Journal of Computational Physics 283 (2015) 408–413

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Short note

Short note on the mass matrix for Gauss–Lobatto grid points

Saul A. Teukolsky

Departments of Physics and Astronomy, Space Sciences Building, Cornell University, Ithaca, NY 14853, United States

ARTICLE INFO

Article history:
Received 16 September 2014
Received in revised form 20 November 2014
Accepted 6 December 2014
Available online 12 December 2014

Keywords:
Mass matrix
Gauss–Lobatto quadrature
Spectral methods
Discontinuous Galerkin methods
Finite element methods

ABSTRACT

The mass matrix for Gauss–Lobatto grid points is usually approximated by Gauss–Lobatto quadrature because this leads to a diagonal matrix that is easy to invert. The exact mass matrix and its inverse are full. We show that the exact mass matrix *and* its inverse differ from the approximate diagonal ones by a simple rank-1 update (outer product). They can thus be applied to an arbitrary vector in $O(N)$ operations instead of $O(N^2)$.

1. Motivation

With the increased emphasis on higher-order methods for solving partial differential equations, methods that divide the domain into subdomains and represent the solution as an expansion in basis functions have become more and more important. These include spectral element methods (penalty-based or continuous) and discontinuous Galerkin methods. To handle nonlinearities, collocation schemes are often the method of choice. In such methods, the expansion coefficients are replaced by function values at specially chosen grid points as the fundamental unknowns. In one dimension, the grid points are universally chosen to be the Gaussian quadrature points corresponding to the basis functions. This connects the expansion coefficients in spectral space to the function values in physical space by a discrete transform and leads to rapidly convergent and stable methods for smooth solutions.

In two and three dimensions, if the subdomains can be mapped to squares or cubes, then basis functions that are tensor products of one-dimensional basis functions are almost always used because of the resulting simplification of element-wise operations. Unless the problem requires the flexibility of grids constructed using triangles or tetrahedra, this approach is again almost universal. The key result of this note applies to any one-dimensional set of grid points that define a Gaussian quadrature or are part of a tensor product of such grid points. It does not apply to typical basis sets for triangles, where the quadrature rule and the choice of grid points are not directly connected.

For many problems, the simplest formulation uses Gauss–Lobatto collocation points since having grid points on the boundaries makes it easy to impose boundary conditions. In such a formulation, the exact mass matrix and its inverse are full. Thus it is natural to approximate the mass matrix by Gauss–Lobatto quadrature, which leads to a diagonal matrix that is easy to invert. By contrast, using Gauss collocation points with ordinary Gauss quadrature gives the exact mass matrix, which is diagonal. This makes the comparison between the two choices tricky. On the one hand, Gauss–Lobatto avoids interpolation from the interior points to the boundaries, but on the other hand it may require more collocation points

E-mail address: saul@astro.cornell.edu.

http://dx.doi.org/10.1016/j.jcp.2014.12.012
0021-9991/

to achieve the same accuracy as using Gauss points if you use the approximate mass matrix for efficiency. This point is discussed further in Section 4.3.

We show that there is a simple expression for the exact mass matrix and its inverse for Gauss–Lobatto collocation. Multiplying a vector by one of these expressions can be done in $O(N)$ operations, just as for a diagonal matrix. This suggests that efficiency versus accuracy results for implementations of spectral methods should be reconsidered. Of course, for large values of N the spectral convergence of Gaussian quadrature is likely to make the difference between the exact and approximate mass matrices irrelevant. However, for small or moderate N the situation is not clear.

2. Spectral approximation

This section summarizes some standard material [1–4] on spectral approximations in order to derive the key result in the next section.

Consider approximations of functions by expansions in orthogonal polynomials:

$$u(x) = \sum_{k=0}^{N} b_k p_k(x) \tag{1}$$

where

$$\int_{-1}^{1} p_j(x) p_k(x) W(x)\, dx = h_k \delta_{jk} \tag{2}$$

The associated inner product is

$$\langle u|v\rangle \equiv \int_{-1}^{1} u(x) v(x) W(x)\, dx \tag{3}$$

For simplicity, we will take the weight function $W(x) = 1$, in which case the basis functions are Legendre polynomials. However, almost everything in this note goes through for other systems of orthogonal polynomials.

The set of orthogonal polynomials determines a Gaussian quadrature formula with weights w_j and grid points x_j:

$$\int_{-1}^{1} f(x)\, dx \approx \sum_{j=0}^{N} w_j f(x_j) \tag{4}$$

The Gauss–Lobatto version of this quadrature arranges for the endpoints of the interval to be included in the set x_j. Having collocation points on the boundary can make the application of boundary conditions easier. The quadrature (4) is exact for polynomials of degree no more than $2N+1$ for the Gauss case and $2N-1$ for the Gauss–Lobatto case. Use the quadrature to define the discrete inner product as the analog of (3):

$$\langle u|v\rangle_{\mathrm{G}} = \sum_{j=0}^{N} w_j u(x_j) v(x_j) \tag{5}$$

The continuous and discrete inner products are the same if the product uv is a polynomial of degree no more than $2N+1$ (Gauss) or $2N-1$ (Gauss–Lobatto).

Eq. (1) is called a modal expansion. In collocation methods, instead of regarding the $N+1$ modal coefficients b_k as fundamental, we choose a set of $N+1$ collocation points x_j. Typically these are the Gauss or Gauss–Lobatto points associated with the orthogonal polynomials. The corresponding nodal expansion is

$$u(x) = \sum_{j=0}^{N} u_j \ell_j(x) \tag{6}$$

where $u_j \equiv u(x_j)$. The basis functions $\ell_j(x)$ are called cardinal functions and are simply the Lagrange interpolating polynomials based on the grid points x_j, with $\ell_j(x_i) = \delta_{ij}$:

$$\ell_j(x) = \prod_{\substack{i=0 \\ i\neq j}}^{N} \frac{x - x_i}{x_j - x_i} \tag{7}$$

The nodal expansion (6) is just an approximation of a continuous function $u(x)$ by its interpolating polynomial, so that $u(x_i) = u_i$. Note that in the discrete inner product (5) of any two continuous functions, we may replace u, say, by its

interpolating polynomial, since only the values u_j contribute to the sum. Thus with collocation methods we don't distinguish between a function and its expansion when using discrete inner products.

Since the discrete and continuous inner products are the same for polynomial integrands up to degree $2N-1$, the p_k's satisfy the discrete orthogonality condition

$$\langle p_j|p_k\rangle_{\rm G} = \gamma_k \delta_{jk} \tag{8}$$

where

$$\gamma_k \equiv \langle p_k|p_k\rangle_{\rm G} = \sum_{j=0}^{N} w_k p_k(x_j)^2 \tag{9}$$

For Gauss–Legendre quadrature,

$$\gamma_k = h_k = 2/(2k+1) \quad \text{(Gauss–Legendre)} \tag{10}$$

where h_k is the normalization defined in (2):

$$h_k \equiv \langle p_k|p_k\rangle \tag{11}$$

This is because the integration in (9) is exact since the degree of the polynomial in the integrand in (11) is $2N$.

For Gauss–Legendre–Lobatto (GLL) points, by contrast,

$$\gamma_k = \begin{cases} 2/(2k+1), & 0 \le k < N \\ 2/N, & k = N \end{cases} \tag{12}$$

In this case, the degree of exactness is only $2N-1$, and so $\gamma_N \neq h_N$. This simple fact is at the root of the "difficulties" of using GLL points.

In a collocation method, we regard the expansion (1) not as an independent alternative, but as the corresponding expansion of the interpolating polynomial (6). Evaluating the expression (1) at the grid points gives the relation

$$u_i = \sum_{k=0}^{N} b_k p_k(x_i) \tag{13}$$

This can be regarded as a discrete transform from spectral space, characterized by the representation b_k, to physical space, characterized by the u_i. To find the inverse transform, consider

$$\langle u|p_k\rangle_{\rm G} = \sum_{j=0}^{N} b_j \langle p_j|p_k\rangle_{\rm G} = \sum_{j=0}^{N} b_j \gamma_k \delta_{jk} = b_k \gamma_k \tag{14}$$

Thus

$$b_k = \frac{1}{\gamma_k}\langle u|p_k\rangle_{\rm G} = \frac{1}{\gamma_k}\sum_{j=0}^{N} w_j p_k(x_j) u_j \tag{15}$$

This is the transform from physical to spectral space.

The cardinal functions $\ell_j(x)$ are polynomials of degree N and so they can be expanded as

$$\ell_j(x) = \sum_{k=0}^{N} a_k p_k(x) \tag{16}$$

where by (15)

$$a_k = \frac{1}{\gamma_k}\sum_{i=0}^{N} w_i p_k(x_i)\ell_j(x_i) = \frac{1}{\gamma_k}\sum_{i=0}^{N} w_i p_k(x_i)\delta_{ij} = \frac{1}{\gamma_k} w_j p_k(x_j) \tag{17}$$

Substituting this in Eq. (16) gives

$$\ell_j(x) = w_j \sum_{k=0}^{N} \frac{1}{\gamma_k} p_k(x_j) p_k(x) \tag{18}$$

This expansion for the cardinal functions will be extremely useful in what follows.

3. Exact expressions for the mass matrix and its inverse

3.1. The mass matrix

The mass matrix is defined as

$$M_{ij} = \int_{-1}^{1} \ell_i(x)\ell_j(x)\,dx = \langle \ell_i | \ell_j \rangle \tag{19}$$

Here we have taken the range of x to be $[-1, 1]$. The derivation below goes through even when a weight function $W(x) \neq 1$ is included in (19).

Evaluating the mass matrix by Gaussian quadrature gives a diagonal matrix:

$$M_{ij} = \sum_{k=0}^{N} w_k \ell_i(x_k)\ell_j(x_k) = \sum_{k=0}^{N} w_k \delta_{ik}\delta_{jk} = w_i \delta_{ij} \tag{20}$$

This expression is exact for Gaussian quadrature, but not for the Gauss–Lobatto case because the integrand is of degree $2N$.

Let's derive an exact expression for the Lobatto case. Substituting expression (18) for the cardinal functions gives

$$\begin{aligned}
M_{ij} &= \sum_{k=0}^{N}\sum_{l=0}^{N} w_i w_j \frac{1}{\gamma_k \gamma_l} p_k(x_i) p_l(x_j) \int_{-1}^{1} p_k(x) p_l(x)\,dx \\
&= \sum_{k=0}^{N}\sum_{l=0}^{N} w_i w_j \frac{1}{\gamma_k \gamma_l} p_k(x_i) p_l(x_j) \delta_{kl} h_k \\
&= \sum_{k=0}^{N} w_i w_j \frac{h_k}{\gamma_k^2} p_k(x_i) p_k(x_j) \\
&= \sum_{k=0}^{N} w_i w_j \frac{1}{\gamma_k} p_k(x_i) p_k(x_j) + \left(\frac{h_N}{\gamma_N^2} - \frac{1}{\gamma_N}\right) w_i w_j p_N(x_i) p_N(x_j) \qquad (21) \\
&= w_i \ell_j(x_i) + \left(\frac{h_N}{\gamma_N^2} - \frac{1}{\gamma_N}\right) w_i w_j p_N(x_i) p_N(x_j) \\
&= w_i \delta_{ij} + \alpha w_i w_j p_N(x_i) p_N(x_j) \qquad (22)
\end{aligned}$$

where we have defined

$$\alpha = \frac{h_N - \gamma_N}{\gamma_N^2} \tag{23}$$

Eq. (21) follows from the previous line because $\gamma_k = h_k$ for $k < N$.

Eq. (22) reduces to Eq. (20) if $\gamma_N = h_N$, as for Gauss points. But we see that for the Lobatto case, where it is convenient to use the diagonal expression (20) because it is easy to invert, we introduce an error because of our "quadrature crime". Since the error in applying the approximate mass matrix to a vector converges away spectrally fast for smooth problems as we increase N, it is customary to ignore this error because of the other benefits of Lobatto points. However, there is no need to do this: the extra term in Eq. (22) is proportional to the outer product of the vector $w_i p_N(x_i)$ with itself. This means that in applying the mass matrix to a vector in a matrix–vector multiply, the extra term can be computed as a dot product of $w_j p_N(x_j)$ with the vector and then a scaling of the vector $\alpha w_i p_N(x_i)$ by the dot product. The operation count is $O(N)$, the same as from the diagonal term $w_i \delta_{ij}$.

More importantly, the inverse of the mass matrix is equally simple, as we now show.

3.2. Inverse mass matrix

The inverse of the mass matrix follows from the Sherman–Morrison formula:

$$(\mathbf{A} + \mathbf{u} \otimes \mathbf{v})^{-1} = \mathbf{A}^{-1} - \frac{(\mathbf{A}^{-1} \cdot \mathbf{u}) \otimes (\mathbf{v} \cdot \mathbf{A}^{-1})}{1 + \mathbf{v} \cdot \mathbf{A}^{-1} \cdot \mathbf{u}} \tag{24}$$

In our case,

$$\mathbf{A} = \mathrm{diag}(w_i), \qquad u_i = w_i p_N(x_i), \qquad v_i = \alpha u_i \tag{25}$$

We find

$$M_{ij}^{-1} = \frac{1}{w_i}\delta_{ij} + \beta p_N(x_i) p_N(x_j), \qquad \beta \equiv -\frac{h_N - \gamma_N}{\gamma_N h_N} \tag{26}$$

The simple form of the extra off-diagonal term in Eq. (26) makes it easy to use the exact inverse in applications. Once again, applying the inverse matrix to a vector is an $O(N)$ operation.

4. Applications

4.1. The differentiation matrix

As a trivial application, consider the differentiation matrix that appears when solving partial differential equations:

$$D_{ij} = \ell'_j(x_i) \tag{27}$$

where a prime denotes a derivative. The differentiation matrix typically appears via the stiffness matrix $\mathbf{S}$:

$$\mathbf{D} = \mathbf{M}^{-1} \cdot \mathbf{S} \tag{28}$$

where

$$S_{jk} = \int_{-1}^{1} \ell_j(x)\ell'_k(x)\,dx = \langle \ell_j | \ell'_k \rangle = \langle \ell_j | \ell'_k \rangle_{\mathrm{G}} \tag{29}$$

The last equality follows since the degree of the polynomial in the integrand of (29) is $2N-1$. So carrying out the quadrature gives the exact result

$$S_{jk} = \sum_m w_m \ell_j(x_m)\ell'_k(x_m) = \sum_m w_m \delta_{mj}\ell'_k(x_m) = w_j \ell'_k(x_j) \tag{30}$$

It is well known (e.g., [5]) that if the approximate mass matrix (20) is used in (28), one gets the exact result for the differentiation matrix:

$$\sum_j \left(\mathbf{M}_{\mathrm{GLL}}^{-1}\right)_{ij} S_{jk} = \sum_j \frac{1}{w_i}\delta_{ij} w_j \ell'_k(x_j) = \ell'_k(x_i) = D_{ik} \tag{31}$$

But why exactly do we get the right answer without using the exact mass matrix? One way of seeing this is to show explicitly that the "extra" terms in (26) give no additional contribution:

$$\sum_j \beta p_N(x_i) p_N(x_j) S_{jk} \propto \sum_j p_N(x_j) w_j \ell'_k(x_j) = \langle p_N | \ell'_k \rangle_{\mathrm{G}} = 0 \tag{32}$$

Here the quadrature gives zero by orthogonality because the degree of ℓ'_k is less than the degree of p_N.

4.2. Projection in hp-refinement

An advantage of methods like the DG method is that it is relatively straightforward to implement adaptive mesh refinement, including full *hp*-refinement. With refinement, there are two methods for communicating fluxes across subdomain faces: interpolation and projection. Interpolation is simpler, but for marginally resolved problems the inherent aliasing can lead to an instability. Moreover, interpolation does not guarantee conservation and so can be less robust than projection, especially for problems with shocks.

A convenient way to implement projection is with mortars, auxiliary slices inserted at boundary interfaces. A full discussion with explicit formulas is given in [6,7]. We note here that when projecting the solution from the subdomain to the mortar to be able to compute the flux, one gets the exact projection matrix using Gauss–Lobatto quadrature even when using the approximate mass matrix. The proof of this result is similar to that of Section 4.1: the extra outer product terms give no contribution. The resulting expression then shows that the result is the *same* as using interpolation. It is only when transferring the flux back from the mortar to the subdomain that there is a difference between projection and interpolation, and only when the polynomial degree of the subdomain is less than that of the mortar. This observation can be used to greatly simplify the implementation of projection for DG fluxes as given, for example, in [6,7]. In retrospect, it is "obvious" that interpolation from a coarse grid to a finer one introduces no aliasing, and so projection should be the same as interpolation, but this fact has not been used before in the literature on *hp*-refinement, to the best of my knowledge.

4.3. Efficiency of finite element methods

The question of whether to use Gauss points or Gauss–Lobatto points in spectral collocation methods is not always clear-cut, especially for small or moderate numbers of grid points. Gauss points typically require interpolation to impose boundary conditions, but their higher degree of exactness may allow a smaller number of points to be used for a given accuracy. There have been a number of studies of this question [8,9]. Kopriva and Gassner [8] concluded that for a simple linear wave equation, the two approximations have comparable efficiency, but for a nonlinear steady-state example Gauss approximation was faster for a desired error. Bassi et al. [9] concluded that Gauss nodes have "a clear advantage" for steady-state problems, and ascribed this to under-integration with Gauss–Lobatto nodes. As already mentioned in Section 1, this question should be re-examined in the light of the result of this paper, which allows the exact mass matrix or its inverse to be used efficiently in the Lobatto case.

4.4. Dispersion and dissipation

For wave propagation problems, dispersion and dissipation errors are important properties of any numerical scheme. Gassner and Kopriva [10] showed that the error introduced by using the approximate Gauss–Lobatto mass matrix can be interpreted as a modal filter applied to the highest polynomial mode (since $\gamma_N \neq h_N$). This filtering greatly increases the dispersion and dissipation errors compared to the Gauss case. It would be worthwhile to re-examine this question with the exact mass matrix.

4.5. Roundoff errors

It has been noted in [11] that different ways of computing the terms in a spectral element or DG method can affect roundoff errors as N increases. It may be worth examining whether the numerical behavior is affected by the different ways of computing the mass matrix.

Acknowledgements

I thank Jan Hesthaven for several helpful comments, including pointing out that interpolation from a coarse grid to a finer one introduces no aliasing. This work was supported in part by NSF Grants PHY-1306125 and AST-1333129 at Cornell University, and by a grant from the Sherman Fairchild Foundation.

References

[1] J.S. Hesthaven, S. Gottlieb, D. Gottlieb, Spectral Methods for Time-Dependent Problems, Cambridge University Press, Cambridge, UK, 2007.
[2] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods: Fundamentals in Single Domains, Springer, Berlin, 2006.
[3] J.P. Boyd, Chebyshev and Fourier Spectral Methods, 2nd edition, Dover Publications, New York, 2001.
[4] B. Fornberg, A Practical Guide to Pseudospectral Methods, Cambridge University Press, Cambridge, UK, 1996.
[5] M.H. Carpenter, D. Gottlieb, Spectral methods on arbitrary grids, J. Comput. Phys. 129 (1996) 74–86.
[6] D.A. Kopriva, A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method, J. Comput. Phys. 128 (1996) 475–488.
[7] D.A. Kopriva, S.L. Woodruff, M.Y. Hussaini, Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method, Int. J. Numer. Methods Eng. 53 (2002) 105–122.
[8] D.A. Kopriva, G. Gassner, On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods, J. Sci. Comput. 44 (2) (2010) 136–155.
[9] F. Bassi, N. Franchina, A. Ghidoni, S. Rebay, A numerical investigation of a spectral-type nodal collocation discontinuous Galerkin approximation of the Euler and Navier–Stokes equations, Int. J. Numer. Methods Fluids 71 (2013) 1322–1339.
[10] G. Gassner, D.A. Kopriva, A comparison of the dispersion and dissipation errors of Gauss and Gauss–Lobatto discontinuous Galerkin spectral element methods, SIAM J. Sci. Comput. 33 (5) (2011) 2560–2579.
[11] S. Gottlieb, J.-H. Jung, Numerical issues in the implementation of high order polynomial multi-domain penalty spectral Galerkin methods for hyperbolic conservation laws, Commun. Comput. Phys. 5 (2009) 600–619.