Uludağ University Journal of The Faculty of Engineering, Vol. 29, No. 1, 2024 RESEARCH

DOI: 10.17482/uumfd.1325198

PARALLEL MESHLESS RADIAL BASIS FUNCTION COLLOCATION METHOD FOR NEUTRON DIFFUSION PROBLEMS

Tayfun TANBAY [*]

Received: 10.07.2023; revised: 22.02.2024; accepted: 27.02.2024

Abstract: The meshless global radial basis function (RBF) collocation method is widely used to model physical phenomena in science and engineering. The method produces highly accurate solutions with an exponential convergence rate. However, due to the global approximation structure of the method, dense node distributions lead to long computation times and hinder the applicability of the technique. In order to overcome this issue, this study proposes a parallel meshless global RBF collocation algorithm. The algorithm is applied to 2-D neutron diffusion problems. The multiquadric is used as the RBF. The algorithm is developed with Mathematica and eight virtual processors are used in calculations on a multicore computer with four physical cores. The method provides accurate numerical results in a stable manner. Parallel speedup increases with the number of processors up to five and seven processors for external and fission source problems, respectively. The speedup values are limited by the constrained resource sharing of the multicore computer's memory. On the other hand, significant time savings are achieved with parallel computation. For the four-group fission source problem, when 4316 interpolation nodes are employed, the utilization of seven processors instead of sequential computation decreases the computation time of the meshless approach by 716 s.

Keywords: Parallel, Meshless, Radial basis function, Collocation, Neutron diffusion

Nötron difüzyon problemleri için paralel ağsız radyal baz fonksiyonu kollokasyon yöntemi

Öz: Ağsız global radyal baz fonksiyonu (RBF) kollokasyon yöntemi bilim ve mühendislikte karşılaşılan fiziksel olayların modellenmesinde yaygın bir şekilde kullanılmaktadır. Yöntem, üstel bir yakınsama hızı ile yüksek doğruluğa sahip çözümler üretir. Fakat, yöntemin global yaklaşım yapısı nedeniyle, çok sayıda ayrıklaştırma noktası kullanılması hesaplama sürelerini uzatmakta ve yöntemin uygulanabilirliğini kısıtlamaktadır. Söz konusu sorunun üstesinden gelebilmek için bu çalışmada bir paralel ağsız global RBF kollokasyon algoritması önerilmiştir. Algoritma iki boyutlu nötron difüzyon problemlerine uygulanmıştır. Multikuadrik fonksiyonu RBF olarak kullanılmıştır. Algoritma, Mathematica yazılımı ile geliştirilmiş ve hesaplamalar dört fiziksel çekirdeğe sahip çok çekirdekli bir bilgisayar ile sekiz sanal işlemci ile gerçekleştirilmiştir. Yöntem, doğru sayısal sonuçları kararlı bir şekilde sunmuştur. Paralel hızlanma işlemci sayısıyla, dış ve fisyon kaynağı problemleri için, sırasıyla beş ve yedi işlemciye kadar artmaktadır. Hızlanma değerleri çok çekirdekli bilgisayar belleğinin sınırlı kaynak paylaşımı nedeniyle kısıtlanmıştır. Diğer taraftan, paralel hesaplama ile önemli zaman kazanımları elde edilmiştir. Dört-grup fisyon kaynağı problemi için, 4316 interpolasyon noktası kullanılması durumunda, seri hesaplama yerine yedi işlemci kullanılması ağsız yöntemin hesaplama süresini 716 s kısaltmıştır.

Anahtar kelimeler: Paralel, Ağsız, Radyal baz fonksiyonu, Kollokasyon, Nötron difüzyonu

[*] Bursa Technical University, Faculty of Engineering and Natural Sciences, Department of Mechanical Engineering, Mimar Sinan Campus, 16310, Yıldırım/Bursa
Corresponding author: Tayfun Tanbay (tayfun.tanbay@btu.edu.tr)

1. INTRODUCTION

Meshless methods emerged in the 1970s for solving astrophysics problems (Lucy, 1977). Since their appearance in the literature, these methods were used to numerically solve partial differential equations (PDE) encountered in different branches of physics and became an alternative to classical mesh-based approaches such as the finite element method (FEM). The fundamental difference between meshless and mesh-based methods is that the discretization nodes of a meshless approach can be generated without any preliminary definition. This paved the way for the development of more flexible algorithms than mesh-based methods.

Based on their formulation, meshless techniques are classified into three groups (Liu, 2010): (i) Within the strong-form methods, the shape functions are substituted into the PDE and boundary conditions. The RBF collocation method, which was first developed for the solution of hydrodynamics problems by Kansa (1986), is the most prominent member of this class. (ii) On the other hand, the governing equation is multiplied by a weight function and the resulting expression is integrated over the entire problem region when a weak formulation is adopted. The integration weakens the continuity requirements of both the shape and weight functions. The radial version of point interpolation (Liu and Gu, 2005), Element free Galerkin (EFG) (Belytschko et al., 1994), and meshless local Petrov-Galerkin (MLPG) (Atluri and Zhu, 1998) methods are the most well-known weak-form based meshless approaches. (iii) Finally, within the hybrid methods, strong and weak-form algorithms are used together for different subregions of the problem to benefit from the advantageous characteristics of the two main classes.

Weak-form methods are inherently more stable than strong-form methods. Duan (2008) carried out a comparative analysis between RBF based strong- and weak-form meshless algorithms and showed that the use of RBFs in a weak formulation rather than the strong-form approach reduces the solution matrix's condition number by one order of magnitude. On the other hand, the strong-form RBF collocation method is an exponentially convergent method and can yield highly accurate solutions with fewer discretization nodes than other meshless and mesh-based methods (Li et al., 2003). Another important advantage of strong-form methods is that a background mesh is not required since there is no integration in their formulation procedures. Therefore, strong-form methods are truly meshless methods.

In the global RBF collocation approach, the computation times increase rapidly with the number of discretization nodes due to the fully populated structure of the collocation matrix. In this context, the use of parallel algorithms is a must to avoid long computation times. There exist several studies on the parallel implementations of meshless approaches. In one of the earliest works, the free mesh method was used in a parallel manner to solve an incompressible fluid flow problem (Shirazaki and Yagawa, 1999). Induced damage simulations of composites were carried out with the parallelized smoothed particle hydrodynamics (SPH) by Medina and Chen (2000). Parallel SPH was also used to model fluid flow problems with a high speedup performance (Ferrari et al., 2009, Ihmsen et al., 2011, Marrone et al., 2012, Dominguez et al., 2013a, Dominguez et al., 2013b, Cercos-Pita, 2015, Crespo et al., 2015). The parallel reproducing kernel particle method (RKPM) was used to model structural mechanics problems (Danielson et al., 2000) and supersonic flow over a NACA airfoil (Günther et al., 2000). Essential boundary conditions were implemented on a single processor by Danielson et al. (2000) and Günther et al. (2000). A hierarchical enrichment method was proposed by Zhang et al. (2002) for the complete parallelization of the RKPM to carry out 3-D CFD studies. In another work, the parallel version of the partition of unity approach was used to solve elliptic PDEs (Griebel and Schweitzer, 2003).

3-D transient heat conduction problems were solved with a coupled method of fundamental and particular solutions approach by Ingber et al. (2004) where parallel domain decomposition was applied to stabilize the interpolation matrix and decrease the computation time. The EFG method was applied to heat transfer (Singh and Jain, 2005a) and fluid dynamics (Singh and Jain, 2005b) in a parallel manner. The computational efforts showed that parallelization improved the

speedup and efficiency of calculation as the number of discretization nodes increased, where speedup is the ratio of wall clock times of sequential to parallel computation and computational efficiency is the ratio of speedup to number of processors. However, the efficiency decreased as more processors were employed in computations. Parallel RKPM was used for bulk forming simulation by Hu et al. (2007a), where the Taylor bar, back extrusion analysis, and wheel forging analysis problems were solved with the meshless method. Unlike the general trend of monotonic decrease with increasing processor number, the parallel computation efficiency had its lowest value when 16 processors were used in calculations for all the cases. The back extrusion and wheel forging cases were also studied with the parallel radial point interpolation approach (Hu et al., 2007b), and the results indicated that the computation efficiency decreased with the number of processors for the wheel forging problem, while a minimum efficiency was obtained with 32 processors for the back extrusion analysis.

The complexity of the parallel MLPG method were studied and compared with FEM and finite difference method (FDM) by Trobec et al. (2009). The 2-D time-dependent diffusion equation was solved, and parallel computation efficiencies of 70-80% were found in the numerical experiments. A parallel RBF interpolation algorithm having $O(N)$ complexity (i.e., the calculation load increases linearly with the number of interpolation nodes) and storage requirement was proposed by Yokota et al. (2010). Gaussian RBF was chosen as the radial function and using a supercomputer the authors reported parallel efficiencies of 84% and 79% for strong and weak scalings, respectively. A node pair-wise approach was proposed by Karatarakis et al. (2013) to decrease the computation time for constructing the stiffness matrix of the EFG. 2- and 3-D elasticity problems were solved in a parallel manner and significant speedup values were found in the numerical experiments. RBF collocation method was employed locally to model two-phase incompressible fluid flow (Kelly et al., 2014). The results of sequential and parallel computations showed that speedup factors improved as the interpolation nodes were refined. The meshless finite point method (FPM) was compared with the FEM for modeling 3-D aerodynamics by Ortega et al. (2014). The FPM was parallelized and simulations were carried out on a multicore computer. The speedup of parallel computations was limited due to the constant memory bandwidth and forced sharing of processor resources.

The local RBF collocation method was used in a parallel manner to solve a natural convection problem (Kosec et al., 2014). The simulations showed that the computation time could be decreased significantly with the parallel implementation of the meshless approach. In another CFD application, a parallel least squares fit based meshless method was used to solve compressible flow problems (Ma et al., 2014), where Hilbert space filling curves were used to enhance the algorithm for randomly distributed nodes. Later, the same researchers used the parallel meshless dynamic point cloud method to model compressible flow problems with moving boundaries (Ma et al., 2015). The results of Ma et al. (2014) and Ma et al. (2015) showed that the computation time and speedup depend strongly on the type of processor. The approach presented by Ma et al. (2014) was further improved with an implicit method and extended to solve 3-D cases by Zhang et al. (2018a).

In order to model elasticity problems with meshless and hybrid FEM-meshless methods, a parallel algorithm is presented by Ullah et al. (2016). Parallelization of the methods decreased the computation time, however, the computation efficiency decreased as the number of processors increased. Inviscid compressible flow was modeled with the meshless weighted least squares curve fit (WLSCF) method by Cao et al. (2019). A parallel algorithm was used with multi-layered point reordering to deal with the negative impact of irregularly distributed nodes on the memory of the processor unit. In another work, the parallel meshless WLSCF method was used to model turbulent flows (Zhang et al., 2020), where high levels of speedup were obtained for 2- and 3-D problems. A functional programming based parallel algorithm for meshless methods was presented by Barbosa et al. (2021). The MLPG method was chosen as the meshless approach, and it was found that an increment in number of processors improved the speedup, while it had a

negative impact on the computation efficiency. A Poisson disc sampling based parallel domain discretization approach was proposed by Depolli et al. (2022) for the hybrid RBF-FDM and meshless methods. In a recent work, the discrete least squares meshless method was used within a parallel computation strategy for the numerical solution of PDEs (Sefidgar et al., 2022). Poisson equation, dam-break, and sediment transport problems were studied, and it was found that the parallel computation efficiency decreased with an increasing number of processors. Finally, the meshless solutions of neutron diffusion and transport equations were presented by Rokrok et al. (2012), Tanbay and Ozgener (2013), Tanbay and Ozgener (2014), Tanbay (2018), Tanbay and Ozgener (2019), Tanbay (2019), Tanbay and Ozgener (2020), Kim et al. (2017), Kashi et al. (2017), Tayefi et al. (2018), Zhang et al. (2018b), Bassett and Kiedrowski (2019), Khuat and Kim (2019), Khuat et al. (2019), Alizadeh et al. (2021), and Bassett and Owen (2022). Among these studies, Tayefi et al. (2018) and Bassett and Kiedrowski (2019) presented parallel MLPG algorithms for neutron diffusion and transport, respectively.

The literature review showed that there exist no works on parallelization of the global RBF collocation. Therefore, the novelties of this paper can be expressed as follows:

- In this study, a parallel meshless global RBF collocation algorithm is proposed for the first time.
- The method is used to solve the 2-D multigroup neutron diffusion problems. To the best of author's knowledge, the parallel global RBF collocation method is implemented to model neutron diffusion for the first time.
- Three problems are solved with the developed parallel technique, and the impacts of the number of interpolation nodes and processors on the speedup and computational efficiency of the algorithm are investigated in detail.

2. MATERIAL AND METHODS

In this section, first, the meshless global RBF collocation method and its application to the multigroup neutron diffusion equation will be described. Then, information on the parallel implementation of the proposed algorithm within Mathematica will be provided.

2.1. Global RBF Collocation Solution of the Neutron Diffusion Equation

Consider a PDE and its boundary conditions in operator form:

$$\begin{aligned} L[u(\boldsymbol{x})] &= f(\boldsymbol{x}), \boldsymbol{x} \in \Omega \\ H[u(\boldsymbol{x})] &= g(\boldsymbol{x}), \boldsymbol{x} \in \Gamma \end{aligned} \tag{1}$$

In Eq. (1), L is a partial differential operator, H is a partial differential and/or algebraic operator, u is the dependent variable, f and g are driving functions, and $\boldsymbol{x} = (x, y)$ are spatial Cartesian coordinates. In order to solve Eq. (1) with RBF collocation, firstly, interpolation nodes with N_D members on the domain and N_B members on the boundary are generated as

$$\begin{aligned} D &= \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{N_D}\} \\ B &= \{\boldsymbol{x}_{N_D+1}, \dots, \boldsymbol{x}_{N_D+N_B}\} \\ E &= \{\boldsymbol{x}_{N_D+N_B+1}, \dots, \boldsymbol{x}_{N_T}\} \end{aligned} \tag{2}$$

where D and B are the sets of interpolation nodes for Ω and Γ, respectively. When the problem includes Neumann type boundary conditions, method's accuracy can be enhanced by creating interpolation nodes outside the problem region, and solving the PDE on the boundary (Fedoseyev

et al., 2002). This approach is used in this study and E, with N_E members, represents the interpolation nodes generated outside the problem domain. Hence the total number of interpolation nodes is $N_T = N_D + N_B + N_E$. The nodes can have a uniform or random distribution.

Following the generation of the interpolation nodes, the field variable of the PDE is approximated with a finite series of RBFs:

$$u(\boldsymbol{x}) \cong \sum_{j=1}^{N_T} a_j \psi_j(\boldsymbol{x}) \tag{3}$$

In Eq. (3), ψ is the RBF and the coefficients $a_j, j = 1, \dots, N_T$ are calculated at the end of the numerical solution. Approximating the field variable of Eq. (1) with Eq. (3), and collocating at $\boldsymbol{x}_i$ gives

$$\begin{aligned} \sum_{j=1}^{N_T} a_j L(\psi_{ij}) &= f_i, \qquad i = 1, \dots, N_D + N_B \\ \sum_{j=1}^{N_T} a_j H(\psi_{ij}) &= g_i, \qquad i = N_D + 1, \dots, N_D + N_B \end{aligned} \tag{4}$$

or in matrix form

$$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{b} \tag{5}$$

where $\psi_{ij} = \psi(\boldsymbol{x}_i)$, $f_i = f(\boldsymbol{x}_i)$, $g_i = g(\boldsymbol{x}_i)$ and the elements of the collocation matrix $\boldsymbol{K}$, and the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are determined by

$$\begin{aligned} k_{ij} &= \begin{cases} L(\psi_{ij}), & i = 1, \dots, N_D + N_B, \ \ j = 1, \dots, N_T \\ H(\psi_{ij}), & i = N_D + 1, \dots, N_D + N_B, \ \ j = 1, \dots, N_T \end{cases} \\ \boldsymbol{a} &= [a_1 \cdots a_{N_T}]^T, \ \ \boldsymbol{b} = \left[f(\boldsymbol{x}_1) \cdots f(\boldsymbol{x}_{N_D+N_B})\ g(\boldsymbol{x}_{N_D+1}) \cdots g(\boldsymbol{x}_{N_D+N_B})\right]^T \end{aligned} \tag{6}$$

The solution of Eq. (5) yields the coefficients $a_j, j = 1, \dots, N_T$, and thus the numerical result. Interpolation nodes can be used as the collocation nodes or a different set of nodes can be generated for collocation. In this work, the same nodes are utilized for both interpolation and collocation. The collocation matrix $\boldsymbol{K}$ is a fully populated matrix in the global RBF collocation method. By defining a support domain for all the interpolation nodes that cover the nearest neighbours of $\boldsymbol{x}_j$, the approach can be localized. The localization of the method leads to a sparse collocation matrix with higher stability, however, it also degrades the accuracy and the convergence characteristics of the global method.

Although various radial functions have been proposed, the most widely utilized radial function for RBF collocation is the multiquadric function, which was first proposed by Hardy (1971) for approximating geographic surfaces. Multiquadric is defined by

$$\psi_j(\boldsymbol{x}) = \sqrt{(x - x_j)^2 + (y - y_j)^2 + c^2} \tag{7}$$

In Eq. (7), c is the shape parameter and has a substantial impact on the numerical solution. Theoretically, it is shown by Madych (1992) that for function interpolation, the numerical error would vanish as $c \to \infty$ if the calculation could be performed with infinite precision arithmetic. In practice, increasing c decreases the approximation error and leads to a faster convergence, however, it also has an adverse effect on the stability. In fact, the shape parameter has an optimum value that provides a balance between accuracy and stability.

The neutron diffusion equation is a PDE that governs the interaction of neutrons with various materials in a nuclear reactor core. Modeling this behavior in an accurate and computationally efficient manner has a crucial role in the design of nuclear power plants. Within the multi-energy group approach having G energy groups, the operators, field variable, and right-hand side functions of Eq. (1) take the following form for the time-independent neutron diffusion equation in 2-D rectangular geometry:

$$\begin{gathered} L[\phi_g] = -D_g \nabla^2 \phi_g{}^{(m)} + \Sigma_{r,g} \phi_g{}^{(m)}, 0 \le x, y \le a \\ H[\phi_g] = \begin{cases} \phi_g, & x, y \in \Gamma_D \\ \dfrac{\partial \phi_g}{\partial n}, & x, y \in \Gamma_N \end{cases} \\ f = \sum_{g'=1}^{g-1} \Sigma_{s,g' \to g} \phi_{g'}^{(m)} + \frac{\chi_g}{k^{(m-1)}} \sum_{g'=1}^{G} \nu_{g'} \Sigma_{f,g'} \phi_{g'}^{(m-1)} + S_g \end{gathered} \tag{8}$$

where ϕ_g, is the neutron flux distribution, D_g is the diffusion constant, $\Sigma_{r,g}$ is the removal cross section, χ_g is the fission spectrum function, ν_g is the number of neutrons released per fission for group g, $\Sigma_{s,g' \to g}$ is a group-to-group scattering cross section representing the probability that a neutron would scatter from group g' down to group g, k is the multiplication factor, ∇^2 is the Laplacian, $\partial / \partial n$ represents the derivative in the normal direction, and m is the iteration index. Γ_D and Γ_N represent the Dirichlet and Neumann type boundaries, respectively. Two types of problems are encountered in nuclear reactor physics. If there is no fissile material (i.e. non-multiplying medium), the neutrons are supplied by an external source, S_g, and the neutron diffusion equation is solved directly with $f = S_g$. On the other hand, if the medium is multiplying, then the coupled G PDEs are solved with fission source iteration (Tanbay and Ozgener, 2014).

2.2. Parallel Computation with Mathematica

The parallel meshless RBF collocation method is implemented with an in-house code that is newly developed with Mathematica. Calculations are performed with a personal computer having a multicore Intel i5 1.60GHz processor with 4 cores and 8 threads. The processor architecture is x86-64, and the random access memory capacity is 8GB.

Mathematica uses the Wolfram Symbolic Transfer Protocol (WSTP) for processor communication. Resources are shared through a virtual shared memory and the message passing is based on the WSTP. By default, the number of processors utilized in calculations is equivalent to the number of physical cores. However, additional virtual processors can be activated by using the built-in function LaunchKernels[$N_{Processor}$] where $N_{Processor}$ represents the desired number of virtual processors (kernels in Mathematica's terminology). Since the processor has 8 threads, in this work 8 kernels are used in calculations.

RBF collocation solution of a PDE consists of three main steps:

1) Generating the discretization nodes of Eq. (2) utilized for interpolation and collocation
2) Calculating the elements of the collocation matrix, k_{ij} and the elements of the source vector, b_i of Eq. (6)

3) Solving Eq. (5) to obtain a_i

The generation of the discretization nodes and the calculation of k_{ij} and b_i with the developed parallel algorithm are performed with the built-in function ParallelTable. The arguments computed with the ParallelTable function are distributed to the kernels with another built-in function, DistributeDefinitions. The linear system is solved with the LinearSolve function, which employs efficient algorithms and does not have an effect on the parallel computation efficiency.

3. RESULTS AND DISCUSSION

The performance of the parallel RBF collocation algorithm is investigated through the solution of three test cases. A single energy group external source case, and two multigroup fission source cases are considered. Dirichlet type boundary conditions exist at $\boldsymbol{x} \in (a, y) \cup (x, a)$, while Neumann type boundary conditions exist at $\boldsymbol{x} \in (0, y) \cup (x, 0)$ for all the problems. Uniformly distributed interpolation nodes with a spacing of h are used, and the members of E are located at a distance of h from the boundaries. Parallel speedup and efficiency are defined as

$$\begin{aligned} \text{Speedup} &= \frac{T_1}{T_p} \\ \text{Efficiency} &= \frac{\text{Speedup}}{N_p} \end{aligned} \tag{9}$$

where T_1 and T_p represent the wall clock times with 1 and N_p processors, respectively. Accuracy of the solution is determined via the maximum and root mean square (RMS) errors in neutron flux for the external source problem

$$\begin{aligned} \epsilon_{max} &= \max_{1 \le i \le N_T} [|\phi_a(\boldsymbol{x}_i) - \phi_n(\boldsymbol{x}_i)|] \\ \epsilon_{RMS} &= \sqrt{\frac{1}{N_T} \sum_{i=1}^{N_T} [\phi_a(\boldsymbol{x}_i) - \phi_n(\boldsymbol{x}_i)]^2} \end{aligned} \tag{10}$$

while the accuracy of fission source problems is examined with the relative error in multiplication factor

$$\epsilon_k = \frac{|k_a - k_n|}{k_a} \tag{11}$$

where n and a refer to numerical and analytical, respectively. The performance of the method can be studied with both constant and node number dependent shape parameter strategies. With the variable shape parameter strategy, c is decreased with the number of nodes as $c = a/N_T^{1/2}$. Decreasing c with N_T has a stabilizing effect on the method and it was efficiently used for neutron transport equation (Tanbay and Ozgener, 2020).

3.1. External Source Problem

The following trigonometric source is considered for the external source case

$$S = \cos\left(\frac{\pi x}{2a}\right)\cos\left(\frac{\pi y}{2a}\right) \tag{12}$$

which yields an analytical neutron flux distribution expressed by

$$\phi_a = \frac{\cos\left(\frac{\pi x}{2a}\right)\cos\left(\frac{\pi y}{2a}\right)}{\Sigma_a + 2D\left(\frac{\pi}{2a}\right)^2} \tag{13}$$

where $\Sigma_a = \Sigma_r$. The geometric and nuclear parameters are chosen as $a = 0.25m$, $\Sigma_a = 1.43676\ m^{-1}$, and $D = 0.0177764m$ (Tanbay and Ozgener, 2014). Figure 1 shows the variation of RMS and maximum errors in neutron flux with the number of interpolation nodes on a semi-logarithmic scale. These results are obtained with a constant shape parameter of $c = a\sqrt{0.01}$. The convergence curves indicate that the parallel RBF collocation algorithm produces accurate flux distributions in a stable manner.

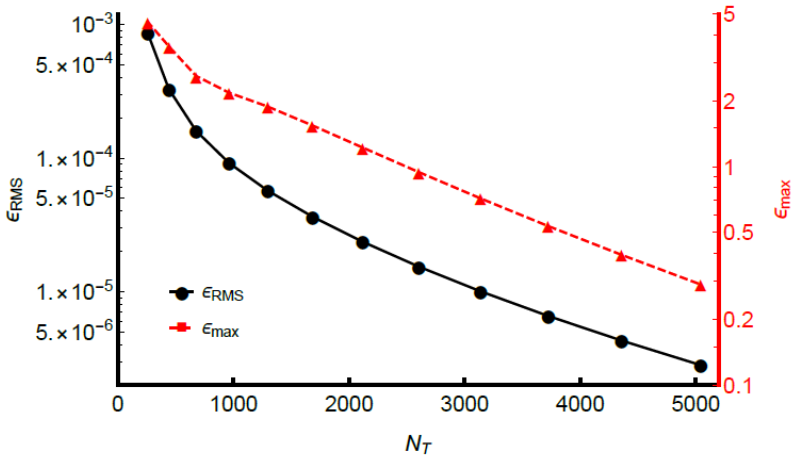

Figure 1:
Variation of ϵ_{max} and ϵ_{Rms} with N_T for the external source case

Speedup and efficiency of parallel computation of the one-energy group external source problem are presented in Figure 2 for four sets of interpolation nodes. Speedup of the computation improved with the number of processors up to $N_p = 5$ for all values of N_T. The communication between the processors becomes apparent for $N_p \geq 6$ which leads to an asymptotic behavior of the speedup. Similar to the findings of Ortega et al. (2014), the speedup is limited due to the fact that a multicore computer was used for the calculations. The efficiency of the parallel algorithm decreases as N_p increases, and such a trend is frequently observed in parallel implementations of numerical methods.

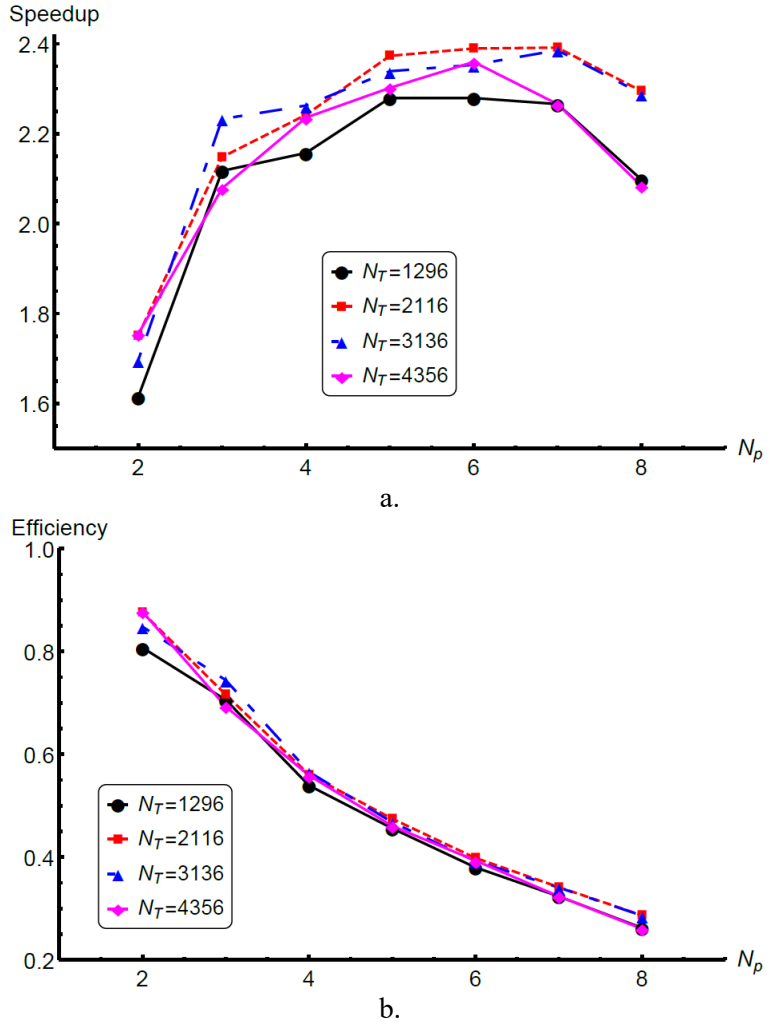

Figure 2:
a. Speedup and b. efficiency of parallel computation for the external source case

3.2. Two-group Fission Source Problem

For the two-group problem, the domain size is $a = 0.25m$ and the nuclear constants are presented in Table 1, which yield a multiplication factor of $k_a = 1.96413$. A convergence parameter of 10^{-6} is used for the iterative solution. Figure 3 demonstrates the variation of ϵ_k with N_T where $c = a/N_T^{1/2}$, and the results are found with 29 iterations. The error values clearly display the accuracy and stability of the meshless algorithm.

Table 1. Nuclear parameters for the two-group fission source problem (Tanbay and Ozgener, 2013)

Group	$D(m)$	ν	$\Sigma_f(m^{-1})$	$\Sigma_r(m^{-1})$	$\Sigma_{s,g\to g+1}(m^{-1})$	χ
1	0.012245	2.65	6.300	13.552	6.776	0.575
2	0.012245	2.55	6.776	8.228	-	0.425

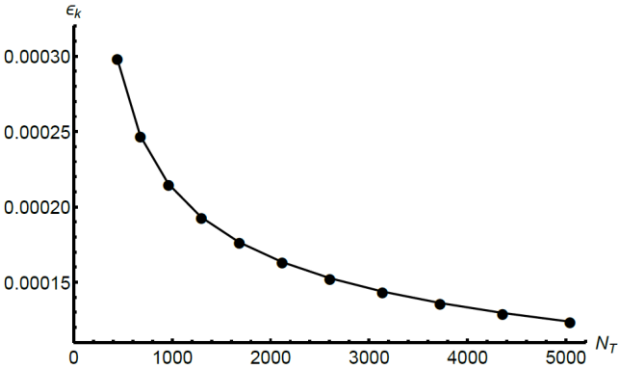

Figure 3:
Convergence of the two-group fission source problem

Figure 4 illustrates the parallelization speedup and efficiency of the RBF collocation approach. The speedup of the algorithm improves up to 7 processors, and the efficiency decreases with increasing N_p. A comparison with the external source problem shows that the speedup and efficiency of parallelization are higher for the two-group fission source case. The highest speedup values for the first and second cases are found to be 2.537 with $N_T = 1681$ and $N_p = 5$ and 3.158 with $N_T = 1296$ and $N_p = 7$, respectively. This improvement in parallel speedup is due to the fact that the fission source problem is much more CPU-intensive than the external source case. Two coupled PDEs are solved iteratively for the fission source case, while a single PDE is solved directly for the external source problem.

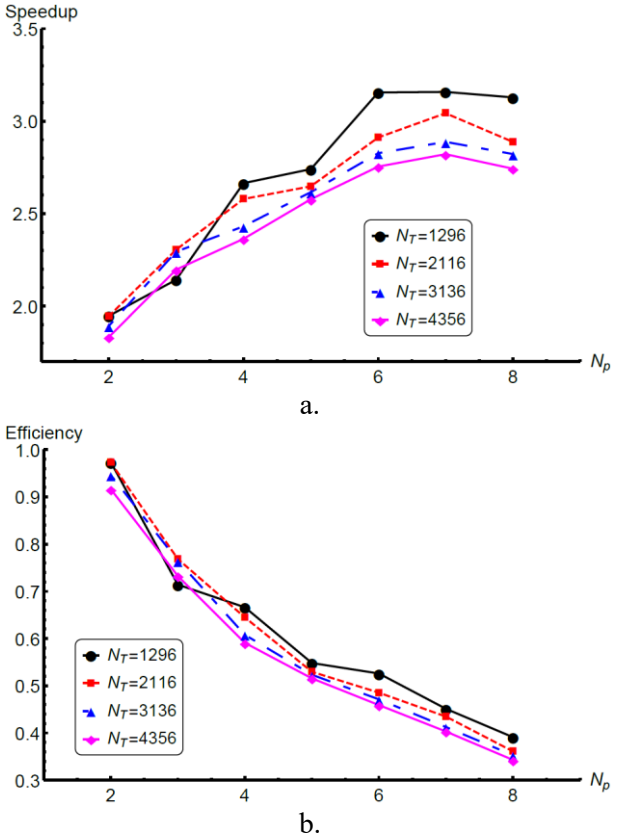

Figure 4:
a. Speedup and b. efficiency of parallel computation for the two-group fission source problem

3.3. Four-group Fission Source Problem

The third and final case is a four-group problem for which $a = 0.5m$. The nuclear parameters are provided in Table 2, where D and Σ have dimensions of m and m^{-1}, respectively. These data yield an exact multiplication factor of $k_a = 0.87227$. The numerical solutions are obtained with 13 iterations where the iteration convergence parameter is 10^{-6}. The parallel approach produces accurate and stable results as illustrated with the convergence curve in Figure 5, which is obtained with $c = a/N_T^{1/2}$.

Table 2. Nuclear parameters for the four-group fission source problem (Tanbay and Ozgener, 2014)

Group	D	ν	Σ_f	Σ_r	$\Sigma_{s,g\to g+1}$	$\Sigma_{s,g\to g+2}$	$\Sigma_{s,g\to g+3}$	χ
1	2.876787	2.4	0.0049492	0.028204	0.023597	4.079×10^{-6}	4.449×10^{-8}	0.768
2	1.570845	2.4	0.0022188	0.005275	0.001615	4.231×10^{-8}	-	0.232
3	0.722486	2.4	0.0043629	0.017612	0.004684	-	-	0
4	0.964199	2.4	0.0110879	0.026546	-	-	-	0

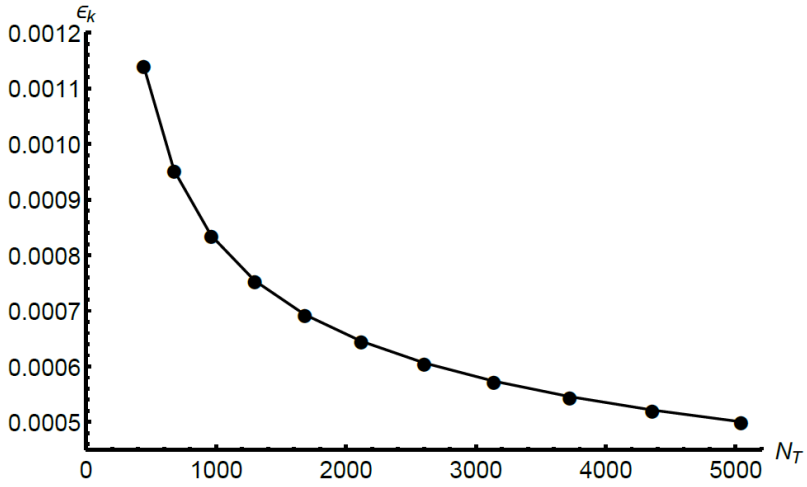

Figure 5:
Convergence of the four-group fission source problem

The wall clock time, speedup, and computational efficiency of the parallel algorithm for the four-group case are presented in Table 3 for three sets of interpolation nodes. The speedup of the method improves as N_p increases except at $N_p = 8$ where overhead occurs due to communication. Although the speedup is limited by the multicore processor, the results show the significance of parallel computation. For instance, the wall clock time of the RBF collocation decreased by 716 s for $N_T = 4356$ when 7 virtual processors are utilized instead of sequential computation.

Table 3. Wall clock time, speedup, and computational efficiency of the method for the four-group fission source problem

N_p	1	2	3	4	5	6	7	8
$N_T = 2116$								
Time (s)	272.855	142.481	113.544	107.071	99.619	96.276	90.840	93.355
Speedup	1	1.915	2.403	2.548	2.739	2.834	3.004	2.923
Efficiency	1	0.957	0.801	0.637	0.548	0.472	0.429	0.365
$N_T = 3136$								
Time (s)	571.171	311.132	256.496	236.229	217.572	205.266	199.596	203.313
Speedup	1	1.836	2.227	2.418	2.625	2.783	2.862	2.809
Efficiency	1	0.918	0.742	0.604	0.525	0.463	0.409	0.351
$N_T = 4356$								
Time (s)	1100.160	600.250	505.445	459.450	419.364	396.436	384.242	396.828
Speedup	1	1.833	2.177	2.394	2.623	2.775	2.863	2.772
Efficiency	1	0.916	0.725	0.599	0.525	0.462	0.409	0.346

4. CONCLUSIONS

The global radial basis function collocation method is an exponentially convergent and highly accurate technique to solve partial differential equations. However, the fully populated collocation matrix of the approach increases the computation time in a fast manner as the interpolation node set gets denser. Therefore, parallelization of the method is imperative for cases that require a large number of discretization nodes, such as multi-region configurations where the dependent variable may change sharply between different regions, to reflect the physical phenomena accurately.

In this work, a parallel meshless global RBF collocation algorithm is developed and it is utilized for the solution of the 2-D neutron diffusion equation. Multiquadric is employed as the RBF and three problems, including a one-group external source case, a two-group fission source case and a four-group fission source case, are solved with the proposed algorithm. The speedup and computation efficiency are used as the criteria to assess the performance of parallelization while maximum and RMS errors in neutron flux distribution and relative error in multiplication factor are considered to measure accuracy. The RBF collocation method produces accurate and stable numerical solutions. Eight virtual processors are used in the numerical experiments, and the results show that the speedup of the method improves up to five and seven processors for the one-group external and multigroup fission source problems, respectively. Although speedup values are limited by the memory constraint of the multicore computer, the parallel approach provides significant time savings for CPU-intensive problems.

In recent years, graphics processing units (GPU) become indispensable hardware elements for modeling complex physical phenomena. Parallelization of meshless methods with the aid of GPU has resulted in high levels of speedup and efficiency in computational fluid dynamics applications. Therefore, future works can be performed on implementing the parallel global RBF collocation method with GPU clusters. High performance computation with this parallel meshless technique has the potential to produce exceptionally accurate solutions for 3-D, multi-region nuclear reactor physics problems.

CONFLICT OF INTEREST

The author acknowledges that there is no known conflict of interest or common interest with any institution/organization or person.

AUTHOR CONTRIBUTION

Determination of the concept and design process of the research, development of the algorithm, simulations, analysis and discussion of the results and preparation of the paper were carried out by Tayfun Tanbay.

REFERENCES

1. Alizadeh, A., Abbasi M., Minuchehr, A. and Zolfaghari, A. (2021) A mesh-free treatment for even parity neutron transport equation, *Annals of Nuclear Energy*, 158, 108292. doi:10.1016/j.anucene.2021.108292

2. Atluri, S.N. and Zhu, T. (1998) A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics, *Computational Mechanics*, 22, 117-127. doi: 10.1007/s004660050346

3. Barbosa, M., Telles, J.C.F., Santiago, J.A.F., Junior, E.F.F. and Costa, E.G.A. (2021) A parallel implementation strategy for meshless methods based on the functional programming paradigm, *Advances in Engineering Software*, 151, 102926. doi: 10.1016/j.advengsoft.2020.102926

4. Bassett, B. and Kiedrowski, B. (2019) Meshless local Petrov-Galerkin solution of the neutron transport equation with streamline-upwind Petrov-Galerkin stabilization, *Journal of Computational Physics*, 377, 1-59. doi:10.1016/j.jcp.2018.10.028

5. Bassett, B. and Owen, J.M. (2022) Meshless discretization of the discrete-ordinates transport equation with integration based on Voronoi cells, *Journal of Computational Physics*, 449, 110697. doi:10.1016/j.jcp.2021.110697

6. Belytschko, T., Lu, Y.Y. and Gu, L. (1994) Element-free Galerkin methods, *International Journal for Numerical Methods in Engineering*, 37, 229-256. doi:10.1002/nme.1620370205

7. Cao, C., Chen, H.Q., Zhang, J.L. and Xu, S.G. (2019) A multi-layered point reordering study of GPU-based meshless method for compressible flow simulations, *Journal of Computational Science*, 33, 45-60. doi:10.1016/j.jocs.2019.04.001

8. Cercos-Pita, J.L. (2015) AQUAgpusph, a new free 3D SPH solver accelerated with OpenCL, *Computer Physics Communications*, 192, 295-312. doi:10.1016/j.cpc.2015.01.026

9. Crespo, A.J.C., Domínguez, J.M., Rogers B.D., Gómez-Gesteira, M., Longshaw, S., Canelas, R., Vacondio, R., Barreiro, A. and García-Feal, O. (2015) DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH), *Computer Physics Communications*, 187, 204-216. doi:10.1016/j.cpc.2014.10.004

10. Danielson, K.T., Hao, S., Liu, W.K., Uras, R.A. and Li, S. (2000) Parallel computation of meshless methods for explicit dynamic analysis, *International Journal for Numerical Methods in Engineering*, 47, 1323-1341. doi:10.1002/(SICI)1097-0207(20000310)47:7<1323::AID-NME827>3.0.CO;2-0

11. Depolli, M., Slak, J. and Kosec, G. (2022) Parallel domain discretization algorithm for RBF-FD and other meshless numerical methods for solving PDEs, *Computers and Structures*, 264, 106773. doi:10.1016/j.compstruc.2022.106773

12. Domínguez, J.M., Crespo A.J.C., Valdez-Balderas, D., Rogers, B.D. and Gómez-Gesteira, M. (2013a) New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters, *Computer Physics Communications*, 184, 1848-1860. doi:10.1016/j.cpc.2013.03.008

13. Domínguez, J.M., Crespo A.J.C. and Gómez-Gesteira, M. (2013b) Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method, *Computer Physics Communications*, 184, 617-627. doi:10.1016/j.cpc.2012.10.015

14. Duan, Y. (2008) A note on the meshless method using radial basis functions, *Computers and Mathematics with Applications*, 55, 66-75. doi:10.1016/j.camwa.2007.03.011

15. Fedoseyev, A.I., Friedman, M.J. and Kansa, E.J. (2002) Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary*, Computers and Mathematics with Applications*, 43, 439-455. doi:10.1016/S0898-1221(01)00297-8

16. Ferrari, A., Dumbser, M., Toro, E.F. and Armanini, A. (2009) A new 3D parallel SPH scheme for free surface flows, *Computers & Fluids*, 38, 1203-1217. doi:10.1016/j.compfluid.2008.11.012

17. Griebel, M. and Schweitzer, M.A. (2003). A Particle-Partition of Unity Method-Part IV: Parallelization. In: Griebel, M. and Schweitzer, M.A. (eds) *Meshfree Methods for Partial Differential Equations*. Lecture Notes in Computational Science and Engineering, vol 26. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-56103-0_12

18. Günther, F., Liu, W.K., Diachin, D. and Christon, M.A. (2000) Multi-scale meshfree parallel computations for viscous, compressible flows, *Computer Methods in Applied Mechanics and Engineering*, 190, 279-303. doi:10.1016/S0045-7825(00)00202-4

19. Hardy, R.L. (1971) Multiquadric equations of topography and other irregular surfaces, *Journal of Geophysical Research*, 76, 1905-1915. doi:10.1029/JB076i008p01905

20. Hu, W., Yao, L.G., Xu, H. and Hua, Z.Z. (2007a) Development of parallel 3D RKPM meshless bulk forming simulation system, *Advances in Engineering Software*, 38, 87-101. doi:10.1016/j.advengsoft.2006.08.002

21. Hu, W., Yao, L.G. and Hua, Z.Z., (2007b) Parallel point interpolation method for three-dimensional metal forming simulations, *Engineering Analysis with Boundary Elements*, 31, 326-342. doi:10.1016/j.enganabound.2006.09.012

22. Ihmsen, M., Akinci, N., Becker, M. and Teschner, M. (2011) A parallel SPH implementation on multi-core CPUs, *Computer Graphics Forum*, 30, 99-112. doi:10.1111/j.1467-8659.2010.01832.x

23. Ingber, M.S., Chen, C.S. and Tanski, J.A. (2004) A mesh free approach using radial basis functions and parallel domain decomposition for solving three-dimensional diffusion equations, *International Journal for Numerical Methods in Engineering*, 60, 2183-2201. doi:10.1002/nme.1043

24. Kansa, E.J. (1986) Application of Hardy's multiquadric interpolation to hydrodynamics, *Proceedings of the 1986 Summer Computer Simulation Conference*, Society for Computer Simulation, San Diego, 4, 111-117.

25. Kashi, S., Minuchehr, A., Zolfaghari, A. and Rokrok, B. (2017) Mesh-free method for numerical solution of the multi-group discrete ordinate neutron transport equation, *Annals of Nuclear Energy*, 106, 51-63. doi:10.1016/j.anucene.2017.03.034

26. Karatarakis, A., Metsis, P. and Papadrakakis, M. (2013) GPU-acceleration of stiffness matrix calculation and efficient initialization of EFG meshless methods, *Computer Methods in Applied Mechanics and Engineering*, 258, 63-80. doi:10.1016/j.cma.2013.02.011

27. Kelly, J.M., Divo, E.A. and Kassab, A.J. (2014) Numerical solution of the two-phase incompressible Navier-Stokes equations using a GPU-accelerated meshless method, *Engineering Analysis with Boundary Elements*, 40, 36-49. doi:10.1016/j.enganabound.2013.11.015

28. Khuat, Q.H., Hoang, S.M.T., Woo, M.H., Kim, J.H. and Kim, J.K. (2019) Unstructured discrete ordinates method based on radial basis function approximation, *Journal of the Korean Physical Society*, 75, 5-14. doi: 10.3938/jkps.75.5

29. Khuat, Q.H. and Kim, J.K. (2019) A solution to the singularity problem in the meshless method for neutron diffusion equation, *Annals of Nuclear Energy*, 126, 178-185. doi:10.1016/j.anucene.2018.10.054

30. Kim, K., Jeong, H.S. and Jo, D. (2017) Numerical analysis for multi-group neutron-diffusion equation using Radial Point Interpolation Method (RPIM), *Annals of Nuclear Energy*, 99, 193-198. doi:10.1016/j.anucene.2016.08.021

31. Kosec, G., Depolli, M., Rashkovska, A. and Trobec, R. (2014) Super linear speedup in a local parallel meshless solution of thermo-fluid problems, *Computers and Structures*, 133, 30-38. doi:10.1016/j.compstruc.2013.11.016

32. Li, J., Cheng, A.H.D. and Chen, C.S. (2003) A comparison of efficiency and error convergence of multiquadric collocation method and finite element method, *Engineering Analysis with Boundary Elements*, 27, 251-257. doi:10.1016/S0955-7997(02)00081-4

33. Liu, G.R. (2010) *Meshfree Methods: Moving Beyond The Finite Element Method 2nd Edition*, CRC Press, USA.

34. Liu, G.R. and Gu, Y.T. (2005) *An Introduction to Meshfree Methods and Their Programming*, Springer, Dordrecht.

35. Lucy, L.B. (1977) A numerical approach to the testing of the fission hypothesis, *The Astronomical Journal*, 82, 1013-1024. doi:10.1086/112164

36. Ma, Z.H., Wang, H. and Pu, S.H. (2014) GPU computing of compressible flow problems by a meshless method with space-filling curves, *Journal of Computational Physics*, 263, 113-135. doi:10.1016/j.jcp.2014.01.023

37. Ma, Z.H., Wang, H. and Pu, S.H. (2015) A parallel meshless dynamic cloud method on graphic processing units for unsteady compressible flows past moving boundaries, *Computer Methods in Applied Mechanics and Engineering*, 285, 146-165. doi:10.1016/j.cma.2014.11.010

38. Madych, W.R. (1992) Miscellaneous error bounds for multiquadric and related interpolators, *Computers and Mathematics with Applications*, 24, 121-138. doi:10.1016/0898-1221(92)90175-H

39. Marrone, S., Bouscasse, B., Colagrossi, A. and Antuono, M. (2012) Study of ship wave breaking patterns using 3D parallel SPH simulations, *Computers & Fluids*, 69, 54-66. doi:10.1016/j.compfluid.2012.08.008

40. Medina, D.F. and Chen, J.K. (2000) Three-dimensional simulations of impact induced damage in composite structures using the parallelized SPH method, *Composites: Part A*, 31, 853-860. doi:10.1016/S1359-835X(00)00031-2

41. Ortega, E., Oñate, E., Idelsohn, S. and Flores, R. (2014) Comparative accuracy and performance assessment of the finite point method in compressible flow problems, *Computers & Fluids*, 89, 53-65. doi:10.1016/j.compfluid.2013.10.024

42. Rokrok, B., Minuchehr, H. and Zolfaghari, A. (2012) Element-free Galerkin modeling of neutron diffusion equation in X-Y geometry, *Annals of Nuclear Energy*, 43, 39-48. doi:10.1016/j.anucene.2011.12.032

43. Sefidgar, S.M.H., Firoozjaee, A.R. and Dehestani, M. (2022) Sparse discrete least squares meshless method on multicore computers, *Journal of Computational Science*, 62, 101686. doi:10.1016/j.jocs.2022.101686

44. Shirazaki, M. and Yagawa, G. (1999) Large-scale parallel flow analysis based on free mesh method: a virtually meshless method, *Computer Methods in Applied Mechanics and Engineering*, 174, 419-431. doi:10.1016/S0045-7825(98)00307-7

45. Singh, I.V. and Jain, P.K. (2005a) Parallel EFG algorithm for heat transfer problems, *Advances in Engineering Software*, 36, 554-560. doi:10.1016/j.advengsoft.2005.01.009

46. Singh, I.V. and Jain, P.K. (2005b) Parallel meshless EFG solution for fluid flow problems, *Numerical Heat Transfer, Part B: Fundamentals*, 48, 45-66. doi:10.1080/10407790590935993

47. Tanbay, T. (2018) On the accuracy and stability of the meshless RBF collocation method for neutron diffusion calculations, *Journal of Innovative Science and Engineering*, 2, 8-18.

48. Tanbay, T. (2019) Meshless solution of the neutron diffusion equation by the RBF collocation method using optimum shape parameters, *Journal of Innovative Science and Engineering*, 3, 23-31. doi:10.38088/jise.570328

49. Tanbay, T. and Ozgener, B. (2013) Numerical solution of the multigroup neutron diffusion equation by the meshless RBF collocation method, *Mathematical and Computational Applications*, 18, 399-407. doi:10.3390/mca18030399

50. Tanbay, T. and Ozgener, B. (2014) A comparison of the meshless RBF collocation method with finite element and boundary element methods in neutron diffusion calculations, *Engineering Analysis with Boundary Elements*, 46, 30-40. doi:10.1016/j.enganabound.2014.05.005

51. Tanbay, T. and Ozgener, B. (2019) A meshless method based on symmetric RBF collocation for neutron diffusion problems, *Acta Physica Polonica A*, 135, 661-663. doi:10.12693/APhysPolA.135.661

52. Tanbay, T. and Ozgener, B. (2020) Fully meshless solution of the one-dimensional multigroup neutron transport equation with the radial basis function collocation method, *Computers and Mathematics with Applications*, 79, 1266-1286. doi:10.1016/j.camwa.2019.08.037

53. Tayefi, S., Pazirandeh, A. and Saadi, M.K. (2018) A meshless local Petrov-Galerkin method for solving the neutron diffusion equation, *Nuclear Science and Techniques*, 29, 169. doi:10.1007/s41365-018-0506-x

54. Trobec, R., Šterk, M. and Robic, B. (2009) Computational complexity and parallelization of the meshless local Petrov-Galerkin method, *Computers and Structures*, 87, 81-90. doi:10.1016/j.compstruc.2008.08.003

55. Ullah, Z., Coombs, W. and Augarde, C. (2016) Parallel computations in nonlinear solid mechanics using adaptive finite element and meshless methods, *Engineering Computations*, 33, 1161-1191. doi:10.1108/EC-06-2015-0166

56. Yokota, R., Barba, L.A. and Knepley, M.G. (2010) PetRBF – A parallel O(N) algorithm for radial basis function interpolation with Gaussians, *Computer Methods in Applied Mechanics and Engineering*, 199, 1793-1804. doi:10.1016/j.cma.2010.02.008

57. Zhang, L.T., Wagner, G.J. and Liu, W.K. (2002) A parallelized meshfree method with boundary enrichment for large-scale CFD, *Journal of Computational Physics*, 176, 483-506. doi:10.1006/jcph.2002.6999

58. Zhang, J.L., Ma, Z.H., Chen, H.Q. and Cao, C. (2018a) A GPU-accelerated implicit meshless method for compressible flows, *Journal of Computational Physics*, 360, 39-56. doi:10.1016/j.jcp.2018.01.037

59. Zhang, Y.N. Zhang, H.C. Zhang, X. Yu, H.X. and Zhao, G.B. (2018b) Block Radial Basis Function Collocation Meshless method applied to steady and transient neutronics problem solutions in multi-material reactor cores, *Progress in Nuclear Energy*, 109, 83-96. doi:10.1016/j.pnucene.2018.08.010

60. Zhang, J.L., Chen, H.Q., Xu, S.G. and Gao, H.Q. (2020) A novel GPU-parallelized meshless method for solving compressible turbulent flows, *Computers and Mathematics with Applications*, 80, 2738-2763. doi:10.1016/j.camwa.2020.08.030