On performance portability and library reuse: A Navier-Stokes miniapp

Valeria Barra[1], Jed Brown[1], Jeremy Thompson[1], Yohann Dudouit[2]

[1] University of Colorado at Boulder
[2] Lawrence Livermore National Lab

University of Colorado at Boulder (Computer Science Dept.)

ICIAM 2019, Valencia, Spain — July 17, 2019

CENTER FOR EFFICIENT EXASCALE DISCRETIZATIONS

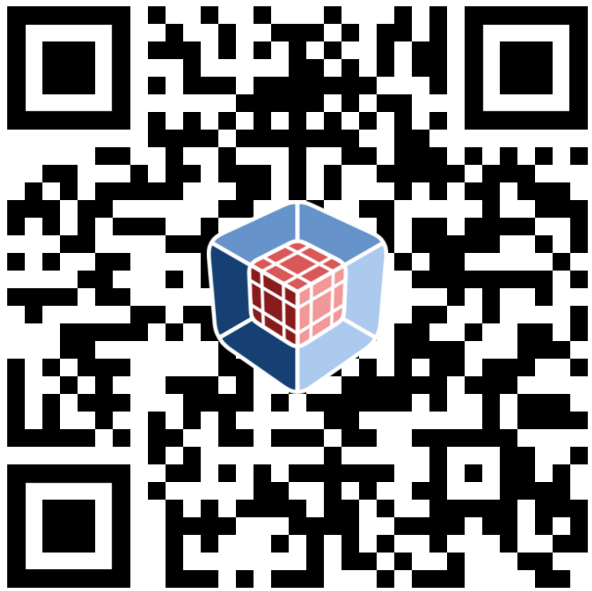
Overview

- We will introduce libCEED: the library of the CEED
- For decades, high-order numerical methods have been considered too expensive. But we now know that a sparse matrix is no longer a good representation for high-order operators
- libCEED uses a matrix-free operator description, based on a purely algebraic interface, where user only specifies action of weak form operators
- libCEED operator representation is optimal with respect to the FLOPs needed for its evaluation, as well as the memory transfer needed for operator evaluations (matvec)
  - For a 3D Jacobian operator, matrix-free operators that exploit tensor-product structures reduce the work load from $\mathrm{O}(\mathrm{p}^6)$ (for sparse matrix) to $\mathrm{O}(\mathrm{p}^4)$, and memory storage from $\mathrm{O}(\mathrm{p}^6)$ to $\mathrm{O}(\mathrm{p}^3)$
- We demonstrate the usage of libCEED with PETSc for a compressible Navier-Stokes solver

University of Colorado Boulder

libCEED: the library of the CEED (Center for Efficient Extensible Discretizations)

- Primary target: high-order finite element/spectral element methods exploiting tensor product structure
- Open source (BSD-2 license) C library with Fortran interface
- Releases: v0.1 (January 2018), v0.2 (March 2018), v.0.3 (September 2018), v0.4 (March 2019)



For latest release:
Brown J., Abdelfattah A., **Barra V.**, Dobrev V., Dudouit Y., Fischer P., et al., *CEED ECP Milestone Report: Public release of CEED 2.0* (2019, March 31) DOI: `http://doi.org/10.5281/zenodo.2641316`

University of Colorado Boulder

libCEED: the library of the CEED (Center for Efficient Extensible Discretizations)

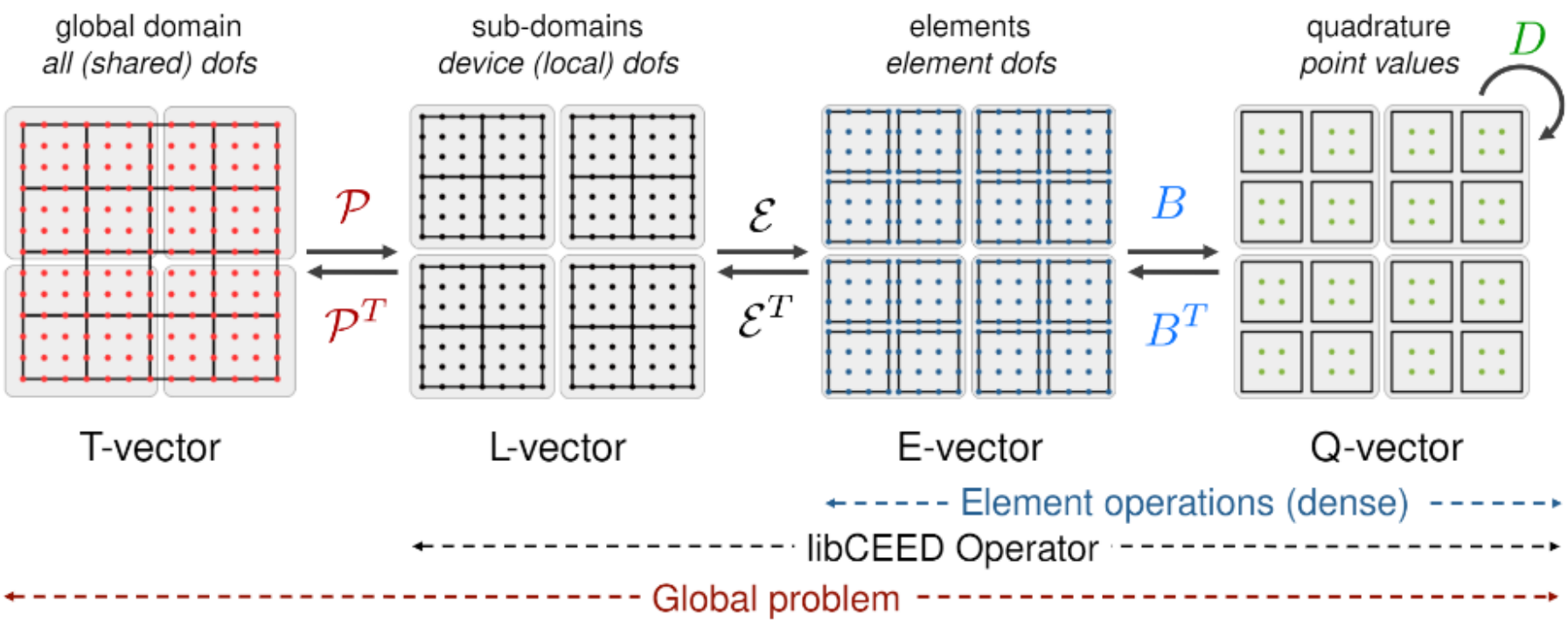
`/cpu/self/ref/*`:	with `*` reference serial and blcoked implementations
`/cpu/self/avx/*`:	AVX (Advanced Vector Extensions instruction sets) with `*` reference serial and blocked implementations
`/cpu/self/xsmm/*`:	LIBXSMM (Intel library for small dense/sparse mat-multiply) with `*` reference serial and blocked implementations
`/*/occa`:	OCCA (just-in-time compilation) with `*`: CPU, GPU, OpenMP (Open Multi-Processing: API), OpenCL (framework for CPUs, GPUs, etc.)
`/gpu/magma`:	MAGMA (dense Linear Algebra library for GPUs and multicore architectures)
`/gpu/cuda/*`:	CUDA with `*`: `ref` (reference pure CUDA kernels), `reg` (pure CUDA kernels using one thread per element), `shared`, optimized pure CUDA kernels using shared memory

Same source code can call multiple CEEDs with different backends. On-device operator implementation with unique interface

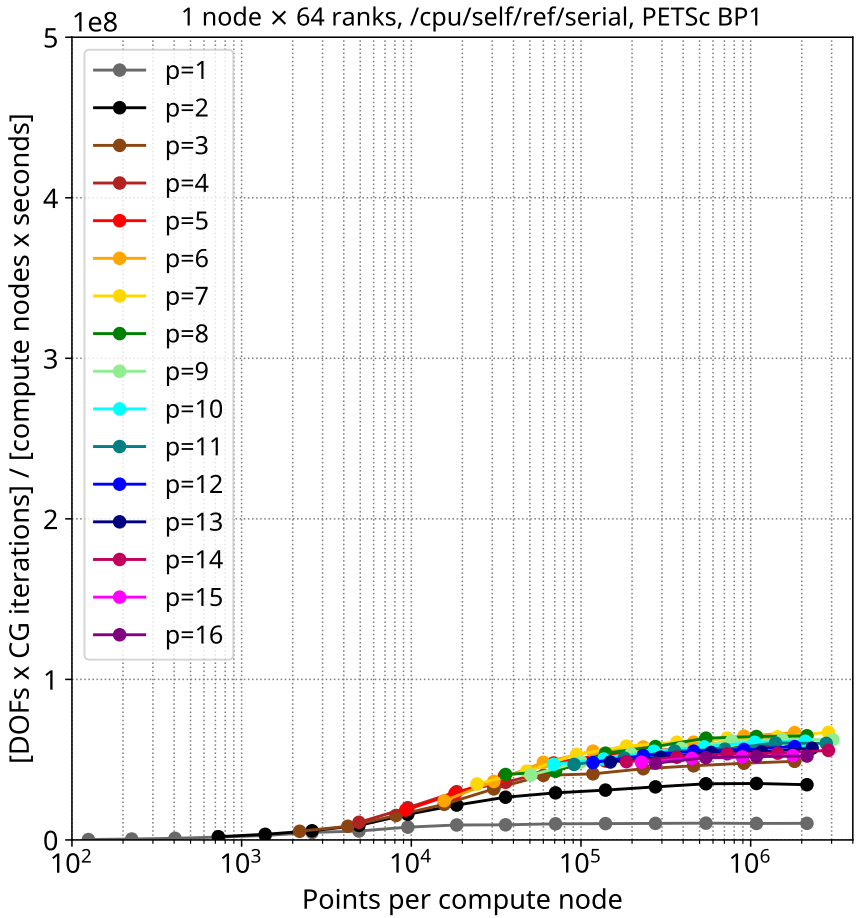
University of Colorado Boulder

libCEED decomposition

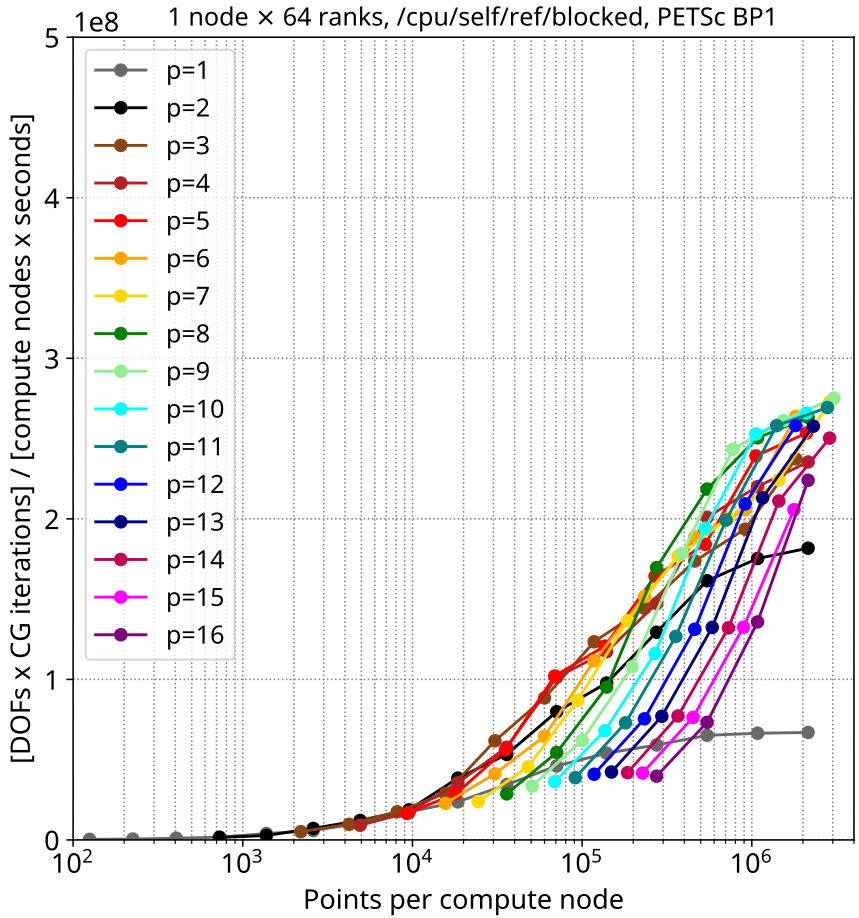
$$A = \mathcal{P}^T \mathcal{E}^T B^T D B \mathcal{E} \mathcal{P}$$



Performance w. r. t. space, on KNL



(a)



(b)

Figure: Knight Landing (Intel Xeon Phi 7230 SKU 1.3 GHz) with intel-18 compiler. In (a) serial implementation; in (b) blocked implementation ($q = P + 1$, $P = p + 1$)

University of Colorado Boulder

Performance w. r. t. time, on KNL

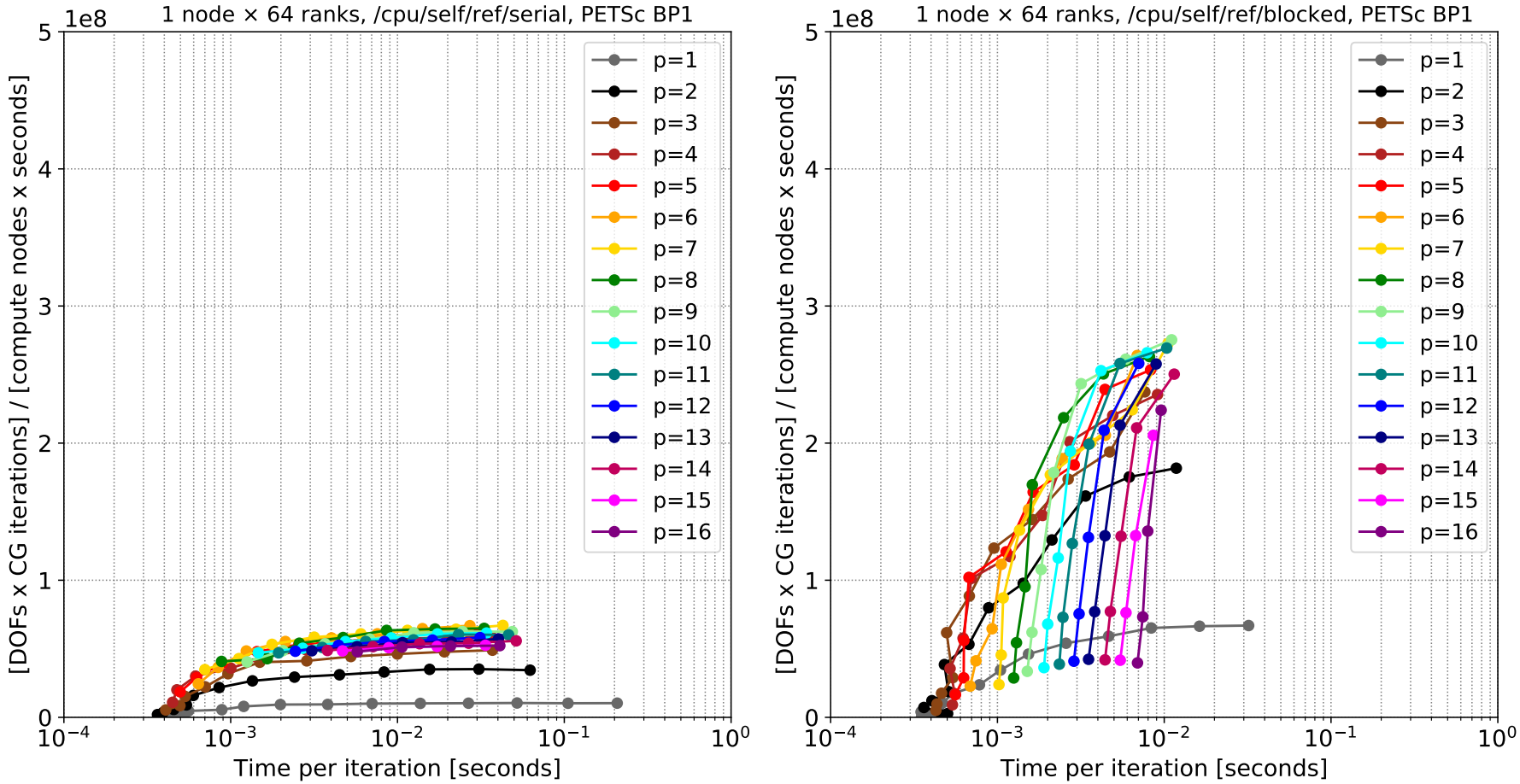
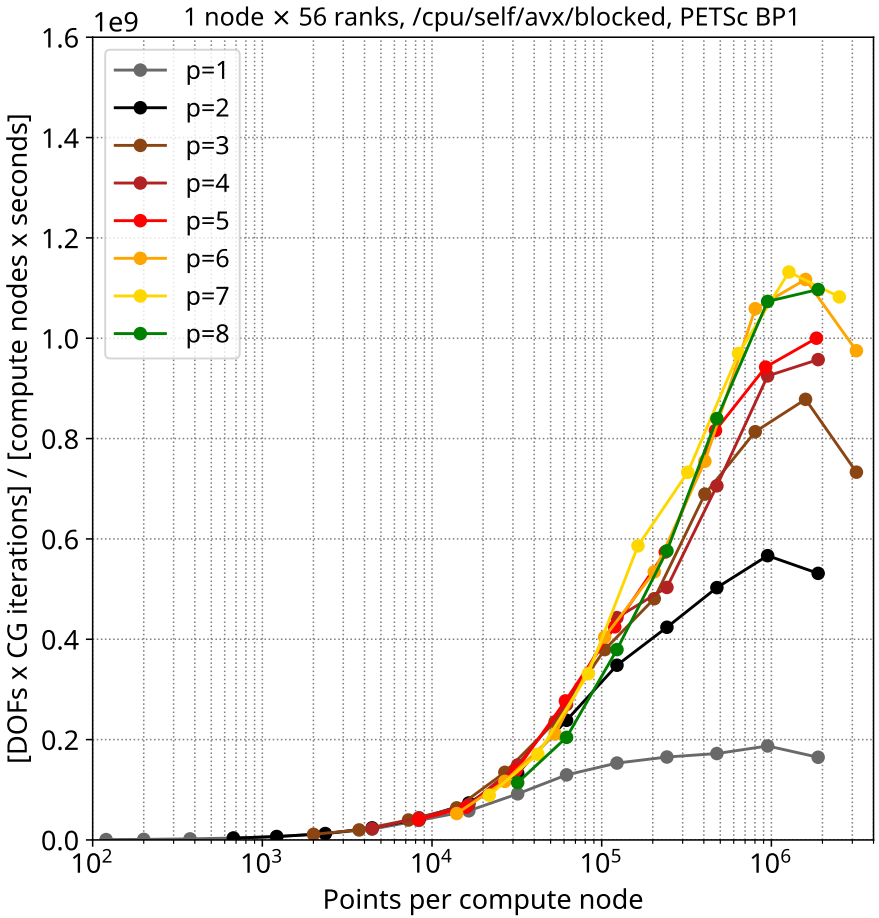


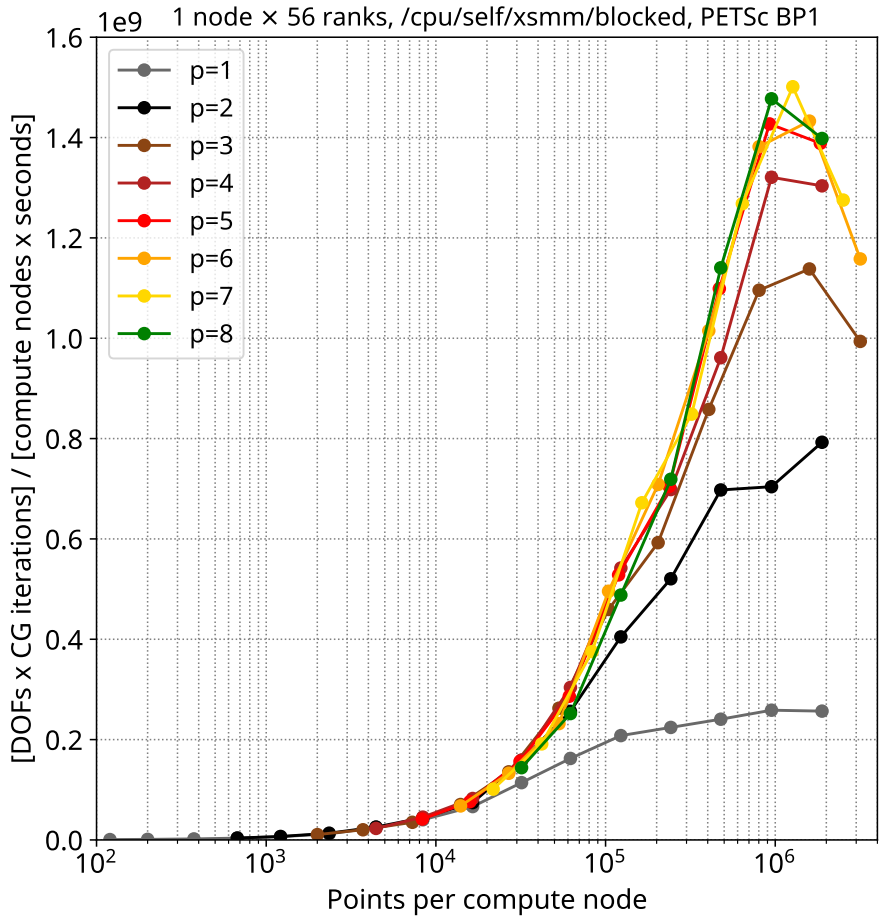
Figure: Knight Landing (Intel Xeon Phi 7230 SKU 1.3 GHz) with intel-18 compiler. In (a) serial implementation; in (b) blocked implementation ($q = P + 1$, $P = p + 1$)

University of Colorado Boulder

Performance w. r. t. space, on Skylake: blocked



(a)

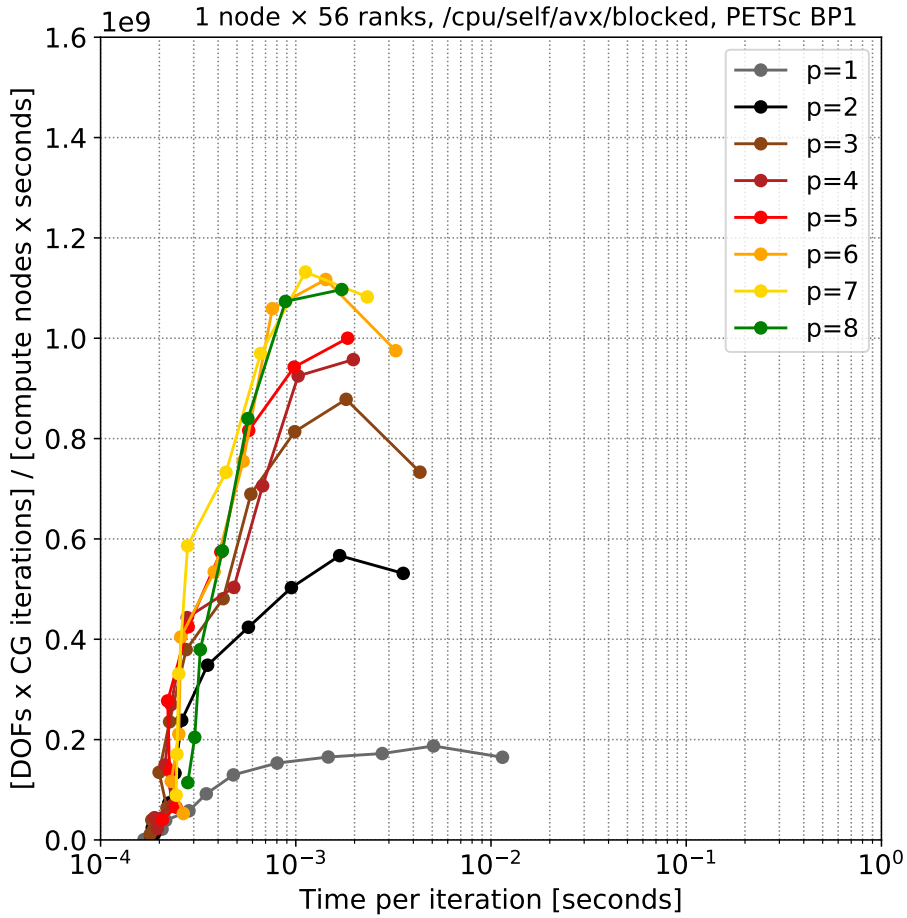


(b)

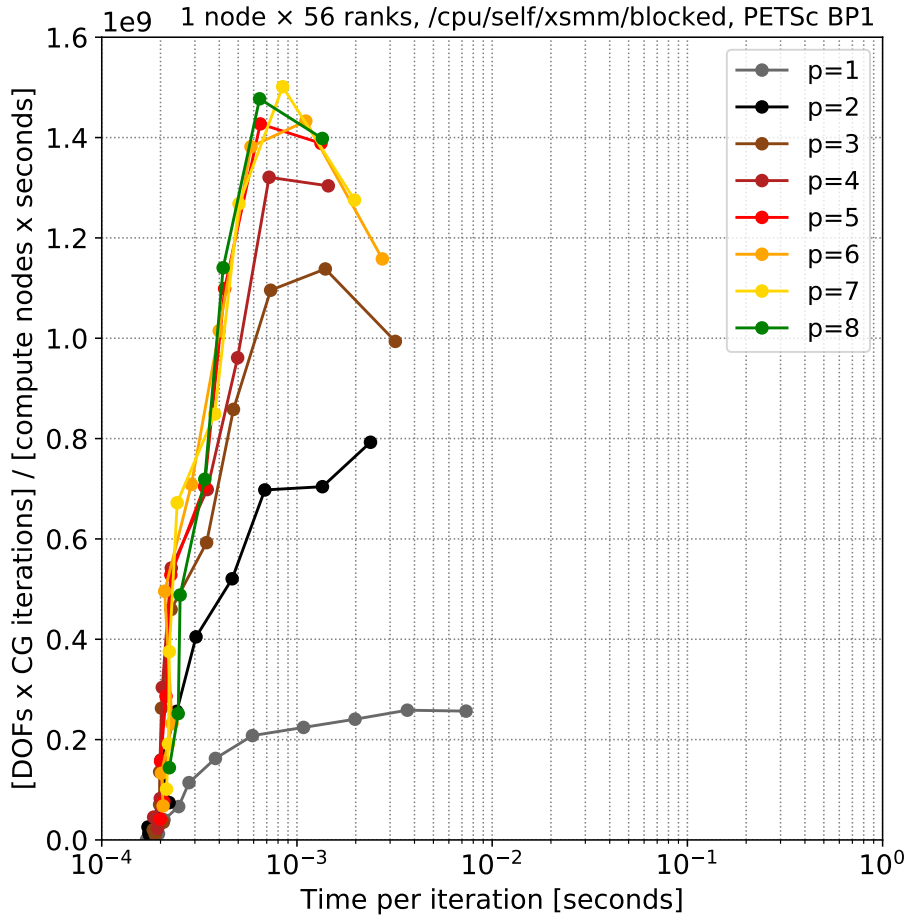
Figure: Skylake (2x Intel Xeon Platinum 8180M CPU 2.50GHz) with intel-19 compiler. Backends: in (a) AVX; in (b) libXSMM ($q = P + 1$, $P = p + 1$)

University of Colorado Boulder

Performance w. r. t. time, on Skylake: blocked



(a)

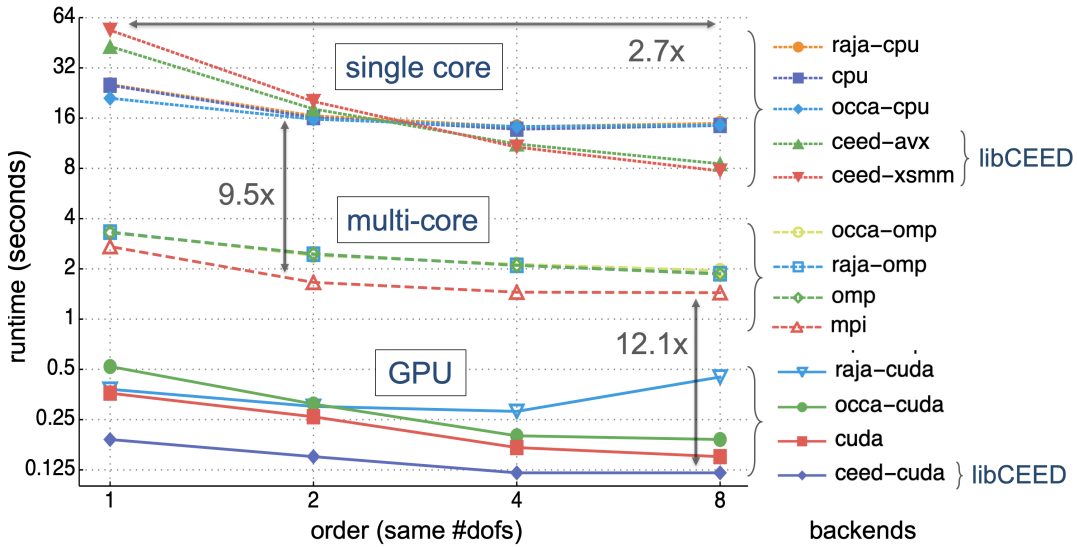


(b)

Figure: Skylake (2x Intel Xeon Platinum 8180M CPU 2.50GHz) with intel-19 compiler.
Backends: in (a) AVX; in (b) libXSMM ($q = P + 1$, $P = p + 1$)

University of Colorado Boulder

Preliminary GPU results: MFEM + libCEED



	1	2	4	8
occa-cuda =	00.52	00.31	00.20	00.19
raja-cuda =	00.38	00.30	00.28	00.45
cuda =	00.36	00.26	00.17	00.15
ceed-cuda =	**00.19**	**00.15**	**00.12**	**00.12**
occa-omp =	03.34	02.41	02.13	01.95
raja-omp =	03.32	02.45	02.10	01.87
omp =	03.30	02.46	02.10	01.86
mpi =	**02.72**	**01.66**	**01.45**	**01.44**
occa-cpu =	**21.05**	**15.77**	14.23	14.53
raja-cpu =	25.42	16.53	14.22	14.88
cpu =	25.18	16.11	13.73	14.45
ceed-avx =	43.04	18.16	11.20	08.53
ceed-xsmm =	53.80	20.13	**10.73**	**07.72**

single-GPU, multi-core CPU, and single-core CPU,
for 1.3 millions DOFs in 2D.
Results by Yohann Dudouit on a Linux desktop with a Quadro GV100 GPU,
sm_70, CUDA 10.1, and Intel Xeon Gold 6130 CPU @ 2.10GHz.

University of Colorado Boulder

Towards a libCEED miniapp: a Navier-Stokes solver

Compressible Navier-Stokes equations in conservation form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{U} = 0\,, \tag{1a}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \left(\frac{\mathbf{U} \otimes \mathbf{U}}{\rho} + P\mathbf{I}_3 \right) + \rho g \mathbf{k} = \nabla \cdot \boldsymbol{\sigma}\,, \tag{1b}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left(\frac{(E + P)\mathbf{U}}{\rho} \right) = \nabla \cdot (\mathbf{u} \cdot \boldsymbol{\sigma} + k \nabla T)\,, \tag{1c}$$

Towards a libCEED miniapp: a Navier-Stokes solver

Compressible Navier-Stokes equations in conservation form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{U} = 0\,, \tag{1a}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \left(\frac{\mathbf{U} \otimes \mathbf{U}}{\rho} + P\mathbf{I}_3 \right) + \rho g \mathbf{k} = \nabla \cdot \boldsymbol{\sigma}\,, \tag{1b}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left(\frac{(E+P)\mathbf{U}}{\rho} \right) = \nabla \cdot (\mathbf{u} \cdot \boldsymbol{\sigma} + k \nabla T)\,, \tag{1c}$$

where $\boldsymbol{\sigma} = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T + \lambda(\nabla \cdot \mathbf{u})\mathbf{I}_3)$, and

University of Colorado Boulder

Towards a libCEED miniapp: a Navier-Stokes solver

Compressible Navier-Stokes equations in conservation form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{U} = 0\,, \tag{1a}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \left(\frac{\mathbf{U} \otimes \mathbf{U}}{\rho} + P\mathbf{I}_3 \right) + \rho g \mathbf{k} = \nabla \cdot \boldsymbol{\sigma}\,, \tag{1b}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left(\frac{(E+P)\mathbf{U}}{\rho} \right) = \nabla \cdot (\mathbf{u} \cdot \boldsymbol{\sigma} + k \nabla T)\,, \tag{1c}$$

where $\boldsymbol{\sigma} = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T + \lambda(\nabla \cdot \mathbf{u})\mathbf{I}_3)$, and

$$(c_p/c_v - 1)\,(E - \mathbf{U} \cdot \mathbf{U}/(2\rho) - \rho g z) = P \quad \leftarrow \text{ pressure}$$

$\mu \leftarrow$ dynamic viscosity

$g \leftarrow$ gravitational acceleration

$k \leftarrow$ thermal conductivity

$\lambda \leftarrow$ Stokes hypothesis constant

$c_p \leftarrow$ specific heat, constant pressure

$c_v \leftarrow$ specific heat, constant volume

University of Colorado Boulder

Vector form

The system (1) can be rewritten in vector form

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = S(\mathbf{q}), \tag{2}$$

for the state variables

$$\mathbf{q} = \begin{pmatrix} \rho \\ \mathbf{U} \equiv \rho \mathbf{u} \\ E \equiv \rho e \end{pmatrix} \begin{matrix} \leftarrow \text{ volume mass density} \\ \leftarrow \text{ momentum density} \\ \leftarrow \text{ energy density} \end{matrix} \tag{3}$$

Vector form

The system (1) can be rewritten in vector form

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = S(\mathbf{q}), \tag{2}$$

for the state variables

$$\mathbf{q} = \begin{pmatrix} \rho \\ \mathbf{U} \equiv \rho \mathbf{u} \\ E \equiv \rho e \end{pmatrix} \begin{array}{l} \leftarrow \text{ volume mass density} \\ \leftarrow \text{ momentum density} \\ \leftarrow \text{ energy density} \end{array} \tag{3}$$

where

$$\mathbf{F}(\mathbf{q}) = \begin{pmatrix} \mathbf{U} \\ (\mathbf{U} \otimes \mathbf{U})/\rho + P\mathbf{I}_3 - \boldsymbol{\sigma} \\ (E + P)\mathbf{U}/\rho - (\mathbf{u} \cdot \boldsymbol{\sigma} + k\nabla T) \end{pmatrix},$$

$$S(\mathbf{q}) = -\begin{pmatrix} 0 \\ \rho g \hat{\mathbf{k}} \\ 0 \end{pmatrix}$$

University of Colorado Boulder

Application example: Density current

A cold air bubble drops by convection in a neutrally stratified atmosphere.

Its initial condition is defined in terms of the Exner pressure, $\pi(\mathbf{x}, t)$, and potential temperature, $\theta(\mathbf{x}, t)$, that relate to the state variables via

$$\rho = \frac{P_0}{(c_p - c_v)\theta(\mathbf{x}, t)} \pi(\mathbf{x}, t)^{\frac{c_v}{c_p - c_v}}, \tag{4a}$$

$$e = c_v \theta(\mathbf{x}, t)\pi(\mathbf{x}, t) + \mathbf{u} \cdot \mathbf{u}/2 + gz, \tag{4b}$$

where P_0 is the atmospheric pressure.

BCs: no-slip and non-penetration for $\mathbf{u}$, no-flux for mass and energy densities.

University of Colorado Boulder

Density current

order: $p = 8$, $P = p + 1$, $q = P + 4$, $\Omega = [0, 6000]^2$ m $\times [0, 3000]$ m,
elem. resolution: 125 m, Global DOFs: 1 081 665

University of Colorado Boulder

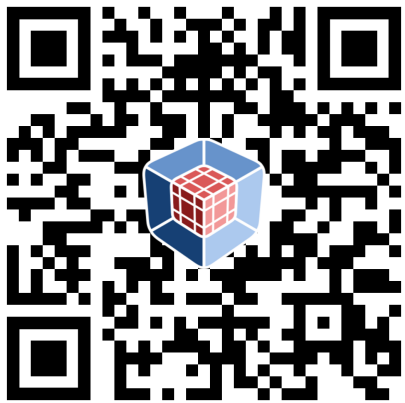
Conclusions and outlook

- We have demonstrated the use of libCEED with PETSc for the numerical high-order solutions of the full compressible Navier-Stokes equations
- libCEED always welcomes contributors and friendly users
  `https://github.com/CEED/libCEED`

University of Colorado Boulder

Conclusions and outlook

- We have demonstrated the use of libCEED with PETSc for the numerical high-order solutions of the full compressible Navier-Stokes equations
- libCEED always welcomes contributors and friendly users
  `https://github.com/CEED/libCEED`



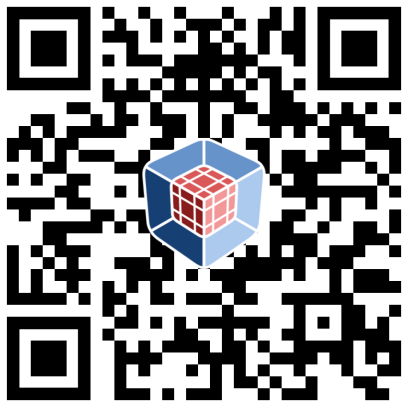
Ongoing and future work:

- Improved non-conforming and mixed-mesh support (integration of PETSc's DMPlex)
- Implicit-explicit (IMEX) and fully implicit time discretizations
- Algorithmic differentiation (AD) of Q-functions
- Ongoing work on CUDA and HIP optimizations

University of Colorado Boulder

Conclusions and outlook

- We have demonstrated the use of libCEED with PETSc for the numerical high-order solutions of the full compressible Navier-Stokes equations
- libCEED always welcomes contributors and friendly users
  `https://github.com/CEED/libCEED`



Ongoing and future work:

- Improved non-conforming and mixed-mesh support (integration of PETSc's DMPlex)
- Implicit-explicit (IMEX) and fully implicit time discretizations
- Algorithmic differentiation (AD) of Q-functions
- Ongoing work on CUDA and HIP optimizations

Acknowledgements: Exascale Computing Project (17-SC-20-SC), NSF

Thank you!

University of Colorado Boulder