Computational error-analysis for large-eddy simulation

Bernard J. Geurts

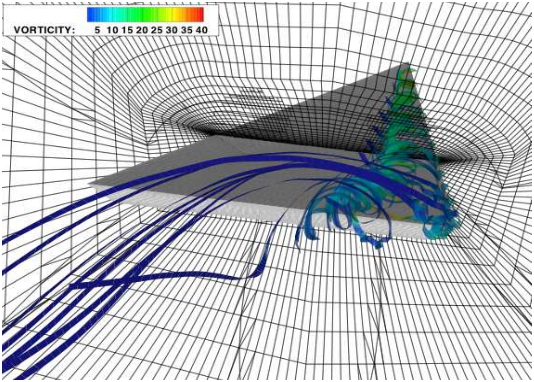
Multiscale Modeling and Simulation (Twente)
Anisotropic Turbulence (Eindhoven)

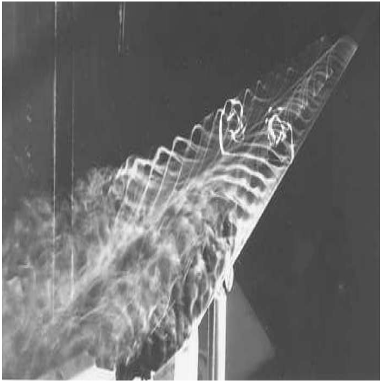
Saarbrücken, June 23-24, 2008

Everyday flows: Wake-Vortex Hazard

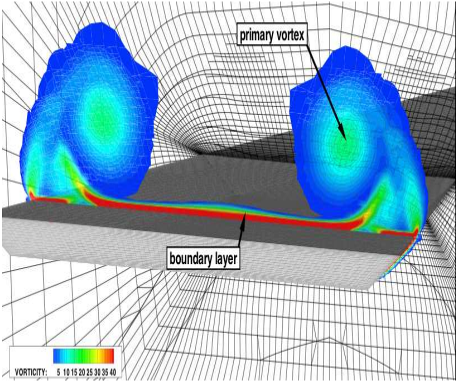
Airport throughput limitations: separation up to 10 km

Flow over delta wing



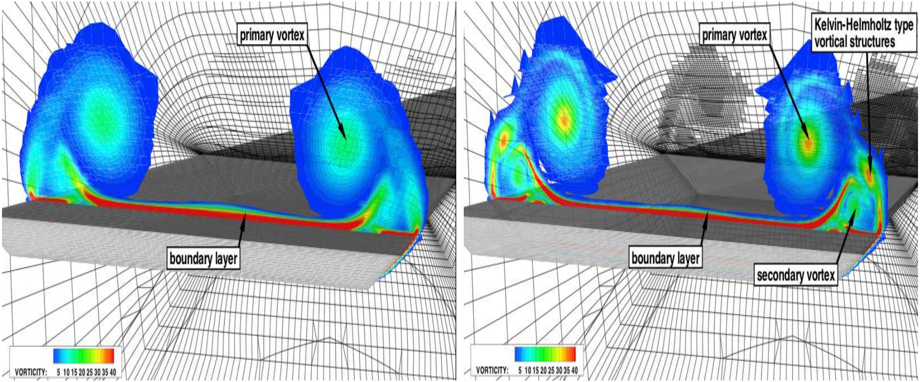


Grid dependence



Reliability - Error-bounds - Computational costs?

Grid dependence



Reliability - Error-bounds - Computational costs?

Outline

1. Role of numerics in LES
2. Pragmatic LES
3. Concluding remarks

Outline

Filtering Navier-Stokes equations

$$\partial_j u_j = 0 \quad ; \quad \partial_t u_i + \partial_j(u_i u_j) + \partial_i p - \frac{1}{Re}\partial_{jj} u_i = 0$$

Convolution-Filtering: *filter-kernel G*

$$\overline{u}_i = L(u_i) = \int G(x-\xi)u(\xi)\, d\xi \quad ; \quad L(1) = 1$$

Large-eddy equations:

$$\partial_j \overline{u}_j = 0$$

$$\partial_t \overline{u}_i + \partial_j(\overline{u}_i \overline{u}_j) + \partial_i \overline{p} - \frac{1}{Re}\partial_{jj}\overline{u}_i = -\partial_j(\overline{u_i u_j} - \overline{u}_i \overline{u}_j)$$

Sub-filter stress tensor

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j$$

Filtering Navier-Stokes equations

$$\partial_j u_j = 0 \quad ; \quad \partial_t u_i + \partial_j (u_i u_j) + \partial_i p - \frac{1}{Re} \partial_{jj} u_i = 0$$

Convolution-Filtering: filter-kernel G

$$\overline{u}_i = L(u_i) = \int G(x - \xi) u(\xi)\, d\xi \quad ; \quad L(1) = 1$$

Large-eddy equations:

$$\partial_j \overline{u}_j = 0$$

$$\partial_t \overline{u}_i + \partial_j (\overline{u}_i \overline{u}_j) + \partial_i \overline{p} - \frac{1}{Re} \partial_{jj} \overline{u}_i = -\partial_j (\overline{u_i u_j} - \overline{u}_i \overline{u}_j)$$

Sub-filter stress tensor

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j$$

Spatial filtering, closure problem

Shorthand notation:

$$NS(\mathbf{u}) = 0 \quad \Rightarrow \quad NS(\overline{\mathbf{u}}) = -\nabla \cdot \tau(\mathbf{u}, \overline{\mathbf{u}}) \; \Leftarrow \; -\nabla \cdot M(\overline{\mathbf{u}})$$

Basic LES formulation

$$\textit{Find } \mathbf{v} : \quad NS(\mathbf{v}) = -\nabla \cdot M(\mathbf{v})$$

After closure system of PDE's results:

- dynamic range restricted primarily to scales $> \Delta$
- does solution $\mathbf{v}$ of closed system resemble $\overline{\mathbf{u}}$?

Goal in LES: *determine the unique solution to system of PDE's that results after adopting explicit closure model*

Spatial filtering, closure problem

Shorthand notation:

$$NS(\mathbf{u}) = 0 \quad \Rightarrow \quad NS(\overline{\mathbf{u}}) = -\nabla \cdot \tau(\mathbf{u}, \overline{\mathbf{u}}) \Leftarrow -\nabla \cdot M(\overline{\mathbf{u}})$$

Basic LES formulation

$$\textit{Find } \mathbf{v} : \quad NS(\mathbf{v}) = -\nabla \cdot M(\mathbf{v})$$

After closure system of PDE's results:

- dynamic range restricted primarily to scales $> \Delta$
- does solution $\mathbf{v}$ of closed system resemble $\overline{\mathbf{u}}$?

Goal in LES: *determine the unique solution to system of PDE's that results after adopting explicit closure model*

Spatial filtering, closure problem

Shorthand notation:

$$NS(\mathbf{u}) = 0 \quad \Rightarrow \quad NS(\overline{\mathbf{u}}) = -\nabla \cdot \tau(\mathbf{u}, \overline{\mathbf{u}}) \;\Leftarrow\; -\nabla \cdot M(\overline{\mathbf{u}})$$

Basic LES formulation

$$\textit{Find } \mathbf{v}: \quad NS(\mathbf{v}) = -\nabla \cdot M(\mathbf{v})$$

After closure system of PDE's results:

- dynamic range restricted primarily to scales $> \Delta$
- does solution $\mathbf{v}$ of closed system resemble $\overline{\mathbf{u}}$?

Goal in LES: *determine the unique solution to system of PDE's that results after adopting explicit closure model*

Spatial filtering, closure problem

Shorthand notation:

$$NS(\mathbf{u}) = 0 \quad \Rightarrow \quad NS(\overline{\mathbf{u}}) = -\nabla \cdot \tau(\mathbf{u}, \overline{\mathbf{u}}) \;\Leftarrow\; -\nabla \cdot M(\overline{\mathbf{u}})$$

Basic LES formulation

$$\textit{Find } \mathbf{v} : \quad NS(\mathbf{v}) = -\nabla \cdot M(\mathbf{v})$$

After closure system of PDE's results:

- dynamic range restricted primarily to scales $> \Delta$
- does solution $\mathbf{v}$ of closed system resemble $\overline{\mathbf{u}}$?

Goal in LES: *determine the unique solution to system of PDE's that results after adopting explicit closure model*

Spatial filtering, closure problem

Shorthand notation:

$$NS(\mathbf{u}) = 0 \quad \Rightarrow \quad NS(\overline{\mathbf{u}}) = -\nabla \cdot \tau(\mathbf{u}, \overline{\mathbf{u}}) \Leftarrow -\nabla \cdot M(\overline{\mathbf{u}})$$

Basic LES formulation

$$\textit{Find } \mathbf{v} : \quad NS(\mathbf{v}) = -\nabla \cdot M(\mathbf{v})$$

After closure system of PDE's results:

- dynamic range restricted primarily to scales $> \Delta$
- does solution $\mathbf{v}$ of closed system resemble $\overline{\mathbf{u}}$?

Goal in LES: *determine the unique solution to system of PDE's that results after adopting explicit closure model*

Spatial filtering, closure problem

Shorthand notation:

$$NS(\mathbf{u}) = 0 \quad \Rightarrow \quad NS(\overline{\mathbf{u}}) = -\nabla \cdot \tau(\mathbf{u}, \overline{\mathbf{u}}) \ \Leftarrow \ -\nabla \cdot M(\overline{\mathbf{u}})$$

Basic LES formulation

$$\mathit{Find}\ \mathbf{v}: \quad NS(\mathbf{v}) = -\nabla \cdot M(\mathbf{v})$$

After closure system of PDE's results:

- dynamic range restricted primarily to scales $> \Delta$
- does solution $\mathbf{v}$ of closed system resemble $\overline{\mathbf{u}}$?

Goal in LES: *determine the unique solution to system of PDE's that results after adopting explicit closure model*

Numerics in academic LES setting

Goal: *approximate the unique solution to system of PDE's resulting after adopting explicit closure model*

General (textbook) requirements:

- Filter separates scales $> \Delta$ from scales $< \Delta$
- Computational grid provides additional length-scale h
- Require Δ/h to be sufficiently large ($\Delta/h \to \infty$)
- Good numerics: $v(x,t:\Delta,h) \to v(x,t:\Delta,0)$ rapidly

However:

- computational costs $\sim N^4$: implies modest Δ/h
- potentially large role of numerical method in computational dynamics because of marginal resolution

Numerics in academic LES setting

Goal: *approximate the unique solution to system of PDE's resulting after adopting explicit closure model*

General (textbook) requirements:

- Filter separates scales $> \Delta$ from scales $< \Delta$
- Computational grid provides additional length-scale h
- Require Δ/h to be sufficiently large ($\Delta/h \to \infty$)
- Good numerics: $v(x,t:\Delta,h) \to v(x,t:\Delta,0)$ rapidly

However:

- computational costs $\sim N^4$: implies modest Δ/h
- potentially large role of numerical method in computational dynamics because of marginal resolution

Numerics in academic LES setting

Goal: *approximate the unique solution to system of PDE's resulting after adopting explicit closure model*

General (textbook) requirements:

- Filter separates scales $> \Delta$ from scales $< \Delta$
- Computational grid provides additional length-scale h
- Require Δ/h to be sufficiently large $(\Delta/h \to \infty)$
- Good numerics: $v(x,t:\Delta,h) \to v(x,t:\Delta,0)$ rapidly

However:

- computational costs $\sim N^4$: implies modest Δ/h
- potentially large role of numerical method in computational dynamics because of marginal resolution

Numerics in academic LES setting

Goal: *approximate the unique solution to system of PDE's resulting after adopting explicit closure model*

General (textbook) requirements:

- Filter separates scales $> \Delta$ from scales $< \Delta$
- Computational grid provides additional length-scale h
- Require Δ/h to be sufficiently large ($\Delta/h \to \infty$)
- Good numerics: $v(x, t : \Delta, h) \to v(x, t : \Delta, 0)$ rapidly

However:

- computational costs $\sim N^4$: implies modest Δ/h
- potentially large role of numerical method in computational dynamics because of marginal resolution

Numerics in academic LES setting

Goal: *approximate the unique solution to system of PDE's resulting after adopting explicit closure model*

General (textbook) requirements:

- Filter separates scales $> \Delta$ from scales $< \Delta$
- Computational grid provides additional length-scale h
- Require Δ/h to be sufficiently large ($\Delta/h \to \infty$)
- Good numerics: $v(x, t : \Delta, h) \to v(x, t : \Delta, 0)$ rapidly

However:

- computational costs $\sim N^4$: implies modest Δ/h
- potentially large role of numerical method in computational dynamics because of marginal resolution

Numerics in academic LES setting

Goal: *approximate the unique solution to system of PDE's resulting after adopting explicit closure model*

General (textbook) requirements:

- Filter separates scales $> \Delta$ from scales $< \Delta$
- Computational grid provides additional length-scale h
- Require Δ/h to be sufficiently large ($\Delta/h \to \infty$)
- Good numerics: $v(x,t:\Delta,h) \to v(x,t:\Delta,0)$ rapidly

However:

- computational costs $\sim N^4$: implies modest Δ/h
- potentially large role of numerical method in computational dynamics because of marginal resolution

Discretization induces spatial filter

Consider central discretization:

$$
\begin{aligned}
\delta_x(u)_j &= \frac{1}{2h}\Big(u_{j+1} - u_{j-1}\Big) \\
&= \frac{1}{2h}\int_{x_j-h}^{x_j+h} \partial_x u(\xi)\, d\xi \\
&= \frac{\partial}{\partial x}\Big(\int_{x_j-h}^{x_j+h} \frac{u(\xi)}{2h}\Big) \\
&= \partial_x\Big(L_{2h}(u)\Big) = \partial_x(\widehat{u})
\end{aligned}
$$

Coarse grids with $h \approx \Delta$ imply both implicit and explicit filter

Discretization induces spatial filter

Consider central discretization:

$$
\begin{aligned}
\delta_x(u)_j &= \frac{1}{2h}\left(u_{j+1} - u_{j-1}\right) \\
&= \frac{1}{2h}\int_{x_j-h}^{x_j+h} \partial_x u(\xi)\, d\xi \\
&= \frac{\partial}{\partial x}\left(\int_{x_j-h}^{x_j+h} \frac{u(\xi)}{2h}\right) \\
&= \partial_x\left(L_{2h}(u)\right) = \partial_x(\widehat{u})
\end{aligned}
$$

Coarse grids with $h \approx \Delta$ imply both implicit and explicit filter

Discretization induces spatial filter

Consider central discretization:

$$
\begin{aligned}
\delta_x(u)_j &= \frac{1}{2h}\Big(u_{j+1} - u_{j-1}\Big) \\
&= \frac{1}{2h}\int_{x_j-h}^{x_j+h} \partial_x u(\xi)\, d\xi \\
&= \frac{\partial}{\partial x}\Big(\int_{x_j-h}^{x_j+h} \frac{u(\xi)}{2h}\Big) \\
&= \partial_x\Big(L_{2h}(u)\Big) = \partial_x(\widehat{u})
\end{aligned}
$$

Coarse grids with $h \approx \Delta$ imply both implicit and explicit filter

Discretization induces spatial filter

Consider central discretization:

$$\begin{aligned}
\delta_x(u)_j &= \frac{1}{2h}\Big(u_{j+1} - u_{j-1}\Big) \\
&= \frac{1}{2h}\int_{x_j-h}^{x_j+h} \partial_x u(\xi)\, d\xi \\
&= \frac{\partial}{\partial x}\Big(\int_{x_j-h}^{x_j+h} \frac{u(\xi)}{2h}\Big) \\
&= \partial_x\Big(L_{2h}(u)\Big) = \partial_x(\widehat{u})
\end{aligned}$$

Coarse grids with $h \approx \Delta$ imply both implicit and explicit filter

Discretization induces spatial filter

Consider central discretization:

$$\begin{aligned}\delta_x(u)_j &= \frac{1}{2h}\Big(u_{j+1} - u_{j-1}\Big) \\ &= \frac{1}{2h}\int_{x_j-h}^{x_j+h} \partial_x u(\xi)\, d\xi \\ &= \frac{\partial}{\partial x}\Big(\int_{x_j-h}^{x_j+h} \frac{u(\xi)}{2h}\Big) \\ &= \partial_x\Big(L_{2h}(u)\Big) = \partial_x(\widehat{u})\end{aligned}$$

Coarse grids with $h \approx \Delta$ imply both implicit and explicit filter

Modified closure problem

Discretization induces spatial filter: $\delta_x u = \partial_x(\widehat{u})$

Convective contribution:

$$
\begin{aligned}
\partial_x(\overline{u^2}) &= \delta_x(\overline{u}^2) + \left[\partial_x(\overline{u^2}) - \delta_x(\overline{u}^2)\right] \\
&= \delta_x(\overline{u}^2) + \partial_x(\overline{u^2} - \widehat{\overline{u}^2}) = \partial_x(\overline{u}^2) + \partial_x(\xi)
\end{aligned}
$$

- Modified mean flux
- Computational Turbulent Stress Tensor

$$
\xi = \overline{u^2} - \widehat{\overline{u}^2} = (\overline{u^2} - \overline{u}^2) + (\overline{u}^2 - \widehat{\overline{u}^2}) = \tau + \mathcal{H}(\overline{u}^2)
$$

Numerically induced high-pass filter:

$$
\mathcal{H}(f) = f - \widehat{f} \to 0 \text{ as } r = \Delta/h \gg 1
$$

Modified closure problem

Discretization induces spatial filter: $\delta_x u = \partial_x(\widehat{u})$
Convective contribution:

$$\partial_x(\overline{u^2}) = \delta_x(\overline{u}^2) + \left[\partial_x(\overline{u^2}) - \delta_x(\overline{u}^2)\right]$$

$$= \delta_x(\overline{u}^2) + \partial_x(\overline{u^2} - \widehat{\overline{u}^2}) = \partial_x(\widehat{\overline{u}^2}) + \partial_x(\xi)$$

- Modified mean flux
- Computational Turbulent Stress Tensor

$$\xi = \overline{u^2} - \widehat{\overline{u}^2} = (\overline{u^2} - \overline{u}^2) + (\overline{u}^2 - \widehat{\overline{u}^2}) = \tau + \mathcal{H}(\overline{u}^2)$$

Numerically induced high-pass filter:

$$\mathcal{H}(f) = f - \widehat{f} \to 0 \text{ as } r = \Delta/h \gg 1$$

Modified closure problem

Discretization induces spatial filter: $\delta_x u = \partial_x(\widehat{u})$
Convective contribution:

$$\begin{aligned} \partial_x(\overline{u^2}) &= \delta_x(\overline{u}^2) + \left[\partial_x(\overline{u^2}) - \delta_x(\overline{u}^2)\right] \\ &= \delta_x(\overline{u}^2) + \partial_x(\overline{u^2} - \widehat{\overline{u}^2}) = \partial_x(\widehat{\overline{u}^2}) + \partial_x(\xi) \end{aligned}$$

- Modified mean flux
- Computational Turbulent Stress Tensor

$$\xi = \overline{u^2} - \widehat{\overline{u}^2} = (\overline{u^2} - \overline{u}^2) + (\overline{u}^2 - \widehat{\overline{u}^2}) = \tau + \mathcal{H}(\overline{u}^2)$$

Numerically induced high-pass filter:

$$\mathcal{H}(f) = f - \widehat{f} \to 0 \text{ as } r = \Delta/h \gg 1$$

Modified closure problem

Discretization induces spatial filter: $\delta_x u = \partial_x(\widehat{u})$
Convective contribution:

$$\begin{aligned}\partial_x(\overline{u^2}) &= \delta_x(\overline{u}^2) + \left[\partial_x(\overline{u^2}) - \delta_x(\overline{u}^2)\right] \\ &= \delta_x(\overline{u}^2) + \partial_x(\overline{u^2} - \widehat{\overline{u}^2}) = \partial_x(\widehat{\overline{u}^2}) + \partial_x(\xi)\end{aligned}$$

- Modified mean flux
- Computational Turbulent Stress Tensor

$$\xi = \overline{u^2} - \widehat{\overline{u}^2} = (\overline{u^2} - \overline{u}^2) + (\overline{u}^2 - \widehat{\overline{u}^2}) = \tau + \mathcal{H}(\overline{u}^2)$$

Numerically induced high-pass filter:

$$\mathcal{H}(f) = f - \widehat{f} \rightarrow 0 \text{ as } r = \Delta/h \gg 1$$

Dynamic importance - subgrid resolution

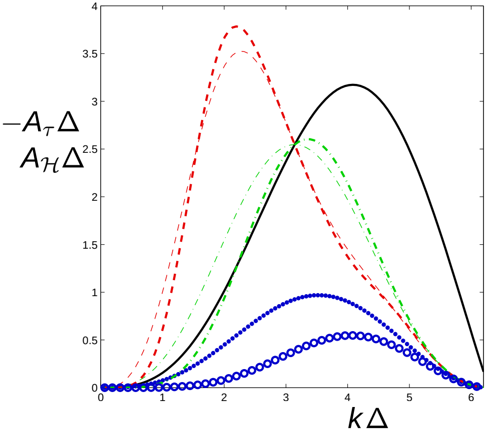
Contributions associated with $u = e^{\imath kx}$:

$$\tau = A_\tau(k\Delta)e^{2\imath kx} \;\; ; \;\; \mathcal{H}(\overline{u}^2) = A_{\mathcal{H}}(k\Delta, r)e^{2\imath kx}$$

Dynamic importance - subgrid resolution

Contributions associated with $u = e^{\imath kx}$:

$$\tau = A_\tau(k\Delta)e^{2\imath kx} \;\; ; \;\; \mathcal{H}(\overline{u}^2) = A_\mathcal{H}(k\Delta, r)e^{2\imath kx}$$

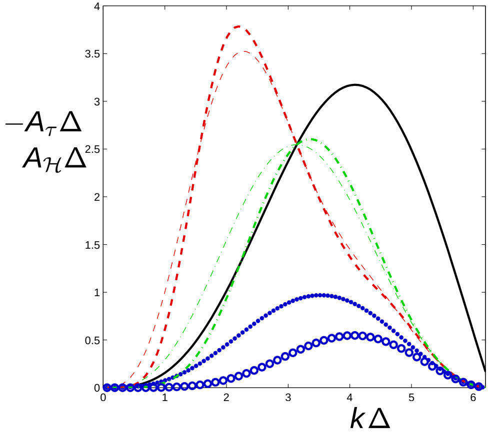


$A_\tau\Delta$: solid

Dynamic importance - subgrid resolution

Contributions associated with $u = e^{\imath kx}$:

$$\tau = A_\tau(k\Delta)e^{2\imath kx} \;\; ; \;\; \mathcal{H}(\overline{u}^2) = A_\mathcal{H}(k\Delta, r)e^{2\imath kx}$$

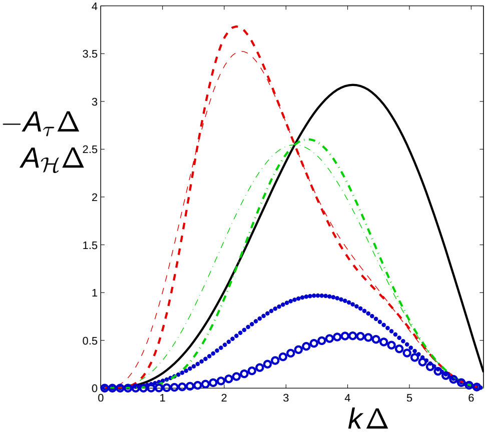


$A_\tau\Delta$: solid
$A_\mathcal{H}\Delta$ at $r = 1$ (- -)
$r = 2$ (-.-) and $r = 4$ (○)

Dynamic importance - subgrid resolution

Contributions associated with $u = e^{\imath kx}$:

$$\tau = A_\tau(k\Delta)e^{2\imath kx} \quad ; \quad \mathcal{H}(\overline{u}^2) = A_\mathcal{H}(k\Delta, r)e^{2\imath kx}$$



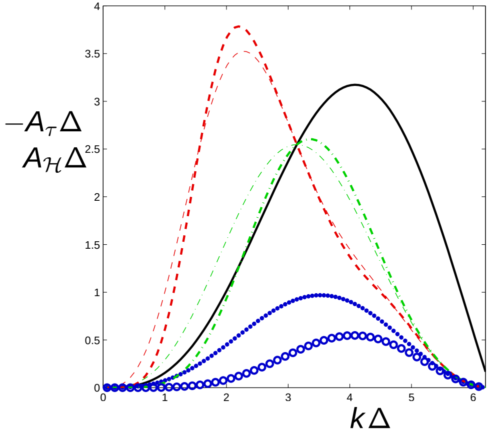
$A_\tau\Delta$: solid
$A_\mathcal{H}\Delta$ at $r = 1$ (- -)
$r = 2$ (-.-) and $r = 4$ (○)

2nd order (thin)
4th order (thick)

Dynamic importance - subgrid resolution

Contributions associated with $u = e^{\imath kx}$:

$$\tau = A_\tau(k\Delta)e^{2\imath kx} \;\; ; \;\; \mathcal{H}(\overline{u}^2) = A_{\mathcal{H}}(k\Delta, r)e^{2\imath kx}$$



$A_\tau\Delta$: solid
$A_{\mathcal{H}}\Delta$ at $r = 1$ (- -)
$r = 2$ (-.-) and $r = 4$ (∘)

2nd order (thin)
4th order (thick)

Strong effect $r = 1 - 2$;
reduction as $r \geq 4$

LES treatment of convective term

Discretization and modeling introduce errors:

$$
\begin{aligned}
\partial_j(\overline{u_i u_j}) &= \left[\delta_j(\overline{u}_i\overline{u}_j) + \mathcal{D}_i\right] + \partial_j \tau_{ij} \\
&= \left[\delta_j(\overline{u}_i\overline{u}_j) + \mathcal{D}_i\right] + \left\{\partial_j m_{ij} + \mathcal{R}_i\right\} \\
&= \delta_j(\overline{u}_i\overline{u}_j) + \delta_j m_{ij} + \left(\mathcal{D}_i + \mathcal{R}_i + \mathcal{D}_i^{(m)}\right)
\end{aligned}
$$

Distinguish:

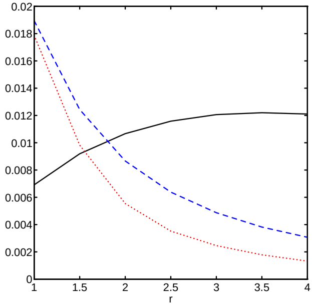
- $\mathcal{D}_i$: discretization error from using method δ_j
- $\mathcal{R}_i = \partial_j(\tau_{ij} - m_{ij})$: total 'model-residue'
- $\mathcal{D}_i^{(m)}$: error when treating model m_{ij}, e.g., filtering

Q1: Justified to ignore $\mathcal{D}_i$? Grid-(in)-dependent LES ?

Q2: Interacting errors? Error-decomposition? Dominance?

A priori test: snapshot turbulent mixing

Comparison discretization error and sub-filter flux



Sub-filter flux (solid): fixed grid, increasing Δ
Discretization error: 2nd order (- -), 4th order (dotted)

At $r = 1$ discretization dominant – relevance modeling?

Total error decomposition

Total (ε_t)=Discretization (ε_d) $+$ Modeling (ε_m)

Decomposition requires: DNS and LES at various $r = \Delta/h$

- Reference via (filtered) DNS data
- LES without discretization errors: fixed Δ, $r \to \infty$
- LES with both types of errors

Provides a posteriori decomposition:

$$\varepsilon_d(E) = E_{\text{LES}}(\Delta, r) - E_{\text{LES}}(\Delta, \infty)$$
$$\varepsilon_m(E) = E_{\text{LES}}(\Delta, \infty) - E_{\overline{\text{DNS}}}(\Delta, r)$$
$$\varepsilon_t(E) = \varepsilon_d(E) + \varepsilon_m(E) = E_{\text{LES}}(\Delta, r) - E_{\overline{\text{DNS}}}(\Delta, r)$$

Requires number of LES: various Δ, h, models, schemes ...

Total error decomposition

Total (ε_t)=Discretization (ε_d) + Modeling (ε_m)

Decomposition requires: DNS and LES at various $r = \Delta/h$

- Reference via (filtered) DNS data
- LES without discretization errors: fixed Δ, $r \to \infty$
- LES with both types of errors

Provides a posteriori decomposition:

$$\varepsilon_d(E) = E_{\text{LES}}(\Delta, r) - E_{\text{LES}}(\Delta, \infty)$$
$$\varepsilon_m(E) = E_{\text{LES}}(\Delta, \infty) - E_{\overline{\text{DNS}}}(\Delta, r)$$
$$\varepsilon_t(E) = \varepsilon_d(E) + \varepsilon_m(E) = E_{\text{LES}}(\Delta, r) - E_{\overline{\text{DNS}}}(\Delta, r)$$

Requires number of LES: various Δ, h, models, schemes ...

Total error decomposition

Total (ε_t)=Discretization (ε_d) $+$ Modeling (ε_m)

Decomposition requires: DNS and LES at various $r = \Delta/h$

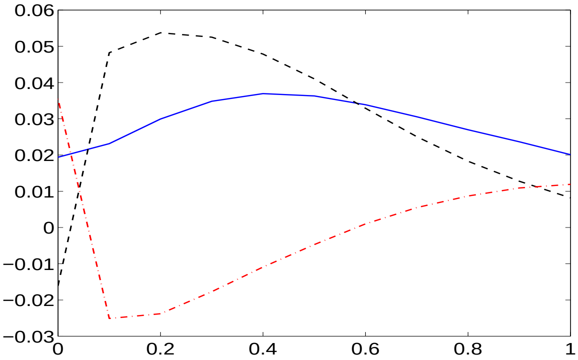
- Reference via (filtered) DNS data
- LES without discretization errors: fixed Δ, $r \to \infty$
- LES with both types of errors

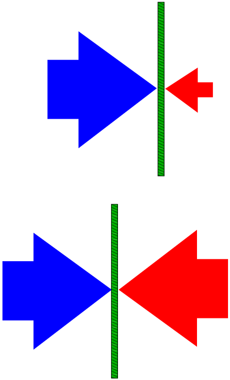
Provides a posteriori decomposition:

$$\varepsilon_d(E) = E_{\text{LES}}(\Delta, r) - E_{\text{LES}}(\Delta, \infty)$$
$$\varepsilon_m(E) = E_{\text{LES}}(\Delta, \infty) - E_{\overline{\text{DNS}}}(\Delta, r)$$
$$\varepsilon_t(E) = \varepsilon_d(E) + \varepsilon_m(E) = E_{\text{LES}}(\Delta, r) - E_{\overline{\text{DNS}}}(\Delta, r)$$

Requires number of LES: various Δ, h, models, schemes ...

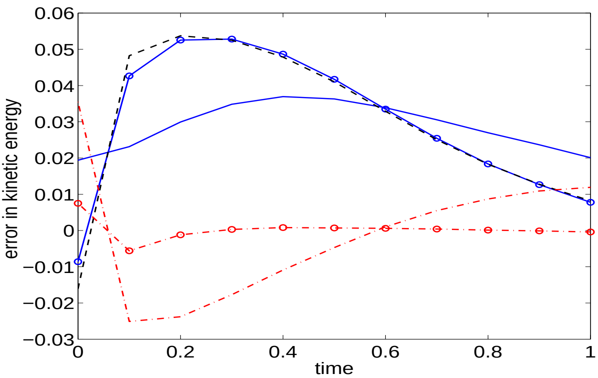
Counter-acting errors: LES-paradoxes

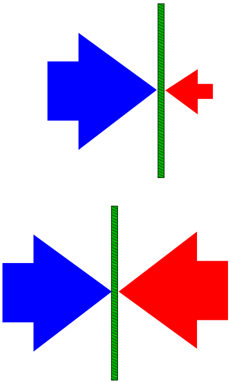




Decaying turbulence: discretization, modeling and total-error

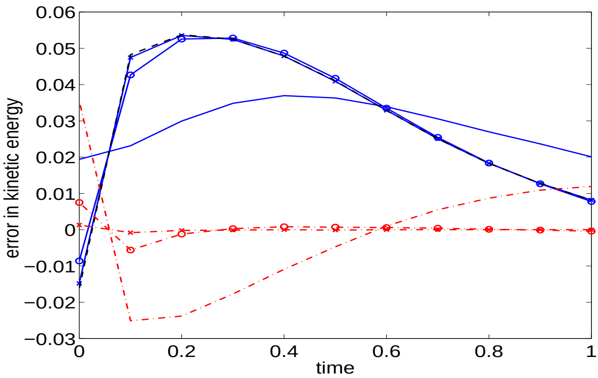
Counter-acting errors: LES-paradoxes

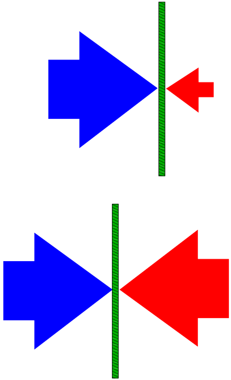




Decaying turbulence: discretization, modeling and total-error

Counter-acting errors: LES-paradoxes

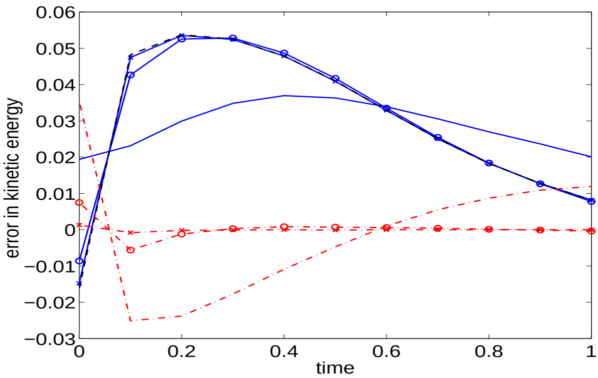


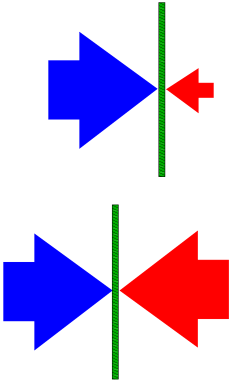


Decaying turbulence: discretization, modeling and total-error

- better model may result in worse predictions
- better numerics may result in worse predictions

Counter-acting errors: LES-paradoxes

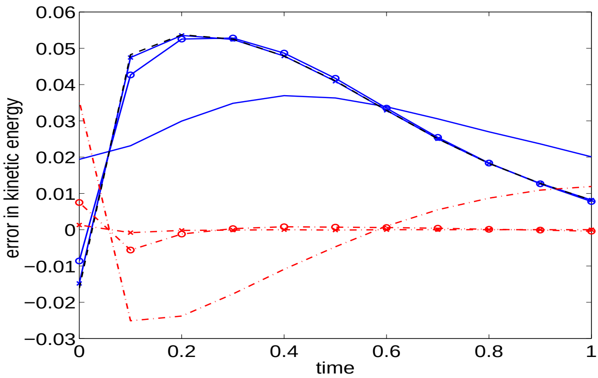


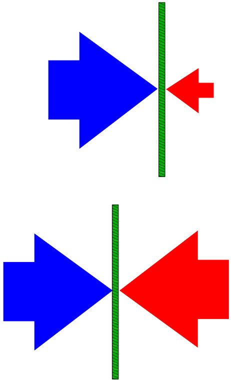


Decaying turbulence: discretization, modeling and total-error

- better model may result in worse predictions
- better numerics may result in worse predictions

Counter-acting errors: LES-paradoxes





Decaying turbulence: discretization, modeling and total-error

- better model may result in worse predictions
- better numerics may result in worse predictions

Numerics or modeling or both ?

Observe:

- At marginal resolution the numerics strongly modifies the equations that should be solved
- Likewise, the introduction of a subgrid model modifies these equations

Dilemma: which is to be preferred?

- Pragmatic guideline: minimal total error at given computational costs
- NOT: simply combine best numerics and best model

Numerics or modeling or both ?

Observe:

- At marginal resolution the numerics strongly modifies the equations that should be solved
- Likewise, the introduction of a subgrid model modifies these equations

Dilemma: which is to be preferred?

- Pragmatic guideline: minimal total error at given computational costs
- NOT: simply combine best numerics and best model

Numerics or modeling or both ?

Observe:

- At marginal resolution the numerics strongly modifies the equations that should be solved
- Likewise, the introduction of a subgrid model modifies these equations

Dilemma: which is to be preferred?

- Pragmatic guideline: minimal total error at given computational costs
- NOT: simply combine best numerics and best model

Numerics or modeling or both ?

Observe:

- At marginal resolution the numerics strongly modifies the equations that should be solved
- Likewise, the introduction of a subgrid model modifies these equations

Dilemma: which is to be preferred?

- Pragmatic guideline: minimal total error at given computational costs
- NOT: simply combine best numerics and best model

Numerics or modeling or both ?

Observe:

- At marginal resolution the numerics strongly modifies the equations that should be solved
- Likewise, the introduction of a subgrid model modifies these equations

Dilemma: which is to be preferred?

- Pragmatic guideline: minimal total error at given computational costs
- NOT: simply combine best numerics and best model

Outline

Experimental error-assessment

Pragmatic: minimal total error at given computational costs

Discuss:

- error-landscape/optimal refinement strategy
- optimality of MILES in DG-FEM

Experimental error-assessment

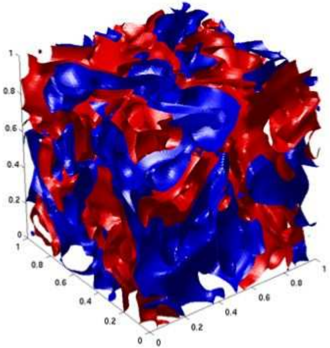
Pragmatic: minimal total error at given computational costs

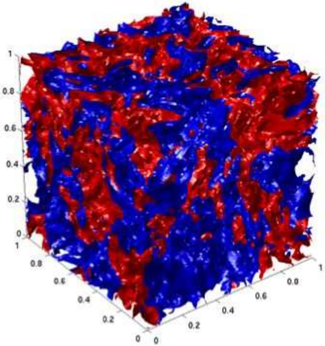
Discuss:

- error-landscape/optimal refinement strategy
- optimality of MILES in DG-FEM

Smagorinsky fluid

Homogeneous decaying turbulence at $Re_\lambda = 50,\ 100$





- Smagorinsky fluid — subgrid model:

$$m_{ij}^S = -2(C_S\overline{\Delta})^2|\overline{S}|\overline{S}_{ij} = -2\ell_S^2|\overline{S}|\overline{S}_{ij}$$

introduces Smagorinsky-length ℓ_S

TU/e

Accuracy measures

Monitor resolved kinetic energy

$$E = \frac{1}{|\Omega|}\int_{\Omega}\frac{1}{2}\overline{\mathbf{u}}\cdot\overline{\mathbf{u}}\,d\mathbf{x} = \frac{1}{2}\langle\overline{\mathbf{u}}\cdot\overline{\mathbf{u}}\rangle$$

Measure relative error: top-hat filter Δ, grid $h = \Delta/r$

$$\delta_E(\Delta, r) = \left\|\frac{E_{LES}(\Delta, r) - E_{\overline{DNS}}(\Delta, r)}{E_{\overline{DNS}}(\Delta, r)}\right\|$$

with error integrated over time

$$\|f\|^2 = \frac{1}{t_1 - t_0}\int_{t_0}^{t_1} f^2(t)dt$$

- each simulation represented by **single** number
- concise representation facilitates comparison

Accuracy measures

Monitor resolved kinetic energy

$$E = \frac{1}{|\Omega|}\int_{\Omega}\frac{1}{2}\overline{\mathbf{u}}\cdot\overline{\mathbf{u}}\,d\mathbf{x} = \frac{1}{2}\langle\overline{\mathbf{u}}\cdot\overline{\mathbf{u}}\rangle$$

Measure relative error: top-hat filter Δ, grid $h = \Delta/r$

$$\delta_E(\Delta, r) = \left\| \frac{E_{LES}(\Delta, r) - E_{\overline{DNS}}(\Delta, r)}{E_{\overline{DNS}}(\Delta, r)} \right\|$$

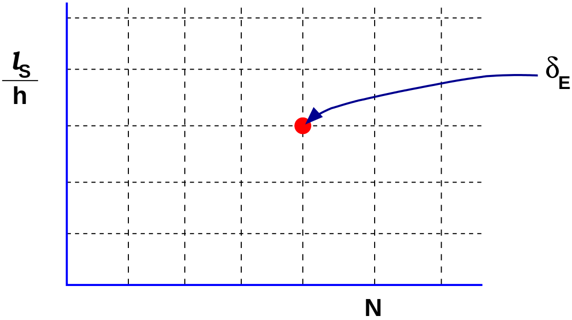
with error integrated over time

$$\|f\|^2 = \frac{1}{t_1 - t_0}\int_{t_0}^{t_1} f^2(t)dt$$

- each simulation represented by **single** number
- concise representation facilitates comparison

Error-landscape: Definition

Framework for collecting error information:



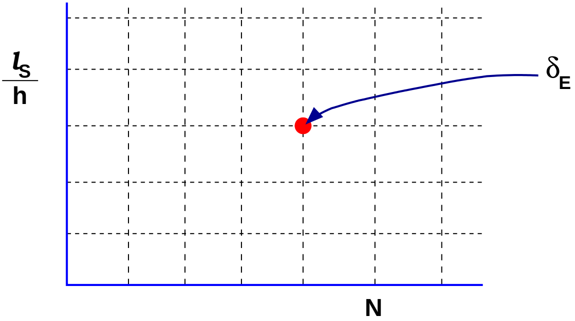
Each Smagorinsky LES corresponds to single point:

$$\left(N, \frac{\ell_S}{h}\right) \quad ; \quad error : \quad \delta_E$$

Contours of δ_E — fingerprint of LES

Error-landscape: Definition

Framework for collecting error information:

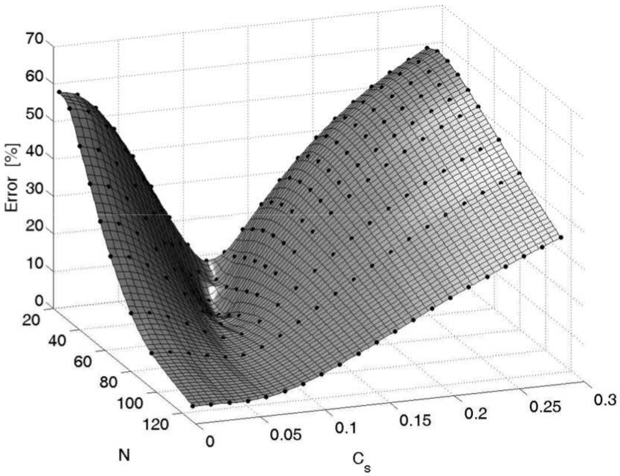


Each Smagorinsky LES corresponds to single point:

$$\left(N, \frac{\ell_S}{h}\right) \quad ; \quad error : \quad \delta_E$$

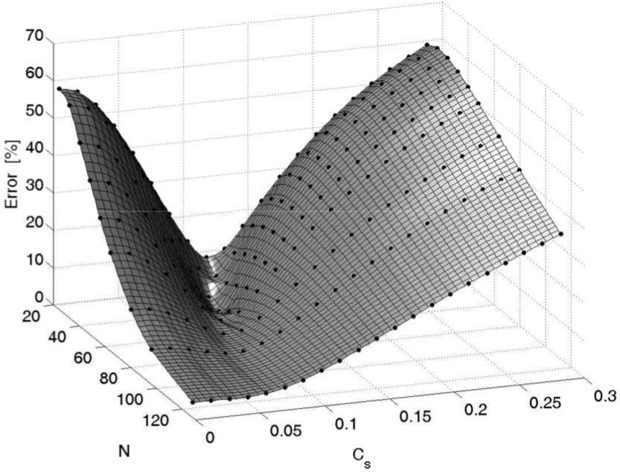
Contours of δ_E — fingerprint of LES

Total error-landscape



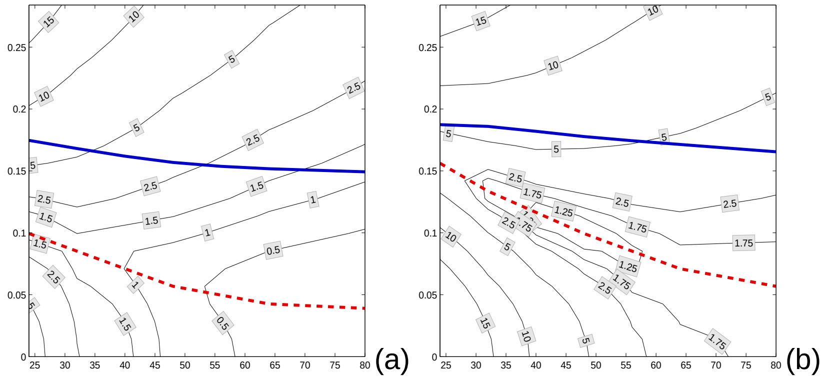
- combination of central discretization and Smagorinsky
- optimum at $C_S > 0$: SGS modeling is viable here

Total error-landscape



- combination of central discretization and Smagorinsky
- optimum at $C_S > 0$: SGS modeling is viable here

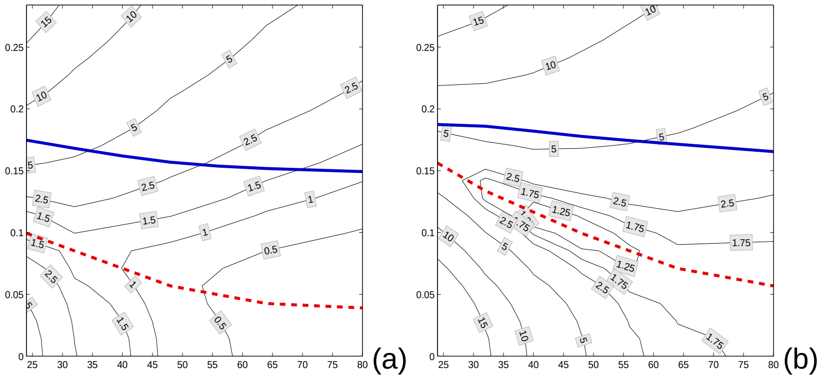
Error-landscape: optimal refinement



Optimal trajectory for $Re_\lambda = 50$ (a) and $Re_\lambda = 100$ (b)

- Under-resolution leads to strong error-increase
- Dynamic procedure over-estimates viscosity

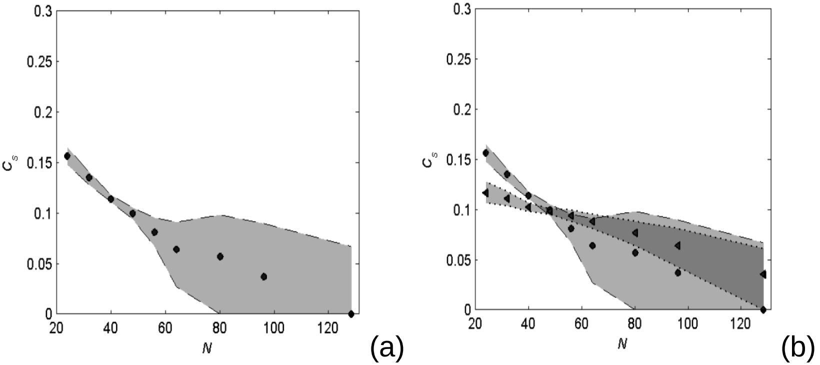
Error-landscape: optimal refinement



Optimal trajectory for $Re_\lambda = 50$ (a) and $Re_\lambda = 100$ (b)

- Under-resolution leads to strong error-increase
- Dynamic procedure over-estimates viscosity

Near optimal parameter regions



(a) energy, (b) energy and enstrophy. In overlap both accurate.

- connected, overlapping region $N \geq 48$
- weighing of errors leads to ‘multi-objective near optimum’

MILES philosophy

Observation:

- practical LES implies marginal resolution
- which implies large role of specific numerical discretization
- next to dynamics due to subgrid model
- and leads to strong interactions and complex error-accumulation

Proposal:

- obtain smoothing via appropriate numerical method alone
- accept that there is no grid-independent solution, other than DNS
- accept that predictions become discretization dependent

Is 'no-model/just numerics' option optimal/viable ?

Consider example: DG-FEM and homogeneous turbulence

MILES philosophy

Observation:

- practical LES implies marginal resolution
- which implies large role of specific numerical discretization
- next to dynamics due to subgrid model
- and leads to strong interactions and complex error-accumulation

Proposal:

- obtain smoothing via appropriate numerical method alone
- accept that there is no grid-independent solution, other than DNS
- accept that predictions become discretization dependent

Is ‘no-model/just numerics’ option optimal/viable ?

Consider example: DG-FEM and homogeneous turbulence

MILES philosophy

Observation:

- practical LES implies marginal resolution
- which implies large role of specific numerical discretization
- next to dynamics due to subgrid model
- and leads to strong interactions and complex error-accumulation

Proposal:

- obtain smoothing via appropriate numerical method alone
- accept that there is no grid-independent solution, other than DNS
- accept that predictions become discretization dependent

Is ‘no-model/just numerics’ option optimal/viable ?

Consider example: DG-FEM and homogeneous turbulence

MILES philosophy

Observation:

- practical LES implies marginal resolution
- which implies large role of specific numerical discretization
- next to dynamics due to subgrid model
- and leads to strong interactions and complex error-accumulation

Proposal:

- obtain smoothing via appropriate numerical method alone
- accept that there is no grid-independent solution, other than DNS
- accept that predictions become discretization dependent

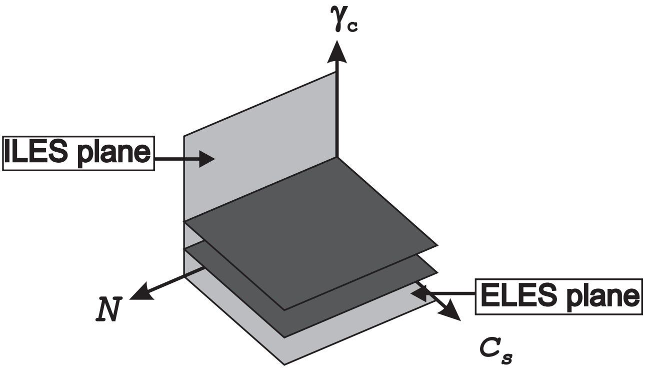
Is 'no-model/just numerics' option optimal/viable ?

Consider example: DG-FEM and homogeneous turbulence

DG-FEM of homogeneous turbulence

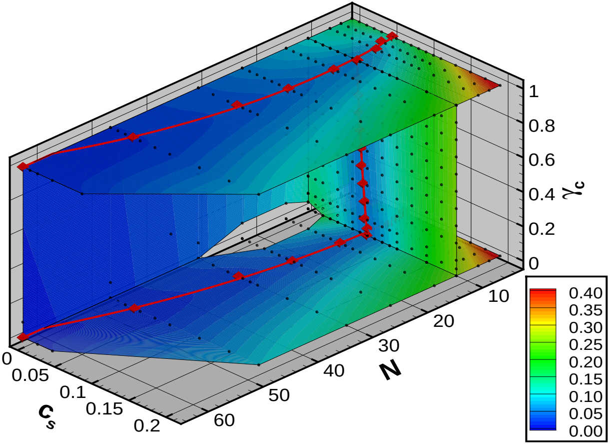
Discretization: Approximate Riemann solver

$$F = F_{central} + \gamma F_{dissipative} \quad ; \quad HLLC - flux$$



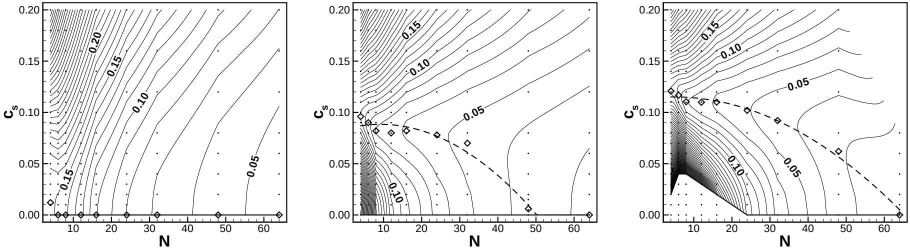
Three-dimensional accuracy charts

LES with DG-FEM: dissipative numerics



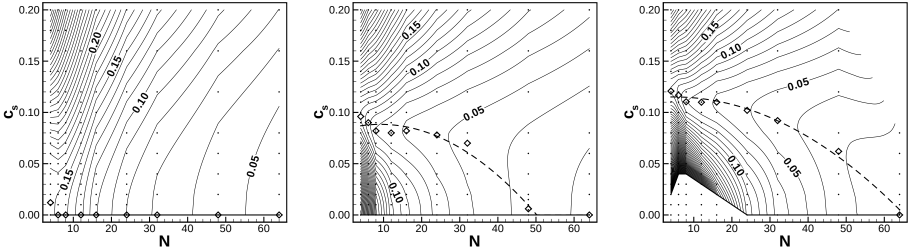
Third order DG-FEM at $Re_\lambda = 100$

Optimal refinement strategies: 2nd order



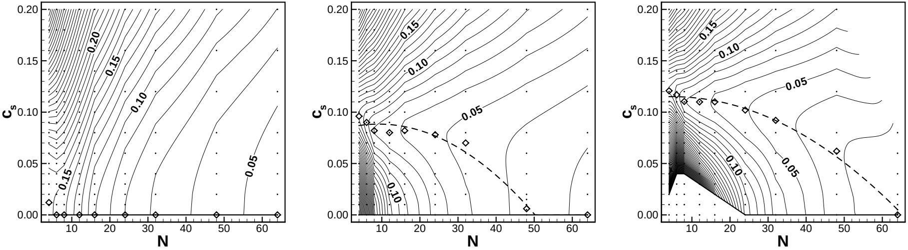
- left-to-right: $\gamma_c = 1$, $\gamma_c = 0.1$, $\gamma_c = 0.01$
- $C_S^* = 0$ at $\gamma_c = 1$: MILES best option
- $\gamma_c < 1$ implies $C_S^* \geq 0$: MILES sub-optimal
- decrease γ_c implies increase C_S^*: exchange of dissipation?

Optimal refinement strategies: 2nd order



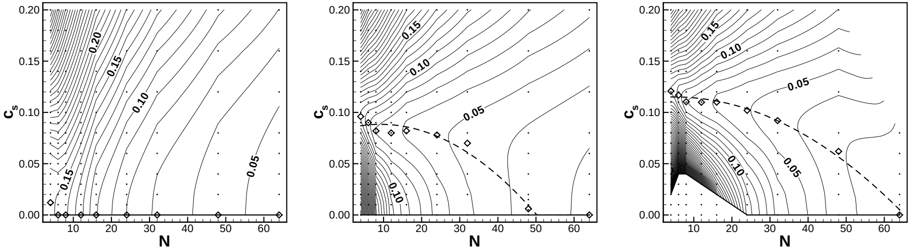
- left-to-right: $\gamma_c = 1$, $\gamma_c = 0.1$, $\gamma_c = 0.01$
- $C_S^* = 0$ at $\gamma_c = 1$: MILES best option
- $\gamma_c < 1$ implies $C_S^* \geq 0$: MILES sub-optimal
- decrease γ_c implies increase C_S^*: exchange of dissipation?

Optimal refinement strategies: 2nd order



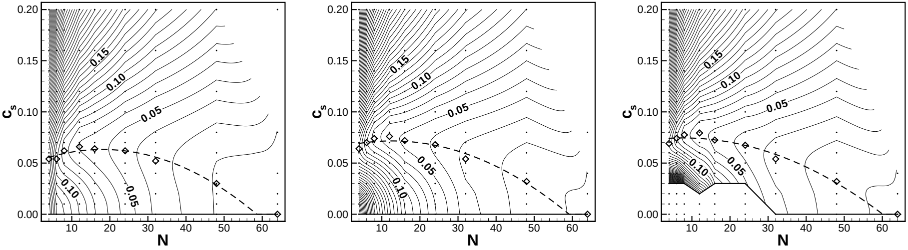
- left-to-right: $\gamma_c = 1$, $\gamma_c = 0.1$, $\gamma_c = 0.01$
- $C_S^* = 0$ at $\gamma_c = 1$: MILES best option
- $\gamma_c < 1$ implies $C_S^* \geq 0$: MILES sub-optimal
- decrease γ_c implies increase C_S^*: exchange of dissipation?

Optimal refinement strategies: 2nd order



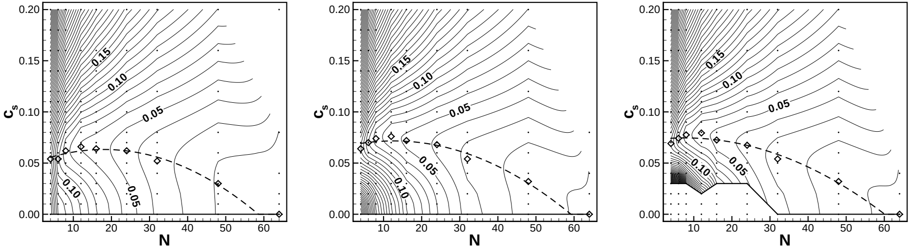
- left-to-right: $\gamma_c = 1$, $\gamma_c = 0.1$, $\gamma_c = 0.01$
- $C_S^* = 0$ at $\gamma_c = 1$: MILES best option
- $\gamma_c < 1$ implies $C_S^* \geq 0$: MILES sub-optimal
- decrease γ_c implies increase C_S^*: exchange of dissipation?

Optimal refinement strategies: 3rd order



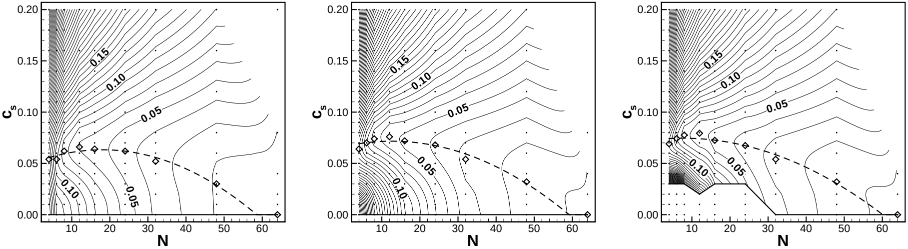
- left-to-right: $\gamma_c = 1$, $\gamma_c = 0.1$, $\gamma_c = 0.01$
- for all $\gamma_c \in [0, 1]$ find $C_S^* \geq 0$: MILES sub-optimal
- optimal C_S is less sensitive to γ_c value than 2nd order

Optimal refinement strategies: 3rd order



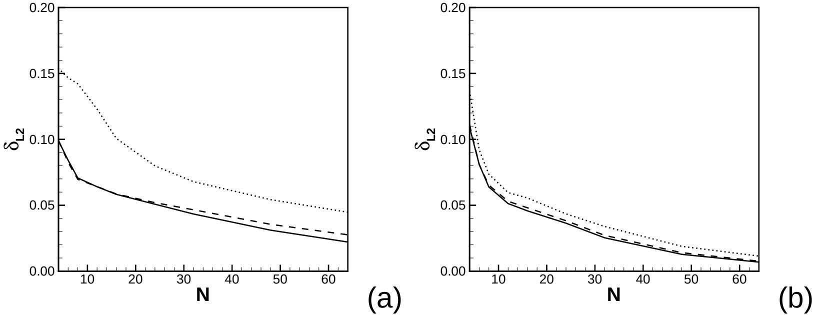
- left-to-right: $\gamma_c = 1$, $\gamma_c = 0.1$, $\gamma_c = 0.01$
- for all $\gamma_c \in [0, 1]$ find $C_S^* \geq 0$: MILES sub-optimal
- optimal C_S is less sensitive to γ_c value than 2nd order

Optimal refinement strategies: 3rd order



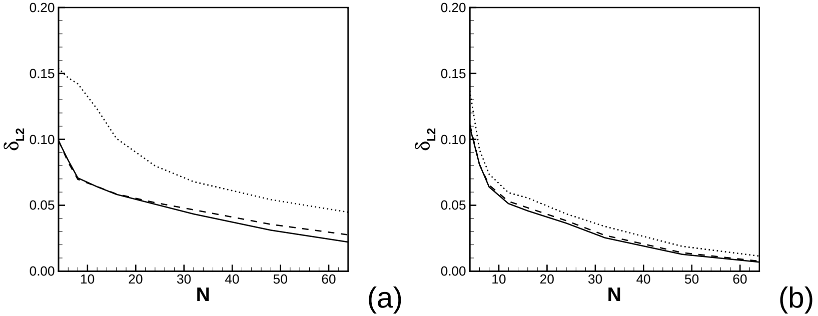
- left-to-right: $\gamma_c = 1$, $\gamma_c = 0.1$, $\gamma_c = 0.01$
- for all $\gamma_c \in [0, 1]$ find $C_S^* \geq 0$: MILES sub-optimal
- optimal C_S is less sensitive to γ_c value than 2nd order

Optimality of MILES ?



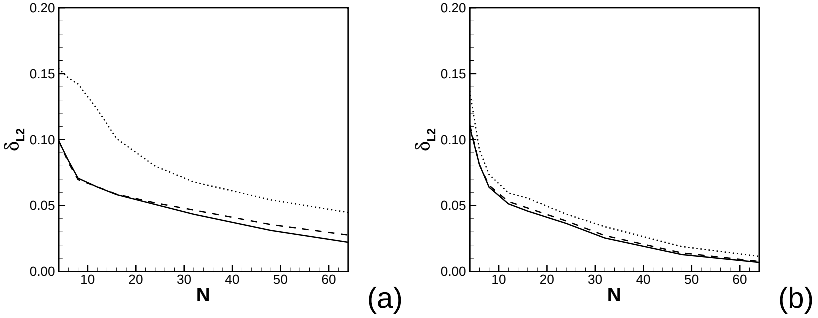
- (a): 2nd order ; (b): 3rd order
- $\gamma_c = 1.00$ (dot), $\gamma_c = 0.10$ (dash) and $\gamma_c = 0.01$ (–)
- 2nd: MILES-error larger than with explicit SGS model
- 3rd: optimum requires explicit SGS model

Optimality of MILES ?



- (a): 2nd order ; (b): 3rd order
- $\gamma_c = 1.00$ (dot), $\gamma_c = 0.10$ (dash) and $\gamma_c = 0.01$ (–)
- 2nd: MILES-error larger than with explicit SGS model
- 3rd: optimum requires explicit SGS model

Optimality of MILES ?



- (a): 2nd order ; (b): 3rd order
- $\gamma_c = 1.00$ (dot), $\gamma_c = 0.10$ (dash) and $\gamma_c = 0.01$ (–)
- 2nd: MILES-error larger than with explicit SGS model
- 3rd: optimum requires explicit SGS model

Outline

1. Role of numerics in LES
2. Pragmatic LES
3. **Concluding remarks**

Concluding remarks

- error-decomposition: modeling, discretization effects
- LES-paradoxes and interacting errors: better models/numerics may not lead to better predictions
- error-landscape – optimal refinement strategy
- MILES – sub-optimal: examples in which explicit modeling more efficient/accurate

- Error-interaction and *a priori* error-bounds hard to include: direct minimization to account for modeling and numerics

Concluding remarks

- error-decomposition: modeling, discretization effects
- LES-paradoxes and interacting errors: better models/numerics may not lead to better predictions
- error-landscape – optimal refinement strategy
- MILES – sub-optimal: examples in which explicit modeling more efficient/accurate

- Error-interaction and *a priori* error-bounds hard to include: direct minimization to account for modeling and numerics

Concluding remarks

- error-decomposition: modeling, discretization effects
- LES-paradoxes and interacting errors: better models/numerics may not lead to better predictions
- error-landscape – optimal refinement strategy
- MILES – sub-optimal: examples in which explicit modeling more efficient/accurate

- Error-interaction and *a priori* error-bounds hard to include: direct minimization to account for modeling and numerics

Concluding remarks

- error-decomposition: modeling, discretization effects
- LES-paradoxes and interacting errors: better models/numerics may not lead to better predictions
- error-landscape – optimal refinement strategy
- MILES – sub-optimal: examples in which explicit modeling more efficient/accurate

- Error-interaction and *a priori* error-bounds hard to include: direct minimization to account for modeling and numerics

Concluding remarks

- error-decomposition: modeling, discretization effects
- LES-paradoxes and interacting errors: better models/numerics may not lead to better predictions
- error-landscape – optimal refinement strategy
- MILES – sub-optimal: examples in which explicit modeling more efficient/accurate

- Error-interaction and *a priori* error-bounds hard to include: direct minimization to account for modeling and numerics

Concluding remarks

- error-decomposition: modeling, discretization effects
- LES-paradoxes and interacting errors: better models/numerics may not lead to better predictions
- error-landscape – optimal refinement strategy
- MILES – sub-optimal: examples in which explicit modeling more efficient/accurate

- Error-interaction and *a priori* error-bounds hard to include: direct minimization to account for modeling and numerics