HEAT DIFFUSION IN FRACTAL GEOMETRY COOLING SURFACE

by

Matjaz RAMŠAK * ***and Leopold ŠKERGET***
Faculty of Mechanical Engineering, University of Maribor, Maribor, Slovenia

Original scientific paper
DOI: 10.2298/TSCI1204955R

In the paper the numerical simulation of heat diffusion in the fractal geometry of Koch snowflake is presented using multidomain mixed Boundary Element Method. The idea and motivation of work is to improve the cooling of small electronic devices using fractal geometry of surface similar to cooling ribs. The heat diffusion is assumed as the only principle of heat transfer. The results are compared to the heat flux of a flat surface. The limiting case of infinite small fractal element is computed using Richardson extrapolation.

Key words: *heat transfer, cooling of electronic devices, boundary element method, fractals*

Introduction

There are many fields where fractals are used in engineering applications such as porous media modelling, nano fluids, fracture mechanics, and many more. For example, Huang *et al*. [1] used fractals to model surface morphology in order to numerically compute thermal contact resistance in the field of heat transfer as in our case. His principle assumption is to neglect heat convection and radiation thus the heat conduction is the only mechanism for heat transfer in small air pockets between two surfaces. The same assumption is applied in this article too.

We found the idea of using fractals to model cooling surface from the fundamental question: "How long is the coast of Britain?" stated in paper by Mandelbrot [2]. Using smaller and smaller ruler the coast length limits to infinite, while the area is obvious finite. If this logic is applied to small electronic device cooler geometry, large efficiencies should be obtained using "infinite" cooling surface area in 3-D or length in 2-D.

The Boundary Element Method (BEM) has a long tradition in our research group. Presented numerical scheme is based on our work [3], where multidomain BEM is used to solve unsteady laminar flow. Two ideas are used in this work: mixed boundary elements and multidomain method. Both ideas lead to a sparse overdetermined system matrix solved in linear least square manner using fast iterative solver. In contrast to the classic boundary element method large problems could be solved consisted of a million mesh nodes. In the next our work [4], the ideas are extended to a 3-D numerical algorithm. The leading numerical example solved is a potential flow past a complex geometry of an airplane.

* Corresponding author; e-mail: matjaz.ramsak@uni-mb.si

Problem definition

The problem considered is a cut out of a small electronic device as shown in fig. 1.

The physical model assumptions follow. First, the major assumption is that the heat conduction is the only mechanism of heat transfer thus neglecting heat convection in air and radiation. The basis of this assumption is that the dimension of the problem is small, for example 1 cm, and still air around this device. Another reason is small temperature difference causing small buoyant forces and causing minimal heat radiation. Second assumption is a steady-state condition.

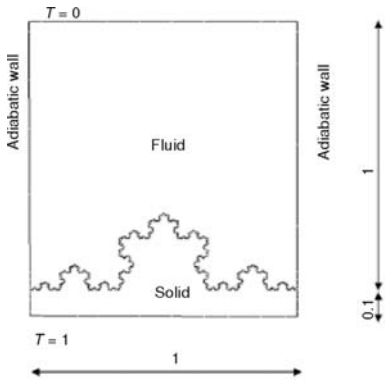

Figure 1. The geometry and boundary conditions of the air cooled fractal electronic device

The problem has been non-dimensionalised as follows. Length quantities x and y are scaled with the length of the cut-off. Normalized temperature is obtained by scaling with the temperature difference between hot and cold walls. The material properties could be treated in two ways. Since the steady-state is computed the solution is only heat conductivity depended. If governing equation is non-dimensionalised using air conductivity, *e. g.* fluid non-dimensional conductivity becomes 1 and solid conductivity is equal to the ratio of solid and air conductivity. In the case of Cu, Al or steel cooler this ratio is 16000, 9000, and 2300, respectively. Such high values could cause numerical instabilities. The second possibility of non-dimensionalising is to use heat diffusivity a instead conductivity. In this manner the diffusivity ratio is only 5.4, 4.4, and 0.8. Anyway, interface boundary condition equals heat flux between solid and fluid resulting in high value heat conductivity ratio in system matrix in both cases of non-dimensionalising. This is the case, where complicated physic could not be overcome by numerical trick.

The non-dimensional governing equation is a Laplace equation:

$$\frac{\partial}{\partial x_j}\left(\lambda \frac{\partial T}{\partial x_j}\right) = 0 \tag{1}$$

where λ is a conductivity as a function of place. The boundary conditions are strait forward as shown in fig. 1.

Boundary element method

For introduction on BEM please refer to the basic literature such as [5]. Multidomain mixed BEM numerical algorithm for solving 2-D diffusion problems will be described in this section.

The developed solver has to fulfill following demands: accuracy, stability at high conductivity ratios, economy of the solution and ability to solve high grid density with high longest/shortest element ratio. The accuracy is a well-known quality of BEM. In comparison to the other numerical methods, the accuracy is more evident using low grid densities, see [3]. The stability is a common problem using BEM. The stability of our BEM numerical algorithm is the main topic in our previous work [8], where considerable effort was done to increase the stability at a convection dominated laminar flows. The result was a stable algorithm solving a driven cav-

ity case up to Reynolds number value 50.000. The usual maximal value is 10.000 to 15.000 using BEM.

Differential form

The Laplace equation can be written in general form as

$$\frac{\partial}{\partial x_j}\left(a\frac{\partial u}{\partial x_j}\right)=0 \tag{2}$$

where u is the scalar field function and a – a diffusivity for the sake of generality. In our case a is equal λ. Boundary conditions on the boundary Γ must be known:

$$u=\overline{u}, \quad \text{on} \quad \Gamma_1 \tag{3}$$

and

$$\frac{\partial u}{\partial n}=\overline{\frac{\partial u}{\partial n}} \quad \text{on} \quad \Gamma_2 \tag{4}$$

The main reason to apply the subdomain technique is to account different material properties since the Greens functions are available only for differential equations with constant coefficients, which is one of the restrictions of BEM. Another crucial reason for multidomain BEM is to make system and integral matrices sparse, a result of small dense blocks arising from connections of subdomain nodes.

As each subdomain is treated as a single entity, the interface nodes have to communicate results between subdomains. This is possible in an iterative way (classical iterative Schwarz algorithm) or in an implicit way, where at the interface boundaries between the subdomains I and II the compatibility interface condition for u are applied:

$$u_{\mathrm{I}}=u_{\mathrm{II}} \tag{5}$$

as well as the equilibrium interface condition:

$$\lambda_{\mathrm{I}}\frac{\partial u}{\partial n_{\mathrm{I}}}=-\lambda_{\mathrm{II}}\frac{\partial u}{\partial n_{\mathrm{II}}} \tag{6}$$

where λ is a diffusion factor for the sake of generality.

Integral form

The general form of the differential diffusion eq. (2) can be transformed into an equivalent integral statement, see [5]:

$$0=-c(\xi)u(\xi)-a\int_{\Gamma}u\frac{\partial u^*}{\partial n}\mathrm{d}\Gamma+a\int_{\Gamma}u\frac{\partial u^*}{\partial n}u^*\mathrm{d}\Gamma \tag{7}$$

where a is a constant diffusivity over subdomain. The variable u^* is the elliptic fundamental solution, *i. e*. the solution of the equation:

$$\frac{\partial^2 u^*}{\partial x_j\,\partial x_j}+\delta(\xi,s)=0 \tag{8}$$

and given for the plane case as:

$$u^* = \frac{1}{2\pi}\ln\left(\frac{1}{r(\xi,s)}\right)$$
$$\frac{\partial u^*}{\partial x_j} n_j = \frac{r_j n_j}{2\pi r^2} \tag{9}$$

where $r_j(\xi, s)$ is the vector from the source point ξ to the reference field point s, *e. g.*, $r_j = x_j(\xi) - x_j(s)$ for $j = 1, 2$ in planar case while r is its magnitude $r = |r_j|$, and n_j – the unit normal vector to the boundary element at the nodal point.

Discrete form

Let us introduce some notation comments. The integral eq. (7) is valid for any arbitrary geometry. From now on, in order to distinguish the solution domain Ω and its exterior boundary Γ from the subdomain geometry, the subdomain domain is denoted by Ω_s and the subdomain boundary by Γ_s consistently.

The idea of mixed elements is an alternative approach in BEM, [6]. The basic idea of mixed elements is to split the field function and normal flux nodal points in order to keep the advantages of function continuous approximation. If the flux is approximated using continuous approximation then the unit normal vectors are non-uniquely defined at the interpolation nodal point at boundary element corner. Therefore, the function is approximated using continuous interpolation polynomials while the normal flux is interpolated using discontinuous interpolation polynomials. As a consequence, the advantages of the continuous field function approximation are retained and its conservation property is preserved while the normal flux values are modeled in a proper way. When using continuous elements, the application of the matching conditions of common interfaces, *i. e*. the matrix assembly, leads to an over-determined system of algebraic equations. If the over-determined system is reduced to a square one by neglecting some of the boundary integral equations, the reduced system matrix is not consistent with the initial one, resulting in unstable numerical scheme for Reynolds number higher than 1,000 in the case of the driven cavity flow, [7]. Instead of using one of the several schemes that reduces the over-determined system to a closed system the original over-determined system matrix is solved in a least squares sense in this work.

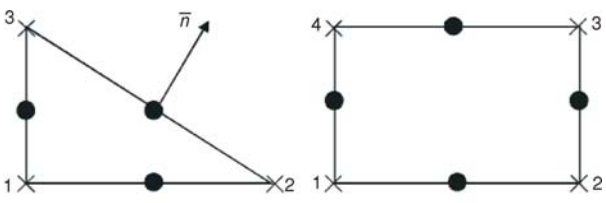

Figure 2. Subdomain approximation using triangular $N = 3$ and quadrilateral $N = 4$ cell. The × denotes function u boundary nodes and the • denotes normal flux $\partial u/\partial n$ boundary nodes. Discontinuous constant normal flux approximation over boundary element ($n = 1$) with unique normal direction $\vec{n}$ and continuous linear field function approximation over a boundary element $n = 2$

Following the mixed boundary elements idea the simplest possible discretization is the continuous linear field function approximation over the boundary element and the constant approximation of its derivative in the normal direction to the boundary element (normal flux), fig. 2.

An unknown field function is approximated with continuous linear interpolation polynomials over the boundary element $\Phi^{n=2}$, fig. 2:

$$\Phi^{n=2} = \frac{1}{2}\begin{Bmatrix} 1-\xi \\ 1+\xi \end{Bmatrix} \tag{10}$$

where $\xi \in [-1, 1]$ is the local co-ordinate system and $n = 2$ – the degree of freedom. The unknown normal flux is approximated with constant interpolation polynomials $\Phi^{n=1}$ as:

$$\Phi^{n=1} = \{1\} \tag{11}$$

Computing Laplace eq. (2) only there is no need to the function domain approximation. Using Greens functions the domain problem is transferred to the boundary completely, see eq. (7). The boundary integrals over boundary element Γ_e can be written as:

$$h^n = \int_{\Gamma_e} \Phi^n \frac{\partial u^*}{\partial n} \mathrm{d}\Gamma_e, \quad g^n = \int_{\Gamma_e} \Phi^n u^* \mathrm{d}\Gamma_e \tag{12}$$

Using elliptic fundamental solution, the integrals are functions of the geometry only.

The complete boundary integral equation over the subdomain boundary Γ_s will be written as the sum of all individual boundary integrals Γ_e surrounding the subdomain. In the vertex nodal point the contribution $\{h\}$ of both neighboring boundary integrals Γ_e could be summed up as $\{h'\}$. The obtained boundary integral over the subdomain boundary Γ_s has N degrees of freedom. The boundary integral discretization has the following form:

$$\int_{\Gamma_s} \frac{\partial u^*}{\partial n} u \mathrm{d}\Gamma_s = \sum_{e=1}^{N} \left\{ \int_{\Gamma_e} \Phi^n \frac{\partial u^*}{\partial n} \mathrm{d}\Gamma_e \right\}^T \{u\}^n = \sum_{e=1}^{N} \{h^n\}^T \{u\}^n = \sum_{i=1}^{N} \{h'\}^T \{u\}^N \tag{13}$$

where the last sum N represents the loop over subdomain interpolation boundary nodal points. In the case of triangular cells $N = 3$ and quadrilateral cells $N = 4$.

The normal flux interpolation is simple. As mentioned before, the constant approximation of normal flux is prescribed to the boundary element, eq. (11). The boundary integral of the normal flux over the complete boundary of subdomain Γ_s is computed as a sum of all individual boundary integrals Γ_e. Thus the N elements with only one nodal point each, are written as

$$\int_{\Gamma_s} \frac{\partial u^*}{\partial n} u^* \mathrm{d}\Gamma_s = \sum_{e=1}^{N} \left\{ \int_{\Gamma_e} \Phi^n u^* \mathrm{d}\Gamma_e \right\}^T \left\{ \frac{\partial u}{\partial n} \right\}^n = \sum_{e=1}^{N} \{g^n\}^T \left\{ \frac{\partial u}{\partial n} \right\}^n = \sum_{i=1}^{N} \{g\}^T \left\{ \frac{\partial u}{\partial n} \right\}^M \tag{14}$$

where the last sum represents the loop overall M boundary normal flux nodal points. In order to distinguish the function and flux nodes, the summation index M is used although its value is equal to N.

Let us write the boundary integral eq. (7) for an individual subdomain in discrete form using discrete equations for function (13) and the normal flux (14) as:

$$0 = -c(\xi)u(\xi) - a\sum_{i=1}^{N} \{h'\}^T \{u\}^N + a\sum_{i=1}^{M} \{g\}^T \left\{ \frac{\partial u}{\partial n} \right\}^M \tag{15}$$

where index i means the sum over N function nodal points and M normal flux nodal points of individual subdomain, see fig. 2. The new variable e is introduced as

$$e = -[c(\xi) + ah']$$

Using new variables the discretized integral equation for the subdomain is written as:

$$\sum_{i=1}^{N} \{e\}^T \{u\}^N - \sum_{i=1}^{M} \{g\}^T \left\{ \frac{\partial u}{\partial n} \right\}^M = 0 \tag{16}$$

The eq. (16) is the discrete form of the integral boundary eq. (7) at source point ξ. The complete system matrix for one subdomain is obtained by writing the eq. (16) for all function and normal flux boundary nodal points. The source point is thus located in $\xi = 1$; N function nodal points and $\xi = 1$; M normal flux nodal points resulting in $2 \times N$ equations for subdomain written as:

$$[\mathrm{E}]\{u\} = [\mathrm{G}]\left\{\frac{\partial u}{\partial n}\right\} = \{0\} \tag{17}$$

Integral accuracy

Integral accuracy is the crucial for accurate and stable final solution. The integrals of fundamental solution over boundary elements eq. (12) are computed numerically using Gauss integration. In order to increase accuracy the element subdivision is applied rather than using large number of gauss integration points, see [3]. The integral accuracy is some kind of compromise regarding computational time. Anyway the measure of integral accuracy is needed. If at all subdomain function nodes the potential 1 is prescribed, than the resulting flux is 0. In this case the discrete boundary integral eq. (16) is reducing to:

$$\sum_{\Gamma_S} e = 0 \tag{18}$$

At each subdomain cell source node, the sum of e integrals must be zero. Checking this over all subdomain source nodes $N + M$ and over all solution domain cells NC the standard deviation of integral error INT_{RMS} is computed as:

$$INT_{\mathrm{RMS}} = \sqrt{\frac{1}{NC(N+M)}\sum_{\Gamma_S} e^2} \tag{19}$$

This is a measure of some kind overall average integral accuracy error. To be more specic, 68% of integrals are of better accuracy than INT_{RMS}. Another useful measure is also the maximum error denoted as INT_{MAX}. The MAX error is usual 10 times higher than RMS error.

Implementation of boundary conditions on boundaries of computational domain

The nodal points on boundaries of computational domain are described first, as they are dependent on the physical conditions, relating computational domain to its surroundings. On the contrary, the interface conditions, which have to be imposed between the subdomains, are formally the same for all nodal points on subdomain interface boundaries.

The matrix form of discrete boundary integral eq. (17) is transformed to the system of algebraic equations by applying the boundary conditions (3), the known function value $\overline{u}$ on the boundary Γ_1 and the known normal flux value $\overline{\partial u/\partial n}$ on Γ_2 as defined in eq. (4):

$$\begin{bmatrix} [-\mathrm{G}]^{\Gamma_1} \\ [\mathrm{E}]^{\Gamma_2} \end{bmatrix} \begin{Bmatrix} \left\{\frac{\partial u}{\partial n}\right\}^{\Gamma_1} \\ \{u\}^{\Gamma_2} \end{Bmatrix} = \begin{Bmatrix} -[\mathrm{E}]\{\overline{u}\}^{\Gamma_1} \\ [\mathrm{G}]\left\{\frac{\partial u}{\partial n}\right\}^{\Gamma_2} \end{Bmatrix} = \{f\} \tag{20}$$

and further to the

$$[\mathrm{A}]\{x\} = \{F\} \tag{21}$$

Implementation of interface conditions between subdomains

The normal flux and function nodal points on the boundary element are treated differently, which comes from the order of interpolation functions used. We will consider them separately, see fig. 3.

Normal flux boundary element nodal points. Let us consider two flux points in contact on the interface between two subdomains I and II, see fig. 3 (left). In subdomain I the unknown flux value at nodal point I is denoted as $\partial u/\partial n_{\mathrm{I}}$. Following the same notation $\partial u/\partial n_{\mathrm{II}}$ is the unknown value in subdomain II. The discretized integral boundary eq. (16) can be written for subdomain I as:

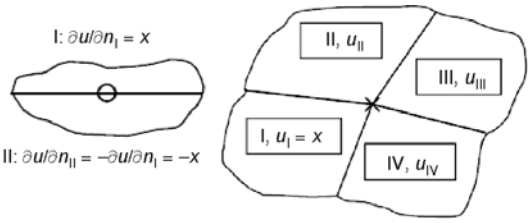

Figure 3. The normal flux (left) and the function (right) nodal points on the boundary element and implementation of the interface boundary condition. A subdomain indexes are I, II, III, and IV for example if four subdomains are connected at function nodal point

$$\left(\sum_{i=1}^{N} \{e\}^T \{u\}^N \right)_{\mathrm{I}} - \left(\sum_{i=1}^{M-1} \{g\}^T \left\{ \frac{\partial u}{\partial n} \right\}^{M-1} + g_{\mathrm{I}} \frac{\partial u}{\partial n_{\mathrm{I}}} \right)_{\mathrm{I}} = f_{\mathrm{I}} \tag{22}$$

and for the subdomain II as:

$$\left(\sum_{i=1}^{N} \{e\}^T \{u\}^N \right)_{\mathrm{II}} - \left(\sum_{i=1}^{M-1} \{g\}^T \left\{ \frac{\partial u}{\partial n} \right\}^{M-1} + g_{\mathrm{II}} \frac{\partial u}{\partial n_{\mathrm{II}}} \right)_{\mathrm{II}} = f_{\mathrm{II}} \tag{23}$$

where the integrals with index I are computed for subdomain I and the integrals II for subdomain II. For the sake of simplicity the sum terms on the left side of equations will be omitted in the further text. From the equilibrium interface condition (6) an additional equation is obtained which reduces the number of unknowns from two to one. Let us choose the unknown flux to be the value of $\partial u/\partial n_{\mathrm{I}}$ at subdomain I. With this unknown chosen, the above equations can be rewritten as:

$$-g_{\mathrm{I}} \frac{\partial u}{\partial n_{\mathrm{I}}} = f_{\mathrm{I}} \tag{24}$$

$$+\frac{\lambda_{\mathrm{I}}}{\lambda_{\mathrm{II}}} g_{\mathrm{II}} \frac{\partial u}{\partial n_{\mathrm{I}}} = f_{\mathrm{II}} \tag{25}$$

to form an over-determined system of two equations with only one unknown.

Function boundary element nodal points. Because of topological aspects of the vertex points, see fig. 3 (right), the application of the interface boundary conditions to the subdomain vertex nodal points are not so straight forward as for the flux points. For example shown in fig. 3, where vertex function nodal point is surrounded by 4 subdomains, the discrete form of the integral boundary eq. (16) for subdomain I at the vertex nodal point I can be written as:

$$e_{\mathrm{I}} u_{\mathrm{I}} = f_{\mathrm{I}}$$

where the rest sums are omitted again. In a similar way, the other three equations are obtained at subdomains II, III, and IV:

$$e_{II}u_{II} = f_{II}$$
$$e_{III}u_{III} = f_{III}$$
$$e_{IV}u_{IV} = f_{IV}$$

forming together four linear independent equations. The implementation of the function compatibility interface condition is straightforward. Let us set the unknown function value at the vertex nodal point as u_I and rewrite the compatibility interface condition as:

$$u_I = u_{II} = u_{III} = u_{IV} = u_I \tag{26}$$

To summarize, four equations are available with only one unknown, leading to the over-determined system matrix.

Solving the over-determined system of equations

The over-determined system matrix A is sparse and block structured. The iterative linear least squares solver of [9] is used to obtain the solution to the system of equations. The method is based in bidiagonalisation procedure. It is analytically equivalent to the standard method of conjugate gradients, for details see [9]. To accelerate the convergence a diagonal preconditioning method is applied, see [10].

The basic question connected with using the over-determined matrix is the influence of additional equations on CPU consumption. To find the answer to this question the comparison is made with the classic subdomain BEM using discontinuous boundary elements with ILU preconditioned iterative conjugate gradient solver for sparse matrices. The matrix which was supposed to be of equal size was defined based on the same number of unknowns NU. For example a square matrix of size NU × NU is equivalent to the over-determined matrix size NEQ × NU where the number of equations NEQ is higher than NU. It has to be emphasized that the square matrix is obtained using a 12-node quadratic discontinuous subdomain, which is approximately equal to four 4-node continuous linear subdomains. For example the over-determined matrix resulting from a grid of 16 × 16 linear subdomains has the same number of unknowns NU as 8 × 8 quadratic subdomains. The number of non-zero matrix elements is approximately 10% lower in the present BEM numerical scheme. The steady-state scalar diffusion problem is solved. The CPU time necessary to achieve approximately the same solution accuracy is lower for the over-determined matrix than for the square matrix, leading to the conclusion that the over determination of the system does not negatively influence the computational time needed for the solution of the system of equations, see fig. 4.

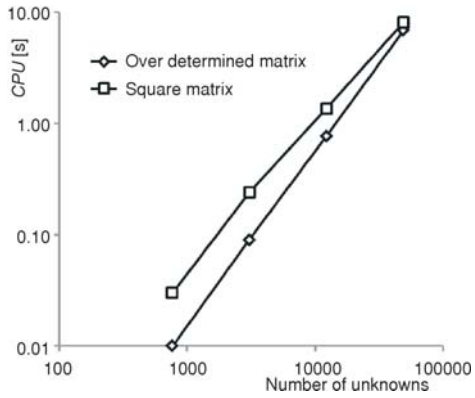

Figure 4. CPU comparison between the present over-determined matrix solver and the iterative solver for the square matrix with the same number of unknowns

Verification of developed numerical method

The developed numerical method is verified and validated using simple flat cooling

surface geometry. The boundary conditions are the same as shown in fig. 1. The test case is equivalent to heat diffusion in two successive walls with the wall heat conductivity ratio as a parameter. The analytical solution is elementary. The accuracy of computed results is measured using maximal error and its standard deviation denoted as ERR computed as:

$$ERR = \sqrt{\frac{1}{NM}\sum_{i=1}^{NM}(T_{\mathrm{BEM}} - T_{\mathrm{EXACT}})^2} \tag{27}$$

where summation is performed over all mesh points NM. Since the temperatures are non-dimensionalized there is no need for ERR normalization.

Computing a million mesh nodes

The first aim is to demonstrate accuracy using higher mesh densities. The results are graphically presented in fig. 5 for quadrilateral mesh and in fig. 6 for triangular mesh. The heat conductivity ratio is 1.

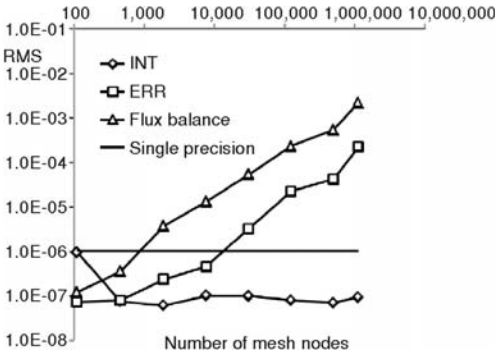

Figure 5. Influence of mesh density for uniform quadrilateral mesh. INT is integral accuracy computed as stated in eq. (19), ERR is obtained result error using eq. (27). Flux balance is error of BEM computed flux conservation

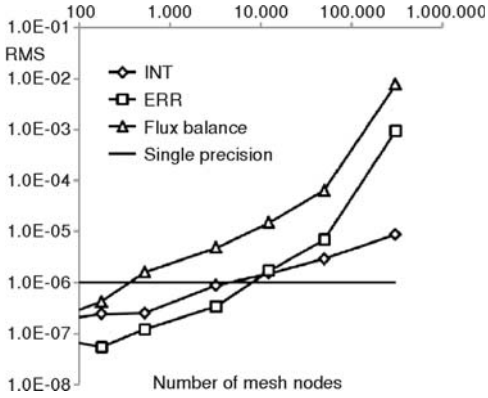

Figure 6. Influence of mesh density for triangular mesh (see caption at fig. 5)

Results using both types of meshes show similar behavior. Results are similar to our 3-D BEM algorithm presented in [4]. While integral accuracy INT is of single precision order of accuracy, the results ERR is higher using higher mesh density. Naturally, one should expect better results using higher mesh density. Using meshes up to 50,000 mesh points the single precision accuracy is obtained. The reader should be reminded that all used FORTRAN program variables are stored as a single precision values. Using higher mesh densities, the results are losing their accuracy. The authors suspect, that the main reason lies in system matrix and its condition number. Namely, system matrix becomes extremely large. While using FEM and FVM multimillion meshes are common, this is not the case using BEM. The authors could not find a million mesh nodes solved using BEM in literature. In our biggest case, a 1.1 million mesh nodes 2 GB of computer memory and one hour of 3 GHz CPU time is used. The results accuracy is of 1e-4 order, which is equal to 0.01.

Ratio longest/shortest boundary element

In order to compute fractal geometries, the non-uniform meshes are unavoidable with large ratio longest/shortest element. While very short elements are necessary to represent fractal edge, long elements are good enough for straight edges, see fig. 1. The aim of present section is to investigate the influence of overall ratio longest/shortest boundary element used. The test case is the simplest geometry with straight fluid solid interface. The discretization of solution domain is performed using quadrilateral mesh with constant number of mesh nodes and variable non-uniformity, see fig. 7. The main reason is to have a higher control in the quadrilateral mesh creation in contrast to triangular. In this manner high ratio values are obtained and extreme distorted cells. The overall ratio longest/shortest boundary element is equal to the area ratio between largest and smallest cell since the element distribution in horizontal direction is uniform. The cell distorted ratio defined as long/short size is approximately the same as overall longest/shortest ratio. The conductivity ratio is kept to 1.

The results of numerical experiment follow (fig. 8). Using ratios between 1 and 10,000 the results are practically of the same accuracy. This is the advantage of developed BEM in contrast to FEM or FVM which suers using high distorted elements. Similar numerical experiment was done using triangular mesh as shown in fig. 7 obtaining similar conclusion. The key of such ratio independency is an excellent fundamental solution integration algorithm using cell subdivision according to its distortion, for details see [8].

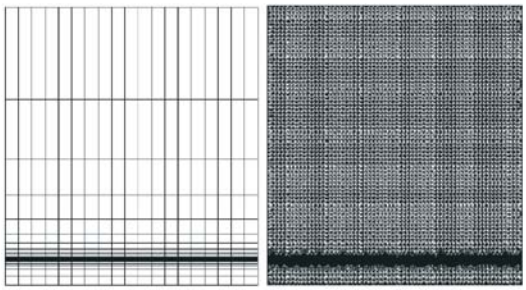

Figure 7. Extreme quadrilateral mesh with large ratio longest/shortest element of value 4000 and 600 mesh nodes (left hand side). Typical triangular mesh used for fractal geometries with increased density along fluid solid interface (right hand side). Number of mesh nodes is less than 10,000

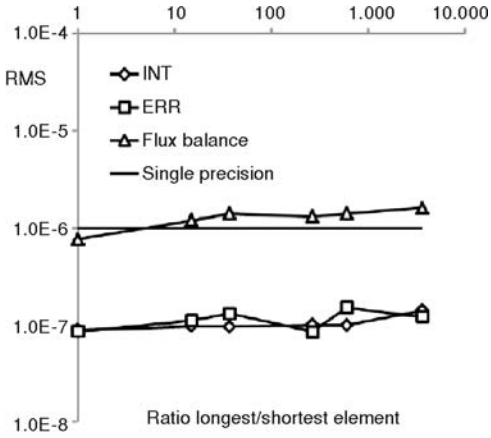

Figure 8. Results for variable ratio longest/shortest element

Conductivity ratio

The aim of present numerical experiment is to validate developed algorithm on fluid/solid conductivity ratio. This ratio is similar to Reynolds number value in computational fluid dynamics (CFD) applications. It is well known, that high conductivity ratios between cells are causing numerical instabilities even in simple heat diffusion problems using all numerical methods. The source of instabilities is numerical fundamental problem. In contrast to the previous numerical example, the non-uniform triangular mesh is used, see fig. 6. The topology of the mesh is similar to those used in fractal geometries in next section.

Results of conductivity ratio test are shown in fig. 9. The fundamental solution integrals are independent on ratio in this case. Using higher ratio values the accuracy is worse as suspected. The reason lies in interface boundary condition implying heat flux equilibrium thus involving the conductivity ratio. Increasing the ratio the ERR is increased in linear manner. Approaching to the extreme ratio value 10^6 the flux balance is approaching to the unity 1. The explanation is simple. If to the system matrix value approx. 1 in single precision another value of 10^{-6} is added, the original value does not change. Another manifest of this fact is flux balance. In this test instead of overall flux balance, the separated flux balance is shown for fluid and solid. While solid flux balance is fulfilled excellently and independently on ratio, fluid flux balance is bad, similar to ERR. Again the reason lies in different values orders which are summed. Heat flux, *e. g*. product of conductivity and temperature gradient, equilibrium at solid fluid interface is enforced by interface boundary condition. At fluid side of interface conductivity and temperature gradient should have value approx. 1, while in solid side these values should be 10^6 and 10^{-6}, respectively. Temperature gradient is implicit result obtained from system matrix as its solution. In solid side the flux error is approx. 100%, while in fluid side the solution is almost exact. In contrast to the computed flux accuracy the flux balance in fluid is bad while in solid is fulfilled perfectly.

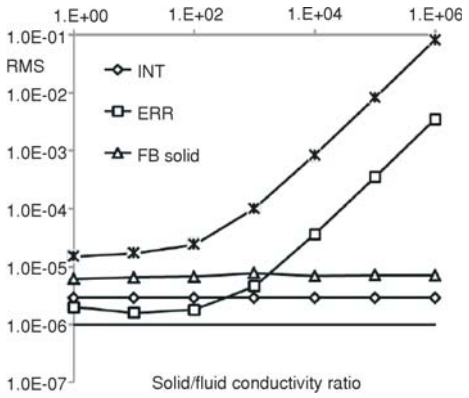

Figure 9. Results for variable solid/fluid conductivity ratio. Flux balance is plotted for solid and fluid separately

Results and discussion

In this section the results of fractal geometry cooling surface are presented. The solid fluid ratio is 10^4 which suits to the aluminum air conductivity ratio. Results are presented graphically in fig. 10 and numerically in tab. 1. Each line in table, leading with fractal length represents the computed case in fig. 11. Flat cooling surface is denoted with fractal length value 1.0 and first plot in fig. 12, *etc*. Temperature contours for the smalest fractal lenght are shown in fig. 12.

Table 1. Result for fractal geometry sequence, according to fig. 10. The flat cooling surface is presented using fractal length 1.0. Delta is heat flux improvement in successive manner

Fractal length	Interface length	Heat flux	Delta [%]
1.000	1.000	1.000	
0.333	1.333	1.157	15.70
0.111	1.778	1.179	2.22
0.037	2.370	1.192	1.29
0.012	3.160	1.202	1.03
0.004	4.214	1.212	0.97

According to the basic idea stated in introduction, increasing the number of fractal elements the interface length is increased and cooling heat flux is increased too, see tab. 1 and fig. 11.

Fluid solid interface length is increased for factor 4/3 for each

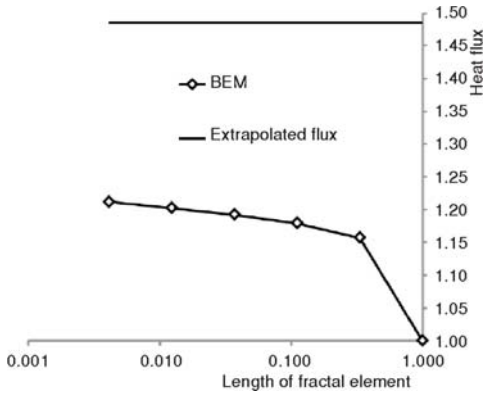

Figure 10. Cooling heat flux depending on fractal element length. The straight line is extrapolated value to zero fractal element length

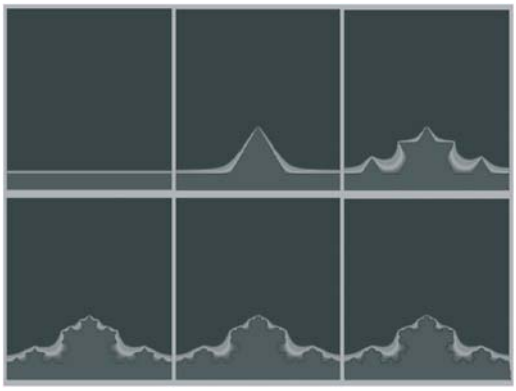

Figure 11. Temperature contours between 0.98 and 1.00 for sequence of fractal geometries. First figure top left is at surface representing fractal length 1.0. Top right figure is second fractal 0.333. Next fractals are 0.111, 0.037, 0.012 and finally 0.004 length units

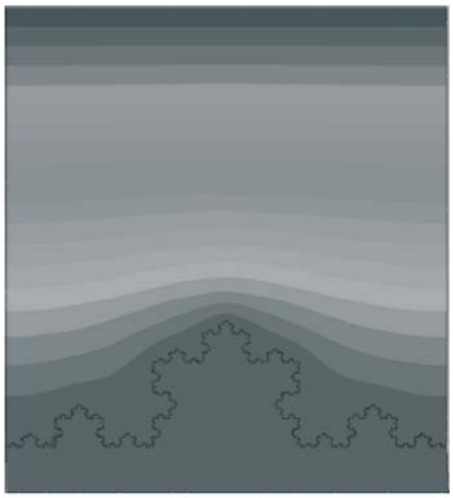

Figure 12. Temperature contours between 0 and 1 for the smallest computed fractal length 0.004 units

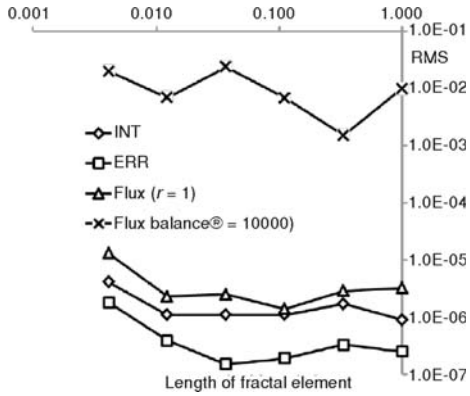

Figure 13. Results for fractal geometries. Comparison to analytical solution for conductivity ratio $r = 1$. Flux balance for conductivity ratio $r = 1$ and $r = 10^4$

fractal length iteration (second column in tab. 1), while heat flux is increased less each iteration (third column in tab. 1). Now, the self-arising question is the behavior of heat flux reducing the fractal length to zero. Interface length limits to infinity, but computed heat flux limits to a finite value as expected naturally. The first evidence of this is decreasing rate of heat ux change denoted by Delta in tab. 1. The second evidence is extrapolated value to zero fractal element length using Richardson extrapolation, [11]. The method is used for estimation and reporting of uncertainty due to discretization in CFD applications in Journal of Fluids Engineering. Our case does not match the explained method completely since the geometry of the mesh is changing. However, the nature of breaking the geometry to smaller fractal elements is similar to the mesh size reducing at fixed geometry. Extrapolated maximal heat flux value is 1:49 for zero fractal ele-

ment length. In this manner the main conclusion is approx. 50% gain of cooling heat flux according to the at surface case. The smallest computed fractal element is 0:004 non-dimensional units. If the cooling surface is 1cm long, the smallest computed fractal element length would be 0.04 mm. This is comparable to the lower bound of the heat diffusion scale represented by the length of the thermal phonon which is in the range between 0.1mm and 0.01 mm.

Computing the case, two numerical problems are found. First, extremely high conductivity ratio as discussed in previous section. In order to demonstrate this error source as the major one in this case, the conductivity ratio 1 is computed too. Having the conductivity ratio 1, the case is reducing to the simple heat conduction in homogeneous square with linear analytical solution. The results are excellent, see fig. 13. ERR and flux balance is of the same order as single precision 10^{-6}. Increasing the ratio up to 10^4 the flux balance is the only measure of results accuracy. Having the flux balance well fulfilled the results might be well to. Better conclusion is: if the flux balance is bad, the results are bad. Following this and based on our experience gained during the verification process the numerical results may be a few percent accurate.

The second source of error is enormous number of mesh nodes necessary to discretized small fractal elements, as discussed in the section *Computing a million mesh nodes*. This is indicated as slightest increasing error for conductivity ratio 1 and smallest fractal elements in fig. 13.

Conclusions

The heat conduction of fractal geometry cooling surface is solved using BEM. Three numerical problems are successfully solved using developed multidomain BEM during verification process.

First problem is computing high mesh density up to a million mesh nodes, second is highly distorted cells and last problem is high conductivity ratio between fluid and solid region.

Using Richardson extrapolation for determining the zero length fractal elements, the cooling capacity is 50% higher comparing to the at surface cooler. Much higher efficiency should be obtained using classic ribs. Another possibility would be using a variation of Koch snowflake with longer and slender ribs. In the future the heat convection will be included in fluid region.

References

[1] Huang, H., Xu, X., Effects of Surface Morphology on Thermal Contact Resistance, *Thermal Science 15* (2011), Suppl. 1, pp. S33-S38
[2] Mandelbrot, B., How Long is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension, *Science*, *New Series, 156* (1967), 3775, pp. 636-638
[3] Ramsak, M., *et al*., A Multidomain Boundary Element Method for Unsteady Laminar Flow Using Stream Function-Vorticity Equations, *Eng Anal Bound Elem, 29* (2005), 1, pp. 1-14
[4] Ramsak, M., Skerget, L., 3-D Multidomain Bem for Solving the Laplace Equation, *Eng Anal Bound Elem, 31* (2007), 6, pp. 528-538
[5] Brebbia, C. A., The Boundary Element Method for Engineers, Pentech Press, London, 1978
[6] Ramsak, M., Skerget, L., Mixed Boundary Elements for Laminar Flows, *Int. J Num Meth. Fluids, 31* (1999), 5, pp. 861-877
[7] Grigoriev, M. M., Dargush, G. F., A Poly-Region Boundary Elementh Method for Incompressible Viscous Fluid Flows, *Int. J. Num. Meth. Fluids, 46* (1999), 7, pp. 1127-1158
[8] Ramsak, M., Škerget, L., A Subdomain Boundary Element Method for High-Reynolds Laminar Flow Using Stream Function-Vorticity Formulation, *Int J Num Meth Fluids, 46* (2004), 8, pp. 815-847
[9] Paige, C. C., Saunders, M. A., LSQR: Sparse Linear Equations and Least-Squares Problems, *ACM Transactions on Mathematical Software, 8* (1982), 1, pp. 195-209

[10] Ramsak, M., Skerget, L., Iterative Least Squares Methods for Solving over-Determined Matrix for Mixed Boundary Elements, *ZAMM, 80* (2000), Suppl. 3, pp. S657-S658
[11] Celik, I., Numerical Uncertainty in Fluid Flow Calculations: Needs for Future Research, *ASME Journal of Fluids Engineering, 115* (1993), 2, pp. 194-195

Paper submitted: April 4, 2012
Paper revised: June 26, 2012
Paper accepted: June 27, 2012