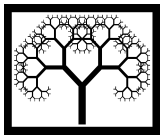
Paper 9



Proceedings of the Eighth International Conference on Engineering Computational Technology, B.H.V. Topping, (Editor), Civil-Comp Press, Stirlingshire, Scotland

A Parallel Meshless Numerical Approach for the Solution of Transport Phenomena

G. Kosec and R. Trobec
Department of Communication Systems
Jožef Stefan Institute, Ljubljana, Slovenia

Abstract

The application of the local meshless numerical method (LRBFCM) for solving a system of coupled partial differential equations (PDE) is explored. The numerical approach is tested on the natural convection based fluid flow problems. The fluid flow part of the solution procedure is coupled locally despite its global nature. Such an approach makes the computations convenient for an implementation on parallel computers. In this paper, the OpenMP based parallelization of the proposed numerical approach is demonstrated. On two cores, a superlinear speedup of 2.5 is confirmed by the performance analysis. The parallelization performance is explored for the classical de Vahl Davis natural convection case. The usability of the meshless numerical framework is demonstrated on highly non-linear and coupled case of solidification of binary alloy, where energy and solute transport govern double natural convection in a domain filled with porous media and free fluid with moving interphases.

Keywords: meshfree, LRBFCM, convective-diffusive problems, fluid flow, de Vahl Davis, natural convection, porous, solidification.

1 Introduction

The computational modelling has been, for the last few decades, drawing much attention to researches due to its pronounced contribution in better understanding of the nature as well as in the development of advanced technologies. The modelling of more and more complex transport physical systems helps the community to address important issues like identifying environmental problems, improving technological processes, developing biomedical applications, etc. However, the physical modelling is only one part of the entire problem. In the majority of cases even the simplest useful physical models cannot be solved in a closed form. The development of efficient, simple and robust numerical methods for solving governing systems of

partial differential equations (PDEs) is therefore a major interest in applied sciences and engineering.

The classical numerical methods, such as the Finite Element Method (FEM) and Finite Difference Methods (FDM), are commonly used for solving physical models. Despite the powerful features of traditional methods, there are substantial difficulties in their application to realistic, geometrically complex transient problems. A promising alternative is a class of numerical methods, referred to as the meshless methods [1, 2], where an arbitrarily distributed set of points is used for the discretization, instead of fully structured mesh. Such methods enable techniques that do not need any polygonisation of the domain. Among the simplest and most promising meshless methods is the Local Radial Basis Function Collocation Methods, which can be regarded as a generalization of FDM..

Computational time is an important factor in numerical simulations and it is often not addressed adequately. Parallel computers can compensate for the lack of single computer performance, but only in cases where an efficient parallelization method is known. Parallel FEM algorithms for solving PDEs have been implemented on eight parallel computers with efficiency of 0.85 [3]. The parallelization of a direct solver within the EFG method has been published in [4] with measured speedup and efficiency, 7.1 and 0.89, respectively, for eight processors and systems with up to 2000 equations. Large sparse linear systems obtained by Free Mesh Method, which is a virtually meshless method based on FEM [5], have been parallelized on up to 64 processors [6]. Construction of the linear system was ideally parallelizable on smaller number of processors, while the parallel efficiency for solving the system on all available processors was 0.6. The parallel RKPM in 3-D was implemented with efficiency of 0.6 on 64 processors [7]. The listed examples indicate that several demanding numerical applications need parallel implementations of meshless methods. Present paper tackles the OpenMP based parallelization of a local meshless solution procedure for fluid flow problems.

In this paper we focus on a simple but effective Local Radial Basis Function Collocation Method (LRBFCM) [8]. Besides local discretization scheme the local solution procedure for a pressure velocity coupling is considered [9]. The local solution procedure is needed in order to achieve optimal parallel efficiency with minimal communication as the global approach requires complex parallelization code and a significant amount of interprocessor communication [10]. We demonstrate the efficiency of the OpenMP based local parallel implementation on the classical de Vahl Davis benchmark test case. The results show superlinear regimes in computational speedup, which are further investigated.

2 Test case

The classical de Vahl Davis [11] 2-D natural convection problem (NC) is considered for benchmarking purposes. The problem domain is a closed air-filled square-shaped cavity with differentially heated vertical walls with temperature difference ΔT and

insulated horizontal walls. The non-permeable and no-slip velocity boundaries are assumed. There are several numerical solutions published in the literature [12-15] that make the test convenient for benchmarking purposes. The NC problem is described by three coupled PDE equations: mass (1), momentum (2) and energy conservation (3) equations, where all material properties are considered to be constant. The Boussinesq approximation (4) is used for the treatment of body force in the momentum equation. The natural convection is thus described by the following system of equations

$$\nabla \cdot \mathbf{v} = 0 \tag{1}$$

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \nabla \cdot (\mathbf{v}\mathbf{v}) = -\nabla P + \nabla \cdot \left(\mu \nabla \mathbf{v}\right) + \mathbf{b} \tag{2}$$

$$\rho \frac{\partial \left(c_p T\right)}{\partial t} + \rho \nabla \cdot (c_p T \mathbf{v}) = \nabla \cdot \left(\lambda \nabla T\right) \tag{3}$$

$$\mathbf{b} = \rho\left[1 - \beta_T (T - T_{\text{ref}})\right]\mathbf{g} \tag{4}$$

with $\mathbf{v}, P, T, \lambda, c_p, \mathbf{g}, \rho, \beta_B, T_{\text{ref}}, \mu$ and $\mathbf{b}$ standing for velocity, pressure, temperature, thermal conductivity, specific heat, gravitational acceleration, density, coefficient of thermal expansion, reference temperature for Boussinesq approximation, viscosity and body force, respectively. The case is characterized by two dimensionless values

$$\mathrm{Ra}_T = \frac{|\mathbf{g}|\,\beta_T \Delta T \Omega_H{}^3 \rho^2 c_p}{\lambda \mu}, \tag{5}$$

$$\mathrm{Pr} = \frac{\mu c_p}{\lambda}, \tag{6}$$

referred as Rayleigh and Prandtl numbers. Ω_H stands for domain height (Figure 1).

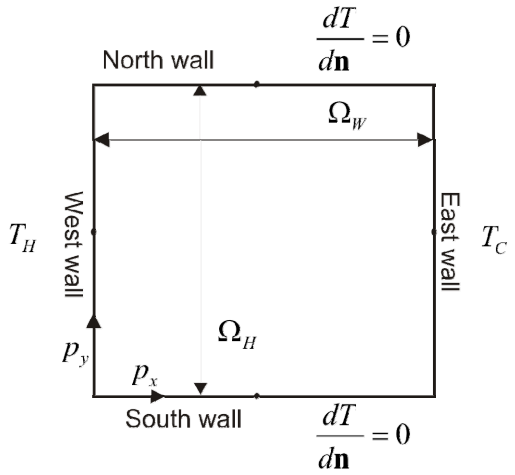


Figure 1: The natural convection benchmark test schematics.

3 Solution procedure

In this work we use a novel local meshless numerical method with local pressure velocity coupling. A general idea behind the local meshless methods is the use of local sub clusters of domain nodes, named local support domains.With the selected support domain, an approximation function is introduced as a linear combination of weighted basis functions

$$\theta(\mathbf{p}) = \sum_{n=1}^{N} \alpha_n \Psi_n(\mathbf{p}) \tag{7}$$

where θ, N, α_n and $\Psi_n, \mathbf{p}$ stand for the approximation function, the number of basis functions, the approximation coefficients and the basis functions, position vector, respectively. In each node such approximation function is created. The basis functions could be selected arbitrarily, e.g., monomials, radial basis function, etc., however in this work Hardy's Multiquadrics (MQs)

$$\Psi_n(\mathbf{p}) = \sqrt{(\mathbf{p}-\mathbf{p}^n)\cdot(\mathbf{p}-\mathbf{p}^n)/\sigma_C^2 + 1} \tag{8}$$

with σ_C standing for the free shape parameter of the basis function, are used, based on the results of the study by Franke [16]. By taking into account all support domain nodes and equation (7) the approximation linear system of equations is obtained. In this work we use the collocation approach, where the number of support nodes is the same as the number of the basis functions, in present work five nodded support domains are used. The system solution provides us with the unknown interpolation coefficients. An arbitrary spatial differential operation (L) can be applied on the interpolating function in the following way

$$L\theta(\mathbf{p}) = \sum_{n=1}^{N_{Basis}} \alpha_n L\Psi_n(\mathbf{p}) \tag{9}$$

In general, the system (7) has to be solved only if the position of the support domain nodes is changed. As a consequence the computation can be optimized by computing $\mathbf{\Psi}^{-1}$ in a pre-process. Furthermore, the computation of the coefficients and the evaluation of differential operators can be combined in a single operation. The differential operator (χ_m^L) vector is thus introduced as a

$$\chi_m^L(\mathbf{p}) = \sum_{n=1}^{N} \Psi_{nm}^{-1} L\left(\Psi_n(\mathbf{p})\right) \tag{10}$$

The introduced formalism holds in general and therefore the general notation for partial differential operator (L) is used. However, in the present work, only operators $\partial/\partial p_\varepsilon$ and ∇^2 are employed.

$$\chi_m^{\nabla^2}(\mathbf{p}) = \sum_{n=1}^{N} \Psi_{nm}^{-1} \sum_{\varepsilon} \frac{\partial^2}{\partial p_\varepsilon^2} \Psi_n(\mathbf{p}) \tag{11}$$

$$\chi_m^{\partial/\partial p_\varepsilon}(\mathbf{p}) = \sum_{n=1}^{N} \Psi_{nm}^{-1} \frac{\partial}{\partial p_\varepsilon} \Psi_n(\mathbf{p}) \tag{12}$$

The structured formulation is convenient since most of the complex and CPU demanding operations are performed in the pre-process phase. For all inner temporal loop operations only N floating point operations (FLOPS) are need for the evaluation of an arbitrary partial differential operator. The implementation of the Dirichlet boundary condition is straightforward by setting the field values in boundary nodes to the boundary conditions. In order to implement Neumann and Robin boundary conditions a special case of interpolation is needed. In these boundary nodes the function directional derivative instead of the function value is known and therefore the equation in the interpolation system changes to

$$\theta_{BC} = \sum_{n=1}^{N_{Basis}} \alpha_n \left(a \frac{\partial}{\partial \mathbf{n}} \Psi_n(\mathbf{p}) + b \Psi_n(\mathbf{p}) \right) \tag{13}$$

In the presented numerical framework the computation of Neumann and Robin boundary conditions can be simplified through the usage of the differential operator vector. Consider the Neumann boundary condition

$$a \frac{\partial \theta}{\partial \mathbf{n}} + b\theta = \theta_{BC} \tag{14}$$

$$\theta_0 = \frac{\theta_{BC} - a \sum_{m=2}^{N_{Sub}} \chi_m^{\partial/\partial \mathbf{n}} \theta_m}{a \chi_0^{\partial/\partial \mathbf{n}} + b} \tag{15}$$

where θ_0 stands for boundary node. Equation (15) simplifies to Neumann boundary condition computation if b is set to zero. Such an approach makes the Neumann and the Robin boundary condition computation straightforward and CPU effective, again only N FLOPS are needed to evaluate them, without any special computational treatment on, or near, boundaries.

For the temporal discretization we use a two-level explicit time stepping

$$\rho_0 \frac{\theta - \theta_0}{\Delta t} = \nabla \cdot \left(D_0 \nabla \theta_0 \right) - \nabla \cdot \left(\rho_0 \mathbf{v}_0 \theta_0 \right) + S_0 \tag{16}$$

where zero-indexed quantities stand for the values at the initial time, and D, S for general diffusion coefficient, and source term, respectively. The time step is denoted with Δt. The pressure-velocity coupling is performed through the correction of the intermediate velocity $(\hat{\mathbf{v}})$

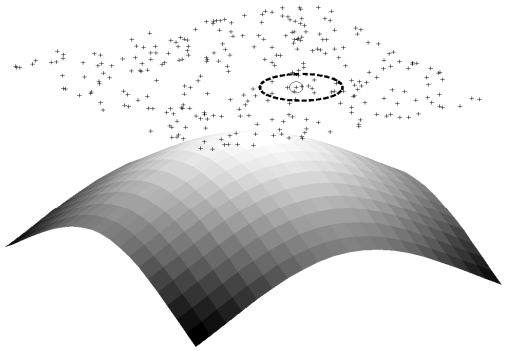


Figure 2: Schematic representation of meshless numerical principle. A domain node (small circle) is considered only through the local influence domain (dashed circle).

$$\hat{\mathbf{v}} = \mathbf{v}_0 + \frac{\Delta t}{\rho}\left(-\nabla P_0 + \nabla\cdot\left(\mu\nabla\mathbf{v}_0\right) + \mathbf{b}_0 - \nabla\cdot(\rho\mathbf{v}_0\mathbf{v}_0)\right) \tag{17}$$

The equation (17) did not take into account the mass continuity and two respective corrections need to be applied

$$\hat{\mathbf{v}}^{m+1} = \hat{\mathbf{v}}^m + \hat{\mathbf{v}} \tag{18}$$

$$\hat{P}^{m+1} = \hat{P}^m + \hat{P} \tag{19}$$

where m, $\hat{v}$ and $\hat{P}$ stand for iteration index, velocity correction and pressure correction, respectively. By combining the momentum and the mass continuity equations, the pressure correction Poisson equation emerges

$$\nabla\hat{\mathbf{v}}^m = \frac{\Delta t}{\rho}\nabla^2\hat{P} \tag{20}$$

Instead of solving the global Poisson equation exactly, the pressure correction is guessed from the divergence of the intermediate velocity.

$$\hat{P} = \ell^2\frac{\rho}{\Delta t}\nabla\cdot\hat{\mathbf{v}}^m \tag{21}$$

The proposed assumption makes the solution, of the pressure-velocity coupling iteration, local. Such an approach is CPU efficient, as it needs only a single computation for each pressure correction. With the computed pressure correction the pressure and the velocity can be corrected as

$$\hat{P}^{m+1} = \hat{P}^{m} + \zeta \widehat{P} \tag{22}$$

where ζ stands for the relaxation parameter. The iteration is performed until the criterion $\nabla \cdot \hat{\mathbf{v}} < \varepsilon_V$ is met in all computational nodes. The approach is similar to the artificial compressibility method (ACM), which has been recently under intense research [17-19] and in the framework of the FDM to the SOLA approach [20]. However, the proposed approach retains the correct time transient which is not the case in the SOLA and the ACM approaches. In the present work we are particularly interested in the proper transient response of computations.

4 Implementation

The Open Multi-Processing (OpenMP) is an Application Programming Interface (API) that supports multi-platform shared-memory multiprocessing programming in C, C++, and Fortran, on most processor architectures and operating systems, including Unix, Mac OS, Microsoft Windows and others [21]. The OpenMP consists of a set of compiler directives, library routines, and environment variables that influence run-time behaviour. In the OpenMP implementation the master thread forks off a specified number of slave threads and divides the task among them. The threads run concurrently on the shared data, with the runtime environment allocating threads to the available processors. Each section of the code that is designed to run in parallel is marked with a pre-processor directive. After the execution of the parallelized code, the slave threads join back into the master thread, and the program executes serially from there on. The OpenMP can implement the task parallelism and the data parallelism at the same time. It runs particularly effective on multicore computer architectures based on the multilevel memory hierarchy and fast shared-memory caches. For the purpose of the present analysis we use *#pragma omp parallel for* directive with the static scheduling. The implementation is shown schematically in Figure 3.

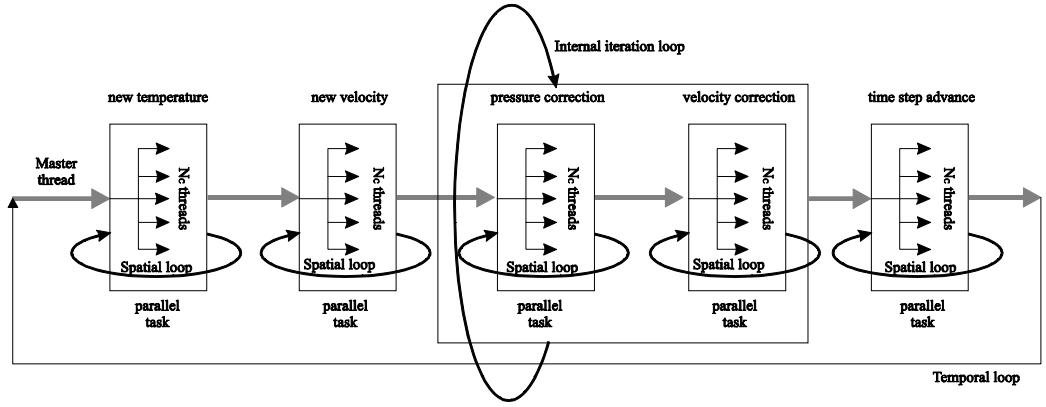


Figure 3: Scheme of parallel solution procedure for the natural convection case.

We run tests on a computer system with four Intel Xeon E7450 processors, each with six cores, system clock of 2.40 GHz, 1066 MHz front side bus (FSB), and 64 GB of shared main memory. The system has three levels of cache hierarchy: each core has 32 kB of L1 instruction cache and 32 kB of L1 data cache. Typical L1 latency is 2 clock cycles. Each pair of cores shares 3 MB of L2 cache, for a total of 9 MB of shared L2 cache per processor. Typical L2 latency in the case of L2 cache hit is 6 clock cycles. Each processor has 12 MB of shared L3 cache. Typical L3 latency in the case of L3 cache hit is 60 cycles or 120 clock cycles in the case of L3 cache miss and main memory hit.

In addition, we create a preliminary cache miss analysis on a computer system with two quad-core Intel Xeon 5520 2.26 GHz processors, 1066 MHz FSB, and 24 GB of shared main memory. Each core has 32 kB L1 instruction cache, 32 kB L1 data cache, and 256 kB of L2 cache. Each CPU has 8 Mb of L3 cache that is shared among its four cores. Cache latencies, in the case of cache hit, are 4 clock cycles for L1 data cache, 10 cycles for L2 cache, and 40 cycles for L3 cache.

5 Results

The results of numerical integration for the NC case have been already explored in details in our previous work [22] are shown in the left part of Figure 4 with cavity streamlines and temperature contours of the stationary state solution for Ra=10^8. The presented solution procedure has been also successfully applied on several other thermo-fluid problems, where the latest and also the most complex simulation was solidification of a binary alloy [23]. In addition to already performed numerical methodology assessment [9] we analyse the parallel efficiency of the proposed solution approach. The performance is evaluated through the speedup of computation defined as

$$S = \frac{t_1}{t_{Nc}} \tag{23}$$

With t_1 and t_{N_c} standing for the computational time on a single core and on N_C cores. In the right part of Figure 4 the speedup is shown for the different number of domain nodes N_D. On the left part of Figure 4 streamlines and temperature contour plots for considered case are presented.

The fluid flow computations show superlinear speedup when working with high number of N_D. As expected, with increasing number of cores the superlinear effect is lost. To explain the range of superlinear regime we focus on the computations with one and two cores $\left(N_C = 1,2\right)$. Besides the computational time we measure L3 cache hit rate defined as

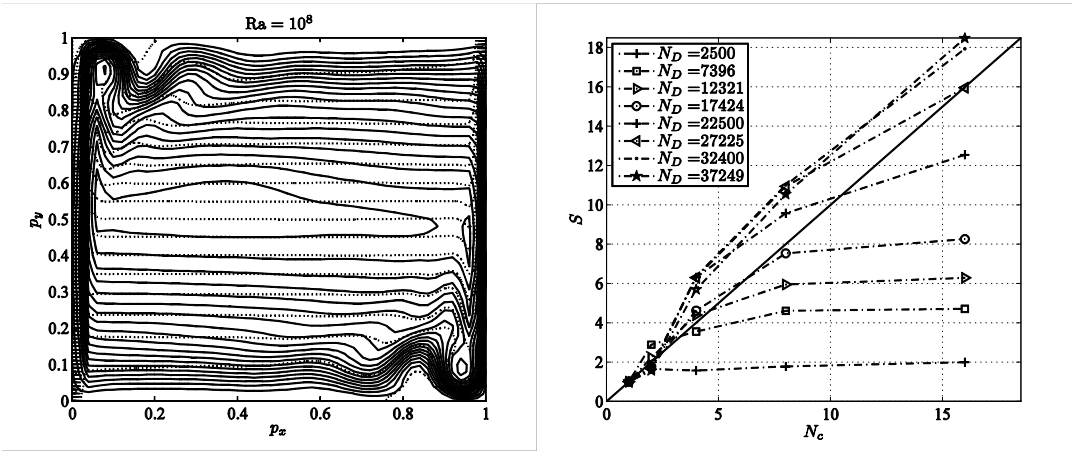


Figure 4: Cavity streamlines and temperature contours for Ra=10^8 (left) and speedup with respect to N_D and N_C (right).

$$H_{N_C} = \frac{number\ of\ hits}{number\ of\ all\ accesess}. \tag{24}$$

where the subscript denotes the number of cores engaged in the computation.. The CPU performance counters that record cache hit and miss events were accessed via Intel Performance Counter Monitor [24].

In Figure 5 the computation time and the speedup with the respect to N_D for $N_C = \{1,2\}$ is shown together with the cache hit rate measurements. It is clearly visible that the superlinear speedup of almost 2.5 occurs when the L3 cache hit rate starts to decline on the single core while it stays slightly above 0.9 on two cores. In this regime, an increased N_D causes the working data set to become too large for the L3 cache of single core while it still fits into two L3 caches that are available by two core system. The single-threaded program slows down comparing to the two-threaded program as a result of an increased number of L3 cache accesses. Roughly after $N_D = 15000$ the working data set becomes too large even for both L3 caches, therefore the main memory must be accesses regardless of the number of threads. From that point on the speedup cannot be superlinear on the account of accumulating L3 caches. Note that in Figure 5 dimensionless computation times are used, as the absolute values do not supply any additional information, yet the dimensionless values are more convenient for presentation. The speedup values are not scaled in any way.

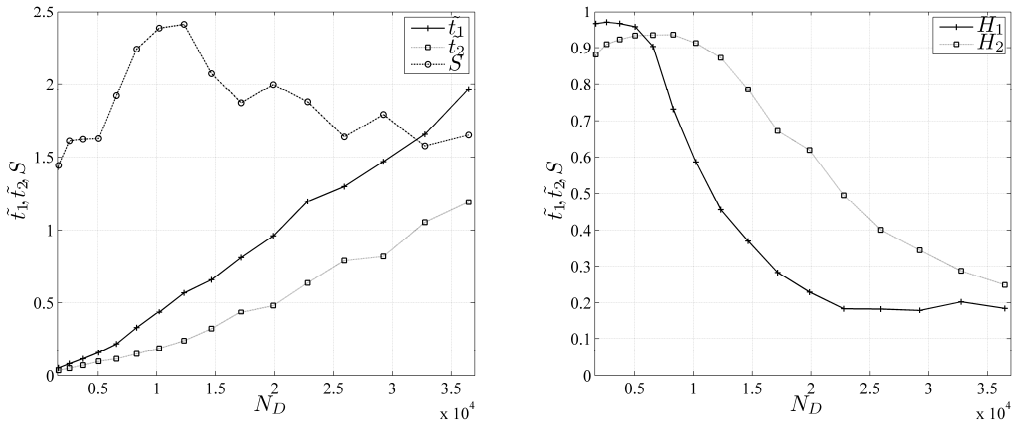


Figure 5: Computation time and speedup (left) and corresponding L3 hit rate (right).

6 Conclusions

We have proposed and implemented an OpenMP based parallelization of a local meshless numerical solution procedure. The fluid mechanics problem is considered as a test problem. To construct an effective parallelization, we introduce a local pressure correction algorithm, which requires none of global communication. The classical de Vahl Davis benchmark test at Ra=10^8 is computed by an OpenMP parallel implementation. A superlinear speedup is captured on the performance analysis. Through the analysis of computational problem the background of superlinearity is explained. The accumulating L3 caches are identified as a source of such behaviour.

Our results stand for a preliminary analysis as the analysis lacks of detailed quantitative assessment. Our future work will be focused in testing of more different computing architectures based on many/multi core interconnected processors.

Acknowledgment

The authors acknowledge the financial support from the state budget by the Slovenian Research Agency under Grant P2-0095.

References

[1] Kosec, G.; Šarler, B., "Numerical solution of natural convection problems by a meshless method", Convection and Conduction Heat Transfer, 108-132, 2011.

[2] Atluri, S. N. and Shen, S., "The Meshless Method", Tech Science Press, Encino, 2002.
[3] Nikishkov, G. P. , Makinouchi, A., Yagawa, G. and S., Yoshimura., "Performance study of the domain decomposition method with direct equation solver for parallel finite element analysis", Computational Mechanics, vol. 19, pp. 84-93, 1996.
[4] Singh, V., "Parallel implementation of the EFG Method for heat transfer and fluid flow problems", Computational Mechanics, vol. 34, pp. 453–463, 2004.
[5] Yagawa, G. and Yamada, T., "Free mesh method: A new meshless method", Computational Mechanics, vol. 18, pp. 383-386, 1996.
[6] Shirazaki, M. and Yagawa, G., "Large-scale parallel flow analysis based on free mesh method: a virtually meshless method", Computer Methods in Applied Mechanics and Engineering, vol. 174, pp. 419-431, 1999.
[7] Zhang, L., Wagner, G. J. and Liu, W. K., "A parallelized meshfree method with boundary enrichment for large-scale CFD", Journal of Computational Physics, vol. 176, pp. 483–506, 2002.
[8] Šarler, B., "From global to local radial basis function collocation method for transport phenomena", Advances in Meshfree Techniques, 257-282, 2007.
[9] Kosec, G. and Šarler, B., "Solution of thermo-fluid problems by collocation with local pressure correction", International Journal of Numerical Methods for Heat and Fluid Flow, vol. 18, pp. 868-882, 2008.
[10] Trobec, R., Sterk, M. and Robic, B., "Computational complexity and parallelization of the meshless local Petrov-Galerkin method", Computers & Structures, vol. 87, pp. 81-90, 2009.
[11] de Vahl Davis, G., "Natural convection of air in a square cavity: a bench mark numerical solution", International Journal of Numerical Methods in Fluids, vol. 3, pp. 249-264, 1983.
[12] Prax, C., Sadat, H. and Salagnac, P., "Diffuse approximation method for solving natural convection in porous Media", Transport in Porous Media, vol. 22, pp. 215-223, 1996.
[13] Sadat, H. and Couturier, S., "Performance and accuracy of a meshless method for laminar natural convection", Numerical Heat Transfer, vol. B37, pp. 455-467, 2000.
[14] Divo, E. and Kassab, A. J., "Localized meshless modeling of natural-convective viscous flows", Numerical Heat Transfer, vol. B129, pp. 486-509, 2007.
[15] Wan, D. C., Patnaik, B. S. V. and Wei, G. W., "A new benchmark quality solution for the buoyancy-driven cavity by discrete singular convolution", Numerical Heat Transfer, vol. B40, pp. 199-228, 2001.
[16] Franke, J., "Scattered data interpolation: tests of some methods", Mathematics of Computation, vol. 48, pp. 181-200, 1982.
[17] Rahman, M. M. and Siikonen, T., "An artificial compressibility method for viscous incompressible and low Mach number flows", Internation Journal of Numerical Methods in Engineering, vol. 75, pp. 1320-1340, 2008.
[18] Malan, A. G. and Lewis, R. W., "An artificial compressibility CBS method for modelling heat transfer and fluid flow in heterogeneous porous materials",

International Journal for Numerical Methods in Engineering, vol. 87, pp. 412-423, 2011.
[19] Traivivatana, S., Boonmarlert, P., Thee, P., Phongthanapanich, S. and Dechaumphai, P., "Combined adaptive meshing technique and characteristic-based split algorithm for viscous incompressible flow analysis", Applied Mathematics and Mechanics, vol. 28, pp. 1163-1172, 2007.
[20] Hong, C. P., "Computer Modelling of Heat and Fluid Flow Materials Processing", Institute of Physics Publishing Bristol, 2004.
[21] Chandra, R, Dagum, L., Kohr, D., Maydan, D., McDonald, J. and Menon, R., "Parallel Programming in OpenMP", Academic Press, San Diego, 2001.
[22] Kosec, G. and Šarler, B., "Solution of heat transfer and fluid flow problems by the simplified explicit local radial basis functon collocation method", 14th international Conference on FE in Flow Problems, 2007.
[23] Kosec, G., Založnik, M., Šarler, B. and Combeau, H., "A meshless approach towards solution of macrosegregation phenomena", Computers, Materials and Continua, vol. 22, pp. 169-195, 2011.
[24] INTEL, http://software.intel.com/en-us/articles/ intel-performance-counter-monitor/.