A Discontinuous Galerkin Spectral Element Method for the simulation of turbulent compressible flows

A. Beck, C. Altmann, G. Gassner, F. Hindenlang, M. Staudenmaier, C.-D. Munz

Institute of Aerodynamics and Gasdynamics, Universität Stuttgart

August 31, 2011
Content

Introduction and Overview

DG Spectral Element Method

Accuracy and Efficiency

High Performance Computing

Application to Turbulent Flows

Current Research
Introduction

- Work of research group “Numerical Methods of Fluid Dynamics” (Prof. Munz)
- Work funded in part by the DFG SPP 1276 “MetStröm”: Multiple Scales in Fluid Mechanics and Meteorology
- Subproject “Discrete-continuous hybrid models based on integral conservation laws“ in cooperation with Prof. Klein (FU Berlin) and Prof. Horenko (U Lugano)
  - Theoretical and practical development and implementation of scale-based closure concepts for underresolved simulations of multiscale phenomena (LES-like approach)
  - Investigation into interactions of numerical and modelling errors
Basic Idea and Motivation

- Very briefly: Just the basic idea
  - Underresolution of multiscale (turbulent) flows: range of scales and associated effects not captured by resolved solution field
  - Need to account for these effects by suitable model (closure)
  - Our approach: Interpret effect of missing scales as a flux correction term on resolved scales
  - Develop data-based stochastic models for flux correction term
  - Training of models by making them "learn" from DNS data
  - Highly efficient framework for the generation of "training data" (DNS of canonical turbulent flows) essential!

- We need an accurate (→ DGSEM formulation) and highly efficient (→ HPC aspects) DNS solver for canonical turbulent flows (→ Isotropic turbulence, shear layer, boundary layer) to train the model and to investigate the interactions of numerics, physics and model (→ current research)!
Problem Definition: The Compressible Navier Stokes Equations

- **Conservation law** of the form

\[ U_t + \nabla \cdot F(U, \nabla U) = 0 \]  \hspace{1cm} (1)

- Find \( U(x, t) = (\rho, \rho \mathbf{u}, \rho e)^T \) determined by eq. (1) + suitable initial and boundary conditions

- Physical flux is given by

\[ F(U, \nabla U) = F^A(U) - F^D(U, \nabla U), \]  \hspace{1cm} (2)

where

\[ F^A(U) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p I \\ \rho \mathbf{u} H \end{pmatrix} \]  \hspace{1cm} (3)

and

\[ F^D(U, \nabla U) = \begin{pmatrix} 0 \\ -\tau \\ \tau \cdot \mathbf{u} + k \nabla T \end{pmatrix}. \]  \hspace{1cm} (4)
DGSEM: Multi-Domain Decomposition

- Discretize space into non-overlapping hexahedra (or quads) with mapping \( x = X(\xi, \eta, \zeta) \)
  - with \( X_\xi \): Metrics of the mapping
  - with \( J \): Jacobian of the transformation
  - for curved elements: Mapping is a polynomial of arbitrary order

- Curved elements and complex geometries possible!
DGSEM: Approximation

- Transformed equations

\[ J U_t + \nabla_\xi \cdot \tilde{F} = 0 \quad \text{with} \quad \tilde{F}^i = \sum_{n=1}^{3} J a_n \tilde{F}_n \]  

- Tensor-product ansatz in reference element

\[ U(\xi, t)|_Q \approx U_h(\xi, t) := \sum_{i,j,k=0}^{N} (U_{ijk})(t) \psi_i(\xi_1)\psi_j(\xi_2)\psi_k(\xi_3) \]  

with \( \psi_i, \psi_j, \psi_k: 1D\)-Lagrange interpolating polynomials with \( \psi_i(\xi_j) = \delta_{ij} \)

- Approximation space is polynomial (inside a grid cell) but discontinuous across grid cell interfaces
DGSEM: Formulation

- Projection of transformed equation onto test function, partial integration yields DG-Formulation

\[
\langle J U_t, \psi \rangle_E + (\tilde{f}^* n_{\xi}, \psi)_{\partial E} - \langle \tilde{F}, \nabla_{\xi} \psi \rangle_E = 0
\]  

(7)

- Tensor-product structure of DGSEM operator

\[
\begin{aligned}
(f^{(\xi_1,1)} \xi^2) & f^{(1,\xi^2)} \\
\end{aligned}
\]

- Nodal basis functions with interpolation points=integration points

\[
(U_{ij})_t + \frac{1}{J_{ij}} \left[ \tilde{f}^* (1, \eta_j) \hat{\psi}_i(1) - \tilde{f}^* (-1, \eta_j) \hat{\psi}_i(-1) + \sum_{k=0}^{N} \hat{D}_{ik} \tilde{F}_{kj} \right] \\
+ \frac{1}{J_{ij}} \left[ \tilde{g}^* (\xi_i, 1) \hat{\psi}_j(1) - \tilde{g}^* (\xi_i, -1) \hat{\psi}_j(-1) + \sum_{k=0}^{N} \hat{D}_{jk} \tilde{G}_{ik} \right] = 0
\]  

(8)
DGSEM: Dispersion and Dissipation Properties

- Comparison of dispersion and dissipation behavior (normalized to DOF)

- Points per wavelength for given error

| $|\text{Re}(\Omega^*) - K|$ | Gauss(N=5) | Gauss-Lobatto(N=6) | cFD(O6) |
|---|---|---|---|
| 0.001 | 6.75 | 7.11 | 5.76 |
| 0.0001 | 8.22 | 8.88 | 7.93 |
| 0.00001 | 9.89 | 10.80 | 10.92 |
DGSEM: Efficiency

- Tensor product reduces work per DOF from $O(N^3)$ to $O(N)$!
- Specific CPU time
  (CPU time/(DOF * Runge-Kutta stage) for 3D compressible NSE)

<table>
<thead>
<tr>
<th>Method</th>
<th>spec. CPU time (Nehalem) [μs]</th>
</tr>
</thead>
<tbody>
<tr>
<td>cFD (O6)</td>
<td>4</td>
</tr>
<tr>
<td>Modal DG (N=5)</td>
<td>10</td>
</tr>
<tr>
<td>DGSEM (G, N=5)</td>
<td>2</td>
</tr>
<tr>
<td>DGSEM (GL, N=5)</td>
<td>1.6</td>
</tr>
</tbody>
</table>
High Performance Computing: Some remarks

- We use explicit time integration
- DG Operator is split in two parts
  - Volume parts: depend only on local data and are computationally expensive
  - Surface parts: depend also on neighbor data
- We are particularly interested in high order computations ($N > 5$), as we believe that this is efficient!
  - Higher accuracy per DOF (low dispersion and dissipation errors)
  - Data layout more compact
  - Operations more dense
  - Ratio of volume to surface data favorable
- We are interested in pushing parallelization and 'domain granularity' to the limit
- Our parallelization does not introduce extra operations, only MPI communication
- It is possible to hide communication latency by doing 'inner' work (volume parts) using non-blocking communication!
- We use "dim-by-dim" structure of operator to generate buffers for communication: Overlap message passing time and computation time
High Performance Computing: Parallel Scaling

- IBM Blue Gene system JUGENE (1 Petaflops peak)
  - Computing center Jülich
  - 294,912 processors!
  - Power PC 450 about a factor 5 slower than Intel Xeon (X5560) Nehalem!

- Weak and Strong Scaling

⇒ up to 87% scaling with only one element per processor on 131072 processors!! (sustained about 26 Teraflops)
Canonical Turbulence I: Taylor Green Vortex

- Analytic initialization of vortex field
- Homogeneous isotropic decaying turbulence
- Computation with $2.16 \times 10^8$ DOF and 200,000 time steps on 32k processors:
Canonical Turbulence I: Analysis

- Wide range of spatial and temporal scales
- High resolution numerics necessary!
  - Parallel analyze tools necessary!
  - Postprocessing and data management at least as important as actual computation itself!!
- Spectra of the kinetic energy (parallel FFT):

- Turbulent energy cascade with $\sim$Kolmogorov $-\frac{5}{3}$ range
 Canonical Turbulence II: Compressible Shear Layer Flow

- Parameters: $Ma_1 = 0.5$, $Ma_2 = 0.25$, $Re = 500$
- Blasius solution as initial condition
- Inflow forcing using eigenfunctions from linear stability theory
  - Distribution of z-component of vorticity

- Isosurfaces of $\lambda_2$ vortex detection criterium
Canonical Turbulence II: Comparison Results DGSEM and cFD O6

▶ Time Fourier Analysis of the flow field
▶ Compare maxima of $v$-velocity amplitudes
Comparison of computational effort

Computation on Nehalem cluster (Computing center in Stuttgart, HLRS)

<table>
<thead>
<tr>
<th></th>
<th>DGSEM (Gauss, N=5)</th>
<th>cFD O6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>424 × 144 × 2</td>
<td>2500 × 850 × 9</td>
</tr>
<tr>
<td>Points</td>
<td>2544 × 864 × 12</td>
<td>2500 × 850 × 9</td>
</tr>
<tr>
<td>DOF</td>
<td>26,376,192</td>
<td>19,125,000</td>
</tr>
<tr>
<td>Time Step</td>
<td>0.0138</td>
<td>0.00998</td>
</tr>
<tr>
<td>CPU Time</td>
<td>5056h</td>
<td>6215h</td>
</tr>
</tbody>
</table>

Framework as efficient and accurate as state-of-the-art Finite Difference code designed especially for DNS of canonical turbulence!

DGSEM ideally suited for HPC computations!
Current Research: Motivation

- Taylor-Green Vortex dissipation rate: LES and DNS-type runs

- Dissipation rate: Measure of physical dissipation through resolved scales
- High order vs. low order scheme: Significant gain in resolution quality at reduced cost for higher order schemes!
- High order LES looks promising
- → Benefit in high order for underresolved/LES-type turbulence simulations?
Current Research: High order DG for underresolved turbulence simulations

- Question: Spectral accuracy for DG for $\Delta h \rightarrow 0$: But what happens for large $\Delta h$ as in LES?

- A qualitative hint for same number of DOF (no SGS model!):

- Second order scheme too dissipative, high order scheme with superior accuracy but stability problem

- **Aliasing errors** in high order schemes need to be controlled!
Current Research: Stabilized high order schemes

- Two stabilization mechanisms: Filtering of higher modes or alias prevention by overintegration

Taylor-Green Vortex ($Re = 800$). Isocontours of $\lambda_2 = -1.5$ at $t = 8.5$. First plot shows 2nd order calculation with $64^3$ DOF. Second plot shows stabilized 16th order computation with $64^3$ DOF. Last plot shows the reference DNS result.
Current Research: Stabilized high order schemes

- Kinetic energy decay for increasing Reynolds number (more scales!)
- Comparison with state-of-the-art explicit and implicit LES-models for the same no. of DOF!

- Stabilized high order scheme at least as good!
- High order DG schemes a viable candidate for LES
Thank you for your attention!

Numerics Research Group at the Institute of Aerodynamics and Gasdynamics

→ www.iag.uni-stuttgart.de/nrg ←