Adjoint Sensitivities and Calibration of Hypersonic Flow Problems in SPARC

Eric Phipps, Patrick Blonigan, Kathryn Maupin, and Jaideep Ray

Trilinos User's Group 2022

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY
NNSA National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2022-14373 C

SPARC: Sandia Parallel Aerodynamics and Reentry Code

(T. Fischer, et al)

Goal: Create a credible full-system virtual flight-testing platform for hypersonic vehicles

Modeling

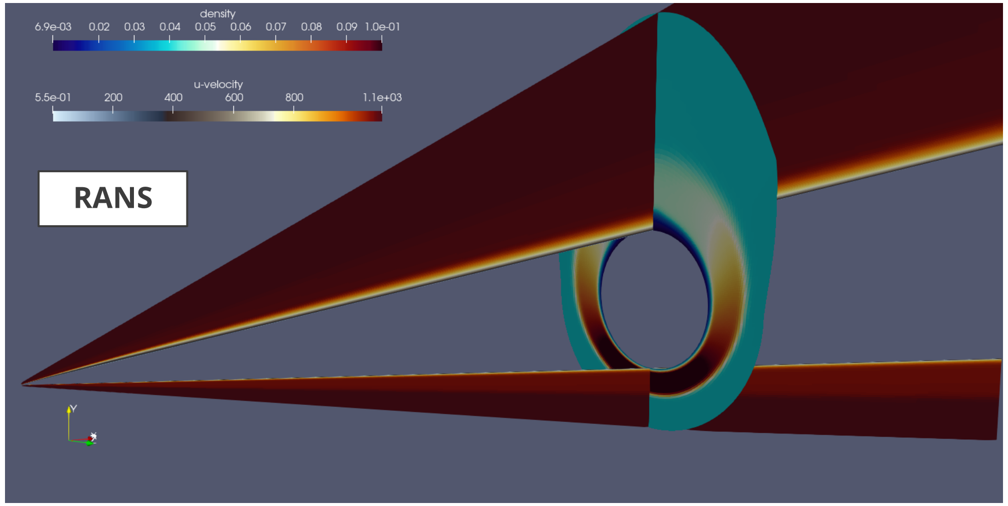
- Perfect and non-equilibrium thermal and chemical gas models
- Euler, Laminar, RANS, Hybrid RANS/LES, LES, and DNS
- Structured and Unstructured Finite Volume methods
- R&D in structured and unstructured high-order methods
- Simulate coupled ablation
- Couples to SIERRA for full-system thermal and structural analyses

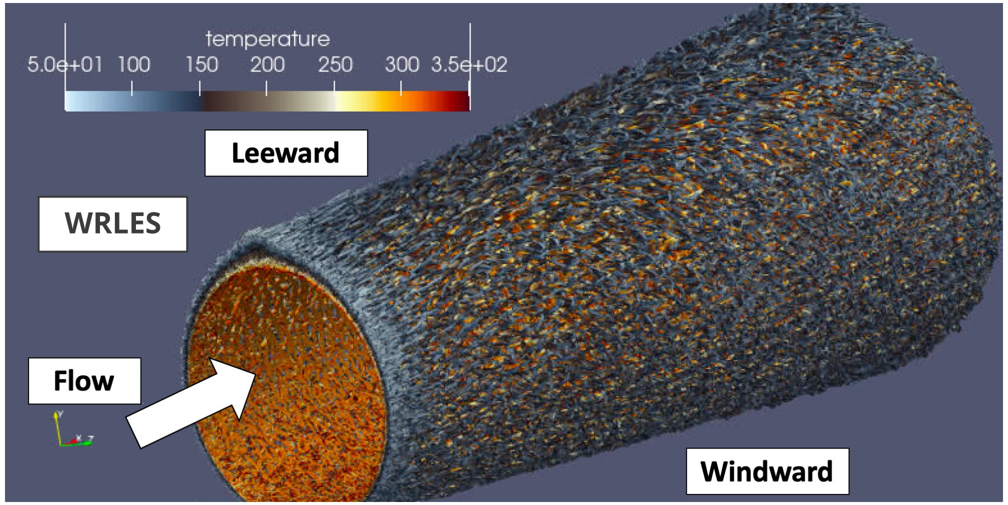
Performance and Portability

- Performance Portability through Kokkos
- Good performance on x86, Arm, and GPU platforms
- Uses performance portable/scalable linear solvers from Trilinos
- Uses embedded geometry and inline mesh refinement

Credibility

- Validation with UQ against wind tunnel and flight test data
- Visibility and peer review by external hypersonics community

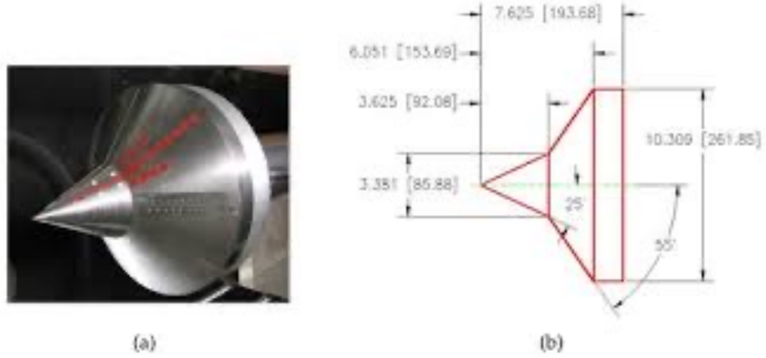


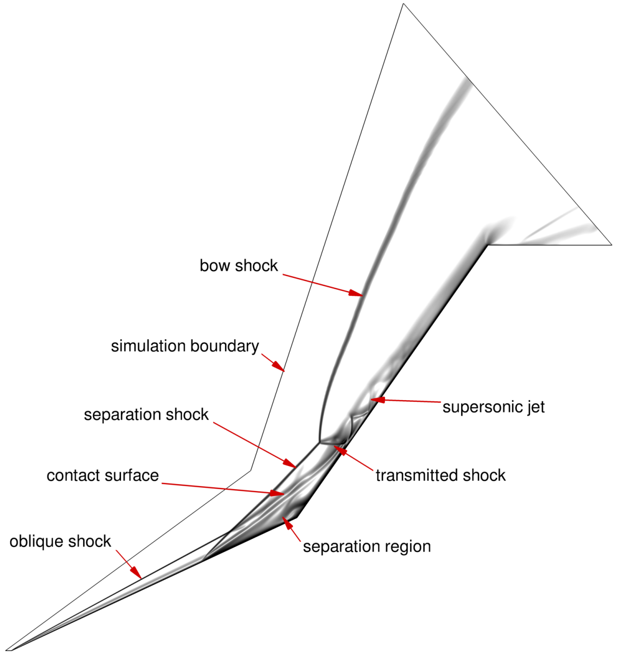


Background

- SPARC validated by comparison to wind tunnel experiments at CUBRC LENS-I/XX facilities for hypersonic flow over a double-cone geometry
  - Uncertainties in inflow (boundary) conditions required calibrating them against subset of data
  - Problems are steady-state, but complex shock interactions requires use of time integration methods to find solutions (pseudo-transient)
  - Due to simulation expense, calibration was conducted using surrogate models trained from samples of SPARC simulations
- ATDM/DPC project investigated tools for “embedded analysis”
  - Embedded sensitivity analysis via Sacado
  - Embedded derivative-based optimization/calibration using ROL
- FY22 ATDM Milestone:
  - Investigate embedded workflows for formulating and solving these calibration problems
  - Investigate the feasibility of solving these problems based on adjoint sensitivities to provide a foundation for distinguishing future capabilities
  - Leverage Sacado automatic differentiation (AD), Tempus time integration, and ROL optimization components

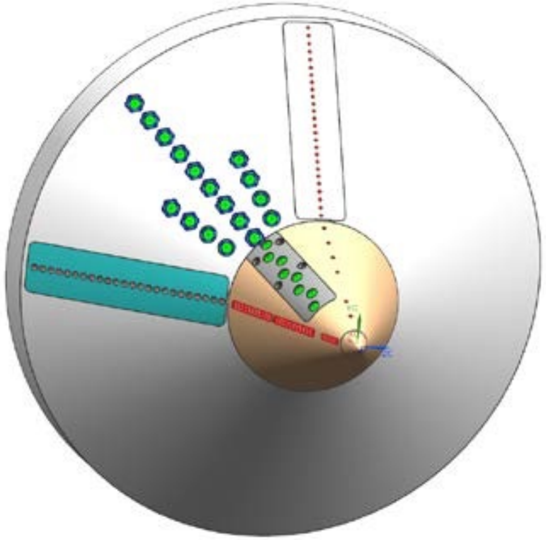
Double-cone geometry

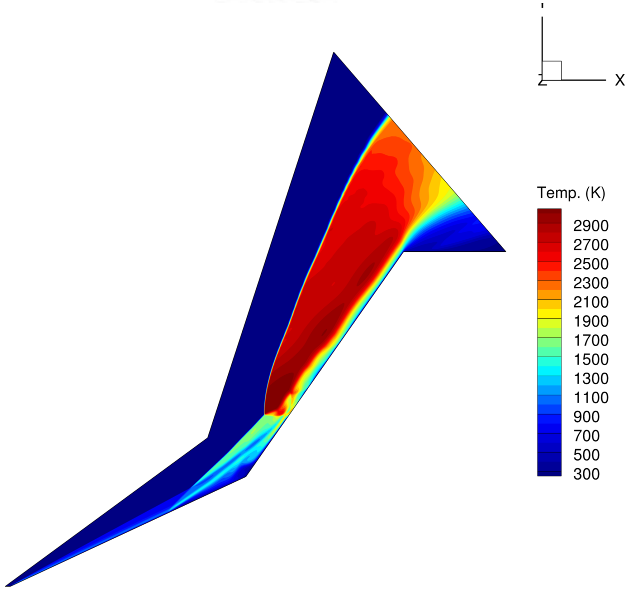




Double Cone calibrations

- Several experimental data sets used for prior validation of SPARC
  - Run35 (easy) – CUBRC LENS-I shock tunnel, perfect gas, vibration & reaction equilibrium, Mach 11
  - Case 1 (moderate) – CUBRC LENS-XX expansion tunnel, real gas, vibrational non-equilibrium, reaction equilibrium, Mach 12
  - Case 4 (hard)– CUBRC LENS-XX expansion tunnel, real gas, vibration & reaction non-equilibrium, Mach 12
- Physics: laminar flow, some dissociation, separation & reattachment, shock interactions
- Measurements: pressure and heat-flux on surface of the cone at several locations
  - Due to model form error, only use probe locations ahead of the separation region
- Prior inference with Bayesian methods and surrogate models complete and published (DOI:10.2514/1.J059033)





J. Ray et al, "Estimation of Inflow Uncertainties in Laminar Hypersonic Double-Cone Experiments", *AIAA Journal* 2020 58:10, 4461-4474

Embedded Optimization with ROL

- ROL Rapid Optimization Library (D. Ridzal et al)
  - Derivative-based optimization library in Trilinos focusing on PDE-constrained optimization

$$\min_{y} \; g(u,y) = 0$$
$$\text{s.t.} \; f(u,y) = 0$$

- Here we use the trust region-CG optimization method using BFGS approximation to the Hessian and reduced-space formulation:

$$\min_{y} h(y), \quad h(y) = g(u(y),y) \;\; \text{s.t.} \;\; f(u(y),y) = 0$$

$$\frac{\partial h}{\partial y} = \frac{\partial g}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial g}{\partial y} \;\; \text{s.t.} \;\; \frac{\partial f}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial f}{\partial y} = 0$$

- Double cone calibration objective function:

$$g(u,y) = \sum_{i=1}^{N_q} \left[s_q(q(x_i;u,y) - \tilde{q}_i)\right]^2 + \sum_{i=1}^{N_p} \left[s_p(p(x_i;u,y) - \tilde{p}_i)\right]^2 + \left[s_h(h_0(y) - \tilde{h}_0)\right]^2 + \left[s_P(P_{Pitot}(y) - \tilde{P}_{Pitot})\right]^2$$

- Calibration parameters: inflow boundary conditions for density, velocity
  - Determined objective function is insensitive to temperature through local sensitivity analysis

Pseudo-transient Nonlinear Solvers

- SPARC employs *pseudo-transient* nonlinear solvers to compute steady-state flows:

$$M\dot{u} + f(u, y) = 0 \implies \frac{1}{\Delta t_k}(Mu_{k+1} - Mu_k) + f(u_{k+1}, y) = 0 \quad \text{(BDF1)}$$

- Called pseudo-transient because time-step residual is not driven to zero each step
  - Typically small number of Newton iterations per time step, using an approximate Jacobian:

$$\tilde{f}_k(u_{k+1}, y) \equiv \frac{1}{\Delta t_k}(Mu_{k+1} - Mu_k) + f(u_{k+1}, y),$$

$$\left(\frac{1}{\Delta t_k} M + J(u^l_{k+1}, y)\right) \Delta u^l_{k+1} = -\tilde{f}_k(u^l_{k+1}, y), \quad u^{l+1}_{k+1} = u^l_{k+1} + \Delta u^l_{k+1},$$

$$J(u^l_{k+1}, y) \approx \frac{\partial f}{\partial u}(u^l_{k+1}, y)$$

  - Each linear system approximately solved using a light-weight linear solver (typically block tri-diagonal solver provided by Ifpack2)

- Manually specified sequence of increasing time-step sizes (called a run-schedule)
  - Requires O(10,000-100,000) time steps

Pseudo-transient Sensitivities

- Similar pseudo-transient approach possible for forward sensitivity equations

$$M\dot{Z} + \frac{\partial f}{\partial u}(u,y)Z + \frac{\partial f}{\partial y} = 0, \quad Z = \frac{\partial u}{\partial y} \implies \frac{1}{\Delta t_k}(MZ_{k+1} - MZ_k) + \frac{\partial f}{\partial u}(u_{k+1},y)Z_{k+1} + \frac{\partial f}{\partial y}(u_{k+1},y) = 0$$

- Results in similar solution approach:

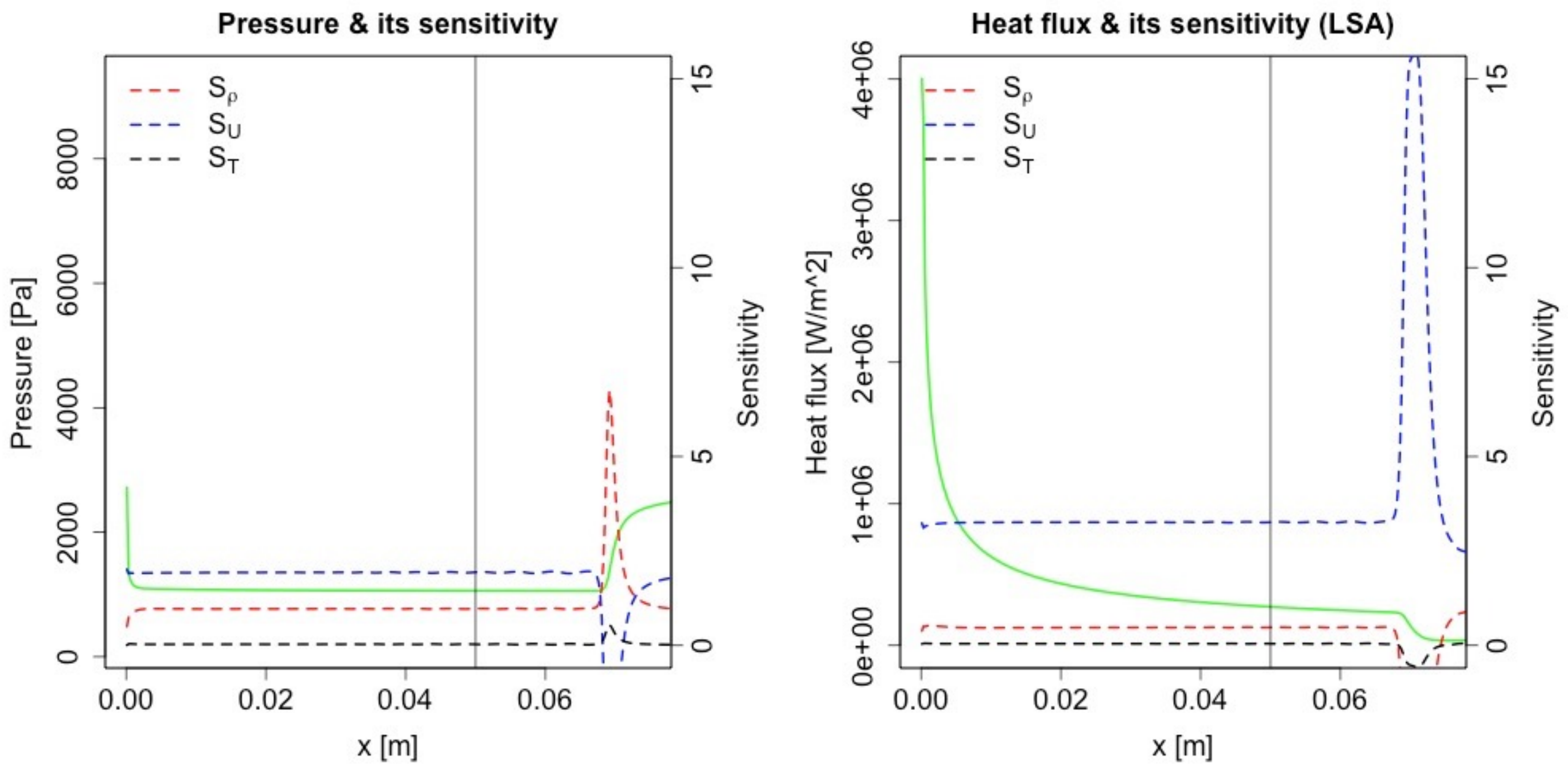
$$F_k(Z_{k+1},u_{k+1},y) \equiv \frac{1}{\Delta t_k}(MZ_{k+1} - MZ_k) + \frac{\partial f}{\partial u}(u_{k+1},y)Z_{k+1} + \frac{\partial f}{\partial y}(u_{k+1},y),$$

$$\left(\frac{1}{\Delta t_k}M + J(u^l_{k+1},y)\right)\Delta Z^l_{k+1} = -F_k(Z^l_{k+1},u^l_{k+1},y), \quad Z^{l+1}_{k+1} = Z^l_{k+1} + \Delta Z^l_{k+1},$$

  - Combined approach: simultaneously solve state and sensitivity equations each time-step
  - Allows use of same run-schedule for state and sensitivity equations
- Sensitivity approach provided by Tempus time integration package (C. Ober et al)
  - Requires users to implement needed partial derivatives
- Sensitivity residual computed using forward-mode Automatic Differentiation (AD) with Sacado package
  - Use ”tangent mode” to compute $\left(\frac{\partial f}{\partial u}\right)Z + \frac{\partial f}{\partial y}$ without explicitly forming $\frac{\partial f}{\partial u}$

Double Cone Sensitivity Analysis

- Sensitivity of objective function with respect to inflow density, velocity, and temperature for Case 1
  - Shows objective function is not sensitive to temperature and can be removed from calibration set



Adjoint Sensitivities

- Recall ROL requires computing the reduced-gradient $\frac{\partial h}{\partial y} = \frac{\partial g}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial g}{\partial y}$ s.t. $\frac{\partial f}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial f}{\partial y} = 0$

- Adjoint sensitivity approach: $\frac{\partial h}{\partial y} = \frac{\partial g}{\partial u}\left(\frac{\partial f}{\partial u}\right)^{-1}\frac{\partial f}{\partial y} + \frac{\partial g}{\partial y} \implies \left(\frac{\partial h}{\partial y}\right)^T = \left(\frac{\partial f}{\partial y}\right)^T\left(\frac{\partial f}{\partial u}\right)^{-T}\left(\frac{\partial g}{\partial u}\right)^T + \left(\frac{\partial g}{\partial y}\right)^T$

- Requires solving linear system of the form: $\left(\frac{\partial f}{\partial u}(u_\infty, y)\right)^T w = \left(\frac{\partial g}{\partial u}(u_\infty, y)\right)^T$

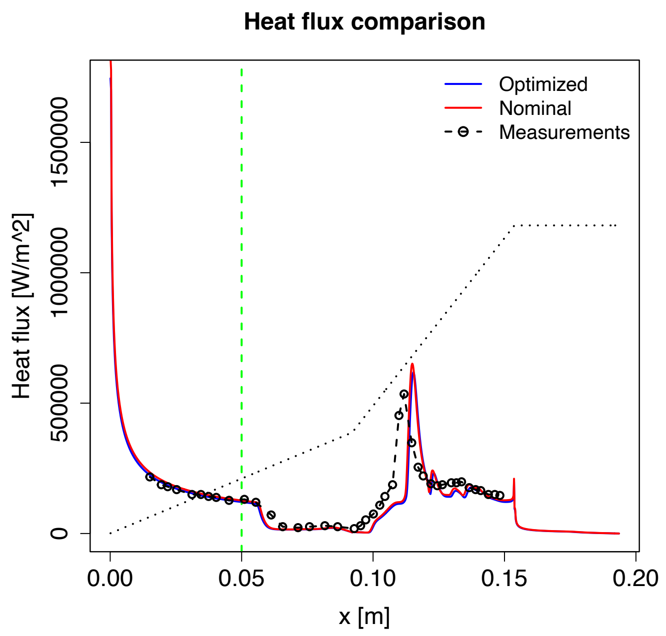
- While a pseudo-transient approach similar to forward sensitivities is possible, found it was not effective on these problems
  - Full transient adjoint and pseudo-transient adjoint capabilities provided by Tempus

- Found a Newton-GMRES approach to be the most effective
  - Apply Newton's method to linear system and solve linear system at each step using GMRES (provided by Belos)
  - Precondition GMRES using SPARC's block tri-diagonal solver applied to the native (approximate) Jacobian-transpose
  - Because of ill-conditioning, multiple Newton iterations (O(10)) are required to achieve small linear residuals with a bounded number of linear iterations per solve (O(100)) (equivalent to iterative refinement and restarted GMRES)
  - Implemented through Tempus interface to SPARC using a single time-step, not including transient terms

- Need to compute analytic adjoint matrix $\left(\frac{\partial f}{\partial u}\right)^T$
  - Leverage existing Sacado tangent capabilities to compute $\frac{\partial f}{\partial u}V$ for any matrix V
  - Use graph coloring provided by Zoltan2 to find V with a small number of columns
  - Form adjoint through explicit transpose

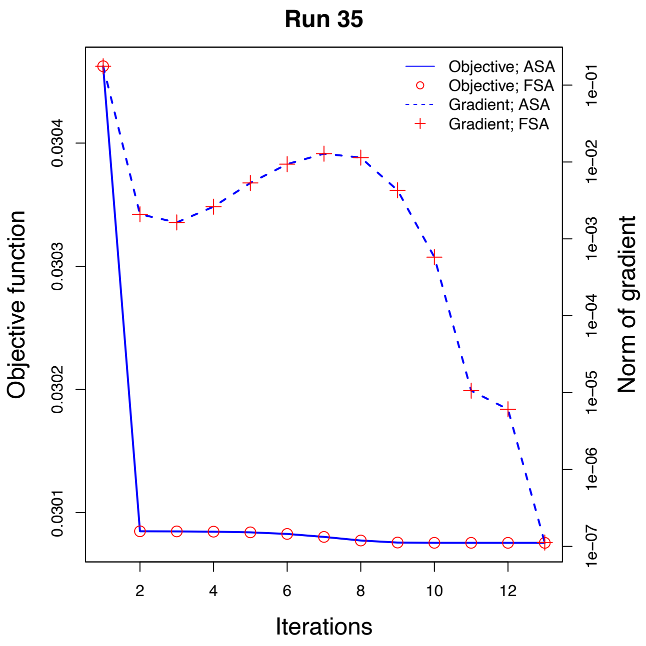
Run 35 Calibration (Fine Mesh)

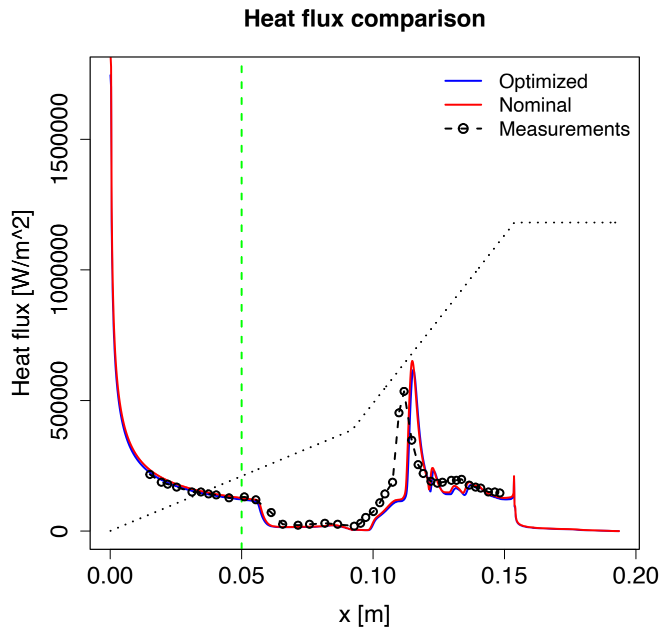
	Experiment	ROL Converged	Bayesian (90% CI)
Density [g/m^3]	0.5848 (0.5439, 0.6257)	0.589	0.574 (0.5471, 0.6209)
Velocity [m/s]	2545 (2469, 2621)	2506	2490 (2441, 2653)



Run 35 data set

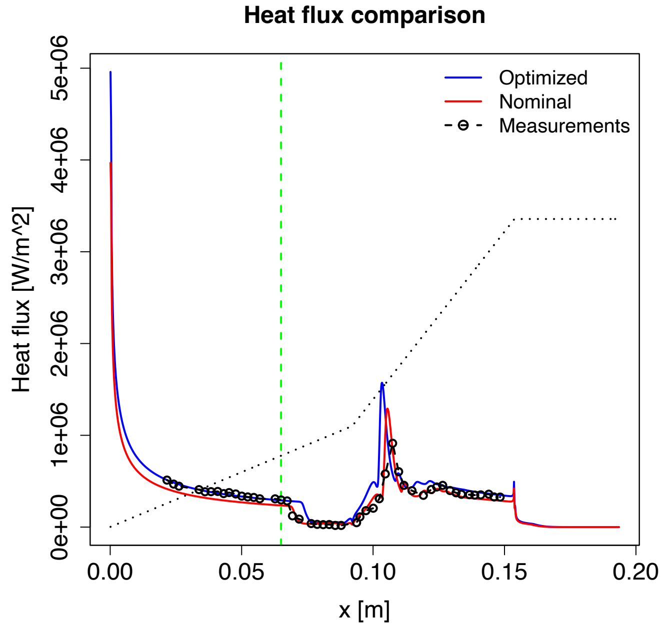
- **Easiest problem**
- Perfect gas
- Vibration equilibrium
- Reaction equilibrium





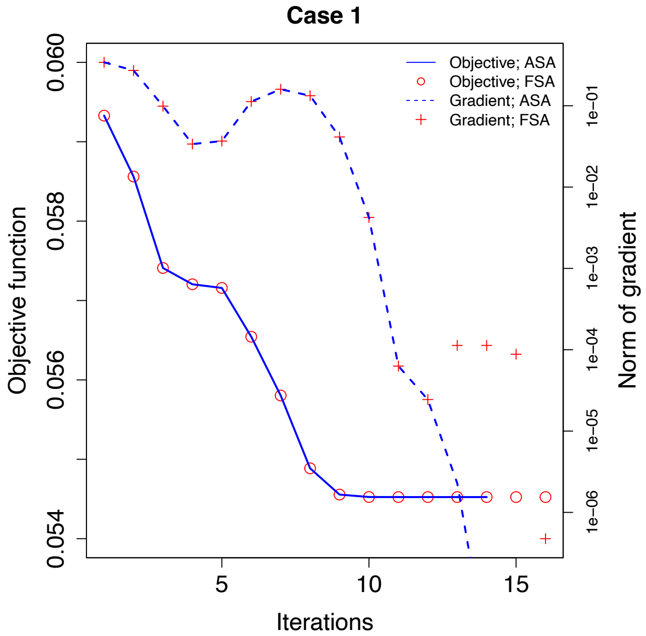
Case 1 Calibration (Medium Mesh)

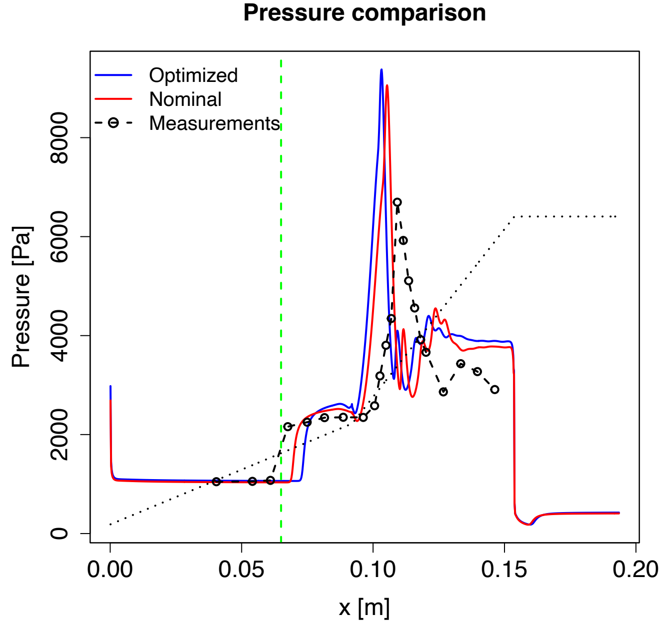
	Experiment	ROL Converged	Bayesian (90% CI)
Density [g/m^3]	0.4990 (0.4641, 0.5339)	0.433	0.4897 (0.4328, 0.5645)
Velocity [m/s]	3246 (3149, 3343)	3540	3340 (3211, 3654)



Case 1 data set

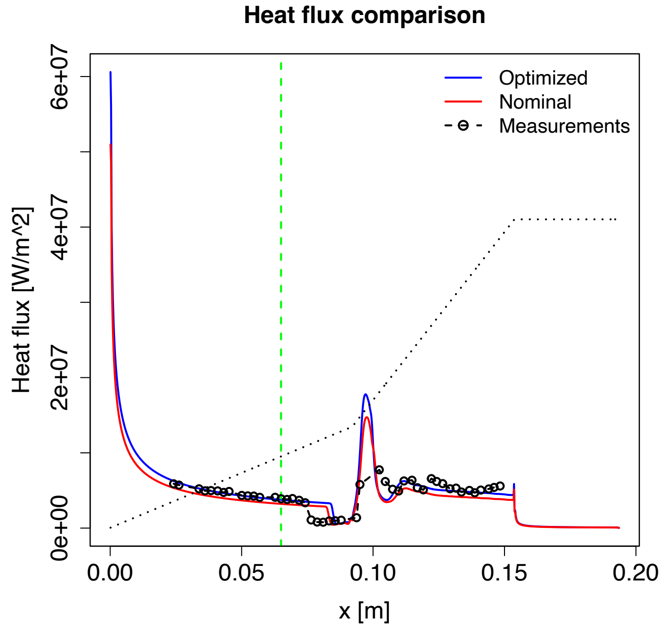
- **Moderate problem**
- Real gas
- Vibration non-equilibrium
- Reaction equilibrium





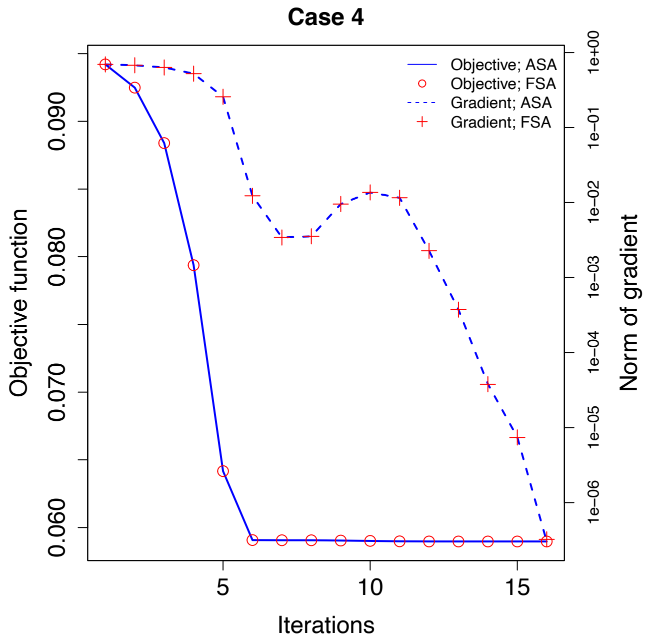
Case 4 Calibration (Coarse Mesh)

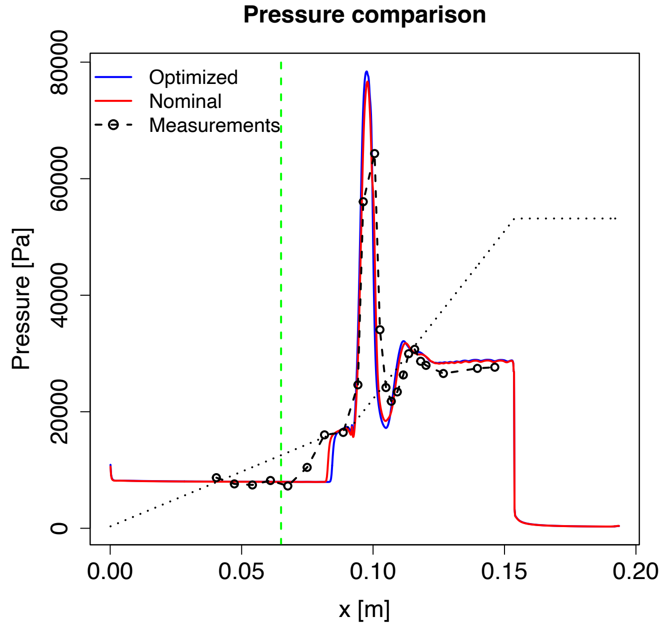
	Experiment	ROL Converged	Bayesian (90% CI)
Density [g/m^3]	0.9840 (0.9151, 1.053)	0.866	0.8608 (0.7996, 1.0396)
Velocity [m/s]	6479 (6285, 6673)	6940	7060 (6380, 7089)



Case 4 data set

- **Hardest problem**
- Real gas
- Vibration non-equilibrium
- Reaction non-equilibrium





Computational Cost Comparisons

Run-time (in seconds) for forward (FSA) and adjoint (ASA) sensitivity computations.

Problem	Mesh	State (SPARC)	State (Tempus)	Adjoint	ASA Total	FSA Total	x-Speedup
Run 35	Coarse	58	79	3	82	208	2.5
	Medium	110	149	3	152	474	3.1
	Fine	622	829	13	842	2311	2.7
Case 1	Coarse	440	527	6	533	1890	3.5
	Medium	1709	2146	12	2158	7393	3.4
	Fine	12440	15219	41	15259	51628	3.4
Case 4	Coarse	558	727	7	734	2861	3.9
	Medium	7139	9081	17	9099	33121	3.6
	Fine	15024	18976	44	19020	68221	3.6

Memory high-water mark (in MB) for forward and adjoint sensitivity computations.

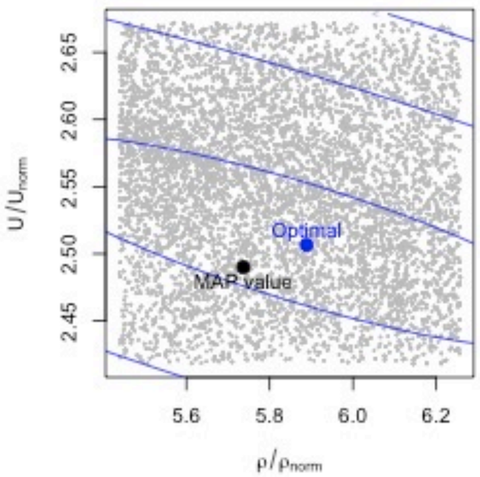
Problem	Mesh	State (SPARC)	ASA	FSA
Run 35	Coarse	253	323	265
	Medium	276	395	297
	Fine	287	534	325
Case 1	Coarse	257	400	275
	Medium	282	555	317
	Fine	302	824	371
Case 4	Coarse	257	403	275
	Medium	281	515	317
	Fine	303	883	371

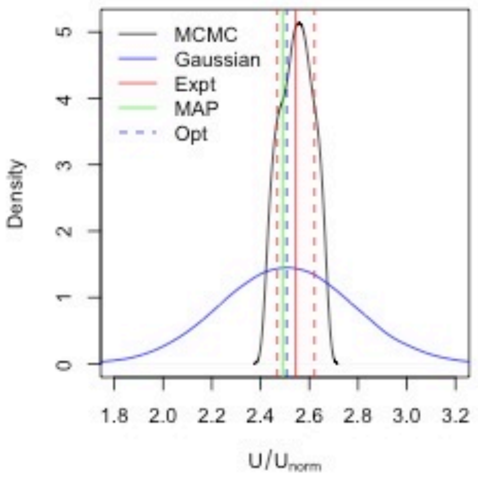
- Tempus-based state integration about 25% slower than SPARC-native implementation
  - Due to extra residual computations for consistent computation of CFL-limited time step required by Tempus
- Cost of the adjoint solve is insignificant compared to forward state integration
- Adjoint sensitivity is about 3 times faster than forward approach on these problems
  - Directly translates into comparable speedup for ROL calibration
- Adjoint approach requires substantially more memory
  - Due to cost of storing true adjoint matrix and Tempus implementation requiring several copies of this matrix
  - A limiting factor for small memory environments such as GPUs

ROL and Surrogate-based Inversion Comparisons

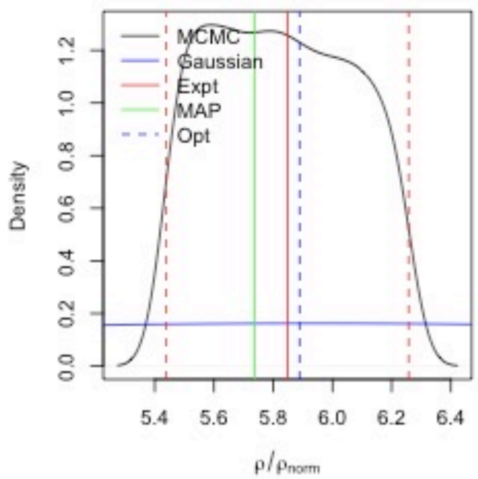
- The ROL-based calibration is a deterministic inversion approach which doesn't directly provide estimates of uncertainty
- Assuming
  - QoIs (heat flux, pressure, total enthalpy, Pitot pressure) differ from experimental values by additive Gaussian noise
  - This noise is sufficiently small such that QoIs depend approximately linearly on the calibrated parameters
- Then
  - The posterior of the calibrated parameters is (approximately) Gaussian
  - Solution of the ROL calibration problem is equivalent to MLE for the mean of the posterior
  - The inverse of the Hessian of the objective function (negative log likelihood) at the calibrated parameters is an estimate of the covariance of the posterior
- Does this provide a useful estimate of uncertainty?
- Estimate Hessian by differencing gradient after termination of ROL calibration (3 extra gradient computations)

Run 35





Summary	ρ_∞/ρ_{norm}		U_∞/U_{norm}	
	MCMC	Deterministic	MCMC	Deterministic
MAP	5.74	5.89	2.490	2.506
Mean	5.83	5.89	2.548	2.506
Median	5.82	5.89	2.55	2.506
IQR	(5.62, 6.03)	(4.24, 7.54)	(2.5, 2.6)	(2.32, 2.69)
90% CI	(5.47, 6.21)	(1.88, 9.9)	(2.44, 2.65)	(2.06, 2.95)

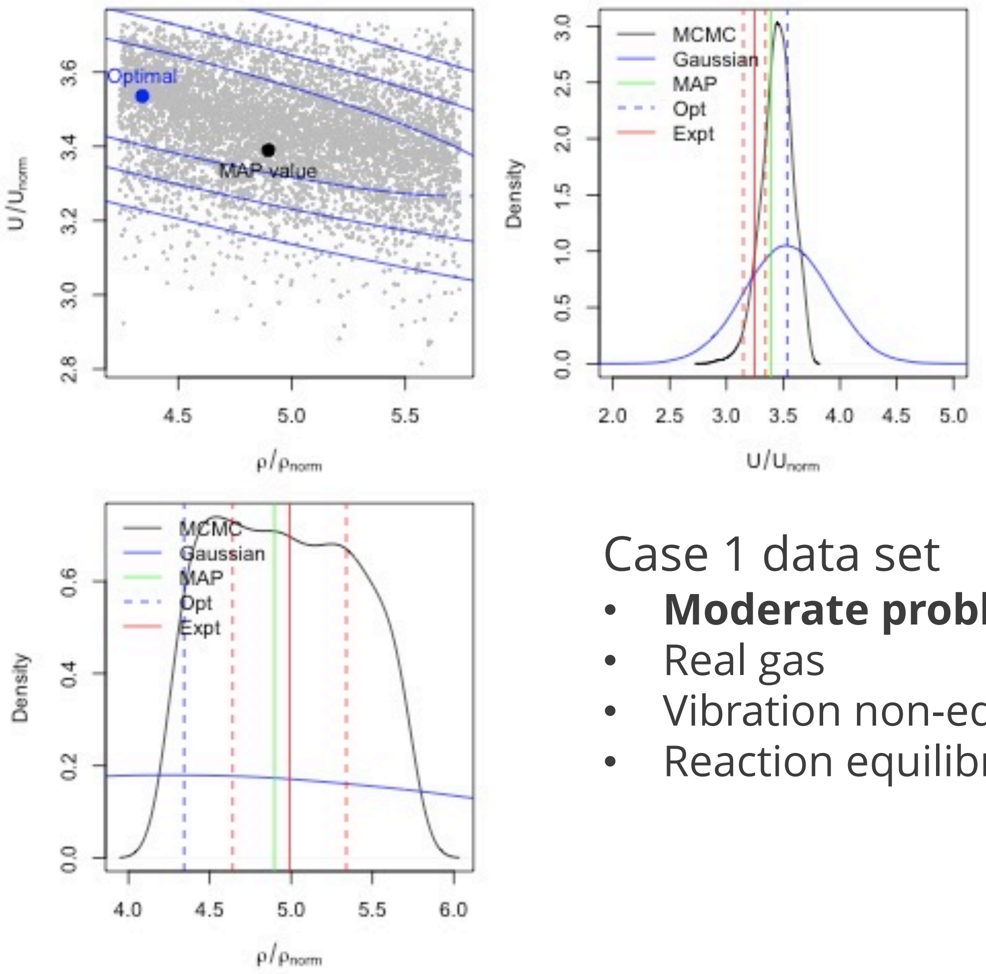


Run 35 data set

- **Easiest problem**
- Perfect gas
- Vibration equilibrium
- Reaction equilibrium

- Deterministic calibrated parameters
  - Agree well with MCMC Map/Mean/Median
  - Are within the range of experimental uncertainty
- IQR for velocity comparable, but 90% CI is too wide
- Both IQR and 90% CI for density are way too wide

Case 1



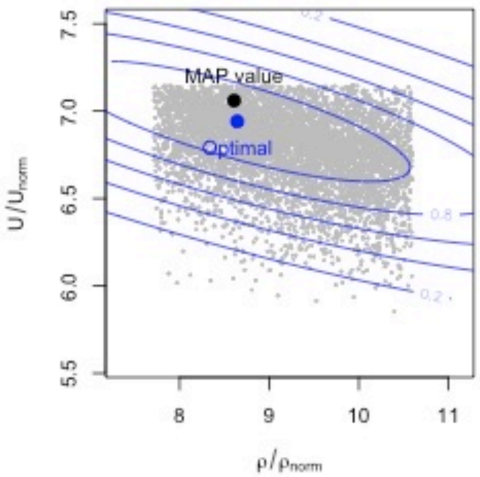
Summary	ρ_∞/ρ_{norm}		U_∞/U_{norm}	
	MCMC	Deterministic	MCMC	Deterministic
MAP	4.897	4.33	3.34	3.54
Mean	4.96	4.33	3.44	3.54
Median	4.95	4.33	3.45	3.54
IQR	(4.6, 5.31)	(2.86, 5.79)	(3.36, 3.53)	(3.28, 3.79)
90% CI	(4.33, 5.64)	(0.73, 7.96)	(3.21, 3.65)	(2.92, 4.16)

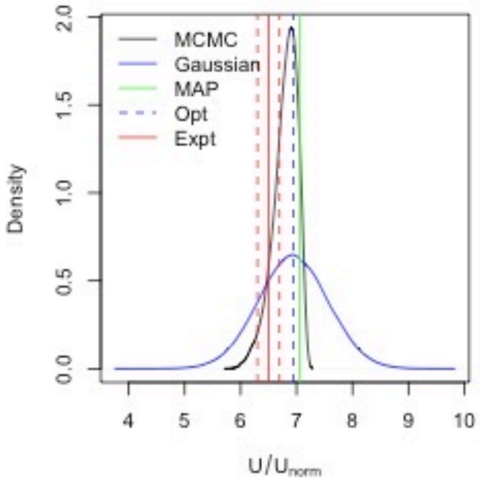
Case 1 data set

- **Moderate problem**
- Real gas
- Vibration non-equilibrium
- Reaction equilibrium

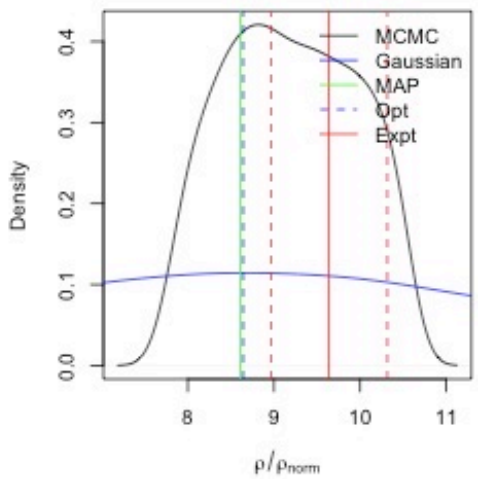
- Deterministic calibrated parameters
  - Agree somewhat with MCMC Map/Mean/Median
  - Are outside the range of experimental uncertainty
- IQR for velocity somewhat comparable, but 90% CI is too wide
- Both IQR and 90% CI for density are way too wide

Case 4





Summary	ρ_∞/ρ_{norm}		U_∞/U_{norm}	
	MCMC	Deterministic	MCMC	Deterministic
MAP	8.608	8.66	7.06	6.94
Mean	9.186	8.66	6.8	6.94
Median	9.169	8.66	6.834	6.94
IQR	(8.57, 9.81)	(6.32, 11.0)	(6.67, 6.96)	(6.52, 7.36)
90% CI	(8.0, 10.4)	(2.98, 14.37)	(6.38, 7.09)	(5.92, 7.95))



Case 4 data set

- **Hardest problem**
- Real gas
- Vibration non-equilibrium
- Reaction non-equilibrium

- Deterministic calibrated parameters
  - Agree somewhat with MCMC Map/Mean/Median
  - Are outside the range of experimental uncertainty
- IQR for velocity somewhat comparable, but 90% CI is too wide
- Both IQR and 90% CI for density are way too wide

Conclusions

- Demonstrated adjoint sensitivities for hypersonic flows can be successful and facilitate embedded calibration/optimization
  - Capabilities are built-in to SPARC and don't require construction of surrogate models
  - Provides similar calibrated parameters to surrogate/Bayesian approach (uncertainty estimates not useful though)
  - Adjoint sensitivity calculation is essentially free compared to forward state solve
- Approach leverages multiple Trilinos capabilities
  - Sacado AD
  - ROL embedded optimization
  - Tempus time integration
  - Zoltan2 graph coloring
  - Ifpack2 block tri-diagonal solver
  - Belos GMRES
- Techniques provide a foundation for future distinguishing capabilities
  - Field inversion
  - Model form error estimation
  - Construction of ML-based turbulence model closures
  - Shape optimization

Backup Slides

Sacado: AD Tools for C++ Applications

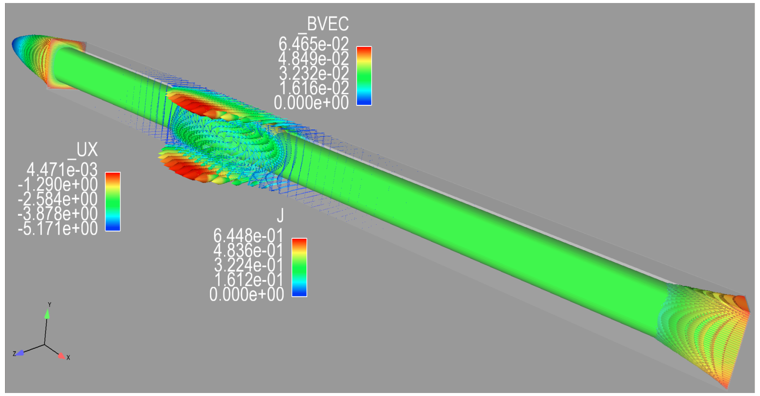
Automatic differentiation package in Trilinos (Phipps and Gay)

Operator overloading-based approach
- Sacado provides C++ data types implementing AD
- Type of variables in code replaced by AD data type
- AD object for each variable stores value of that variable and its derivatives
- Mathematical operations replaced by overloaded versions implementing chain-rule
- Expression templates reduce overhead

Primary tools are Sacado's forward mode (a.k.a. tangent mode) AD tools
- Integrates with Kokkos for efficient differentiation of thread-parallel programs
- Compute sparse Jacobian's for finite element-type codes by differentiating at element level and manually assembling global Jacobian
- Global Jacobian vector products

TRILINOS

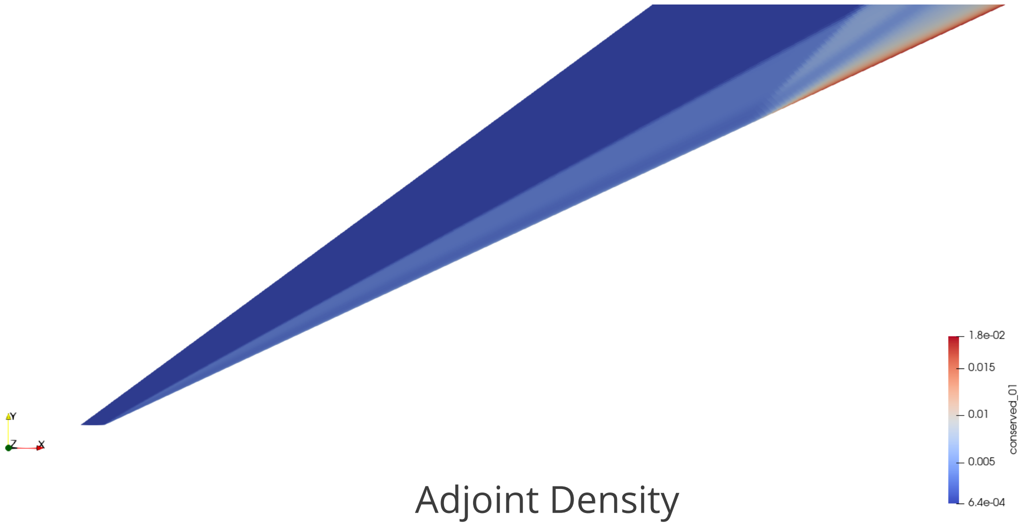
http://github.com/trilinos



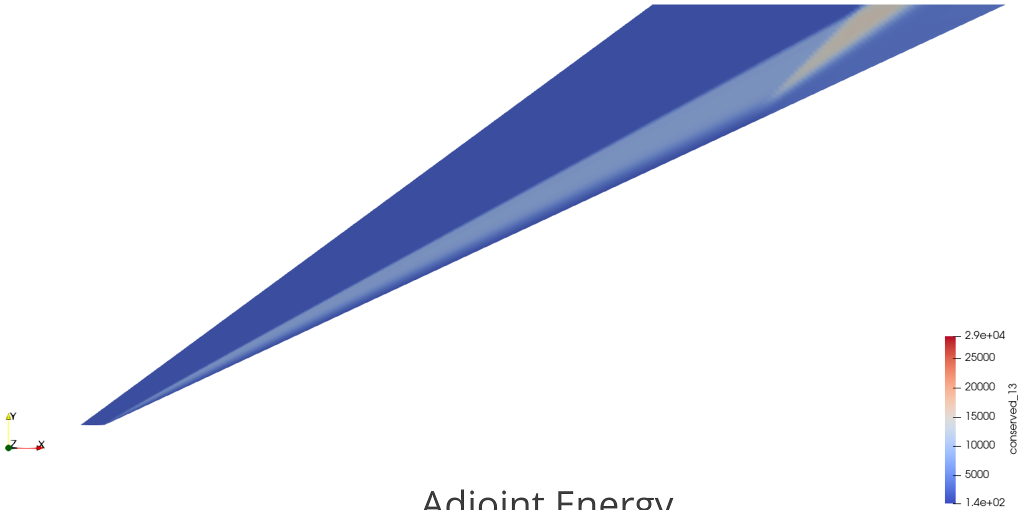
Iso-velocity adjoint surface for fluid flow in a 3D steady MHD generator in Drekar computed via Sacado (Courtesy of T. Wildey)

Adjoint Flow Visualizations (Run35, Medium Mesh)

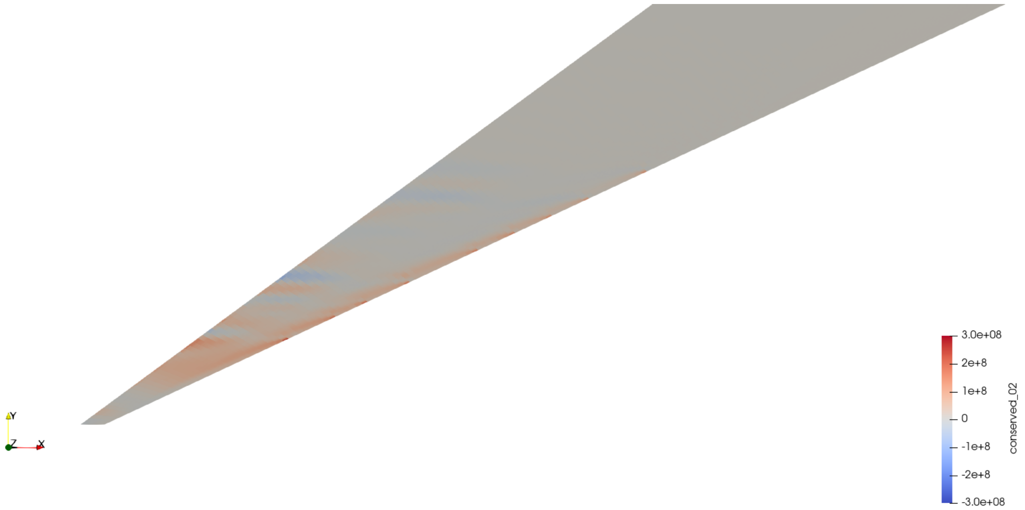
Primal Density



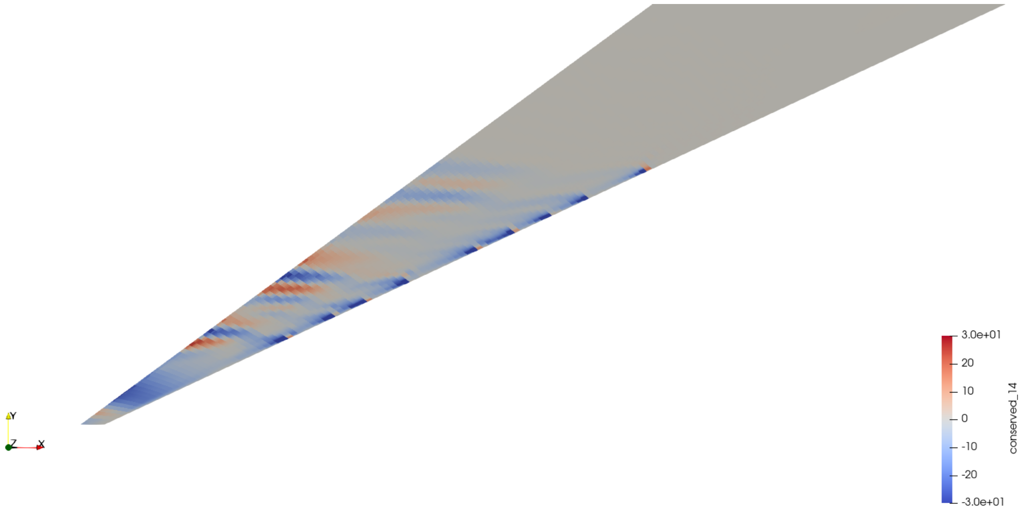
Primal Energy



Adjoint Density

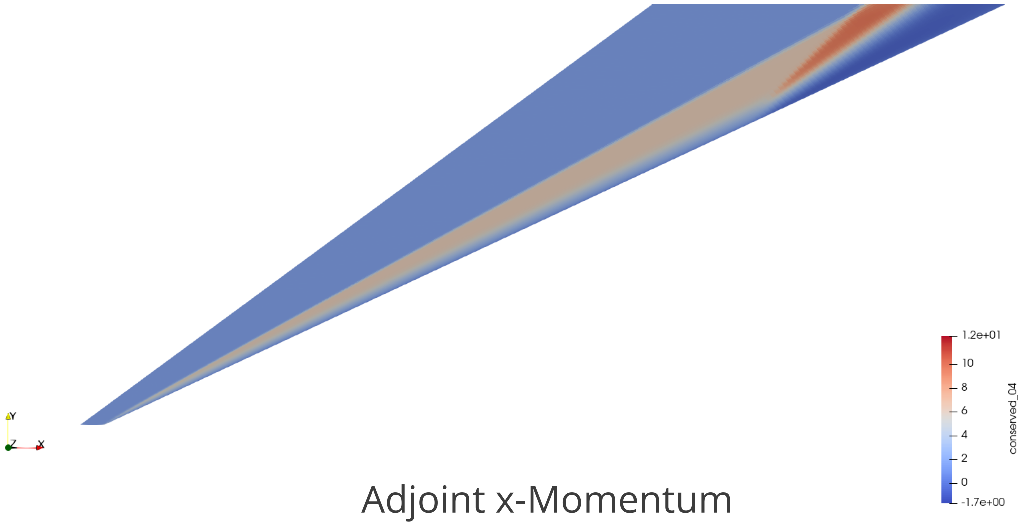


Adjoint Energy

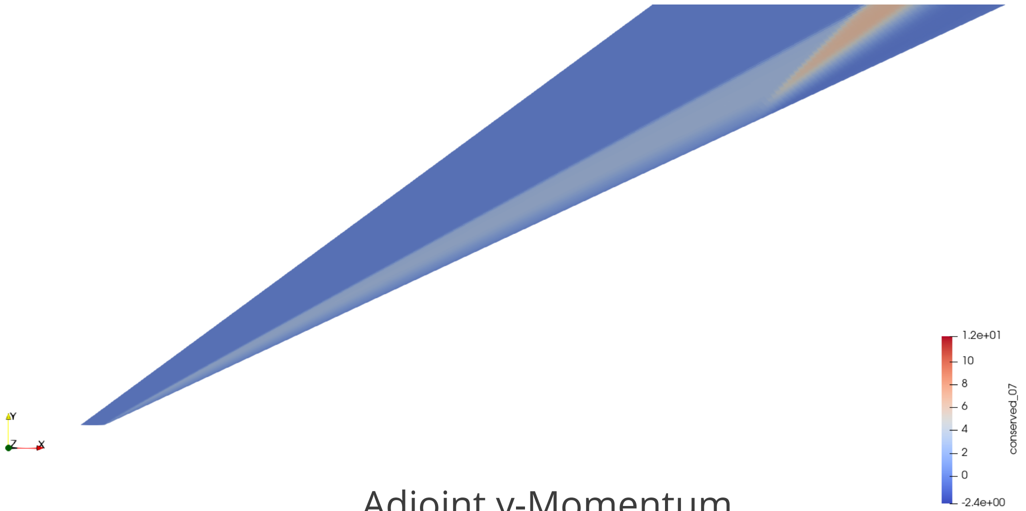


Adjoint Flow Visualizations (Run35, Medium Mesh)

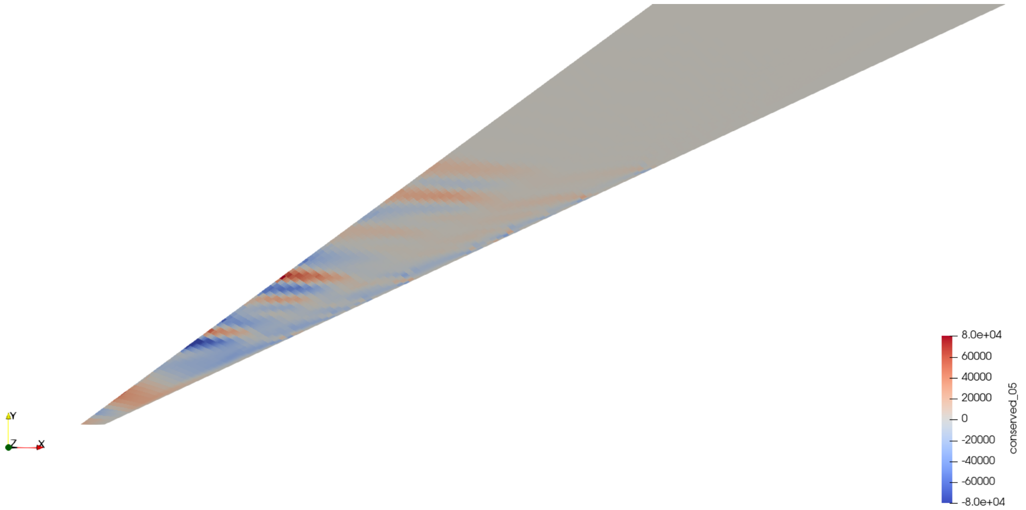
Primal x-Momentum



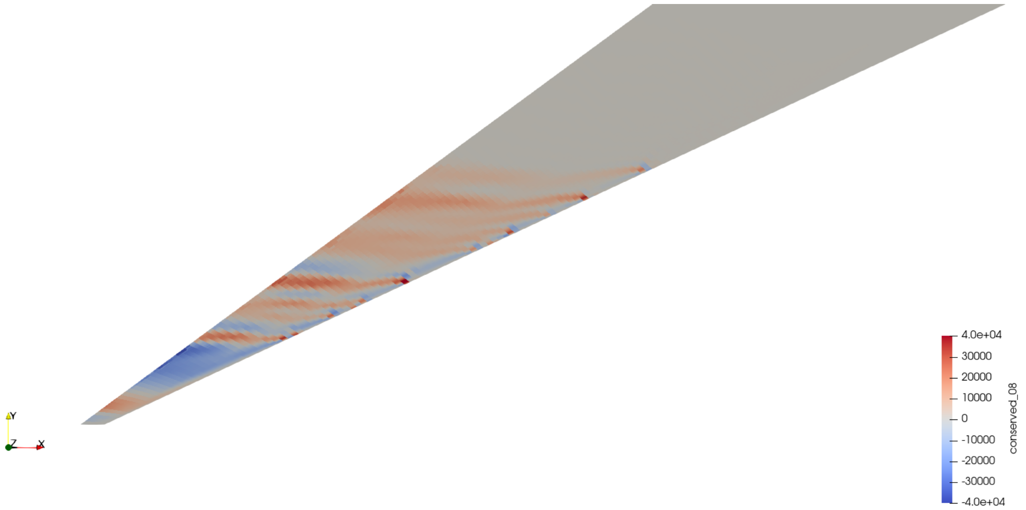
Primal y-Momentum



Adjoint x-Momentum

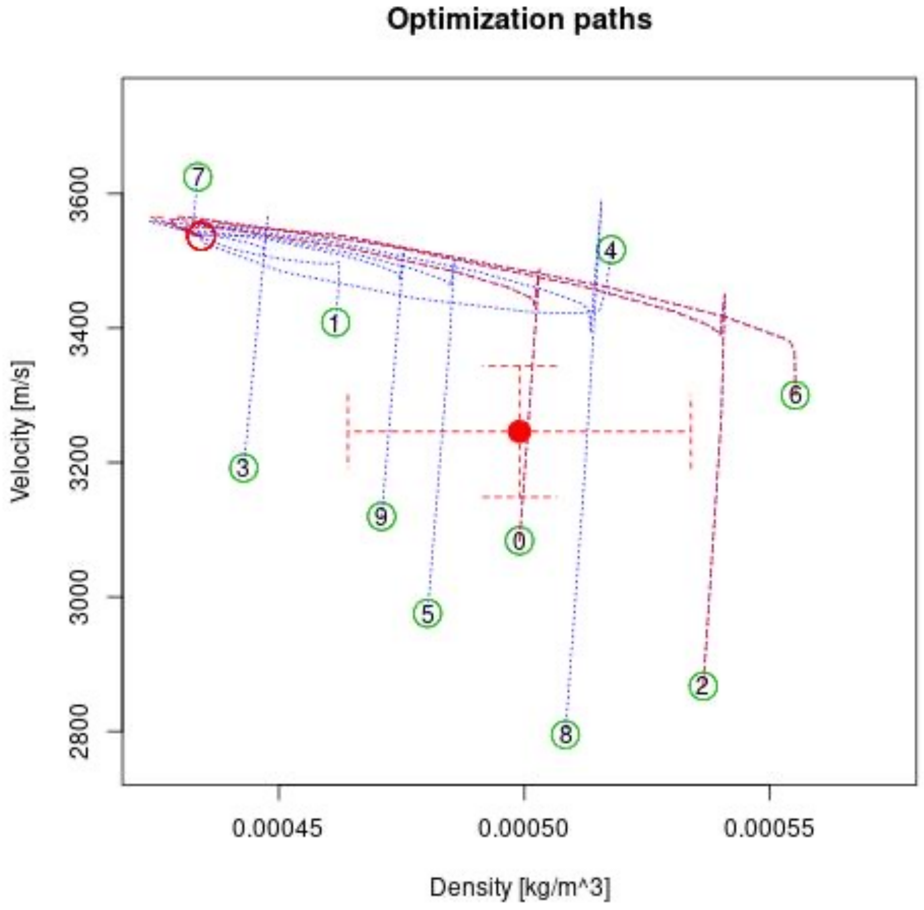


Adjoint y-Momentum

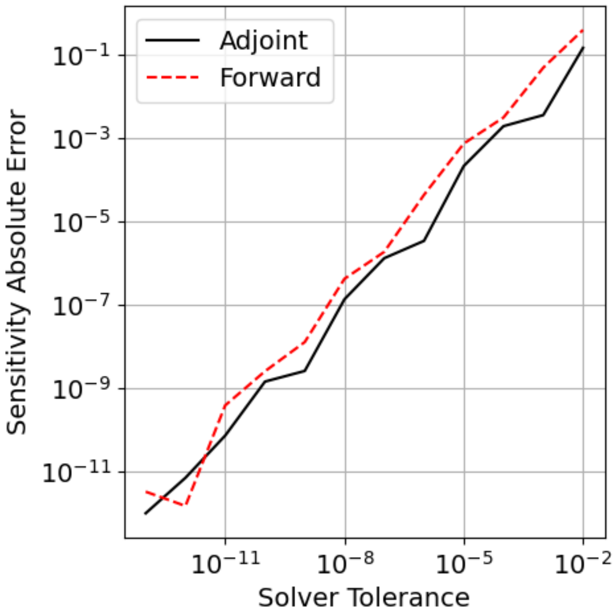


Multi-start Optimization

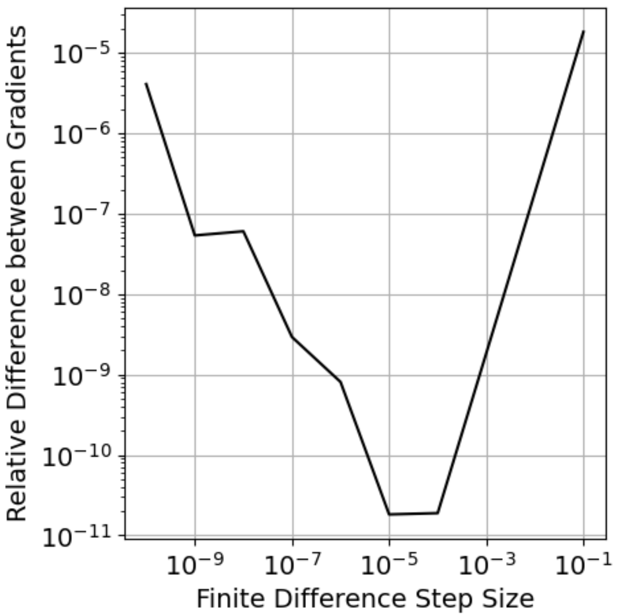
- Case 1 ROL-based calibration results in calibrated parameters outside the range of experimental uncertainty
  - Whereas surrogate-based calibration was on the edge
- Is this due to the problem formulation, or are there possibly multiple local minima?
- Executed ROL inversions (on medium mesh) with 10 initial guesses to find out
- All initial guesses converge to same minimum
- As in Case 4, the freestream conditions quoted by experimentalists are likely wrong



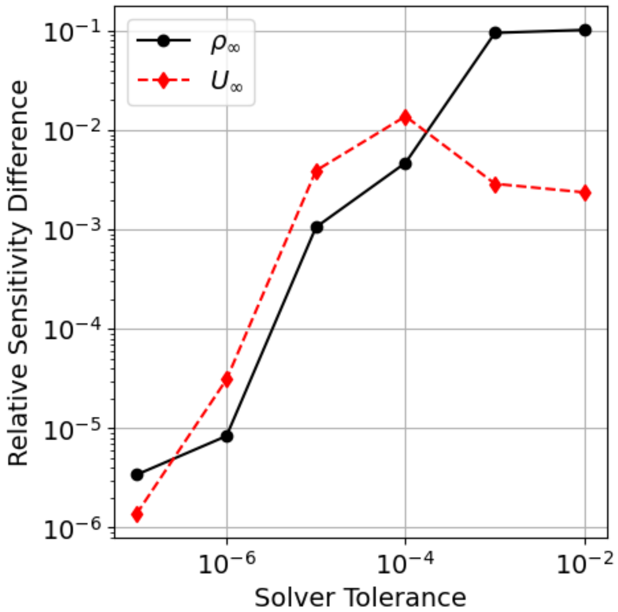
Adjoint Sensitivity Accuracy Comparisons



Forward and adjoint sensitivity error on a coarse mesh blunt wedge regression test problem.



Comparison between adjoint sensitivities and finite differences on a coarse mesh blunt wedge regression test problem.



Difference between forward and adjoint sensitivities on coarse mesh Run 35 problem

- Explored (combined) forward and adjoint sensitivity accuracy on several test problems
  - Coarse mesh blunt wedge regression test problem
  - Coarse mesh Run 35 problem
- Very similar accuracy between methods across wide range of solver tolerances.
- Sensitivities also agree with finite differences to expected precision, verifying correctness