Wavelet compression of parabolic diffusion integral kernels

J. Ravnik & L. Škerget
Faculty of Mechanical Engineering, University of Maribor, Slovenia

Abstract

In this paper we examine the properties of steady and unsteady diffusion fundamental solutions in the framework of fast BEM. The fundamental solutions and their kernels form integral kernels in boundary integral equations. The properties of these kernels define the relationship between CPU time and storage gain versus the accuracy of sparse approximations in fast BEM.

We show that, when solving a diffusion type problem, using unsteady fundamental solutions is advantageous over steady fundamental solutions. We examine the behaviour of fundamental solutions to show why the unsteady fundamental solution is better for fast BEM. Furthermore, we confirm the theoretical findings by simulation of viscous fluid flow and heat transfer. We consider steady cases, where a false transient approach is used so unsteady diffusion fundamental solutions may be employed. Finally, we examine cases of increasing non-linearity, to highlight that these findings apply to strongly non-linear problems as well.

Keywords: steady diffusion, unsteady diffusion, wavelet transform, sparse approximation, integral kernel, fast BEM.

1 Introduction

One of the main impediments of the boundary element method (BEM) is the need for solving large, fully populated, linear systems of equations resulting from numerical discretization of the governing equations. In cases of non-linear or non-homogenous partial differential equations domain discretization is needed, which requires even larger fully populated matrices.

In the last decade several approaches have been proposed to overcome this disadvantage. They are known as fast BEM as their purpose is to reduce the

WIT Transactions on Modelling and Simulation, Vol 57, © 2014 WIT Press
www.witpress.com, ISSN 1743-355X (on-line)
doi:10.2495/BE370141

required computational resources from the $\mathcal{O}(n^2)$ dependence of the original BEM to $\mathcal{O}(n \log n)$ or $\mathcal{O}(n)$ of the fast BEM. Here, n represents the number of nodes in the computational mesh.

One of the ideas of fast BEM is the construction of data sparse approximations of fully populated matrices. Hackbusch and Nowak [10] developed a panel clustering method, which enables approximate matrix vector multiplications with decreased amount of arithmetical work. A class of hierarchical matrices was introduced by Hackbusch [9] with the aim of reducing the complexity of matrix-vector multiplications. Bebendorf and Rjasanow [2] developed an algebraic approach for solving integral equations using collocation methods with almost linear complexity. Ravnik *et al.* [14] developed a wavelet compression method and used it for compression of single domain BEM in 2D. The fast multipole method, which has been developed for particle simulations by Greengard and Rokhlin [6] has also been used with BEM. The algorithm decreases the amount of work required to evaluate mutual interaction of particles by reducing the complexity of the problem from quadratic to linear. Ever since, the method was used by many authors for a wide variety of problems using different expansion strategies. Bui *et al.* [4] combined FMM with the Fourier transform to study multiple bubbles dynamics. Gumerov and Duraiswani [8] applied the FMM for the biharmonic equation in three dimensions. The boundary integral Laplace equation was accelerated with FMM by Popov *et al.* [13]. Ravnik *et al.* [16] compared wavelet and fast multipole data sparse approximations for boundary–domain integral equations of Poisson type. Koro and Abe worked on developing a practical determination strategy of optimal threshold parameter for matrix compression in wavelet BEM [11]. Xiao *et al.* [21] developed an a-posteriori technique for BEM compression. Bucher *et al.* [3] studied the application of fast wavelet transforms.

Although these methods all yield nearly linear or linear scaling of computing resources with mesh size, their success still depends on the type of the integral kernel. Namely, when n is not very large, the compression ratios of these methods may not be very good, i.e., maybe a sparse representation of the BEM matrix has 50% of the data of the original matrix. In such cases, these methods are useless, since a 50% reduction does not enable a mesh density increase, which would have a significant effect on the physics that needs to be simulated. In other words, by only halving the computer resources, we are not able to simulate any new physical phenomena.

Fortunately, the compression ratio depends on the choice of the fundamental solution. In this work, we analyse the effect of making the steady diffusion equation unsteady by introducing an artificial accumulation term, i.e. we use the false transient approach. In this case, parabolic diffusion fundamental solution may be used instead of the Laplace fundamental solution. We will show, that the compression ratios achieved in this case are much greater than when using the Laplace fundamental solution.

2 Analysis of diffusion fundamental solutions

2.1 Steady diffusion

The steady state diffusion equation reads

$$\alpha_0 \nabla^2 u(\vec{r}) = 0, \qquad \vec{r} \in \Omega \tag{1}$$

where α_0 is a constant diffusivity term, $\vec{r}$ is the location vector defined in the domain Ω and $u(\vec{r})$ is the field function. The $3D$ and $2D$ fundamental solutions and their gradients of this equation are

$$u^\star_{3D} = \frac{1}{4\pi\alpha_0|\vec{\xi} - \vec{r}|}, \qquad u^\star_{2D} = \frac{1}{2\pi\alpha_0} \ln \frac{1}{|\vec{\xi} - \vec{r}|}, \tag{2}$$

$$\vec{\nabla} u^\star_{3D} = \frac{\vec{\xi} - \vec{r}}{4\pi\alpha_0|\vec{\xi} - \vec{r}|^3}, \qquad \vec{\nabla} u^\star_{2D} = \frac{\vec{\xi} - \vec{r}}{2\pi\alpha_0|\vec{\xi} - \vec{r}|^2}, \tag{3}$$

where $\vec{\xi}$ is the source point. All expressions in (2) and (3) are singular at $\vec{r} = \vec{\xi}$ and diminish as $|\vec{\xi} - \vec{r}|$ increases.

2.2 Unsteady diffusion

The fundamental solution of the unsteady diffusion equation

$$\frac{\partial u}{\partial t} = \alpha_0 \nabla^2 u \tag{4}$$

is a free-space Green's function, describing the function field generated by a unit source applied at a point $\vec{\xi}$ at time t_0, i.e.

$$u^\star(\vec{r}, \vec{\xi}, t, t_F) = \frac{1}{(4\pi\alpha_0(t_F - t))^{3/2}} \exp\left(-\frac{(\vec{\xi} - \vec{r})^2}{4\alpha_0(t_F - t)}\right), \qquad t \geq t_0. \tag{5}$$

The gradient of the fundamental solution is

$$\vec{\nabla} u^\star(\vec{r}, \vec{\xi}, t, t_F) = \frac{(\vec{\xi} - \vec{r})}{16\pi^{3/2}(\alpha_0(t_F - t))^{5/2}} \exp\left(-\frac{(\vec{\xi} - \vec{r})^2}{4\alpha_0(t_F - t)}\right) \tag{6}$$

The integral form of equation (4) is (Wrobel [20]):

$$c(\vec{\xi})u(\vec{\xi}, t_F) + \alpha_0 \int_{t_0}^{t_F} \int_\Gamma u(\vec{r}, t) \vec{\nabla} u^\star \cdot d\vec{\Gamma} dt = \alpha_0 \int_{t_0}^{t_F} \int_\Gamma u^\star \vec{\nabla} u(\vec{r}, t) \cdot d\vec{\Gamma} dt$$
$$+ \int_\Omega u(\vec{r}, t_0) u^\star(\vec{r}, \vec{\xi}, t_0, t_F) d\Omega, \tag{7}$$

where $c(\vec{\xi})$ is the geometric factor defined as $c(\vec{\xi}) = \beta/4\pi$, where β is the inner angle with origin in $\vec{\xi}$. If $\vec{\xi}$ lies inside of the domain then $c(\vec{\xi}) = 1$; $c(\vec{\xi}) = 1/2$, if $\vec{\xi}$ lies on a smooth boundary; and $\Gamma = \partial\Omega$ is the boundary of the domain. If we consider that the field function does not change within a time step (i.e. we consider constant time interpolation) we may change the order of integrals and integrate over time:

$$\tilde{u}^\star = \lim_{t_1 \to t_F} \int_{t_0}^{t_1} u^\star(\vec{r}, \vec{\xi}, t, t_F)dt = \frac{1}{4\pi\alpha_0|\vec{\xi} - \vec{r}|}\operatorname{erfc}\left(\sqrt{\frac{(\vec{\xi} - \vec{r})^2}{4\alpha_0\Delta t}}\right), \tag{8}$$

where $\Delta t = t_1 - t_0$. For long time steps $\tilde{u}^\star$ reduces to the Laplace fundamental solution (2). The time integral of the gradient of $u^\star$ is

$$\vec{\nabla}\tilde{u}^\star = \lim_{t_1 \to t_F} \int_{t_0}^{t_1} \vec{\nabla}u^\star(\vec{r}, \vec{\xi}, t, t_F)dt = \frac{(\vec{\xi} - \vec{r})}{2\pi^{3/2}\alpha_0|\vec{\xi} - \vec{r}|^3} \cdot$$
$$\cdot \left[\left(\sqrt{\frac{(\vec{\xi} - \vec{r})^2}{4\alpha_0\Delta t}}\right)\exp\left(-\frac{(\vec{\xi} - \vec{r})^2}{4\alpha_0\Delta t}\right) + \frac{\sqrt{\pi}}{2}\operatorname{erfc}\left(\sqrt{\frac{(\vec{\xi} - \vec{r})^2}{4\alpha_0\Delta t}}\right)\right], \tag{9}$$

which for long time steps reduces to the gradient of the diffusion fundamental solution (3).

Using the calculated time integrals, we may write the integral form of the governing equations as:

$$c(\vec{\xi})u(\vec{\xi}, t_F) + \alpha_0 \int_\Gamma u(\vec{r}, t)\vec{\nabla}\tilde{u}^\star \cdot d\vec{\Gamma} = \alpha_0 \int_\Gamma \tilde{u}^\star\vec{\nabla}u(\vec{r}, t) \cdot d\vec{\Gamma}$$
$$+ \int_\Omega u(\vec{r}, t_0)u^\star(\vec{r}, \vec{\xi}, t_0, t_1)d\Omega. \tag{10}$$

During discretization of equation (10) the boundary and domain integrals having $u^\star$, $\tilde{u}^\star$ and $\vec{\nabla}\tilde{u}^\star$ as kernels must be evaluated. In order to observe how these functions behave, we define a new parameter γ as

$$\gamma = \sqrt{\frac{(\vec{\xi} - \vec{r})^2}{4\alpha_0\Delta t}}. \tag{11}$$

Then we may express the kernels with γ as:

$$u^\star = \frac{1}{(4\pi\alpha_0\Delta t)^{3/2}}\exp\left(-\gamma^2\right), \tag{12}$$

$$\tilde{u}^\star = \frac{1}{8\pi\alpha_0^{3/2}\sqrt{\Delta t}}\frac{\operatorname{erfc}(\gamma)}{\gamma}, \tag{13}$$

$$\vec{\nabla}\tilde{u}^\star = \frac{1}{16\pi\alpha_0^2\Delta t}\left[\frac{2}{\sqrt{\pi}}\frac{\exp(-\gamma^2)}{\gamma} + \frac{\operatorname{erfc}(\gamma)}{\gamma^2}\right]\frac{\vec{\xi} - \vec{r}}{|\vec{\xi} - \vec{r}|}. \tag{14}$$

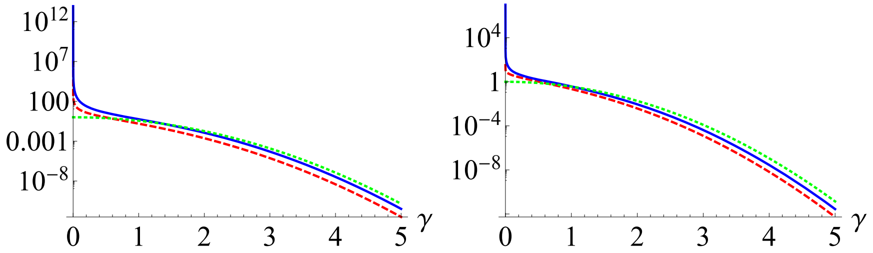

Figure 1: Fundamental solutions and their derivatives versus γ; left: $3D$, right $2D$. Dotted line $u^\star$, dashed line $\tilde{u}^\star$ and solid line $\vec{\nabla}\tilde{u}^\star$.

Similar derivation may be performed in 2D, yielding:

$$u^\star_{2D} = \frac{1}{4\pi\alpha_0\Delta t}\exp(-\gamma^2), \tag{15}$$

$$\tilde{u}^\star_{2D} = \frac{-\mathrm{Ei}(-\gamma^2)}{4\pi\alpha_0}, \tag{16}$$

$$\vec{\nabla}\tilde{u}^\star_{2D} = \frac{1}{4\pi\alpha_0^{3/2}\sqrt{\Delta t}}\frac{\exp\left(-\gamma^2\right)}{\gamma}\frac{\vec{\xi}-\vec{r}}{|\vec{\xi}-\vec{r}|}, \tag{17}$$

where Ei is the exponential integral function.

In equations (12)–(17) the fundamental solutions and their gradients are expressed in terms of the parameter γ. Figure 1 shows this dependence in graphs. We observe, that the values of all functions rapidly decrease as γ increases. At $\gamma = 5$ all are under 10^{-8}.

To get a physical interpretation of the parameter γ let us consider a domain with a characteristic size L. Usually, L is about equal to the size of the domain (for internal problems) or it is smaller then the domain size (for external problems), but it is never significantly larger than the domain size. Thus, we may estimate the largest distance in the domain to be $|\vec{\xi}-\vec{r}| \approx L$.

On the other hand, the characteristic diffusion time is defined as $t_d = \frac{L^2}{\alpha_0}$. In order to have a sufficiently small time step, let us foresee, for example, 100 time steps to reach the characteristic time, i.e. $t_d/\Delta t = 100$. In this case, we can examine γ at $|\vec{\xi}-\vec{r}| = L$, which is

$$\gamma = \sqrt{\frac{(\vec{\xi}-\vec{r})^2}{4\alpha_0\Delta t}} = \sqrt{\frac{L^2}{4\alpha_0\Delta t}} = \sqrt{\frac{t_d}{4\Delta t}} = 5. \tag{18}$$

Thus in a typical setting, we can expect values of γ to reach at least 5 or more. As the integral kernels in equations (12)– (17) are very small for $\gamma >\approx 2$, this means that integrations over $2 \leq \gamma \leq 5$ will make a negligibly small contribution to the final result. This makes kernels (12)–(17) very suitable for compression or sparse approximation.

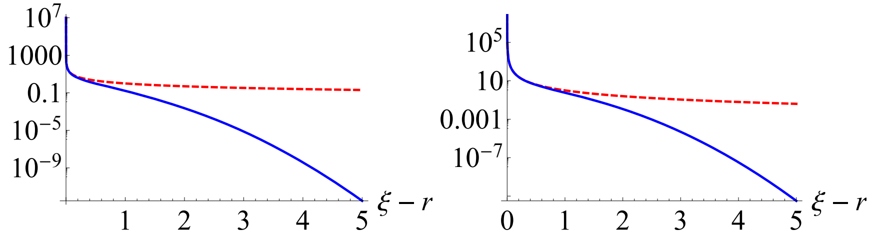

Figure 2: Steady kernels in equations (2) and (3) (dashed) compared to unsteady kernels (13) and (14) (solid). Left: kernels, right: their gradients.

2.3 Comparison

In order to compare the steady kernels (2)–(3) and unsteady kernels (12)–(17) we set $\Delta t = 1$ and $\alpha_0 = 1$ and compare the kernels versus the distance between source and field points. The comparison is shown in Figure 2, where it is evident that although both types of kernels diminish with distance, the unsteady kernels diminish faster.

3 Numerical experiments

In our previous work (Ravnik *et al.* [14, 16, 17]) we have developed and used wavelet compression and sparse approximation techniques aimed at reducing the computational requirements of our BEM based flow solver (Škerget *et al.* [18, 19], Ravnik *et al.* [15]). The methods were aimed at the steady Laplace kernel and the following conclusions were found: the methods are successful in lowering the computational requirements and scale nearly linearly. When solving non-linear fluid dynamics problems, it was found the increasing the non-linearity of the problem results in lower compression ratios in order to keep the simulation results accurate. Unfortunately, low compression ratios do not enable a significant improvement in flow modelling.

In this work, we present a series of tests, where we applied wavelet transform to a unsteady diffusion type problem. In order to convert a steady Poisson type PDE into unsteady one, we use the false transient approach [7, 18], where a false time included in the equation. In this way the equation must be solved in transient manner until a final steady state solution is reached. The tests were carried out on the kinematics equation for the solution of boundary vorticity values (Boit-Savart law). The kinematics equation is

$$\nabla^2 \vec{v} + \vec{\nabla} \times \vec{\omega} = 0, \tag{19}$$

and its false transient counterpart

$$-\frac{1}{\alpha}\frac{\partial \vec{v}}{\partial t} + \nabla^2 \vec{v} + \vec{\nabla} \times \vec{\omega} = 0, \tag{20}$$

where α is a relaxation parameter, $\vec{v}$ fluid velocity and $\vec{\omega}$ fluid vorticity. Steady fundamental solution was used for (19) and the unsteady one for (20). Further details on the implementation of the solution of velocity-vorticity formulation for natural convection may be found in references.

We consider a $2D$ natural convection problem in a closed square cavity. The fluid in the cavity is air and it is put in motion due to buoyancy forces resulting in heating one vertical wall and cooling the second vertical wall. We assume the fluid is incompressible and the material properties are constant and do not depend on temperature. Variations of fluid density are considered only in the buoyancy term through the Boussinesq approximation. The vertical walls are kept at constant temperatures, having a temperature difference of ΔT. The top and bottom walls are adiabatic. No-slip velocity boundary condition is applied on all walls. Non-dimensional units are used, taking the cavity's height as the characteristic length L. In this setting the problem is governed by two nondimensional numbers, the Prandtl number and the Rayleigh number,

$$Pr = \frac{\nu_0}{\alpha_0} = 0.71, \qquad Ra = \frac{g_0 \beta_T \Delta T L^3}{\nu_0 \alpha_0}, \tag{21}$$

where the viscosity of air is ν_0, the thermal diffusivity is α_0, the thermal volume expansion coefficient is β_T and $g_0 = 9.81 m/s^2$.

We have simulated the phenomenon at temperature differences yielding $Ra = 10^4$, $Ra = 10^5$ and $Ra = 10^6$. The computational mesh had 161^2 nodes. The nodes were concentrated towards the corners of the domain using a geometrical series where ratio between sizes of the largest and smallest element was 10.

In Table 1 we compare the heat flux through the vertical walls expressed in terms of the Nusselt number value with benchmark results of other authors. We observe good agreement.

Next, we performed a series of experiments aimed at investigating the relationship between the compression and accuracy of simulations. We used time steps $\Delta t = 10^{-2}$, $\Delta t = 10^{-3}$ and $\Delta t = 10^{-4}$ for calculation of integral kernels in equation (11). We used wavelet transform for matrices of arbitrary size, developed by Ravnik *et al.* [14], to compress the resulting integral matrices. Up to nine different levels of compression were used for each Rayleigh number and time step, amounting to a total of 81 simulations.

Figure 3 displays Nusselt number value versus compression. Compression ratio is defined as the amount of computer storage needed to store the compressed matrix divided by the computer storage needed for the original fully populated matrix. We observe that for all Rayleigh numbers considered, the Nusselt number result is stable up to compression 0.1. For better compression ratio, we observe a dependence of the Nusselt number on the time step. However, all results are within 1% of the benchmark results obtained without compression. Furthermore, since non-linearity of the problem increases with Rayleigh number (higher temperature difference yields stronger buoyancy forces), that the largest differences in Nusselt number value are found at the highest Rayleigh number value.

Table 1: Comparison of Nusselt number with benchmark results, no compression was used.

Ra	Lo *et al.*	de Vahl	Barakos *et al.*	present
	[12]	Davies [5]	[1]	
	31 × 31	81 × 81		161 × 161
10^4	2.244	2.243	2.245	2.2433
10^5	4.521	4.519	4.510	4.5205
10^6	8.823	8.8	8.806	8.8198

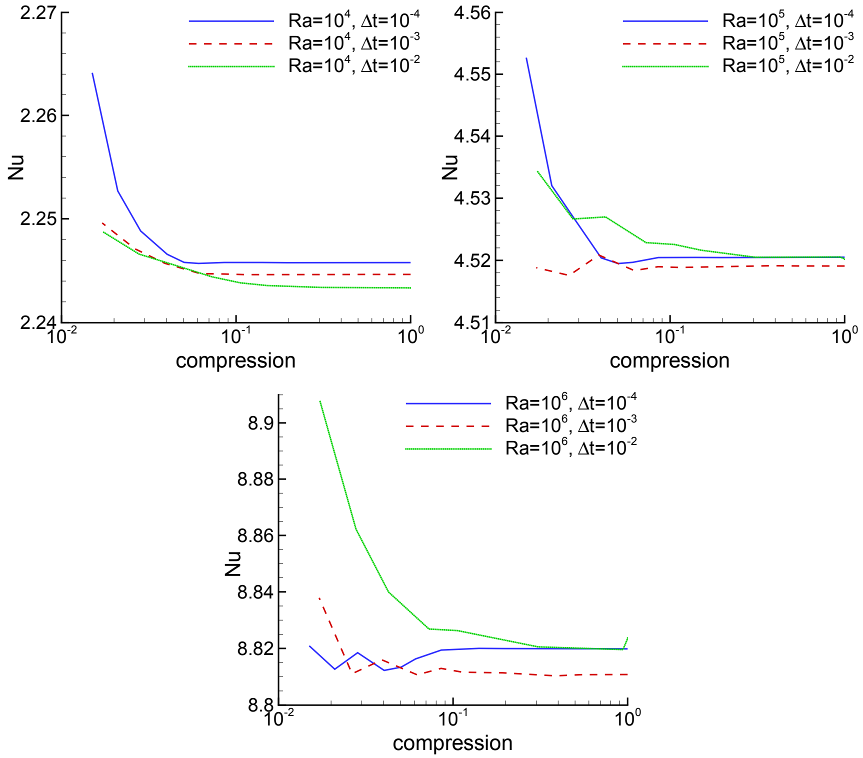

Figure 3: Effect of compression on the Nusselt number value.

Figure 4 displays the CPU time needed for the simulation to reach the steady state result divided by the number of iterations. We observe a big decrease of the CPU time up to compression ratio 0.1. At higher compression, the CPU time still decreases, but the additional gain is small.

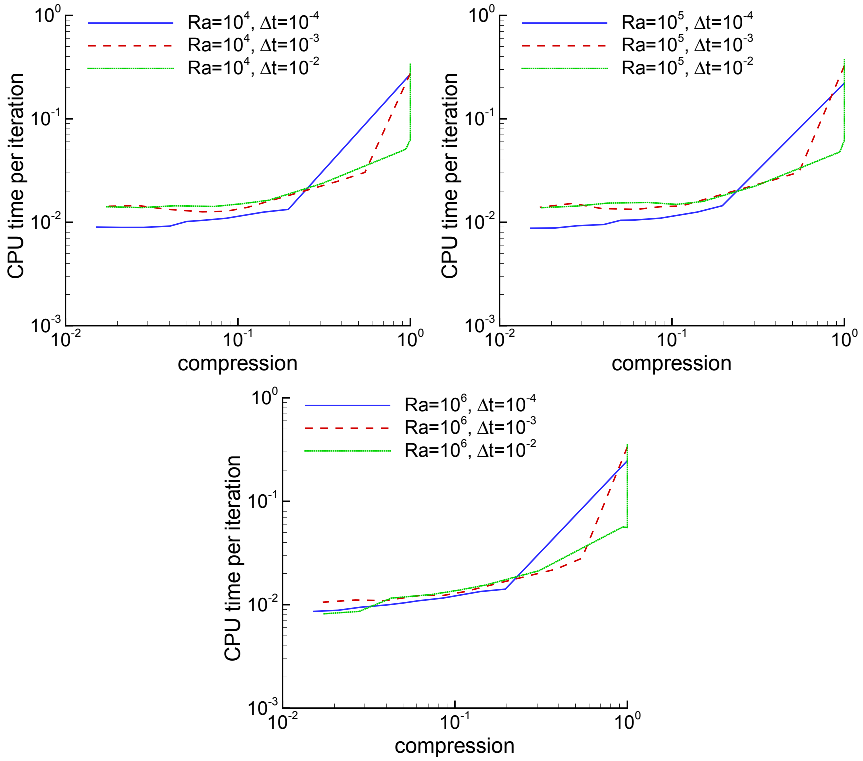

Figure 4: Effect of compression on the CPU time per iteration.

Looking at Nusselt numbers in Figure 3 and at CPU times in Figure 4 we can conclude, that there exists an optimal compression ratio, which maximises the gain in CPU time and at the same time keeps the simulation accuracy high. In the cases studied here, this value is 0.1 compression ratio.

Looking at the wall CPU time to perform simulations (Figure 5) we observe, as expected, a decrease in CPU time with increasing compression. In terms of total CPU time, the simulations that use the shortest time step require the most CPU time. This is due to the fact that short time step needs many steps to reach steady state, this the total CPU time is longer, while the CPU time per iteration (Figure 4) depends solely on the compression ratio and is independent of the time step.

Examining the $Ra = 10^6$, $\Delta t = 10^{-2}$ case, we observe that at highest compression the CPU time is very long (Figure 5) and the simulation accuracy is diminished (Figure 3). In this case the compression had an effect on the non-linear solution procedure, raising the number of iterations needed and influencing accuracy. This happens in the case of the longest time step. A long time step yields small γ values in the domain (see equation (11)) and thus, according to Figure 1,

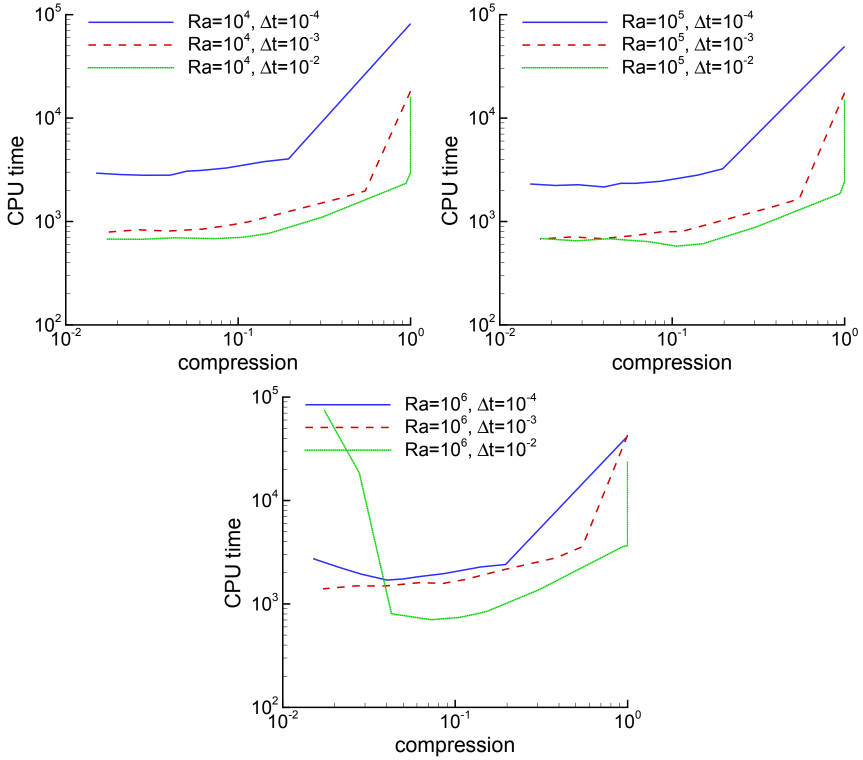

Figure 5: Effect of compression on the CPU time per iteration.

very small integral kernel values are not reached. This results in diminishing accuracy at high compression ratios.

4 Conclusions

We have examined the properties of fundamental solutions and their gradients for unsteady and steady diffusion problems. We have shown, that the unsteady kernel show greater promise for obtaining higher compression when making a sparse approximation of the kernels. The reason for this is the fact, that the unsteady kernels diminish quickly with the distance away from the source point. Thus, compression is successfully, since a large portion of the domain may be neglected, due to an almost negligible contribution to the kernel integral.

The theoretical findings have been confirmed by constructing wavelet transform based sparse approximations in integral kernels in an existing flow and heat transfer solver. We observed a major drop in computer resources needed for simulation. As the problem simulated was non-linear, we have observed the increase in the non-linearity requires lower compression to achieve accurate results.

In the future, we plan to integrate this technique into out in-house boundary element method based laminar and turbulent viscous fluid and heat transfer solver.

References

[1] G. Barakos, E. Mitsoulis, and D. Assimacopoulos. Natural convection flow in a square cavity revisited: laminar and turbulent models with wall function. *Int. J. Numerical Method Fluids*, 18:695–719, 1994.

[2] M. Bebendorf and S. Rjasanow. Adaptive low rank approximation of collocation matrices. *Computing*, 70:1–24, 2003.

[3] H. F. Bucher, L. C. Wrobel, W. J. Mansur, and C. Magluta. A novel approach to applying fast wavelet transforms in boundary element method. *El. J. of Bound. Elements*, BETEQ 2001(2):187–195, 2002.

[4] T. T. Bui, E. T. Ong, B. C. Khoo, E. Klaseboer, and K. C. Hung. A fast algorithm for modeling multiple bubbles dynamics. *J. Comput. Phys.*, 216:430–453, 2006.

[5] G. de Vahl Davies. Natural convection of air in a square cavity: a bench mark numerical solution. *Int. J. Numer. Meth. Fl.*, 3:249–264, 1983.

[6] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[7] G. Guj and F.A. Stella. A velocity–vorticity for the numerical solution of 3d incompressible flows. *Journal of Computational Physics*, 106:286–298, 1993.

[8] N. A. Gumerov and R. Duraiswami. Fast multipole method for the biharmonic equation in three dimensions. *J. Comput. Phys.*, 215:363–383, 2006.

[9] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62:89–108, 1999.

[10] W. Hackbusch and Z. P. Nowak. On the fast multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54:463–491, 1989.

[11] K. Koro and K. Abe. A practical determination strategy of optimal threshold parameter for matrix compression in wavelet BEM. *Int. J. Numer. Meth. Engng.*, 57:169–191, 2003.

[12] D.C. Lo, D.L. Young, K. Murugesan, C.C. Tsai, and M.H. Gou. Velocity-vorticity formulation for 3D natural convection in an inclined cavity by DQ method. *Int. J. Heat Mass Transfer*, 50:479–491, 2007.

[13] V. Popov, H. Power, and S. P. Walker. Numerical comparison between two possible multipole alternatives for the BEM solution of 3D elasticity problems based upon Taylor series expansions. *Eng. Anal. Bound. Elem.*, 27:521–531, 2003.

[14] J. Ravnik, L. Škerget, and M. Hriberšek. The wavelet transform for BEM computational fluid dynamics. *Eng. Anal. Bound. Elem.*, 28:1303–1314, 2004.

[15] J. Ravnik, L. Škerget, and Z. Žunič. Velocity-vorticity formulation for 3D natural convection in an inclined enclosure by BEM. *Int. J. Heat Mass Transfer*, 51:4517–4527, 2008.

[16] J. Ravnik, L. Škerget, and Z. Žunič. Comparison between wavelet and fast multipole data sparse approximations for Poisson and kinematics boundary – domain integral equations. *Comput. Meth. Appl. Mech. Engrg.*, 198:1473–1485, 2009.

[17] J. Ravnik, L. Škerget, and Z. Žunič. Fast single domain–subdomain BEM algorithm for 3D incompressible fluid flow and heat transfer. *Int. J. Numer. Meth. Engng.*, 77:1627–1645, 2009.

[18] L. Škerget, M. Hriberšek, and G. Kuhn. Computational fluid dynamics by boundary domain integral method. *Int. J. Num. Meth. Eng.*, 46:1291–1311, 1999.

[19] L. Škerget, M. Hriberšek, and Z. Žunič. Natural convection flows in complex cavities by BEM. *Int. J. Num. Meth. Heat & Fluid Fl.*, 13:720–735, 2003.

[20] L. C. Wrobel. *The Boundary Element Method*. John Willey & Sons, Ltd, 2002.

[21] J. Xiao, J. Tausch, and Y. Hu. A-posteriori compression of wavelet-BEM matrices. *Computational Mechanics*, 44:705–715, 2009.