SAND2015-6755C

Higher-Order Particle Weighting for Unstructured Meshes on Modern Architectures

Scott High[1,2] Matthew T. Bettencourt[1]

[1]Electromagnetic Theory,
Sandia National Laboratories,
Albuquerque, New Mexico, USA
[2]University of Illinois
Urbana–Champaign, Ilinios , USA

International Conference on Numerical Simulation of Plasmas
Golden Colorado
August 13, 2015

Center for Exascale Simulation
XPACC
of Plasma-Coupled Combustion

Work partially supported under
Department of Energy, National Nuclear Security Administration, under Award Number DE-NA0002374

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

NNSA National Nuclear Security Administration

SAND 2015-3757 C

Sandia National Laboratories

Background

- US DoE Advanced Scientific Computing (ASC) is refreshing their major computing resources
  - FY16 Trinity – Self booted Xeon Phi cluster
  - FY18 Sierra – IBM Power with Nvidia chips -150 PF
  - FY21 ATS-3 – In bidding process
- Current software stack cannot take advantage of these platforms
- New program element to develop new applications
  - Advanced Technology Demonstration and Mitigation
- Sandia National Laboratories is developing a new code for plasma simulation EMPIRE
  - ElectroMagnetic Plasma In Radiation Environments

EMPIRE

- EMPIRE is a new open source application being developed for next generation platforms
- Physics and simulation goals –
  - Electromagnetic and electrostatic
  - Kinetic and fluid based plasma descriptions
  - Radiation transport and gas chemistry - DSMC
  - Beyond forward simulation – Sensitivities, UQ, optimization, …
- Computation capabilities –
  - Hybrid structured/unstructured mesh
  - Hybrid PIC-fluid description
  - In-situ meshing, mesh refinement
  - In-situ analysis, visualization and adjoint methods
  - Asynchronous Multi-Tasking (AMT), dynamic balancing

Mini-PIC

- Goal – Develop a reduced physics application to act as a testbed for development of next generation PIC models
- Continuum models in FEM, and solver technologies are being examined by a large number of people
  - Focus on the PIC part of the solve
  - MPI+X
    - Domain Decomposition for MPI
    - Kokkos for cross platform threading
  - Uses unstructured tet or hex elements
  - ES and EM field solves
    - High order particle and current weighting
  - DSMC under development
  - Being used as a test platform for embedded sensitivity propagation research

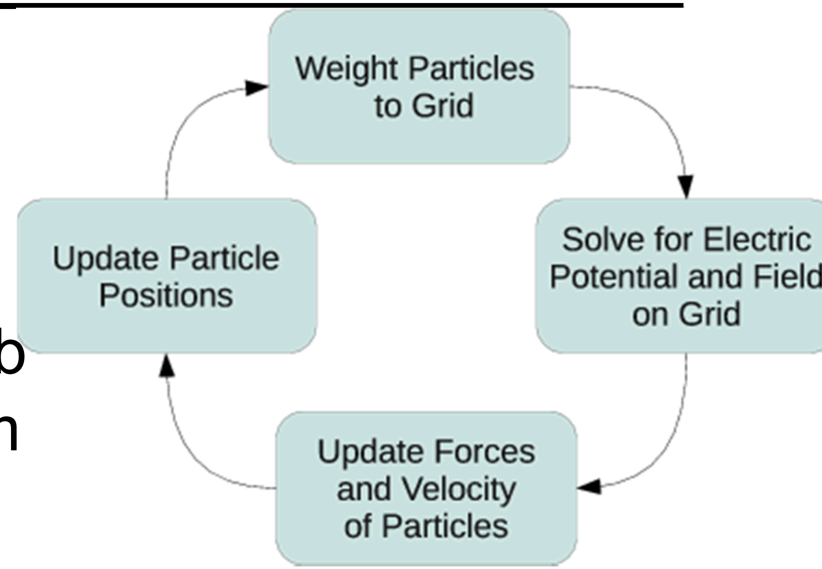
Kokkos: C++ Library / Programming Model for Manycore Performance Portability

- Portable to Advanced Manycore Architectures
  - Multicore CPU, NVidia GPU, Intel Xeon Phi
    - Backends – Cuda, OpenMP, pthreads and serial
  - Maximize amount of user (application/library) code that can be compiled without modification and run on these architectures
  - Minimize amount of architecture-specific knowledge that a user is required to have
  - Allow architecture-specific tuning to easily co-exist
  - Requires a C++11 compiler
- Performant
  - Portable user code performs as well as architecture-specific code
  - Thread scalable – not just thread safety (no locking!)
- Usable
  - Small, straight-forward application programmer interface (API)
  - Constraint: don't compromise portability and performance

Kokkos: Collection of Libraries

- Core – lowest level portability layer
  - Portable data-parallel dispatch: parallel_for, parallel_reduce, parallel_scan
  - Multidimensional arrays with device-polymorphic layout for transparent and device-optimal memory access patterns
  - Access to hardware atomic operations
- Containers – built on core arrays
  - UnorderedMap – fast find and thread scalable insertion
  - Vector – subset of std::vector functionality to ease porting
  - Compress Row Storage (CRS) graph
- Linear Algebra
  - Sparse matrices and linear algebra operations
  - Wrappers to vendors' libraries
  - Portability layer for Trilinos manycore solvers
  - Trilinos/Tpetra built on top of Kokkos for MPI+X linear algebra

Time Update



- Mini-PIC uses the standard operator split, leap-frog, time update scheme
- Elliptic solve uses a Bi-CGStab which extracts parallelism from Tpetra
- Particle position update requires element by element tracking
  - Needed for EM current deposition and thin feature detection
  - Iterative for hex elements, analytic for tet's
- Weighting algorithms can cause conflicts

Electromagnetic Formulation

- Starting with Maxwell's equations

$$\frac{\partial B}{\partial t}+\nabla\times E=0;\frac{1}{c^2}\frac{\partial E}{\partial t}-\nabla\times B=-\mu J;\nabla\cdot B=0;\nabla\cdot E=\frac{\rho}{\varepsilon}$$

- One can put this into a second order formulation

$$\frac{1}{c^2}\frac{\partial E}{\partial t}+\nabla\times\nabla\times E=-\mu\frac{\partial J}{\partial t}$$

- Which can be updated with unconditional stability $\alpha=0.25$

$$\frac{1}{c^2}\frac{E^{n+1}-2E^n+E^{n-1}}{\Delta t^2}+\nabla\times\nabla\times\left[\alpha E^{n+1}+(1-2\alpha)E^n+\alpha E^{n-1}\right]=-\mu\frac{J^{n+1/2}-J^{n-1/2}}{\Delta t}$$

- Which can be put into the weak form and integrated by parts

$$\frac{1}{c^2\Delta t^2}\int\left(E^{n+1}-2E^n+E^{n-1}\right)v\,dV+\int(\nabla\times v)\cdot\left(\nabla\times\left[\alpha E^{n+1}+(1-2\alpha)E^n+\alpha E^{n-1}\right]\right)dV=-\frac{\mu}{\Delta t}\int\left(J^{n+1/2}-J^{n-1/2}\right)dV$$

Electromagnetic Formulation Cont.

- Expanding the electric field in the basis function v

$$E = \sum e_i v_i = \hat{e} \cdot \hat{v}$$

- Defining mass and stiffness matrices

$$M_{(i,j)} = \int v_i \cdot v_j dV ; S_{(i,j)} = \int \nabla \times v_i \cdot \nabla \times v_j dV$$

- Yields the following matrix equation

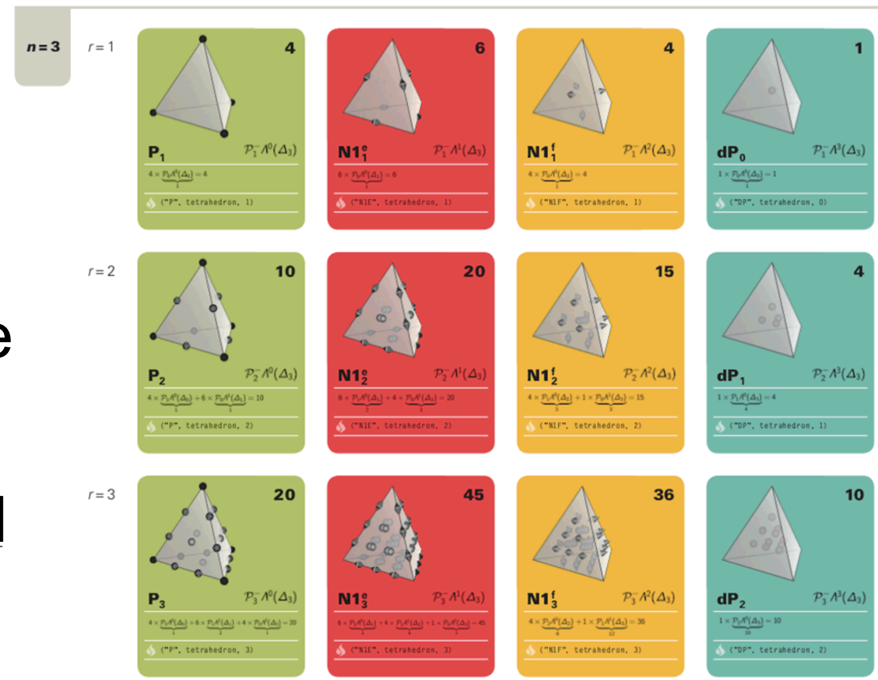
$$\left(M + c^2 \Delta t^2 \alpha S\right) \hat{e}^{n+1} = -c^2 \Delta t^2 S\left((1-2\alpha)\hat{e}^n + \alpha \hat{e}^{n-1}\right) + M\left(2\hat{e}^n - \hat{e}^{n-1}\right) - \int \Delta t \mu \left(J^{n+1/2} - J^{n-1/2}\right) v dV$$

- Current is represented as a delta function moving though space $J(x,t) = qu(t)\delta(x,t)$
  - Thus the integral is just an evaluation of the basis, lumped mass formulation $j_i^{n+1/2} = qu^{n+1/2} v_i (x^{n+1/2})$
- Electrostatic formulation uses standard mass matrix and lumped mass charge weighting

Element Space

- Currently using Nedelec elements
  - Form an incomplete representation of the space on simplex elements
  - Naturally satisfies the divergence constraints

- Convergence for EM only first order for N1 elements
  - 2^{nd} order convergence for potential in ES solve
  - E field still 1^{st} order
- Looking at the second kind of Nedelec elements
  - 2x DOFs



Particle Movement

- Particles are moved in “reference space”
  - Lower round-off, simpler conditionals
- On tet (simplex) mesh elements this requires a multiplication by the element Jacobian matrix
- On hex mesh elements, elements are non-affine and therefore one needs to iterate
  - Jacobian matrix is a function of position within the element
  - Even if velocity is constant within an element, reference velocity may not be, straight physical lines are curved in reference space
    - Midpoint integration not sufficient in hex elements
  - Iterate reference velocity until starting and ending points in physical and reference space coincide
    - Reference velocity represents transformed physical velocity at a point along the trajectory.
    - Reference velocity can be used to weight current

Particle Migration

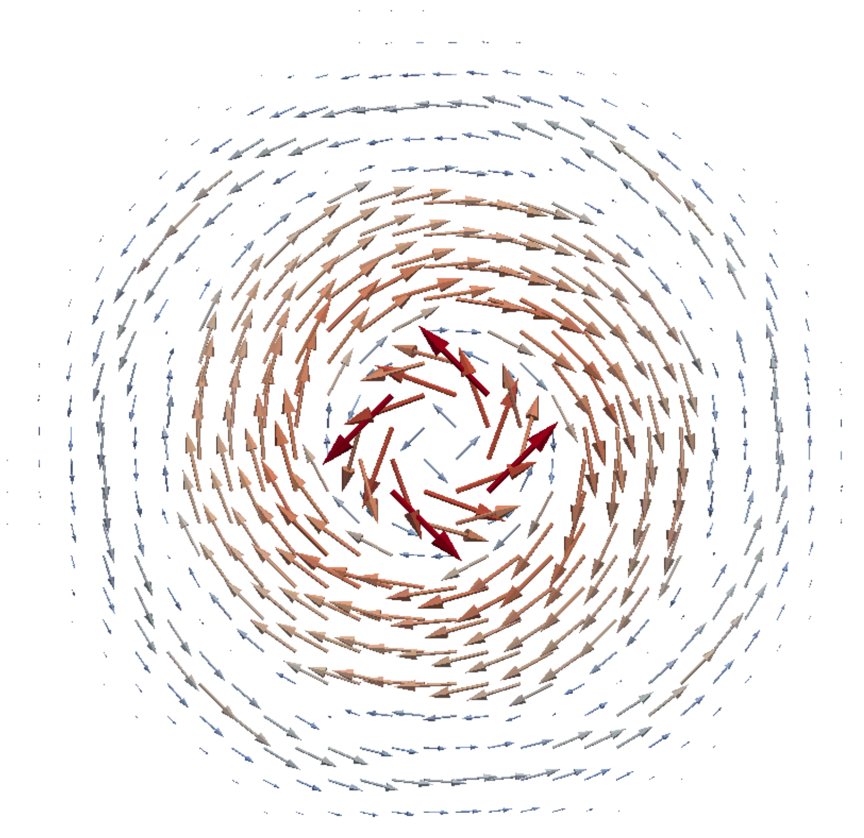
- At a high level particles are pushed to the end of their timestep or until they hit a processor boundary
  - Particles are packed into migrate buffers and migrated to neighboring processors
  - Particles are unpacked on destination processor and the remaining of the push is conducted
  - This is repeated until all the particles reach their final location
  - MPI calls are done in the host space, copied to/from device space

Charge Weighting

- Weighting charge in a threaded environment can result in data write conflicts
  - Atomic operations are typically used in summations
    - Lots of conflicts and poor performance on Cuda
  - Alternative – sort the particles by element and sum into local variables
    - Fewer atomic operations
    - Sort can be expensive
    - Not extendable to EM solve and currents due to element crossing
    - Only practical on Nvidia, sorting too slow on other platforms

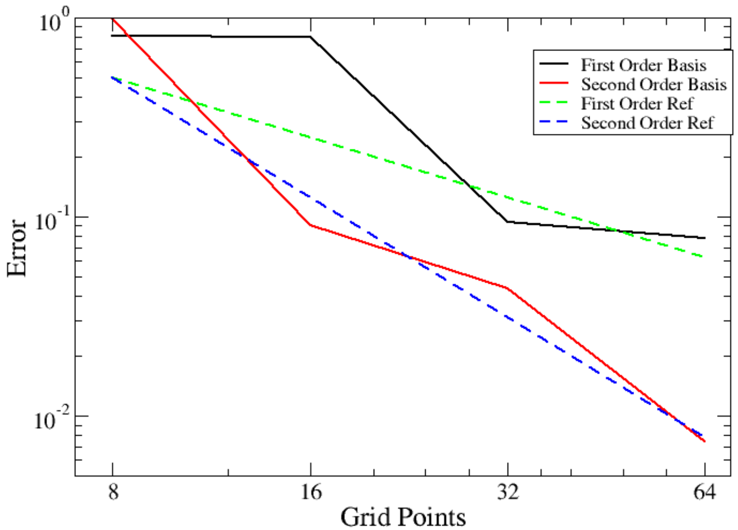
Results

- Circular current
  - Gaussian current pulse
  - 3 dimensional box with Dirichlet conditions
  - 1.3M elements

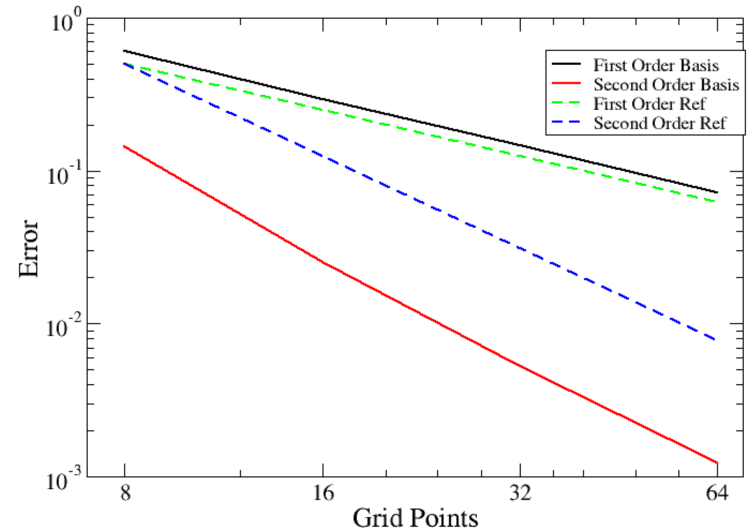


Convergence Results

Electrostatic – Electric field between two particles

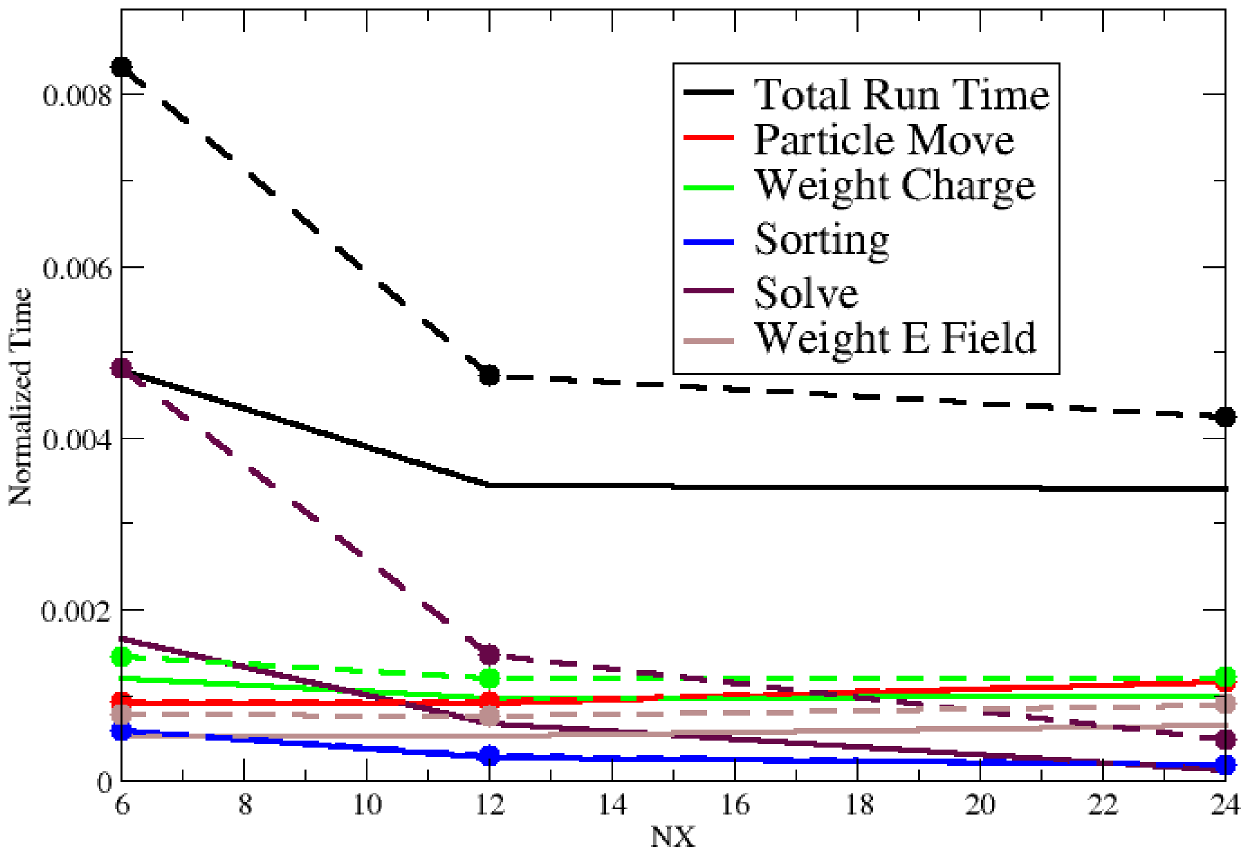


Electromagnetic – Method of Manufactured Solutions



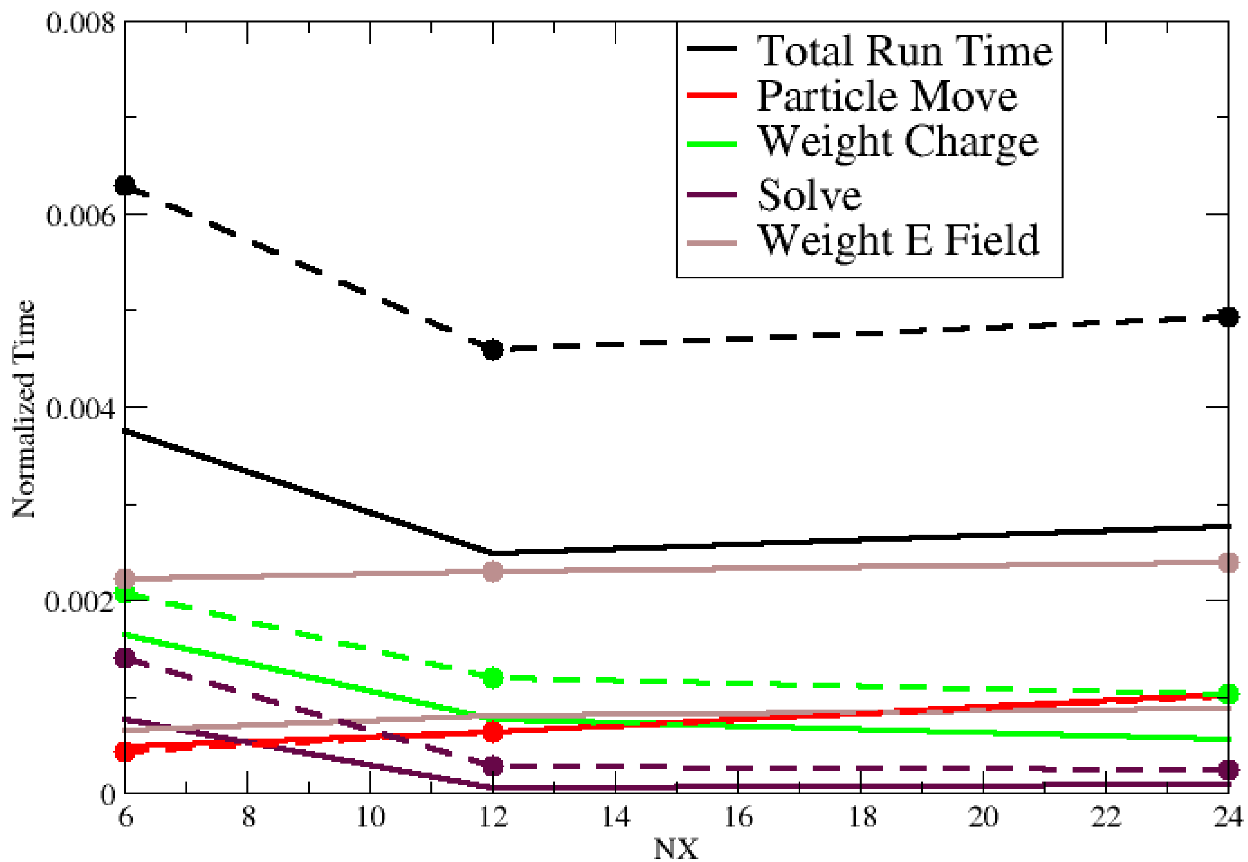
Runtime Comparison for Electrostatic Nvidia

– Runtime for different parts of PIC algorithm
  – Scaled by number of cells, 100 particles/cell
  – Circle points use higher order basis



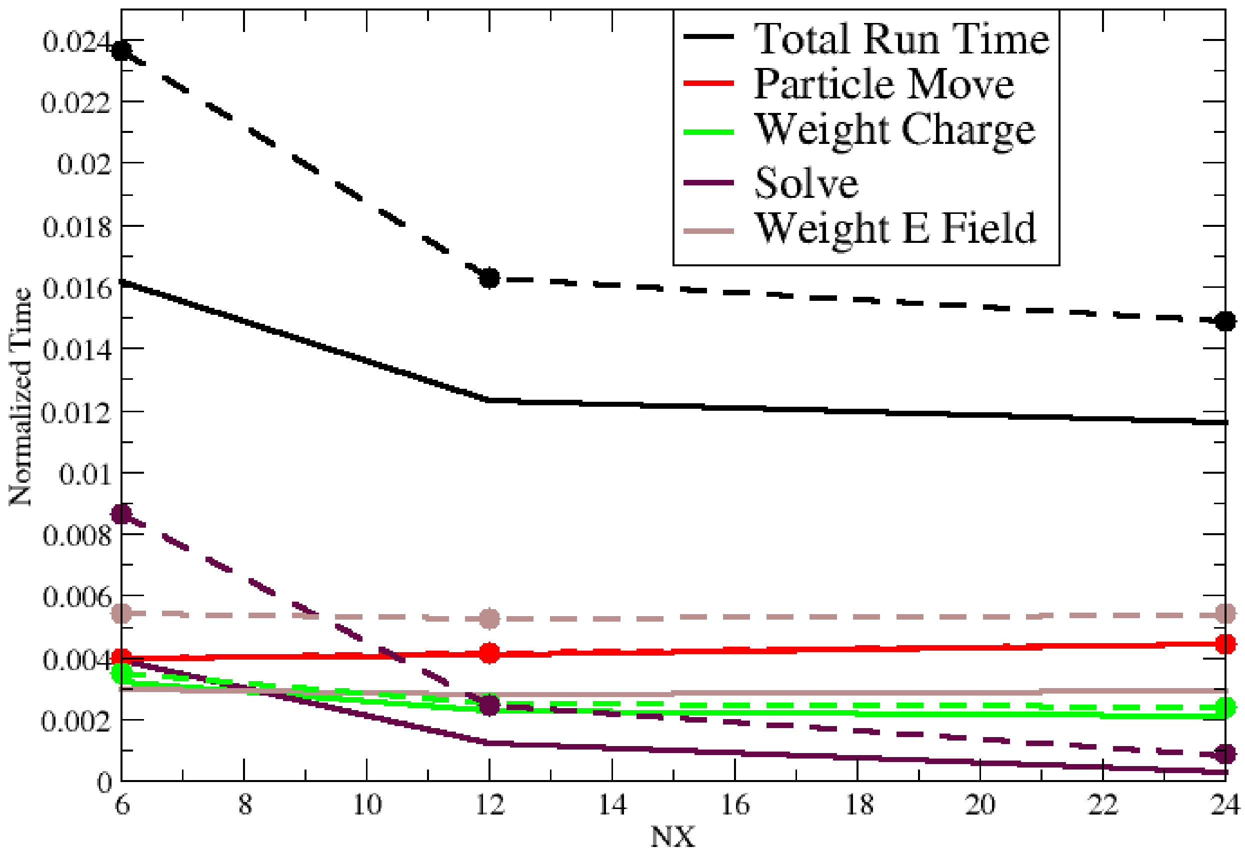
Runtime Comparison for Electrostatic Intel SandyBridge

- Runtime for different parts of PIC algorithm
  - Scaled by number of cells, 100 particles/cell
  - Circle points use higher order basis



Runtime Comparison for Electrostatic Intel Knights Corner (Xeon Phi)

- Runtime for different parts of PIC algorithm
  - Scaled by number of cells, 100 particles/cell
  - Circle points use higher order basis



Summary

- Linear and second order FEM discretizations developed for both EM and ES PIC
  - Uses compatible discretizations with Nedelec elements
- Fields converge at expected order
- Run-times roughly two times slower for higher order on GPUs and other cached systems
  - More efficient at higher spatial resolution
- On tetrahedral mesh electric field converges at the order of the elements
  - In electrostatics – E is derivative of solution
  - In electromagnetics – Basis is incomplete
- Looking at using the second kind of Nedelec elements