

# Simulation of flow of nanofluids by BEM

J. Ravnik & L. Škerget

*University of Maribor, Faculty of Mechanical Engineering, Slovenia*

## Abstract

Heat transfer is one of the major engineering problems, which has to be addressed when progress is made in engineering technology. Low thermal conductivity of fluids such as water or oil, led to the introduction of nanofluids – stable suspensions of nanosized particles in base fluid. Nanofluids have very high thermal conductivities at very low nanoparticle concentrations and are able to substantially increase heat transfer.

Numerical solver for simulation of flow and heat transfer of nanofluids was developed using Boundary Element Method. Velocity-vorticity formulation of Navier-Stokes equations is proposed for nanofluids and solved by a combination of single domain and sub-domain boundary element method. The developed solver was used to simulate three-dimensional natural convection of nanofluids. Results of simulations show that a substantial heat transfer enhancement can be observed and attributed to the usage of nanofluids. We observed that the nanofluid particle volume fraction also plays an important role – increasing the volume fraction increases the heat transfer for all Rayleigh number values considered.

*Keywords: nanofluids, boundary element method, natural convection.*

## 1 Introduction

Nanofluid is a suspension consisting of uniformly dispersed and suspended nanometre-sized (10–50 nm) particles in base fluid, pioneered by Choi [1]. Nanofluids are used to replace low thermal conductivity working fluids such as water, oil or ethylene glycol. Nanofluids have very high thermal conductivities at very low nanoparticle concentrations.

Buoyancy induced flow and heat transfer is an important phenomenon used in various engineering systems. Increased thermal conductivity of nanofluids plays a crucial role in the enhancement of heat transfer especially in slow, low Rayleigh



number, flows. In this regime, usage of nanofluids instead of classical working fluids, is most advantageous.

Numerous numerical methods have been proposed to simulate the natural convection phenomena of pure fluids. Our, boundary element based methods, was presented in Ravnik et al. [2]. We employ the velocity-vorticity formulation of Navier-Stokes equations coupled with the energy equation. The unknown field functions are the velocity, vorticity and temperature. We use single domain BEM to solve the kinematics equation for boundary vorticity values (Škerget et al. [3]). Apart for the boundary vorticity values, all other flow fields are solved for by subdomain BEM (Popov et al. [4]). Subdomain BEM solution of a partial differential equation leads to an over-determined sparse system of linear equations. A sparse system enables fast algebraic operations and does not require a lot of storage.

In this work we implemented the velocity vorticity formulation written for nanofluids and derived a BEM based numerical algorithm to find a solution.

## 2 Governing equations

We consider water based nanofluids. Their thermophysical properties are given in Table 1. Effective properties of the nanofluid are: density  $\rho_{nf}$ , dynamic viscosity  $\mu_{nf}$ , heat capacitance  $(c_p)_{nf}$ , thermal expansion coefficient  $\beta_{nf}$  and thermal conductivity  $k_{nf}$ , where subscript  $nf$  is used to denote effective i.e. nanofluid properties. The properties are all assumed constant throughout the flow domain. The non-dimensional velocity-vorticity formulation of Navier-Stokes equations for simulation of nanofluids consists of the kinematics equation, the vorticity transport equation and the energy equation:

$$\nabla^2 \vec{v} + \vec{\nabla} \times \vec{\omega} = 0, \quad (1)$$

$$\frac{\partial \vec{\omega}}{\partial t} + (\vec{v} \cdot \vec{\nabla}) \vec{\omega} = (\vec{\omega} \cdot \vec{\nabla}) \vec{v} + Pr \frac{\mu_{nf}}{\mu_f} \frac{\rho_f}{\rho_{nf}} \nabla^2 \vec{\omega} - Pr Ra \frac{\beta_{nf}}{\beta_f} \vec{\nabla} \times T \vec{g}, \quad (2)$$

Table 1: Thermophysical properties of water based nanofluids (Oztop and Abu-Nada [6]).

	pure water	$Cu$	$Al_2O_3$	$TiO_2$
$c_p [J/kgK]$	4179	385	765	686.2
$\rho [kg/m^3]$	997.1	8933	3970	4250
$k [W/mK]$	0.613	400	40	8.9538
$\beta [\cdot 10^{-5} K^{-1}]$	21	1.67	0.85	0.9
$\alpha [\cdot 10^{-7} m^2/s]$	1.47	1163	131.7	30.7



$$\frac{\partial T}{\partial t} + (\vec{v} \cdot \vec{\nabla})T = \frac{k_{nf}}{k_f} \frac{(\rho c_p)_f}{(\rho c_p)_{nf}} \nabla^2 T. \quad (3)$$

The flow and heat transfer of a nanofluid is defined by specifying the pure fluid Rayleigh  $Ra$  and Prandtl  $Pr$  number values. The nanofluid properties are evaluated using the following models (Khanafar et al. [5]): density from (4), viscosity from (5), thermal capacity from (6), thermal expansion from (7) and thermal conductivity from (8):

$$\rho_{nf} = (1 - \varphi)\rho_f + \varphi\rho_s, \quad (4)$$

$$\mu_{nf} = \frac{\mu_f}{(1 - \varphi)^{2.5}}, \quad (5)$$

$$(\rho c_p)_{nf} = (1 - \varphi)(\rho c_p)_f + \varphi(\rho c_p)_s, \quad (6)$$

$$\beta_{nf} = \beta_f \left[ \frac{1}{1 + \frac{(1-\varphi)\rho_f}{\varphi\rho_s}} \frac{\beta_s}{\beta_f} + \frac{1}{1 + \frac{\varphi}{1-\varphi} \frac{\rho_s}{\rho_f}} \right], \quad (7)$$

$$k_{nf} = k_f \frac{k_s + 2k_f - 2\varphi(k_f - k_s)}{k_s + 2k_f + \varphi(k_f - k_s)}. \quad (8)$$

### 3 Numerical method

The algorithm used to solve the set of governing equations (1)–(3) is devised as follows. Either Dirichlet or Neumann type boundary conditions for velocity and temperature must be known. In this paper we use the no-slip boundary condition on all solid walls and prescribe temperature or temperature flux. Boundary conditions for vorticity are unknown and are calculated as a part of the algorithm. The following steps are performed.

1. Use models (4)–(8) to calculate ratios of nanofluid to pure fluid material properties
2. Calculate vorticity values on the boundary by single domain BEM from the kinematics equation (1).
3. Calculate velocity values by sub-domain BEM from the kinematics equation (1).
4. Calculate temperature values by sub-domain BEM from the energy equation (3).
5. Calculate vorticity values in the domain by sub-domain BEM from the vorticity transport equation (2).
6. Check convergence. If all flow fields converged to  $10^{-6}$  stop, else go to 2.

The three-dimensional solver capable of simulating flow and heat transfer by solving velocity-vorticity formulation of Navier-Stokes equations by a combination of single and sub-domain BEM was developed by Ravnik et al. [2, 7].



The solver has been adapted for simulation of flow and heat transfer of nanofluids. Governing equations are written in integral form. The kinematics equation is

$$c(\vec{\xi})\vec{n}(\vec{\xi}) \times \vec{v}(\vec{\xi}) + \vec{n}(\vec{\xi}) \times \int_{\Gamma} \vec{v}\vec{\nabla}u^* \cdot \vec{n}d\Gamma \\ = \vec{n}(\vec{\xi}) \times \int_{\Gamma} \vec{v} \times (\vec{n} \times \vec{\nabla})u^*d\Gamma + \vec{n}(\vec{\xi}) \times \int_{\Omega} (\vec{\omega} \times \vec{\nabla}u^*)d\Omega, \quad (9)$$

where  $\vec{\xi}$  is the source or collocation point,  $\vec{n}$  is a vector normal to the boundary, pointing out of the domain and  $u^*$  is the fundamental solution for the diffusion operator:  $u^* = 1/4\pi|\vec{\xi} - \vec{r}|$ .  $c(\vec{\xi})$  is the geometric factor defined as  $c(\vec{\xi}) = \alpha/4\pi$ , where  $\alpha$  is the inner angle with origin in  $\vec{\xi}$ . In order to write a linear system of equations for the unknown boundary vorticity values, we set the source point into every boundary node of the whole computational domain. This yields a full system matrix where number of rows and columns is equal to number of boundary nodes. It is solved using a LU decomposition method.

The partial derivative with respect to time in the kinetics equations is approximated by second order three point finite difference scheme  $\partial T/\partial t = (3T - 4T' + T'')/2\Delta t$ , where  $\Delta t$  is the time step and prime and double prime denote field functions in two earlier time steps. The final forms of vorticity transport and energy equation are

$$c(\vec{\xi})\omega_j(\vec{\xi}) + \int_{\Gamma} \omega_j\vec{\nabla}u^* \cdot \vec{n}d\Gamma = \int_{\Gamma} u^*q_jd\Gamma + \frac{1}{Pr} \frac{\mu_f}{\mu_{nf}} \frac{\rho_{nf}}{\rho_f} \cdot \\ \cdot \left( \int_{\Gamma} \vec{n} \cdot \{u^*(\vec{v}\omega_j - \vec{\omega}v_j)\} d\Gamma - \int_{\Omega} (\vec{v}\omega_j - \vec{\omega}v_j) \cdot \vec{\nabla}u^*d\Omega \right) \\ - Ra \frac{\beta_{nf}}{\beta_f} \frac{\mu_f}{\mu_{nf}} \frac{\rho_{nf}}{\rho_f} \left( \int_{\Gamma} (u^*T\vec{g} \times \vec{n})_jd\Gamma + \int_{\Omega} (T\vec{\nabla} \times u^*\vec{g})_jd\Omega \right) \\ + \frac{1}{Pr} \frac{\mu_f}{\mu_{nf}} \frac{\rho_{nf}}{\rho_f} \frac{1}{2\Delta t} \int_{\Omega} (3\omega_j - 4\omega'_j + \omega''_j)u^*d\Omega, \quad (10)$$

$$c(\vec{\xi})T(\vec{\xi}) + \int_{\Gamma} T\vec{\nabla}u^* \cdot \vec{n}d\Gamma = \int_{\Gamma} u^*T_qd\Gamma \\ + \frac{k_f}{k_{nf}} \frac{(\rho c_p)_{nf}}{(\rho c_p)_f} \left( \int_{\Gamma} \vec{n} \cdot \{u^*(\vec{v}T)\} d\Gamma - \int_{\Omega} (\vec{v}T) \cdot \vec{\nabla}u^*d\Omega \right) \\ + \frac{k_f}{k_{nf}} \frac{(\rho c_p)_{nf}}{(\rho c_p)_f} \frac{1}{2\Delta t} \int_{\Omega} (3T - 4T' + T'')u^*d\Omega, \quad (11)$$

where  $\omega_j$  is the  $j^{th}$  component of vorticity.

In the subdomain BEM method we make a mesh of the entire domain  $\Omega$  and name each mesh element a subdomain. Equation (10) is written for each of the

subdomains. In order to obtain a discrete version of (10) we use shape functions to interpolate field functions and flux across the boundary and inside of the subdomain. In this work we used hexahedral subdomains, which enable continuous quadratic interpolation of field functions. On each boundary element we interpolate the flux using discontinuous linear interpolation scheme. By using discontinuous interpolation we avoid flux definition problems in corners and edges. A function, e.g. temperature, is interpolated over a boundary elements as  $T = \sum \varphi_i T_i$ , inside each subdomain as  $T = \sum \Phi_i T_i$ , while flux is interpolated over boundary elements as  $q = \sum \phi_i q_i$ . The following integrals must be calculated:

$$[H] = \int_{\Gamma} \varphi_i \vec{\nabla} u^* \cdot \vec{n} d\Gamma, \quad [G] = \int_{\Gamma} \phi_i u^* d\Gamma, \quad [\vec{A}] = \int_{\Gamma} \varphi_i \vec{n} u^* d\Gamma, \quad (12)$$

$$[B] = \int_{\Omega} \Phi_i u^* d\Omega, \quad [\vec{D}] = \int_{\Omega} \Phi_i \vec{\nabla} u^* d\Omega. \quad (13)$$

The square brackets denote integral matrices. In order to calculate the integrals, a Gaussian quadrature algorithm is used. Calculation of the free coefficient  $c(\vec{\xi})$  is performed indirectly considering rigid body movement problem solution. The calculated  $c(\vec{\xi})$  are added to the diagonal terms of the  $[H]$  matrix.

The source point is set to all function and flux node in each subdomain. By letting curly brackets denote vectors of nodal values of field functions, we may write the discrete vorticity transport equation for  $x$  component as:

$$\begin{aligned} [H]\{\omega_x\} &= [G]\{q_x\} \\ &+ \frac{1}{Pr} \frac{\mu_f}{\mu_{nf}} \frac{\rho_{nf}}{\rho_f} ([A_y]\{v_y \omega_x - \omega_y v_x\} + [A_z]\{v_z \omega_x - \omega_z v_x\}) \\ &- \frac{1}{Pr} \frac{\mu_f}{\mu_{nf}} \frac{\rho_{nf}}{\rho_f} ([D_y]\{v_y \omega_x - \omega_y v_x\} + [D_z]\{v_z \omega_x - \omega_z v_x\}) \\ &+ Ra \frac{\beta_{nf}}{\beta_f} \frac{\mu_f}{\mu_{nf}} \frac{\rho_{nf}}{\rho_f} (g_z [A_y]\{T\} - g_y [A_z]\{T\} - g_z [D_y]\{T\} + g_y [D_z]\{T\}) \\ &+ \frac{1}{Pr} \frac{\mu_f}{\mu_{nf}} \frac{\rho_{nf}}{\rho_f} \frac{1}{2\Delta t} [B](3\{\omega_x\} - 4\{\omega'_x\} + \{\omega''_x\}). \end{aligned} \quad (14)$$

Equations for  $y$  and  $z$  components are obtained analogously. Similarly, the discrete counterpart of the energy equation (11) is:

$$\begin{aligned} [H]\{T\} &= [G]\{q_T\} \\ &+ \frac{k_f}{k_{nf}} \frac{(\rho c_p)_{nf}}{(\rho c_p)_f} ([A_x]\{v_x T\} + [A_y]\{v_y T\} + [A_z]\{v_z T\}) \\ &- \frac{k_f}{k_{nf}} \frac{(\rho c_p)_{nf}}{(\rho c_p)_f} ([D_x]\{v_x T\} + [D_y]\{v_y T\} + [D_z]\{v_z T\}) \\ &+ \frac{k_f}{k_{nf}} \frac{(\rho c_p)_{nf}}{(\rho c_p)_f} \frac{1}{2\Delta t} [B](3\{T\} - 4\{T'\} + \{T''\}), \end{aligned} \quad (15)$$

where  $\{q_T\}$  is a nodal vector of temperature flux.



Since neighbouring subdomains share nodes, the resulting systems of linear equations are over-determined. After taking into account the boundary conditions, we solve them using a least squares solver (Paige and Saunders [8]). All integrals depend only on the shape of subdomains and as such may be calculated only once, prior to the start of the nonlinear iterative process.

### 3.1 Acceleration of computation

The numerical algorithm solves the above equations sequentially within a time step. At the end of each iteration, RMS difference between flow fields of current and previous iteration is calculated. When RMS difference for all flow fields reaches  $\epsilon_{err} = 10^{-6}$ , we stop to iterate. Within each iteration 7 over-determined systems of linear equations must be solved (three for domain velocity, three for domain vorticity and one for temperature). The LSQR solver [8] with diagonal preconditioning is used. The solver requires a large number of iterations in order to converge to a predefined convergence criteria  $\epsilon$ . The original algorithm, as proposed by Ravnik et al. [7], uses a constant convergence criteria. Value, which is 10 times less than the required RMS criteria  $\epsilon_{err}$  was usually used, i.e.  $\epsilon = \epsilon_{err}/10 = 10^{-7}$ .

We argue, that it is not necessary to keep  $\epsilon = 10^{-7}$  during the whole iterative process, since at the beginning, when RMS differences  $\epsilon_{err}$  are large, we do not require a very accurate solution of linear systems of equations. Considering this  $\epsilon$  may set larger. Since the number of iterations of the solver of linear systems of equation depends strongly on the required accuracy and since the most CPU time is used in these routines, we anticipate a large decrease of CPU time.

In order to accelerate the solver, we used the following steps. Instead of keeping  $\epsilon$  constant, we change its value every iteration. We set the range for  $\epsilon$  between  $\epsilon_{min} = 10^{-7}$  and  $\epsilon_{max} = 10^{-3}$ . We introduce a parameter  $R$ ;  $1 \leq R \leq 100$ . At the end of each iteration, we calculate RMS differences  $\epsilon_{err}$  for all flow fields and use the following algorithm to determine the new  $\epsilon$ :

---

DO  $\forall$  equations

$\epsilon_{old} = \epsilon$

IF ( $\epsilon_{err}/R$ .LT. $\epsilon$ ) THEN  $\epsilon = \epsilon_{err}/R$  ELSE  $\epsilon = \epsilon_{max}$

IF ( $\epsilon_{old}$ .LT. $\epsilon$ )  $\epsilon = \epsilon_{old}$

IF ( $\epsilon$ .LT. $\epsilon_{min}$ )  $\epsilon = \epsilon_{min}$

END DO

---

We keep the linear solver accuracy between the minimum and maximum value at  $\epsilon = \epsilon_{err}/R$ . In the algorithm we do not allow for decrease of linear solver accuracy. If at some stage during the nonlinear loop RMS difference increases, the linear solver accuracy is not increased.

In order to test the algorithm, natural convection of air was simulated. Simulations were run for  $Ra = 10^3, 10^4$  and  $10^5$  using parameter  $R$  ranging between  $R = 0.1$  and  $R = 100$ . Table 2 shows the number of iterations of the LSQR solver [8], CPU time and the number of iterations of the nonlinear loop.



Table 2: Comparison of convergence performance and the number of iterations of LSQR solver [8] for different values of parameter  $R$ . All simulations were run until RMS difference reached  $\epsilon_{err} = 10^{-6}$ .

$\epsilon_{min}$	$\epsilon_{max}$	$R$	LSQR nit [ $\cdot 10^3$ ]	CPU time [min]	No. non-linear iterations
<i>Ra</i> = $10^3$					
$10^{-7}$	$10^{-7}$	-	103.5	75	131
$10^{-7}$	$10^{-3}$	100	73.6	55	132
$10^{-7}$	$10^{-3}$	10	57.2	45.7	138
$10^{-7}$	$10^{-3}$	1	37.9	35.5	154
<i>Ra</i> = $10^4$					
$10^{-7}$	$10^{-7}$	-	148.8	108	205
$10^{-7}$	$10^{-3}$	10	79.9	66.3	207
<i>Ra</i> = $10^5$					
$10^{-7}$	$10^{-7}$	-	759.9	570	1235
$10^{-7}$	$10^{-3}$	10	410.6	363	1255
$10^{-7}$	$10^{-3}$	1	202.9	228	1179
$10^{-7}$	$10^{-3}$	0.1	150.6	241	1774

Within each non-linear loop iteration, seven linear systems of equations must be solved using LSQR solver. The table presents cumulative values for number of LSQR iterations.

At  $Ra = 10^3$ , the total number of iterations of the LSQR solver required for the computation to converge at constant  $\epsilon$  drops to more than one half of its value when using newly proposed dynamic solver accuracy algorithm with  $R = 1$ . At the same time, the CPU time of the whole non-linear loop is also decreased by one half. At higher Rayleigh number values even better decrease of CPU time is observed. The gain in CPU time seems to be unaffected by the nonlinearity of the problem, i.e. the Rayleigh number. The number of non-linear iterations needed to reach the solution is also stable. Based on this analysis we decided to use dynamic solver accuracy algorithm and we chose the value  $R = 10$  for all further analyses.

## 4 Test case

We used the developed numerical algorithm to simulate natural convection of nanofluids in a cubic cavity. The cavity is filled with fluid and subjected to a temperature difference on two opposite vertical sides, while the other four are adiabatic. Water ( $Pr = 6.2$ ) and water based nanofluids (Table 1) for several



Rayleigh number values were considered. In our previous work (Ravnik et al. [2]) we examined this problem using air as the working fluid. We established, by comparison with other authors (Tric et al. [9]), that a grid with  $25^3$  nodes was sufficient. In this paper we increased the grid density to  $41^3$  nodes in order to further increase the accuracy of computations. Nodes were concentrated towards the hot and cold walls in such a way that the ratio between the largest and the smallest element length was 7.

In order to compare effectiveness of nanofluids in different simulations, wall heat flux is calculated. Usually, the heat flux  $\dot{Q}$  is expressed in terms of pure fluid thermal conductivity, characteristic flow scales and a non-dimensional Nusselt number, i.e.  $\dot{Q} = k_f L \Delta T \cdot Nu$ . The Nusselt number,  $Nu$ , is defined as the integral of the temperature flux through a wall. For a nanofluid, it is written as

$$Nu = \frac{k_{nf}}{k_f} \int_{\Gamma} \vec{\nabla} T \cdot \vec{n} dA, \quad (16)$$

where  $\Gamma$  is the surface through which we calculate the heat flux and  $\vec{n}$  is a unit normal to this surface. We study local variation of heat flux using the local Nusselt number defined as  $Nu_i(x, y, z) = k_{nf}/k_f \vec{\nabla} T \cdot \vec{n}$ .

#### 4.1 Results

Applying a temperature difference on two opposite walls of an otherwise insulated cavity starts up natural convection producing a large vortex in the main part of the enclosure. At low Rayleigh number values the vortex is weak and the heat is transferred predominately by diffusion. Convection dominates at  $Ra = 10^6$ , where temperature stratification may be observed. The flow becomes unsteady for higher  $Ra$  values with vortices forming along the hot and cold walls. Due to high thermal conductivity of nanofluids we expect to observe the largest improvement in heat transfer of nanofluids for cases where diffusion play a non negligible role. Thus, simulations were performed for Rayleigh number values between  $Ra = 10^3$  and  $Ra = 10^6$  for three nanofluids. Two solid nanoparticle volume fractions in nanofluids were considered:  $\varphi = 0.1$  and  $\varphi = 0.2$ .

Nusselt number values for the natural convection in a cube are shown in Table 3. Using water based nanofluids instead of pure water increases heat transfer in all cases. For low Rayleigh number, where diffusion is the predominant heat transfer mechanism, the enhancement is the largest. For  $Cu$  nanofluid at  $Ra = 10^3$  we observe an 27.2% increase in heat transfer for  $\varphi = 0.1$  and 64.1% for  $\varphi = 0.2$ . Similar findings were reported by Abu-Nada and Oztop [10] for 2D inclined cavity case.

As the Rayleigh number increases, convection becomes the dominant heat transfer mechanism, while diffusion is negligible. Thus, the increased thermal conductivity of nanofluids plays a less important role in the overall heat balance. All nanofluids exhibit smaller heat transfer enhancement as compared to the low Rayleigh number case. At  $Ra = 10^6$   $Cu$  nanofluid increases heat transfer at  $\varphi = 0.1$  for 11.6% and at  $\varphi = 0.2$  for 21.6%.





Table 3: Nusselt number values for the natural convection in a cube. Solid particle volume fraction is denoted by  $\varphi$ .

$Ra$	Water	Water+Cu		Water+Al <sub>2</sub> O <sub>3</sub>		Water+TiO <sub>2</sub>	
	pure	$\varphi = 0.1$	$\varphi = 0.2$	$\varphi = 0.1$	$\varphi = 0.2$	$\varphi = 0.1$	$\varphi = 0.2$
$10^3$	1.071	1.363	1.758	1.345	1.718	1.297	1.598
$10^4$	2.078	2.237	2.381	2.168	2.244	2.115	2.132
$10^5$	4.510	4.946	5.278	4.806	4.968	4.684	4.732
$10^6$	9.032	10.08	10.98	9.817	10.39	9.556	9.871

In our geometry the hot and cold walls face each other in  $x$  direction. Thus, the main vortex, which is induced by the onset of natural convection is located in the  $y$  plane. We chose the  $y = 0.5H$  plane to study the two velocity profiles: vertical velocity  $v_z(x, 0.5H, 0.5H)$  and horizontal velocity  $v_x(0.5H, 0.5H, z)$  across the centre of the enclosure. The comparison of profiles for  $Ra = 10^3$  and  $Ra = 10^6$  for water and nanofluids is shown in Figure 1.

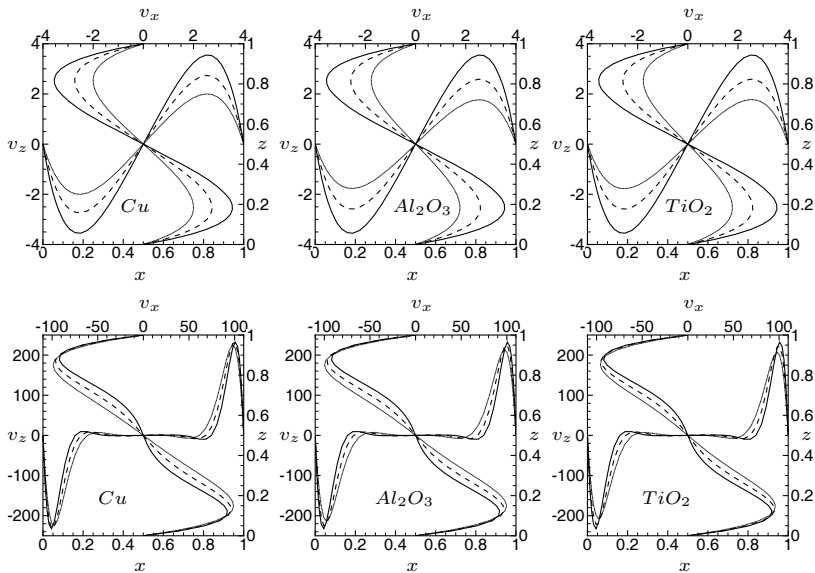


Figure 1: Velocity profiles  $v_x(z)$  and  $v_z(x)$  through a centre of the  $y = 0.5H$  plane for natural convection in a differentially heated cubic cavity. Top row  $Ra = 10^3$ , bottom row  $Ra = 10^6$ . Solid line denotes pure water, dashed line  $\varphi = 0.1$  nanofluid and dotted line  $\varphi = 0.2$  nanofluid.

When diffusion dominates ( $Ra = 10^3$ ) we observe that pure water reaches the highest velocities, while addition of solid particles slows down the flow. The decreased velocity results in decreased convective heat transfer. However, since in this regime the majority of heat is transported by diffusion, the decrease due to lower velocity is almost negligible and the overall heat transfer of nanofluids is still very large due to higher thermal conductivity of a nanofluid. In the  $Ra = 10^6$  case convection dominates. Here we observe that velocities reached by nanofluids are higher than the velocities of pure water. Thus, using nanofluids, the velocity profiles, and in consequence, temperature profiles and heat transfer are increased. The relative increase of heat transfer in the convection dominated case is smaller than the increase in the diffusion dominated case because the increased thermal conductivity does not play an important role in the convection dominated heat transfer. When comparing velocity profiles between different nanofluids we observe only slight differences. The  $Cu$  nanofluid reaches the highest velocities, while the highest velocities for  $Al_2O_3$  nanofluid are about 4% lower and for  $TiO_2$  approximately 9% lower.

Figure 2 displays temperature contours in the central  $y = 0.5H$  plane. Comparing the temperature fields for different nanofluids we observe almost identical temperature distribution in the central part of the enclosure. Differences are larger closer to the walls, although their magnitude is still small. Since heat transfer depends on the temperature gradient on the walls, we examined the temperature contour closest to the hot and cold walls.

Heat flux distribution on a profile across the hot wall is shown in Figure 3. The heat flux is smaller at the top of the hot wall and larger at the bottom, since the hot fluid is rising and decreasing the temperature gradient at the top. Comparing the  $Ra = 10^3$  and  $Ra = 10^6$  cases, we see that the heat transfer enhancement is, relatively speaking, larger in the diffusion dominated  $Ra = 10^3$  case than the convection dominated  $Ra = 10^6$  case.

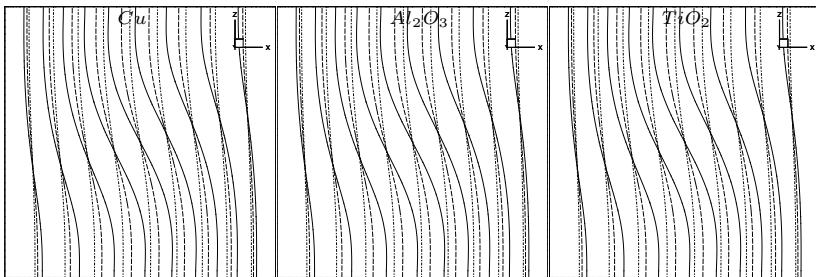


Figure 2: Temperature contours on the central  $y = 0.5H$  plane. Contour values are  $-0.4(0.1)0.4$ ;  $Ra = 10^3$ . Solid line denotes pure water, dashed line  $\varphi = 0.1$  nanofluid and dotted line  $\varphi = 0.2$  nanofluid.

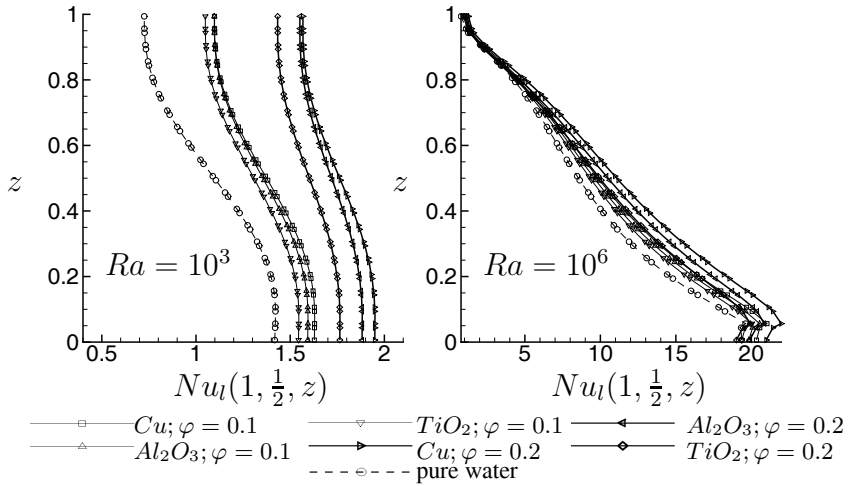


Figure 3: Heat flux shown in terms of local Nusselt number on a profile across hot wall on the  $y = 0.5H$  plane.

## 5 Summary

The paper presents a numerical method for the simulation of flow of nanofluids. The method is based on the BEM solution of the velocity vorticity formulation of Navier-Stokes equations. A combination of single domain and sub-domain BEM was used. A dynamic solver accuracy algorithm was implemented to speed up simulations.

The developed algorithm was used to simulate natural convection in a three-dimensional differentially heated cavity of three types of water based nanofluids. Heat transfer, expressed with Nusselt number values, was presented for water and three nanofluids ( $Cu$ ,  $Al_2O_3$  and  $TiO_2$ ). Results show that using water based nanofluids instead of pure water enhances heat transfer and that the enhancement is largest when diffusion is the dominant heat transfer mechanism. Simulations also revealed that heat transfer enhancement grows with increasing solid nano-particle volume fraction in the nanofluid.

## References

- [1] Choi, S.U.S., Enhancing thermal conductivity of fluids with nanoparticles. *Develop Appl Non Newtonian Flows*, **66**, pp. 99–106, 1995.
- [2] Ravnik, J., Škerget, L. & Žunič, Z., Velocity-vorticity formulation for 3D natural convection in an inclined enclosure by BEM. *Int J Heat Mass Transfer*, **51**, pp. 4517–4527, 2008.
- [3] Škerget, L., Hriberšek, M. & Žunič, Z., Natural convection flows in complex



- cavities by BEM. *Int J Num Meth Heat & Fluid Fl*, **13**, pp. 720–735, 2003.
- [4] Popov, V., Power, H. & Škerget, L., (eds.) *Domain Decomposition Techniques for Boundary Elements: Applications to fluid flow*. WIT press, 2007.
- [5] Khanafer, K., Vafai, K. & Lightstone, M., Buoyancy-driven heat transfer enhancement in a two-dimensional enclosure utilizing nanofluids. *Int J Heat Mass Transfer*, **46**, pp. 3639–3653, 2003.
- [6] Oztop, H.F. & Abu-Nada, E., Natural convection of water-based nanofluids in an inclined enclosure with a heat source. *Int J Heat Fluid Flow*, **29**, pp. 1326–1336, 2008.
- [7] Ravnik, J., Škerget, L. & Žunič, Z., Combined single domain and subdomain BEM for 3D laminar viscous flow. *Eng Anal Bound Elem*, **33**, pp. 420–424, 2009.
- [8] Paige, C.C. & Saunders, M.A., LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, **8**, pp. 43–71, 1982.
- [9] Tric, E., Labrosse, G. & Betrouni, M., A first incursion into the 3D structure of natural convection of air in a differentially heated cubic cavity, from accurate numerical simulations. *Int J Heat Mass Transfer*, **43**, pp. 4034–4056, 2000.
- [10] Abu-Nada, E. & Oztop, H.F., Effects of inclination angle on natural convection in enclosures filled with cuwater nanofluid. *Int J Heat Fluid Fl*, **30**, pp. 669–678, 2009.

