Exact calculation of Fourier series in nonconforming spectral-element methods

Aimé Fournier

NCAR Institute for Mathematics Applied to Geosciences
Boulder CO 80307-3000 USA

Key words: adaptive mesh refinement, Fourier analysis, spectral-element method

PACS: 02.30.Nw, 02.60.Cb, 02.60.Jh, 02.70.Dh, 02.70.Jn

1 Usefulness of calculating Fourier series in the SEM

In this note is presented a method, given nodal values on multidimensional nonconforming spectral elements, for calculating global Fourier-series coefficients. This method is “exact” in that given the approximation inherent in the spectral-element method (SEM), no further approximation is introduced that exceeds computer round-off error. The method is very useful when the SEM has yielded an adaptive-mesh representation of a spatial function whose global Fourier spectrum must be examined, e.g., in dynamically adaptive fluid-dynamics simulations such as [7].

Email address: fournier@ucar.edu (Aimé Fournier).
URL: http://www.asp.ucar.edu/gtp/fournier (Aimé Fournier).
2 Derivation of an exact transform

Suppose we have some functional problem in a spatial domain \( \mathbb{D} := [-\pi, \pi]^d \) (possibly including toroidal geometry) and use coordinate transformations

\[
\vec{\vartheta}_k \quad \text{from} \quad \vec{\xi} \in \mathbb{E}_0 := [-1, 1]^d \quad \text{to} \quad \vec{x} \in \mathbb{E}_k
\]  

(1)

to partition \( \mathbb{D} = \bigcup_{k=1}^{K} \mathbb{E}_k \) by \( K \) elements \( \mathbb{E}_k := \vec{\vartheta}_k(\mathbb{E}_0) \) with disjoint \(^1\) interiors. Typically the SEM approximates the exact solution by its piecewise polynomial representation of degree \( P \):

\[
uex(\vec{x}) \approx u(\vec{x}) = \sum_{k=1}^{K} \sum_{\vec{\jmath} \in \mathbb{J}_k} u_{\vec{\jmath}, k} \phi_{\vec{\jmath}, k}(\vec{x}), \tag{2}\]

where \( \mathbb{J} := \{0, \ldots, P\}^d \) indexes the values \( u_{\vec{\jmath}, k} := u(\vec{x}_{\vec{\jmath}, k}) \) and nodes \( \vec{x}_{\vec{\jmath}, k} := \vec{\vartheta}_k(\vec{\xi}_{\alpha}) \) mapped from the \( d \)-dimensional Gauss-Lobatto-Legendre (GLL) quadrature nodes \( \vec{\xi}_{\alpha} := \xi_{\alpha} \in [-1, 1] \),

\[
\phi_{\vec{\jmath}, k}(\vec{x}) := \begin{cases} 
\phi_{\vec{\jmath}} \circ \vec{\vartheta}^{-1}_k(\vec{x}), & \vec{x} \in \mathbb{E}_k \\
0, & \vec{x} \not\in \mathbb{E}_k 
\end{cases} \tag{3}
\]

is the \( \vec{x}_{\vec{\jmath}, k} \)-interpolating piecewise-polynomial,

\[
\phi_{\vec{\jmath}}(\vec{\xi}) := \prod_{\alpha=1}^{d} \phi_{\vec{\jmath}, \alpha}(\xi_{\alpha}) \quad \text{and} \quad \phi_{\vec{\jmath}}(\xi) := \sum_{p=0}^{P} \hat{\phi}_{\vec{\jmath}, p} L_p(\xi) \tag{4}
\]

are \( \vec{\xi}_{\vec{\jmath}} \)- and \( \xi_{\vec{\jmath}} \)-interpolating polynomials, \( \hat{\phi}_{\vec{\jmath}, p} \equiv w_j L_p(\xi_j) / \sum_{j'=0}^{P} w_{j'} L_{p}(\xi_{j'})^2 \) is a Legendre coefficient [e.g., 4, (B.3.15)], \( \sqrt{p + \frac{1}{2}} L_p(\xi) \) is the orthonormal Legendre polynomial of degree \( p \) on \([-1, 1]\] and \( w_j \) is the GLL quadrature weight.

In many cases a physically interesting quantity is the global Fourier-series coefficient \( \hat{u}_q \) at integer wavenumber components \( q^\alpha \), usually approximated by

\(^1\) \( \mathbb{E}_k \cap \mathbb{E}_{k'} = \emptyset \) if \( k \neq k' \)
$M^d$-point trigonometric $d$-cubature in such manner as

$$
\hat{u}_q := \frac{1}{(2\pi)^d} \int_{\mathbb{D}} u(\vec{x}) e^{-i\vec{q} \cdot \vec{x}} \, dv(\vec{x}) \equiv \sum_{k=1}^{K} \sum_{\vec{m} \in M} \hat{\phi}_{\vec{j},k,\vec{q}} u_{\vec{m}},
$$

(5)

where

$$
\hat{\phi}_{\vec{j},k,\vec{q}} = \frac{1}{M^d} \sum_{\vec{m} \in M} \phi_{\vec{j},k}(\vec{x},\vec{m}) e^{-i\vec{q} \cdot \vec{x},\vec{m}} - E_{\vec{q}} \phi_{\vec{j},k},
$$

(6)

d$v(\vec{x}) := \prod_{\alpha=1}^{d} dx^{\alpha}$ is the volume differential and $M := \{1, \ldots M\}^d$ indexes trigonometric nodes $x_{\vec{m}}^{\alpha} := (2m^{\alpha}/M - 1)\pi$. Note whenever $\mathbb{D}$ is adaptively repartitioned there is an additional computation cost of $O(M^d)$ per node to use (2) to provide in (6) the values $\phi_{\vec{j},k}(\vec{x},\vec{m})$, as well as a $d$-cubature error [generalizing 3, theorem 4.7]

$$
E_{\vec{q}} \equiv \sum_{\vec{r} \in \mathbb{Z}^d \setminus \{\vec{0}\}} \hat{u}_{\vec{q}+\vec{r}}
$$

that in general converges no faster than $O(M^{-2})$, because $C^1$ discontinuities of (2) across element boundaries cause $|\hat{u}_q|$ to decay only as $O(|\vec{q}|^{-2})$. We discover a more accurate method by substituting (3) into (5) to yield

$$
\hat{\phi}_{\vec{j},k,\vec{q}} = \frac{1}{(2\pi)^d} \int_{E_k} e^{-i\vec{q} \cdot \vec{x}} \phi_{\vec{j} \circ \vec{a}_k^{-1}}(\vec{x}) \, dv(\vec{x})
$$

$$
\equiv \left( \frac{1}{(2\pi)^d} \int_{E_k} e^{-i\vec{q} \cdot \vec{a}_k(\vec{\xi})} \phi_{\vec{j}}(\vec{\xi}) \left| \frac{\partial \vec{a}_k}{\partial \vec{\xi}} \right| \, dv(\vec{\xi}) \right)
$$

$$
\equiv \left( \frac{1}{(2\pi)^d} \int_{E_0} e^{-i\vec{q} \cdot \vec{a}_k(\vec{\xi})} \left( \prod_{\alpha=1}^{d} \sum_{p=0}^{P} \phi_{\vec{j}^{\alpha,p}} L_{p}(\xi^{\alpha}) \right) \left| \frac{\partial \vec{a}_k}{\partial \vec{\xi}} \right| \, dv(\vec{\xi}) \right).
$$

In many applications, especially when $u$-structure rather than domain geometry is guiding the mesh adaption, each $E_k$ is a $d$-parallelepiped with center $\vec{a}_k$ and $d$ legs $2\vec{h}_k^{\alpha}$, so we have an affinity $\vec{a}_k(\vec{\xi}) := \vec{a}_k + \vec{h}_k \cdot \vec{\xi}$, where $\vec{h}_k^{\alpha}$ make up the columns of $\vec{h}_k$. Then we obtain

$$
\hat{\phi}_{\vec{j},k,\vec{q}} = \frac{1}{(2\pi)^d} \left| \vec{h}_k \right| \prod_{\alpha=1}^{d} \sum_{p=0}^{P} \phi_{\vec{j}^{\alpha,p}} \int_{-1}^{1} e^{-i\vec{q} \cdot \vec{h}_k^{\alpha} \xi} L_{p}(\xi) \, d\xi.
$$
Finally, recalling the classical identity [e.g., exercise 12.4.9] for the spherical Bessel function \( B_p(r) \) of the first kind,

\[
B_p(r) = \frac{ip}{2} \int_{-1}^{1} e^{-ir\xi} L_p(\xi) \, d\xi,
\]

we obtain

\[
\hat{\phi}_{j,k;\vec{q}} = \frac{1}{\pi d} \left| \vec{h}_k \right| e^{-i\vec{q}\cdot\vec{a}_k} \prod_{\alpha=1}^{d} \sum_{p=0}^{P} \hat{\phi}_{p,\vec{q}} i^{-p} B_p(\vec{q} \cdot \vec{h}_k^\alpha). \tag{8}
\]

Note that most expressions in (8) can be precomputed; objects that may vary during a dynamically adaptive computation, such as \( \vec{a}_k \) or \( \vec{h}_k^\alpha \), typically take values from a sparse set, e.g., a collection of powers of 2. The computation of (5) now incurs no additional error beyond that of (2). Also note, to generalize to the case \( P = P_k^\alpha \) is straightforward.

### 3 Accuracy of transform for 1D & 2D test cases

Equation (8) was implemented in MatLab\textsuperscript{®} and tested using known results for (5). The most immediate test follows from (7), namely

\[
\hat{u}_q^{ex} = \hat{L}_p(\cdot/\pi)_q = i^{-p} B_p(\pi q).
\]

In this case (5) was found to reproduce (7) to 12-16 digits for \( K = 1, P \leq 18 \), implying similar performance for any polynomial \( u(\vec{x}) \) in this range. The next test was to put \( u^{ex}(x) = \sin x \), or \( \hat{u}_q^{ex} = (\delta_{q,1} - \delta_{q,-1})/2i \). Since this is not a polynomial we should expect at best to see algebraic convergence w.r.t. \( K \) in a uniform meshing \( a_k = (k - 1)h_k - \pi, h_k = 2\pi/K \) and exponential convergence w.r.t. \( P \), as verified in Fig. 1. Note there is no need to test \( u^{ex}(x) = \sin rx \) for \( r > 1 \) because of scaling.

We conclude by examining three 2D tests with adaptive meshing in the fashion of [5], using MatLab\textsuperscript{®}. Fig. 2 confirms (5) in the case \[6, (19)]

\[
u^{ex}(\vec{x}) \equiv \sum_{\vec{q} \in \mathbb{Z}^2} e^{b_1 |q^1| + b_2 |q^2| + i\vec{q} \cdot \vec{f} \vec{x}}, \tag{9}\]
Fig. 1. Surface plot (blue low to red high) of \( \log_{10} \) relative r.m.s. error in (5) for \( u^{\text{ex}}(x) = \sin x \), vs \( \log_2 K \) and \( P \).

where \( b^\alpha = -\frac{2}{5} \) and \( \vec{l} \equiv \left( \begin{array}{c} l_1 \\ l_2 \end{array} \right) = (\begin{array}{c} 1 \\ 2 \end{array}) \) is a biperiodicity-preserving “rotation”. As expected, the red curve (connecting the \( |\hat{u}_q| \) peaks) shows a power-law decay in \( \vec{q} \)-space. Note, in this plot and those below the \( \vec{l} \)-operation helps instigate mesh adaption but has the consequence of leaving \( \vec{q} \) undersampled in \( \mathbb{Z}^2 \). In Fig. 3 is shown an initial condition \([6, (22)]\)

\[
\vec{u}^{\text{ex}}(0, \vec{x}) := -\vec{l} \sin \vec{l} \cdot \vec{x}
\]  
(10)

for the 2D Burgers eq. As expected, \( \hat{u}_q \) almost vanishes for \( \vec{q} \neq \pm \vec{l} \). Finally, at time \( t = 1.6037/\pi |l|^2 \) the analytic solution generalizing \([2, (2.5)]\) to 2D is shown in Fig. 4. As expected for the nearly \( C^0 \)-discontinuous fronts \( \perp \vec{l} \) seen at left, \( |\hat{u}_q|^1 \) decays slightly faster than \( O(|q|^{-1}) \) but only for wavevectors \( \vec{q}||\vec{l} \) (red curve).

References

[1] George Arfken, 1985: *Mathematical methods for physicists*, 3rd ed., Academic Press, 985 pp.
Fig. 2. Left, $u$ (9) over the spatial $\vec{x}$ domain, increasing from blue to red; yellow lines indicate element boundaries, black lines show nodes $\vec{x}_{jk}$ with $P = 5$. Right, surface plot of $|\hat{u}_q|$ from (5) vs $q^1$ and $q^2$.

Fig. 3. As in Fig. 2 but for the $t = 0$ state given by (10), in $K = 2^6$ elements.

[2] C. Basdevant, M. Deville, P. Haldenwang, J.M. Lacroix, J. Ouazzani, R. Peyret & P. Orlandi, 1986: Spectral and finite difference solutions of the Burgers equation. Computers & Fluids, 14, 23–41.

[3] John P. Boyd, 1989: Chebyshev & Fourier spectral methods, Springer, 793 pp.
Fig. 4. As in Fig. 3 but for $t = 1.6037/5\pi$.

[4] M.O. Deville, P.F. Fischer & E.H. Mund, 2002: *High-order methods for incompressible fluid flow*, Cambridge, 499 pp.

[5] A. Fournier and D. Rosenberg, 2005: “Multiresolution adaptive spectral-element modeling of geophysical fluid dynamics using GASpAR,” in preparation.

[6] Fournier, A., G. Beylkin and V. Cheruvu, 2005: Multiresolution adaptive space refinement in geophysical fluid dynamics simulation. *Lecture Notes Comp. Sci. Eng.*, **41**, 161–170.

[7] Rosenberg, D., A. Fournier, P.F. Fischer & A. Pouquet, 2005: Geophysical-astrophysical spectral-element adaptive refinement (GASpAR): Object-oriented $h$-adaptive code for geophysical fluid dynamics simulation. *J. Comp. Phys.*, submitted.