Fast Field Solvers for Thermal and Electrostatic Analysis

V. Székely and M. Rencz

Technical University of Budapest, Department of Electron Devices

1521 Budapest, Hungary

Abstract

Two different field solver tools have been developed in order to facilitate fast thermal and electro-static simulation of microsystem elements. The µS-THERMANAL program is capable for the fast steady-state and dynamic simulation of suspended multilayered microsystem structures. The 2D-SUNRED program is the first version of a general field solver based on an original method, the successive node reduction. SUNRED offers a very fast and accurate substitute of FEM programs for the solution of the Poisson equation. Steady-state and dynamic simulation examples demonstrate the usability of the novel tool.

1. Introduction

Thermal and electro-static effects play fundamental role in the operation of many microsystem elements, as e.g. infrared sensors, thermal rms meters, capacitive displacement sensors, actuators based on the electrostatic force, etc. All these problems require fast and reliable field solver programs capable to solve the following equation:

$$p(x,y,z) + c\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\lambda\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(\lambda\frac{\partial T}{\partial z}\right) \qquad (1)$$

where λ is the heat conductivity, c is the heat capacitance per unit volume, $T(x,y,z)$ is the thermal field and $p(x,y,z)$ is the incoming heat flux density. FEM methods can always be used for this purpose, but the drawbacks are well known: first of all the huge amount of time required to define the problem.

Two entirely different programs have been developed at TU Budapest for the solution of Eq. (1) aimed at microsystem applications [1]. These two programs are presented in this paper.

2. The µS-THERMANAL program

The µS-THERMANAL program has been developed for the thermal simulation of suspended microsystem elements as cantilevers, membranes, bridges, together with the investigation of the conventional IC chips [2].

The ancestor of this program, the THERMANAL program was an early realization of the well known algorithm of Kokkas [3], with the extension for unlimited number of layers [4]. In the model of Kokkas the structure consists of equally shaped rectangular layers stacked on an ideal heat sink. The dissipating elements are on the surface of the uppermost layer only, and the heat is removed via the bottom surface, the sidewalls are adiabatic. Heat transfer is assumed only by conduction. This model is a good one for conventional ICs, this explains that several realizations have been reported even recently.

In the algorithm of [3] the solution of the differential equation is constructed in the form of two-dimensional Fourier-cosine series. In our program this series is calculated by Fast Fourier Transformation (FFT) method – resulting in a quick solution both in steady-state and in the frequency-domain.

The Fourier algorithm relatively easily can be extended to calculate the temperature distribution on multi-layer structured membranes as well. In this case the boundary conditions are different from the usual IC structure: the top and the bottom surfaces of a membrane are adiabatic, while the sidewalls are isothermal, considering the bulk silicon nearly ideal heat sink.
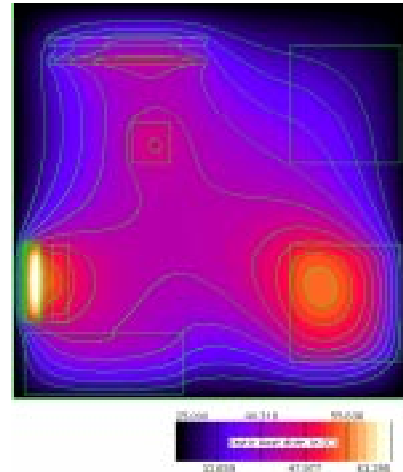
Fig.1. Steady-state temperature distribution on a membrane surface. The rectangular shapes are the dissipating elements.

These boundary conditions are fulfilled by using Fourier-*sine* expansion instead of Fourier-cosine [2]. Simulation results of a membrane structure are presented in Fig.1.

Rectangular bridge and cantilever structures require the consideration of other, special boundary conditions. The Fourier method can be matched to these conditions as well, see [2].

We have extended this model by the consideration of free convection cooling, taking into account the heat transfer by convection on the top and bottom surface of the membrane. Moreover we modified the original algorithm also towards considering dissipators inside the layered structure. The detailed description of these algorithms can be found in [2].

From heat transfer point of view a special class of microsystem elements are the parts suspended on thin and

narrow strips. H-shaped membranes and cantilevers with a narrow neck belong to this class. In these structures the leads act as heat sinking elements. In the μS-THERMANAL program we consider them as pseudo-dissipating elements with negative dissipation. The algorithm for the exact calculation of these dissipation values is given in [5].

3. SUNRED: a 2D thermal and electrostatic simulator

During our work in the thermal simulation of 3D SOI structures we encountered the problem, that the accuracy of the applied FEM simulation tool had to be limited down to an unacceptable value if we wanted to have the simulation results in a reasonable time. To overcome this problem we have developed a new dedicated field solver program SUNRED, which works currently in two dimensions, but it is expandable to 3D. The applied method is a finite difference method. The algorithm is the *SUccessive Node REDuction*, leading to the acronym SUNRED.

This tool is designed especially for the fast calculation of the thermal behavior of arbitrary shape integrated microstructures. A special requirement was to calculate on a grid that is fine enough to obtain not only the temperature distribution but the accurate streamlines of the heat-flow as well.

An interesting feature of this simulator is that characteristic methods of three distinct disciplines are combined in it. These fields are
- the electromagnetic field theory,
- the linear network theory and
- image processing.

Combination of these methods resulted in a very useful tool.

3.1. The model

The current 2D version of the program treats the linear heat conduction problems in two dimensions. Anisotropy can be taken into account. The equation being solved is

$$p(x,y) + c\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\lambda\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) \qquad (2)$$

and in the steady-state case

$$p(x,y) = \frac{\partial}{\partial x}\left(\lambda\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda\frac{\partial T}{\partial y}\right) \qquad (3)$$

This is the 2D form of the well-known Poisson equation, the mathematical description of many physical phenomenon. Electrical current-stream fields and electrostatic potential fields can be described by the same equation, which means that the program is capable to solve such problems as well. Thus, investigation of capacitive sensor and actuator structures can be also performed with the help of this program.

The investigated area is a rectangle. A dense equidistant grid is spawned to this area defining a cell matrix. The suggested grid size is either 128×128 or 256×256 (for a PC, higher resolution can be used on workstations). A material type is assigned to each cell. This assignment is performed by constructing an image – in the sense of the digital image handling methods. Each pixel of this digital image corresponds to a grid-cell whereas the material type constituting the cell is coded by the color of the pixel. Thus, in order to enter a problem two files have to prepared:

- the "problem-image" which can be in any usual image format (the suggested format is the BMP),

- the "material-table" assigning different material parameters to each color.

This method of problem definition provides a very easy and fast input of complex geometrical arrangements (using any general picture editing tools). Almost arbitrarily shaped structures can be investigated, limitation is coming only from the finite resolution of the digital image. Real images, e.g. a microscopic image of some IC structure may also be used as geometry input.

On the edges of the investigated rectangular area either forced temperature or zero heat-flow can be prescribed – individually, for any gridpoints of the boundary. Excitations can be defined in the interior of the investigated area as well, forcing a given temperature or a given heat-flux to any cell. Obviously a new "color" should be introduced for each excitation value in the problem image.

The solution of Eq.(2) is accomplished using the method of *finite differences*, and applying a network model for the thermal field. The cells of the field are described by an electrical model. The cells are squares (or rectangles), with a node in their center (Fig.2a.). Heat flux can be forced into them – this corresponds to the current flowing in this node. Forced temperature means the forced value of the cell node.
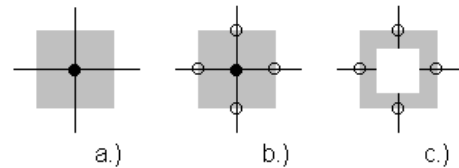
Fig.2. Cell, center node and terminal nodes

The boundary between different materials is lying always on the cell edges. In other words: each cell is "filled" by a single material. Each cell has four terminals in the direction of its four neighbors (Fig.2b.). On the terminals each cell can be described by a 4×4 matrix. This way the center node is hidden, but knowing the terminal temperatures the temperature of the center node can be back-calculated. Fig.2c. presents that the cell shows four terminals to the outside and the inner node is hidden.

The steady-state model of the cell is shown in Fig.3. It contains four thermal conductances. The value of these conductances depend on the thermal conductivity of the material filling the cell and on the geometry. This basic cell can be described by an admittance matrix of 4×4 size.
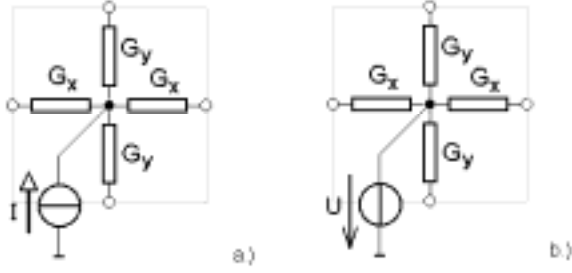
Fig.3. Steady-state circuit models of a single cell.
a.) Current excitation, b.) forced voltage

3.2. The solution algorithm

The solution of the problem is done by the electrical solution of the whole model network. This raises serious problems because of the size of this network. Using a grid of 128×128 lines the model network consists of 32768 nodes. For a 256×256 grid arrangement this number is 131072. Although the corresponding circuit matrix is extremely sparse the solution of such a big network is a hard problem.

In order to avoid the troublesome "when to finish the iteration" problems we have not considered iterative solutions − only direct methods have been investigated. Similarly to the idea described in [6], a successive procedure has been developed for the network reduction. The essential features of this algorithm are briefly presented in this paragraph.

Four basic cells can be assembled to form a block or macrocell as shown in Fig.4a. In other words: a *1st order cell* has been built from four *zero-order cells*. The four interior connecting terminals of the cells can be eliminated; they will not appear in the outside-directed description.

a.)   b.)
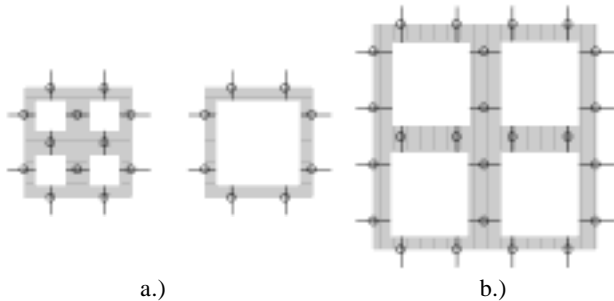
Fig 4. Network reduction. a.) Four basic cells will constitute a 1st level cell, b.) Building a 2nd level cell

Using four 1st level cells we can assemble a 2nd level cell as shown in Fig.4b. The inner terminals can be eliminated again.

Continuing this successive construction of higher and higher level cells we obtain finally the matrix of a single cell − the terminals of which are lying on the four edges of the investigated rectangular field. Matching with the boundary conditions means the solution of this matrix for the U or I constraints, given individually for the terminals lying on the boundaries of the investigated field. The voltages of all the inside nodes can then be calculated by a successive back-substitution.

Let us present the procedure in terms of the data flow and arithmetic operations used. The cells are described by their admittance matrices **Y** relating to the boundary nodes and by their inhomogeneous vectors **J** representing the excitations on all the inside nodes but reduced to the boundary nodes. For the 0th level cells shown in Fig.3. **Y** and **J** can be generated by using elementary calculations. The connection of two cells as shown in Fig.5 is equivalent to the addition of their **Y** matrices and **J** vectors where the areas of the connected nodes overlap as shown in Fig.5. as well.
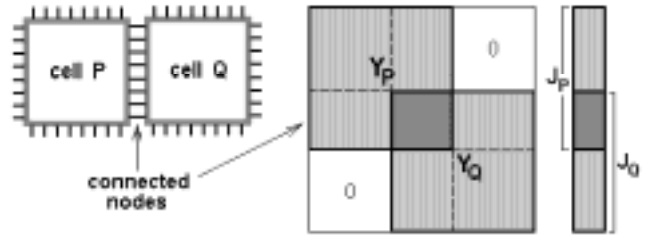
Fig.5. Connection of two cells and the resulting Y matrix and J vector

The next step is to eliminate the inside (≡connected) nodes. For sake of better understanding **Y** and **J** are visualized in Fig.6. in a rearranged node order. The first N nodes are the boundary nodes (that should be kept), M are the inside nodes that being eliminated. Partitions of **Y** are denoted by **YA**, **YB**, **X** and **X^t** as shown in Fig.6.
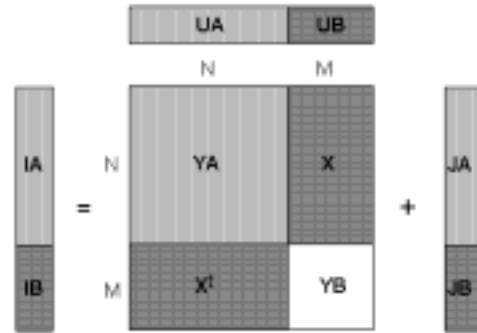
Fig.6. Partitions of the admittance matrix

The nodal voltages and nodal currents are represented by the vectors **U** and **I**, respectively. The **J**, **U** and **I** vectors are partitioned similarly to **Y**. The linear matrix equation for the two connected cells is

$$\mathbf{IA} = \mathbf{YA} \cdot \mathbf{UA} + \mathbf{X} \cdot \mathbf{UB} + \mathbf{JA} \qquad (4)$$

$$\mathbf{IB} = \mathbf{X^t} \cdot \mathbf{UA} + \mathbf{YB} \cdot \mathbf{UB} + \mathbf{JB} \qquad (5)$$

Elementary rearrangements of these equations result in the formula for the reduced admittance matrix

$$\mathbf{Y_{RED}} = \mathbf{YA} - \mathbf{X} \cdot \mathbf{ZB} \cdot \mathbf{X^t} \qquad (6)$$

where $\mathbf{ZB} = \mathbf{YB^{-1}}$. The new inhomogeneous part is

$$\mathbf{J_{RED}} = \mathbf{JA} - \mathbf{X} \cdot \mathbf{ZB} \cdot \mathbf{JB} \qquad (7)$$

During the back-substitution step $[\mathbf{UA}] \to [\mathbf{UA,UB}]$

$$\mathbf{UB} = - \mathbf{ZB}{\cdot}\mathbf{X^t} \cdot \mathbf{UA} - \mathbf{ZB} \cdot \mathbf{JB} \quad . \tag{8}$$

It is worthy for note that the same matrix is appearing in (7) and (8) because

$$\mathbf{ZB}{\cdot}\mathbf{X^t} = (\mathbf{X}{\cdot}\mathbf{ZB})^\mathbf{t} \quad . \tag{9}$$

All the $\mathbf{Y}$ matrices are symmetrical. This permits a saving of about 50% both in storage and in arithmetic operations.

The description of all cells by their $\mathbf{Y}$ matrices represents a huge amount of data. As the processing is essentially serial, it is advantageous to store these data streams in files. Thus the organization of the program is mainly *pipelined*: the program segments read one or more streams from files and writes the results into further files. This way a quite large number of nodes can be handled on computers having only limited amount of memory.

The short description of the main program segments demonstrates clearly the pipelined process:

1. Network reduction. The segment reads the queue of the $\mathbf{Y}$ matrices of nth level cells, reduces them in fours by using three times Eq. (6) and writes the resulting, $n+1$th level $\mathbf{Y}$ matrices into a new file. The $\mathbf{ZB}$ and $\mathbf{X}{\cdot}\mathbf{ZB}$ matrices are calculated and stored into a further file as well. The number of runs of this segment is $\log_2(K)$, where K is the number of the pixels in one edge of the problem-image.

2. Forward substitution. This segment reads the file of the $\mathbf{X}{\cdot}\mathbf{ZB}$ matrices, reads the queue of the $\mathbf{J}$ inhomogeneous vectors of nth level cells, reduces them in fours by using Eq. (7) and writes the resulted, $n+1$th level $\mathbf{J}$ vectors into a new file. The number of required runs is $\log_2(K)$ again.

3. Solution. This segment uses the uppermost level $\mathbf{Y}$ matrix and $\mathbf{J}$ vector and solves the corresponding system of linear equations, taking into account the actual boundary parameters.

4. Backward substitution. This segment calculates the voltages on the internal nodes in a hierarchical top-down order, by using Eq. (8) and two files: the queues of the $\mathbf{J}$ vectors and the $\mathbf{X}{\cdot}\mathbf{ZB}$ matrices.

The advantage of this ordering of the calculus lies in the fact that the first and most time consuming step has to be repeated only in the case when the investigated structure has been changed. When the excitations are only changed, steps 2, 3 and 4 have to be repeated. In the case when only the boundary conditions are modified repeating of the steps 3 and 4 is sufficient.

The hierarchical network reduction requires $\log_2(128)=7$ successive steps for the $128{\times}128$ grid, 8 steps for the grid-size of 256 and so on. The detailed analysis of the t total computing time gives

$$t = 63.5 \, P^{3/2} \, (t_{*+}) \tag{10}$$

where P is the node number for the whole model network and (t_{*+}) is the time of one multiplication and one addition. This time should be compared to the Ordo(P^3) time requirement of a "brutal force" Gauss elimination.

For a 32768 node problem the solution time is only 6 minutes on a 586 PC and 50 sec on a SUN20 workstation.

The program provides both the steady-state analysis and the transient (time-domain) simulation. In the latter case the reverse-Euler integration scheme is used. This method reduces the problem of the transient analysis to a d.c. solution for each time-step.

A useful feature of the program is that 3D structures having cylindrical symmetry can be calculated with the same 2D algorithm. The basic idea of the corresponding algorithm is a transformation of the heat-conduction equation. Both the λ heat conductivity and the c heat capacitance have to be substituted by

$$\lambda\text{'} = \lambda \, x \, , \quad c\text{'} = c \, x \tag{11}$$

in Eq.(2) where x is the radial coordinate. This transforms Eq.(2) into the heat-conduction equation written in cylindrical coordinates. This means that by applying (11) during the generation of the model network, the results will be valid automatically for the cylindrical structure.

Finally a comparison has to be given with the ancestor of this algorithm described in [6]. Our realization provides the advantages that (i) there is no need to handle the node-incidence matrix of the network, only the admittance matrix is used, (ii) current sources buried within the interior of the cells can also be handled, (iii) the transient simulation can also be realized by using the described calculation scheme.

3.3. Presentation of the results

The results of the simulation are treated as images again. The temperature (or potential) fields which are essentially 2D scalar functions can be considered as black-and-white images. The brightness of the image points is proportional to the temperature (or potential) of each point. The program provides the results in form of digital images, in the standard BMP image format. Such a potential image is shown in Fig.7.

Although this image gives a good qualitative view of the potential field, the potential values can not be read from this picture. A very basic procedure of the image processing can help to overcome this problem. This procedure is the intensity transformation: an arbitrary $b_{tr}=f(b)$ function can be used to map the original b pixel intensities into the b_{tr} brightness values. A set of appropriately chosen functions offer a rich variety of presentations for the same temperature (potential) field, and provide the good quantitative evaluation at the same time. Fig.8. is an example for this: it is the same image as in Fig.7. but after a suitable intensity transformation.

The temperature field or potential image is generally not enough to visualize thermal, electrostatic or streaming fields. Tracing of the heat-stream lines (or electrical field lines) is an often encountered requirement. Especially in the case of temperature fields the streamlines provide an easy way to "discover" how and where the heat flux is streaming.

To obtain streamline pictures further image processing steps are required. The simulator provides the
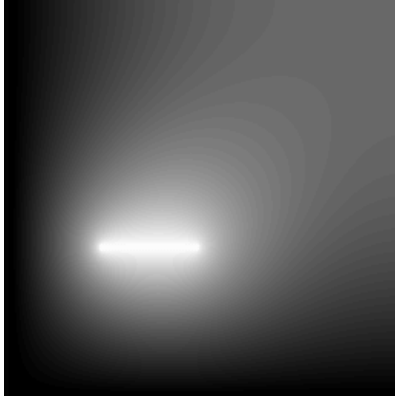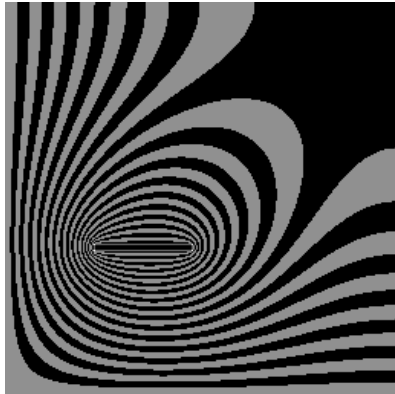
Fig.7. Grayscale image of a potential field)

Fig.8. Potential or temperature field using a telegraph signal like intensity mapping.

J_x, J_y components of the J current density vector in form of two intensity images. These vectors have to be turned by 90°. This can be made simply by interchanging the two images and negating one of them:

$$G_x = J_y$$
$$G_y = -J_x$$
(12)

It can be easily proven that the $P(x,y)$ potential function of this G field is suitable to trace the streamlines. The equipotential lines of this P *pseudopotential* are the streamlines of the original field. Visualization of these lines can be proceeded in the same way as in case of the real potential field.

This pseudopotential can be constructed if and only if there is no divergence in the $J(x,y)$ vector-field. This means that the procedure can be applied only on the divergence-free regions of the field. Pictures generated by using this pseudopotential are shown e.g. in Figs. 9 and 10.

3.4. Examples

In the first example a 3D stacked submicron CMOS SOI structure [7] is investigated. The heat is generated in the upper transistor. The results of the steady-state simulation are shown in Fig.9. The highly different thermal conductivity of the silicon and the SiO_2 leads to the surprising fact that the heat stream makes a detour (see the white arrows) around the SiO_2 region.
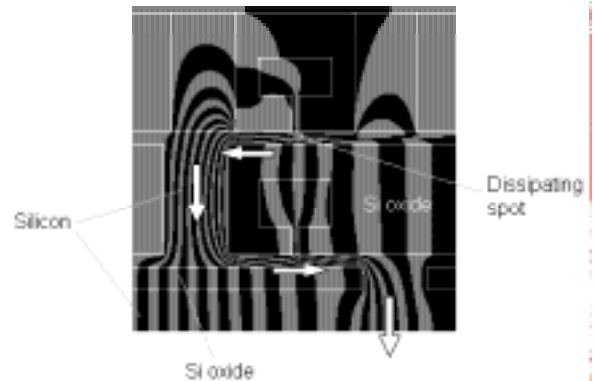

Fig.9. Heat-flow in a stacked SOI structure

The second example demonstrates how a microphotograph can serve as the input for problem statement. The electron-microscopic image of the facet of a special laser diode design [8] is shown in Fig.10a[1]. Image processing methods were used to extract the contours of the areas representing the different materials and to color them appropriately. The resulting image was used as the input of the simulation, the result of which is shown in Fig.10b.
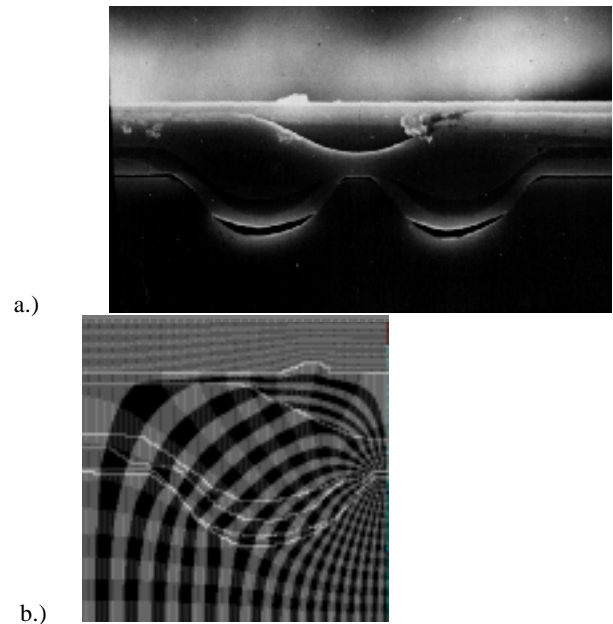

a.)

b.)

Fig.10. Thermal simulation of a laser diode.
a.) microphotograph, b.) simulation results of the left hand side of the structure (the isotherms and the streamlines)
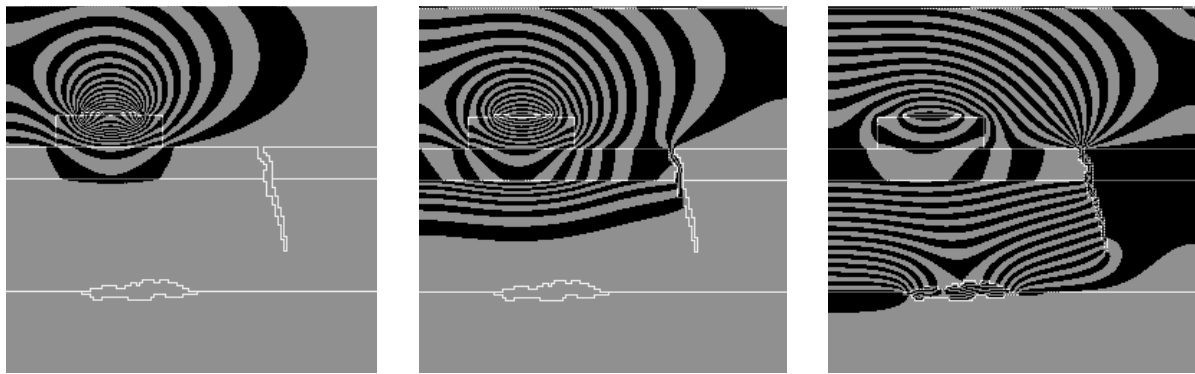
In the third example a semiconductor chip mounted on a multi-layered metal + ceramics substrate is investigated. The heat-flow is perturbed by an irregularly shaped cracking and by a cavity at the interface of the two bottom layers. The steady-state thermal field is demonstrated in Fig.11. by the heat-flow lines.

---

[1] Courtesy of I. Habermajer, TU Budapest.
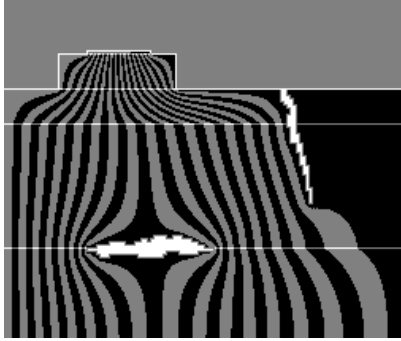
Fig.11. Heat streamlines in a chip mounting

The transient analysis generates a frame sequence that can be played as a movie. Three frames of such a sequence are shown in Fig. 12. Both time and spatial

functions can be extracted from the frame sequence. In Fig.13. the time functions of the temperature are visualized referring to 4 prescribed locations.

Fig.13. Time diagrams for the problem of Fig.11.

Fig.12. Three frames from a transient sequence (t= 0.4ms, 5 ms, 156 ms)
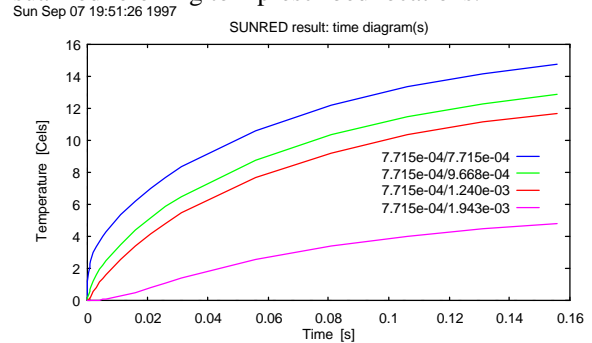
4. Conclusions

Two new, fast and easy-to-use thermal simulation tools have been developed for the accurate thermal simulation of dedicated microsystem elements and special 3D structures. With the help of the novel 2D-SUNRED program arbitrarily shaped structures can be thermally analyzed in minutes, currently in two dimensions. The development of the 3D version of the program is in progress. Using the μS-THERMANAL simulator, both the steady-state and the frequency-domain behavior and even the accurate dynamic thermal model of microsystem elements can be obtained.

References

[1] **V. Székely, M. Rencz:** Fast field solver programs for thermal and electrostatic analysis of microsystem elements, ICCAD'97 Int. Conf. on Computer-Aided Design, Nov. 9-13 1997, San Jose, Ca, USA, Proc. pp.684-689

[2] **V. Székely, A. Csendes, M. Rencz:** μS-THERMANAL: An efficient thermal simulation tool for microsystem elements and MCM's. SPIE'96 Symposium on Micro-machining and Microfabrication, 14-15 October 1996, Austin, Texas, USA, SPIE Proc. Vol. 2880, pp. 64-75.

[3] **A.G. Kokkas:** Thermal analysis of multiple-layer structures, IEEE Trans. on El. Dev., V. ED-21, No. 11, pp. 674-681 (1974)

[4] **V. Székely, P. Baji, M. Rencz:** Graphical computer methods in the design of integrated circuits, Periodica Polytechnica, V. 23, No. 3-4, pp. 331-338 (1979)

[5] **V. Székely, A. Poppe:** Novel tools for thermal and electrical analysis of circuits, Electrosoft, V. 1. No. 4. pp. 234-252 (1990)

[6] **T. A. Johnson, R. W. Knepper, V. Marcello and W. Wang:** Chip substrate resistance modeling technique for integrated circuit design, IEEE Trans. on Computer-Aided Design, Vol.CAD-3, No.2, pp. 126-134 (1984)

[7] **V. Dudek, V.O. Keck, G. Mayer, B. Höfflinger:** Design considerations for high performance low-power Silicon-on-insulator gate arrays, Custom Integrated Circuits Conference 1995, Santa Clara, CA, May 1995

[8] **I. Habermajer, Ö.Lendvay, Z.Lábadi and V.Rakovics:** Double hetero structure InP/GaInAsP laser diode with double channel substrate, built-in reverse-biased pn junction and buried active layer, Hungarian Patent No. 206 565., 1988