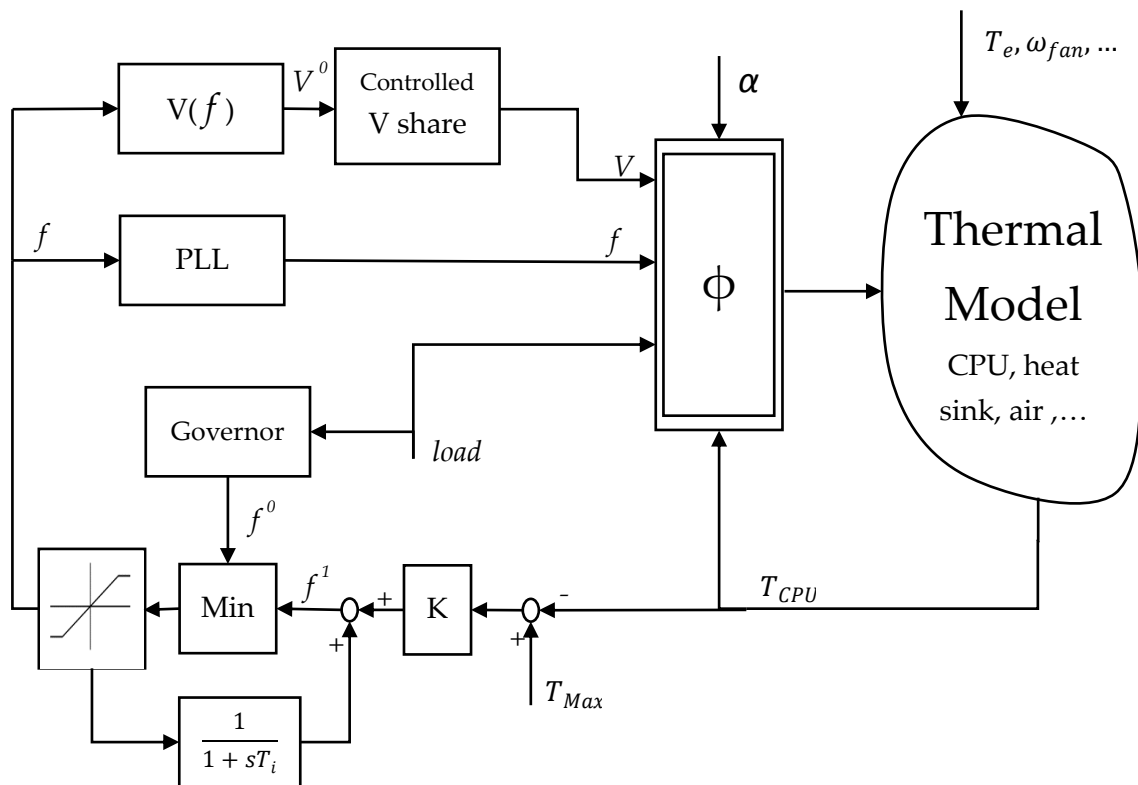
Part I
Introduction

Description of the complete system

Since the early 1980s the growing of new technologies and applications shifted the scientific interest on the power electronics field. The necessity of industry to develop micro devices with a high heat dissipation per unit of volume justifies the need of more compact and high heat flux cooling systems and more complex management of hardware allocation.

Nowadays research and development on pre-existent structure are aimed to maximize efficiency controlling power dissipation and improve computation management of programs of increasing complexity and computational weight.

In this thesis will be treated a part of a very complex system such the heat and software dynamics involving an operating CPU inside a computer. Here will be briefly described through an abstract and general scheme the contours of our problem, this is for conceptual purpose and lack of many details but is used to give a reasonable definition of the problem and neglects not fundamental information:

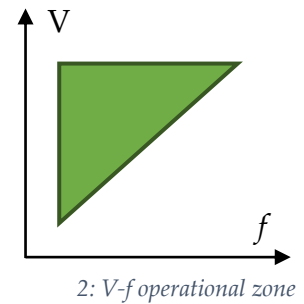


1: General System's Scheme

Starting from the top left is located the "Governor" that impose a frequency f^0 depending on the *load* of the actual computational request. Recently is been developed a new part of the system that take this frequency f^0 and compare it with another one f^1 coming from a block that coexist with the governor improving the performance. This second frequency is computed with a anti wind-up PI control scheme using the difference between the actual and the maximum CPU temperature. In this way the

frequency imposed to the system is not only based on the computational load but also to the actual CPU temperature.

At this point the frequency is fed in two section “PLL” (Phase Lock Loop) and a voltage controller that impose a voltage V^0 using a relationship like the one in figure 2. These two blocks are used to impose a relationship between frequency and voltage that improve operation done by the CPU, indeed being at high frequency with low voltage the processor won't work due to critical path failure. Frequency f , voltage V , computational load (% of not idle time), parameter α (Utilization Factor) and the CPU temperature are then fed to a block ϕ that generate a power flow P .



This power is input of a thermal model containing the CPU itself, the heatsink, the thermal interface material between them, the air and so on and its function of the external temperature, the presence and intensity of fan cooling action and so on.

This last part of the scheme is the one that will be further investigate.

Part of the system considered

Each power electronic device must be cooled to avoid excessive increase in temperature, which leads to failure of the device. Typically this kind of applications features high rates of heat generation and high-power densities, hence making straightforward the necessity of developing optimal cooling systems.

All the cooling procedures exploit the convection heat transfer phenomenon, which occurs between a fluid in motion and a bounding surface when there is a gap of temperature between them. According to the Newton's law of cooling.

$$q = Ah (T_s - T_1)$$

the convective heat power q is proportional to the difference between the surface and the fluid temperature, T_s and T_1 respectively. The proportionally constant h is called convection heat transfer coefficient and A is the heat exchange surface.

There are different possibilities in the structure of cooling devices, the most used are presented below.

The water-cooled and air-cooled systems are the most exploited in power electronics Applications.

Water-coolers are used commonly for cooling combustion engines in automobiles and electrical generators. The water flowing in the pipes transports away the heat from the heat source to a secondary cooling surface. These systems are usually quite expensive because they need a moving pump that has a finite life and must be periodically serviced. The water-cooling systems, compared to the air cooling systems, need less volumetric flow and reduced temperature differences for

transmitting heat over greater distances, having higher specific heat capacity and thermal conductivity.

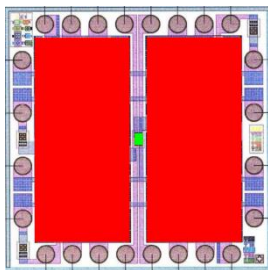
The convectional air-cooled systems, called heat sink, consist of an array of fins, extended from a plate base connected to the electronic device.

If a fan is used for increasing the air flow, the heat transfer phenomenon is due to forced convection and no longer to natural convection. The first purpose of the fin is to enhance the heat exchange surface, because, according to the Newton's law of cooling, at a fixed temperature difference the increase of surface A is proportional to the increase of heat flux. In fact, following this approach also the surface of each fin is enhanced usually by increasing the roughness of the surface.

However, reducing the thickness and increasing at the same time the fin surface we obtain a progressive reduction of the fin efficiency, due to the increase of variation of temperature in the extended surface.

In this thesis is been faced the modeling and the simulation of the thermal part of the described system. The major part of the work is focus to the part of the heat sink facing the CPU with the thermal gel and the array of fins on the other side.

Even if the problem is small, the complexity is however high, indeed the CPU dissipate heat in a non-homogeneous region and each of those is characterize by fast, high and highly variable power generation dynamics. For data collecting and simulation is been used a surrogate of the real system: in the figures 5 and 6 is reported the actual system that mimic a real CPU. In detail the equipment is composed by 16 power supplier (below),



5: Heating cell

a fan that can be used to add additional cooling action, a copper heat sink that will be the one considered in modeling examples, and beneath is located a special board containing 16 heating elements, each of which is made of 2 heating areas and a temperature sensor (central green dot in Figure 4). This board reproduce a record file in an SD card, this record is generated by a software tool that can emulate the power dissipated in a CPU on which is running a specific program or excite any combination of the heat generator for modeling tests. In this way is possible to use the heating chip as a controlled thermal actuator during experiment without relying on an unpredictable real processor.

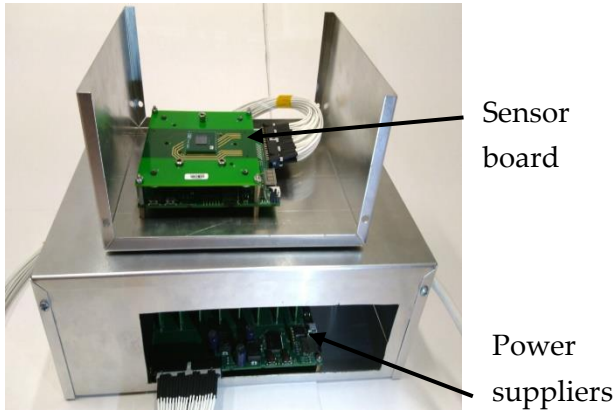
Moreover, having a "controlled" CPU processor that can reproduce and test real program execution is suitable for commercial purposes.

3: Heat sink in 3D

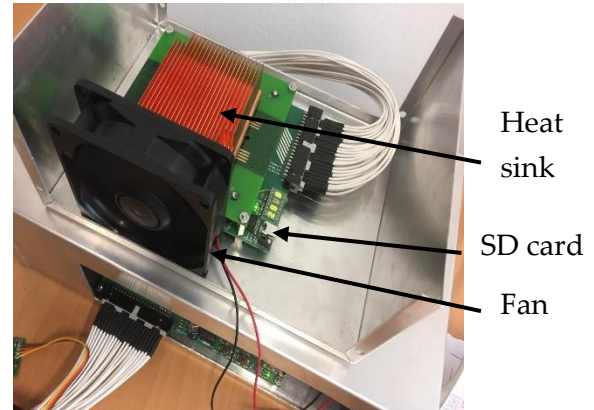


4: Heatsink with separated base

Modeling and simulation are been done using Modelica language with Open Modelica editor to compile and run the code, while CodeBlocks (C++ language) is been used to write some part of the code as explained in detail below(page 29) the rest of the programs is been wrote directly in Modelica language.



7: Equipment photo 1



6: Equipment photo 2

Technological Motivation

This thesis is collocated in a major study with the purpose to improve the performance of the actual implementation for the management of the hardware inside most of the computers. In particular, will be possible to deeply study the behavior of system that usually cannot be observed or controlled, with real and practical application in system running specific programs having, also, commercial purpose as explained before.

Concerning the objective of this work is to realize a solution for modelling the connecting part between the heat source and the heat sink's fins, so that future development will use this simulator to realize and study a control action like the one expose in the first chapter.

For this are fundamental the capability to adapt any solution to a scalable and complexity-variable model. The great challenge is not to face the trade-off of having a precise model but too much complex and computational slower than the system dynamics, or having a fast computational model but with high imprecision; instead will be to adopt easily scalable approach that has as result the lighter possible model, leaving the choice on how precise and heavy the simulator should be to future users.

Part II
Mathematical Model

Starting consideration

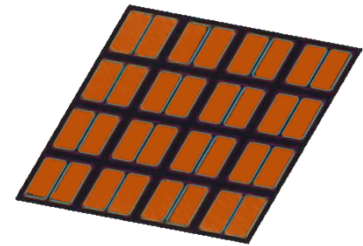
We will now consider the thermal behavior of the base part of the heat sink associable to a parallelogram (real dimension example: 60x60x2.8 figure 8) of copper.



8: Heatsink with highlighted base

Throughout this part will be considered adiabatic all the external faced since the dimension among the height is much inferior with respect the other dimension so the only conducting faces will the top and the bottom ones.

Moreover, the bottom part of the heat sink facing the 32-heat sources will be considered adiabatic if is not facing a power source (dark area in figure 9) meaning that the thermal gel between the processor and heat sink, principally due to its high thermal conductance and its thickness, is been neglected.



9: Map of heating areas of the chip

In the simulation will be used 16 training set of data, one of each heating cell; meaning that the power dissipated will be considered uniform on each couple of the 32 heating areas.

Is being considered that the thermal properties of the object remain constant with the varying of the temperature so density, dimensions, heat capacity and heat conductance are not function of the temperature.

Finite element methods

For the modelling part is been used a finite element discretization in which we decide to not fix any partition among any direction so to face this characteristic the volume will be divided in parallelogram with no fixed dimension but depending on the geometry of the object and the partition adopted. In the following will be used and referred for simplicity as "cubes" to identify a finite element but each direction (x,y,z) will be distinguished.

Concerning the physical description of the heat propagation is been considered a three directional heat propagation meaning that two adjacent cubes will exchange heat only through their adjacent faces and all the heat will be stored at the center of the cubes.

To describe and characterize the thermal dynamics is been adopter an electric equivalent, since will be considered the energy transfer linearly related to the difference of an effort variable and the energy storage depending on the derivative of the same effort variable, the temperature in our case.

The result as example for a single cube is:

$$\text{Heat Capacity: } C = \frac{\rho V}{c_p}$$

where:

$$\rho = \text{Copper Density } \left[\frac{\text{Kg}}{\text{m}^3} \right]$$

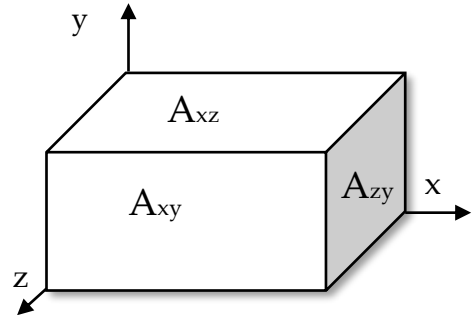
$$V = \text{Volume } [\text{m}^3]$$

$$c_p = \text{Specific Heat Capacity } \left[\frac{\text{J}}{\text{Kg} * \text{K}} \right]$$

$$\text{Heat Conductance among X: } G_x = \lambda \frac{A_{yz}}{s_x}$$

$$\text{Heat Conductance among Y: } G_y = \lambda \frac{A_{zx}}{s_y}$$

$$\text{Heat Conductance among Z: } G_z = \lambda \frac{A_{yx}}{s_z}$$



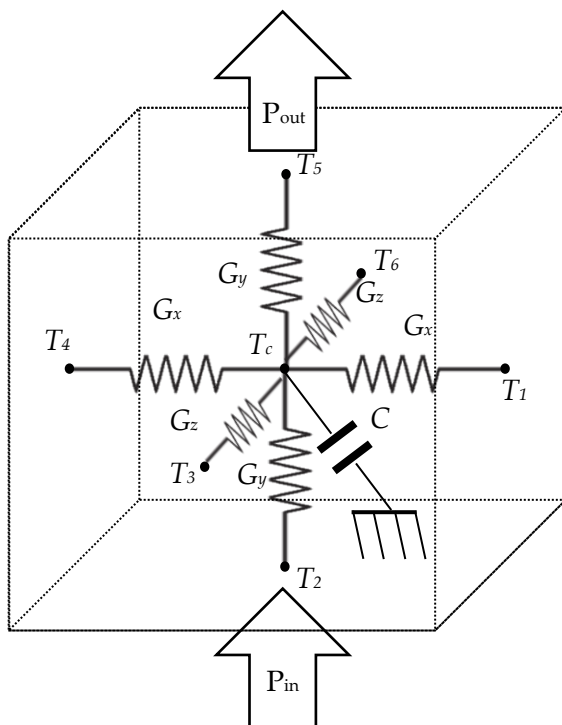
10: Scheme of axis and surfaces' names

where:

$$\lambda = \text{copper thermal conductivity } \left[\frac{\text{W}}{\text{m} * \text{K}} \right]$$

A_{ij} [m^2] and s_k [m] for each direction x, y and z are the area and half of the thickness of each element.

Assuming a power flow entering from the bottom and exiting from the top the electric equivalent element (figure 11) has the following equations:



11: Electrical scheme of a single finite element

$$\begin{aligned} C\dot{T}_c = P_{in} - P_{out} - \\ - G_x * (T_1 - T_c) - G_x * (T_c - T_4) - \\ - G_y * (T_5 - T_c) - G_y * (T_c - T_2) - \\ - G_z * (T_6 - T_c) - G_z * (T_c - T_3) \end{aligned}$$

This formulation is a general form and cannot be applied among all the elements of the cubes, but it will be adapted depending on the section of the object considered. Will be described deeply this aspect below (page 19).

Study of the approximations

Using the previous discretization are introduced two approximation:

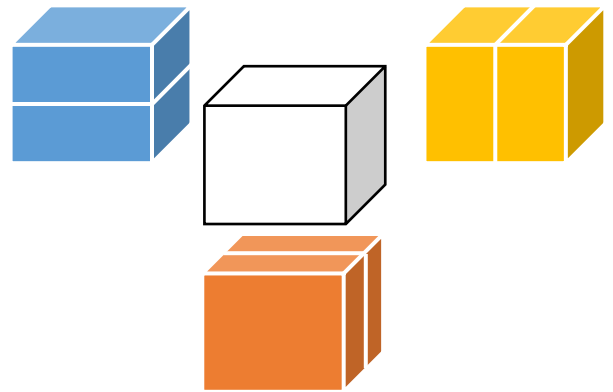
- The heat is stored in a single point at the center of the cube
- The thermal resistance between two cubes is calculated using half of the cube volume meaning that the same portion of the cube is been considered once for each direction(as showed in figure 12)

The first approximation can be reasonably acceptable since all the heat capacity of the object is partitioned in smaller capacity, so the overall value is the same, with finer partition this solution tends to the real one.

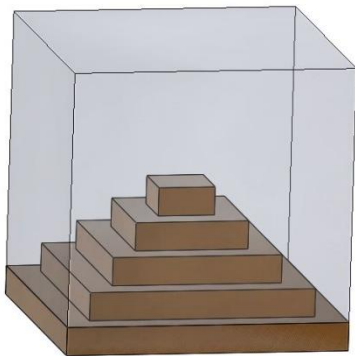
For the second approximation we try to improve the computation of the thermal conductance of each direction (G_x , G_y and G_z), searching for a better approximation even if involves complex computation . This is because since the thermal conductance is constant and equal for each cube the overall system is not affected from this calculus.

We then try to minimize the overlapping of the volume used for the thermal conductance, so we follow with different partition of the same element resulting in three attempts:

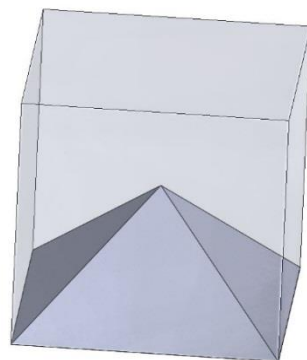
- “Ziggurat” based partition (Figure 13)
- “Pyramid” based partition (Figure 14)
- Decreased area partition (Figure 15)



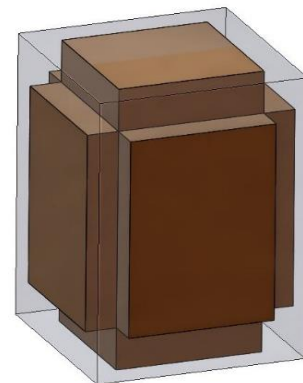
12: Three subdivision of a finite element for the computation of the conductance among x , y and z



15: 3D section for the base conductance computation with Ziggurat partition



14: 3D section for the base conductance computation with Pyramid partition



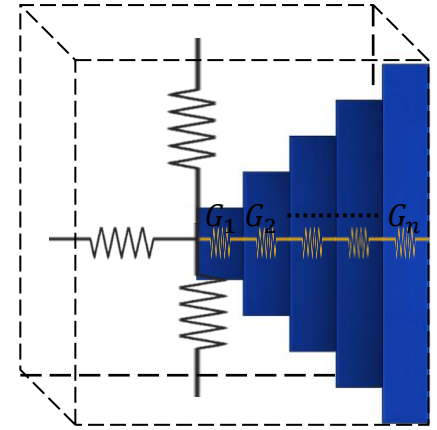
13: Volume considered for the decreased area partition

Ziggurat-based partition

The idea was to divide the internal part in 6 “Ziggurat” structure with the smallest top level at the center like in the figure 6. This method follows the principle of serial conductance calculated on decreasing areas levels.

$$G_i = \sum_{levels} \lambda \frac{A_{jk}}{S_i}$$

The problem that we encountered is that with decreasing the area the conductance tends to zero so the conductance close to the center reach too small value very far from the correct value leading to an incorrect model



16: Side electrical scheme with Ziggurat partition

Pyramid based partition

As well as for the previous one computing the conductance using the defined integral, we have the confirm that this partition cannot be faced with these formulae

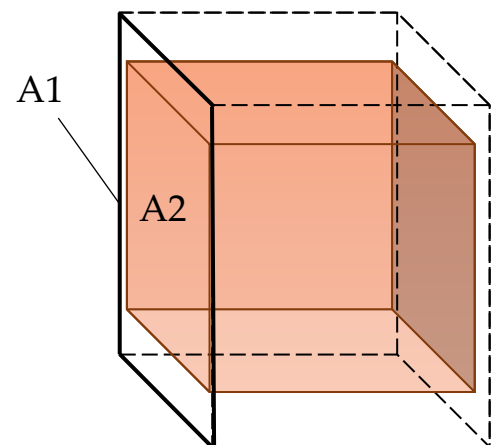
$$G_i = \int_0^{S_i} \lambda \frac{A(x)}{x} \xrightarrow{x \rightarrow 0} \infty$$

This partition has the good property of having no overlaps in the internal subdivisions. Unfortunately the applied formula is not correct with decreasing area geometry.

Decreased area partition

This partition born from the concept that all the cube’s half-volume could overlaps but to come closer to the real conductance could be possible to consider a smaller area A2 from the cubes one A1, then calculate a decreased conductance. The only problem is that is not possible to estimate the rate between the area A1 and A2 without aimed tests. With a different test case should be possible study this relationship also depending on the partition choose for the finite element discretization. Only knowing the real value of the thermal conductance this approach can be improved and used.

The result of this partition applied to all the faces look like as in figure 15 or in figure 17 showing one direction subdivision.



17: Single volume considered with decreased area partition

From the above consideration we decided to keep the standard approximation since the last one requires a different study far from our test case while the first two request different mathematical formulation, a task far from our objective. Nevertheless this aspect can be improved in the future with further study updating the result of our work.

Electrical model result

The result of the previous approach is a Modelica program for the thermal simulation of the base of the heat sink. Such program will not be reported here but only its structure with some equation as example. To face the programming of this part we proceed with order defining a name for each face like for a dice and splitting the base in 4 type of section:

- 8 Edges (“Spigoli”): numbered as the intersection of faces that generate them
- 12 Sides (“Lati”): numbered as the couple of face that generate them
- 6 Faces (“Faccie”): from 1 to 6
- 1 Core all the rest of the cube inside

The difference among these sections lays in the type of equations that have to be written for each of them, moreover each element of a type is slightly different from the one of the same types. Meaning that it was not possible to write the equation for all the edges in only one cycle for example.

Here below are reported some example for each type of element and a couple of sides to appreciate the difference between two elements of the same type.

Is also reported the presence of power flow at the bottom and at the top but not on the side as previously described.

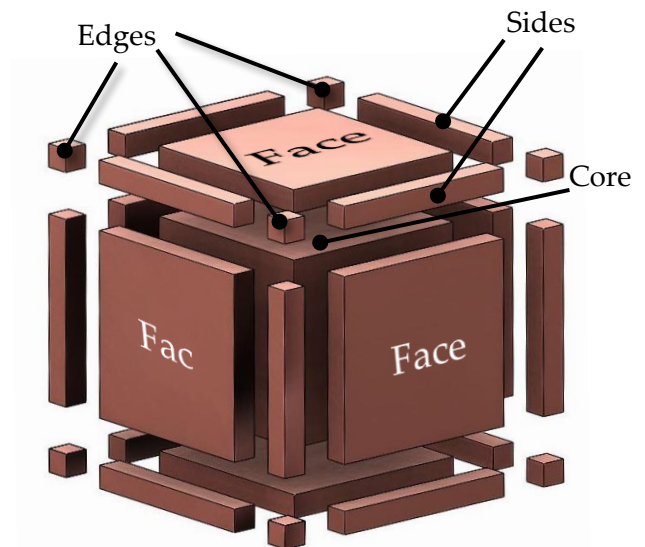
Side 21 equation:

$$\begin{aligned}
 CT_c[x_i, 1, 1] = & P_{in}[x, 1] - \\
 & G_x * (T[x_{i-1}, 1, 1] - T[x_i, 1, 1]) - \\
 & G_x * (T[x_i, 1, 1] - T[x_{i+1}, 1, 1]) - \\
 & G_y * (T[x_i, 1, 1] - T[x_i, 2, 1]) - \\
 & G_z * (T[x_i, 1, 1] - T[x_i, 1, 2])
 \end{aligned}$$

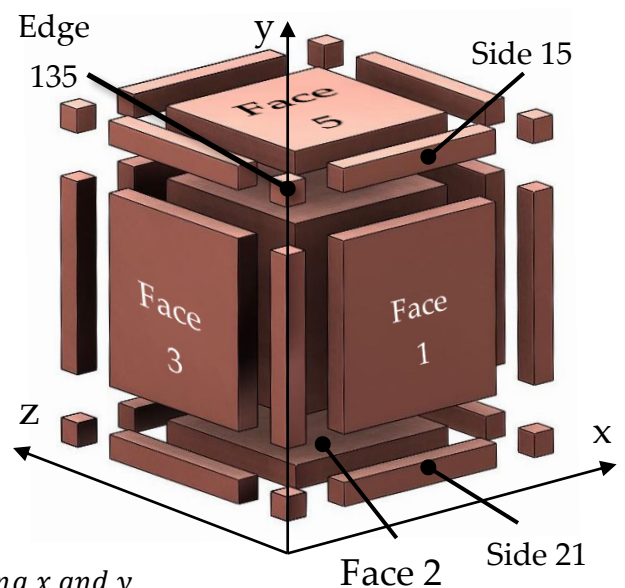
Side 15 equation:

$$\begin{aligned}
 CT_c[x_i, ny, 1] = & P_{out}[x, 1] - \\
 & G_x * (T[x_{i-1}, ny, 1] - T[x_i, ny, 1]) - \\
 & G_x * (T[x_i, ny, 1] - T[x_{i+1}, ny, 1]) - \\
 & G_y * (T[x_i, ny - 1, 1] - T[x_i, ny, 1]) - \\
 & G_z * (T[x_i, ny, 1] - T[x_i, ny, 2])
 \end{aligned}$$

wehre $x_i = 2 \dots nx$ and nx, ny are the division among x and y



18: Base's types of sections each one divided in finite elements (array for sides, 2D matrices for faces...)



19: Base's section partition with axes and name's examples

Part III
Interfaces

Introduction

Now that the central part of the simulation has been explained, will be shown the 2 interfaces used to simulate correctly the interaction of the base with the processor and the fins.

The first interface is called “Causal” and is used to map and correctly direct the sources from the processor.

The second is called “Acausal” and is created to interface two objects with the same contact surface but with different grids size.

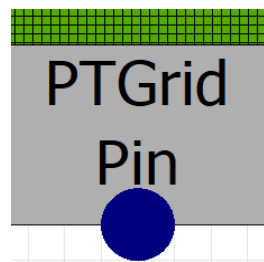
We’ll proceed to detailed description of both interfaces.

Causal interface

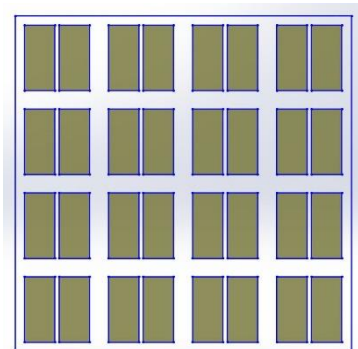
This interface receives as input a vector of data ($S[i]$ with $i=16$ in our case) coming from experiment and simulation with equipment and has as output a Power and Temperature 2D grid (PTgrid) with the same dimension of the bottom of the thermal base model.

The connection between the PTgrid of the Causal interface and the PTgrid of the base in our experiment will impose the power flow from the Causal block and the temperature will be set from the above block leading to a balanced system of equation.

Here is reported (figure 20) the Modelica block of the system created using the blocks mentioned and the geometry of the distribution of the power sources for our experiment (figure 21), of course any regular plane geometry can be used.



21: Modelica Causal block



20: Heating areas

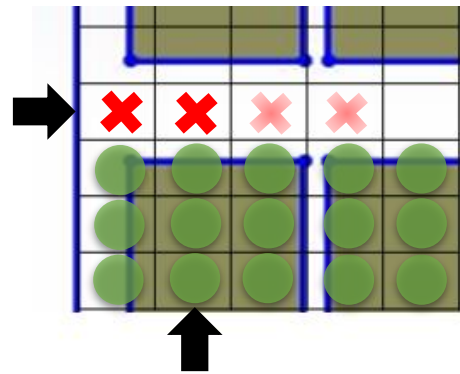
To solve this part are been designed three solution:

- Point-point and inspection;
- Line-Line inspection;
- Covering Coefficient approach;

Point-point and inspection approach

For this solution we used the geometry of the sources mapped with their edge's coordinates. In this way we can examine all the point using two for cycle and when we found a grid point between two source coordinates among a direction ($X^{1_{si}} < x < X^{2_{si}}$) and the other ($Y^{1_{si}} < y < Y^{2_{si}}$), then we impose the connection among that source and the PTgrid port.

A scheme of a possible iteration during the geometry inspection for the connection creation in figure 22.



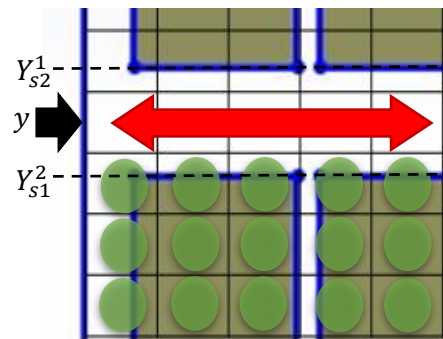
22: Point-Point Inspection procedure's step (setting adiabatic cell)

Line-Line inspection approach

This solution improves the precedent since we notice that the geometry of the source can be used to connect quickly some PTgrid port.

To do this instead of examining all the line (for each y) and then all the point (scrolling among x) we consider all the line that don't satisfy $Y^{1_{si}} < y < Y^{2_{si}}$ directly adiabatic while we scroll among the x all the others as before. In this way we set directly more port adiabatic instead of one at a time (figure 23).

This solution optimizes slightly the connection instantiation that for certain geometry can results advantageous but is also true that in some geometry this don't give any benefits.



23: Line-Line Inspection procedure's step (setting adiabatic line)

Coefficient approach

This approach is very different from the other and require a condition to be applied. Suppose to have a coefficient that indicates the percentage of area of each source covered by a PTgrid cell, in this way should be possible to write the power to connect to each port like a linear combination of the source's data in input. Still remain a problem, meaning that having all the coefficient stored in a matrix of variable length vectors (list of coefficients for each PTgrid cell) in Modelica is not easy to calculate such coefficients and is not possible to write equation using variable length vector.

To solve this issue we adopt a metaprogramming approach that will be describe exhaustively below in the following section (page 31).

The result is a block that has directly wrote inside, from a program, the equation of the powers of each PTgrid port.

The formulation and the definition of the coefficients are :

$$B.P[i,j] = \sum_K \frac{A_{S_k(i,j)}}{A_c} * P_{S_k}$$

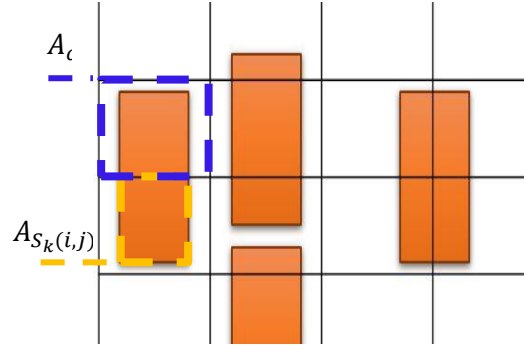
Where:

$B.P[i,j]$: Power flow of a cell i,j -th of a PTgrid "B" (for bottom)

$A_{S_k(i,j)}$: Portion of area of a k -th source (S_k) covered by a cell i,j -th

A_c : Area of each cell

P_{S_k} : Power coming from a k -th source (S_k)



24: Example's scheme with possible cases of covering areas

Since all the sources area must be covered by at least one cell is holds :

$$\sum_{i,j} \frac{A_{S_k(i,j)}}{A_c} = 1 \quad \forall k$$

Acausal interface

Is used to connect two grids with different partition so has two PTgrid as connection port.

This interface is important for our test case because the two body that will connect (base and fins) can have different partition since the geometry and the thermal dynamics of the two objects are usually very different. Is also true that in other thermal case in which is used a finite element partition this interface can be useful.

For the realization of this interface two solution are proposed.

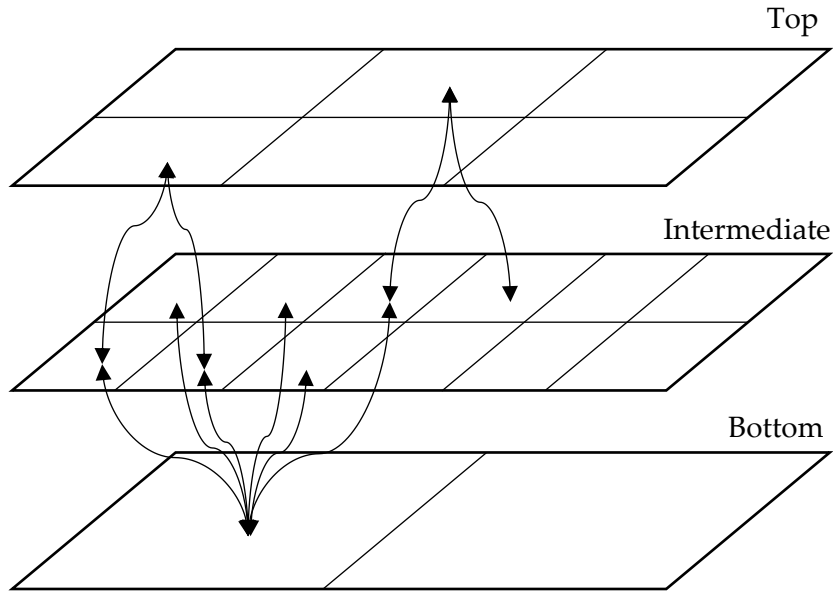
- "lcm" approach;
- Area Coefficient approach;

Least Common Multiple approach

Consider having two PTgrids "B" (standing for bottom) and "T" (for top) with division among x and y respectively: X_B, Y_B and X_T, Y_T .

We can think to introduce an intermediate "I" PTgrid with the same dimension of the others but with partition among x and y with the least common multiple of the partition of T and B such that: $X_I = \text{lcm}(X_B, X_T)$ and $Y_I = \text{lcm}(Y_B, Y_T)$

Now is important to instantiate the relationship between B-I and I-T following the idea that the Temperature ($B.T(i,j)$) of a boundary PTgrid will be equal for each cell of I covered by a cell of B or T while Powers ($B.P(i,j)$) will be distributed from a cell of B or T equally to all the cell of I covered by them. Is reported an example where $B=1 \times 2$ and $T=2 \times 3$ (figure 25)



25: Acausal lcm partial connection scheme

It's possible to write down the mathematical formulation of this solution:

$$B = X_B \times Y_B$$

$$T = X_T \times Y_T$$

$$I = X_I \times Y_I: XI = mcm(XB, XT) \wedge YI = mcm(YB, YT)$$

B-I formulation:

$$\# \text{ of cell of } I \text{ covered by a cell of } B \text{ among } X : X_{BI} = \frac{X_I}{X_B}$$

$$\# \text{ of cell of } I \text{ covered by a cell of } B \text{ among } Y : Y_{BI} = \frac{Y_I}{Y_B}$$

$$I.T[(h-1)X_{BI} + i, (k-1)Y_{BI} + j] = B.T[h, k]$$

$$I.P[(h-1)X_{BI} + i, (k-1)Y_{BI} + j] = B.P[h, k]/(X_{BI} * Y_{BI})$$

with $i = 1, \dots, X_{BI}$ $j = 1, \dots, Y_{BI}$ $h = 1, \dots, X_B$ $k = 1, \dots, Y_B$

I-T formulation

$$\# \text{ of cell of } I \text{ covered by a cell of } T \text{ among } X : X_{TI} = \frac{X_I}{X_T}$$

$$\# \text{ of cell of } I \text{ covered by a cell of } T \text{ among } Y : Y_{TI} = \frac{Y_I}{Y_T}$$

$$I.T[(h-1)X_{TI} + i, (k-1)Y_{TI} + j] = T.T[h, k]$$

$$I.P[(h-1)X_{TI} + i, (k-1)Y_{TI} + j] = T.P[h, k]/(X_{TI} * Y_{TI})$$

with $i = 1, \dots, X_{TI}$ $j = 1, \dots, Y_{TI}$ $h = 1, \dots, X_T$ $k = 1, \dots, Y_T$

It's also possible to reformulate the problem looking at the boundary PTgrid as sum of the covered I Powers and mean of the covered temperature.

This solution would be feasible but has the disadvantages to introduce additional state variable not needed, more over with certain combination of partition the complexity would explode (i.e.: $B = 17 \times 18$ and $T = 19 \times 21 \rightarrow I = 323 \times 126$). Of course this detail can be handled in a wise choice of the partition, but this was unsatisfying.

Area coefficient approach

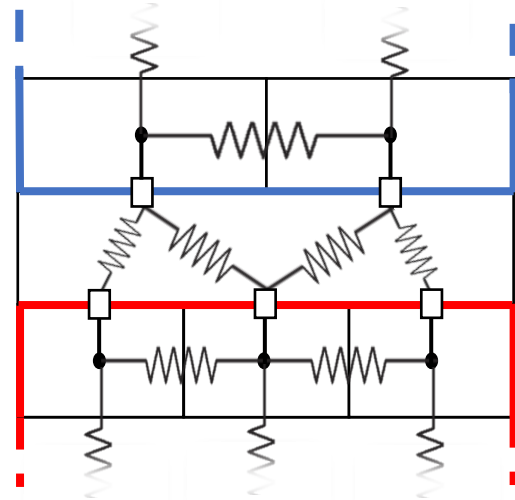
Like the third solution for the Causal interface, imagine having a coefficient that describe the area in common between two PTcell for each couple of facing cell.

Of course the two grids must have the same dimension but with any kind of partition.

A scheme for an example of such problem could be the one in figure 26 where a grid 2x2 (blue) face a grid 3x3 (red) see from the Top/Bottom and the circuitual appearance see in a side section in figure 27.

A11	A12	A21	A23
A14	A15	A25	A26
A34	A3	A45	A46
A37	A38	A48	A49

27: Different facing grids (red 3x3 with blue 2x2)



26: Side section of connections between different grids

In this formulation we will link the two body with special conductance calculated using apposite coefficient. Reminding the formula of the conductance:

$$G = \lambda \frac{A}{s}$$

Can be adapted to the scheme above reminding that the resulting conductance will be the series (parallel formula for conductances) of two conductance: one from half distance from the bottom and one from half distance of the top:

$$G_{ij} = \frac{1}{\frac{\lambda_i A_{ij}}{s_i}} + \frac{1}{\frac{\lambda_j A_{ij}}{s_j}}$$

Where is not assumed that the two body are made of the same material ($\lambda_i \neq \lambda_j$)

A_{ij} is the common portion of area of the two facing cell ij

s_i, s_j are half distance from the center of each element.

Manipulating the formula can be written like:

$$G_{ij} = \frac{s_i s_j}{s_i \lambda_j + s_j \lambda_i} * \frac{1}{A_{ij}} = \frac{G_{base}}{A_{ij}}$$

In this way is even clear that the problem can be handled easily containing the difference among different G_{ij} in the area A_{ij} .

At this point having a conductance G_{ij} wherever two cell overlap allow us to write the equation of the powers between the PTgrid port [B,T] of the Acausal block:

$$B.P[i] = \sum_{j \text{ covered by } i} G_{ij} * (T.T[j] - B.T[i])$$

For clearness and lightness of notation is been used a numeration of single subscript ($i = 1, \dots, 4$) having $G_{ij} \rightarrow G_{15}$ instead of $G_{i,j} \rightarrow G_{1,1} = G_{15}$ used in the programs

As before the explanation on how we determine the area A_{ij} is contained in the following chapter on metaprogramming.

Part IV

Metaprogramming

Introduction

During the work was always needed some programming freedom that Modelica in some operations do not permit.

We than decide to use a metaprogramming approach to face this problem, in general we build one program using C++ language that take as input a Modelica model file (.mo) and generate a second one containing some precise modification when the metacompiler finds a metaconstruct that in our implementation is marked by the @ character

We apply this general procedure to solve two problems already mentioned:

- In the Coefficient approach in Causal interface
- In the Area coefficient approach in Acausal interface

We will explain in detail for both approach all the requirement and information needed to generate blocks previously described. The result of using meta-construct are been already formulated at page 24 (causal formulation) and pages 27-28 (acausal formulation); while examples are reported below at pages 35-36 (causal result) and pages 36-37 (acausal result)

Coefficient approach with metaprogramming

As said before inside the Causal interface we use the coefficient that indicate the area of a source covered by a PTcell.

Firstly how the meta-construct is recall: imagine to have a Modelica block already instantiated with a vector input data port "IN" (data coming from experiment in our case) and a PTgrid "OUT" with partition (X_o by Y_o) as output.

The correct syntax for the meta-construct is:

```
//@causalmaptiletouniformgrid("Name.flp",IN,OUT,Xo,Yo)
```

(Space between '@' and 'caus...' are irrelevant but allowed for clearness and order)

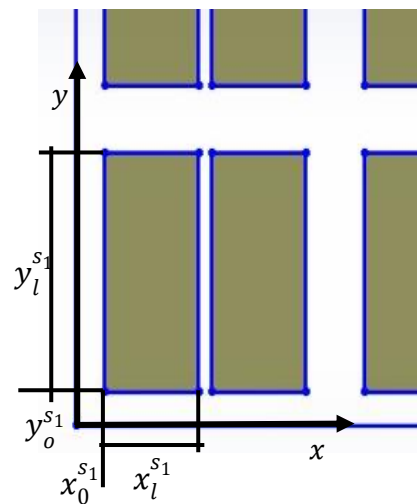
"Name.flp" is a file containing the listed sources organized like in figure 28;

IN and OUT are the name used from the program to write correctly the equations;

X_o and Y_o are the partition of OUT, is implicit that the dimension of OUT and the mapped sourced are the same or some of them will be cut.

Using this syntax when the file is passed to our program it will identify all the needed information to write the equations and where write them, so the position of this string is also important.

28: Scheme of a file .flp



$$\begin{aligned}
 s_i &: x_0^{s_i} \ y_0^{s_i} \ x_l^{s_i} \ y_l^{s_i} \\
 &\quad \downarrow \\
 s_1 &: x_0^{s_1} \ y_0^{s_1} \ x_l^{s_1} \ y_l^{s_1}
 \end{aligned}$$

Area coefficient approach with metaprogramming

For the acausal interface the structure and the behavior of the metaconstruct are the same but the information and how are used change.

The syntax for the meta-construct is:

```
// @acausaluniformagrids(Bottom,Top,X,Y,ZT,ZB,λT, λB,#XT,#YT,#XB,#YB)
```

Bottom and Top are the name of the PTgrids of the block

X and Y are the dimension of the grids

Z_T,Z_B : are the half thickness of an element of the Top and Bottom objects

λ_T, λ_B are the thermal conductivity of the material of the Top and Bottom objects

#X_T,#Y_T,#X_B,#Y_B are the number of divisions among X and Y of the Top and Bottom objects.

With the above data the program will compute the generic conductance between the two object and the areas covering coefficient as formulated at page 28.

Part V
Blocks in Modelica

Block summary

All the previous solutions are been applied to the following Modelica blocks. Colors and designs have been set and defined by us, so all the following blocks are been created from scratch.

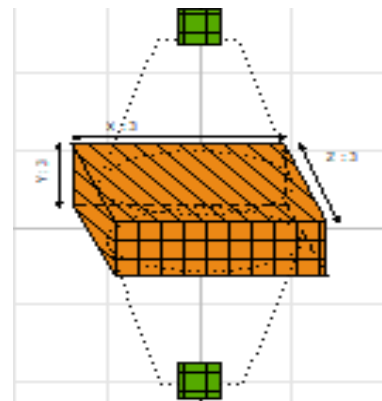
Base Block

This block is characterizing by two PTgrids (green square grids): one for the bottom and one for the top used to simulate the flowing power from the sources below and the fins above, under the assumption of adiabatic lateral sides.

It's possible to modify dimension and partition among all three direction as well as all the other parameter (heat conductivity, heat capacity...) using the window reported in figure 29.

Inside it is possible to see (in figure 30 is reported a short part) the implementation of all the possible type of series of elements (Edges , Faces, Core and Sides). They all follow

the same base equation but, as is possible to see in the figure and reported in formulae above (page 19) for two examples of Sides, the same type of element requires a specific indexing that make the programming and verification of this block cumbersome.

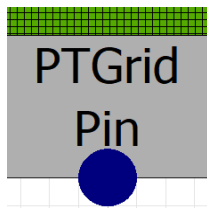


29: Base icon in Modelica

```
816 //Edge456 v
817   c * der(T[nx, ny, nz]) = (-Pout.P[nx, nz]) + gx * (T[nx - 1, ny, nz] - T[nx, ny, nz]) + gy * (T[nx,
ny - 1, nz] - T[nx, ny, nz]) + gz * (T[nx, ny, nz - 1] - T[nx, ny, nz]);
818 //Edge156 v
819   c * der(T[nx, ny, 1]) = (-Pout.P[nx, 1]) + gx * (T[nx - 1, ny, 1] - T[nx, ny, 1]) + gy * (T[nx, ny -
1, 1] - T[nx, ny, 1]) - gz * (T[nx, ny, 1] - T[nx, ny, 2]);
820 //Side21 v
821   for x in 2:nx - 1 loop
822     c * der(T[x, 1, 1]) = Pin.P[x, 1] + gx * (T[x - 1, 1, 1] - T[x, 1, 1]) - gx * (T[x, 1, 1] - T[x +
1, 1, 1]) - gy * (T[x, 1, 1] - T[x, 2, 1]) - gz * (T[x, 1, 1] - T[x, 1, 2]);
823   end for;
824 // Side 15 v
825   for x in 2:nx - 1 loop
826     c * der(T[x, ny, 1]) = (-Pout.P[x, 1]) + gx * (T[x - 1, ny, 1] - T[x, ny, 1]) - gx * (T[x, ny, 1] -
T[x + 1, ny, 1]) + gy * (T[x, ny - 1, 1] - T[x, ny, 1]) - gz * (T[x, ny, 1] - T[x, ny, 2]);
827   end for;
828 // Side 54 v
```

30: Actual code contained in the base model (same formulation of page 19)

Causal Block



This block is characterized by a “RealVectorInput” port for data values (blue circle) and a PTgrid port to interface with the base (green rectangle grid).

31: Causal block in Modelica

Of this and the following block exist two version:

- A version that is not been still modified by our programs so do not contain any equation (example in figure 32)
- The same block that is been updated by our program (example in figure 33)

For this block the parameters must be set directly inside since our meta-construct cannot receive parametric information expressed as parameter in Modelica.

```

block CausalP3x3
  //      z /
  //      /a
  //      /
  //      /_____b_/_____>x
  Modelica.Blocks.Interfaces.RealVectorInput Pin[9] annotation( ...);
  Tesi.PTGrid Source(nx = 3, nz = 3) annotation( ...);
equation
  // @causalmaptiletouniformgrid("3x3source.flp",Pin,Source,3,3)
  annotation( ...);
end CausalP3x3;

```

33: Code example before meta-program action in Causal interface

```

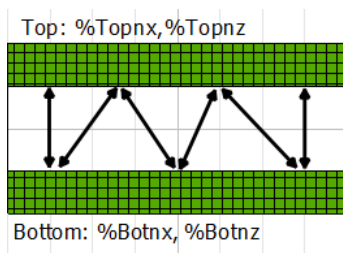
Tesi.PTGrid Source(nx = 3, nz = 3) annotation( ...);
equation
  //Causalmaptiletouniformgrid(3x3source.flp,Pin,Source,3,3) begin
  Source.P[1,1]=0+Pin[1]*1;
  Source.P[1,2]=0+Pin[4]*1;
  Source.P[1,3]=0+Pin[7]*1;
  Source.P[2,1]=0+Pin[2]*1;
  Source.P[2,2]=0+Pin[5]*1;
  Source.P[2,3]=0+Pin[8]*1;
  Source.P[3,1]=0+Pin[3]*1;
  Source.P[3,2]=0+Pin[6]*1;
  Source.P[3,3]=0+Pin[9]*1;
  //end
  annotation( ...);
end SourceP3x3;

```

32: Code example after meta-program action in Causal interface

The equations start with “0+” because it must be possible that a PTcell is not “connected” to any power source, most commonly because it doesn’t cover any supply geometry; in those case the PTcell is assumed adiabatic.

Acausal Block



34: Acausal Modelica block

Similarly to the previous one exists two version of this block one processed and on that is not. It's possible to set division of the two PTGrids (Top, Bottom) but only for graphical appearance. As before the actual parameter for the meta-construct must be set inside the block.

Here is reported the effect of our program with an example:

```

block Acausal3x3to3x1
  Tesi.PTGrid Bottom(nx = 3, nz = 3) annotation( ...);
  Tesi.PTGrid Top(nx = 3, nz = 1) annotation( ...);
equation
  // @acausaluniformgrids(Bottom,Top,0.12,0.12,0.4,0.4,1,1,3,3,3,1)
  annotation( ...);
end Acausal3x3to3x1;

```

35: Code example before meta-program action in Acausal interface

It's possible to see that the equations are written in a procedural way as expressed in the formulation at page 28.

```

# Tesi.PTGrid Top(nx = Topnx, nz = Topnz) annotation( ...);
equation
  //acausaluniformgrids(Bottom,Top,0.12,0.12,0.4,0.4,1,1,3,3,3,1) begin with Gbase =0.2
  Bottom.P[1,1]=0+125*(Top.T[1,1]-Bottom.T[1,1]);
  Bottom.P[1,2]=0+125*(Top.T[1,1]-Bottom.T[1,2]);
  Bottom.P[1,3]=0+125*(Top.T[1,1]-Bottom.T[1,3]);
  Bottom.P[2,1]=0+125*(Top.T[2,1]-Bottom.T[2,1]);
  Bottom.P[2,2]=0+125*(Top.T[2,1]-Bottom.T[2,2]);
  Bottom.P[2,3]=0+125*(Top.T[2,1]-Bottom.T[2,3]);
  Bottom.P[3,1]=0+125*(Top.T[3,1]-Bottom.T[3,1]);
  Bottom.P[3,2]=0+125*(Top.T[3,1]-Bottom.T[3,2]);
  Bottom.P[3,3]=0+125*(Top.T[3,1]-Bottom.T[3,3]);
  Top.P[1,1]=0+125*(Bottom.T[1,1]-Top.T[1,1])+125*(Bottom.T[1,2]-Top.T[1,1])+125*(Bottom.T[1,3]-Top.T[1,1]);
  Top.P[2,1]=0+125*(Bottom.T[2,1]-Top.T[2,1])+125*(Bottom.T[2,2]-Top.T[2,1])+125*(Bottom.T[2,3]-Top.T[2,1]);
  Top.P[3,1]=0+125*(Bottom.T[3,1]-Top.T[3,1])+125*(Bottom.T[3,2]-Top.T[3,1])+125*(Bottom.T[3,3]-Top.T[3,1]);
  //end
# annotation( ...);

```

36: Code example after meta-program action in Acausal interface

Part VI
Tests and Conclusions

Test summary

For the verification of the work done are been performed 3 type of test:

All are not used to validate the model with real data in input since is still too early for such level of experiment, however, is possible to show that all the block created are working. In the following tests are been set inputs and non-realistic parameters in order to be able to predict the result easily and compare the result with the expect behavior.

3x3x3 Test with sinusoidal (0 offset)

We supply a sinusoidal power flow to a base block divided 3x3x3 so that all the 4 types of element (Edges , Faces, Core and Sides) where present.

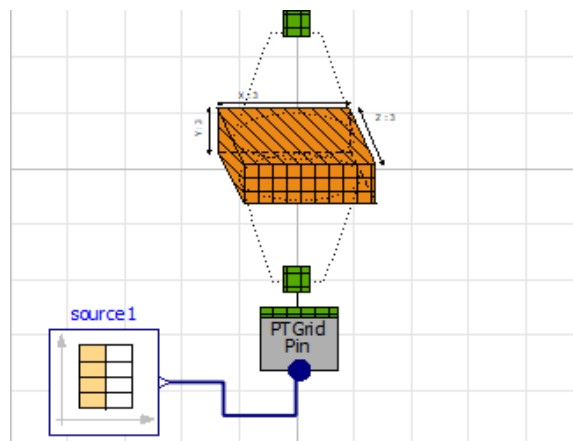
The sinusoidal power is directed with a floor plan like in figure 37, so only one cell of the PTgrid is active, in this way it will be possible to appreciate the temperature distribution inside the Base block.

Is been choose a zero offset sinusoidal power flow since the aim of this test is to appreciate the effective action of the RC circuit nest of the 3x3x3-base block, in this way the block is completely adiabatic except for the input power, so is necessary to use a zero mean signal in order to do not insert additional block to prevent a temperature drift.

In Modelica the blocks present and their setting were:

1	2	3
4	5	6
7	8	9

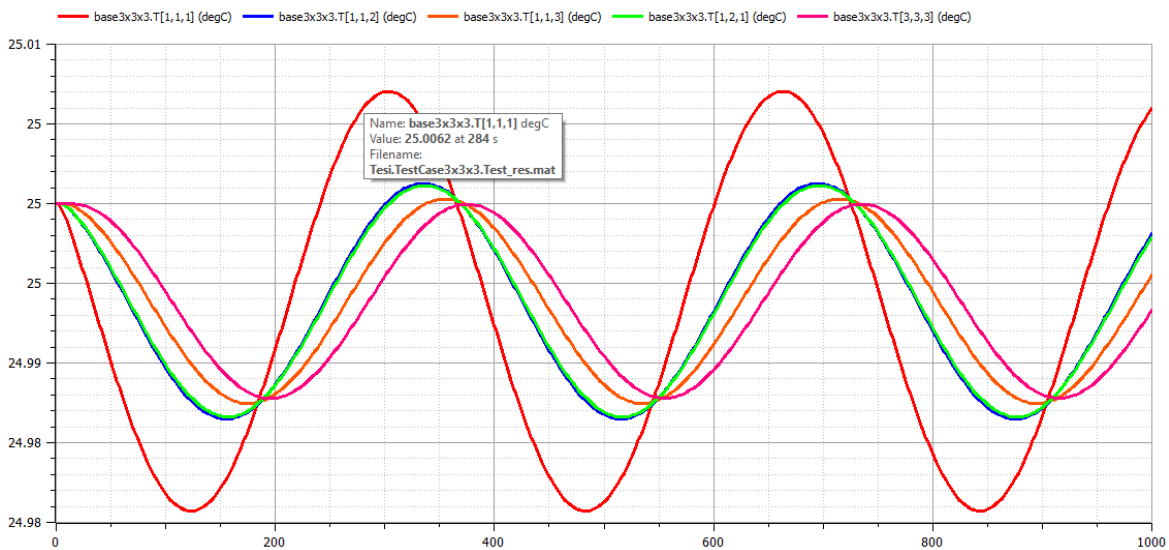
37: Scheme of heating sources map during tests



38: Modelica blocks setting for 3x3x3 tests

Results

39: Test 3x3x3 (no offset) results



In the above graph are represented the behavior of 5 different temperature cells of the base. Are showed the one directly supplied in the test (red), then two adjacent to the previous (Blue and Green) a third at 2 cell distance (Orange) and at last the most far cell from the first one so in in the higher opposite corner (Purple).

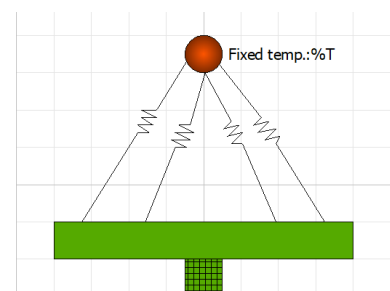
As expected, the first temperature had the greater amplitude and is not phased from the actual driving power. The blue and green line are correctly overlapped since the signal that reach them traverse a symmetric RC circuit. The effect of all the sinusoidal are that flowing inside the RC 3D circuit are a diminished amplitude and an increased phased as expected from such circuit. This confirm that the base block is correctly build.

Test 3x3x3 (with offset)

The previous test highlights the RC circuit behavior, now in the second test the focus will be on the temperature distribution when a power flow establishes through a base block.

Unfortunately due to the non-realistic parameter is been chosen a low value for the thermal capacity. Indeed, in order to perform a test where the behavior of interest could be seen even if the number of partitions is low and at the same time having all the previous effects was too cumbersome.

To realize this experiment is been added a block acting like a power sink that could be identified by the fins' action. The mentioned block is a rough but functional implementation where a PTgrid port is attached to a fixed temperature (25°C for the test) with a conductance.



40: Modelica power sink block

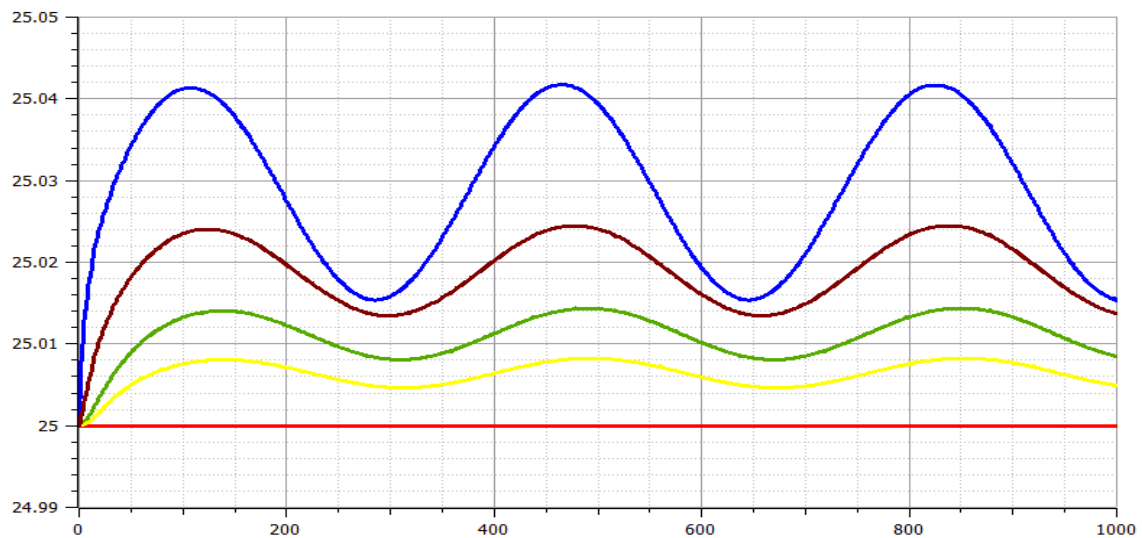
Result

In the following graph is represented the temperature of different point inside of the base block: starting from the blue line, that is the temperature of the element directly reached by the power source, all the line below are the temperature of adjacent and further blocks up to the red line that is the temperature of a element at the surface of the base directly attached to the power sink.

The last line is at 25°C showing the ambient temperature and allowing to appreciate the trend of the temperature going from the hottest spot towards the ambient.

As previously is difficult to see the effect of the heat capacity since the peak of the sinusoidal are almost aligned.

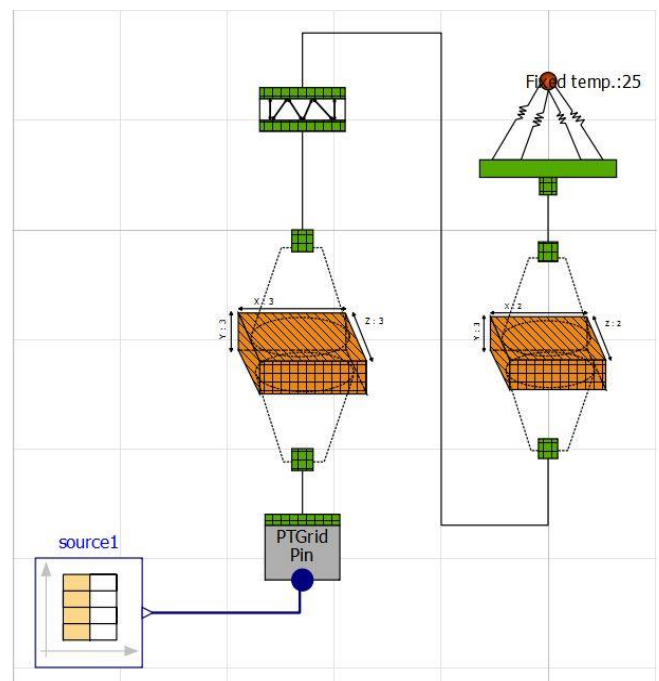
41: Test 3x3x3 (with offset) results



Test 3x3x3 and 2x3x3

In this test is been added and addition base block with a different partition from the first one allowing to use the acausal interface between them.

As before the settings for this test was too difficult to collect a single result highlighting all the behavior in one graph, so as before the heat capacity effect will be imperceptible and also the conductance of the acausal block are been set with low value so the temperature of the facing element will be almost the same.



42: Test 3x3x3 and 2x3x2 Modelica set-up

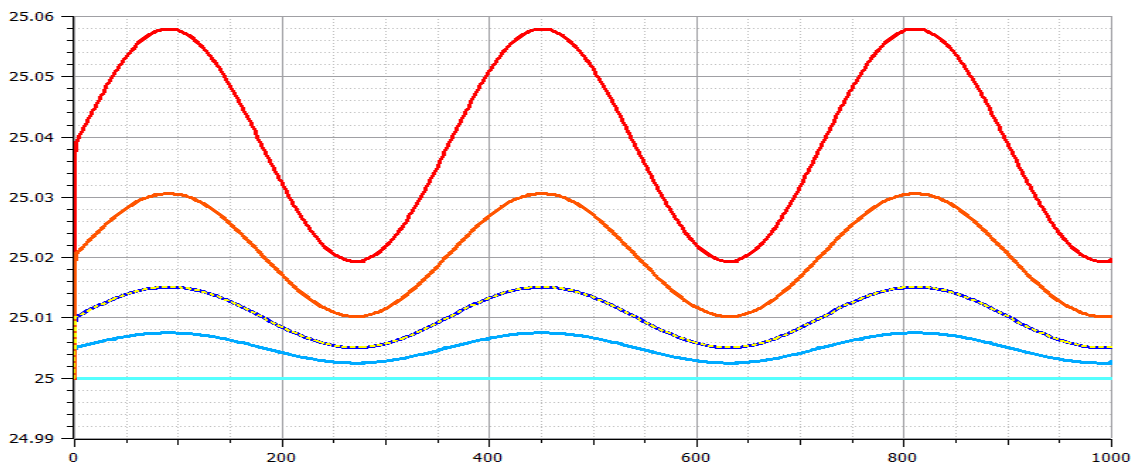
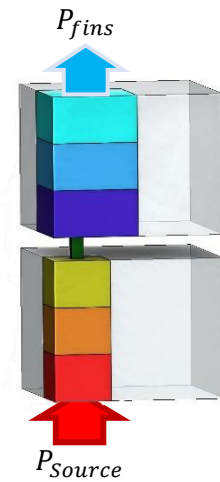
Result

In the following graph are represented the temperature of the column of element above the heating source. The colors settings are explained by the figure 43.

As before the distribution of the temperature is similar as expected since the test is almost the same as the second one but with additional element between the power source and the power sink.

As mentioned before the yellow line and the blue one overlap.

The focus of this last test was to verify the correct built of the acausal interface and we are satisfied by the result.



44: Test 3x3x3 and 2x3x2 (with offset) results

Conclusions

As said in the introduction the importance of large-scale model finds always more application in an enormous variety of fields from mechanics to thermal modeling of semi-conductor the applicability of such approach as the finite element methods in incredible.

It is also true that the software support for this kind of implementation is still at an embryonal stage and require lot of work to being functional and practice. With the results of this work we do a great step in that direction introducing an easier and lighter way to build and connect block using finite element partition, and also the possibility to link such block with a different type of interface like the power source map used for the development.

Interesting the ideas and hits born during this thesis like the necessity of implement a more accurate conductance computation for the single element.

The test result show a good behavior of the meta-program and we hope in a future and more concrete implementation of this program.

Bibliography

A. Leva, F. Terraneo, I. Giacomello, and W. Fornaciari “Event-Based Power/Performance-Aware Thermal Management for High-Density Microprocessors”, IEEE transactions on control systems technology, Milano, Italy, 2017.

D. Brooks and M. Martonosi, “Dynamic thermal management for highperformance microprocessors,” in Proc. 7th Int. Symp. High Perform. Comput. Archit., Nuevo Leone, Mexico, Jan. 2001.

J. Donald and M. Martonosi, “Techniques for multicore thermal management: Classification and new exploration,” in Proc. 33rd Int. Symp. Comput. Archit., Boston, MA, USA, 2006.

W. Hu et al., “An efficient task mapping algorithm with power-aware optimization for network on chip,” J. Syst. Archit., vol. 70, Oct. 2016.

W. L. Hung, G. M. Link, Y. Xie, N. Vijaykrishnan, and M. J. Irwin, “Interconnect and thermal-aware floorplanning for 3D microprocessors,” in Proc. 7th Int. Symp. Quality Electron. Design, San Jose, CA, USA, 2006.

S. Niknam, A. Asad, M. Fathy, and A.-M. Rahmani, “Energy efficient 3D hybrid processor-memory architecture for the dark silicon age,” in Proc. 10th Int. Symp. Reconfigurable Commun.-Centric Syst.-Chip, Tallinn, Estonia, Jun. 2015.

F. Tavakkoli, S. Ebrahimi, S. Wang, and K. Vafai, “Analysis of critical thermal issues in 3D integrated circuits,” Int. J. Heat Mass Transf., vol. 97, Jun. 2016

S. Murali et al., “Temperature control of high-performance multi-core platforms using convex optimization,” in Proc. IEEE Design, Autom. Test Eur. (DATE), Munich, Germany, Mar. 2008.