Solution-verification, grid-adaptation and uncertainty quantification for chaotic turbulent flow problems

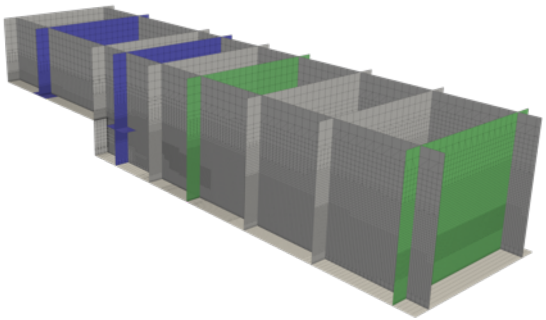
Johan Larsson
University of Maryland

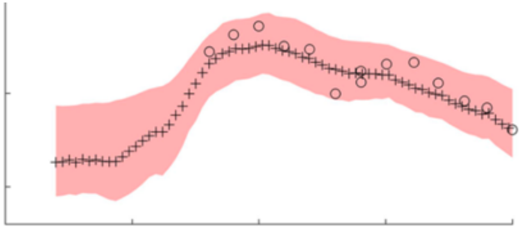
Qiqi Wang
MIT

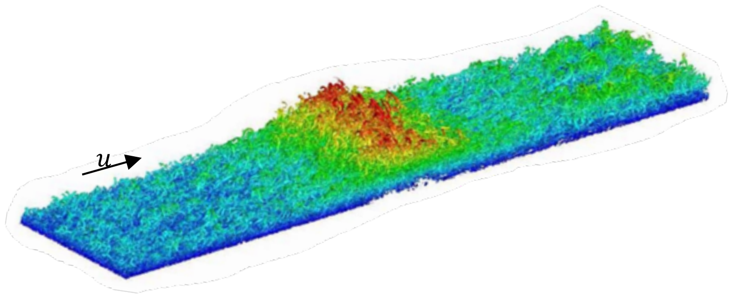
Ivan Bermejo-Moreno
University of Southern California

PSAAP-3 Kickoff Meeting

Aug 18-19, 2020







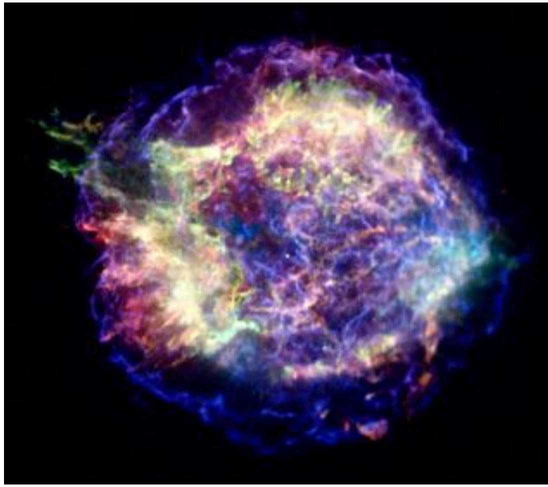
Motivation and target problems

PDEs exhibiting turbulence – chaotic, truly multiscale (broadband) solutions

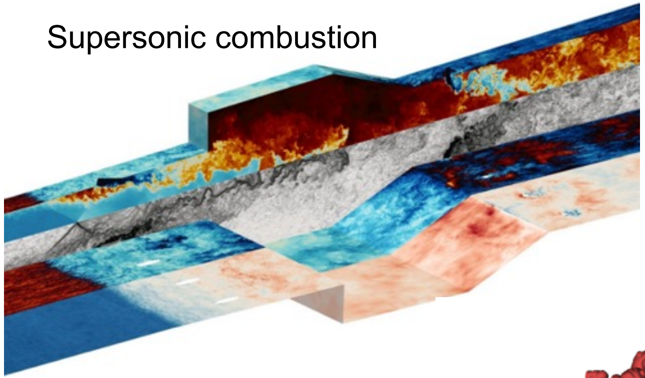
In general, turbulence + other physics effects (shocks, chemistry, …) – this work is focused on the turbulence

Typically with many problem parameters:
Reynolds number, initial disturbance, geometry,
fluid properties, model form and coefficients, …

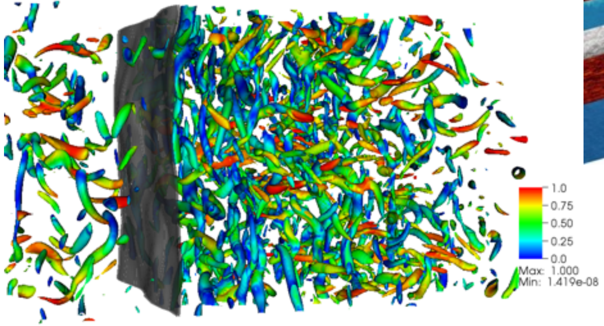
Supernovae explosions



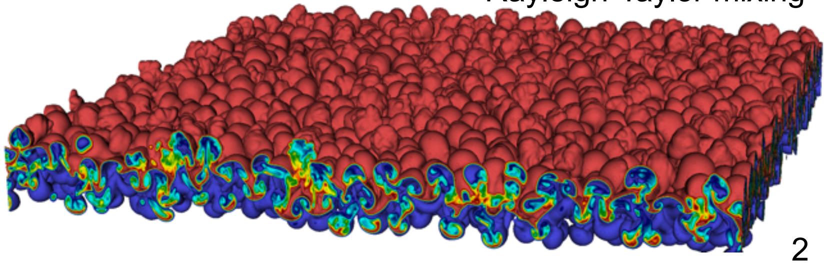
Supersonic combustion



Shock/turbulence interactions



Rayleigh-Taylor mixing



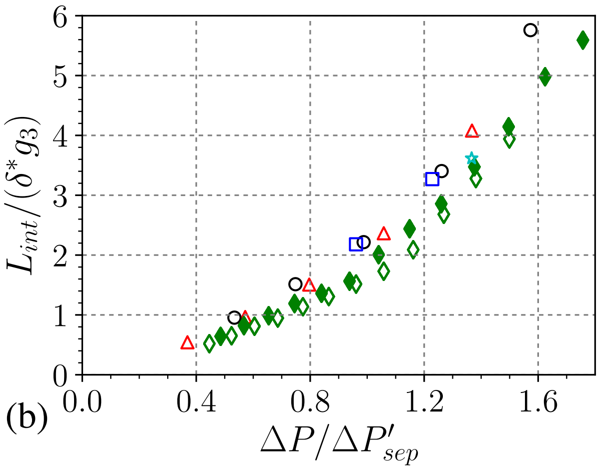
The need for the sensitivity gradient ∇J in UQ of turbulent flows

- $\mathcal{O}(100)$ or more parameters – must avoid "curse-of-dimensionality" (e.g., no polynomial chaos).
- Different engineering challenges require different levels of UQ:

a)	Approximate error bars	Approximate σ_J sufficient	Sensitivity (∇J) is sufficient
b)	Confidence intervals	Approximate σ_J, approx. pdf	With ∇J, gradient-enhanced Kriging
c)	Full pdf, long-tail events,…	Must sample	With ∇J, can focus sampling

- Error attribution – ∇J provides direct information
- Estimating the effects of grid-induced errors – requires ∇J due to the extremely high dimensionality

Sample VVUQ problem:
Does DNS (red/blue/black) agree with experiment (green)?



Project objectives

Main goal: to develop a mathematical foundation for VVUQ of PDEs that exhibit chaotic/broadband dynamics

Specific objectives:

1. Develop algorithms for computing the **sensitivity gradient ∇J** for chaotic turbulence problems
   Consider methods at different levels of accuracy and cost

   $\nabla J \equiv$ gradient in parameter space of a QoI J

2. Develop algorithms that use the sensitivity gradient (of different levels of accuracy) to perform:
   - **uncertainty estimation** - how large is the error in J?
   - **error attribution** - what factors contribute the most to my estimated error/uncertainty?
   - **grid adaptation** - how to design/refine grid to optimally reduce the error in J?

Focus application: turbulent boundary layer over flat plate, with varying pressure gradients

The developed methods will be applicable to a wide range of chaotic problems

Overall approach

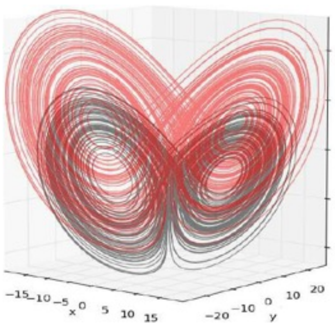
The key challenge: In realistic chaotic/broadband (turbulence) problems with many parameters, computing ∇J is only affordable via adjoints – but the **adjoint equations diverge exponentially**, and the available solutions (shadowing, ensembles of short-duration adjoints) are infeasibly costly or inapplicable in general.

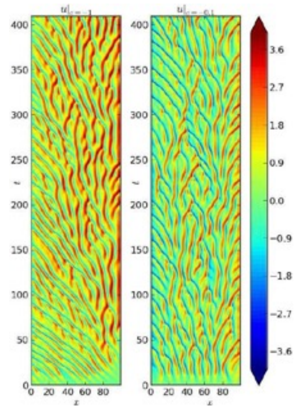
Overall **approach**:

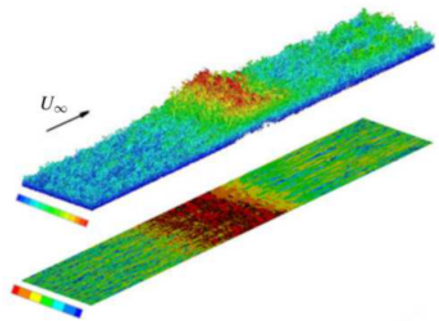
1. Develop multiple methods to compute ∇J stemming from different approximations and trade-offs:
   a) Methods starting from chaos-theory (shadowing and others)
   b) Methods starting from physics-inspired reduced-order modeling
   c) Methods combining ideas from these philosophically different approaches
2. Local error estimation, for use in both uncertainty estimation and grid-adaptation
3. Compare these methods on a validation / benchmark problem – learn about strengths, weaknesses, inherent limitations, possible improvements, ...

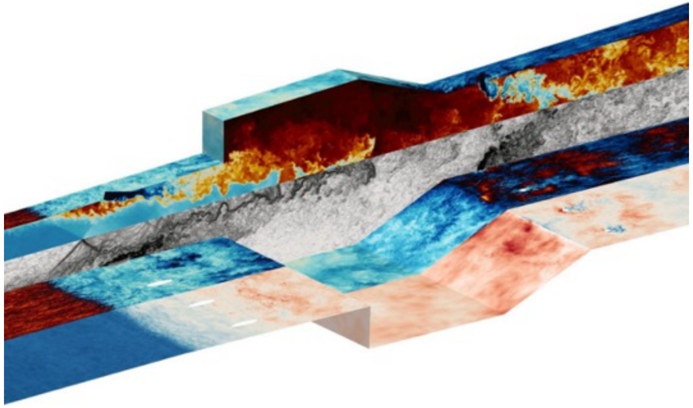
Validation/benchmark problem

Chaotic ODEs/PDEs:









Lorenz
(ODE)

Kuramoto-
Sivashinsky
(2D PDE)

Incompressible
Navier-Stokes
(broadband
turbulence)

Compressible
and/or multi-phase
Navier-Stokes
(discontinuities)

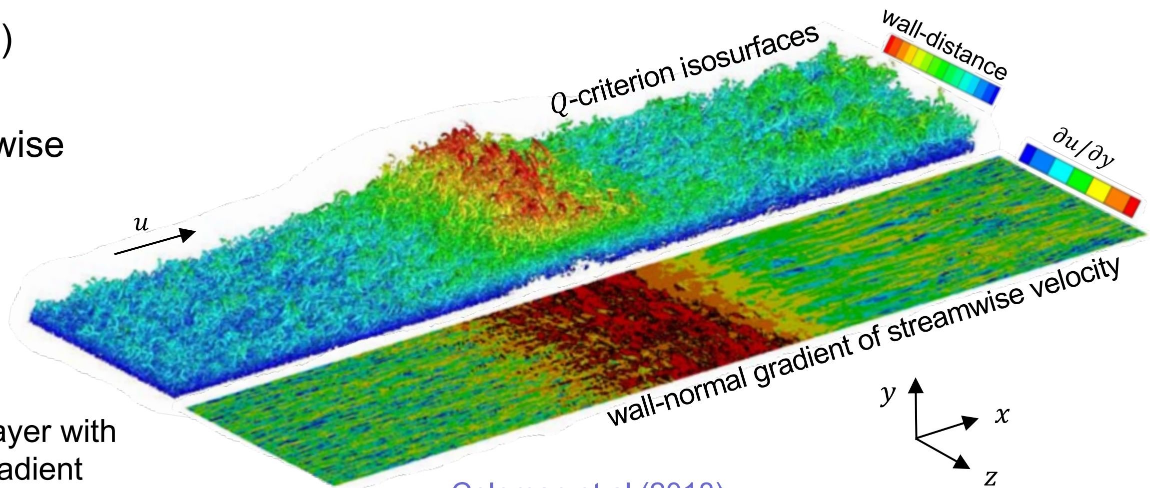
With chemistry,
tabulated EoS,
radiation, particles
(the real NNSA
applications)

"Everything should be made as simple as possible, but no simpler"
Albert Einstein

Validation/benchmark problem

Our choice: **Flat-plate turbulent boundary layer with spatial pressure gradient**

- Space of parameters for sensitivity studies includes:
  - Natural parameters – can be highly controlled in our own reference simulations:
    - Reynolds number
    - State of incoming boundary layer (thickness + mean & turbulent wall-normal profiles)
    - Spatial mean pressure profiles in streamwise and spanwise directions
  - Modeling parameters (evaluation of epistemic uncertainties):
    - Subgrid-scale (SGS) model
    - Wall model (future studies)
- Quantities of Interest:
  - Drag force streamwise & spanwise
  - Size of separation bubble

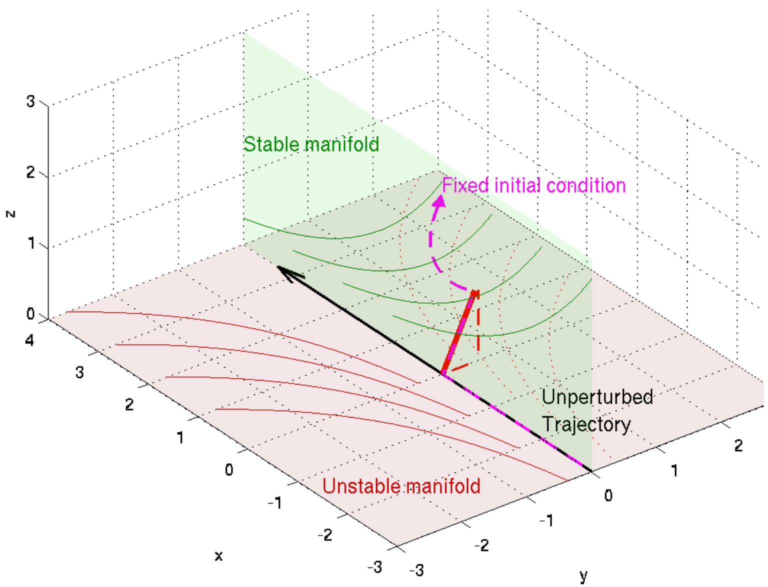


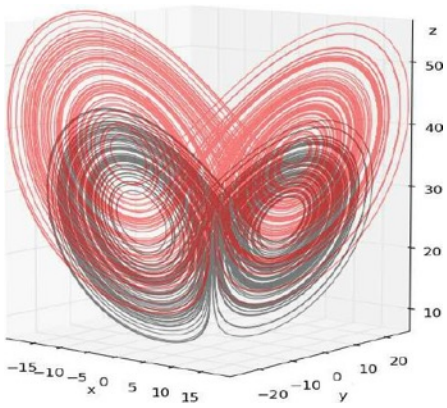
Flat plate boundary layer with non-zero pressure gradient

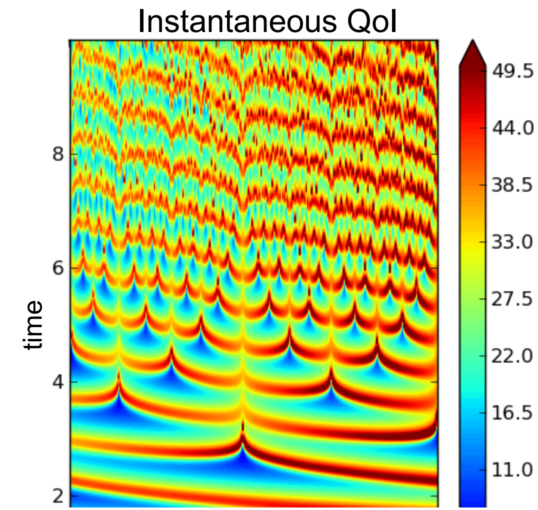
Coleman et al (2018)

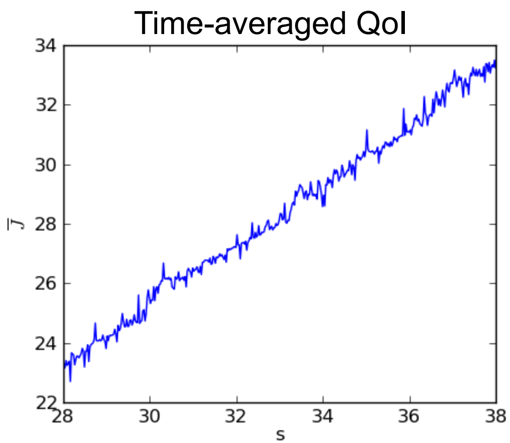
Sensitivity divergence in long-time-averaged chaotic systems[1]

Chaotic problems have unstable manifolds, which by definition cause divergence of the linearized initial-value problem









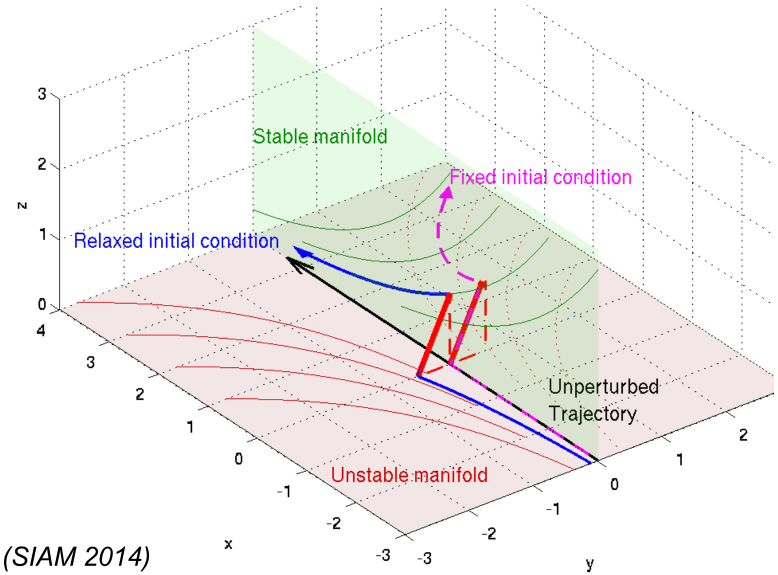
[1] *e.g., Lea et al (Tellus 2000)*

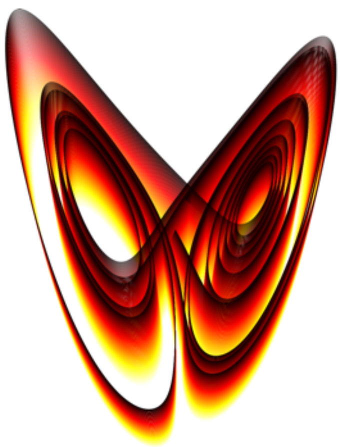
One solution: Least squares shadowing (LSS)[1]

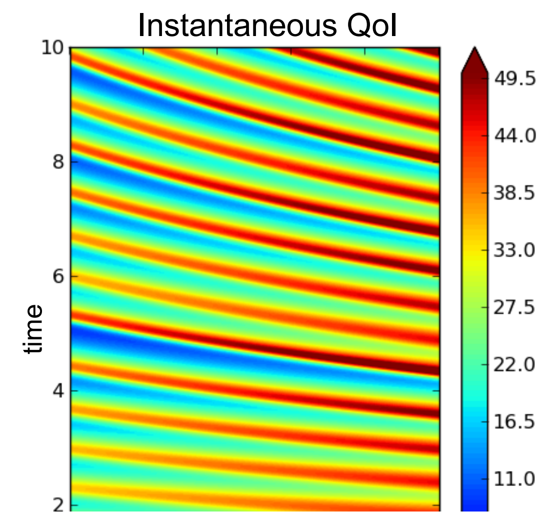
A "shadow trajectory" is defined to stay close to the reference solution at all times, while satisfying the governing equation

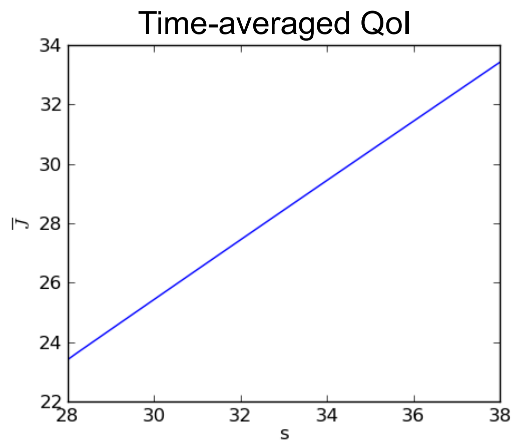
$$\min \frac{1}{2}\int_{T_0}^{T_1} \|u(t) - u_r(t)\|^2 dt$$

$$\text{s.t.} \quad \frac{du}{dt} = f(u; s + \delta s)$$









[1] *Wang (SIAM 2014)*

Shadowing adjoint algorithms

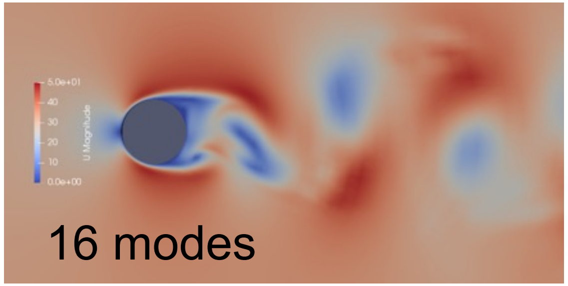
Non-intrusive Least Squares Shadowing[1]

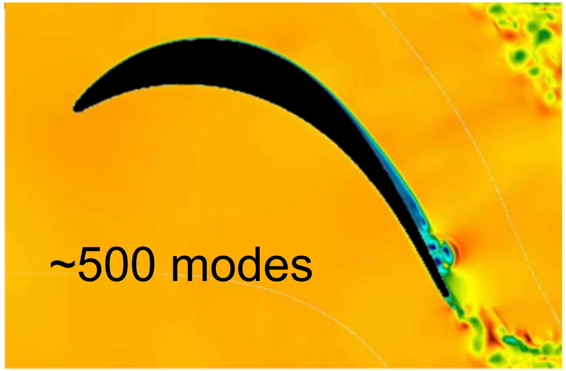
- Non-intrusive (for code with existing adjoint)
- Cost proportional to number of unstable modes

Turbulence has many unstable modes!

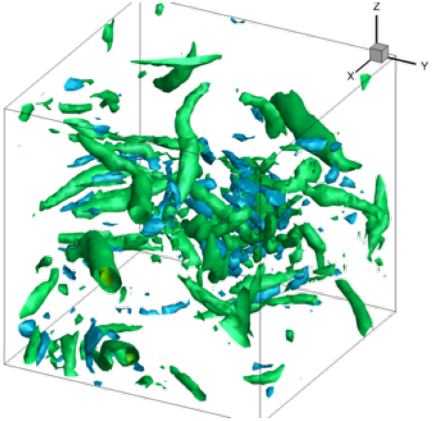
Space-time multigrid[2]

- Insensitive to number of unstable modes
- Intrusive modification of solver





$$\min \frac{1}{2}\int_{T_0}^{T_1} \|u(t) - u_r(t)\|^2 dt$$
$$\text{s.t.} \quad \frac{du}{dt} = f(u; s + \delta s)$$



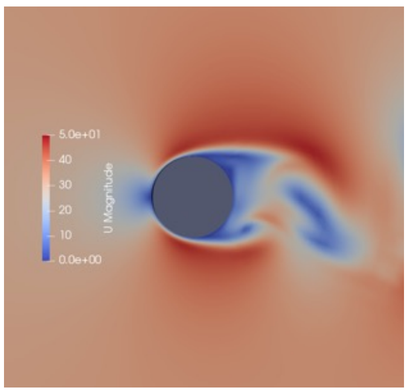
[1] *Ni & Wang (JCP 2017)*
[2] *Blonigan & Wang (NLA 2014)*

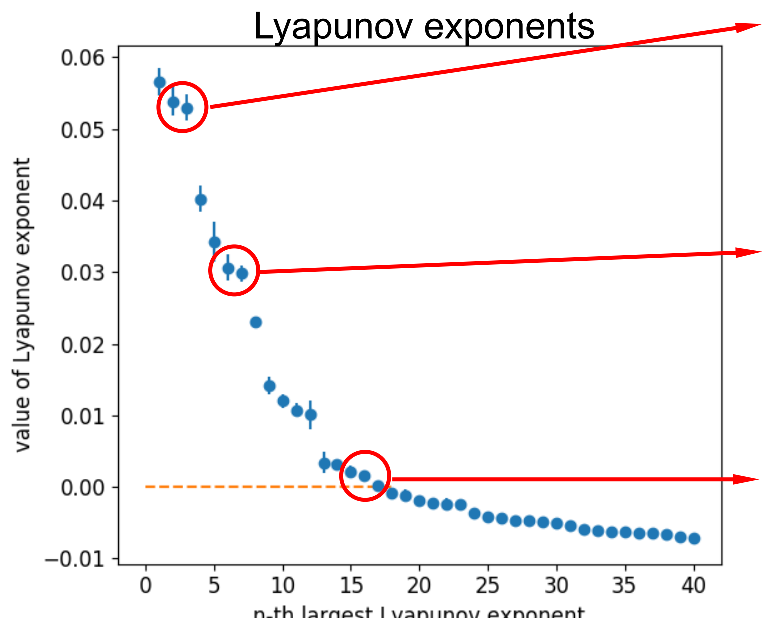
Cost reduction ideas for shadowing algorithm

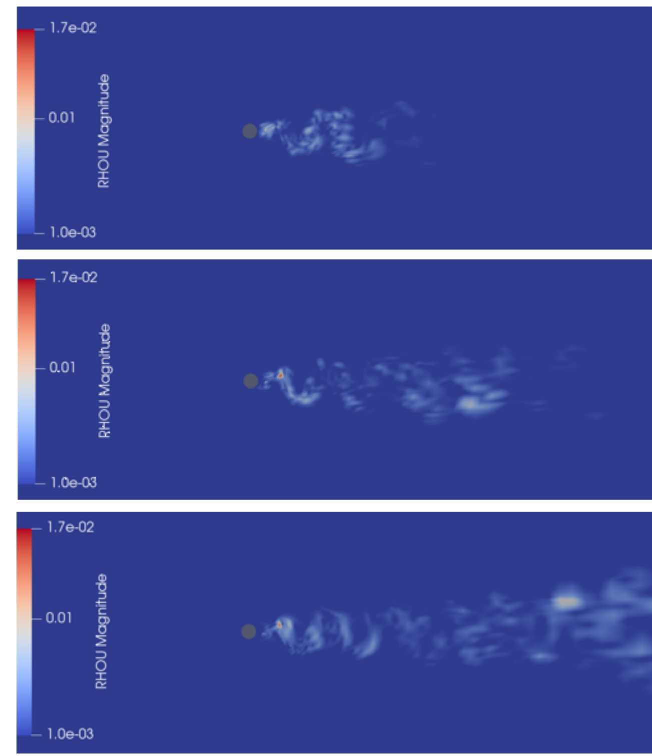
Theorem: a shadowing tangent solution can be found within the affine space of:

1. the tangent restricted to the stable subspace, and
2. the span of all unstable modes

Example: low-Re flow over a cylinder







Cost reduction ideas for shadowing algorithm

Theorem: a shadowing tangent solution can be found within the affine space of:

1. the tangent restricted to the stable subspace, and
2. the span of all unstable modes

Observation: Many unstable modes can have the same high-wavenumber content and differ from each other only in their "modulation", a low-wavenumber multiplicative factor.

Hypothesis: Cost reduction can be achieved through computing multiple unstable modes by solving a single tangent equation, then apply different low-wavenumber modulations to the solution.

In addition: There are other approaches (beyond shadowing) based on similar ideas from chaos theory – we will pursue these too.

Another possible solution: Adjoint of a reduced-order model

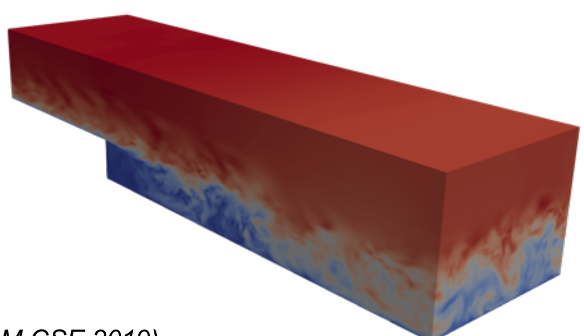
Ideas first developed in the context of output-based grid-adaptation

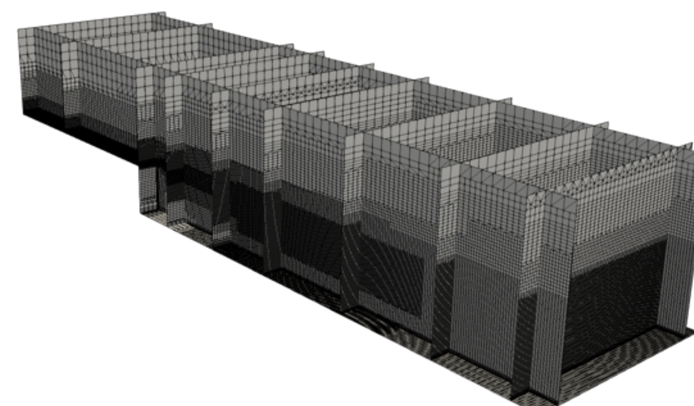
General idea: expand outputs in terms of deterministic statistics of the solution, e.g.:

$$J_i(u_{\widehat{H}}) \approx J_i(u_{\widehat{h}}) + \left\langle \frac{\partial J_i}{\partial \overline{u}}, \overline{u}_{\widehat{H}} - \overline{u}_{\widehat{h}} \right\rangle + \left\langle \frac{\partial J_i}{\partial \overline{u'u'}}, \overline{u'u'}_{\widehat{H}} - \overline{u'u'}_{\widehat{h}} \right\rangle + \dots ,$$

Then model difference in statistics in a mean sense, from exact corresponding equations (mean, variance, etc) with modeled source terms. Corresponding adjoint equations are steady in time, and deterministic – very cheap to solve… but possibly poorly linking local errors to the outputs.

Proof of feasibility (in the context of grid-adaptation) on a modified Kuramoto-Sivashinsky equation[1] and flow over a backward-facing step[2]





[1] *Larsson (AIAA SciTech 2018, SIAM CSE 2019)*
[2] *Toosi & Larsson (AIAA Aviation 2018)*

Adjoint of a reduced-order model

In this project, use this general idea of expanding the output in terms of deterministic statistics

$$J_i(u_{\widehat{H}}) \approx J_i(u_{\widehat{h}}) + \left\langle \frac{\partial J_i}{\partial \overline{u}}, \overline{u}_{\widehat{H}} - \overline{u}_{\widehat{h}} \right\rangle + \left\langle \frac{\partial J_i}{\partial \overline{u'u'}}, \overline{u'u'}_{\widehat{H}} - \overline{u'u'}_{\widehat{h}} \right\rangle + \dots ,$$

and expand the range of accuracy/cost possibilities:

- Where to truncate the expansion? After mean, (co-)variances, higher-order?
- How to model unclosed terms? Inferred eddy-viscosity and analogously for higher moments?
- How does the accuracy depend on the flow, the parameter, and the QoI?
- Consider averages other than in time, for use in unsteady problems – e.g., applicable to Rayleigh-Taylor etc

Consider complexity-reduction that retains some chaotic character but with very few unstable modes – could then be combined with the chaos-theory-based methods?

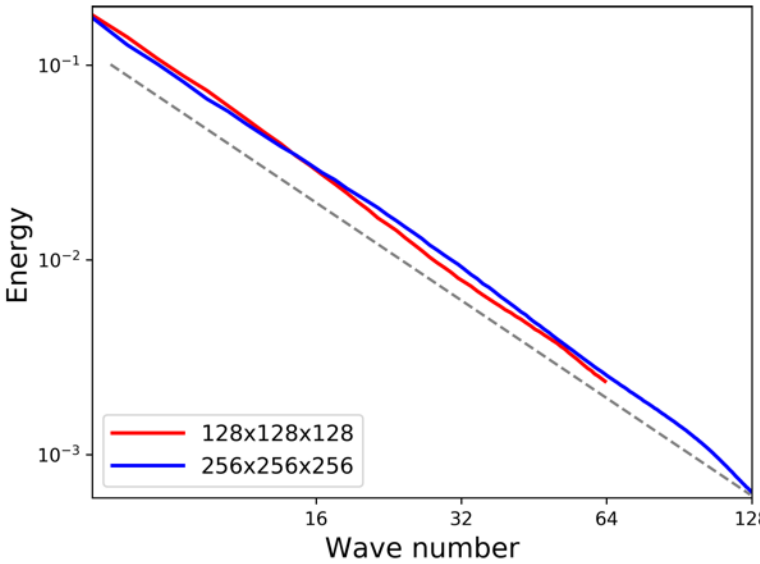
Local error estimation in turbulence simulations

At high Reynolds numbers, coarse-grained simulations of turbulence (“large eddy simulations”, or LES) are never in the asymptotic range of convergence – by definition!

Implies that:

- Refining the grid affects both the numerical and the modeling errors
- Traditional error estimation techniques (e.g., to evaluate the local residual on a refined mesh) fail to account for the solution developing finer scales on such a mesh

Energy spectrum in LES of isotropic turbulence



Local error estimation in turbulence simulations

Approaches:

1. Traditional residual-based: interpolate solution onto refined grid, evaluate the residual. But does this estimate what would *actually* happen on a refined grid, where smaller scales would develop?

   Grid-refinement of LES with an eddy-viscosity model:

		Actual LES	Frozen u_i
$\overline{u_i' u_j'}$		increases	unchanged
$\tau_{ij}^{\text{model}} \sim C\Delta^2 \left\lvert \frac{\partial u_i}{\partial x_j} \right\rvert \frac{\partial u_i}{\partial x_j}$	Δ	decreases	decreases
	$\partial u_i / \partial x_j$	increases	unchanged

2. Estimation of the time-averaged error by comparing averages (and co-variances, etc) on different grids – natural consequence of the “averaged adjoint” approach to sensitivity.
   - Works in channel flow and backward-facing step flow, but what about more general types of flows?

3. “Coarse-graining consistency” error estimation

Software integration (and essentially the overall work plan...)

Exascale approach: Kokkos, Trilinos

Multi-solver software framework for integrated VVUQ on high-fidelity simulation of chaotic, turbulent flow

- Setup of simulation parameter sweeps
- Ensemble jobs, and run pipeline.

1) Interface with existing resource manager frameworks (e.g., Flux)
2) MPI-based embedded many-run execution within single job submission

Forward problem

DNS/LES flat-plate flow (primal) solver

- Cartesian, co-located structured grid
- Incompressible Navier-Stokes
- Finite difference (2nd – 6th order)
- Fractional step time integration
- DFT, pencil-based, null-space pressure Poisson solver

LES mode

UQ and sensitivity analysis

Chaos-theory-based approaches

- Space-time multigrid type solvers
- Non-intrusive solvers, utilizing Jacobian
- Phase-space based solvers

Adjoint of ROM approaches

- Adjoint of non-chaotic (RANS) averaged forward equations
- Eddy-viscosity inference from LES data
- Higher-order statistics based

Error and solution verification

Local error estimators

- Numerical residual
- Small-scale resolved energy
- Solution differences in everywhere-dissimilar grids

Grid convergence and adaptation

UMD
MIT
USC

DNS mode

Verification of methodology with DNS `truth'

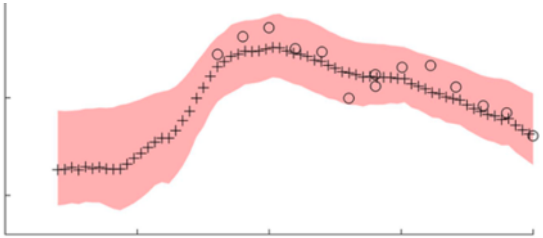
N_p forward DNS simulations for perturbed input parameters.

Finite-difference based sensitivity and error estimation

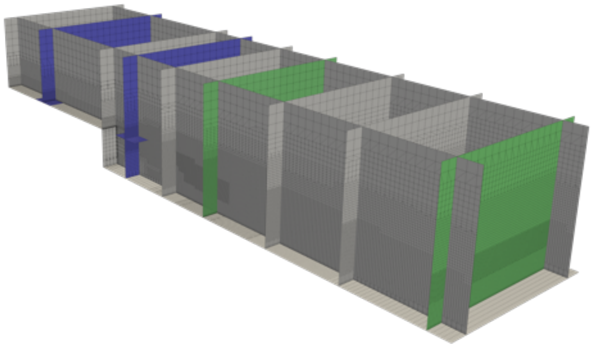
Comparison and assessment of VVUQ methodology

What constitutes success of this project?

- Algorithm(s)/theory for approximate sensitivity with feasible cost for incompressible turbulence



- Algorithm(s)/theory for local error estimation and grid-adaptation in LES



- A reference data set with highly controlled parametric variations for other research groups to test sensitivity algorithms