Available online at www.sciencedirect.com
ScienceDirect
Procedia Engineering 6 (2010) 19–27
Procedia Engineering
www.elsevier.com/locate/procedia

IUTAM Symposium on Computational Aero-Acoustics
for Aircraft Noise Prediction

Development of Arbitrary-Order Hermite Methods for Simulation and Analysis of Turbulent Jet Noise

Daniel Appelö[a], Tim Colonius[a], Thomas Hagstrom[b], Matthew Inkman[a]

[a]*Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125*
[b]*Department of Mathematics, Southern Methodist University, Dallas, TX 75275*

Abstract

In this short note a brief description of Hermite methods is given. Previous and ongoing development of arbitrary order Hermite methods for the simulation of turbulent jets is also presented. In addition we outline how Hermite methods can be hybridized with discontinuous Galerkin methods to handle boundary conditions in a straightforward way. © 2010 Published by Elsevier Ltd.

Keywords:

1. Introduction

New understanding of turbulence mixing noise has high scientific and technological value. Jet noise has traditionally been reduced by increasing the bypass ratio of turbofan engines, following Lighthill's [1] developments in the 1950's. This trend cannot be continued indefinitely without significant changes to the traditional airframe configuration and engine cycles. New theoretical and computational approaches are needed to find optimal designs.

In the intermediate and long time frames, Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS), respectively, *will* provide direct prediction of the sound generated by large-scale turbulence. DNS requires that all dynamically relevant scales of motion be directly resolved and that a portion of the radiated acoustic field be included in the computational domain. The acoustic radiation is several orders of magnitude smaller than energetic turbulent structures and high resolution methods are favored since they provide minimal dispersion and dissipation to both turbulent flow structures and their weak radiated sound [2].

The development of accurate algorithms [2], and increasing computational capability permit DNS of turbulent flows at increasingly higher Reynolds numbers. However as outlined in the recent DARPA report, [3], future exascale computing will require new algorithmic approaches. It is clear that the limiting factor in scientific computations today and more so tomorrow is the so called von Neumann bottleneck, i.e. limited data transfer rate between the processor and memory compared to the amount of memory. The following historical context taken from [3] illustrates the future need for algorithms with a high number of arithmetic operations per memory access, *"... the Cray XMP, the worlds fastest supercomputer in 1983-1985 ... could perform 2 fetches and 1 store ... as well as up to 2 floating-point mathematical operations, per cycle. ... the IBM BG/L, can accomplish 4 floating-point mathematical operations in 1 (much shorter) cycle yet requires around 100 such cycles to fetch just 1 argument from memory ... This explains why the XMP ... spend .. 50% of its time moving data ... but BG/L may spend 99% of its time moving data"*.

1877-7058 © 2010 Published by Elsevier Ltd.
doi:10.1016/j.proeng.2010.09.003

Efficient utilization of future computer architectures require algorithms that achieve accurate answers by operating many times on local data. Hermite methods have exactly this property. In addition they are easy to formulate and implement to arbitrary order on unbounded or periodic domains and with the hybridization approach, described below, on non rectangular domains with physical boundary conditions.

2. Hermite Methods

A brief description of how to approximate $u_t = u_x$ using Hermite methods with Taylor time series evolution or Runge-Kutta time evolution now follows. A complete analysis of Hermite methods for linear hyperbolic systems is presented in [4]. Descriptions of the application of Hermite methods to compressible flow can be found in [5, 6, 7].

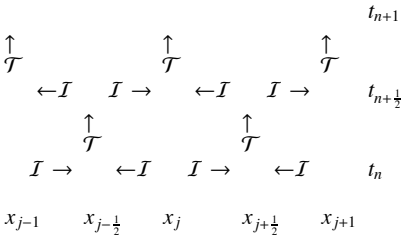

Figure 1: Schematic description of the numerical process for a full time step. Solid circles represent the base mesh and open circles represent the dual mesh. $\mathcal{I}$ is the Hermite interpolation operator and $\mathcal{T}$ is the time evolution operator.

2.1. Hermite Taylor Algorithm

1. At the start of the computation (time t_n) the initial data

$$u, \partial_x u, \ldots, \partial_x^m u, \text{ at all grid points} \ldots, x_{j-1}, x_j, x_{j+1}, \ldots.$$

are used to expand the approximate solution as a *local* power series (a Hermite polynomial) around $x_{j+1/2}$

$$p_{j+1/2}(x, t_n) = \sum_{l=0}^{2m+1} c_{l0} \, (x - x_{j+1/2})^l. \tag{1}$$

Note the simple relation between the polynomial coefficients and the derivative data:

$$p(x_{j+1/2}, t_n) = c_{00}, \;\; \partial_x p(x_{j+1/2}, t_n) = c_{10}, \;\; \ldots, \;\; (l!)^{-1} \, \partial_x^l p(x_{j+1/2}, t_n) = c_{l0}.$$

2. To advance the solution it is expanded as a Taylor series in time

$$p^n_{j+1/2}(x, t) = \sum_{l=0}^{2m+1} \sum_{s=0}^{2m+1-l} c_{ls} \, (x - x_{j+1/2})^l (t - t_n)^s. \tag{2}$$

If c_{ls} were known the solution could be advanced by evaluating $p^n_{j+1/2}(x, t)$ at $t = t_{n+1/2}$.

3. Depending on the equation of interest different recursions for c_{ls} are obtained. For example, for the transport equation

$$\partial_t u = \partial_x u, \tag{3}$$

taking $s - 1$ time derivatives of (3) gives

$$\partial_t^s u = \partial_t^{s-1} \partial_x u. \tag{4}$$

Replacing u by $p^n_{j+1/2}(x,t)$ in the above equality yields the recursion

$$s\, c_{ls} = (l+1)\, c_{l+1\, s-1}, \quad l = 0, \ldots, 2m+1-s. \tag{5}$$

As the coefficients c_{l0} are known from the initial data thus c_{ls}, $s > 0$ can be easily computed.

4. To advance the solution at the open circles to time $t_{n+\frac{1}{2}}$ (see, Fig. 1) simply evaluate

$$(l\,!)^{-1}\partial_x^l u(x_{j+1/2}, t_{n+1/2}) = \sum_{s=0}^{2m+1-l} c_{ls}\,(\frac{\Delta t}{2})^s, \quad l = 0, \ldots, m. \tag{6}$$

5. Repeat from 1 using the data on the staggered grid to advance the second half time step.

Runge-Kutta Time Stepping

For linear problems it is advantageous to evaluate all the terms in the series (6) resulting in a method of order $\sim (\Delta t^{2m+1} + \Delta x^{2m+1})$. For nonlinear and variable coefficient problems the recursion relation for the coefficients (5) becomes more involved and expensive to evaluate. In particular for large m and s higher order accurate Taylor methods do not justify the additional computational expense. Instead any explicit Runge-Kutta method that only uses the least expensive case, $s = 1$, should be used. For applications involving nonlinear problems we currently use 2-6 substeps of the standard 4th order Runge-Kutta method within each cell before communicating, see [7]. We note that the efficiency can be further enhanced by use of adaptive Runge-Kutta methods. We plan to incorporate adaptive time stepping in the future.

We emphasize that no matter how many stages in the Runge-Kutta formula or how many substeps we use, the time evolution over a global half-step is purely local to each cell. This fact is central to our claim that Hermite methods are particularly well-suited to current and future architectures. If, for example, methods of order 15 in space are employed, the local problem for a system with p variables is to independently evolve $p \cdot 8^3$ degrees-of-freedom in each cell over a time step which scales like 8 times the spatial density of the degrees-of-freedom. This is ideal for exploiting the memory hierarchy and minimizing communications between processor nodes. In addition, although we may use a formula with many stages, they are not stored globally. Thus the storage requirements are optimal, essentially only a single copy of the fields is needed, better by a factor of two or more in comparison with low storage approaches (e.g. [8]) no matter how high a temporal order is chosen.

3. Performance

To assess the performance of the Hermite-Runge-Kutta method we solve Euler's equations in two dimensions. The equations are reformulated so that only multiplicative nonlinearities appear, which results in a recursion relation (5) that is faster to evaluate. Thus we solve for specific volume, $r = \frac{1}{\rho}$, velocities, v_k, and pressure, p:

$$\frac{Dr}{Dt} - r\nabla\cdot v = 0, \quad \frac{Dv}{Dt} + r\nabla p = 0, \quad \frac{Dp}{Dt} + \gamma p \nabla\cdot v = 0, \quad (x,y) \in [-5,5]\times[-5,5], \quad t \geq 0. \tag{7}$$

Here D/Dt denotes the material derivative. The boundary conditions are taken to be periodic and the initial data are

$$\rho = 1 - \frac{A_v^2}{2} e^{1-x^2-y^2}, \quad v_1 = 1 - yA_v e^{\frac{1-x^2-y^2}{2}}, \quad v_2 = xA_v e^{\frac{1-x^2-y^2}{2}}, \quad p = 1 + \frac{\gamma A_v^2}{2} e^{1-x^2-y^2}, \tag{8}$$

with $A_v = 1$.

For the problem (7)-(8) solutions at time $t = 10$ (one "period") are computed for the Hermite-Taylor method and the Hermite-Runge-Kutta method with a time-step close to the CFL limit for two substeps. To find the error, computations using $m = 3, 7, 11$ are performed on a uniform 20×20 grid and compared to a reference solution computed on a much finer grid. In Table 1 we see that the six sub-step Runge-Kutta computation is 1.4 times faster than the Hermite-Taylor method (at roughly the same error level). We also see that the gain in using the Runge-Kutta time-stepping is even greater for larger m: 2.75 times faster for $m = 7$ and 19.5 times faster for $m = 11$. See [7] for further details.

Table 1: Results for the weak vortex (8), q/n_s refers to number of terms in the Taylor series or number of local sub-steps in the Runge-Kutta method and n_t is the number of global time-steps used.

Method	cpu-time	m	q/n_s	n_t	l_2 error
T	0.248(3)	3	8	80	1.3686(-6)
R	0.176(3)	3	6	80	1.3936(-6)
T	0.853(4)	7	16	90	1.3464(-7)
R	0.751(3)	7	2	130	1.7198(-6)
R	0.310(4)	7	2	540	4.1167(-8)
T	0.108(6)	11	24	130	9.6448(-8)
R	0.555(4)	11	2	250	1.3037(-7)

3.1. Scaling

As the Hermite methods have few communications per floating point operation we expect favorable scaling results on parallel architectures. In Table 2 below we see the results of a weak scaling study (fixed number of degrees of freedom per cpu) for our 3D Navier-Stokes solver on two machines with Myrinet networks maintained by UNM's High-Performance Computing Center. The results show perfect scaling. We expect the favorable weak scaling results, which are relevant to our application, will extend to larger values of m and problems of the scale of DNS of high Rejets. In Table 3 below we see the results of a strong scaling study for our 3D Navier-Stokes solver on Ranger at the Texas

Table 2: Weak scaling for the 3D Navier-Stokes code. Average number of micro seconds per timestep for fixed number of degrees of freedom ($\sim 8 \cdot 10^6$) per cpu.

# cpu		1	8	27	64	125
Ristra	μs/DOF	296	297	297	297	297
Nano	μs/DOF	197.0	197.3	197.6	–	–

Advanced Computation Center. Note that there are 16 cores per node, which explains the anomalously fast result for 8 cores.

Table 3: Strong scaling for the 3D Navier-Stokes code. Number of micro seconds per timestep for fixed number of degrees of freedom ($\sim 61 \cdot 10^6$).

# cpu		8	27	64	125	216
Ranger	μs/DOF/cpu	29	69	62	75	86

3.2. Optimization of Subroutines Operating on Polynomials

The computational work within each cell essentially entails two operations: (i) computation of the Hermite interpolant using the vertex data, and (ii) evaluation of terms in the partial differential equation at each stage of the Runge-Kutta process. In three space dimensions these steps involve a large number of operations performed independently on $p(2m+2)^3$ variables. Exploiting the tensor-product structure, the first step requires $O(m^2)$ multiplications by dense $m \times m$ matrices, $O(m^4)$ flops total. As this is only done once per step, it is the second operation which is the primary computational bottleneck.

In our current formulation the degrees-of-freedom are the spatial Taylor coefficients at the cell centers. The cost of evaluating terms in the equation is then highly dependent on their structure. For linear terms or product nonlinearities, the primary operation is to compute the product of polynomials. This can be accomplished using FFTs

in $O(m^3 \ln m)$ flops, leading to a total cost per cell of $O(n_{\text{substeps}} n_{\text{stages}} m^3 \ln m)$ flops. The favorable operation count for the FFT algorithm is only valid for m sufficiently large. The crossover point where the FFT-based algorithm is faster than a standard algorithm employing matrix multiplications will depend on the architecture. In Figure 3.2 a plot of the maximum errors for different m (for the strong vortex problem above) using each of a general purpose direct implementation, a hand tuned and optimized direct implementation of the polynomial multiplications and an implementation based on FFT are presented. For this particular machine, an eight core Opteron, optimized direct algorithms $m = 3$ or 4 are good choices.

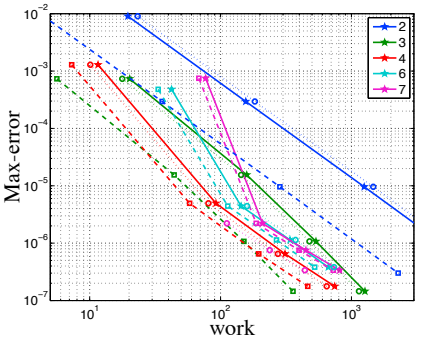

Figure 2: Maximum error after one period as a function of work for the strong vortex using the optimized (dashed), standard (solid) and fftw (dotted) versions of the *polyFun* library.

3.3. Order Adaptivity

Many of the flows of interest exhibit small regions where the fields vary rapidly, a prime example being the shear layers exiting the nozzle and bounding the potential core of a jet. Obviously, the efficiency of the simulations can be greatly improved through the use of automatic adaptivity which concentrates the degrees-of-freedom in these critical regions.

Hermite methods are very well-suited to order adaptive implementations (p-refinement). At each half step the local degree $2m + 1$ approximants are computed in Taylor form, and then updated to degree m. This immediately yields a truncation error estimate (the first term neglected in the update), as well as an estimate of the effect of both changing the degree and changing the mesh spacing. Chen and Hagstrom, [9], study order-adaptive implementations of Hermite methods for hyperbolic and singularly perturbed parabolic initial value problems and demonstrate its utility on a number of model problems posed in $1 + 1$ and $2 + 1$ dimensions.

3.4. Far Field Boundary Conditions and Absorbing Layers

We have also carried out successful tests of various damping layers and radiation boundary conditions for compressible flow calculations [5, 6, 7, 10]. Accurate domain truncation is particularly important for applications in aeroacoustics, as the true energy radiated to the far field, whose prediction is the primary object of the computation, is a minuscule fraction of the energy in the flow and can easily be masked by spurious reflections [11]. We expect that recent developments of far field boundary conditions, [12], and absorbing layers, [13], will mitigate such effects.

3.5. An Example With a Planar Two Dimensional Jet

As an example of a flow simulation we solve the full 2D compressible Navier-Stokes equations, using $m = 5$ and the classic fourth-order Runge-Kutta with 2 local substeps. The equations are solved on a uniform grid, centered around the origin, with gridspacing $h = 0.2$ in x and y. The global timestep is $dt = 0.067$.

To see the influence of the absorbing boundary conditions two cases are considered. The first consists of a single jet contained in a domain where all boundaries are truncated by supergrid absorbing layers [7, 10, 13], and for the second case the vertical boundary conditions are taken to be periodic, yielding an infinite array of jets.

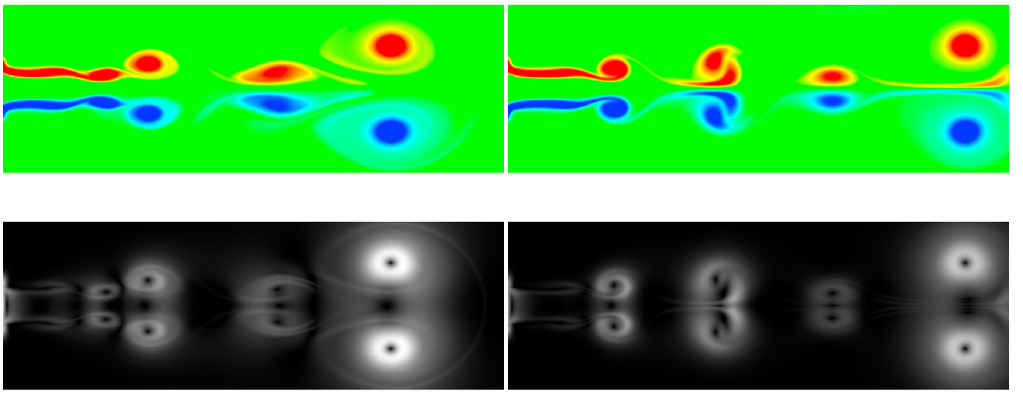

Figure 3: Top: Vorticity in pseudocolor with range -.5 (blue) to .5 (red). Bottom: Schlieren plots with levels ranging from 0 (black) to .05 (white). The left is for an array of jets, the right is for a single jet.

For the case of a single jet, the physical portion of the domain extends 100 gridpoints in x and 35 gridpoints in y, with an absorbing layer of width 10 gridpoints in x and 5 gridpoints in y at each domain boundary. For the array of jets, the physical domain extends 45 gridpoints in y. Simulations are conducted in the absence of freestream flow at acoustic Re = 5000. The flow is driven by adding a nonconservative forcing (see [14]):

$$f_x(r,x,t) = CF(t)\,G(x)\,H(r), \quad F(t) = -\tanh(\alpha_t(t_0 - t)) + \tanh(\alpha_t(t - t_0 - T)),$$
$$H(y) = \tfrac{1}{2}\mathrm{erfc}\Big(\sqrt{\alpha}\tfrac{|y|-R}{\alpha_r}\Big), \qquad G(x) = \frac{\sqrt{\alpha}}{\alpha_x\sqrt{\pi}}\exp\left(-\alpha\left(\frac{x}{\alpha_x}\right)^2\right).$$

Here the parameters are $t_0 = 4.5$, $T = 500$, $C = 0.01$, $R = 1.0$, $\alpha_r = \alpha_x = .2$, $\alpha_t = .3$ and $\alpha = 1.256$.

In Figure 3 we display schlieren and vorticity snapshots at $t = 242.67$ for both cases. Note that the absence of absorbing layers in y alters the flow patterns completely although the vorticity variations are concentrated around the centerline.

4. A hybrid Hermite-Runge-Kutta Discontinuous Galerkin Method for Initial Boundary Value Problems

As the Hermite methods use derivative data as degrees of freedom special care is needed to treat the boundary conditions, see [4]. This is somewhat involved and as an alternative we outline an approach to enforce the boundary conditions by coupling a nodal discontinuous Galerkin (DG) method [15], used to evolve the solution next to the boundary, to our Hermite-Runge-Kutta method.

For simplicity, consider a one-dimensional problem on a uniform grid $x_0, x_1 = x_0 + h, \ldots$, with a boundary at $x = x_0$. To handle the boundary conditions a Legendre-Gauss-Lobatto (LGL) grid is introduced between $x = x_0$ and $x = x_1$, see Figure 4. On this grid we discretize the equations at hand using a (single element) nodal discontinuous Galerkin method with an upwind flux for the boundary conditions. To the left we have the physical boundary conditions but to the right we must transfer data from the interior Hermite data to the element face. Figure 4 outlines the basic principles of the hybrid method. A full timestep starts by advancing the Hermite solution on the dual grid $x_{\frac{1}{2}}, x_{\frac{3}{2}}, \ldots$ using some number of local Runge-Kutta substeps (2 in the figure). At the start and at the end of every substep we extrapolate u and u_t to x_1 to find a fourth order accurate (interpolation) approximation of $u(x_1, t)$, see Figure 5. The approximation is used as boundary data for the time evolution, by a Runge-Kutta method (typically using a larger number of substeps), of the DG data. Once the Hermite and the DG data has been advanced to time $t_{n+\frac{1}{2}}$ the Hermite

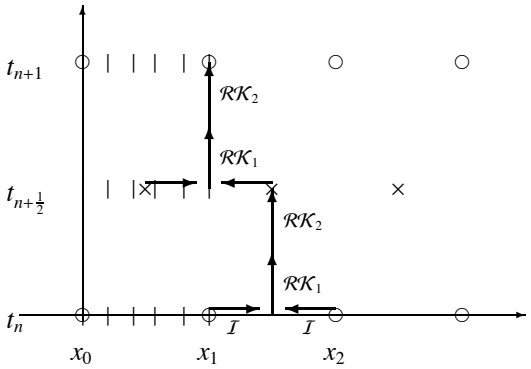

Figure 4: Schematic picture of the hybrid Hermite-DG method.

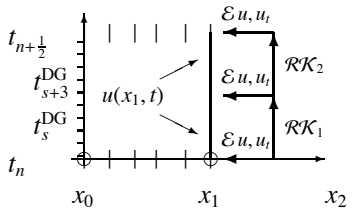

Figure 5: The boundary data for the DG-method along x_1 and for the first timestep is obtained by extrapolating u and u_t to x_1 at all Runge-Kutta substeps in $t \in [t_n, t_{n+1/2}]$ and constructing a fourth order accurate interpolant of u.

data at x_1 is filled in by differentiating the DG data, using the algorithm from Fornberg [16]. The second half step is similar to the first, the exception being that no extrapolation of u, u_t is needed.

4.1. A Numerical Experiment

To test the approach described above we solve

$$u_t = u_x, \;\; v_t = -v_x, \;\; t \geq 0, \;\; -1 \leq x \leq 1, \tag{9}$$

with boundary conditions

$$u(-1, t) = v(-1, t), \;\; u(1, t) = v(1, t),$$

and initial data:

$$u(x, 0) = v(x, 0) = \sin(12.5\pi x),$$

to time 40. The computations are done on a grid with $h = 0.1$. At the end we measure the maximum error in u and v. In Table 4 we present results for a 7th order accurate Hermite method ($m = 3$) combined with an 8th order accurate DG method. Both methods are advanced using the classic fourth order accurate Runge-Kutta method using different number of substeps. The results in Table 4 are obtained in the following way: for a fixed number of substeps of the Hermite and DG method (# RK-H & # DG-RK) the minimal number of global timesteps of size k is found and that error is tabulated. As can be seen, the CFL number k/h depends on both # RK-H and # DG-RK. The results indicate

Table 4: $m = 3, n_{DG} = 7$.

# RK-H	# DG-RK	k/h	k/h/(# RK)	Error	Rate
1	2	0.40	0.40	1.0 (0)	−−
1	3	0.63	0.63	4.5 (-1)	2.0
1	4	0.69	0.69	1.8(-2)	11.2
1	5	0.69	0.69	3.5(-3)	7.3
1	6	0.69	0.69	1.7(-3)	4.0
2	2	0.40	0.20	1.6 (-1)	−−
2	4	0.84	0.42	9.3 (-1)	-2.5
2	6	0.84	0.42	1.4 (-2)	10.3
2	8	0.84	0.42	8.5 (-4)	9.8
2	10	0.84	0.42	1.2 (-4)	8.8
3	3	0.60	0.20	1.1(0)	−−
3	6	0.93	0.31	4.7(-3)	7.9
3	9	0.93	0.31	1.1(-3)	3.5
3	12	0.93	0.31	5.6(-5)	10.3
4	4	0.81	0.20	3.8(-1)	−−
4	8	0.93	0.23	1.9(-3)	7.6
4	12	0.93	0.23	4.6(-5)	9.1
4	16	0.93	0.23	1.6(-5)	3.7

that a low multiple of # RK-H, 1-4, of # RK-DG is sufficient to realize the maximal CFL number for a given # RK-H. The error can be made even smaller by using more substeps in the DG element.

On the left in Figure 6 we have plotted the maximum error as a function of the number of global timesteps for a (# RK-H,# RK-DG)=(1,4), (2,4), (3,12), (marked as circles, stars, and squares) along with a solid line corresponding to a fourth order accurate error.

On the right in Figure 6 the maximum error is plotted as a function of scaled work. The plot shows that we can achieve the same error at the same cost for 1, 2 or 3 Hermite-Runge-Kutta substeps. That is, we can achieve the same error communicating two or three times less often.

5. Acknowledgment

The authors were funded by NSF grant 0905045.

[1] J. Lighthill, On sound generated aerodynamically I. General theory, Proc. Roy. Soc. Lond. A 211 (1952) 564–587.
[2] T. Colonius, S. Lele, Computational aeroacoustics: progress on nonlinear problems of sound generation, Progress in Aerospace Sciences 40 (2004) 345–416.
[3] E. P. Kogge, S. Lead, ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, Tech. rep., DARPA (2009).
[4] J. Goodrich, T. Hagstrom, J. L. and, Hermite methods for hyperbolic initial-boundary value problems, Math. Comp. 75 (254) (2005) 595–630.
[5] T. Hagstrom, J. Goodrich, I. Nazarov, C. Dodson, High-order methods and boundary conditions for simulating subsonic flows, in: 11th AIAA/CEAS Aeroacoustics Conference, no. 2005-2869 in AIAA, 2005.
[6] T. Hagstrom, J. Goodrich, G. Zhu, A Hermite-Taylor algorithm for simulating subsonic shear flows, Tech. Rep. 2006-2572, AIAA (2006).
[7] D. Appelö, T. Hagstrom, Experiments with Hermite methods for simulating compressible flows: Runge-Kutta time-stepping and absorbing layers, in: 13th AIAA/CEAS Aeroacoustics Conference, AIAA, 2007.
[8] C. Kennedy, M. Carpenter, R. Lewis, Low-storage explicit Runge-Kutta schemes for the compressible Navier-Stokes equations, Tech. Rep. 99-22, ICASE (1999).
[9] C. R., H. T., P-adaptive Hermite methods for initial value problems, submitted to MMAS, 2009.
[10] T. Colonius, H. Ran, A super-grid-scale model for simulating compressible flow on unbounded domains., J. Comput. Phys. 182 (1) (2002) 191–212.
[11] T. Colonius, Modeling artificial boundary conditions for compressible flow, Ann. Rev. Fluid Mech. 36 (2004) 315–345.
[12] T. Hagstrom, T. Warburton, Complete radiation boundary conditions: Minimizing the long time error growth of local methods, SIAM Journal on Numerical Analysis 47 (5) (2009) 3678–3704.

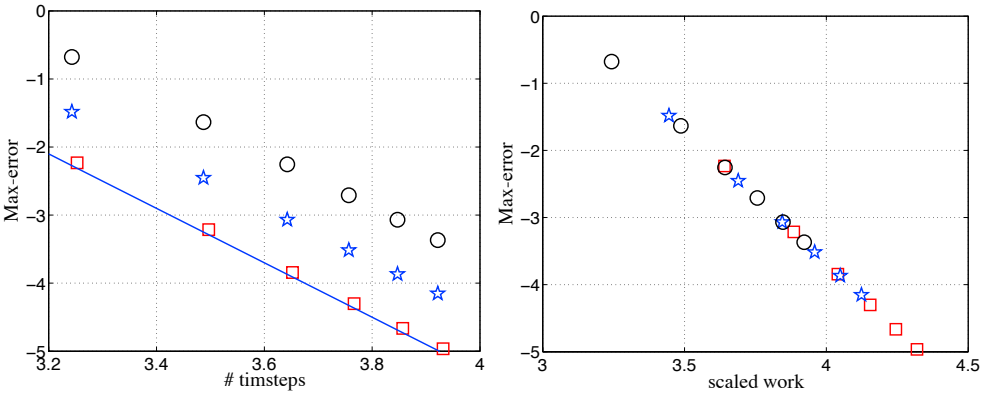

Figure 6: Left: maximum error as a function of the number of global timesteps for a (# RK-H,# RK-DG)=(1,4), (2,4), (3,12), marked as circles stars and squares. The solid line corresponds to a fourth order accurate error. Right: maximum error as a function of scaled work. We can achieve the same error communicating half or a third as often.

[13] D. Appelö, T. Colonius, A high-order super-grid-scale absorbing layer and its application to linear hyperbolic systems, Journal of Computational Physics 228 (11) (2009) 4200–4217.
URL http://www.sciencedirect.com/science/article/B6WHY-4VSB1H1-5/2/4f999773788177c0ff37b66b80e6686f
[14] R. H. . C. T. Mohseni, K., Numerical experiments on vortex ring formation, J. Fluid Mech. 430 (2001) 267–282.
[15] J. S. Hesthaven, T. Warburton, Nodal discontinuous Galerkin methods: algorithms, analysis, and applications, Vol. 54, Springer, New York, 2008.
[16] B. Fornberg, A practical guide to pseudospectral methods, Vol. 1, Cambridge University Press, Cambridge, 1996.
URL http://www.loc.gov/catdir/description/cam027/95031986.html