The Snuffle Problem

Gerstenmaier

for the Heat Conduction in a Power Module with 6 Power Chips

February 2007

Definition of the Snuffle Problem Gerstenmaier

York Christian Gerstenmaier

Siemens AG, Corporate Technology, Munich, Germany
c/o Institute for Physics of Electrotechnology, Munich University of Technology,
Arcisstr. 21, 80290 Munich, Germany, yge@tep.ei.tum.de, phone: +49 89 28923127

1 Introduction

In the following a thermal problem will be presented which is encountered in the thermal predictive simulation of power semiconductor modules (e.g. dc/ac-converters). Heat sources are MOSFET-devices (or other semiconductor devices) on the top side of the module (figure 1 presents an example). The cooling is applied at the bottom side of the module, either by a convective liquid or gas (air) stream. The temperature evolution $T(x,t)$ in the module and in the power dissipating devices is described by the heat conduction equation:

$$\left(\rho(x)c(x)\frac{\partial}{\partial t} - \nabla_x \cdot (\lambda(x)\nabla_x)\right) T(x,t) = H(x,t,T(x,t)) \;. \tag{1}$$

x denotes the 3D position (x, y, z). $H(x,t,T(x,t))$ is the heat generation density [W/cm^3] in the system, which will generally depend on the temperature $T(x,t)$ of the heat source at position and time (x,t), because e.g. on-state and switching losses of the MOSFETs are temperature dependent. ρ, c, λ denotes the local mass density, the specific heat and the thermal conductivity, respectively. Strictly speaking ρ, c, λ also depend on local temperature. However, it turns out that in power-semiconductor-modules material parameters for constant mean temperatures can be used in good approximation.

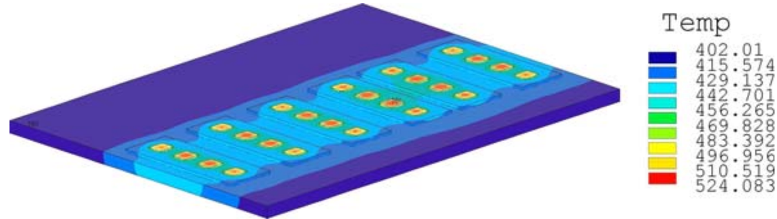

Fig. 1. Steady state, linear FEM-simulation (44104 grid points) of dc/ac-converter module with 24 power-MOSFETs heated with uniform power. Ambient temperature of 400 K applied at the bottom side.

An essential nonlinearity arises due to the convective cooling at the bottom side, which is included by a corresponding boundary condition:

$$\lambda(x)\frac{\partial T(x,t)}{\partial n} = \alpha(x,T,T_a)(T(x,t) - T_a(x,t)) + J(x,t) \;. \tag{2}$$

Here $\alpha(x, T, T_a)$ denotes a local heat-transfer coefficient at an edge point x of the structure with local temperature $T(x, t)$ and local ambient (fluid-)temperature $T_a(x, t)$. $\partial/\partial n$ denotes the normal derivative with outside direction. $J(x, t)$ is an optional surface heat source.

2 Model and Parameters

For a simplified model only cooling at the bottom side is assumed, while the other sides are thermally adiabatic. When the module is mounted on an air cooled radiator, the effective cooling area of the module is increased considerably, e.g. by a factor of 100. This can be described by an effective $\alpha(T)$ in a snuffle problem calculation with the following typical parameters:

Heat flux by black body radiation:

$$\sigma(T^4 - T_a^4) = \alpha_r(T, T_a)(T - T_a) ,$$

such that

$$\alpha(T, T_a) = \sigma(T^2 + T_a^2)(T + T_a)$$

with

$$\sigma = 567 \cdot 10^{-12} [\mathrm{W}/(\mathrm{cm}^2\ \mathrm{K}^4)] \quad \text{(Stefan-Boltzmann constant * 100)} .$$

For natural convection:

$$\alpha_c(T, Ta) = a(T - T_a)^{1/4}$$

with

$$a = 406 \cdot 10^{-4} [\mathrm{W}/(\mathrm{cm}^2\ \mathrm{K}^{5/4})] .$$

The complete heat transfer coefficient is:

$$\alpha(T) = \alpha_r + \alpha_c .$$

The ambient temperature is constant:

$$T_a = 297\,\mathrm{K} .$$

The sidewall and topside boundary conditions of the module are adiabatic ($\partial T/\partial n = 0$).

In order to have simple geometry and boundary conditions the whole module is assumed to be of rectangular structure with uniform material. The dimensions (side lengths) of the module are: 11.8 cm length; 5.8 cm width; 0.5 cm height.

The effective material parameters for module and silicon chips (MOSFETs) are uniform:

Density [g/cm^3] : $\rho = 2.32$;
specific heat [Ws/(g K)] : $c = 0.851$;
thermal conductivity [W/(K cm)] : $\lambda = 1.51$.

There are 6 chip heat sources in the module which do not disturb the geometry of the rectangular parallelepiped and are assumed as embedded regions with heat generation density $H(x,t)$ different from zero. Those 6 "chips" are quadratic and of equal dimension ($0.9 \times 0.9\,\mathrm{cm}^2$) oriented in one row at the module topside. The chip thickness is 0.02 cm (200 μm). The distances of the chips left edge from the left side of the module are in [cm]: x1 = 1.35, x2 = 2.95, x3 = 4.55, x4 = 6.15, x5 = 7.75, x6 = 9.35 (in the distance of 1.6 cm). The chip distance in y-direction from the module side wall is 1.9 cm. The chips topside is on the same plane as the rectangular module top-side.

The following calculations could be performed:

1. At start time $t = 0$ the module has homogeneous temperature $T_a = 297$ K. The chips are turned on with power dissipation of 250 W/Chip. This means a heat generation density H of 250 W/chipvolume = 15432.1 W/cm^3 for all 6 chips.
   What are the temperature contours on the module topside after 50 seconds? What is the heating curve of the center of the 3rd chip from the left (hottest chip) in the time domain of 0 to 50 sec?

2. Additionally to problem 1) now there is in the 6th chip from the left a degraded array of MOSFET cells with lateral dimensions of $0.1 \times 0.1\,\mathrm{cm}^2$, with considerably increased power dissipation. The chips 1–5 are heated again with constant 250 W (starting at $t = 0$), as is chip 6. However, now in chip 6 there is an additional 50 W of dissipated power in the degraded cell array, so that the total power of chip 6 is 300 W. The location of the quadratic degraded cell array is in the distance of 0.65 cm in x- and y-direction from the left lower corner of chip 6. The thickness of the extra heating region is as usual 0.02 cm.
   Temperature contours and heating curves as for problem 1) would be of interest and also the heating curve for the center of the degraded cell array.

Problem 2) poses a special problem in so far as the small degraded heat source will probably necessitate a rather fine grid discretization locally in order to calculate the local temperature correctly. This might cause problems with the grid generation.

Results of the Snuffle Problem Gerstenmaier by the FDEM Program

Torsten Adolph and Willi Schönauer

Forschungszentrum Karlsruhe, Institute for Scientific Computing,
Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
{torsten.adolph,willi.schoenauer}@iwr.fzk.de
http://www.fzk.de/iwr

1 The Snuffle Problem

The snuffle problem from York C. Gerstenmaier from the Institute for Physics of Electrotechnology of the Technische Universität München deals with the simulation of the temperature in a power semiconductor module. On the top side of the module we have 6 MOSFET-devices that are the heat sources, whereas on the bottom side the module is cooled by a convective liquid or gas stream.

This is a 3-D problem, and as we want to compute the temperature distribution on the surface of the module at a given time, it is also time dependent. Furthermore, as the MOSFET-devices are very thin in comparison with the remainder of the module, we separate the module into two subdomains: the upper subdomain is as thick as the MOSFET-devices, the lower subdomain contains the remainder of the module. As we expect greater temperature gradients in the upper subdomain, and as it would be too costly to have the same fine grid in the lower subdomain, we choose different mesh sizes in x- and y-direction in the two subdomains. Thus, we introduce a sliding dividing line (SDL) between the two subdomains; they are coupled by coupling conditions, see [1, Section 2.6].

For this snuffle problem we want to compute the temperature T for the heat conduction in the power module which can be described by the PDE

$$\left(\rho(x)c(x)\frac{\partial}{\partial t} - \nabla_x \cdot (\lambda(x)\nabla_x)\right) T(x,t) = H(x,t,T(x,t)) \ . \tag{1}$$

This PDE holds for both subdomains.

In (1), $T = T(x,t)$ is the unknown temperature, x is the position (x,y,z) in 3-D, $H(x,t,T(x,t))$ is the heat generation density, ρ is the local mass density, c is the specific heat and λ the thermal conductivity.

At the bottom side of the module, we have convective cooling, which results in the boundary condition

$$\lambda(x)\frac{\partial T(x,t)}{\partial n} = \alpha(x,T,T_a)(T(x,t) - T_a(x,t)) + J(x,t) \ , \tag{2}$$

where $\alpha(x,T,T_a)$ denotes a local heat-transfer coefficient at an edge point x of the structure with local temperature $T(x,t)$ and local ambient (fluid-)temperature $T_a(x,t)$. $\partial/\partial n$ denotes

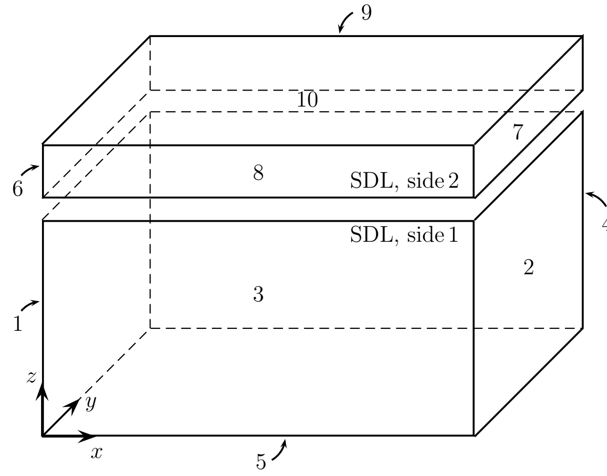

Fig. 1. Illustration of the subdomains with external boundaries and sliding dividing line (SDL).

the normal derivative with outside direction, $J(x,t)$ is an optional surface heat source that is put to zero in our case.

Fig. 1 shows the 10 external boundaries and the SDL of the domain. In Table 1 we present the boundary conditions for the external boundaries. In (2) the heat transfer coefficient α

Table 1. Boundary conditions for the external boundaries of Fig. 1.

Bd.	condition
1	$\frac{\partial T}{\partial x} = 0$
2	$\frac{\partial T}{\partial x} = 0$
3	$\frac{\partial T}{\partial y} = 0$
4	$\frac{\partial T}{\partial y} = 0$
5	Equ. (2)
6	$\frac{\partial T}{\partial x} = 0$
7	$\frac{\partial T}{\partial x} = 0$
8	$\frac{\partial T}{\partial y} = 0$
9	$\frac{\partial T}{\partial y} = 0$
10	$\frac{\partial T}{\partial z} = 0$

is composed from α_r and α_c that can be represented by

$$\alpha(x,T,T_a) = \sigma \cdot \left(T^2 + T_a^2\right)(T + T_a) + a \cdot (T - T_a)^{1/4} \tag{3}$$

so that we get

$$\begin{aligned}\alpha(x,T,T_a)(T-T_a) &= \sigma\cdot\left(T^2+T_a^2\right)\left(T^2-T_a^2\right)+a\cdot(T-T_a)^{5/4}\\ &= \sigma\cdot\left(T^4-T_a^4\right)+a\cdot(T-T_a)^{5/4}\ .\end{aligned} \tag{4}$$

As for the lower boundary 5

$$\frac{\partial T}{\partial n} = -\frac{\partial T}{\partial z} \tag{5}$$

holds, we get as boundary condition for this boundary:

$$-\lambda(x)\frac{\partial T}{\partial z}+\sigma\cdot\left(T^4-T_a^4\right)+a\cdot(T-T_a)^{5/4}=0\ . \tag{6}$$

Table 2. Coupling conditions for the SDL of Fig. 1.

DL	side	condition
1	1	$T_{upper} = T_{lower}$
1	2	$\frac{\partial T_{upper}}{\partial z} = \frac{\partial T_{lower}}{\partial z}$

T_{upper}: T in upper subdomain
T_{lower}: T in lower subdomain

In Table 2 we show the coupling conditions for the SDL. In Table 3 we give the material parameters for the solid materials that we use for the computation.

Table 3. Material parameters.

Param.	Unit	Value
λ	$W/(K\cdot cm)$	1.51
ρ	g/cm^3	2.32
c	$Ws/(g\cdot K)$	0.851
T_a	K	297
a	$W/(cm^2\cdot K^{5/4})$	0.406E-01
σ	$W/(cm^2\cdot K^4)$	0.567E-09
J	W/cm^2	0

As λ, ρ and c do not depend on x, (1) becomes

$$\rho c\,\frac{\partial T}{\partial t}-\lambda\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}\right)-H=0\ . \tag{7}$$

The 6 chips are assumed as embedded regions with heat generation density $H>0$. They are quadratic and of equal dimensions oriented in one row at the module surface, see Fig. 2.

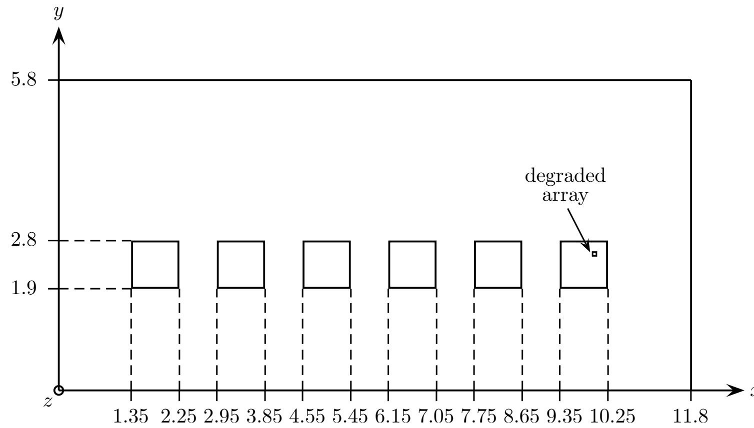

Fig. 2. Illustration of the position of the 6 chips and dimensions.

2 Results

We have to perform two calculations: first, all 6 chips have the same power dissipation of 250 W/Chip. We want to compute the temperature distribution on the surface after 50 sec. and the temperature curve of the centre of the 3rd chip from the left, which is expected to be the hottest chip. For the second calculation we suppose that the rightmost chip has a degraded array where we have an additional 50 W of dissipated power. Here we also want to compute the heating curve of the centre of the 6th chip. For both problems it holds $T(x, t = 0) = T_a$.

Table 4. Grid characteristics.

	length (x-direction)	width (y-direction)	height (z-direction)
upper (MOSFETs)			
dimension [cm]	11.8	5.8	0.02
no. of nodes	237	117	9
mesh size	0.05	0.05	0.0025
lower (remainder)			
dimension [cm]	11.8	5.8	0.48
no. of nodes	119	59	9
mesh size	0.10	0.10	0.06

We carried out the computations on the distributed memory supercomputer SGI Altix 4700 with Itanium2 processors, 1.6 GHz, and NUMAlink interconnect that has been installed at the Leibniz Computing Centre in Munich. We computed in parallel on 32 processors, and we use the consistency orders $q = 2$ and $q = 4$ to get aware of the influence of the order. In Table 4 we present the dimensions of the grid together with the number of nodes and the mesh size in the three space directions for the two subdomains.

Thus, the number of nodes in the upper subdomain is 249561, and the number of nodes in the lower subdomain is 63189. Therefore, the total number of grid points is 312750. For the mesh size h_x in x-direction and the mesh size h_y in y-direction, it holds $h_x = h_y$ in the lower and the upper subdomain, but we get for the ratio of the mesh sizes in the two subdomains $h_{x,u}/h_{x,l} = h_{y,u}/h_{y,l} = 1/2$. The ratio of the mesh sizes in the z-direction in the two subdomains is $h_{z,u}/h_{z,l} = 1/24$.

For the first problem, all 6 power chips have the same power dissipation of 250 W/Chip, so it holds for the heat generation density H

$$H = 250\,W/\text{chipvolume} = 250\,W/(0.9 \cdot 0.9 \cdot 0.02)cm^3 = 15432.1\,W/cm^3\ . \tag{8}$$

In Table 5 we present the results of the first problem where all 6 chips have the same power dissipation. For the consistency orders $q = 2$ and $q = 4$ you can see the maximum temperature T_{max} for each of the two subdomains and the errors of the solution. The maximum error is the maximum of the global relative estimated error, i.e. it is the maximum absolute error in the subdomain divided by the maximum of the temperature. The mean error is the arithmetic mean of all relative errors in the subdomain. The given CPU time is that of the master processor 1. For order $q = 2$, the maximum temperature, which is actually met in

Table 5. Results of the first calculation with $H = 250$ W/Chip.

Order	$q = 2$	$q = 4$
T_{max}, upper	526.4 K	526.2 K
T_{max}, lower	524.2 K	524.1 K
error upper, max	0.11E-01	0.78E-02
mean	0.81E-03	0.17E-03
error lower, max	0.69E-02	0.22E-02
mean	0.17E-03	0.35E-04
CPU time	8.3 h	28.2 h

the centre of the 3rd chip from the left, is 526.4 K after 50 sec. In the lower subdomain the maximum temperature is only slightly smaller. The maximum error in the upper domain is about 1%, in the lower subdomain it is only 0.7%, which means that we have a solution that is accurate to 6 K. The mean errors are much smaller which means that the maximum errors occur only in few nodes. Considering the mean error, the solution is accurate to 0.5 K.

We see that the maximum temperatures for order $q = 2$ and $q = 4$ differ only slightly, but the maximum errors in the subdomains are reduced to 2/3 in the upper and to 1/3 in the lower subdomain if we compute with order $q = 4$ instead of $q = 2$. The mean errors are reduced to about 1/4 in both subdomains. Therefore, for oder $q = 4$ the solution is accurate to 4 K, and if we look at the mean error, it is accurate to 0.1 K.

Figs. 3 and 4 show the temperature T on the surface of the module (a) and its error (b) as well as the heating curve of the centre of the 3rd chip from the left (c) for the computations with consistency order $q = 2$ and $q = 4$, respectively. From the error pictures of each figure we can also see that the maximum errors occur only in few nodes.

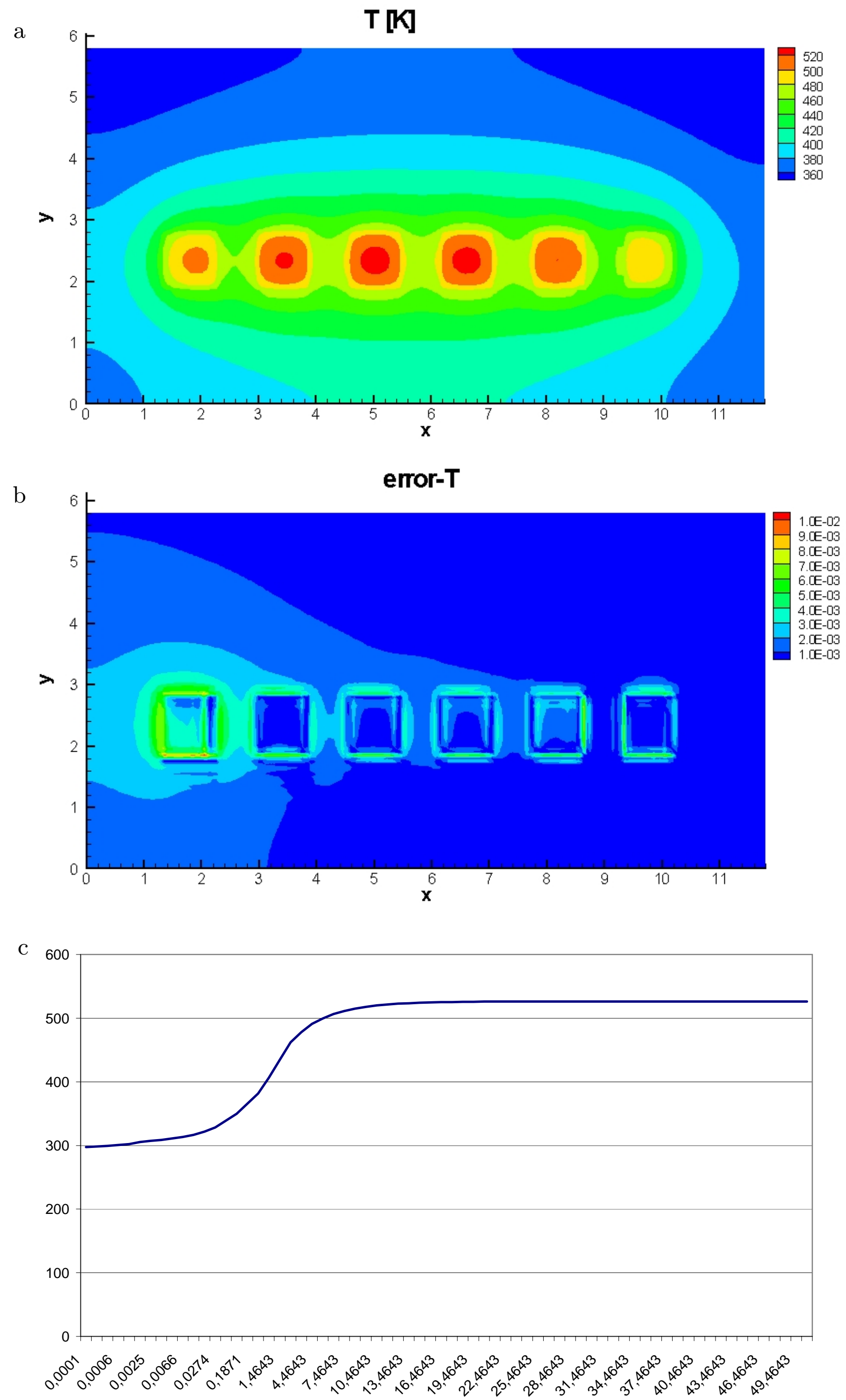

Fig. 3. Contour plot for the temperature T and its error on the top side of the module for $q = 2$ after 50 sec. ($H = 250\,\mathrm{W/Chip}$), heating curve for the centre of the 3rd chip from the left.

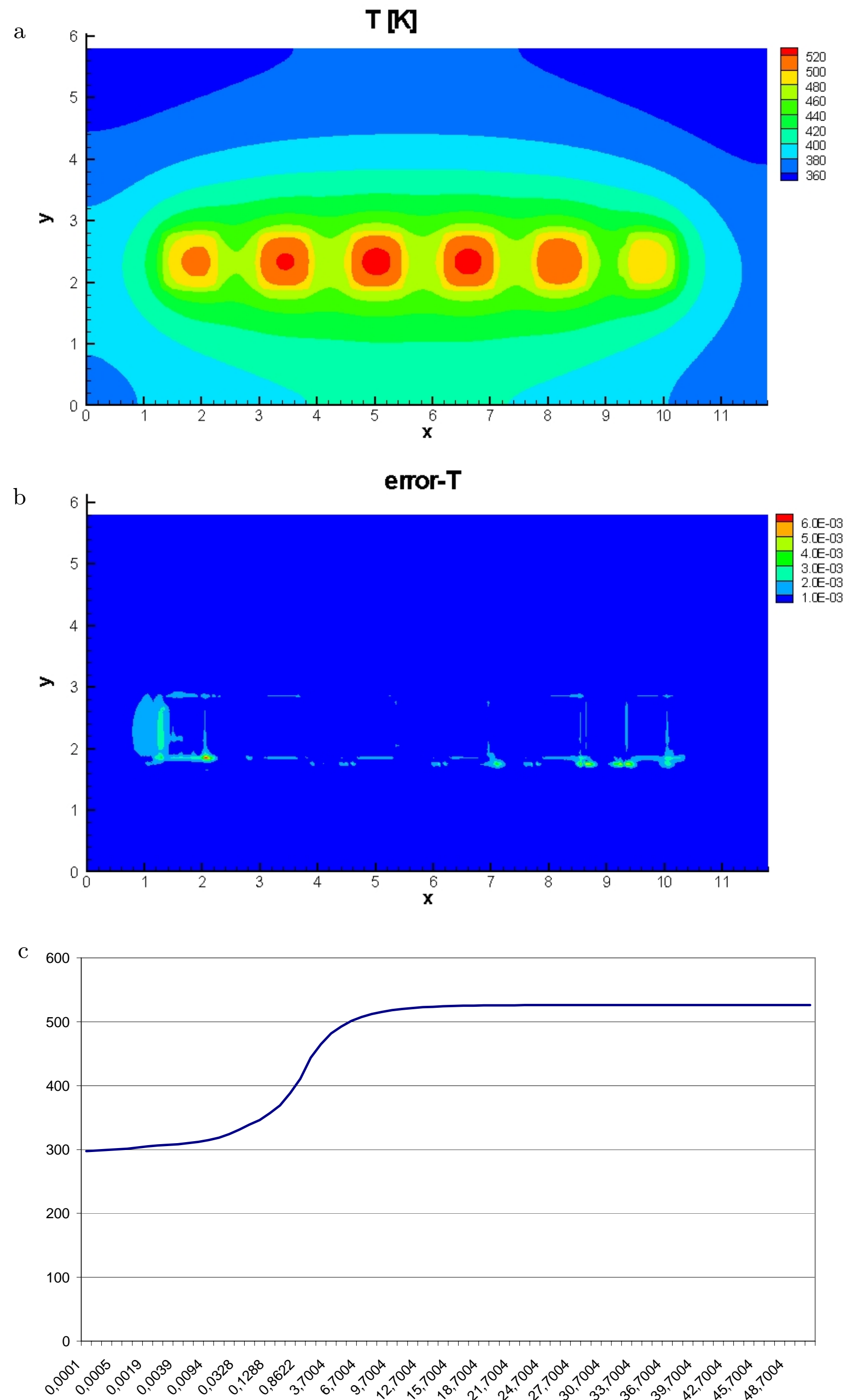

Fig. 4. Contour plot for the temperature T and its error on the top side of the module for $q = 4$ after 50 sec. ($H = 250$ W/Chip), heating curve for the centre of the 3rd chip from the left.

For the solution of the resulting linear system of equations we use the linear solver package LINSOL, and we use the bandwidth optimizer SSP. The CPU time for LINSOL is about 99.5% of the total CPU time, and the bandwidth optimization takes about 25% of the CPU time for LINSOL.

For the second problem, the leftmost 5 power chips have the same power dissipation of 250 W/Chip ($H = 15432.1$, see (8)), the 6th chip has a heat generation density H with

$$\begin{aligned} H &= 250\,W/\text{chipvolume} + 50\,W/\text{degraded array volume} \\ &= 15432.1\,W/cm^3 + 50\,W/(0.1 \cdot 0.1 \cdot 0.02)cm^3 \\ &= 15432.1\,W/cm^3 + 250000.0\,W/cm^3 \\ &= 265432.1\,W/cm^3\ . \end{aligned} \tag{9}$$

In Table 6 we present the results of the second problem where the chips 1–5 have the same power dissipation of $H = 250$ W, but the 6th chip has a degraded array with an additional power dissipation of 50 W. For the consistency orders $q = 2$ and $q = 4$ you can see the maximum temperature T_{max} for each of the two subdomains and the errors of the solution. The given CPU time is that of the master processor 1. For order $q = 2$, the maximum

Table 6. Results of the second calculation with $H = 250$ W/Chip, 6th chip with degraded array.

Order	$q = 2$	$q = 4$
T_{max}, upper	706.3 K	677.6 K
T_{max}, lower	632.7 K	614.3 K
error upper, max	0.56E-01	0.15E-01
mean	0.62E-03	0.13E-03
error lower, max	0.32E-01	0.73E-02
mean	0.14E-03	0.26E-04
CPU time	7.6 h	15.5 h

temperature, which is met in the centre of the degraded array of the 6th chip, is 706.3 K after 50 sec. In the lower subdomain the maximum temperature is only 632.7 K. The maximum error in the upper domain is about 5.6%, in the lower subdomain it is only 3.2%, which means that the solution is accurate to 40 K. The mean errors are much smaller which means that the maximum errors occur only in few nodes. Considering the mean error, the solution is accurate to 0.5 K. The mean errors are much smaller again which means that the maximum errors occur only in few nodes.

As the relative error in the upper subdomain is about 5.6%, i.e. 39.6 K, the actual maximum temperature must be between 666.7 K and 745.9 K. For order $q = 4$ we get a maximum error of 1.5%, so the maximum temperature is in the expected area. We see that the maximum temperature for order $q = 2$ is about 4% higher than that of order $q = 4$. The maximum errors in the subdomains are reduced to 1/4 in both subdomains if we compute with order $q = 4$ instead of $q = 2$. The mean errors are reduced to about 1/5 in both subdomains. Therefore, for oder $q = 4$ the solution is accurate to 10 K, and if we look at the mean error, it is accurate to 0.1 K.

For this case, we optimized the bandwidth only for the computation with consistency order $q = 2$, the consumed CPU time is slightly smaller than that of the first computation with order $q = 2$. For the computation with order $q = 4$, we turned off the bandwidth optimizer and computed on 64 processors, and we see that the CPU time is reduced to 15.5 h.

Figs. 5 and 6 show the temperature T on the surface of the module (a) and its error (b) as well as the heating curve of the centre of the 3rd chip from the left and of the centre of the degraded array of the 6th chip (c) for the computations with consistency order $q = 2$ and $q = 4$, respectively. Again, we see from the error pictures that the maximum errors occur only in few nodes.

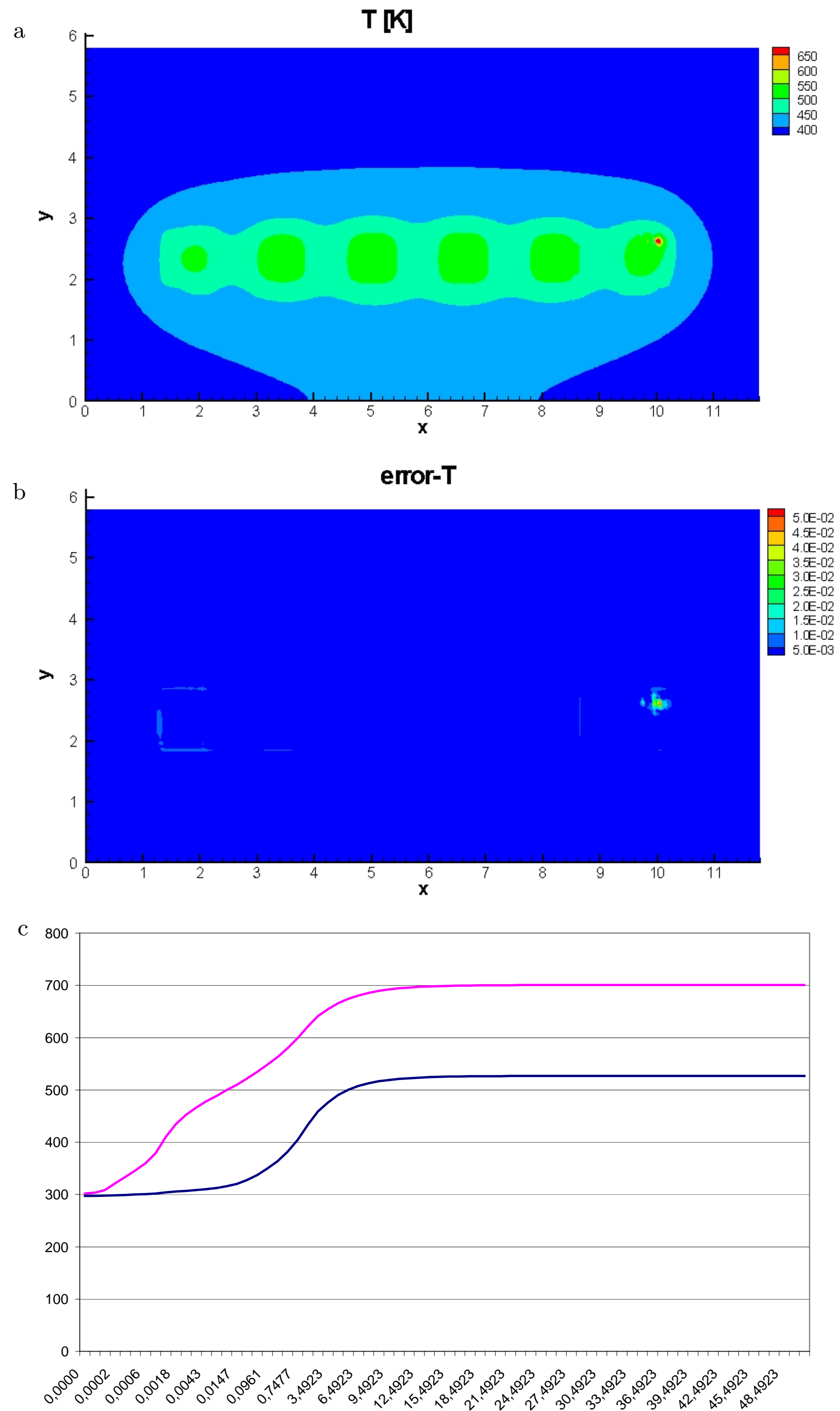

Fig. 5. Contour plot for the temperature T and its error on the top side of the module for $q = 2$ after 50 sec. (degenerated chip), heating curves for the centre of the 3rd chip from the left (blue) and for the centre of the degraded array of the rightmost chip (pink).

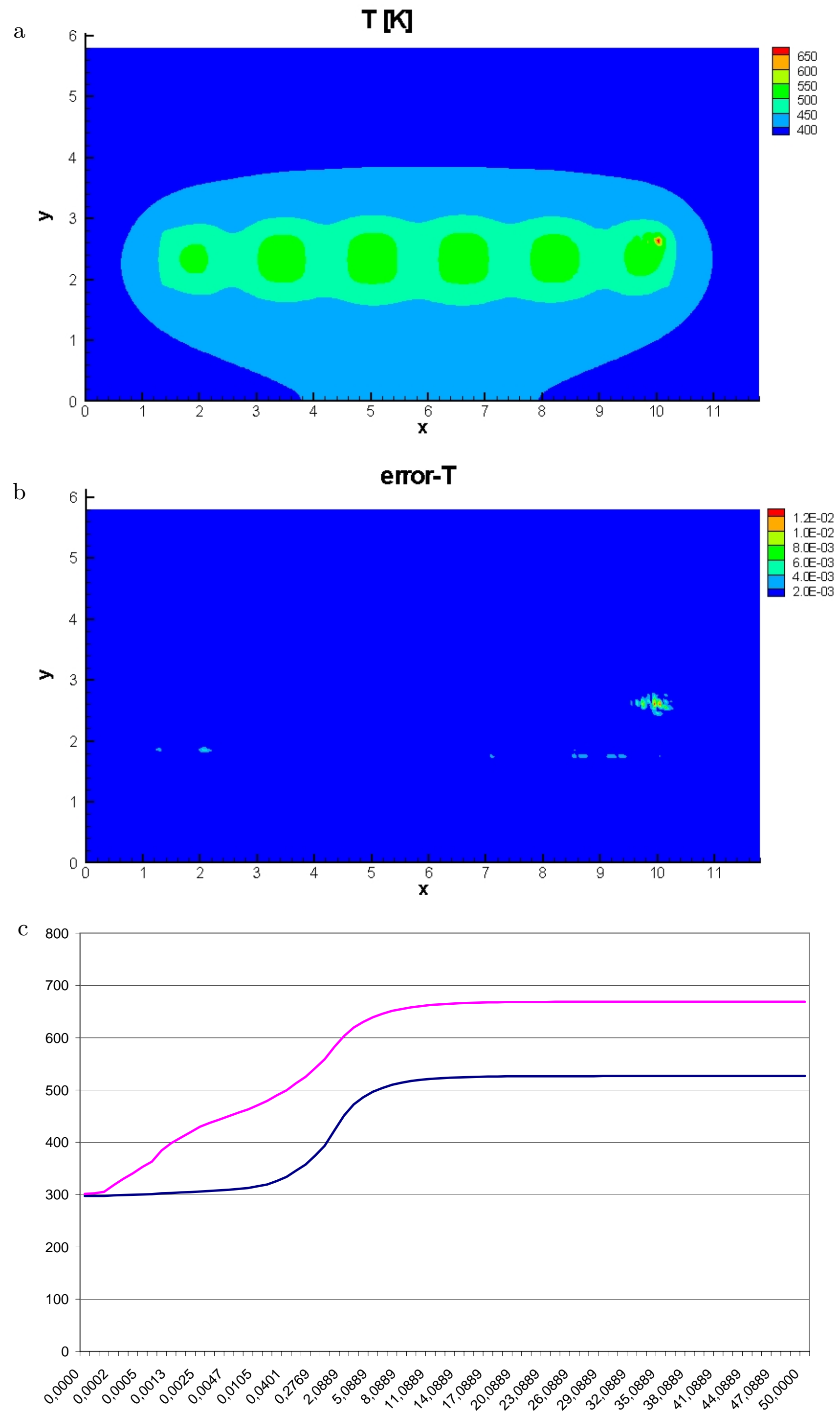

Fig. 6. Contour plot for the temperature T and its error on the top side of the module for $q = 4$ after 50 sec. (degenerated chip), heating curve for the centre of the 3rd chip from the left (blue) and for the centre of the degraded array of the rightmost chip (pink).

3 Concluding Remarks

This seemingly simple heat conduction problem turns out to be a rather critical one if we request a result in the 1% relative error class. At first we tried to generate a grid that increased gradually from the chip area to the interior of the module. However, the accuracy was bad. Our experience has shown us that uniform grids deliver the best accuracy. Therefore, we introduced a "sliding dividing line" (SDL) between the upper part with the chips and the remaining lower part. Here, the dividing line is in reality a dividing area that, in this case, does not slide, but we use its property of non-matching grid. This procedure then leads to acceptable results as shown in Tables 5 and 6. However, we have a 3-D problem that is always expensive in storage and computation. We have used the grids shown in Table 4 for the upper subdomain with 249561 nodes and the lower subdomain with 63189 nodes, totally 312750 nodes. With FDEM we compute with open eyes because our error estimate shows us the quality of the solution. We believe that we are the first to solve such a problem with the inclusion of an error estimate. This is possible because we use a finite difference method where we have an explicit and transparent access to the error, see [1, Sections 2.3 and 2.4]. This is a unique feature of FDEM. Another unique feature is that it is a black-box solver with respect to the PDEs, but also with respect to the solution domain with subdomains, and nevertheless is efficiently parallelized with MPI on distributed memory parallel computers, see [1, Section 2.8]. The $q = 4$ example of Table 6 has been computed on 64 processors.

References

1. Schönauer, W., Adolph, T., FDEM: The Evolution and Application of the Finite Difference Element Method (FDEM) Program Package for the Solution of Partial Differential Equations, Abschlussbericht des Verbundprojekts FDEM, Universität Karlsruhe, 2005, available at http://www.rz.uni-karlsruhe.de/rz/docs/FDEM/Literatur/fdem.pdf .

Supplement June 2007

The relative estimated errors are always relative to the maximum of the solution component. For this heat conduction problem, we want to give another relative error estimate, this time relative to the change in temperature C_{max} between the local ambient temperature T_a and the maximum temperature T_{max} after 50 sec. The most important place in the module is the position with the maximum temperature, so we also want to know the relative error at this position. We provide all this information in Tables 7–8 for the two problems we solved.

Table 7. Results of the first calculation with $H = 250\,\text{W/Chip}$ for all six chips, grid with 312,750 nodes.

			rel. estim. error		
		T_{max} [K]	max.[1]	max.[2]	max.[3]
Order	Subdomain	C_{max} [K]	mean[1]	mean[2]	
$q = 2$	upper	526.4	$0.11 \cdot 10^{-1}$	$0.25 \cdot 10^{-1}$	$0.66 \cdot 10^{-3}$
		229.4	$0.81 \cdot 10^{-3}$	$0.18 \cdot 10^{-2}$	
	lower	524.2	$0.69 \cdot 10^{-2}$	$0.16 \cdot 10^{-1}$	
		227.2	$0.17 \cdot 10^{-3}$	$0.39 \cdot 10^{-3}$	
$q = 4$	upper	526.2	$0.78 \cdot 10^{-2}$	$0.18 \cdot 10^{-1}$	$0.12 \cdot 10^{-3}$
		229.2	$0.17 \cdot 10^{-3}$	$0.40 \cdot 10^{-3}$	
	lower	524.1	$0.22 \cdot 10^{-2}$	$0.50 \cdot 10^{-2}$	
		227.1	$0.35 \cdot 10^{-4}$	$0.81 \cdot 10^{-4}$	

[1] maximum/mean error relative to maximum temperature
[2] maximum/mean error relative to maximum change in temperature
[3] error relative to maximum change in temperature for node with maximum temperature

The relative estimated error $\|\Delta u_d\|_{rel,1}$ is the error we usually present. If we have only one solution component, this global relative error is computed by

$$\|\Delta u_d\|_{rel} = \frac{\underset{k=1,n}{max} |\Delta u_{d,k}|}{\underset{k=1,n}{max} |u_{d,k}|}. \tag{10}$$

For the change in temperature, it holds

$$C_{max} = T_{max} - T_a, \tag{11}$$

and thus we get for the relative estimated error $\|\Delta u_d\|_{rel,2}$

$$\|\Delta u_d\|_{rel,2} = \frac{T_{max}}{C_{max}} \cdot \|\Delta u_d\|_{rel,1}. \tag{12}$$

For the first problem where all six chips have the same power dissipation, the maximum temperature T_{max} occurs in the centre of the third chip from the left. As the module will

suffer a damage if the temperature in only one point exceeds a certain maximum temperature, it is very interesting to know the error in the node k_{max} that is in the centre of the third chip. So it holds for $\|\Delta u_d\|_{rel,3}$:

$$\|\Delta u_d\|_{rel,3} = \frac{|\Delta u_{d,k_{max}}|}{C_{max}}. \tag{13}$$

Table 8. Results of the second calculation with $H = 250\,\text{W/Chip}$, 6th chip with degraded array, grid with 312,750 nodes.

Order	Subdomain	T_{max} [K] / C_{max} [K]	rel. estim. error max.[1] / mean[1]	max.[2] / mean[2]	max.[3]
$q = 2$	upper	706.3	$0.56 \cdot 10^{-1}$	$0.96 \cdot 10^{-1}$	$0.91 \cdot 10^{-1}$
		409.3	$0.62 \cdot 10^{-3}$	$0.11 \cdot 10^{-2}$	
	lower	632.7	$0.32 \cdot 10^{-1}$	$0.61 \cdot 10^{-1}$	
		335.7	$0.14 \cdot 10^{-3}$	$0.27 \cdot 10^{-3}$	
$q = 4$	upper	677.6	$0.15 \cdot 10^{-1}$	$0.27 \cdot 10^{-1}$	$0.24 \cdot 10^{-1}$
		380.6	$0.13 \cdot 10^{-3}$	$0.23 \cdot 10^{-3}$	
	lower	614.3	$0.73 \cdot 10^{-2}$	$0.14 \cdot 10^{-1}$	
		317.3	$0.26 \cdot 10^{-4}$	$0.50 \cdot 10^{-4}$	

[1] maximum/mean error relative to maximum temperature
[2] maximum/mean error relative to maximum change in temperature
[3] error relative to maximum change in temperature for node with maximum temperature

For the second problem with the degraded array on chip 6, the maximum temperature occurs in the centre of the degraded array. Therefore, the relative error $\|\Delta u_d\|_{rel,3}$ is computed in this position.

Additionally, we want to compare the two supercomputers that we mainly use presently. So we carry out the computations on the HP XC4000 with 2.6 GHz AMD Opteron processors and InfiniBand 4X interconnect that has been installed at the University of Karlsruhe, Germany, and on the SGI Altix 4700 with 1.6 GHz Intel Itanium2 Montecito Dual Core processors and NUMAlink 4 interconnect at the LRZ Munich, Germany.

We repeated the computation with consistency order $q = 4$ and the degraded array on chip 6 on 64, 128, 256 and 512 processors for the grid with 312,750 nodes. We use full LU preconditioning, but without any bandwidth optimizer because for this problem the bandwidth optimizer does not yield good results and therefore only consumes CPU time. Here, we sacrifice storage for CPU time. In Table 9, you can see the CPU time for master processor 1 for each computation.

You see that the computation time is reduced roughly by the factor 2 if we double the number of processors, at least up to 256 processors. For 512 processors the communication overhead strongly affects the computation time. As more than 99% of the computation time is comsumed by the linear solver LINSOL, there is still space for improvement.

Table 9. CPU times for the scalability test computations, grid with 312,750 nodes.

	CPU time [h]	
No. of proc.	HP XC4000	SGI Altix 4700
32	40.53	18.55
64	23.29	10.34
128	10.65	7.00
256	5.69	5.96
512	6.25	6.15

This is not the usual way to examine the scalability of a code. So we also tried to double the number of nodes in the three space directions which gives the eightfold number of nodes, and simultaneously compute with the eightfold number of processors. We tried to compute on 512 processors with a grid with 2,344,946 nodes, and wanted to compare the CPU time to that of the computation with 64 processors and 312,750 nodes. However, we must use LU preconditioning to solve the resulting linear system of equations, and then the factorization of the large sparse matrix is expected to consume the $4 \times 8 = 32$-fold computation time as we get the fourfold bandwidth of the matrix, and we have fill-in between the outer diagonals. So it is impossible to perform scalability tests this way.

We finally did measurements for the first time step with the grid with 2,344,946 nodes on 128, 256 and 512 processors. The CPU time for master processor 1 for the three computations are shown in Table 10.

Table 10. CPU times for the scalability test computations, grid with 2,344,946 nodes.

	CPU time [h]	
No. of proc.	HP XC4000	SGI Altix 4700
128	3.13	2.64
256	1.41	1.19
512	0.88	0.75

We see that the computation time is reduced by more than the expected factor 2 on both supercomputers, if we compare the computation on 256 processors to that on 128 processors, probably because of cache effects. For 512 processors, the communication overhead is the cause for the reduction factor of 1.6 instead of 2. For these computations, LINSOL consumed 99.99% of the CPU time.

These numerical experiments show that 3-D calculations are extremely expensive. Doubling the number of nodes in each space direction gives eightfold number of nodes, but as we get large sparse banded matrices at the same time, the bandwidth is the fourfold which results because of the fill-in in the $4 \times 8 = 32$-fold computational amount as explained above. If we would compute the solution for 77 time steps as requested for the snuffle problem, we needed for the grid with 2,344,946 nodes more than 85 hours (SGI) or 99 hours (HP) computation time for 512 processors, respectively. As the communication overhead affects

for this number of processors already significantly the computation time as seen in Table 10 (from 256 to 512 processors reduction factor 1.6 instead of 2 for computation time), we do not expect significant reductions in time for 1024 or 2048 processors.

Again, we want to stress that such technical problems never have been solved with error estimate up to now. Therefore, we repeat our experience that we gained for many technical examples: Happy are those people that do not see the errors.