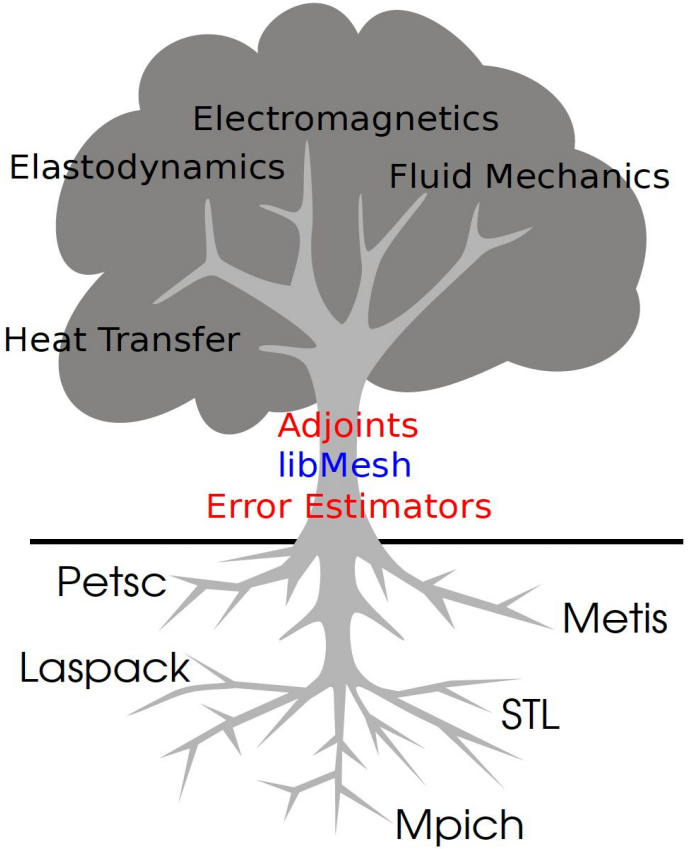
State-of-the-art adjoint capability in libMesh

Vikram V Garg

Oct 1, 2021

libMesh: An Open Source C++ Finite Element library

- Provides core FEM capabilities
- Advanced capabilities
  - AMR, adjoint-based algorithms, reduced basis methods
- Application codes
  - Fluids, materials, heat transfer and optimization problems
- At least two commercial solvers based on libMesh (Akselos, CoreForm).



Differentiating Physics Codes

Simulation models need to provide the pathway from changes in the inputs to changes in Quantities of Interest.

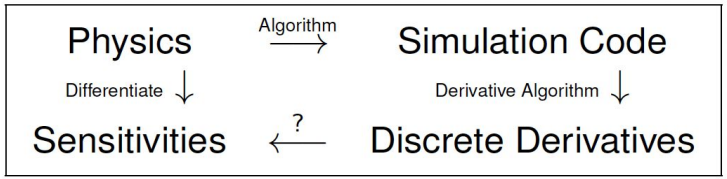


Figure: Obtaining derivatives from simulation codes.

Adjoint Sensitivity

$$\text{Find } u \in U, \text{ such that } \mathcal{R}(u(p), v) = 0 \ \forall v \in V$$

$$\text{Solve } \frac{\partial \mathcal{R}}{\partial u}(v, z(p_0)) = Q_u(v; u(p_0)) \ \forall v \in V$$

$$Q(u(p_1)) - Q(u(p_0)) \approx \frac{\partial \mathcal{R}}{\partial p}(u(p_0), z(p_0))\Delta p$$

Algorithms enabled by adjoint capability

- Sensitivity gradients (optimization, design, analysis)
- Mesh/model refinement (automatic refinement to achieve accuracy in specific QoIs)
- Flux postprocessing (adjoints provide automatic, localizable lift functions for higher order flux evaluation)
- Interpretation of stabilized formulations
- Many more to come ...

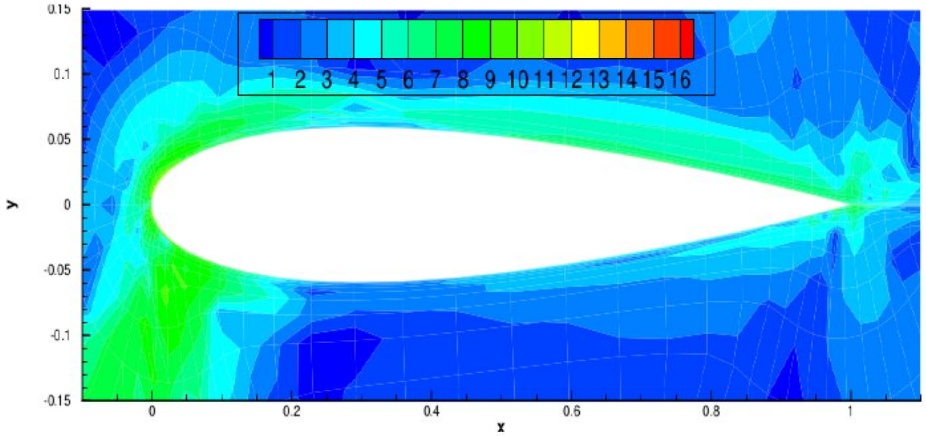
My roles

- Suitable mathematical formulations for adjoint problems
  - Consistency, stability
- Algorithm development for adjoint calculation
  - Preconditioner reuse, adjoint error indicators, adjoint dirichlet boundary conditions
- Object-oriented software design for seamless and intuitive integration of adjoint capability
  - APIs, design patterns, internal memory/IO management, verification and example suite

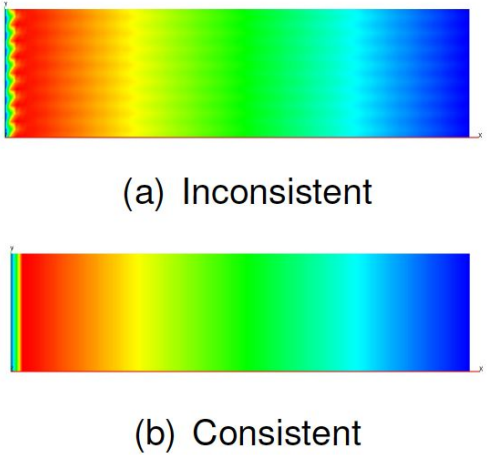
Mathematical Contributions

Direct adjoint calculation not consistent for complex flow problems.

- Avoided instability in adjoint calculation for boundary fluxes.
- Achieved consistency using modified formulations for species transport.



Velocity adjoint for lift QoI at 16 degree AoA



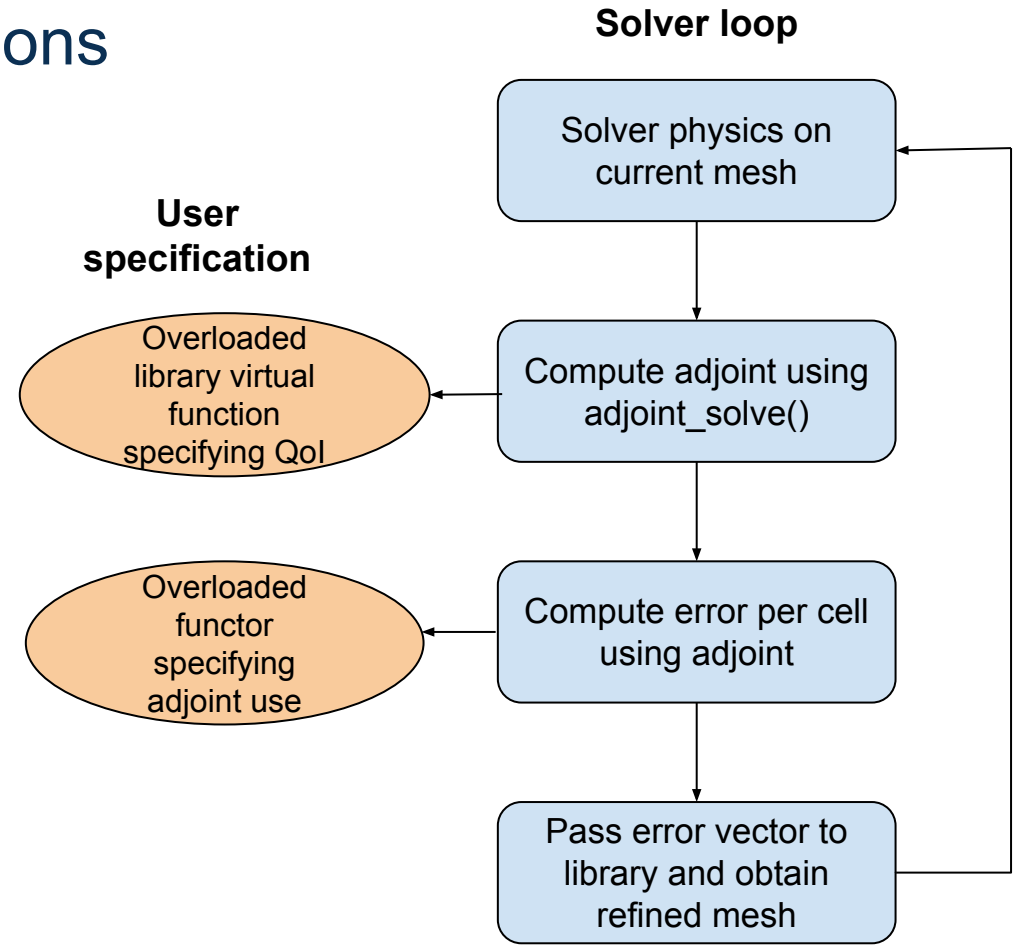
(a) Inconsistent

(b) Consistent

Software Design Contributions

Automatic adjoint capability

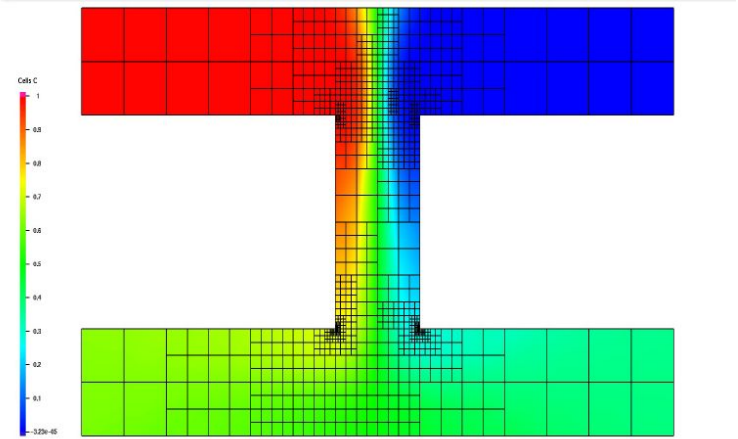
- User only specifies QoI and physics weighting via overloading
- Adjoint computation, storage and postprocessing handled by library
- Model error instead of discretization using physics pointer swapping

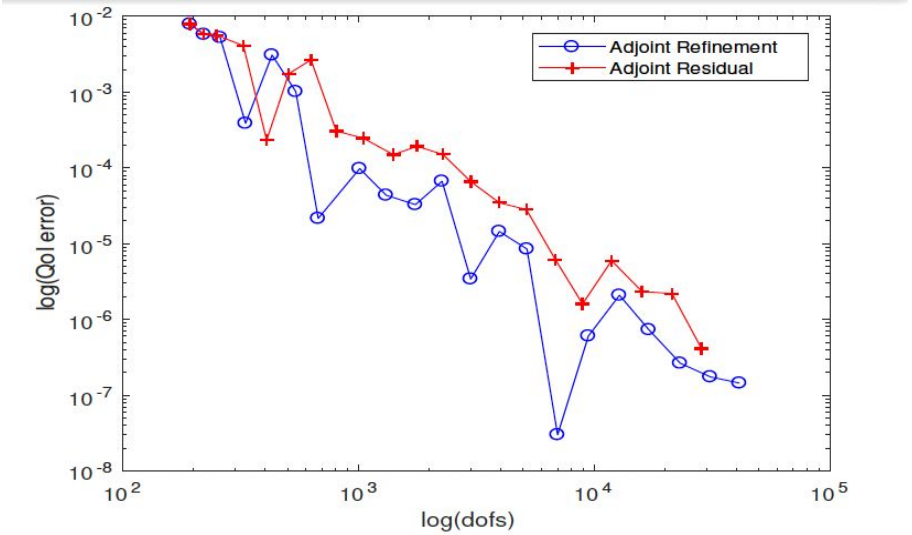


Algorithmic Contributions

Efficient Adjoint Calculation and Use for AMR

- ‘Equal order’ error estimation algorithm improves efficiency by sacrificing accuracy.
- Avoided recalculation of adjoint on finer space for error indicators.
- Reused primal linear solver preconditioner for adjoint problem.





Conclusions

- New generation of simulation technologies, design framework depend on derivative information.
- Mathematical analysis leads to adjoints as the natural counterpart to derivatives for PDEs.
- Efficient software design incorporates adjoint capability efficiently and with minimal intrusion.
- Recent work on unsteady adjoint support in the context of adaptive time solvers and AMR.