Math. Model. Nat. Phenom.
Vol. X, No. X, 20XX, pp. X-X

From h to p efficiently: selecting the optimal spectral/hp discretisation in three dimensions

C. D. Cantwell[a 1], **S. J. Sherwin**[b], **R. M. Kirby**[c] **and P. H. J. Kelly**[d]

[a] Department of Mathematics, Imperial College London, London, SW7 2AZ, UK
[b] Department of Aeronautics, Imperial College London, London, SW7 2AZ, UK
[c] School of Computing, University of Utah, Salt Lake City, Utah, USA
[d] Department of Computing, Imperial College London, London, SW7 2AZ, UK

Abstract. There is a growing interest in high-order finite and spectral/hp element methods using continuous and discontinuous Galerkin formulations. In this paper we investigate the effect of h- and P-type refinement on the relationship between runtime performance and solution accuracy. The broad spectrum of possible domain discretisations makes establishing a performance-optimal selection non-trivial. Through comparing the runtime of different implementations for evaluating operators over the space of discretisations with a desired solution tolerance, we demonstrate how the optimal discretisation and operator implementation may be selected for a specified problem. Furthermore, this demonstrates the need for codes to support both low- and high-order discretisations.

Key words: spectral/hp element, optimisation, code performance
AMS subject classification:

1 Introduction

Spectral/hp element methods have become a mainstream technique for the solution of partial differential equations (PDEs). Their ability to handle complex geometries, provide localised refinement in regions of high solution gradient and attain convergence at exponential rates makes them a valuable tool in a wide variety of applications across academia and industry. These originally included problems in fluid dynamics [11], but spectral/hp element methods are now used in many

[1] Corresponding author. E-mail: c.cantwell@imperial.ac.uk

other fields such as studies of electromagnetics [6], shallow-water problems [1] and structural dynamics [9].

Spectral/hp element methods can be considered as a high-order extension of the traditional low-order finite element methods. They present greater scope for achieving fast convergence of a solution by both refining the mesh (h-refinement) and increasing the order of the polynomial expansions (P-refinement) used to represent the solution. Global spectral techniques can be considered as the limit of P-refinement on a single element. The notion of high order is understood differently depending on the context. Those practising h-type refinement consider high-order expansions to be anything up to fifth-order [7, 16] while at the other end of the spectrum, the global spectral community would consider expansion orders in the region of 100 to be relatively modest [5]. In contrast, the spectral/hp element community [8, 3] would place high order in the region of 15th-order. Spectral/hp discretisations offer a broad performance-tuning capacity, not only through choice of element shape, mesh refinement and expansion order, but through a choice of different implementations for evaluating operators. In a mathematical context, operator evaluation refers to a matrix-vector product, but numerically we may evaluate this operation using a single global matrix, a series of local elemental matrices, or a sum-factorisation approach [10] through exploitation of the tensorial nature of the elemental expansion basis. The resulting size of the parameter space for discretisation and operator implementation is the primary motivation of this work.

Cantwell *et al.* [2] examined the runtime performance of a spectral/hp implementation in actioning four fundamental operators in three dimensions (including the backward transform and evaluation of the Helmholtz linear differential operator). They found that choosing the correct implementation strategy for a given discretisation was essential when evaluating the different operators. This choice also affects solver performance considerably when using iterative solvers and the choice of discretisation controls the solution accuracy. In solving a given PDE using a spectral/hp element framework, one typically strives to achieve a required level of accuracy in the solution. Therefore, one must use a suitable (h, P)-discretisation which achieves this. Furthermore, one would like to minimise the runtime cost of the calculation. For each (h, P)-discretisation there exists an optimal evaluation strategy for each mathematical operator [2] and so we therefore seek the (h, P)-discretisation, from the space of those which provide acceptable accuracy, which minimises the runtime cost when using the optimal implementation. In practice, other factors may influence the selection such as memory limitations, but we discount these for the purposes of this study.

In this paper we establish how one can select the optimal discretisation for solving an elliptic problem, the Helmholtz equation, to a given threshold of accuracy. To achieve this we minimise the runtime cost of evaluating the Helmholtz operator through variation of the discretisation and choice of evaluation strategy across the space of (h, P)-discretisation. We stress that in this study we optimise only the low-level evaluation of the Helmholtz operator required as a building block for an iterative solver and not the complete iterative solution of the linear system. The choice of solver and preconditioner would add an extra layer of complexity onto the analysis. The paper is organised as follows. In Sec. 2 we summarise the spectral/hp element formulation and describe the three operator implementation strategies which may be adopted for a particular discretisation.

We also describe the example problem used to demonstrate the optimal discretisation selection. In Sec. 3 we investigate how the optimal combination of mesh size and expansion order may be selected to solve the example problem to a prescribed level of accuracy in minimal runtime. Finally, we discuss the consequences and limitations of the work in Sec. 4.

2 Methods

2.1 Spectral/hp element framework

The spectral/hp element framework and the tensorial construction of hexahedral and tetrahedral elements is well-established and is documented extensively elsewhere ([8, 11, 12, 13]). Therefore, we only give a brief summary of the formulation here.

A potentially geometrically complex domain is partitioned into a tessellation of elemental subdomains. In a three-dimensional setting we consider an element Ω_e to be either hexahedral or tetrahedral in shape, although prisms and square-based pyramids may also be used. For each Ω_e, a mapping $\chi_e : \Omega_e \rightarrow \Omega_{st}$ maps points in the physical elemental subdomains onto the corresponding shape in a reference space, Ω_{st}. The reference-space hexahedral and tetrahedral regions are defined as

$$\begin{aligned}\mathcal{Q}^3(\xi) &= \left\{(\xi_1, \xi_2, \xi_3) \in [-1, 1]^3\right\},\\ \mathcal{T}^3(\xi) &= \left\{(\xi_1, \xi_2, \xi_3) | -1 \leq \xi_i, i = 1, 2, 3; \xi_1 + \xi_2 + \xi_3 \leq -1\right\},\end{aligned}$$

respectively. For the latter, a coordinate transform [4, 12] leads to a representation of the tetrahedral geometry with fixed limits as

$$\mathcal{T}^3(\eta) = \left\{(\eta_1, \eta_2, \eta_3) \in [-1, 1]^3\right\}.$$

On each reference element, the solution for a given element is expressed in terms of a fixed basis of three-dimensional functions, $\{\phi_n\}$, constructed as a tensor product of P one-dimensional polynomials $\{\psi_p\}$. This leads to the following expansions on the reference elements

$$\mathcal{Q}^3 : \quad u(\xi) = \sum_n \phi_n \hat{u}_n = \sum_p \sum_q \sum_r \psi_p(\xi_1)\psi_q(\xi_2)\psi_r(\xi_3)\hat{u}_{pqr}, \tag{2.1}$$

$$\mathcal{T}^3 : \quad u(\eta) = \sum_n \phi_n \hat{u}_n = \sum_p \sum_q \sum_r \psi_p(\eta_1)\psi_{pq}(\eta_2)\psi_{pqr}(\eta_3)\hat{u}_{pqr}. \tag{2.2}$$

The nature of this construction allows for the sum-factorisation evaluation strategy [10] to be employed, even in the tetrahedral region, which leads to a higher performance in some circumstances [2]. However, to ensure only the minimum necessary degrees of freedom are used in the tetrahedral case (an extension to three dimensions of the triangular case shown in Fig. 1) an interdependency between the one-dimensional modes in the second and third coordinate directions is introduced. This negatively affects the performance of the sum-factorisation strategy in tetrahedra when compared to hexahedra.

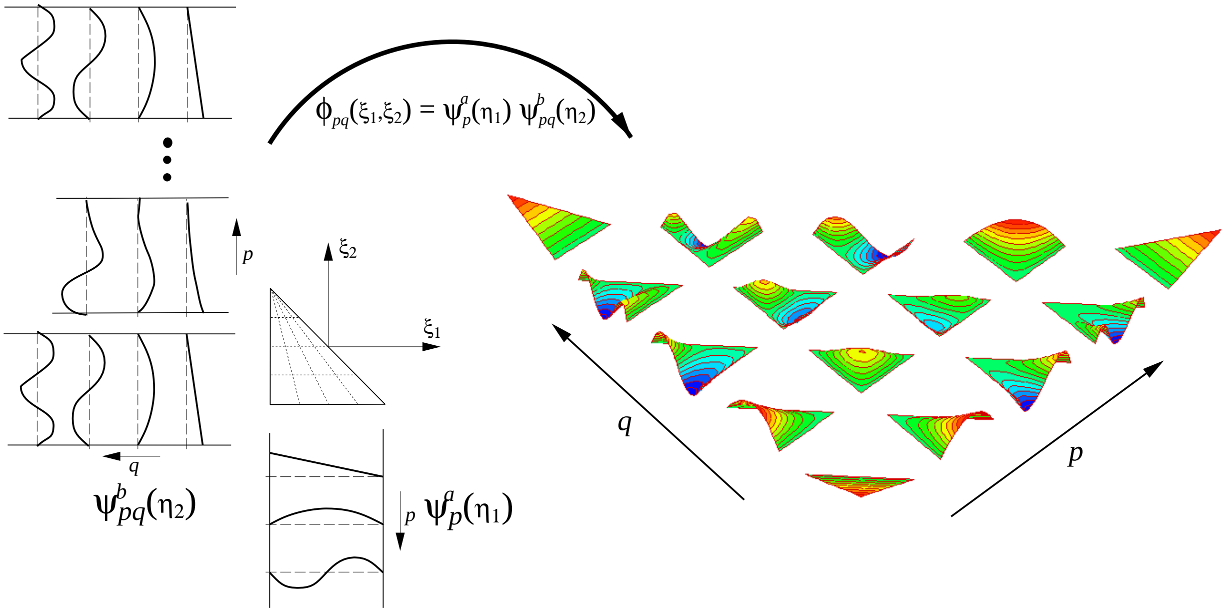
$\phi_{pq}(\xi_1,\xi_2) = \psi^a_p(\eta_1)\, \psi^b_{pq}(\eta_2)$
p
ξ_2
ξ_1
q
$\psi^b_{pq}(\eta_2)$
p
$\psi^a_p(\eta_1)$
q
p

Figure 1: Two-dimensional triangular modes $\phi_{pq}(\xi_1, \xi_2)$ are constructed as a tensor product of one-dimensional modes $\psi_p(\eta_1)$ and $\psi_{pq}(\eta_2)$.

In a continuous Galerkin spectral/hp element formulation the C^0-continuity requirement is imposed across elemental boundaries to enforce the required connectivity. This leads to a global formulation of the problem in which each of the elemental modes are considered as global functions and assigned a global numbering. An assembly matrix $\mathcal{A}$ maps this global mode numbering onto the corresponding local elemental mode numberings,

$$\hat{\mathbf{u}}^l = \mathcal{A}\hat{\mathbf{u}}^g.$$

This is simplified if the polynomial functions used in the construction of the basis support a boundary/interior decomposition. In this case all elemental interior modes are zero on the element boundaries, and consequently interior modes on different elements are globally orthogonal. This leads to a global matrix system that exposes significant structure which may be exploited using substructuring techniques [14] to improve the performance of global operations.

We conclude the formulation with an overview of the implementation strategies with which we may evaluate numerical operators. As shown in [2] for the three-dimensional case, and in [15] in two dimensions, careful selection of the implementation strategy is essential to achieve the best performance at a given (h, P)-discretisation. As well as global operations and local elemental operations the tensor-product construction of the elemental basis modes allows for a third method of implementation known as sum-factorisation [10]. Therefore, the three methods we consider for evaluating operators are

- Global matrix evaluation,

- Local elemental matrix evaluation,

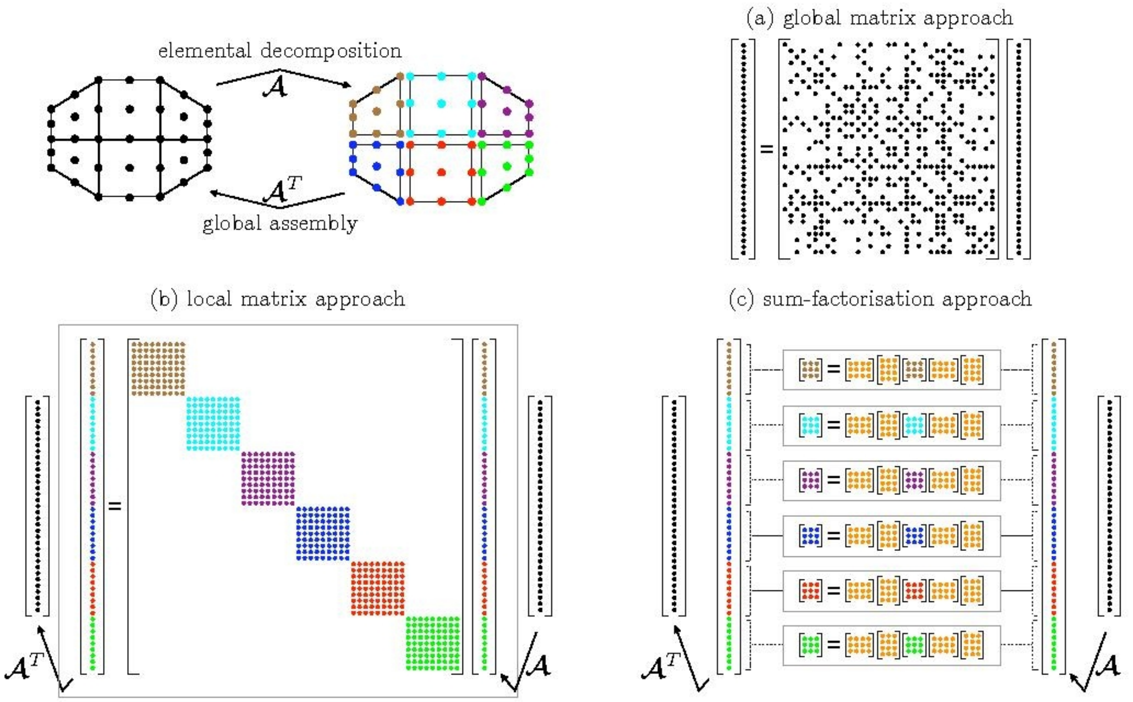

Figure 2: Diagram of the three evaluation strategies: global matrix operation (a), local elemental operations (b) and sum-factorisation (c).

- Local elemental sum-factorisation.

A diagrammatic representation of these is given in Fig. 2 showing the global approach (Fig. 2(a)) and the two elemental approaches (Fig. 2(b) and (c)), as well as the elemental decomposition of the domain.

The global approach evaluates all degrees of freedom in the domain simultaneously through the use of a global matrix system. The local strategies perform the evaluation at the elemental level. The sum-factorisation approach factorises the fully three-dimensional $\mathcal{O}(P^6)$ elemental operation as a series of three one-dimensional $\mathcal{O}(P^4)$ operations. This provides significantly improved performance with some discretisations [2], reducing both memory storage and computational cost. This is not always the case, and with some linear differential operators with certain discretisations it may provide a significantly lower performance than the local matrix or global strategies. To illustrate the sum-factorisation implementation, consider the backward transform operation for the hexahedral region in Eqn. 2.2. This is factorised as

$$u(\xi_{1i}, \xi_{2j}, \xi_{3k}) = \sum_{p=0}^{P} \psi_p(\xi_{1i}) \left\{ \sum_{q=0}^{P} \psi_q(\xi_{2j}) \left\{ \sum_{r=0}^{P} \hat{u}_{pqr} \psi_r(\xi_{3k}) \right\} \right\}. \tag{2.3}$$

In matrix form, and following the notation described in [2], the three-dimensional backward transform,

$$\mathbf{u} = \mathbf{B}\hat{\mathbf{u}} \equiv (\mathbf{B}_0 \otimes \mathbf{B}_1 \otimes \mathbf{B}_2)\hat{\mathbf{u}},$$

a tensor product of the one-dimensional backward transforms $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_2$, may be expressed as a sequence of three matrix-matrix multiplications,

$$\begin{aligned}
\mathbf{q}_0^{P_1P_2;Q_0} &= \left[\hat{\mathbf{u}}^{P_0;P_1P_2}\right]^\top \mathbf{B}_0^\top, \\
\mathbf{q}_1^{P_2Q_0;Q_1} &= \left[\mathbf{q}_0^{P_1;P_2Q_0}\right]^\top \mathbf{B}_1^\top, \\
\mathbf{u}^{Q_0Q_1;Q_2} &= \left[\mathbf{q}_1^{P_2;Q_0Q_1}\right]^\top \mathbf{B}_2^\top, \\
\mathbf{u} &= \mathbf{u}^{Q_0Q_1Q_2;1}.
\end{aligned}$$

Here, a vector $\hat{\mathbf{u}}^{P_0;P_1P_2}$ denotes an $P_0P_1P_2$-length column vector arranged as a matrix of dimensions $P_0 \times P_1P_2$. We note that, for example, reshaping $\mathbf{q}^{P_1P_2;Q_0}$ to $\mathbf{q}^{P_1;P_2Q_0}$ requires only a change in the stride and does not necessitate the reordering of data in memory. In this form, the operation may be computed particularly efficiently at higher orders in the hexahedral region, especially if an optimised BLAS is available.

2.2 Example problem

The problem considered is that of solving the Helmholtz equation

$$\nabla^2 u - \lambda u = f,$$

on the unit cube $[0, 1]^3$ with $\lambda > 0$. This can be written more compactly as

$$\mathcal{H}u = f \tag{2.4}$$

with positive-definite $\mathcal{H} = \nabla^2 - \lambda$. This is a typical calculation encountered in a broad range of applications, such as in the solution of the incompressible Navier-Stokes equations, and therefore makes an effective example of the optimisation techniques which may be applied within a spectral/hp framework. We optimise the choice of (h, P)-discretisation so as to attain the best performance when evaluating the matrix-vector product $\mathbf{H}\tilde{\mathbf{u}}$, where $\mathbf{H}$ is the spectral/hp discretisation of the continuous operator $\mathcal{H}$, while $\tilde{\mathbf{u}}$ denotes an intermediate residual vector in an iterative conjugate gradient algorithm. Repeated calculation of this matrix-vector product dominates the iterative solution of Eqn. 2.4. The domain is equipped with suitable Dirichlet boundary conditions appropriate to the forcing function f. For this study we consider the forcing function to be of the form

$$f = -(\lambda + 3n^2\pi^2)\sin(n\pi x)\sin(n\pi y)\sin(n\pi z), \tag{2.5}$$

which has the exact smooth solution

$$u(x, y, z) = \sin(n\pi x)\sin(n\pi y)\sin(n\pi z). \tag{2.6}$$

The integer n parametrises the family of problems, with high frequency forcing functions leading to larger solution errors. These are measured in the L_2-norm and defined as

$$E = \left[\int_{\Omega} (u_{exact} - u)^2 \right]^{\frac{1}{2}} .$$

For each (h, P) pair, f is projected onto the spectral/hp discretisation, the time taken to evaluate $\mathbf{H}\tilde{\mathbf{u}}$ is measured and Eqn. 2.4 is solved for u. The solution error is calculated as the difference between this solution and a discretisation of the analytic solution. This error is calculated using 30 quadrature points, rather than the $P+1$ points necessary to support an order P basis, to avoid poor measurement through aliasing effects with low-order expansions. The choice of n controls the solution frequency and high frequency solutions will be captured less accurately by a fixed discretisation. While for increasing n there is a quantitative difference in the convergence of the error as the discretisation is refined, the qualitative picture remains comparable. Therefore, for the results presented in the following section, the choice of $n = 3$ is used.

2.3 Test system

The dependence on hardware of the timings makes the specification of the system particularly relevant. All calculations presented in this paper are conducted on the same Mac Pro 64-bit system with dual quad-core 2.26GHz Intel Xeon processors and 16GB system memory. The implementation of BLAS is the Accelerate framework distributed with OS X. Matrix solves, used to determine the solution accuracy of a given discretisation, are computed using iterative methods. The tolerance for the iterative procedure is set to 10^{-14} to ensure the solution accuracy is correctly determined for all discretisations considered.

3 Optimal Strategy in three dimensions

We define the *optimal strategy* for an (h, P)-pair to be the strategy which minimises runtime for that discretisation. This is shown for hexahedral and tetrahedral elements in Fig. 3 and will be the strategy used in selecting the optimal discretisation. In summary, global strategies offer better performance at low expansion orders, while sum-factorisation or local-matrix approaches offer better performance in high-order regimes.

Figure 4 shows logarithmic contours of runtime (dotted lines) overlaid with logarithmic contours of solution error (solid lines) measured in the L_2-norm. The first three plots (Fig. 4(a-c)) show contours of runtime for sum-factorisation, local elemental evaluation and global evaluation. Figure 4(d) shows the minimum runtime of these strategies for each discretisation, that is the runtime for the optimal strategy as described in Fig. 3. Contours of error are the same on all plots in the figure. The apparent distortion of the 10^{-3} error contour in the hexahedral case is an artifact of the $(h, P) = (\frac{1}{3}, 4)$ case representing the $n = 3$ solution particularly accurately due to the alignment of the h-mesh with our chosen solution. A similar artifact may be observed at $(h, P) = (\frac{1}{3}, 6)$.

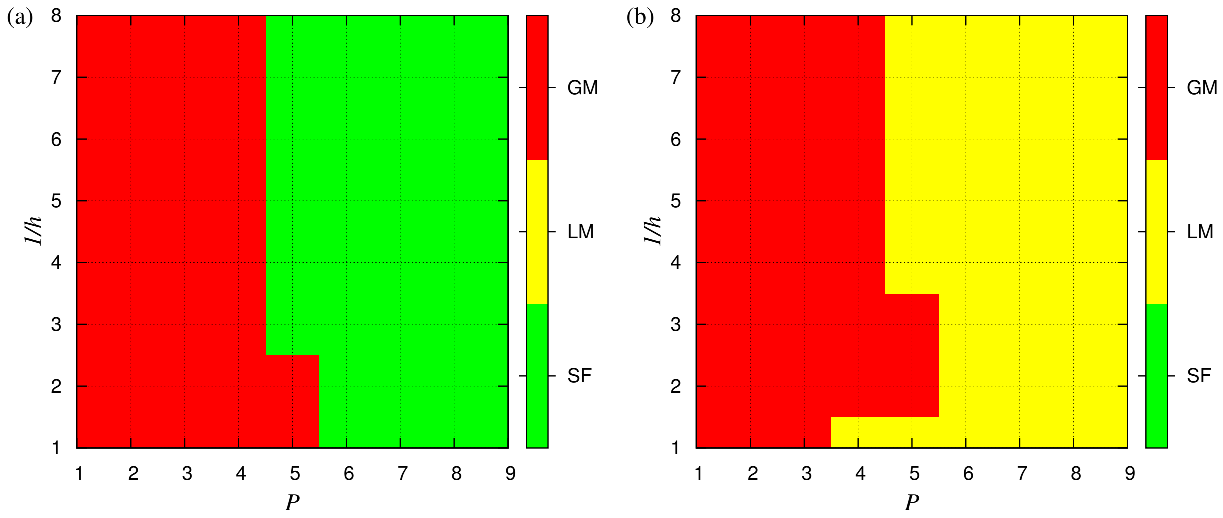

Figure 3: Optimal implementation strategies for the space of (h, P) discretisations on (a) hexahedral and (b) tetrahedral elements. As expected, the global strategy (GM) offers the highest performance for low-order expansions, while local-matrix (LM) and sum-factorisation (SF) strategies offer better performance at high orders.

We begin by briefly relating our present understanding of strategy performance from the results for the hexahedral region presented in Cantwell *et al.* [2], to the data in Fig. 4. They observe the sum-factorisation strategy for the Helmholtz operator performs poorly at low orders. In Fig. 4 this is apparent from a quantitative comparison of the runtime contours across the first three plots. The global strategy performs best at low orders, again notable from the low runtime-contour values. The vertical orientation of the runtime contours at low expansion orders suggests that this strategy also scales well with increasing mesh density in this low-order regime. This is in contrast to the sum-factorisation contours which are more diagonally oriented and therefore support scaling with polynomial order. A final point to note from the first three implementation plots is that for the local elemental and global strategies the runtime contours are similarly oriented and with similar contour separation. This aligns with the observation from Cantwell *et al.* [2] that the relative increase in runtime of the global strategy with polynomial order reaches a plateau of approximately twice the runtime of the local elemental approach.

We next examine how the optimal discretisation may be selected for the three strategies separately. For the case of the sum-factorisation, the orientation of the contours of runtime compared with the contours of error leads to an optimal discretisation consisting of a small number of high-order elements. This is true independent of the desired solution accuracy. In contrast, the global strategy is more complicated. The runtime and error contours are largely parallel leading to a broad spectrum of discretisations with favourable performance. However, for large error tolerances the marginally optimal choice is to favour mesh density over high order, while for small

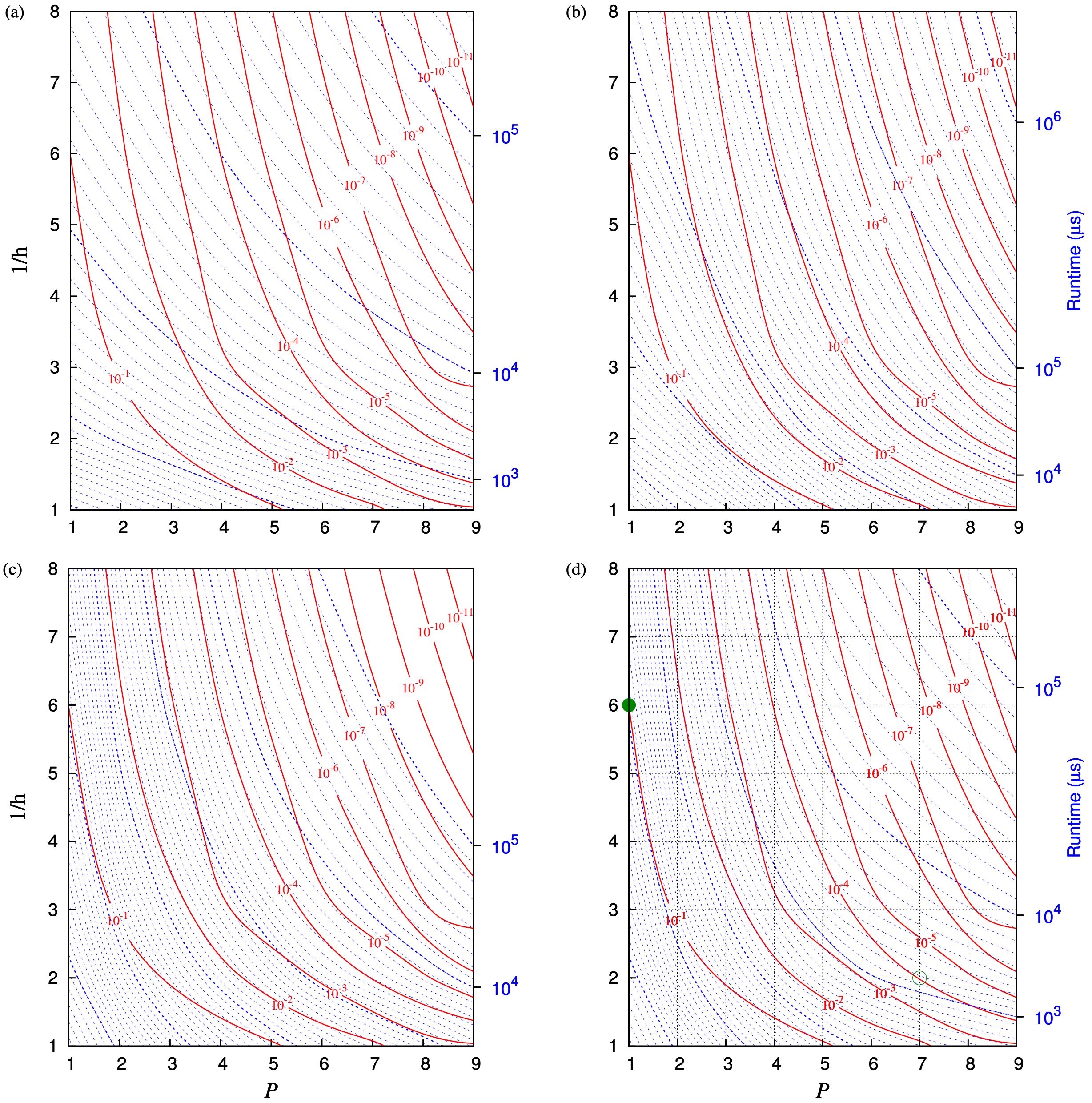

Figure 4: Contour plots showing the runtime of a single operator evaluation (dotted lines) and L_2-error (solid lines and fixed across all plots) in solving the Helmholtz problem for each (h, P)-combination using hexahedral elements. The three evaluation strategies are shown: sum-factorisation (a), elemental matrices (b) and global matrix (c). For the global matrix (c), contours above 10^5 are not shown due to the problem size exceeding the memory of the test system. A comparison with the optimal strategy chosen for each discretisation is shown in (d), where the filled circle marks the optimal discretisation to attain a solution with a 10% error tolerance, while the open circle indicates the optimal discretisation for 0.01%.

tolerances the exponential convergence dominates leading to high order providing better performance. The local elemental approach is more interesting still. For a more relaxed error tolerance neither h-refinement or P-refinement is dominant and often a middle-ground discretisation is more appropriate. However, as with the global strategy, high order dominates when accuracy is required.

The final sub-figure, Fig. 4(d), shows the comparison for the optimal strategy. This essentially merges the results from Fig. 4(a)-(c) to show the discretisation which offers the best possible performance. The character of the optimal strategy contours can be seen to contain aspects from the three base strategies, most notably from the global strategy at low orders and the sum-factorisation strategy at high orders. Of course, in selecting an optimal discretisation only discrete selections of h and P are possible and the intersection of the grid lines denotes the available choices. In particular, we highlight the optimal discretisation for a 10% error tolerance (marked by the filled circle on the figure), and the optimal discretisation for a 0.01% tolerance (marked by the open circle). We will discuss these choices in the next section.

We now address the analysis of tetrahedral elements in Fig. 5. The format of the plots in this figure is identical to Fig. 4, showing error and runtime for the three base strategies and the optimal case. A comparison of the performance for the different strategies is again given in Cantwell *et al.* [2]. The selection of optimal discretisation for these strategies is much more straightforward. For the sum-factorisation strategy a high-order approach is favoured in all cases, while for the local-elemental strategy high order is the better approach in all but cases requiring low solution accuracy. For the global approach a low-order finite element discretisation offers the best performance. The latter is most likely a consequence of the multiplicity of the degrees of freedom in a tetrahedral mesh.

For the optimal implementation, the approach to take is dependent on the solution accuracy required, in much the same way as for the hexahedral case. With a high tolerance of error, one should select a low-order discretisation, while for greater accuracy high-order elements are essential for optimal performance. Again, we show the optimal strategy for the tetrahedral case and mark the optimal discretisations for a 10% and 0.1% error tolerance. Finally, it can be noted that qualitatively, hexahedral elements offer better performance over tetrahedral elements for this particular problem with a difference of approximately half an order of magnitude in runtime for both the 10^{-1} and 10^{-4} error contours.

4 Discussion

The selection of an optimal discretisation for a problem can be appreciated to be a non-trivial task. It is dependent not only on the strategy employed by the spectral/hp solver, but on the nature of the problem and the desired accuracy. In this paper we have computed and analysed how careful selection of the discretisation influences performance of a spectral/hp element solver. The Helmholtz problem provides an effective demonstration of the complexities and caveats of choosing discretisation and implementation to attain the maximum performance to the required accuracy. The discrete nature of the parameters in this process also complicates matters and hinders a precise

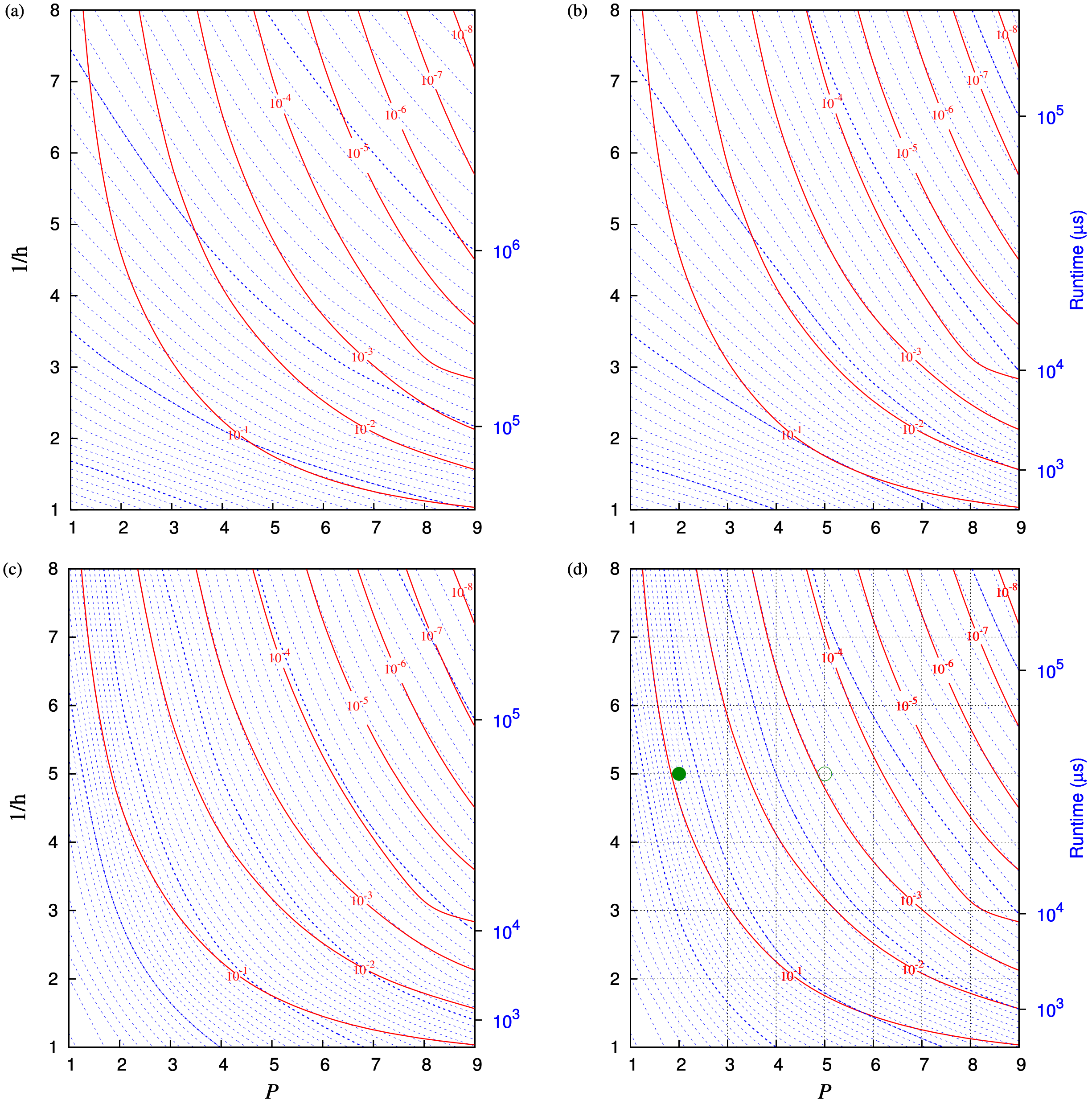

Figure 5: Contour plots showing the runtime of a single operator evaluation (dotted lines) and L_2-error (solid lines and fixed across all plots) in solving the Helmholtz problem for each (h, P)-combination using tetrahedral elements. The three evaluation strategies are shown: sum-factorisation (a), elemental matrices (b) and global matrix (c). A comparison with the optimal strategy chosen for each discretisation is shown in (d), where the filled circle marks the optimal discretisation to attain a solution with a 10% error tolerance, while the open circle indicates the optimal discretisation for 0.1%.

rule being formed. Nevertheless, the results do provide valuable guidelines to the approach which should be followed and significant conclusions may still be drawn from these figures.

Figures 4(d) and 5(d) are the culmination of this analysis where the lowest runtime across the three implementations is selected for each (h, P)-combination. We can directly read off the optimal discretisation with an error tolerance of 10%. For both hexahedral and tetrahedral elements the optimal strategy (which corresponds to a global matrix evaluation in this case) is to use linear finite elements to represent the solution. Interestingly, this is in contrast to what was found in the analysis of the two-dimensional case [15]. If instead we needed a much higher accuracy of 1%, the answer is less obvious and we should take more care. For hexahedral elements, the minimal runtime along the contour is approximately $10^{2.3}$. Of course, we are in fact limited to discrete choices of h and P and consequently $h = \frac{1}{7}, P = 2$ is the optimal choice here with a runtime of $10^{2.4}$. We also quote the optimal choice for 0.01% accuracy as being $(h, P) = (\frac{1}{2}, 7)$.

Finally, we note some limitations of the results presented. It must be remembered that this study, as with any other in which measurements of performance are considered, is both problem and hardware dependent. The exact optimal discretisation determined through this analysis may change when the analysis is performed on a range of alternative architectures. Additionally, different problems may favour particular implementation strategies leading to a performance shift. It would be of interest to quantify the extent to which different problems, for example those without infinitely smooth solutions, affect the selection of the optimal discretisation. Finally, in practice the full solution time of the Helmholtz problem will depend upon the choice of iterative technique employed. The timings presented here reflect only the low-level matrix-vector operation which typically forms the dominant sub-step in these techniques. However, the process and general trends presented here should help guide the selection process and highlight the essential need for modular codes which support both low- and high-order discretisations.

Acknowledgements

CDC acknowledges support from the Wolfson Foundation. SJS acknowledges support from the EPSRC Advanced Research Fellowship.

References

[1] P.E. Bernard, J.F. Remacle, R. Comblen, V. Legat, and K. Hillewaert. High-order discontinuous Galerkin schemes on general 2D manifolds applied to the shallow water equations. *Journal of Computational Physics*, 2009.

[2] C. D. Cantwell, S. J. Sherwin, R. M. Kirby, and P. H. J. Kelly. From h to p efficiently: strategy selection for operator evaluation on hexahedral and tetrahedral elements. *Computers and Fluids*, (submitted), 2010.

[3] M. O. Deville, P. F. Fischer, and E. H. Mund. *High-order methods for incompressible fluid flow*. Cambridge University Press, Cambridge, 2002.

[4] M. Dubiner. Spectral methods on triangles and other domains. *J. Sci. Comp.*, 6(4):345–390, 1991.

[5] D. Gottlieb and S.A. Orszag. *Numerical analysis of spectral methods: theory and applications*. Society for Industrial Mathematics, 1977.

[6] JS Hesthaven and T. Warburton. Nodal High-Order Methods on Unstructured Grids:: I. Time-Domain Solution of Maxwell's Equations. *Journal of Computational Physics*, 181(1):186–221, 2002.

[7] T. J. R. Hughes. *The finite element method.* Prentice-Hall, New Jersey, 1987.

[8] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, Oxford, second edition edition, 2005.

[9] U. Lee. *Spectral element method in structural dynamics*. Wiley, 2009.

[10] S. A. Orszag. Spectral methods for problems in complex geometries. *Advances in computer methods for partial differential equations- III*, pages 148–157, 1979.

[11] A.T. Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.*, 54(3):468–488, 1984.

[12] S. J. Sherwin and G. E. Karniadakis. Tetrahedral *hp* finite elements: Algorithms and flow simulations. *J. Comput. Phys.*, 124:14–45, 1996.

[13] S.J. Sherwin. Hierarchical hp finite elements in hybrid domains. *Finite Elements in Analysis and Design*, 27(1):109–119, 1997.

[14] B.F. Smith, P. Bjorstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge Univ Pr, 2004.

[15] P. E. J. Vos, S. J. Sherwin, and M. Kirby. From h to p efficiently: Implementing finite and spectral/hp element discretisations to achieve optimal performance at low and high order approximations. *J. Comput. Phys.*, 229:5161–5181, 2010.

[16] OC Zienkiewicz, RL Taylor, and JZ Zhu. *The finite element method: its basis and fundamentals.* Elsevier Butterworth Heinemann, 2005.