

LOAD BALANCING ANGULAR ADAPTIVITY ON ENERGY DEPENDENT REACTOR PROBLEMS

**Ioannis Nikiteas¹, Steven Dargaville¹, Paul N. Smith^{1,2},
Richard P. Smedley-Stevenson^{1,3}, Christopher C. Pain¹**

¹Applied Modelling and Computation Group, Imperial College London
London SW7 2AZ, UK

²ANSWERS Software Service, Wood PLC Kimmeridge House,
Dorset Green Technology Park, Winfrith Newburgh, Dorchester, Dorset, DT2 8ZB

³AWE, Aldermaston, Reading, RG7 4PR, UK
ioannis.nikiteas17@imperial.ac.uk

ABSTRACT

This paper presents a novel parallel algorithm capable of solving problems with multiple energy groups, such as reactor core problems, using energy dependent angular h -adaptivity coupled with Dynamic Load Balancing (DLB). We use the C5G7 NEA benchmark as an example with our in-house code FETCH2 and compare runtimes and Continuous Degrees of Freedom (CDOFs) between energy independent angular adapting runs with and without DLB. Our results demonstrate a 32% reduction in CDOFs and a 38% reduction in runtime in favour of the energy dependent angular adaptivity runs with DLB when compared to energy independent angular adaptivity.

KEYWORDS: dynamic load balancing, angular adaptivity, reactor physics

1. INTRODUCTION

Radiation transport phenomena are usually modelled using the Boltzmann Transport Equation (BTE). Computing the BTE can be done using either stochastic or deterministic computational methods. The work presented in this paper focuses on a deterministic approach, which encompasses an angular discretisation with Haar wavelets capable of scalable, angular h -adaptivity on unstructured finite element grids.

Real life applications of computational modelling for radiation phenomena include radiation shielding problems or nuclear reactor simulations which can be challenging to compute due to the large problem size, the complicated underlying geometries and the orders of magnitude differences in the modelled quantities, e.g. neutron energies. To combat large problem sizes, modelling codes can be designed to use multiple processors in order to increase the total computational power. Unstructured meshes can be used to aid with the modelling of irregular and complicated geometries, while adaptivity can help resolve quantities that are greatly varying through the computational phase space.

Combining in a code parallel computing, unstructured meshes, and adaptivity can be challenging, especially if such implementation needs to be performant. Adaptivity in parallel in particular can hinder the performance of a code since its main functionality is to add or remove Degrees of Freedom (DOFs) from the underlying mesh, resulting into mesh regions with greatly varying computational work; thus creating a performance bottleneck. To reduce the computational imbalances across the mesh, a Dynamic Load Balancing (DLB) algorithm can be used. The algorithm aims to minimise the work imbalances across the computational domain by redistributing partitions of the mesh at each processor based on a set of weights representative of the local work at each mesh node.

Even though the implementation of a DLB algorithm on unstructured adapting meshes can greatly improve parallel performance and runtimes, as shown in our previous work [1], the communication costs associated with the migration of mesh elements during a DLB can be equally costly as the computational work.

The aim of this work is to demonstrate how energy dependant angular adaptivity can be used to provide a more robust and accurate solution in reactor physics problems, using a parallelly scalable approach. We use our code FETCH2, developed by the AMCG in Imperial College London to perform DLB on energy dependent angularly adapting discretisations for a reactor physics benchmark and measure the parallel performance and k_{eff} against the reference value.

2. THEORETICAL BACKGROUND

2.1. Angular adaptivity: Haar wavelets

A brief introduction to angular adaptivity with Haar wavelets is provided in this section, but for a complete review we direct the reader to our previous work [2].

Haar wavelets HW_N are hierarchical, with local support and form a non-rotationally invariant angular discretisation. To reduce the discretisation error, or capture highly anisotropic fluxes one can increase the number of wavelets N , thus creating a finer angular discretisation. However, uniformly refining the angular discretisation is computationally unfavourable, hence performing localised refinement is preferred. Localised refinement or coarsening of the angular discretisation is known as angular adaptivity. In general, adaptive methods aim to be implemented in automated and iterative algorithms, without any *a priori* knowledge of where refinement should occur. Therefore to guide adaptivity a quantity calculated *a posteriori* is needed, referred to as an error metric.

Due to the choice of Haar wavelets for our angular discretisation, we can form an error metric that utilises the local support and hierarchical structure of our discretisation, but also wavelet specific properties such as the norm-equivalence, wavelet cancellation and thresholding. As a result the metric is capable of refining in areas where the angular flux is large or discontinuous. In more detail, the error metric is defined using the problem solution ψ that has a residual R and an error ϵ , with the exact solution having a residual equal to $R(\psi_{\text{exact}}) = 0$ and error $\epsilon = \psi_{\text{exact}} - \psi$. A threshold τ can be used to form an approximation e to the exact error given by $e = \epsilon \approx |\psi|/\tau$.

When solving a reactor core problem, particle energies can vary greatly and hence, the optimal angular discretisations between energy groups is likely to be different. More energetic particles,

such as fast neutrons, will require fine angular discretisations to resolve streaming effects, while lower energy particles such as thermal neutrons, can utilise coarser and more uniform angular discretisations due to their diffusive behaviour. Consequently, to decrease computational costs for the already computationally demanding calculations of reactor cores an energy dependent angularly adapting discretisation scheme can be employed.

There have only been two references in the literature where energy dependent angular adaptivity has been used [3, 4]. Both of them by Goffin et al. from the AMCG in Imperial College London. The first implementation used spherical harmonics while the latter used a linear octahedral wavelets to perform goal-based angular adaptivity. Both implementations showed a reduction in the total number of DOFs, with the linear wavelets also showcasing a speed-up of 2 between the energy dependent and the energy independent angular discretisations. Although the results presented were encouraging, the problems modelled were relatively simple, with only two energy groups and no parallel processing or DLB was used.

2.2. Dynamic load balancing

As previously mentioned a DLB algorithm aims to ensure equal distribution of computational work across all processors. This requires for the initial problem to be decomposed in partitions which are then assigned to each processor. There are multiple ways a computational domain decomposition can be performed; our implementation is based on a spatial mesh decomposition. The specific algorithm used to decompose the spatial mesh and drive the core of the DLB algorithm is chosen to be the graph partitioner ParMETIS [5] due to its short partitioning time and high mesh quality.

The DLB algorithm operates using a series of weights to represent the amount of computational work present at each spatial node. The first step in the DLB algorithm is to determine whether load balancing the spatial mesh is necessary hence, the work imbalance needs to be calculated. The computational imbalance is estimated by taking the ratio between the sum of all the angular basis functions in a partition, over the average number of angular basis functions across all partitions. This calculation yields the work imbalance η^* , which is a relative measure of the local partition work compared against the average work in the problem.

If the work imbalance η^* is greater than a user supplied imbalance tolerance η , then the mesh will be repartitioned. For the mesh to be accurately balanced between partitions, work estimates for each spatial node are required. We calculate the amount of work per node by taking the ratio of the number of adapted angular basis functions at the node, normalised over the largest number of angular basis functions across the entire computational domain. The weights are then supplied to ParMETIS which returns a new spatial mesh decomposition.

The DLB algorithm checks for computational imbalances at the inner most loop of our power iteration as it can be seen from Algorithm 1, with the purpose of creating optimally balanced partitions for each Gauss-Seidel sweep through the energies. Performing the repartitioning at every Gauss-Seidel sweep will inadvertently result into increased communications, which for multienergy problems will be prohibitively expensive. To minimise unnecessary communications similar energy groups can be agglomerated into *group sets* thus, sharing their angular discretisation and reducing communication costs.

Algorithm 1: Dynamic Load Balancing algorithm of eigenvalue problems with energy dependent angular adaptivity.

```

do
  /* Start power solve */
  while keff error ≤ keff relative tolerance do
    /* Perform a Gauss-Seidel for all energy groups */
    foreach group set do
      /* Energy groups in a group set share the same angular expansion */
      if work imbalance ≥ imbalance tolerance then
        Compute work weights for spatial mesh;
        Dynamic Load Balance the spatial mesh;
      end
      for energy group in group set do
        Perform a linear solve; /* matrix-free FGMRES(30) */
      end
    end
  end
end
foreach group set do
  Compute angular error metric;
  Adapt angular expansion;
end
while adapt step ≤ final adapt step;

```

Worth noting is the type of iterative method used, a matrix-free FGMRES(30) configured with a multigrid preconditioner. The spatial discretisation is a sub-grid scale finite element discretisation. We redirect the reader to [2, 6] for more details.

3. RESULTS

The algorithm was tested against the C5G7 NEA reactor physics benchmark [7], which has 7 energy groups, using the following set-up parameters. For the spatial mesh, we used an unstructured triangular mesh, with 0.15 element spacing in the fuel assembly regions and reflect boundaries, and 2.0 element spacing for the moderator region, resulting to a final mesh of 350,000 elements. The mesh can be seen in more detail in Figure 1. The adaptivity algorithm was allowed to adapt up to 4 orders (HW_4), using an initial angular discretisation of 4 Continuous DOFs (CDOFs) per spatial node, per energy group (HW_1 which is equivalent to S_2). The error metric tolerance was set to $\tau = 1 \times 10^{-3}$, which if adapted uniformly corresponds to 256 angular basis functions per spatial node, per energy group. Finally for the iterative methods, we solved the power iteration to a relative tolerance of 1×10^{-5} as per the benchmark specifications. Every power iteration included 11 linear solves; four of which were due to upscattering effects from energy groups 4 to 7. All linear solves were resolved to a relative tolerance of 1×10^{-10} . The results were obtained using ARCHER, a Cray XC30 and UK’s Tier 1 Supercomputing service, with 24 cores per compute node. All the results presented where run using 5 nodes (120 cores).

The first important step is to determine whether using a using a energy dependent angular discretisation is of any benefit and if DLB has any positive impact in the overall performance. We compare two runs with energy dependent and independent angular discretisations. The energy dependent

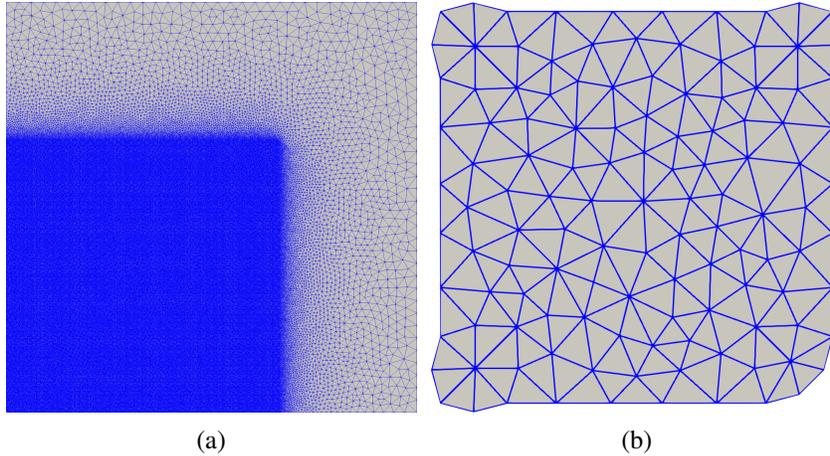


Figure 1: (a) Full mesh showcasing our varying element size from fuel assemblies to the moderator. (b) Unstructured mesh region near a fuel pin.

discretisation, uses 2 group sets, one for fast neutrons, including the first 3 energy groups and one for the thermal neutrons, containing the remaining 4 energy groups. As angular adaptivity occurs, the total problem size is observed to decrease, as shown in Table 1, yielding a reduction of 32% in the number of CDOFs. The k_{eff} is added in the table as an indicator that the adaptivity algorithm converges to the true solution of the benchmark.

Adapt order	Energy independent		Energy dependent	
	Av. CDOFs per spatial node	k_{eff}	Av. CDOFs per spatial node	k_{eff}
1	28	1.20568	28	1.20568
2	111	1.19081	111	1.19085
3	278	1.18774	224	1.18845
4	669	1.18695	451	1.18682

Table 1: Comparison of average number of Continuous (CDOFs) per adapt step with the use of energy dependent angular discretisations for all 7 energy groups, using a reference value 1.18655. The energy dependent angular discretisation results in 32% less the DOFs.

Furthermore, Figures 2b and 2c serve as a depiction of where in the spatial mesh angular adaptivity occurred, when using energy dependent angular discretisations. An expected but noteworthy result are the spatial nodes where the energy independent angular discretisation, shown in Figure 2a, has adapted. For the most part the adapted nodes appear to be just the union of the spatial nodes where the energy dependent angular discretisations adapted.

In addition to the reduction of DOFs, the overall performance of energy dependent angular discretisations with DLB was examined. Four different configurations were tested; an energy independent angular adaptivity scheme with/out DLB and the previously discussed energy dependent scheme

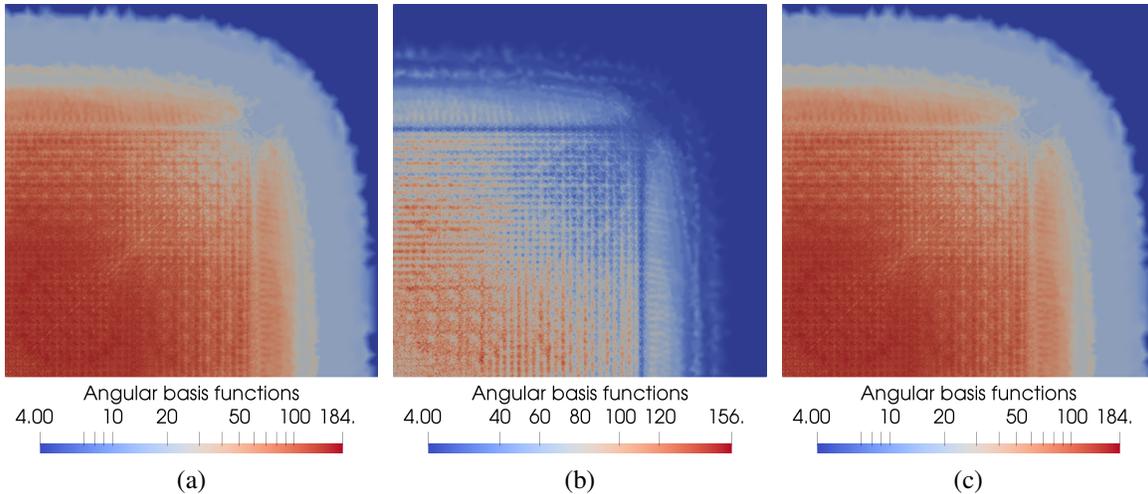


Figure 2: Plots indicating the number of angular basis functions present across the spatial mesh, normalised to one energy group. (a) Energy independent angular adaptivity. (b-c) Energy dependent angular adaptivity split across 2 energy group sets (b) “Fast” energy groups (c) “Thermal” energy groups

with/out DLB. The results are shown in Figure 3 and indicate a 29% reduction in runtime by introducing energy dependent angular discretisations and a 38% improvement in runtime when using energy dependent angular discretisations with DLB compared to the energy independent case, a 9% improvement.

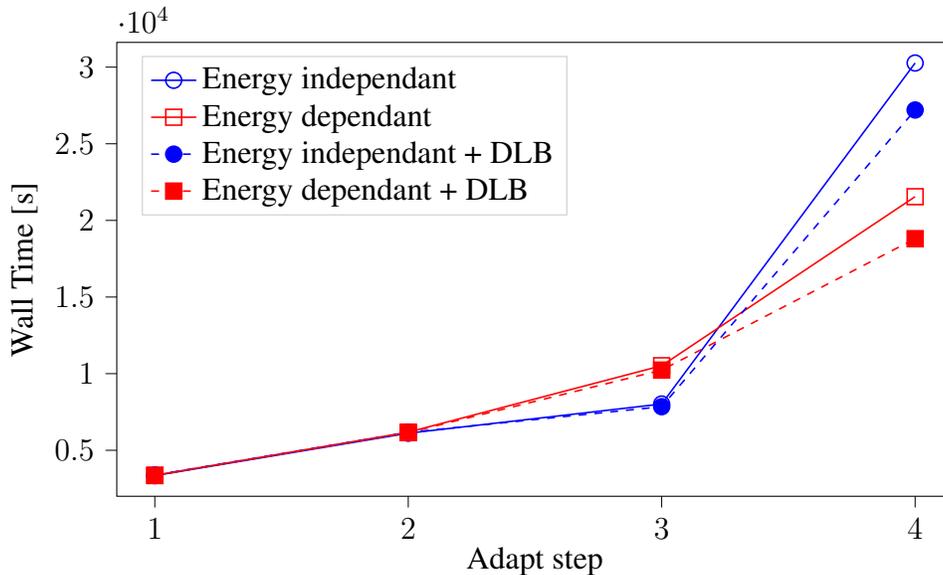


Figure 3: Comparison of runtimes for energy dependent angular discretisation with angular adaptivity and dynamic load balancing (DLB). The energy dependent, load balanced run resulted to 38% faster runtime.

Until now all the presented results had a fixed load imbalance tolerance of $\eta = 1.25$, which implies work imbalances between partitions of up to 25% are tolerable. However, that value is not

necessarily the optimal for the given problem, hence a short sensitivity analysis is performed to determine whether, an optimal imbalance tolerance exists. The results are listed in Table 2.

Similarly to our previous work [1], our DLB algorithm appears to be insensitive to the value of the work imbalance tolerance as shown in Table 2. With the exception of $\eta = 1.05$, where at the fourth adapt order, imbalances due to heavily adapted regions start to grow, resulting in an increase in the number of DLBs and thus a deterioration in the final runtime.

		Tolerance	Adapt order			
			1	2	3	4
Wall Time [s]	$\eta = 1.05$	3310.2	6335.9	7835.0	25344.0	
	$\eta = 1.25$	3431.2	6291.2	10363.4	18808.1	
	$\eta = 1.50$	3362.3	6177.0	10159.6	19229.2	
	$\eta = 2.50$	3425.3	6307.3	10691.2	21759.9	
Number of Dynamic Load Balances per adapt	$\eta = 1.05$	0	4	7	124	
	$\eta = 1.25$	0	0	17	14	
	$\eta = 1.50$	0	0	1	0	
	$\eta = 2.50$	0	0	0	0	
% Time of Dynamic Load Balancing	$\eta = 1.05$	0.083	0.086	0.166	1.159	
	$\eta = 1.25$	0.057	0.056	0.224	0.271	
	$\eta = 1.50$	0.058	0.056	0.053	0.037	
	$\eta = 2.50$	0.056	0.056	0.051	0.047	

Table 2: Sensitivity analysis for imbalance tolerance. Comparison of the effect of various values of imbalance tolerances on the runtime, DLB time and DLB occurrences as a function of adapt order.

Based on the previous testing the most favourable configuration for the C5G7 benchmark uses: energy dependent angular adaptivity with two *group sets* and DLB with $\eta = 1.25$. Using the above configuration we performed a strong scaling study to examine the its scalability. The results from the strong scaling study are listed in Table 3. A decrease in scaling performance is observed by 50% when the number of cores increases by $\times 8$. The reason for performance decrease has been identified in our previous work [1] to be due to increased communication costs arising from the creation of very small spatial mesh partitions.

4. CONCLUSIONS

In this work we present a parallel algorithm suitable for performing heterogeneous reactor core calculations on unstructured meshes by using energy dependent angular adaptivity and DLB, thus resulting in improved runtimes. Our results indicated that is it computationally favourable to use energy dependent angular discretisations, yielding a 29% reduction in runtimes. Applying DLB to

Number of Processors:	120	240	480	960
Strong Scaling Performance:	100%	80.4%	59.3%	50.8%

Table 3: Strong scaling performance of FETCH2 with energy dependent angular adapting discretisations and DLB.

the run, further decreased the runtime by 32% and the CDOFs to 38% when compared to energy independent angular adaptivity runs. Moreover, our strong scaling performance was measured to be around 50% for a $\times 8$ multiplier in the number of processors. A reasonable result given the limitations that we identified in our previous work about adapting one mesh (the angular), but DLB another (the spatial) [1]. It should be noted that the spatial mesh used for the benchmark due to its varying element size in the fuel and moderator regions was balanced *a priori*, effectively reducing the formation of large, disproportionate mesh partitions that were observed in [1]. Overall, we believe that this work showcases the potential benefits of using angular adaptivity in large physical problems such as reactor core calculations. Future work will include combining this work with improved error metric calculations in order to perform space-angle adaptivity on large reactor physics problems.

REFERENCES

- [1] I. Nikiteas, S. Dargaville, C. Pain, P. N. Smith, and R. P. Smedley-Stevenson. “Impact of Load Balancing on Parallel Performance with Haar Wavelets Angular Adaptivity.” In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, volume 1, pp. 2542–2551. Portland, OR (2019).
- [2] S. Dargaville, A. G. Buchan, R. P. Smedley-Stevenson, P. N. Smith, and C. C. Pain. “Scalable angular adaptivity for Boltzmann transport.” *Journal of Computational Physics*, **volume 406**, pp. 109–124 (2020).
- [3] M. A. Goffin, A. G. Buchan, A. C. Belme, C. C. Pain, M. D. Eaton, P. N. Smith, and R. P. Smedley-Stevenson. “Goal-based angular adaptivity applied to the spherical harmonics discretisation of the neutral particle transport equation.” *Annals of Nuclear Energy*, **volume 71**, pp. 60–80 (2014).
- [4] M. A. Goffin, A. G. Buchan, S. Dargaville, C. C. Pain, P. N. Smith, and R. P. Smedley-Stevenson. “Goal-based angular adaptivity applied to a wavelet-based discretisation of the neutral particle transport equation.” *Journal of Computational Physics*, **volume 281**, pp. 1032–1062 (2015).
- [5] G. Karypis and V. Kumar. “Multilevel Graph Partitioning Schemes.” *[ICPP’95] International Conference on Parallel Processing*, **volume 3**, pp. 113–122 (1995).
- [6] A. G. Buchan, A. S. Candy, S. R. Merton, C. C. Pain, J. I. Hadi, M. D. Eaton, A. J. H. Goddard, R. P. Smedley-Stevenson, and G. J. Pearce. “The Inner-Element Subgrid Scale Finite Element Method for the Boltzmann Transport Equation.” *Nuclear Science and Engineering*, **volume 164(2)**, pp. 105–121 (2015).
- [7] Nuclear Energy Agency. “Benchmark on Deterministic Transport Calculations Without Spatial Homogenisation A 2-D/3-D MOX Fuel Assembly Benchmark.” Technical report (2003).