SCI INSTITUTE
TECHNICAL REPORT

Comparison of Consistent Integration Versus Adaptive Quadrature For Taming Aliasing Errors

Hanieh Mirzaee[*], *Claes Eskilsson*[†], *Spencer J. Sherwin*[‡], *Robert M. Kirby*[*]
[*]SCI Institute, University of Utah, Salt Lake City, Utah, USA
[†]Dept. of Shipping and Marine Technology, Chalmers Univ. of Tech., Göteborg, Sweden
[‡]Dept. of Aeronautics, Imperial College London, South Kensington Campus, London, UK

UUSCI-2009-008

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

August 28, 2009

Abstract:

Most spectral/*hp* element methods, whether employing a modal or nodal basis representation, evaluate non-linear differential operators in "physical space" at a collection of collocation or quadrature points. The number of points used is often set to what is needed to represent the original solution over an element (or integrate the square of the function over an element), and not to what is needed to represent the square of the function. This discrepancy leads to aliasing errors, which when the fields are highly resolved have little appreciative impact and hence can be ignored. In under-resolved scenarios, aliasing can pollute the solution leading to decreased accuracy and issues of stability. These errors can be eliminated by consistent integration at the price of increased computational cost. In most engineering simulations, however, the issue is not binary: not all elements within a simulation domain contain under-resolved solutions nor full-resolved solutions. The location and times at which elements support under-resolved solutions varies based upon the dynamics of the system. Hence an efficient mean of taming aliasing errors can be through dynamic quadrature.

In this report, we present analysis that compares the computational efficiency of an adaptive consistent integration strategy that dynamically adapts the level of quadrature based upon *aliasing indicator*, with the traditional consistent integration approach. Hierarchical Gauss-Kronrod quadrature is used, allowing for both error estimation and consistent integration of quadratic non-linearities at a single set of points which have as their subset the classic integration points. Our theoretical estimates indicate for our adaptive scheme to be efficient, linear quadrature needs to be used on the majority of elements to keep the overall computational cost less than the cost of the consistent integration approach. Two dimensional type elements are considered in all the analysis.

THE UNIVERSITY OF UTAH

Comparison of Consistent Integration Versus Adaptive Quadrature For Taming Aliasing Errors

Hanieh Mirzaee[1], Claes Eskilsson[2], Spencer J. Sherwin[3] and Robert M. Kirby[4]

August 28, 2009

Abstract

Most spectral/*hp* element methods, whether employing a modal or nodal basis representation, evaluate non-linear differential operators in "physical space" at a collection of collocation or quadrature points. The number of points used is often set to what is needed to represent the original solution over an element (or integrate the square of the function over an element), and not to what is needed to represent the square of the function. This discrepancy leads to aliasing errors, which when the fields are highly resolved have little appreciative impact and hence can be ignored. In under-resolved scenarios, aliasing can pollute the solution leading to decreased accuracy and issues of stability. These errors can be eliminated by consistent integration at the price of increased computational cost. In most engineering simulations, however, the issue is not binary: not all elements within a simulation domain contain under-resolved solutions

[1]mirzaee@cs.utah.edu, School of Computing, Univ. of Utah, Salt Lake City, Utah, USA

[2]claes.eskilsson@chalmers.se, Dept. of Shipping and Marine Technology, Chalmers Univ. of Tech., SE-412 96 Göteborg, Sweden

[3]s.sherwin@imperial.ac.uk, Dept. of Aeronautics, Imperial College London, South Kensington Campus, London, SW7 2AZ, UK

[4]*Corresponding Author.* kirby@cs.utah.edu, School of Computing, Univ. of Utah, Salt Lake City, Utah, USA

nor full-resolved solutions. The location and times at which elements support under-resolved solutions varies based upon the dynamics of the system. Hence an efficient mean of taming aliasing errors can be through dynamic quadrature.

In this report, we present analysis that compares the computational efficiency of an adaptive consistent integration strategy that dynamically adapts the level of quadrature based upon *aliasing indicator*, with the traditional consistent integration approach. Hierarchical Gauss-Kronrod quadrature is used, allowing for both error estimation and consistent integration of quadratic non-linearities at a single set of points which have as their subset the classic integration points. Our theoretical estimates indicate for our adaptive scheme to be efficient, linear quadrature needs to be used on the majority of elements to keep the overall computational cost less than the cost of the consistent integration approach. Two dimensional type elements are considered in all the analysis.

Keywords: spectral/*hp* elements, high-order finite elements, aliasing, de-aliasing, integration errors, Gaussian integration, Gauss-Kronrod points, consistent integration, over-integration, insufficient quadrature, adaptive quadrature

1 Introduction

In spectral/*hp* element methods, the quadratic non-linearities of the incompressible Navier-Stokes equations and the cubic non-linearities in the compressible Navier-Stokes are typically computed in *physical space* using collocation projections. Specifically, the primary fields (*e.g.*, velocity, pressure, energy) are first transformed into a physical space representation where the fields are discretely evaluated at a set of collocation points.

In spectral/hp element methods following a tensor-product numerical quadrature approach, quadrilateral and hexahedral elements are integrated at tensor-products of Gauss-Legendre (GL) or Gauss-Lobatto-Legendre (GLL) points [8, 11, 12] and triangles, tetrahedra, prisms and pyramids are integrated at tensor-product configurations of Gauss-Lobatto-Legendre (GLL) points and Gauss-Radau-Legendre (GRL) points, GLL-GRL-GRL points, GLL-GRL-GLL points and GLL-GLL-GRL points respectively [11, 12]. Once a collection of collocation/quadrature points have been specified, non-linear products can be obtained at the discrete points in a collocation fashion analogous to the pseudo-spectral evaluation commonly adopted in global spectral methods.

A normal practice in polynomial Galerkin methods is the use of sufficient quadrature to integrate the linear differential terms exactly. In [13, 17] it was argued that employing an insufficient quadrature rule for evaluating non-linear terms leads to an aliasing error that pollutes the solution, possibly degrading the accuracy of the solution and in the worst case leading to issues in the numerical instability of the scheme. In [13], the proposed solution was dubbed "over-integration" – a term implying that de-aliasing of the solution requires the use of more quadrature points than would be necessary to integrate linear differential terms in a polynomial Galerkin method exactly. In this paper we refer to this technique as "insufficient quadrature" (the same language used by [14]) since this terminology more accurately represents the approximation. The errors caused by insufficient quadrature can be bounded by the theoretical estimates of [6], and often this result is used to support the idea that, if the simulation is well-resolved, then the numerical crimes committed by insufficient quadrature are negligible. However, as pointed out in [5] the theory does not address what

happens in the marginally or badly resolved cases frequently encountered in practice.

Although use of consistent integration of the non-linear products eliminates the problems associated with aliasing brought on by insufficient quadrature used when handling non-linear terms, there is a computational penalty (as reported in [9]). Specifically, as was discussed in [13], at least 3/2 times more quadrature points *per direction* are required to properly integrate quadratic non-linearities (two times more points per direction for cubic non-linearities). Hence, there may be a reasonably high computational cost when adopting consistent integration rules for the non-linear terms as compared to a quadrature rule designed to consistently handle the standard linear terms.

The numerical practitioner often wants to know answers to the following questions: (1) When is exact or consistent integration absolutely necessary? (2) Do all elements (or regions of the domain) require the use of consistent integration or can its use be selective? (3) If use of consistent integration can be done on a selective basis, how does one *automatically* determine when it should be used? And lastly, (4) Is there any computationally efficient, dynamic middle-ground between insufficient quadrature and exact integration (for example, that uses consistent integration only when necessary and does so in a way that the decision algorithm is not more costly than doing consistent integration everywhere)?

The main purpose of this report is to understand the computational costs when evaluating quadratic non-linear products on one-dimensional and two-dimensional spectral/*hp* discretisations with the goal of helping the numerical practitioner answer the questions mentioned above. Once solved in one- and two-dimensions, the results of this study extend naturally to the aforementioned three-dimensional discretisation types. Specifically, we attempt to

answer (or review the answers to) the following questions:

- What is the computational cost of consistent integration versus the "under-integration" alternative?
- At the extra cost of evaluating the function at a larger number of Gauss-Kronrod points on *all* elements, can one devise a scheme in which performing consistent integration on only a small proportion of elements is more efficient than consistent integration on all elements?

The report is organized as follows. In Section 2, we present a theoretical discussion of the aliasing error due to the projection of squared polynomials (to mimic quadratic non-linearities) and provide a numerical example to help gain intuition into the results in higher dimensions. We also present an example mesh and flow solution from a published three-dimensional fluid mechanics simulation that helps demonstrate that a large number of the elements within the simulation domain do not need consistent integration, and hence the added cost of the procedure on these elements is a waste of computational resources. The combination of these two examples motivates our search for a dynamically adaptive strategy. In Section 3 we present a review of tensor-product quadrature rules as typically used in spectral/*hp* element discretisations with specific emphasis on the operation counts incurred. In Section 4 we present the core building block of our adaptive strategy, Gauss-Kronrod integration. Gauss-Kronrod integration is a (spatially) hierarchical quadrature rule that augments traditional Gaussian integration rules; the hierarchical nature of the point locations allow one to develop error estimation strategies. In Section 5 we present both the algorithm and operation cost analysis for our dynamic adaptation strategy. In Section 6, we discuss

two alternatives to our initial adaptive scheme. In Section 7, we summarize our findings and provide some guidelines for the numerical methods practitioner to understand the trade-offs between the different strategies presented.

2 Motivation

To motivate our discussion, we first examine the essence of the non-linear evaluation based on a simple illustrative example in one dimension. Assume we are given a single spectral/*hp* element E defined on $[-1,1]$ supporting polynomials up to degree N. We denote by $\{\phi_i(\xi)\}$, $i = 0,\ldots,N$ an orthonormal basis in $L_2[-1,1]$ (*i.e.* the scaled Legendre polynomial basis) that spans the polynomial space $\mathcal{P}^N$; the index i provides the maximum degree of the polynomial expression denoted by the basis function ϕ_i.

We now presume that we have a polynomial expansion of the form:

$$u(\xi) = \sum_{i=0}^{N} \hat{u}_i \phi_i(\xi),$$

and we are interested in obtaining the expansion

$$w(\xi) = \sum_{k=0}^{N} \hat{w}_k \phi_k(\xi)$$

such that $\|w(\xi) - [u(\xi)]^2\|_{L_2}$ is minimized. The modal coefficients $\hat{w}_k$ are uniquely determined through the Galerkin projection which can be determined in this case by the following expression:

$$\hat{w}_k = \sum_{i,j=0}^{N} \hat{u}_i \hat{u}_j \int_{-1}^{1} \phi_i(\xi)\phi_j(\xi)\phi_k(\xi)\, d\xi \tag{1}$$

for $k = 0, \ldots, N$.

Note that the integrand in Equation (1) contains the product of three polynomials from $\mathcal{P}^N$, and hence it is at most a polynomial in $\mathcal{P}^{3N}$. For notational simplicity, let us define the inner product of two functions on $[-1, 1]$ as follows:

$$(f, g) = \int_{-1}^{1} f(\xi) g(\xi) \, d\xi,$$

and the corresponding Gauss-Lobatto-Legendre (GLL) quadrature approximation of the inner product

$$[f, g]_Q = \sum_{i=1}^{Q} \omega_i f(z_i) g(z_i),$$

where z_i and ω_i denote the GLL quadrature points and weights, respectively. The natural number Q denotes the number of points (or weights) used. We know that the GLL quadrature is an exact approximation (to machine precision) of the inner product when the integrand $\mathcal{I} \in \mathcal{P}^{2Q-3}$ and has an error term of the following form [1, 10],

$$E_Q = -\frac{Q(Q-1)^3 2^{2Q-1}[(Q-2)!]^4}{(2Q-1)[(2Q-2)!]^3} \mathcal{I}^{(2Q-2)}(\zeta)$$

where $\zeta \in [-1, 1]$.

Now let us consider the case where an insufficient quadrature is employed. As a common example, we consider a quadrature order sufficient to integrate polynomials in $\mathcal{P}^{2N}$, as typically required for a linear operation in a Galerkin discretisation. The relation between the true modal solution $\hat{w}_k$, defined in Equation (1), and the approximate modal solution $\tilde{w}_k$,

defined as

$$\tilde{w}_k = \sum_{i,j=0}^{P} \hat{u}_i \hat{u}_j [\phi_i(\xi)\phi_j(\xi), \phi_k(\xi)]_Q,$$

is given by the following expression,

$$\tilde{w}_k = \hat{w}_k - \left[\sum_{\substack{i,j=0 \\ i+j>2P-k}}^{N} \hat{u}_i \hat{u}_j \left((\phi_i \phi_j, \phi_k) - [\phi_i \phi_j, \phi_k]_Q \right) \right], \tag{2}$$

for $k = 0, \ldots, N$.

To demonstrate how insufficient integration influences the modes, consider the following "worst-case" numerical example. Suppose that $N = 10$ (*i.e.* 10^{th} degree polynomials) and that the function we are trying to project is given by $u(x) = \sum_{i=0}^{10} \phi_i(\xi)$ – which amounts to all the modal coefficients $\hat{u}_i$ being set to 1.0 and mimics a case in which an element has significantly under-resolved the solution within the element. The exact Galerkin projection yielding the modes $\hat{w}_k, k = 0, \ldots, N$ can be computed when $Q = 17$ GLL points/weights are used to approximate the integral $[\cdot, \cdot]_Q$ given by Equation (1). Suppose, however, that only $Q = 12$ GLL points/weights are used (sufficient for integrating a polynomial in $\mathcal{P}^{2N}$ exactly). Instead of arriving at $\hat{w}_k$, we obtain $\tilde{w}_k$ as given in Equation (2). For this particular example, the difference between the true and the approximate inner product of the triplet is negative for each k. Combining this fact with the observation that $\hat{u}_i \hat{u}_j = 1 \; \forall i, j$ in this example, we find that there is additional energy added to every mode but the mean mode (*i.e.* $k = 0$).

In Figure 1, we present a bar chart of the magnitude of the modal energies when reduced quadrature ($\tilde{w}_k$) and exact quadrature ($\hat{w}_k$) are adopted. Note that modification of the integration rule used (*e.g.* using Gauss-Legendre as opposed to Gauss-Lobatto-Legendre)

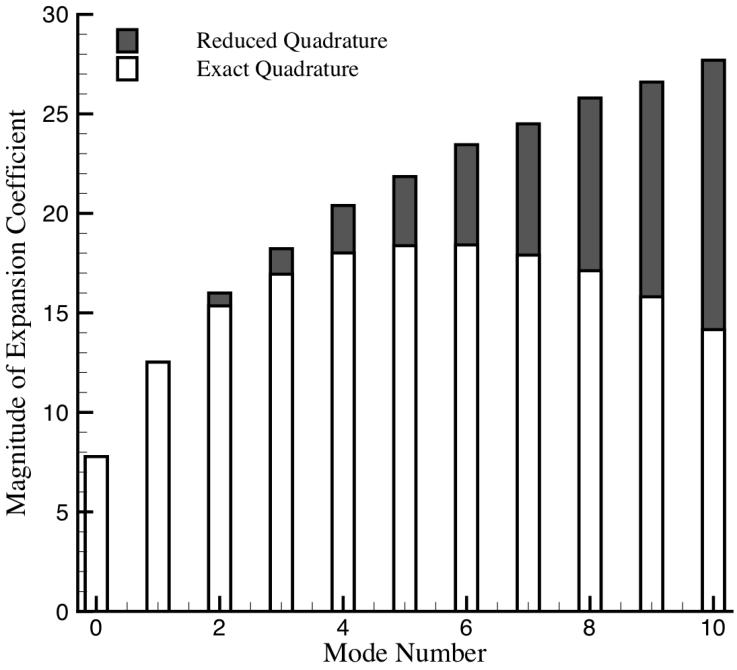

Figure 1: Magnitude of the modal coefficients when exact quadrature ($Q = 17$) and reduced quadrature ($Q = 12$) is used. In this example, energy is added due to insufficient quadrature to all but the mean mode (*i.e.* k=0).

or different choices of function may lead to different aliasing characteristics (such as energy removal).

Whereas our one-dimensional example demonstrates how pernicious aliasing errors can be and provides strong motivation for using consistent integration, practitioners are quick to point out that it demonstrates a worst case scenario that is not often to happen (in the case of being so grossly under-resolved); it does not indeed represent the average case. To understand the practitioners' "average case", one must examine the discretisations and scenarios under which aliasing might be an issue.

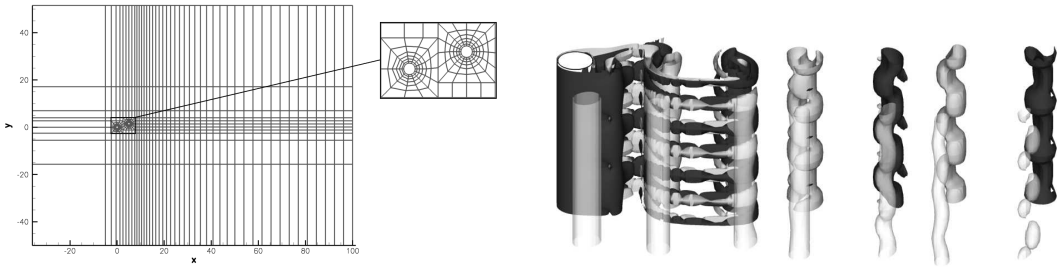

Figure 2: Example of a mesh used in engineering practice (left) and representative flow solution generated on the mesh (right) as presented in [7].

Consider the mesh present in Figure 2 (left) and corresponding flow solution (right) taken from the work of one of the authors [7] in which they are studying the wake dynamics generated by flow around two circular cylinders in a staggered spatial configuration. In large part for numerical reasons, the mesh used for numerical simulation contains a large number of elements upstream of the cylinders (*i.e.* ahead of the cylinders) and flanking the cylinders. The solution in these elements is equally to or is only a small perturbation off of the inflow free-stream condition, which is set to a uniform flow going left to right. Although the full three-dimensional (non-linear) Navier-Stokes equations are being solved in these elements,

the solution is so well resolved that consistent integration provides no appreciative benefit for the required computational cost. Only a small proportion of elements residing around and in the wake of the cylinder cover regions in which the dynamics of the system would even suggest the use of consistent integration.

These two examples provide the motivation for our work. In the worst case, aliasing errors can greatly impact the solution quality in a variety of ways. Consistent integration is one means of avoiding aliasing issues, but is rather expensive compared to what is normally done in practice. As demonstrated by the flow example, large regions may never need the benefits of consistent integration, and hence correspond to a wasted investment of computational resources. The key questions are then how to determine what elements need consistent integration and how to employ it adaptively if and when it is needed. In the next section, we review the quadrature approaches that are commonly used in spectral/hp elements, in Section 4 we introduce Gauss-Kronrod quadrature, what we think to be the building block of an aliasing adaptation strategy, and in Section 5 we try to provide an algorithm for adaptively handling consistent integration.

3 Quadrature Used in Spectral/hp Elements

In this section we review the various quadrature rules used for integrating tensor-product based spectral/hp element methods [11,12]. We start in Section 3.1 with a review of Gaussian quadrature with appropriate definitions of notation that will be used throughout the paper. In Section 3.2 we outline the operator that we will study. In Sections 3.3.1 - 3.3.3, we will present theoretical estimates on the computational cost of the operator under investigation

for segments, quadrilaterals and triangles.

3.1 Brief Review of Gaussian Quadrature

Gaussian quadrature can in some sense be considered the work-horse of almost all polynomial Galerkin spectral/*hp* methods. Almost all simulation codes apply some form of Gaussian quadrature. As its usage is ubiquitous, we will not review all the historical and theoretical properties of the quadrature rule – we refer the interested reader to [5]. In this section, we will review only those components and notations that we require for discussion of our consistent integration scheme.

Let us assume for simplicity that we are interested in the integral of a function given by:

$$\mathcal{I}[f] := \int_{-1}^{1} f(x)\, dx.$$

Without loss of generality, we will limit our discussion to integration on $[-1, 1]$; extensions to compact intervals and to multi-dimensional elements exist through the use of affine maps and polynomial bijective *a.e.* maps respectively.

Gauss-Legendre Quadrature is defined as follows. Given Q Gauss-Legendre points $-1 < x_1 < \ldots < x_k < \ldots < x_Q < 1$ and corresponding weights $\{\alpha_k\}$, $k = 1, \ldots, Q$, the Gaussian quadrature approximation of the integral defined above is given by:

$$\mathcal{G}_Q[f] := \sum_{k=1}^{Q} \alpha_k\, f(x_k).$$

Gaussian integration has the unique property that if $f \in \mathcal{P}^{2Q-1}$, $\mathcal{G}_Q[f] = \mathcal{I}[f]$. For more information on how to calculate the points and weights refer to [12].

Two further quadrature rules are commonly employed. *Gauss-Radau-Legendre Quadrature* establishes a collection of points and weights such that one of the points is pinned at one

end of the interval (*i.e.* $-1 = x_1 < \ldots < x_k < \ldots < x_Q < 1$) and has the unique property that if $f \in \mathcal{P}^{2Q-2}$, $\mathcal{G}_Q[f] = \mathcal{I}[f]$. Similarly, *Gauss-Lobatto-Legendre Quadrature* establishes a collection of points and weights such that $-1 = x_1 < \ldots < x_k < \ldots < x_Q = 1$ and has the unique property that if $f \in \mathcal{P}^{2Q-3}$, $\mathcal{G}_Q[f] = \mathcal{I}[f]$.

3.2 Operator To Be Studied

The fundamental operator that we will study in this report is one which takes as input a vector of modal values on an element and produces a vector of the same dimension denoting the inner product of the basis functions against the result of a non-linear operator acting on the input vector. We will focus all our discussion on the simple squaring of a function, although the results and discussion can easily be extended to more complicated non-linear operators.

Independent of dimension, assume we denote our polynomial function over an element Ω_e by

$$u(\vec{x}) = \sum_{i=0}^{N} \hat{u}_i \phi_i(\vec{x}) \ \vec{x} \in \Omega_e$$

and define our input vector as $\{\hat{u}_k\}_{k=0}^N$. If our non-linear operator is the squaring of the function, then the idealized (exact) output vector of our method $\{\hat{w}_k\}_{k=0}^N$ is given by:

$$\begin{aligned} \hat{w}_k &= \int_{\Omega_e} \phi_i(\vec{x}) \left(u(\vec{x})\right)^2 dx & (3) \\ &= \mathcal{I}[\phi_i(\vec{x}) \left(u(\vec{x})\right)^2] & (4) \end{aligned}$$

where we have assumed appropriate mapping of the element to our reference interval (or reference element).

In practice, the integral above is approximated with Gaussian quadrature: $\hat{w}_k \approx \tilde{w}_k = \mathcal{G}_Q[\phi_i(\vec{x})\,(u(\vec{x}))^2]$. In one-dimension, if $u \in \mathcal{P}^N$ and if we employ Gauss-Legendre-Lobatto quadrature, $\hat{w}_k = \tilde{w}_k$ given $Q \geq \lceil (3N+3)/2 \rceil$. We will refer to this as *consistent integration*. If $Q = N + 2$ – that is, what is typically done for solving linear advection problems as it integrates polynomials in $\mathcal{P}^{2N}$ exactly – we will refer to this as *linear-consistent integration*.

Under the assumption that we can compute and cache the values of the integration points/weights and the basis functions evaluated at those points, the general algorithm we will employ is as follows:

(0) Input a vector of modal coefficients $\{\hat{u}_k\}_{k=0}^N$ representing a polynomial of at most degree N. Set the quadrature order to Q based upon either linear-consistent or consistent integration.

(1) Evaluate the input polynomial at the quadrature points, producing a vector of function values at the points.

(2) Evaluate the non-linear differential operator at the quadrature points, producing a vector containing the squared values.

(3) Form the right-hand-side inner product vector $\{\tilde{w}_k\}_{k=0}^N$ using Equation (4).

Items (1) and (3) can in general be computed through the use of two techniques: matrix-based methods and sum-factorization. In the sections to follow, we will provide the theoretical computational cost estimates for the procedure for these two approaches.

3.3 Cost Functions

Here we examine the different cost functions for the linear and over quadrature mentioned in the previous section considering different element types. In the expressions that follow,

- M is the number of modes,
- $N = M - 1$ is the polynomial order,
- Q is the number of quadrature points,
- M_0 is the number of modes in direction 0,
- M_1 is the number of modes in direction 1,
- N_0 is the expansion order of the basis in direction 0,
- N_1 is the expansion order of the basis in direction 1,
- M_{tot} is the total number of modes of the expansion,
- Q_0 is the number of quadrature points in direction 0,
- Q_1 is the number of quadrature points in direction 1,
- μ is the operational cost of the floating point multiplication, and
- α is the operational cost of the floating point addition.

3.3.1 Segment

We consider the right hand side projection of the polynomial squared,

$$\tilde{w}_k = \int_{\Omega_e} u^2(x)\phi_k(x)dx \tag{5}$$

following the procedure for forming the right-hand-side vector mentioned in the previous section, and we provide theoretical cost estimates when computing this operator on segments. To evaluate (5),

- The cost of item (1), $t_i = \sum_{n=0}^{N} \hat{u}_n \phi_n(x_i), \quad \forall(i)$:

  $C_{item1} = QM\mu + Q(M-1)\alpha$

- The cost of item (2), $\quad t_i = t_i^2, \quad \forall(i)$:

  $C_{item2} = Q * \mu$

- The cost of item (3), $\quad \tilde{w}_k = \sum_i \alpha_i t_i \phi_k(x_i), \quad \forall(k)$:

  $C_{item3} = MQ * 2\mu + M(Q-1)\alpha$

- Total cost:

$$C_{tot} = (3QM + Q)\mu + (Q(M-1) + M(Q-1))\alpha \tag{6}$$

3.3.2 Quadrilateral Elements

Here we again provide analytical cost estimates for the right hand side projection of the polynomial squared,

$$\tilde{w}_{rs} = \int_{\Omega_e} u^2(x,y)\phi_{rs}(x,y)dxdy \tag{7}$$

when using quadrilateral type elements considering first the matrix operations and then the sum-factorization technique.

matrix operations:

To evaluate (7),

- The cost of item (1), $\quad t_{ij} = \sum_{r=0}^{N_0} \sum_{s=0}^{N_1} \hat{u}_{rs} \phi_{rs}(x_i, y_j), \quad \forall (i,j)$:

  $C_{item1} = Q_0 Q_1 M_0 M_1 \mu + Q_0 Q_1 (M_0 M_1 - 1)\alpha.$

- The cost of item (2), $\; t_{ij} = t_{ij}^2, \quad \forall (i,j)$:

  $C_{item2} = Q_0 Q_1 \mu$

- The cost of item (3), $\; \tilde{w}_{rs} = \sum_i \sum_j \alpha_i \alpha_j t_{ij} \phi_{rs}(x_i, y_j), \quad \forall (r,s)$:

  $C_{item3} = 3 M_0 M_1 Q_0 Q_1 \mu + M_0 M_1 (Q_0 Q_1 - 1)\alpha.$

- Total cost:

$$C_{tot} = (4 Q_0 Q_1 M_0 M_1 + Q_0 Q_1)\mu + (Q_0 Q_1 (M_0 M_1 - 1) + M_0 M_1 (Q_0 Q_1 - 1))\alpha \qquad (8)$$

sum-factorization:

To evaluate (7),

- The cost of item (1), $\quad t_{ij} = \sum_{r=0}^{N_0} \psi_r(x_i) \sum_{s=0}^{N_1} \hat{u}_{rs} \psi_s(y_j), \quad \forall (i,j)$:

  - The cost of the inner summation: $C_{sum1} = Q_1 M_0 M_1 \mu + Q_1 M_0 (M_1 - 1)\alpha.$
  - The cost of the outer summation: $C_{sum2} = Q_0 Q_1 M_0 \mu + Q_0 Q_1 (M_0 - 1)\alpha$
  - Hence the total cost of item (1):

    $C_{item} = (Q_1 M_0 M_1 + Q_0 Q_1 M_0)\mu + (Q_1 M_0 (M_1 - 1) + Q_0 Q_1 (M_0 - 1))\alpha.$

- The cost of item (2), $t_{ij} = t_{ij}^2, \quad \forall (i,j)$:

  $C_{item2} = Q_0 Q_1 \mu.$

- The cost of item (3): $\tilde{w}_{rs} = \sum_i \alpha_i \psi_r(x_i) \sum_j \alpha_j t_{ij} \psi_s(y_j), \quad \forall (r,s)$:

  - The cost of the inner summation: $C_{sum1} = Q_0 M_1 Q_1 * 2\mu + Q_0 M_1 (Q_1 - 1)\alpha.$
  - The cost of the outer summation: $C_{sum2} = M_0 M_1 Q_0 * 2\mu + M_0 M_1 (Q_0 - 1)\alpha.$
  - Hence the total cost of item (3):

    $C_{item3} = (2Q_0 M_1 Q_1 + 2M_0 M_1 Q_0)\mu + (Q_0 M_1 (Q_1 - 1) + M_0 M_1 (Q_0 - 1))\alpha.$

- The total cost is:

$$\begin{aligned} C_{tot} &= (2Q_0 M_1 Q_1 + M_0 M_1 Q_0 + Q_1 M_0 M_1 + Q_0 Q_1 M_0 + Q_0 Q_1)\mu \\ &+ (Q_1 M_0 (M_1 - 1) + Q_0 Q_1 (M_0 - 1) + Q_0 M_1 (Q_1 - 1) + M_0 M_1 (Q_0 - 1))\alpha \end{aligned} \tag{9}$$

3.3.3 Triangular Elements

We evaluate the integral mentioned in (7) this time taking into account the triangular element shape. We assume, $M_0 \leq M_1$, so the total number of modes is:

$$M_{tot} = \frac{1}{2} M_0 (M_0 + 1) + M_0 (M_1 - M_0) \tag{10}$$

We consider both matrix operations and sum-factorization technique.

matrix operations

To evaluate (7),

- The cost of item (1), $t_{ij} = \sum_{n=0}^{M_{tot}-1} \phi_n(x_i, y_j)\hat{u}_n, \quad \forall (i,j)$:

  $C_{item1} = M_{tot} Q_0 Q_1 \mu + (M_{tot} - 1) Q_0 Q_1 \alpha.$

- The cost of item (2), $t_{ij} = t_{ij}^2, \quad \forall (i,j)$:

  $C_{item2} = Q_0 Q_1 \mu.$

- The cost of item (3), $\tilde{w}_m = \sum_i \sum_j \alpha_i \alpha_j t_{ij} \phi_m(x_i, y_j) J_{ij}, \quad \forall m \in M_{tot}$:

  $C_{item3} = M_{tot} Q_0 Q_1 * 4\mu + M_{tot}(Q_0 Q_1 - 1)\alpha$

- The total cost is:

$$C_{tot} = (5M_{tot} Q_0 Q_1 + Q_0 Q_1)\mu + ((M_{tot} - 1)Q_0 Q_1 + M_{tot}(Q_0 Q_1 - 1))\alpha \qquad (11)$$

sum-factorization

To evaluate (7),

- The cost of item (1), $t_{ij} = \sum_{r=0}^{N_0} \psi_r(x_i) \sum_{s=0}^{g(r)} \psi_s(y_j) \hat{u}_{rs}, \quad \forall (i,j)$:

  - The cost of inner summation:

    $C_{sum1} = (\frac{1}{2} M_0(M_0 + 1) + M_0(M_1 - M_0))Q_1 \mu$

    $+(\frac{1}{2} M_0(M_0 + 1) + M_0(M_1 - M_0) - M_0)Q_1 \alpha.$

  - The cost of the outer summation: $C_{sum2} = M_0 Q_0 Q_1 \mu + (M_0 - 1)Q_0 Q_1 \alpha.$

  - Hence the total cost of item (1):

    $C_{item1} = (\frac{1}{2} M_0(M_0 + 1) + M_0(M_1 - M_0) + M_0 Q_0)Q_1 \mu$

    $+(\frac{1}{2} M_0(M_0 + 1) + M_0(M_1 - M_0) + (M_0 - 1)Q_0 - M_0)Q_1 \alpha.$

- The cost of item (2), $t_{ij} = t_{ij}^2, \quad \forall (i,j)$:

  $C_{step2} = Q_0 Q_1 \mu.$

- The cost of item (3), $\tilde{w}_{rs} = \sum_j \alpha_j \psi_s(y_j) \sum_i \alpha_i t_{ij} \psi_r(x_i) J_{ij}, \quad \forall (r,s)$:

– The cost of the inner summation: $C_{sum1} = Q_0 M_0 Q_1 * 3\mu + Q_0 M_0 (Q_1 - 1)\alpha$.

– The cost of the outer summation:

$C_{sum2} = (\frac{1}{2} M_0(M_0 + 1) + M_0(M_1 - M_0))Q_1 * 3\mu$

$+(\frac{1}{2} M_0(M_0 + 1) + M_0(M_1 - M_0))(Q_1 - 1)\alpha$.

– Hence the total cost of item (3):

$C_{item3} = 3(\frac{1}{2} M_0(M_0 + 1) + M_0(M_1 - M_0) + Q_0 M_0)Q_1\mu$

$+(\frac{1}{2} M_0(M_0 + 1) + M_0(M_1 - M_0) + Q_0 M_0)(Q_1 - 1)\alpha$.

- The total cost:

$$\begin{aligned} C_{tot} &= 4(M_{tot} + Q_0 M_0 + Q_0)Q_1\mu \\ &+ ((M_{tot} + (M_0 - 1)Q_0 - M_0)Q_1 + (M_{tot} + Q_0 M_0)(Q_1 - 1))\alpha \end{aligned} \quad (12)$$

Figure (3) shows the plots of the cost functions for quadrilateral and triangular elements.

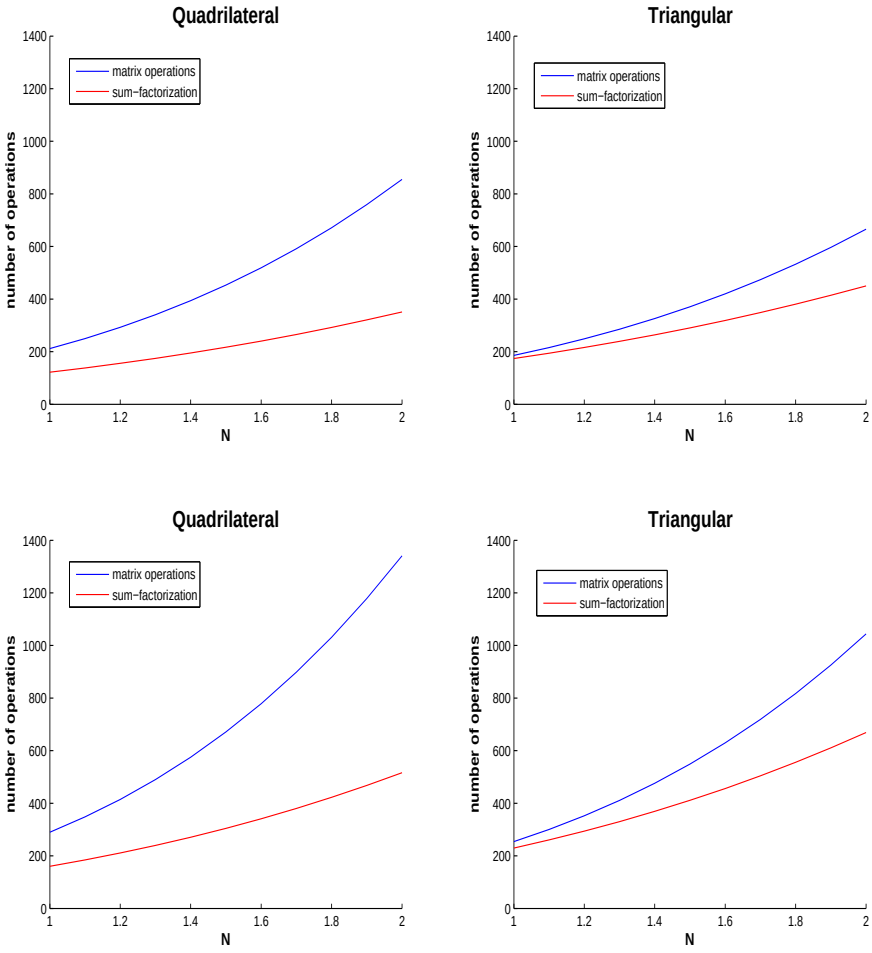

Figure 3: Cost in function of 1D expansion order, top row: linear quadrature; bottom row: consistent quadrature.

4 Review of Gauss-Kronrod Quadrature

The lesser-known area of Gaussian quadrature within the spectral/*hp* element community is that of Gauss-Kronrod quadrature. A review of the historical and theoretical background is provided in [2]. Means computing the Gauss-Kronrod points and weights are provided in [3, 15], with extensions to Kronrod-Radau and Kronrod-Lobatto in [16].

Gauss-Kronrod Quadrature is defined as follows. Given $2Q+1$ Gauss-Kronrod(-Legendre) points $-1 < z_1 < \ldots < z_k < \ldots < z_{2Q+1} < 1$ and corresponding weights $\{\omega_k\}$, $k = 1, \ldots, 2Q+1$, the Gaussian quadrature approximation of the integral defined above is given by:

$$\mathcal{K}_{2Q+1}[f] := \sum_{k=1}^{2Q+1} \omega_k f(z_k)$$

Gauss-Kronrod integration has several unique properties (reviewed in [4]), amongst them are:

- If $f \in \mathcal{P}^{3Q+1}$, $\mathcal{K}_{2Q+1}[f] = \mathcal{I}[f]$.
- The Gauss-Legendre points are a subset of the Gauss-Kronrod points: $\{x_k\}_{k=1}^{Q} \subset \{z_k\}_{k=1}^{2Q+1}$. The weights do not have a similar property; the Gauss-Kronrod weights differ from the Gauss-Legendre weights.
- The points that are "added" to the Gauss points are interlacing (a Gauss-Kronrod point lies in-between two Gauss quadrature). That is, if you consider the $2Q+1$ Gauss-Kronrod points:

$$-1 < z_1 < z_2 < z_3 < z_4 < \ldots < z_{2Q-2} < z_{2Q-1} < z_{2Q} < z_{2Q+1} < 1$$

the Q Gauss(-Legendre) points are interlaced as follows:

$$-1 < z_1 < x_1 < z_3 < x_2 < \ldots < x_{Q-1} < z_{2Q-1} < x_Q < z_{2Q+1} < 1$$

which can be succinctly written as $z_{2j} = x_j, j = 1, \ldots, Q$.

- The weights corresponding to the points that are "added" are all positive.

The first two properties are exploited to allow one to form a quadrature error estimator. Consider the following:

$$\begin{aligned}
\mathcal{K}_{2Q+1}[f] &= \sum_{k=1}^{2Q+1} \omega_k f(z_k) && (13)\\
&= \sum_{k=1}^{Q} \omega_k f(x_k) + \sum_{k=1}^{Q+1} \omega_{2k-1} f(z_{2k-1}) && (14)\\
&= \mathcal{G}_Q[f] + \sum_{k=1}^{Q} (\omega_k - \alpha_k) f(x_k) + \sum_{k=1}^{Q+1} \omega_{2k-1} f(z_{2k-1}) && (15)\\
&= \mathcal{G}_Q[f] + \sum_{k=1}^{2Q+1} \tilde{\omega}_k f(z_k) && (16)
\end{aligned}$$

where $\tilde{\omega}_{2k-1} = \omega_{2k-1}$, $k = 1, Q+1$.

In the application of Gauss-Kronrod rules, both $\mathcal{G}_Q[f]$ and $\mathcal{K}_{2Q+1}[f]$ are evaluated and the pair of approximations of $\mathcal{I}[f]$ is used to estimate the error in $\mathcal{G}_Q[f]$.

5 Adaptation Strategy and Cost Analysis

In this section we present an adaptive quadrature algorithm based upon the use of Gauss-Kronrod quadrature. We propose an "aliasing indicator" to be used for determining on what

elements consistent integration should be used. We show theoretical computational results which give the percentage of elements on which consistent integration can be performed such that the adaptive quadrature scheme is competitive with consistent integration everywhere in the domain using Gaussian quadrature.

5.1 Adaptation Quadrature Algorithm

(0) Input a vector of modal coefficients $\{\hat{u}_k\}_{k=0}^{N}$ representing a polynomial of at most degree N. Set the quadrature order to $Q = N + 2$.

(1) Evaluate the polynomial at $2Q - 1$ Lobatto-Kronrod points, producing a vector of function values $\{u(z_k)\}_{k=1}^{2Q-1}$.

(2) Evaluate the non-linear differential operator at the $2Q-1$ quadrature points, producing a vector $\{[u(z_k)]^2\}_{k=1}^{2Q-1}$.

(3) Use an aliasing indicator to establish whether to use (a) linear-consistent or (b) Kronrod-consistent integration on the element. The aliasing indicator takes as input the vector $\{[u(z_k)]^2\}_{k=1}^{2Q-1}$ and returns a binary response denoting (a) or (b).

(4) Form the right-hand-side inner product vector $\{\hat{w}\}_{k=0}^{N}$ using the strategy selected in (3). The output is a vector denoting the inner product of the square of the input polynomial projected against the basis functions.

5.2 Aliasing Estimator

Following [2], we propose the following aliasing indicator. Let f denote the operator of interest. Our aliasing indicator is as follows:

(1) Compute $\mu = \mathcal{G}_Q[f]$.

(2) Compute $A_1 = \mathcal{K}_{2Q-1}[(f-\mu)^2]$ and $A_2 = \mathcal{G}_Q[(f-\mu)^2]$ denoting a type of "variance".

(3) If $|A_1 - A_2|/|A_1| > tolerance$ do consistent integration using Kronrod quadrature otherwise do linear-consistent integration using Gaussian quadrature.

5.3 Theoretical Cost Analysis

Here we try to find the percentage of using linear quadrature that yields the theoretical cost of the adaptive scheme equivalent to the cost of consistent integration. We perform the analysis for the quadrilateral and triangular element types. We consider the cost of evaluating the aliasing indicator to be zero.

If we denote the cost of forming the right hand side vector using linear integration as $\bar{C}_{lin}$, the same cost using the Kronrod approach as $\bar{C}_k$ and the cost of evaluating the function at the Kronrod points and the non-linear operator together as C_{fix} then we have the following formula for the cost of the adaptive scheme:

$$C_{adapt} = \zeta \bar{C}_{lin} + (1-\zeta)\bar{C}_k + C_{fix}. \tag{17}$$

We denote the cost of the consistent integration as C_{con}. So, in order to have $C_{adapt} =$

C_{con}, we get the following value for ζ:

$$\zeta = \frac{C_{con} - \bar{C}_k - \bar{C}_{fix}}{\bar{C}_{lin} - \bar{C}_k}. \tag{18}$$

.

5.3.1 Quadrilateral elements

To determine each individual cost mentioned above we consider first a quadrilateral type element. Furthermore, like before, we denote the cost of evaluating the polynomial at the quadrature points as C_{item_1}, the cost of evaluating the non-linear operator as C_{item_2} and the cost of forming the right hand side inner product as C_{item_3}.

Using the matrix-vector multiply approach, we have

- $C_{con} = C_{item_1} + C_{item_2} + C_{item_3}$, $Q_0 = Q_1 = 3N/2 + 2$:

$$\begin{aligned}
C_{item_1} &= Q_0 Q_1 M_0 M_1 \mu + Q_0 Q_1 (M_0 M_1 - 1)\alpha, \\
&= (3N/2+2)^2 (N+1)^2 \mu + (3N/2+2)^2 ((N+1)^2 - 1)\alpha. \\
C_{item_2} &= Q_0 Q_1 \mu, \\
&= (3N/2+2)^2. \\
C_{item_3} &= M_0 M_1 Q_0 Q_1 3\mu + M_0 M_1 (Q_0 Q_1 - 1)\alpha, \\
&= (N+1)^2 (3N/2+2)^2 3\mu + (N+1)^2 ((3N/2+2)^2 - 1)\alpha.
\end{aligned} \tag{19}$$

- $\bar{C}_k = C_{item_3}$, $Q_0 = Q_1 = 2(N+2) - 1 = 2N + 3$:

$$\begin{aligned}
C_{item_3} &= M_0 M_1 Q_0 Q_1 3\mu + M_0 M_1 (Q_0 Q_1 - 1)\alpha, \\
&= (N+1)^2 (2N+3)^2 3\mu + (N+1)^2 ((2N+3)^2 - 1)\alpha.
\end{aligned} \tag{20}$$

- $C_{fix} = C_{item_1} + C_{item_2}$, $Q_0 = Q_1 = 2(N+2) - 1 = 2N + 3$:

$$\begin{aligned} C_{item_1} &= Q_0 Q_1 M_0 M_1 \mu + Q_0 Q_1 (M_0 M_1 - 1)\alpha, \\ &= (2N+3)^2 (N+1)^2 \mu + (2N+3)^2 ((N+1)^2 - 1)\alpha. \\ C_{item_2} &= Q_0 Q_1 \mu, \\ &= (2N+3)^2. \end{aligned} \tag{21}$$

- $\bar{C}_{lin} = C_{item_3}$, $Q_0 = Q_1 = (N+2)$:

$$\begin{aligned} C_{item_3} &= M_0 M_1 Q_0 Q_1 3\mu + M_0 M_1 (Q_0 Q_1 - 1)\alpha, \\ &= (N+1)^2 (N+2)^2 3\mu + (N+1)^2 ((N+2)^2 - 1)\alpha. \end{aligned} \tag{22}$$

Substituting the above values into the equation (18), results in the values for ζ mentioned in Table 1. As you can see for some polynomial orders we get $\zeta > 1.0$. This means that for those cases our adaptive scheme is no beneficial comparing to the consistent integration. Also, for the sum-factorization approach we only get $\zeta < 1.0$ for the last 4 polynomial orders. Using the sum-factorization approach we have,

N	ζ	N	ζ
1	1.20	9	0.94
2	1.09	10	0.93
3	1.04	11	0.93
4	1.00	12	0.93
5	0.98	13	0.92
6	0.97	14	0.92
7	0.96	15	0.92
8	0.95	16	0.91

N	ζ	N	ζ
1	1.31	9	1.01
2	1.19	10	1.01
3	1.13	11	1.00
4	1.09	12	1.00
5	1.06	13	0.99
6	1.05	14	0.99
7	1.03	15	0.98
8	1.02	16	0.98

Table 1: ζ values when using matrix-vector multiply approach (left) and sum-factorization approach (right), for quadrilateral elements.

- $C_{con} = C_{item_1} + C_{item_2} + C_{item_3}$, $Q_0 = Q_1 = 3N/2 + 2$:

$$\begin{aligned}
C_{item_1} &= Q_1 M_0 M_1 \mu + Q_1 M_0 (M_1 - 1)\alpha \\
&+ Q_0 Q_1 M_0 \mu + Q_0 Q_1 (M_0 - 1)\alpha, \\
&= (3N/2+2)(N+1)^2\mu + (3N/2+2)(N+1)((N+1)-1)\alpha \\
&+ (3N/2+2)^2(N+1)\mu + (3N/2+2)^2((N+1)-1)\alpha. \\
C_{item_2} &= Q_0 Q_1 \mu, \\
&= (3N/2+2)^2\mu, \\
C_{item_3} &= Q_0 M_1 Q_1 2\mu + Q_0 M_1 (Q_1 - 1)\alpha \\
&+ M_0 M_1 Q_0 2\mu + M_0 M_1 (Q_0 - 1)\alpha, \\
&= (3N/2+2)(N+1)(3N/2+2)2\mu + (3N/2+2)(N+1)((3N/2+2)-1)\alpha \\
&+ (N+1)^2(3N/2+2)2\mu + (N+1)^2((3N/2+2)-1)\alpha
\end{aligned} \tag{23}$$

- $\bar{C}_k = C_{item_3}$, $Q_0 = Q_1 = (2N+3)$:

$$\begin{aligned} C_{item_3} &= Q_0 M_1 Q_1 2\mu + Q_0 M_1 (Q_1 - 1)\alpha \\ &+ M_0 M_1 Q_0 2\mu + M_0 M_1 (Q0 - 1)\alpha \\ &= (2N+3)(N+1)(2N+3)2\mu + (2N+3)(N+1)((2N+3)-1)\alpha \\ &+ (N+1)^2(2N+3)2\mu + (N+1)^2((2N+3)-1)\alpha \end{aligned} \tag{24}$$

- $C_{fix} = C_{item_1} + C_{item_2}$, $Q_0 = Q_1 = (2N+3)$:

$$\begin{aligned} C_{item_1} &= Q_1 M_0 M_1 \mu + Q_1 M_0 (M_1 - 1)\alpha \\ &+ Q_0 Q_1 M_0 \mu + Q_0 Q_1 (M_0 - 1)\alpha \\ &= (2N+3)(N+1)^2\mu + (2N+3)(N+1)((N+1)-1)\alpha \\ &+ (2N+3)^2(N+1)\mu + (2N+3)^2((N+1)-1)\alpha \\ C_{item_2} &= Q_0 Q_1 \mu \\ &= (2N+3)^2\mu \end{aligned} \tag{25}$$

- $\bar{C}_{lin} = C_{item_3}$, $Q_0 = Q_1 = (N+2)$:

$$\begin{aligned} C_{item_3} &= Q_0 M_1 Q_1 2\mu + Q_0 M_1 (Q_1 - 1)\alpha \\ &+ M_0 M_1 Q_0 2\mu + M_0 M_1 (Q_0 - 1)\alpha \\ &= (N+2)(N+1)(N+2)2\mu + (N+2)(N+1)((N+2)-1)\alpha \\ &+ (N+1)^2(N+2)2\mu + (N+1)^2((N+2)-1)\alpha \end{aligned} \tag{26}$$

Table 1, shows the corresponding ζ values.

5.3.2 Triangular elements

In this section we perform a similar analysis as the previous section to determine the values of ζ when using triangular shape elements. We again consider equal number of modes in both directions, i.e. $M_0 = M_1$.

Using matrix-vector multiply approach we have

- $C_{con} = C_{item_1} + C_{item_2} + C_{item_3}$, $Q_0 = Q_1 = 3N/2 + 2$:

$$\begin{aligned}
C_{item_1} &= M_{tot}Q_0Q_1\mu + (M_{tot} - 1)Q_0Q_1\alpha, \\
&= \frac{1}{2}(N+1)(N+2)(3N/2+2)^2\mu + (\frac{1}{2}(N+1)(N+2) - 1)(3N/2+2)^2\alpha. \\
C_{item_2} &= Q_0Q_1\mu, \\
&= (3N/2+2)^2. \\
C_{item_3} &= M_{tot}Q_0Q_1 4\mu + M_{tot}(Q_0Q_1 - 1)\alpha, \\
&= \frac{1}{2}(N+1)(N+2)(3N/2+2)^2 4\mu \\
&+ \frac{1}{2}(N+1)(N+2)((3N/2+2)^2 - 1)\alpha. \qquad (27)
\end{aligned}$$

- $\bar{C}_k = C_{item_3}$, $Q_0 = Q_1 = 2(N+2) - 1 = 2N + 3$:

$$\begin{aligned}
C_{item_3} &= M_{tot}Q_0Q_1 4\mu + M_{tot}(Q_0Q_1 - 1)\alpha, \\
&= \frac{1}{2}(N+1)(N+2)(2N+3)^2 4\mu + \frac{1}{2}(N+1)(N+2)((2N+3)^2 - 1)\alpha \qquad (28)
\end{aligned}$$

- $C_{fix} = C_{item_1} + C_{item_2}$, $Q_0 = Q_1 = 2(N+2) - 1 = 2N + 3$:

$$\begin{aligned} C_{item_1} &= M_{tot} Q_0 Q_1 \mu + (M_{tot} - 1) Q_0 Q_1 \alpha, \\ &= \frac{1}{2}(N+1)(N+2)(2N+3)^2 \mu + (\frac{1}{2}(N+1)(N+2) - 1)(2N+3)^2 \alpha. \\ C_{item_2} &= Q_0 Q_1 \mu, \\ &= (2N+3)^2. \end{aligned} \tag{29}$$

- $\bar{C}_{lin} = C_{item_3}$, $Q_0 = Q_1 = (N+2)$:

$$\begin{aligned} C_{item_3} &= M_{tot} Q_0 Q_1 4\mu + M_{tot}(Q_0 Q_1 - 1)\alpha, \\ &= \frac{1}{2}(N+1)(N+2)(N+2)^2 4\mu + \frac{1}{2}(N+1)(N+2)((N+2)^2 - 1)\alpha \end{aligned} \tag{30}$$

Substituting the above formulas into equation 18 yields the values for ζ mentioned in table 2. Again we see that for some polynomial orders our adaptive scheme is no better than the consistent integration in terms of theoretical costs estimates.

Using the sum-factorization approach we have,

N	ζ	N	ζ
1	1.12	9	0.88
2	1.02	10	0.87
3	0.97	11	0.88
4	0.94	12	0.86
5	0.92	13	0.86
6	0.90	14	0.86
7	0.89	15	0.86
8	0.865	16	0.85

N	ζ	N	ζ
1	1.18	9	0.93
2	1.07	10	0.92
3	1.02	11	0.91
4	0.99	12	0.91
5	0.97	13	0.90
6	0.95	14	0.90
7	0.94	15	0.90
8	0.93	16	0.90

Table 2: ζ values when using matrix-vector multiply approach (left) and sum-factorization approach (right), for triangular elements.

- $C_{con} = C_{item_1} + C_{item_2} + C_{item_3}$, $Q_0 = Q_1 = 3N/2 + 2$:

$$\begin{aligned}
C_{item_1} &= M_{tot}Q_1\mu + (M_{tot} - M_0)Q_1\alpha \\
&+ M_0Q_0Q_1\mu + (M_0 - 1)Q_0Q_1\alpha, \\
&= \frac{1}{2}(N+1)(N+2)(3N/2+2)\mu + (\frac{1}{2}(N+1)(N+2) - (N+1))(3N/2+2)\alpha \\
&+ (N+1)(3N/2+2)^2\mu + ((N+1)-1)(3N/2+2)^2\alpha. \\
C_{item_2} &= Q_0Q_1\mu, \\
&= (3N/2+2)^2\mu, \\
C_{item_3} &= Q_0M_0Q_1 3\mu + Q_0M_0(Q_1 - 1)\alpha \\
&+ M_{tot}Q_1 3\mu + M_{tot}(Q_1 - 1)\alpha, \\
&= (3N/2+2)(N+1)(3N/2+2)3\mu + (3N/2+2)(N+1)((3N/2+2)-1)\alpha \\
&+ \frac{1}{2}(N+1)(N+2)(3N/2+2)3\mu + \frac{1}{2}(N+1)(N+2)((3N/2+2)-1)\alpha \quad (31)
\end{aligned}$$

- $\bar{C}_k = C_{item_3}$, $Q_0 = Q_1 = (2N+3)$:

$$\begin{aligned} C_{item_3} &= Q_0 M_0 Q_1 3\mu + Q_0 M_0 (Q_1 - 1)\alpha \\ &+ M_{tot} Q_1 3\mu + M_{tot}(Q1 - 1)\alpha \\ &= (2N+3)(N+1)(2N+3)3\mu + (2N+3)(N+1)((2N+3)-1)\alpha \\ &+ \frac{1}{2}(N+1)(N+2)(2N+3)3\mu + \frac{1}{2}(N+1)(N+2)((2N+3)-1)\alpha \end{aligned} \quad (32)$$

- $C_{fix} = C_{item_1} + C_{item_2}$, $Q_0 = Q_1 = (2N+3)$:

$$\begin{aligned} C_{item_1} &= M_{tot} Q_1 \mu + (M_{tot} - M_0) Q_1 \alpha \\ &+ M_0 Q_0 Q_1 \mu + (M_0 - 1) Q_0 Q_1 \alpha \\ &= \frac{1}{2}(N+1)(N+2)(2N+3)\mu + (\frac{1}{2}(N+1)(N+2) - (N+1))(2N+3)\alpha \\ &+ (N+1)(2N+3)^2\mu + ((N+1)-1)(2N+3)^2\alpha \\ C_{item_2} &= Q_0 Q_1 \mu \\ &= (2N+3)^2\mu \end{aligned} \quad (33)$$

- $\bar{C}_{lin} = C_{item_3}$, $Q_0 = Q_1 = (N+2)$:

$$\begin{aligned} C_{item_3} &= Q_0 M_0 Q_1 3\mu + Q_0 M_0 (Q_1 - 1)\alpha \\ &+ M_{tot} Q_1 3\mu + M_{tot}(Q_1 - 1)\alpha \\ &= (N+2)(N+1)(N+2)3\mu + (N+2)(N+1)((N+2)-1)\alpha \\ &+ \frac{1}{2}(N+1)(N+2)(N+2)3\mu + \frac{1}{2}(N+1)(N+2)((N+2)-1)\alpha \end{aligned} \quad (34)$$

Table 2, shows the corresponding ζ values.

6 Discussion

The analysis in the previous section, shows that our adaptive scheme will not beat consistent integration for some polynomial orders and only does so if we use high order quadrature on just $10 - 15\%$ of the elements. We again mention that in all the analysis we considered the cost of evaluating the aliasing indicator to be zero. Taking that into account would not provide any improvement concerning the analytical cost estimates. The reason for the theoretical limitation is that we evaluate the function and the non-linear operator at the Lobatto-Kronrod points whether we do linear-consistent or Kronrod integration. Hence there is a constant cost introduced independent of which option is selected by the adaptivity indicator. Therefore, another approach would be to do the assessment (the extra cost that we pay to evaluate the function and the non-linear operator at all the Kronrod points) in one step and then be able to run with linear-consistent quadrature for a few times and then do another check. The reason one might consider this argument is that there are sometimes CFL restrictions for PDEs when the result for an element is already known and we might not need to do any check until the features have been advected across that element.

To simulate the above behavior we consider the following equation as the cost of the modified adaptive scheme,

$$C_{adapt} = \gamma(\zeta \bar{C}_{lin} + (1-\zeta)\bar{C}_k + C_{fix}) + (1-\gamma)(\zeta C_{lin} + (1-\zeta)C_k) \tag{35}$$

In equation (35), γ is a parameter in $[0, 1]$. When $\gamma = 1$ we get the equation mentioned in (17) and when $\gamma = 0$, it means there is no extra cost paid when doing only linear quadrature. The values for $\bar{C}_{lin}$, $\bar{C}_k$ and C_{fix} are mentioned in the previous section. Here we provide formulas for C_{lin} and C_k. We consider only the matrix-vector multiply approach and quadri-

lateral type elements. A similar analysis holds for the sum-factorization technique and triangular elements.

- $C_{lin} = C_{item1} + C_{item2} + C_{item3}$, $Q_0 = Q_1 = N + 2$:

$$\begin{aligned} C_{item_1} &= Q_0 Q_1 M_0 M_1 \mu + Q_0 Q_1 (M_0 M_1 - 1)\alpha, \\ &= (N+2)^2 (N+1)^2 \mu + (N+2)^2 ((N+1)^2 - 1)\alpha. \\ C_{item_2} &= Q_0 Q_1 \mu, \\ &= (N+2)^2. \\ C_{item_3} &= M_0 M_1 Q_0 Q_1 3\mu + M_0 M_1 (Q_0 Q_1 - 1)\alpha, \\ &= (N+1)^2 (N+2)^2 3\mu + (N+1)^2 ((N+2)^2 - 1)\alpha. \end{aligned} \tag{36}$$

- $C_k = C_{item1} + C_{item2} + C_{item3}$, $Q_0 = Q_1 = 2N + 3$:

$$\begin{aligned} C_{item_1} &= Q_0 Q_1 M_0 M_1 \mu + Q_0 Q_1 (M_0 M_1 - 1)\alpha, \\ &= (N+2)^2 (N+1)^2 \mu + (N+2)^2 ((N+1)^2 - 1)\alpha. \\ C_{item_2} &= Q_0 Q_1 \mu, \\ &= (N+2)^2. \\ C_{item_3} &= M_0 M_1 Q_0 Q_1 3\mu + M_0 M_1 (Q_0 Q_1 - 1)\alpha, \\ &= (N+1)^2 (N+2)^2 3\mu + (N+1)^2 ((N+2)^2 - 1)\alpha. \end{aligned} \tag{37}$$

Figure 4 shows the cost plots for the adaptive scheme using three different values of γ. The circles indicate the values of ζ where the cost of the adaptive quadrature equals the cost of the consistent integration (C_{con}). As you can see even with a value of γ as small as 0.1 we still need to perform linear quadrature close to 70% of the time for our adaptive scheme to

be competitive with the consistent integration.

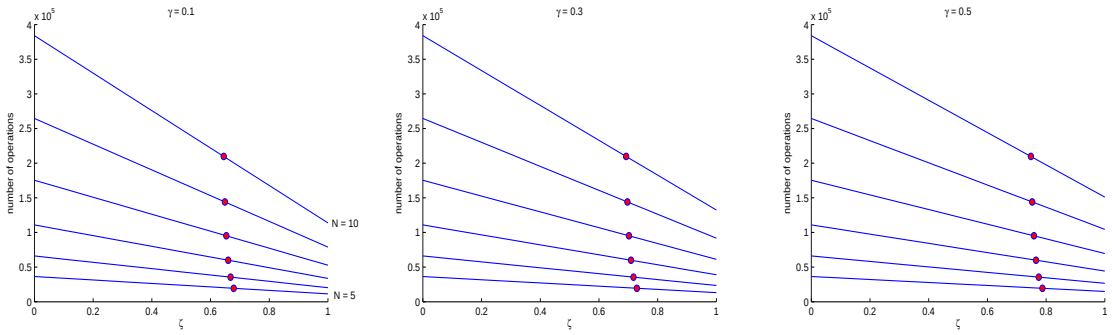

Figure 4: Number of operations vs. ζ for three different values of γ for $5 <= N <= 10$. Circles indicate the cost of the consistent integration.

Another possible approach to avoid the extra cost of evaluating the function and the non-linear operator at all the Kronrod points is to come up with a simpler aliasing indicator routine (in the current format, the indicator requires information form all the Kronrod points). Assuming that we have such an indicator then the adaptive algorithm would be,

(0) Input a vector of modal coefficients $\{\hat{u}_k\}_{k=0}^{N}$ representing a polynomial of at most degree N. Set the quadrature order to $Q = N + 2$.

(1) Evaluate the polynomial at Q Gauss-Lobatto points, producing a vector of function values $\{u(z_k)\}_{k=1}^{Q}$.

(2) Evaluate the non-linear differential operator at the Q quadrature points, producing a vector $\{[u(z_k)]^2\}_{k=1}^{Q}$.

(3) Use an aliasing indicator to establish whether to use (a) linear-consistent or (b) Kronrod-consistent integration on the element. The aliasing indicator takes as input the vector

$\{[u(z_k)]^2\}_{k=1}^{Q}$ and returns a binary response denoting (a) or (b).

- if linear-quadrature,

  (4) Form the right-hand-side inner product vector $\{\hat{w}\}_{k=0}^{N}$ using Q Gauss points. The output is a vector denoting the inner product of the square of the input polynomial projected against the basis functions.

- if Kronrod-quadrature,

  (4) Evaluate the polynomial at the remaining $Q-1$ Kronrod points, producing a vector of function values $\{u(z_k)\}_{k=1}^{2Q-1}$.

  (5) Evaluate the non-linear differential operator at the remaining $Q-1$ Kronrod points, producing a vector $\{[u(z_k)]^2\}_{k=1}^{2Q-1}$.

  (6) Form the right-hand-side inner product vector $\{\hat{w}\}_{k=0}^{N}$ using $2Q-1$ Lobatto-Kronrod points. The output is a vector denoting the inner product of the square of the input polynomial projected against the basis functions.

However, as an example we consider the following calculations for $N = 10$. We assume in item (3) that we have decided to do Kronrod quadrature. Since we already have evaluated the function at 12×12 quadrature points and the total number of Kronrod points is 23×23, this means a further $23 \times 23 - 12 \times 12 = 405$ evaluation points are needed. Whereas, if we were only to consider consistent quadrature we would have required $17 \times 17 = 289$ which is of course less than 405. So, again it would be faster to recompute the entire element using the consistent quadrature.

7 Summary

In this report we have provided analytical estimates to understand the computational costs when evaluating quadratic non-linearities on one-dimensional and two-dimensional spectral/hp discretisations using different quadrature approaches with the goal of presenting an efficient strategy for taming aliasing errors. In particular, we have considered an adaptive scheme that dynamically adapts the level of quadrature based upon an aliasing indicator and compared that with traditional consistent integration approach. Hierarchical Gauss-Kronrod quadrature was used, allowing for both error estimation and consistent integration of quadratic non-linearities at a single set of points which have as their subset the classic integration points.

With the help of the aliasing indicator, our adaptation strategy determines whether to use consistent (Kronrod) or linear quadrature on an element. We provided theoretical computational results that give the percentage of elements on which consistent integration can be performed such that the adaptive quadrature is competitive with consistent integration everywhere in the domain.

Our studies showed that for our adaptive scheme to be efficient we are required to use linear quadrature on more than 85% of elements assuming quadrilateral and triangular element types. This is due to the extra cost of evaluating the function and the non-linear operator at a larger number of Gauss-Kronrod points on all elements regardless of performing linear or consistent quadrature.

Alternatively, we analyzed two different modifications to our initial adaptation algorithm. In the first approach in which we are not paying the extra cost for all the elements, we are

still required to perform linear quadrature on 70% of the elements even when we are only evaluating the function and the non-linear operator at all the Gauss-Kronrod points on 10% of the elements. In the second strategy, we assumed we had an aliasing indicator for which we do not require the values of the functions at all the Gauss-Kronrod points. We again concluded that it is more efficient to recompute the entire domain using consistent quadrature.

References

[1] M. Abramowitz, I. Stegun, Handbook of Mathematical Functions, Dover, 1972.

[2] J. Berntsen, T. O. Espelid, Error estimation in automatic quadrature routines, ACM transactions on Mathematical Software 17 (1991) 233–252.

[3] D. Calvetti, G. Golub, W. Gragg, L. Reichel, Computation of gauss-kronrod quadrature rules, Math. Comput. 69 (2000) 1035–1052.

[4] D. Calvetti, G. Golub, W. Gragg, L. Reichel, Computation of gauss-kronrod quadrature rules, Math. Comput. 69 (2000) 1035–1052.

[5] C. Canuto, M. Hussaini, A. Quarteroni, T. Zang, Spectral Methods in Fluid Mechanics, Springer-Verlag, New York, 1987.

[6] C. Canuto, A. Quarteroni, Approximation results for orthogonal polynomials in sobolev spaces, Mathematics of Computation 38 (157) (1982) 67–86.

[7] B. S. Carmo, S. J. Sherwin, P. W. Bearman, R. H. J. Willden, Wake transition in the flow around two circular cylinders in staggered arrangements, J. Fluid Mech.

[8] M. Deville, E. Mund, P. Fischer, High Order Methods for Incompressible Fluid Flow, Cambridge University Press, 2002.

[9] D. Gottlieb, S. Orszag, Numerical analysis of spectral methods: theory and applications, in: SIAM-CMBS,Philadelphia, 1977.

[10] D. Hunter, G. Nikolov, On the error term of symmetric Gauss-Lobatto quadrature formulae for analytic functions, Math. Comput. 69 (2000) 269–282.

[11] G. Karniadakis, S. Sherwin, Spectral/hp Element Methods for CFD, Oxford University Press, 1999.

[12] G. Karniadakis, S. Sherwin, Spectral/hp element methods for CFD - 2^{nd} Edition, Oxford University Press, UK, 2005.

[13] R. M. Kirby, G. E. Karniadakis, De-aliasing on non-uniform grids: Algorithms and applications, J. Comp. Phys. 191 (2003) 249–264.

[14] R. M. Kirby, S. J. Sherwin, Aliasing errors due to quadratic nonlinearities on triangular spectral/*hp* element discretisations, J. Eng. Math. 56 (2006) 273–288.

[15] D. Laurie, Calculation of gauss-kronrod quadrature formulas, Math. Comput. 66 (1997) 1133–1145.

[16] D. Laurie, Calculation of radau-kronrod and lobatto-kronrod quadrature formulas, Numer Algor 45 (2007) 139–152.

[17] S. Orszag, On the elimination of aliasing in finite-difference schemes by filtering high wavenumber components, J. Atmos. Sciences 28 (1971) 1074.