Sparse Line-Based Discontinuous Galerkin Discretizations and Efficient Time-Integration

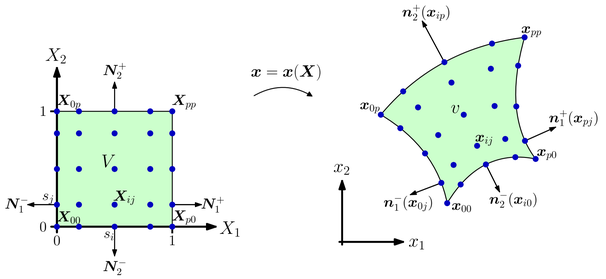
Per-Olof Persson, UC Berkeley and LBNL

Abstract: We present a line-based Discontinuous Galerkin (DG) method, which combines one-dimensional DG discretizations in a finite difference-type fashion. The result is a high-order accurate method for fully unstructured meshes, with computational cost similar to finite difference methods. In particular, the Jacobian matrices have between one and two magnitudes fewer connectivities than for standard nodal DG. This translates directly into a corresponding improvement in execution time for the numerical solvers.

Motivation and goals

- A nodal Galerkin approach typically couples all nodes inside an element, which gives a stencil size of $\mathcal{O}(p^D)$.
- A finite difference approach would apply difference approximations along each coordinate direction, with stencil size $\mathcal{O}(Dp)$.
- GOAL: study unstructured methods with similar stencils.
- LINE-BASED DG: apply 1D DG along each coordinate line.

Line-based discontinuous Galerkin method



Map system of conservation laws from an element v to the reference element V

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{u}) = \boldsymbol{0} \quad \longrightarrow \quad J\frac{\partial \boldsymbol{u}}{\partial t} + \nabla_{\boldsymbol{X}} \cdot \widetilde{\boldsymbol{F}}(\boldsymbol{u}) = \boldsymbol{0}$$

where $\widetilde{\boldsymbol{F}} = (\widetilde{f}_1, \widetilde{f}_2, \widetilde{f}_3) = JG^{-1}\boldsymbol{F}$,
$G = \nabla_{\boldsymbol{X}} \boldsymbol{x}$ and $J = \det(G)$.

Consider a curve $\boldsymbol{x}_{jk}(\xi) = \boldsymbol{x}(\xi, X_j, X_k)$.
Find $\boldsymbol{r}_{jk}(X_1) \approx \partial \widetilde{f}_1 / \partial X_1$ using a 1D DG procedure.
Find $\boldsymbol{r}_{jk}(\xi) \in \mathcal{P}_p([0,1])^m$ such that

$$\begin{aligned}\int_0^1 \boldsymbol{r}_{jk}(\xi) \cdot \boldsymbol{v}(\xi)\, d\xi &= \int_0^1 \frac{d\widetilde{f}_1(\boldsymbol{u}_{jk}(\xi))}{d\xi} \cdot \boldsymbol{v}(\xi)\, d\xi \\ &= \widehat{\widetilde{f}}_1(\boldsymbol{u}_{jk}^+(1), \boldsymbol{u}_{jk}(1)) \cdot \boldsymbol{v}(1) - \widehat{\widetilde{f}}_1(\boldsymbol{u}_{jk}(0), \boldsymbol{u}_{jk}^-(0)) \cdot \boldsymbol{v}(0) \\ &\quad - \int_0^1 \widetilde{f}_1(\boldsymbol{u}_{jk}(\xi)) \cdot \frac{d\boldsymbol{v}}{d\xi}\, d\xi.\end{aligned}$$

Note, that if $\boldsymbol{N} = (1,0,0)$**, then**

$$\widetilde{f}_1 = \widetilde{\boldsymbol{F}} \cdot \boldsymbol{N} = (JG^{-1}\boldsymbol{F}) \cdot \boldsymbol{N} = \boldsymbol{F} \cdot (JG^{-T}\boldsymbol{N}) = \boldsymbol{F} \cdot \boldsymbol{n}$$

with the (non-normalized) normal vector $\boldsymbol{n} = JG^{-T}\boldsymbol{N}$.
Therefore, the numerical flux with $N_1^+ = (1,0,0)$ can be written

$$\widehat{\widetilde{f}}_1(\boldsymbol{u}_R, \boldsymbol{u}_L) = \widehat{\widetilde{\boldsymbol{F}} \cdot \boldsymbol{N}_1^+}(\boldsymbol{u}_R, \boldsymbol{u}_L) = \widehat{\boldsymbol{F} \cdot \boldsymbol{n}_1^+}(\boldsymbol{u}_R, \boldsymbol{u}_L),$$

which involves the original fluxes $\boldsymbol{F}$ in direction $\boldsymbol{n}_1^+ \Longrightarrow$ **Use existing numerical flux functions without change**.

Find $\boldsymbol{r}_{jk}(\xi)$ by a standard finite element procedure.

$$\boldsymbol{u}_{jk}(\xi) = \sum_{i=0}^{p} \boldsymbol{u}_{ijk}\phi_i(\xi), \quad \boldsymbol{r}_{jk}(\xi) = \sum_{i=0}^{p} \boldsymbol{r}_{ijk}\phi_i(\xi)$$

Discrete form $M\boldsymbol{r}_{jk} = \boldsymbol{b}$, find $\boldsymbol{r}_{jk}$ by solving m linear systems with $(p+1)\times(p+1)$ mass matrix $\boldsymbol{M}$.

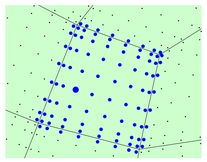
Repeat along each direction to obtain semi-discrete formulation,

$$J_{ijk}\frac{d\boldsymbol{u}_{ijk}}{dt} + \sum_{n=1}^{3} \boldsymbol{r}_{ijk}^{(n)} = \boldsymbol{0}.$$

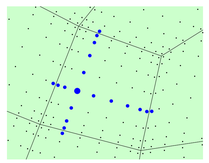
Observations:

- All integrals are one-dimensional.
- No statement about integration/flux points, as integrals are assumed to be exact.
- Numerical fluxes only evaluated point-wise.
- A continuous Galerkin FEM method would have connected neighboring elements globally.

Stencil size and sparsity pattern



Standard Nodal-DG



Line-DG

number of connectivities per node (3D hexahedra)

Polynomial order p	1	2	3	4	5	6	7	8	9	10
Line-DG connectivities	10	13	16	19	22	25	28	31	34	37
Nodal-DG connectivities	32	81	160	275	432	637	896	1215	1600	2057

For $p = 3$ the Line-DG method is **10 times sparser**, and for $p = 10$ it is more than **50 times sparser** than Nodal-DG.

Second Order Systems

For viscous terms, we use an LDG-type approach. Rewrite the system as a set of first order equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{u}, \boldsymbol{q}) = \boldsymbol{0}, \quad \nabla \boldsymbol{u} = \boldsymbol{q}$$

and define the numerical fluxes

$$\begin{aligned}\widehat{\boldsymbol{F} \cdot \boldsymbol{n}}(\boldsymbol{u}, \boldsymbol{q}, \boldsymbol{n}) &= \{\{\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{q}) \cdot \boldsymbol{n}\}\} + C_{11}[\![\boldsymbol{u} \otimes \boldsymbol{n}]\!] + \boldsymbol{C}_{12}[\![\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{q}) \cdot \boldsymbol{n}]\!] \\ \widehat{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{q}, \boldsymbol{n}) &= \{\{\boldsymbol{u}\}\} - \boldsymbol{C}_{12} \cdot [\![\boldsymbol{u} \otimes \boldsymbol{n}]\!] + C_{22}[\![\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{q}) \cdot \boldsymbol{n}]\!]\end{aligned}$$

If $C_{22} = 0$, $\boldsymbol{q}$ can be eliminated locally.
If $\boldsymbol{C}_{12} = S_i^{\pm}\boldsymbol{n}_i^{\pm}/2$ for some *switch function* $S_i^{\pm} \in \{-1, 1\}$, the scheme gets a simple upwind/downwind structure.

Final semi-discrete form becomes

$$\begin{aligned}\frac{d\boldsymbol{u}_{ijk}}{dt} + \frac{1}{J_{ijk}}\sum_{n=1}^{3} \boldsymbol{r}_{ijk}^{(n)} &= \boldsymbol{S}(\boldsymbol{u}_{ijk}, \boldsymbol{q}_{ijk}), \\ \frac{1}{J_{ijk}}\sum_{n=1}^{3} \boldsymbol{d}_{ijk}^{(n)} &= \boldsymbol{q}_{ijk},\end{aligned}$$

where both $\boldsymbol{r}_{ijk}^{(n)}$ and $\boldsymbol{d}_{ijk}^{(n)}$ in general depend on $\boldsymbol{u}$ and $\boldsymbol{q}$.

Time-integration and Solvers

With solution vectors $\boldsymbol{U}, Q$ the system becomes

$$\frac{d\boldsymbol{U}}{dt} = \boldsymbol{R}(\boldsymbol{U}, Q), \qquad Q = \boldsymbol{D}(\boldsymbol{U}, Q).$$

If $C_{22} = 0$, then $\boldsymbol{D}(\boldsymbol{U}, Q) = \boldsymbol{D}(\boldsymbol{U})$ and

$$\frac{d\boldsymbol{U}}{dt} = \boldsymbol{R}(\boldsymbol{U}, \boldsymbol{D}(\boldsymbol{U}))$$

Implicit solvers require solution of equations of the form

$$(\boldsymbol{I} - \alpha\Delta t\boldsymbol{A})\Delta\boldsymbol{U} = \Delta t\boldsymbol{R}(\boldsymbol{U}, \boldsymbol{D}(\boldsymbol{U}))$$

with

$$\boldsymbol{A} = \frac{d\boldsymbol{R}}{d\boldsymbol{U}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} + \frac{\partial \boldsymbol{R}}{\partial Q}\frac{\partial \boldsymbol{D}}{\partial \boldsymbol{U}} = \boldsymbol{K}_{11} + \boldsymbol{K}_{12}\boldsymbol{K}_{21}.$$

Problem: $\boldsymbol{K}_{11}, \boldsymbol{K}_{12}, \boldsymbol{K}_{21}$ sparse but not $\boldsymbol{K}_{12}\boldsymbol{K}_{21}$
Various options for retaining sparsity:

1. Form Krylov matrix-vector products implicitly without forming $\boldsymbol{K}_{12}\boldsymbol{K}_{21}$:

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{K}_{11}\boldsymbol{p} + \boldsymbol{K}_{12}(\boldsymbol{K}_{21}\boldsymbol{p})$$

2. Solve in original split form,which requires the solution of systems involving the matrix

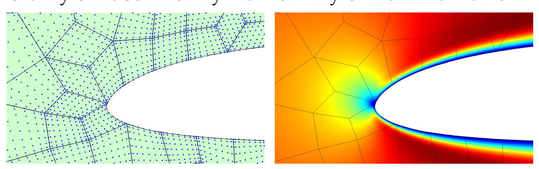
$$\boldsymbol{K} = \begin{bmatrix} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} & \frac{\partial \boldsymbol{R}}{\partial Q} \\ \frac{\partial \boldsymbol{D}}{\partial \boldsymbol{U}} & \frac{\partial \boldsymbol{D}}{\partial Q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{bmatrix}$$

3. Other approaches: subiterations, local timestepping, implicit-explicit, etc

Numerical results

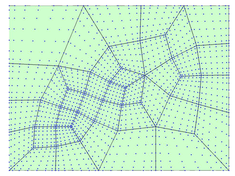
Laminar steady flow around airfoil

Flow around SD7003 foil , with Re $= 5000$, $p = 8$.
Steady-state solution by Newton-Krylov iterative method.

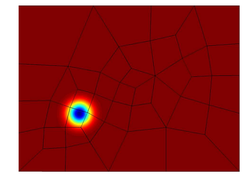


Euler vortex

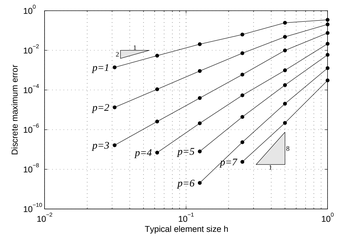
Convergence test for Euler vortex test problem
Optimal convergence of $\mathcal{O}(h^{p+1})$ observed.



Coarsest mesh, with $p = 7$

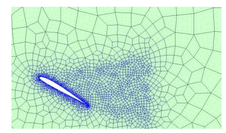


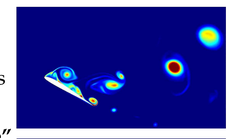
Solution (density)

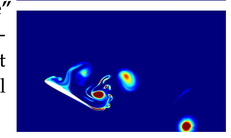


Unsteady LES-type flow around airfoil

Unsteady Navier-Stokes around SD7003 airfoil
Mach 0.2, $Re = 5,000$, AoA = 30°,
$p = 5$ in space, DIRK 3/3 in time,
GMRES + Jacobi preconditioning,
Quasi-Newton method: Re-use Jacobians between iterations and timesteps,
Essentially "explicit performance" for an implicit scheme: majority of compute time spent in the Newton residual evaluations.







Conclusions

A line-based discontinuous Galerkin method:

- Orders of magnitude smaller stencils than nodal-DG
- Very simple scheme structure, with 1D integrals and standard point-wise Riemann solvers
- LDG-type fluxes for viscous terms
- Optimal order of accuracy
- Efficient time-stepping by quasi-Newton

Current work includes:

- Efficient parallel solvers
- Shock capturing and RANS
- Extensions to other elements
- More applications

Department of Mathematics, Lawrence Berkeley National Laboratory. 1 Cyclotron Road, Berkeley, CA 94720, USA. Enquiries to: persson@berkeley.edu