PPNS.1 Pressure Projection Algorithms for RaNS

Pressure projection: RaNS methods enforcing *only* $\mathrm{D}M^h$

$\mathrm{D}M^h$: $\nabla \cdot \mathbf{u} = 0 \Rightarrow \left\| \nabla^h \cdot \mathbf{u}^h \right\| \leq \varepsilon > 0$ iteratively

"famous" named algorithms in the class include

MAC, SMAC	– Los Alamos Nat. Lab
SIMPLE,- ER, -EC, -EST	– Imperial College, UK
PISO	– Imperial College, UK
Operator splitting	– Univ. Houston
Continuity constraint	– Univ. Tennessee

fundamental PPNS theory ingredients

1. measure error in $\nabla^h \cdot \mathbf{u}^h$ via a potential function ϕ^h
2. employ ϕ^h to moderate $\mathrm{D}M^h$ and/or $\mathrm{D}\mathbf{P}^h$ error via
   - velocity correction
   - pressure correction
3. iterate $\mathrm{D}\mathbf{P}^h + \mathrm{D}M^h$ until $\left\| \nabla^h \cdot \mathbf{u}^h \right\| \leq \varepsilon$
4. determine genuine pressure field

PPNS.2 Pressure Projection INS PDE + BC System

For unsteady, laminar-thermal non-D INS in *n*-D

PDEs:

$$\mathrm{L}(u_i) = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} - \mathrm{Re}^{-1}\nabla^2 u_i + \mathrm{Eu}\frac{\partial p}{\partial x_i} + \frac{\mathrm{Gr}}{\mathrm{Re}^2}\Theta \hat{\mathbf{g}}_i = 0$$

$$\mathrm{L}(\Theta) = \frac{\partial \Theta}{\partial t} + u_j \frac{\partial \Theta}{\partial x_j} - \mathrm{Pe}^{-1}\nabla^2 \Theta - s = 0$$

$$\mathrm{L}(\phi) = -\nabla^2 \phi + \nabla \cdot \mathbf{u} = 0$$

$$\mathrm{L}(p) = -\mathrm{Eu}\nabla^2 p - \frac{\partial}{\partial x_i}\left(u_j \frac{\partial u_i}{\partial x_j} + \frac{\mathrm{Gr}}{\mathrm{Re}^2}\Theta \hat{\mathbf{g}}_i \right) = 0$$

BCs:

on $\partial\Omega_{\mathrm{inflow}}$: u_i, Θ, p on $\mathbf{x}_s$ usually given

$$\ell(\phi) = \hat{\mathbf{n}} \cdot \nabla\phi = 0$$

on $\partial\Omega_{\mathrm{outflow}}$: $\ell(q) = \hat{\mathbf{n}} \cdot \nabla(u_i, \Theta) = 0$

$$\phi = 0$$

$$\ell(p) = \hat{\mathbf{n}} \cdot \nabla p + f(\mathrm{Re}^{-1}\nabla^2 \mathbf{u} \cdot \hat{\mathbf{n}}, \partial u_i / \partial t) = 0$$

on $\partial\Omega_{\mathrm{walls}}$: $u_i = 0 = \hat{\mathbf{n}} \cdot \nabla\phi$

$$\ell(\Theta) = \hat{\mathbf{n}} \cdot \nabla\Theta + \mathrm{Nu}(\Theta - \Theta_r) = 0$$

$$\ell(p) = \hat{\mathbf{n}} \cdot \nabla p + f(\mathrm{Re}^{-1}\nabla^2 \mathbf{u} \cdot \hat{\mathbf{n}}) = 0$$

ref. Williams & Baker, IJNMF, Part B, V.29 (1996).

PPNS.3 GWS^h + θTS for Unsteady PP INS

The PPNS PDE + BCs system contains familiar expressions

$$\text{for } q(\mathbf{x},t) \approx q^N \equiv q^h = \cup_e(\{N_k(\eta,\zeta)\}^T\{Q(t)\}_e), \text{ and } \{Q(t)\} \Rightarrow \{\mathrm{U1, U2, U3, TEM}\}^T$$

$$\mathrm{GWS}^N + \theta\mathrm{TS} \Rightarrow \{\mathrm{F}Q\} = [\mathrm{MASS}]\{\Delta Q\} + \Delta t\{\mathrm{RES}(Q)\}\big|_\theta \equiv \{0\}$$

$$\text{then, } \mathrm{GWS}^N \Rightarrow \mathrm{GWS}^h \equiv \mathrm{S}_e\{\mathrm{WS}\}_e = \{0\} \text{ and}$$

$$\{\mathrm{F}Q\}_e = [\mathrm{M200}]_e\{Q\mathrm{P} - \mathrm{QN}\}_e + \Delta t\Big(\{\mathrm{U}J\}_e^T[\mathrm{M300}J]_e\{Q\}_e + \mathrm{Pa}^{-1}[\mathrm{M2}KK]_e\{Q\}_e - \{\mathrm{b}(Q,Q\mathrm{A})\}_e + \{\mathrm{BCs}\}_e\Big)_\theta$$

$$\text{for } q_\mathrm{A}(\mathbf{x},t) \approx q^N = q^h\ldots, \text{ and } \{Q\mathrm{A}\} = \{\mathrm{PHI, PRS}\}^T$$

$$\mathrm{GWS}^N \Rightarrow \mathrm{GWS}^h = \mathrm{S}_e\{\mathrm{WS}\}_e \equiv \{0\}, \text{ and}$$

$$\{\mathrm{F}Q\mathrm{A}\}_e = \mathrm{Pa}[\mathrm{M2}KK]_e\{Q\mathrm{A}\}_e - \{\mathrm{b}(Q)\}_e + \{\mathrm{BCs}\}_e$$

For matrix statement : $[\mathrm{JAC}]\{\delta Q\}^{p+1} = -\{\mathrm{F}Q\}^p$ and $[\mathrm{JAC}] \Rightarrow \mathrm{S}_e[\mathrm{JAC}]_e$

Newton:

$$[\mathrm{JAC}]_e = \begin{bmatrix} \mathrm{JUU} & \mathrm{JUV} & \mathrm{JUW} & \mathrm{JUT} & \mathrm{JU}\phi \\ \cdot & \mathrm{JVV} & \cdot & \cdot & \mathrm{JV}\phi \\ \cdot & \cdot & \mathrm{JWW} & \cdot & \mathrm{JW}\phi \\ \cdot & \cdot & \cdot & \mathrm{JTT} & 0 \\ \mathrm{J}\phi\mathrm{U} & \mathrm{J}\phi\mathrm{V} & \mathrm{J}\phi\mathrm{W} & 0 & \mathrm{J}\phi\phi \end{bmatrix}_e$$

PPNS.4 Iteration Strategy for PPNS GWS^h + θTS Algorithm

PPNS iteration strategy is independent of Newton choice

key formulation issues:

$$\mathrm{D}\mathbf{P}^h = f(\mathrm{D}M^h \text{ via } P^*_{n+1})$$

$$\mathrm{D}M^h = f(\nabla^h \cdot \mathbf{u}^h, \ \phi^h \text{at iteration } p+1)$$

$$P^*_{n+1} = \sum \Phi + (\theta\Delta t)^{-1} \sum_{\alpha=0}^{p} \delta\phi^{\alpha+1}_{n+1}$$

$$p_{n+1} = \mathrm{GWS}^h(\mathrm{L}(p), \left|\phi^{p+1}\right|_E < \varepsilon)_{n+1}$$

solution initiation: ICs for q^h are never (!) available

for $\mathbf{u}\cdot\hat{\mathbf{n}}\big|_{\partial\Omega_{\text{in}}}$ BCs:

iterate $\mathrm{D}\mathbf{P}^h = f(\mathrm{D}M^h, p_0 = 0)$

$$P^*_{n+1} = 0 + (\theta\Delta t)^{-1} \sum_{\alpha=0}^{p} \delta\phi^{\alpha+1}$$

at $\left|\phi^{p+1}\right|_E < \varepsilon$, $\mathbf{u}^h(\mathbf{x}, t_1)$ is initialized

solve $\mathrm{GWS}^h(\mathrm{L}(p)) \Rightarrow p_1$, index n, repeat iteration cycle

for $p\big|_{\partial\Omega_{\text{in, out}}}$ BCs:

solve $\mathrm{GWS}^h(\mathrm{L}(p) = 0$ homogenous $\Rightarrow p(\mathbf{x}, t_0))$

iterate $\mathrm{D}\mathbf{P}^h = f(\mathrm{D}M^h, p_0)$

$$P^*_{n+1} = 0 + (\theta\Delta t)^{-1} \sum_{\alpha=0}^{p} \delta\phi^{\alpha+1}$$

PPNS.5 $\{FQ\}_e$ Template Essence for GWS^h + θTS PPNS Algorithm

GWS^h + θTS for PPNS initial-value PDEs

$$\{FQ\}_e = [M200]_e\{QP - QN\}_e + \Delta t[\{UJ\}_e^T[M300J]_e\{Q\}_e + Pa^{-1}[M2KK]_e\{Q\}_e - \{b(Q)\}_e + \{BCs\}_e]_\theta$$

template essence:

$$\begin{aligned}\{FQ\}_e = (\ )(\ )\{\ \}(0;1)[M200]\{QP - QN\} \\ +(\Delta t)(\ )\{UJ\}(EKJ;0)[M300K]\{QP,QN\}_\theta \\ +(\Delta t, Ra^{-1})(\ )(\ )(EIK,EJK;-1)[M2IJ]\{QP,QN\}_\theta - \{b(Q)\}_\theta\end{aligned}$$

for $\{Q\}_e \Rightarrow \{UI\}_e$:

$$\begin{aligned}\{b(UI)\}_e = ()()\{\}(EKI;\ 0)[M20K]\{PHI + SPHN\} \\ +(\Delta t)()\{\}(EKI;\ 0)[M20K]\{PRESN\} \\ +(\Delta t, Gr/Re^2,\ GDOTI)()\{\}(0;\ 1)[M200]\{TEMP\}_\theta\end{aligned}$$

for $\{Q\}_e \Rightarrow \{TEMP\}_e$:

$$\begin{aligned}\{b(TEMP)\}_e = (\Delta t)(\ )\{\ \}(0;1)[M200]\{SRC\}_\theta \\ +(\Delta t, Nu, Pe^{-1})(\ )\{\ \}(0;1)[N200]\{TEMP - TREF\}_\theta\end{aligned}$$

PPNS.6 $[\mathrm{JAC}]_e$ Template Essence for GWS^h + θTS PPNS Algorithm

The Newton jacobian for $\{QI\}$ in DP^h, I not summed

$$
\begin{aligned}
[\mathrm{JAC}II]_e \equiv \partial\{\mathrm{FU}I\}_e / \partial\{\mathrm{U}I\}_e = (\ \ )(\ \ )\{\ \ \}(0;1)[\mathrm{M200}][\ \ ] \\
+ (\Delta t)(\ \ )\{\mathrm{U}J\}(\mathrm{E}KJ;0)[\mathrm{M300}K][\ \ ] \\
+ (\Delta t)(\ \ )\{\mathrm{UI}\}(\mathrm{E}K\mathrm{I};0)[\mathrm{M3}K00][\ \ ] \\
+ (\Delta t, \mathrm{Re}^{-1})(\ \ )\{\ \ \}(\mathrm{E}IK, \mathrm{E}JK;-1)[\mathrm{M2}IJ][\ \ ]
\end{aligned}
$$

$$
\begin{aligned}
[\mathrm{JAC}IJ]_e = \partial\{\mathrm{FU}I\} / \partial\{\mathrm{U}J\} \\
= (\Delta t)(\ \ )\{\mathrm{UI}\}(\mathrm{E}J\mathrm{I};0)[\mathrm{M3}J00][\ \ ]
\end{aligned}
$$

$$
[\mathrm{JAC}I\phi]_e = (\ \ )(\ \ )\{\ \ \}(\mathrm{E}KI;0)[\mathrm{M20}K][\ \ ]
$$

$$
[\mathrm{JAC}I\Theta]_e = (\Delta t, \mathrm{Gr}/\mathrm{Re}^2, \mathrm{GDOT}I)(\ \ )\{\ \ \}(0;1)[\mathrm{M200}][\ \ ]
$$

PPNS.7 GWS^h ⇒ TWS^h + θTS for PPNS, Accuracy, Stability

TS exercise on $\mathsf{L}(q)$ generates a kinetic flux vector jacobian matrix

DP, DE:

$$\mathsf{L}(q) = \frac{\partial q}{\partial t} + \frac{\partial}{\partial x_j}\left(u_j q - \mathrm{Pa}^{-1} \frac{\partial q}{\partial x_j} \right) - s(q) = 0$$

$$\cong q_t + \partial f_j / \partial x_j \Rightarrow q_t + \mathrm{A}_j \frac{\partial q}{\partial x_j}, \quad f_j \equiv u_j q \ \text{ and } \ \mathrm{A}_j \equiv \partial f_j / \partial q$$

TS:

$$q^{n+1} = q^n + \Delta t q_t^n + 1/2 \Delta t^2 q_{tt}^n + 1/6\, \Delta t^3 q_{ttt}^n + O(\Delta t^4)$$

$$q_t = -\mathrm{A}_j \frac{\partial q}{\partial x_j}$$

$$q_{tt} = \cdots = \frac{\partial}{\partial x_j}\left[\alpha \mathrm{A}_j \frac{\partial q}{\partial t} + \beta \mathrm{A}_j \mathrm{A}_k \frac{\partial q}{\partial x_k} \right]$$

$$q_{ttt} = \cdots = \frac{\partial}{\partial x_j}\left[\gamma \frac{\partial}{\partial x_k}\left(\mathrm{A}_j \mathrm{A}_k \frac{\partial q}{\partial t} \right) + \mu \frac{\partial}{\partial x_k}\left(\mathrm{A}_j \mathrm{A}_k \mathrm{A}_\ell \frac{\partial q}{\partial x_\ell} \right) \right]$$

Substituting into TS, taking lim (TS) ⇒ ε > 0 produces

DP, DE:

$$\mathsf{L}^m(q) = \mathsf{L}(q) - \frac{\Delta t}{2}\frac{\partial}{\partial x_j}\left[\alpha \mathrm{A}_j \frac{\partial q}{\partial t} + \gamma \frac{\Delta t}{3}\frac{\partial}{\partial x_k}\left(\mathrm{A}_j \mathrm{A}_k \frac{\partial q}{\partial t} \right) \right]$$

$$- \frac{\Delta t}{2}\frac{\partial}{\partial x_j}\left[\beta \mathrm{A}_j \mathrm{A}_k \frac{\partial q}{\partial x_k} + \mu \frac{\Delta t}{3}\frac{\partial}{\partial x_k}\left(\mathrm{A}_j \mathrm{A}_k \mathrm{A}_l \frac{\partial q}{\partial x_l} \right) \right] = 0$$

PPNS.8 TWS^h + θTS for PPNS, Kinetic Flux Vector Jacobian

TWS^h requires A_j, A_jA_k and $A_jA_kA_l$ be formed for INS

⇒ for $q = \{u_i, \Theta\}$:
(i not summed)

$$A_j \Rightarrow [A_j] = \begin{bmatrix} u_j + u_1 & & & 0 \\ & u_j + u_2 & & \\ & & u_j + u_3 & \\ 0 & & & u_j \end{bmatrix} \equiv \begin{bmatrix} u_j + u_i\delta_{ij}, & 0 \\ 0 \quad , & u_j \end{bmatrix}$$

$$[A_jA_k] = \begin{bmatrix} u_j + u_i\delta_{ij}, & 0 \\ 0 \quad , & u_j \end{bmatrix} \begin{bmatrix} u_k + u_i\delta_{ik}, & 0 \\ 0 \quad , & u_k \end{bmatrix}$$

$$= \begin{bmatrix} u_ju_k + u_ju_i\delta_{ik} + u_ku_i\delta_{ij} + u_iu_i\delta_{ij}\delta_{ik}, & 0 \\ 0 \quad , & u_ju_k \end{bmatrix}$$

Since $[A_j]$ is diagonal, generates no q cross-coupling and

$$\frac{\partial}{\partial x_j}\left[A_jA_k\frac{\partial q}{\partial x_k}\right] = \frac{\partial}{\partial x_j}\begin{bmatrix} u_ju_k\dfrac{\partial u_i}{\partial x_k} + u_iu_k\dfrac{\partial u_j}{\partial x_k}, & 0 \\ 0 \quad , & u_ju_k\dfrac{\partial \Theta}{\partial x_k} \end{bmatrix}$$

PPNS.9 TWS^h + θTS for PPNS, Alternative β-Term Forms

The TS lead β-term for $q = \{u_i\}$ has been generated many ways

$$\text{balancing tensor diffusivity}: \ \beta\frac{\Delta t}{2}\left[\mathrm{A}_j\mathrm{A}_k\right] \cong \frac{\beta\Delta t}{2}\left[u_j u_k\right]$$

defining local time scale $\Delta t/2 \approx h/|\mathbf{u}_e|$ leads to

$$\text{Petrov}-\text{Galerkin} \quad : \quad \frac{\beta\Delta t}{2}\left[\mathrm{A}_j\mathrm{A}_k\right] \cong \beta h\left[\hat{u}_j u_k\right]$$

$$\text{uniform } \Omega^h \quad : \quad h \cong \mathrm{C}(\det_e)^{1/n}$$

$$\text{non}-\text{uniform } \Omega^h \quad : \quad h \Rightarrow f(h_e \text{ parallel to } \hat{u}_j)$$

for Re >> 1, TS exercise on steady – state INS generates $\left[u_j u_k\right]$ identically

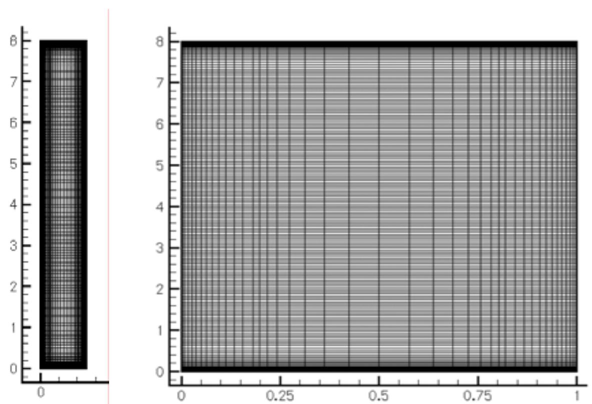
$$\text{uniform } \Omega^h \quad : \quad \beta \Rightarrow h^2\,\mathrm{Re}/12\,, \ \ h \cong \mathrm{C}(\det_e)^{1/n} \ \text{ not arbitrary!}$$

$$\text{non}-\text{uniform } \Omega^h \quad : \quad h \Rightarrow h_e \text{ is not analyzed}$$

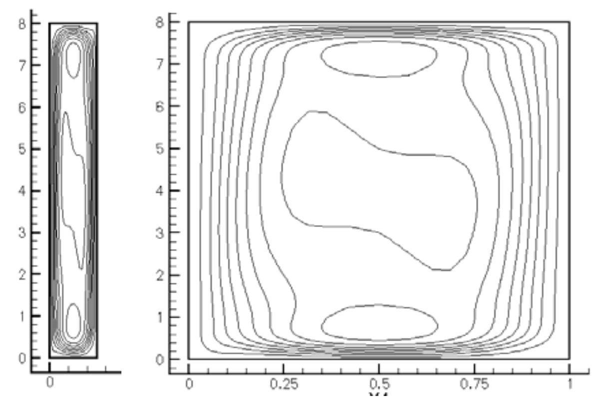
PPNS.10 TWS^h + θTS, Unsteady Thermal Cavity Validation

8:1 thermal cavity flowfield transitions to unsteady for Ra > 3.1E5

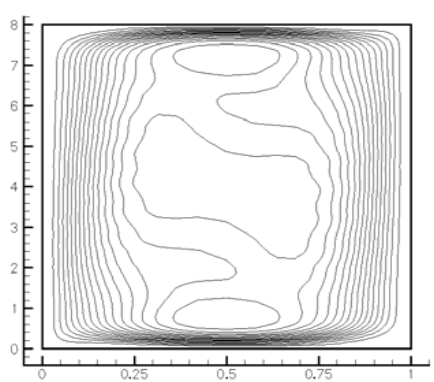
Graded 40 × 200 Ω^h
to scale & rectangularized



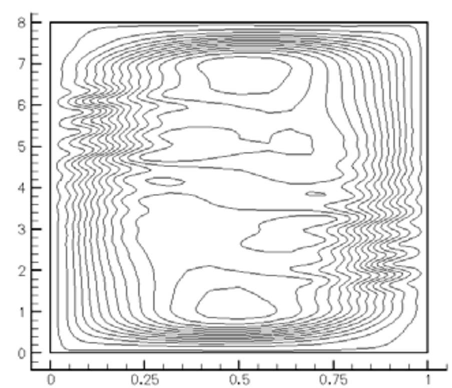
GWS^h + θTS steady solution, θ = 0.55, Ra = 3.4E5



Continuation at θ = 0.5 after 100Δt



Periodic solution snap shot



PPNS.11 TWS^h + θTS for PPNS, Closure for Turbulent Flow

Unsteady, non-D Reynolds-averaged INS with TKE closure

DM: $\nabla \cdot \mathbf{u} = 0$

D**P**: $$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \mathrm{Eu} \nabla p - \nabla \cdot \left(\mathrm{Re}^{-1} + \nu^t \right) \nabla \mathbf{u} + \frac{\mathrm{Gr}}{\mathrm{Re}^2} \Theta \hat{\mathbf{g}} = \mathbf{0}$$

DE: $$\frac{\partial \Theta}{\partial t} + \mathbf{u} \cdot \nabla \Theta - \nabla \cdot \left(\mathrm{Pe}^{-1} + \mathrm{Pr}^{-1} \nu^t \right) \nabla \Theta - s_\Theta = \mathbf{0}$$

D$E(k)$: $$\frac{\partial k}{\partial t} + \mathbf{u} \cdot \nabla k - \nabla \cdot \left(\mathrm{Pe}^{-1} + \nu^t / \mathrm{Pr}^t \right) \nabla k + \mathbf{T} \nabla \mathbf{u} - \varepsilon = \mathbf{0}$$

D$E(\varepsilon)$: $$\frac{\partial \varepsilon}{\partial t} + \mathbf{u} \cdot \nabla \varepsilon - \nabla \cdot \left(\mathrm{C}_\varepsilon \nu^t / \mathrm{Pr}^t \right) \nabla \varepsilon + \mathrm{C}_\varepsilon^1 \mathbf{T} \frac{\varepsilon}{k} \nabla \mathbf{u} - \mathrm{C}_\varepsilon^2 \varepsilon^2 / k = \mathbf{0}$$

PPNS iterative closure strategy

DM^h: $$\mathrm{L}(\phi) = -\nabla^2 \phi + \nabla \cdot \mathbf{u}^h = 0$$
$$\ell(\phi) = -\nabla \phi \cdot \hat{\mathbf{n}} - (\mathbf{u}^{n+1} - \mathbf{u}^h) \cdot \hat{\mathbf{n}} = 0$$

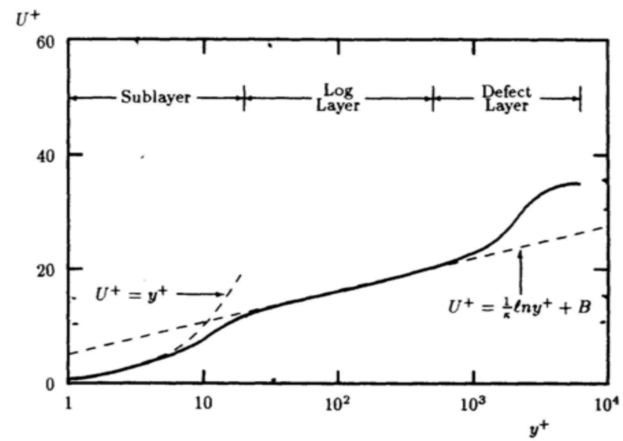
$\nabla \cdot$D**P**: $$\mathrm{L}(p) = -\mathrm{Eu} \nabla^2 p - s(u_i, \Theta) = 0$$
$$\ell(p) = \nabla p \cdot \hat{\mathbf{n}} + f(\mathrm{Re},\ \nabla^2 \mathbf{u} \cdot \hat{\mathbf{n}}) = 0$$

PPNS.12 BCs for *n*-D TKE Closure, Law-of-the-Wall

In *n*-D, low Re^t region resolution is computationally intense

recall Cole's law:

$$U^+ \equiv u / u_\tau = \kappa^{-1} \log(y^+ E) + B$$
$$y^+ \equiv u_\tau y / \upsilon$$



for near-wall production = dissipation

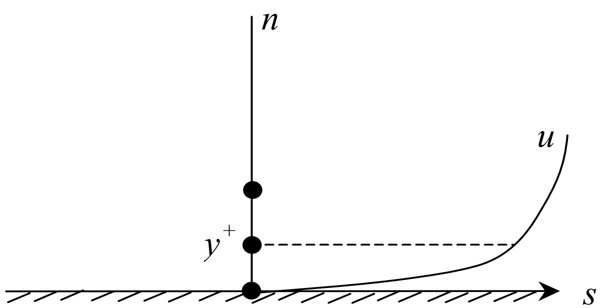
$$DE^m(k) \Rightarrow \quad \begin{aligned} \upsilon^t &= \kappa y u_\tau \\ k &= u_\tau^2 / C_\mu \\ \varepsilon &= (\kappa y)^{-1} |u_\tau|^3 \\ \tau_w &= \sqrt{C_\mu} k = u_\tau (C_\mu)^{-1/2} \end{aligned}$$

law-of-the-wall BC strategy

$$u_i(n_{\text{wall}}) = 0$$
$$k, \varepsilon(n_{\text{wall}+1}) = k, \varepsilon \text{ from } DE^m(k)$$

requires solution for u_τ at each wall + 1 node

$\Rightarrow$ consistancy check mandatory !



PPNS.13 TWS^h + θTS PPNS Algorithm Template Essence

TWS^h + θTS for $q^h \Rightarrow$ {U1, U2, U3, T, PHI; K, EPS, T*IJ*: PRES}^{T}

q-Newton $[\text{JAC}]_e$:

$$\begin{bmatrix} \text{JUU}, & \text{JUV}, & \text{JUW}, & 0, & \text{JU}\phi \\ \text{JVU}, & \text{JVV}, & \text{JVW}, & 0, & \text{JV}\phi \\ \text{JWU}, & \text{JWV}, & \text{JWW}, & \text{JWT}, & \text{JW}\phi \\ \text{JTU}, & \text{JTV}, & \text{JTW}, & \text{JTT}, & 0 \\ \text{J}\phi\text{U}, & \text{J}\phi\text{V}, & \text{J}\phi\text{W}, & 0, & \text{J}\phi\phi \end{bmatrix}_e ; \begin{bmatrix} \text{JKK}, & \text{JKE}, & \text{JKT}_{ij} \\ \text{JEK}, & \text{JEE}, & \text{JET}_{ij} \\ \text{JT}_{ij}\text{K}, & \text{JT}_{ij}\text{E}, & \text{JT}_{ij}\text{T}_{ij} \end{bmatrix}_e ; [\text{JPP}]_e$$

Iteration stabilization accrues to segregated state variable delay

for $\{Q1\}_e^T = \{\text{U}I, \text{T}, \phi\}_e : \ \{\text{F}Q1\}_e^p = \{\text{F}Q1(Q2\text{N})\}$

for $\{Q2\}_e^T = \{\text{K}, \text{EPS}, \text{T}_{ij}\}_e : \ \{\text{F}Q2\}_e^p = \{\text{F}Q2(Q1\text{N})\}$

at convergence for $\{Q1\}$, $\{Q2\}$, solve for $\{\text{PRES}\}_{n+1}$

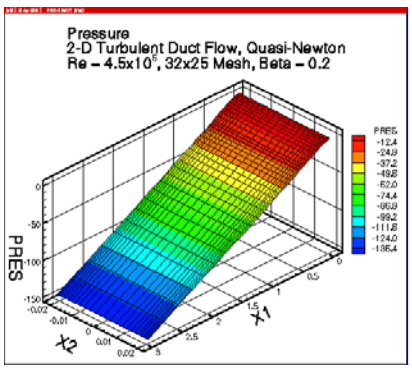
restart iteration loop

Template follows in aPSE area

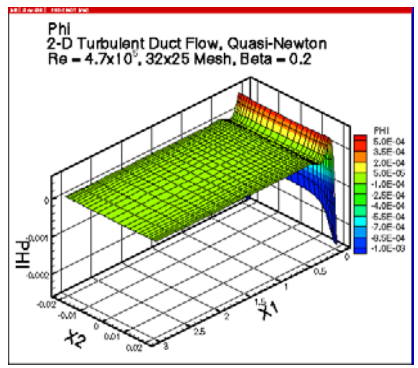
PPNS.14 TWS^h + θTS PPNS Algorithm, Turbulent Duct Flow

Turbulent duct flow, Re/L = 4 $\times 10^6$, TWS $\beta = 0.2$; ϕ, $\Sigma\phi$, pressure, k, ε

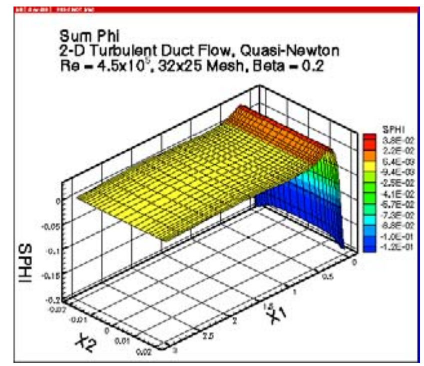
Pressure, E 02



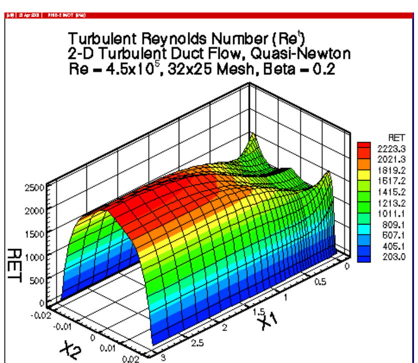
ϕ, E-04



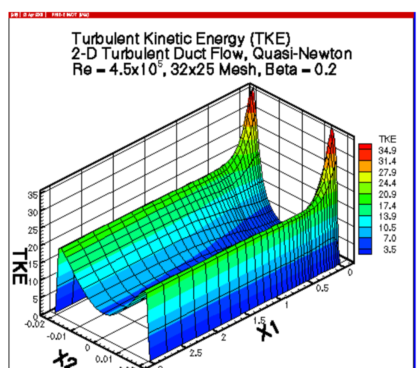
$\Sigma\phi$, E-04



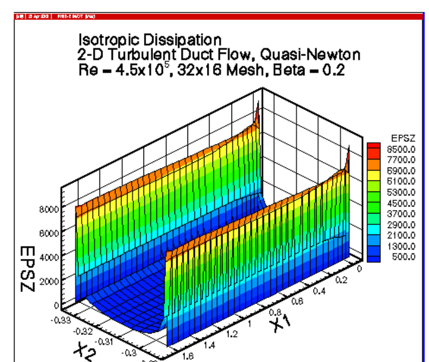
Re-turb, E 04



TKE, E-01



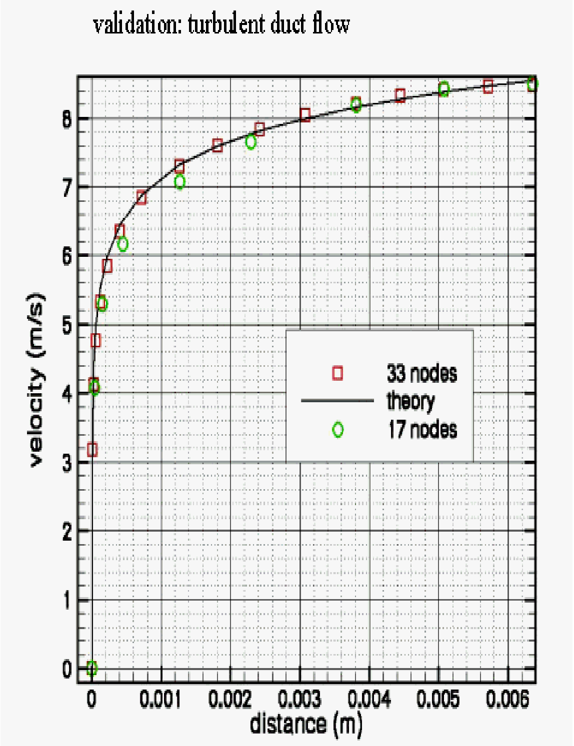
Dissipation, E-04



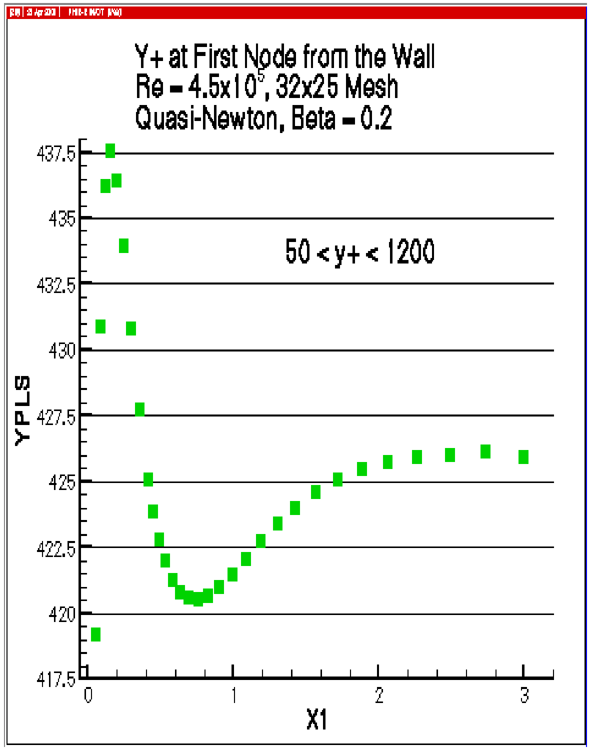
PPNS.15 TWS^h + θTS PPNS Algorithm, Turbulent Duct Flow

Turbulent duct flow, Re/L = 4 ×10^6, BC resolution, iterative convergence

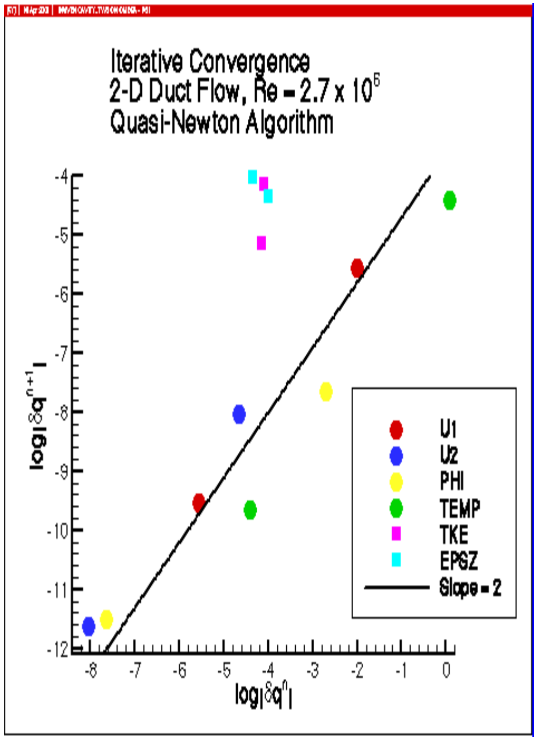
Mesh resolution



y^+ distribution



Iteration convergence



PPNS.16 : RaNS+TKE CFD Prediction of Turbulent Flows

Accurate prediction requires close attention to detail

RaNS

$$D(\bullet) \quad : \mathrm{L}(q) = \frac{\partial q}{\partial t} + \frac{\partial}{\partial x}(f_j - f_j^{v}) - s = 0,\ on\ \Omega \times t \subset \Re^n \times \Re^1$$

$$\text{INS} \quad : \quad q = \{\overline{u}, \overline{v}, \overline{w}, k, \varepsilon, \varphi, \overline{P}\},\ \text{and}\ \nabla \bullet \mathbf{u} = 0$$

$$f_j = f_j(\overline{u_j}, q, p\delta_{ij})$$

$$f_j^{v} = f_j^{v}(q, \mathrm{Re}, \mathrm{Pr}, \mathrm{Re}^t, \tau_{ij}, \mathrm{C}_q^{\alpha}, \beta)$$

$$s = s(q, \mathrm{Gr}, \mathrm{Re}, \tau_{ij}, \varepsilon, S_{ij})$$

Basically a time balance with kinetic and dissipative flux vectors

$$\mathrm{TWS}^h + \theta\mathrm{TS}: \quad \{\mathrm{F}Q\}_e = [\mathrm{M200}(\alpha,\gamma)]_e\{\Delta Q\}_e + \Delta t[\mathrm{M20}J]_e\{\mathrm{F}J\text{-}\mathrm{FV}J\}_e - \{\mathrm{b}\}_e$$

$$\{\mathrm{FV}J\}_e = \{f(\mathrm{Re}^{-1})\, S_{ij},\ (\mathrm{Re}^t/\mathrm{Re})S_{ij},\ (\beta\mathrm{Re})\overline{u_k}, \overline{u_j}, S_{ij})\}_e$$

$\Rightarrow$ one must generate these data for confidence

PPNS.17 RaNS Dissipative Flux Vector GWS Algorithms

$$\mathrm{GWS}^h\,(f_j^{\,\upsilon}) = \int_{\Omega} \Psi_\beta(\mathbf{x})\,\mathrm{L}\,(f_j^{\,\upsilon})\,d\tau = \mathrm{S}_e\{\mathrm{WS}(\cdot)\}_e = \{0\}$$

$$\{\mathrm{WS}(\cdot)\}_e = \int_{\Omega_e} \{N\}\,(f_j^{\upsilon}(\mathrm{Re}, \mathrm{Re}^t, \beta, \ldots. S_{ij}))_e\, d\tau$$

RaNS dissipative flux vector template pseudo-code

$\{\mathrm{F}JI(\upsilon)\}_e = (\mathrm{Re}^{-1})(\ \ )\{\ \ \}(\mathrm{E}JK;0)[\mathrm{M}20K]\{\mathrm{U}I\}$
$\quad +(\mathrm{Re}^{-1})(\ \ )\{\ \ \}(\mathrm{E}IK;0)[\mathrm{M}20K]\{\mathrm{U}J\}$

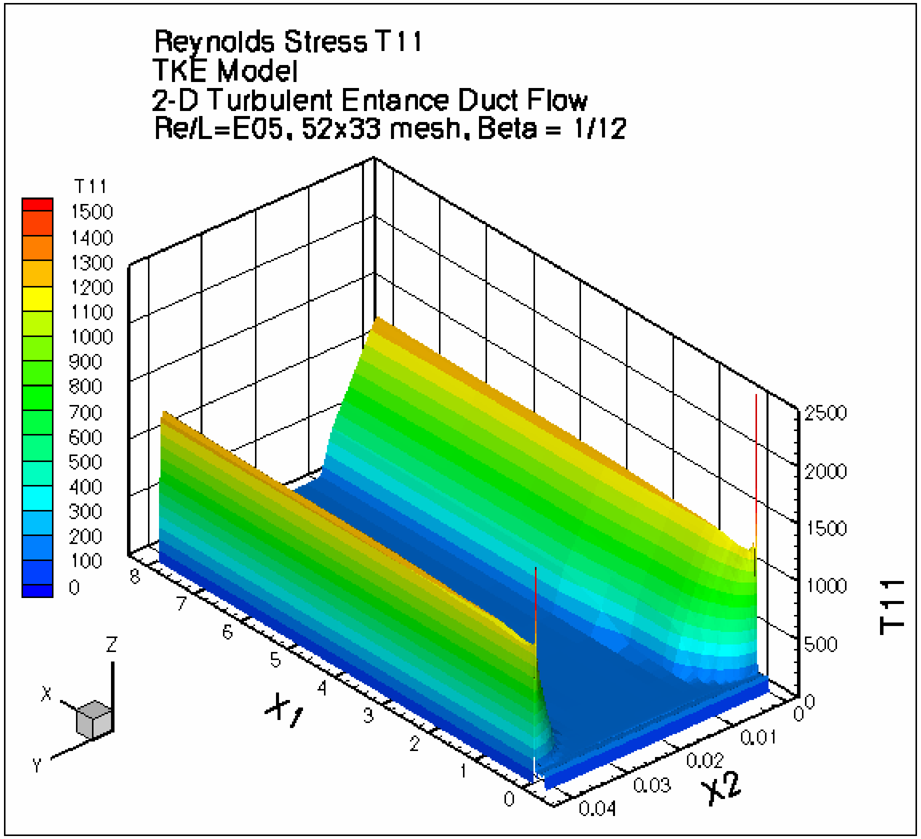
$\{\mathrm{F}JI(\mathrm{Re}^t)\}_e = (\mathrm{Re}^{-1})(\ \ )\{\mathrm{RET}\}(\mathrm{E}JK;0)[\mathrm{M}300K]\{\mathrm{U}I\}$
$\quad +(\mathrm{Re}^{-1})(\ \ )\{\mathrm{RET}\}(\mathrm{E}IK;0)[\mathrm{M}300K]\{\mathrm{U}J\}$

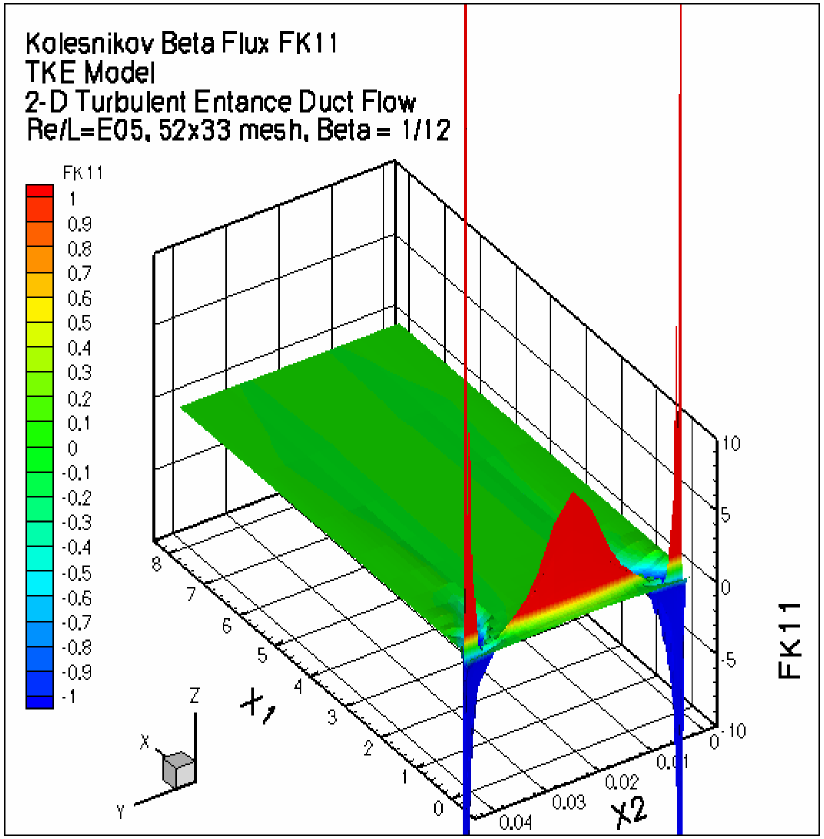
$\{\mathrm{F}JI(\beta)\}_e = (\mathrm{Re}/12)(\mathrm{h}^2)\{\mathrm{U}J,\mathrm{U}K\}(\mathrm{E}KL;0)[\mathrm{M}300L]\{\mathrm{U}I\}$
$\quad +(\mathrm{Re}/12)(\mathrm{h}^2)\{\mathrm{U}I,\mathrm{U}K\}(\mathrm{E}KL;0)[\mathrm{M}300L]\{\mathrm{U}J\}$

boundary conditions : apply zero at nodes where flux vector vanishes(only)
all other boundary nodes float

PPNS.18 RaNS Dissipative Flux Vector Distributions

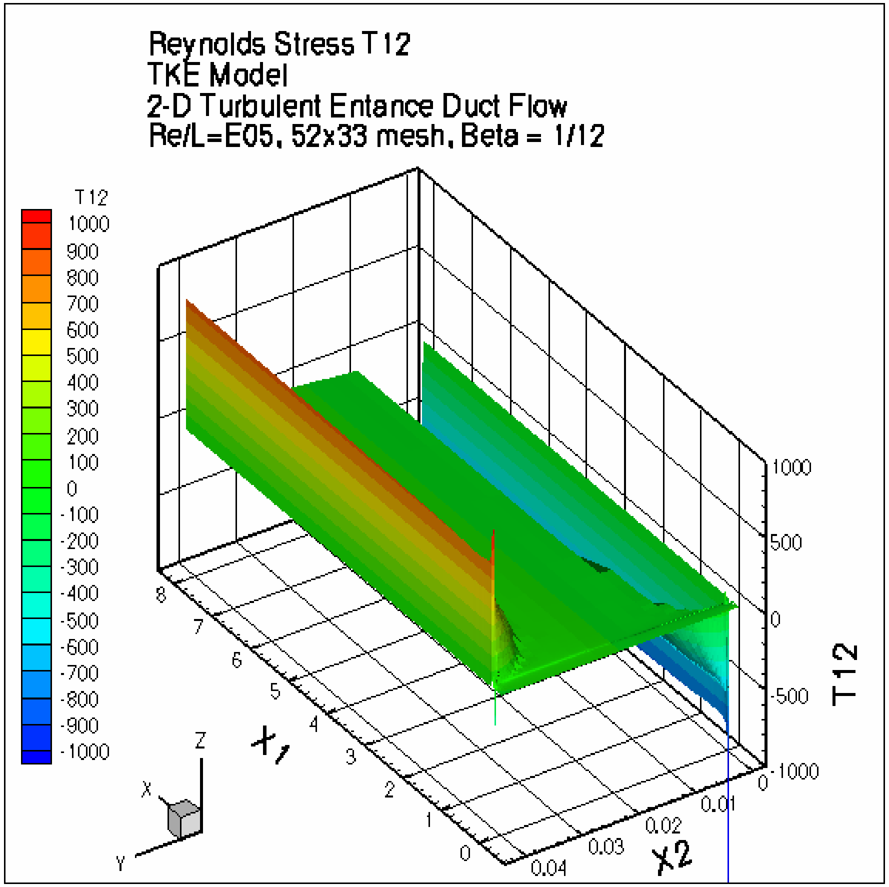
RaNS + TKE turbulent entrance duct TWS^h solution

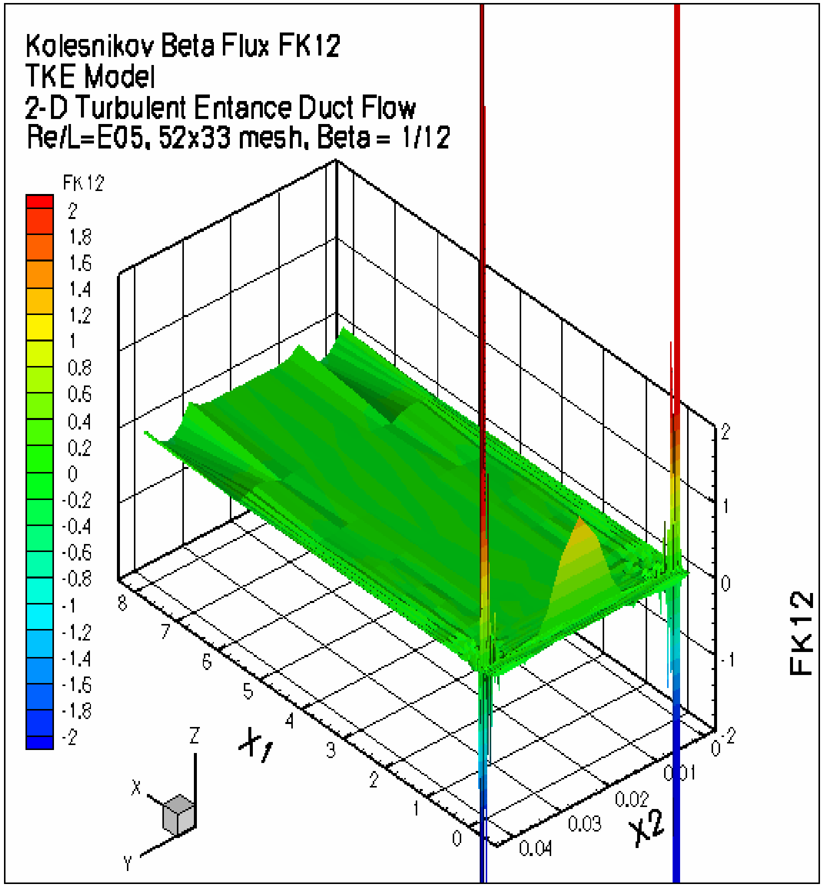




PPNS.18A RaNS Dissipative Flux Vector Distributions

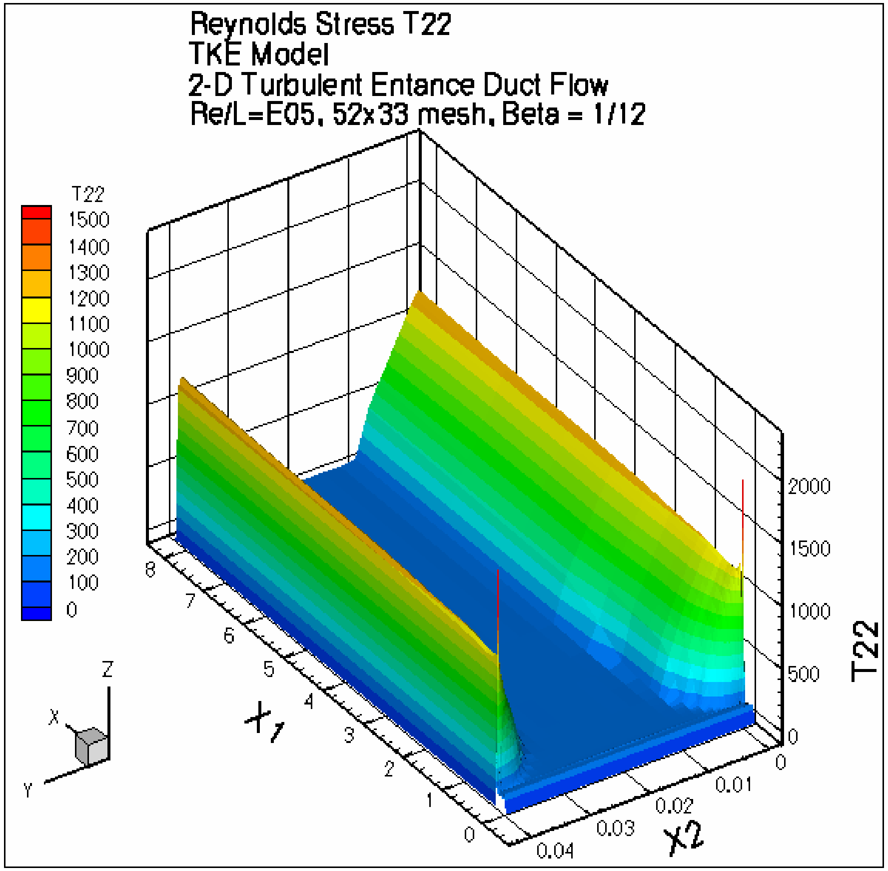
RaNS + TKE turbulent entrance duct TWS^h solution

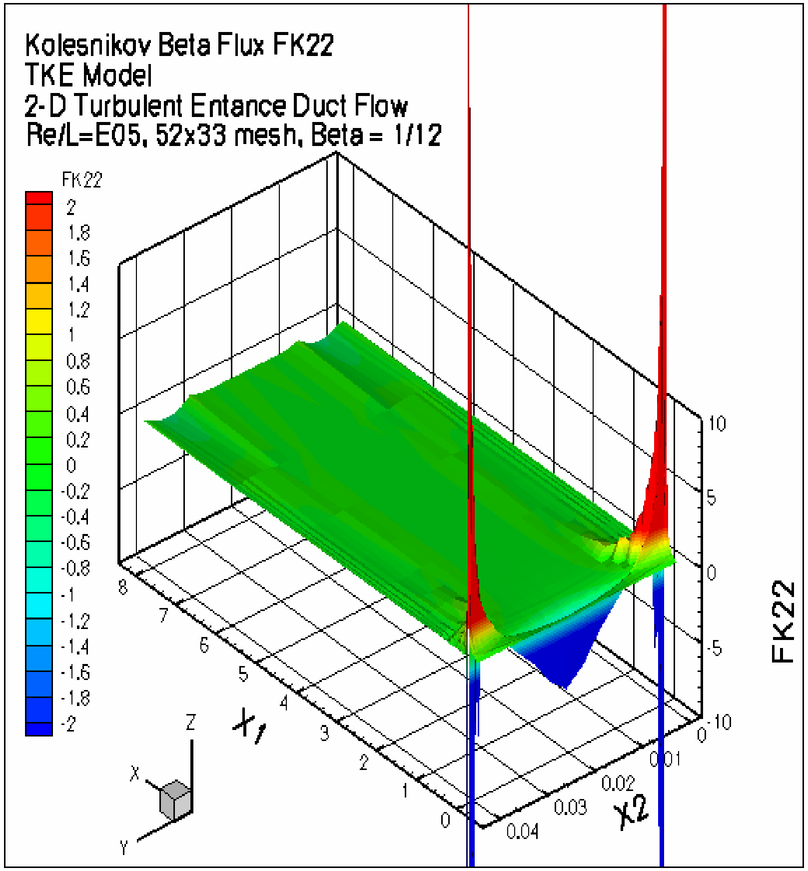




PPNS.18B RaNS Dissipative Flux Vector Distributions

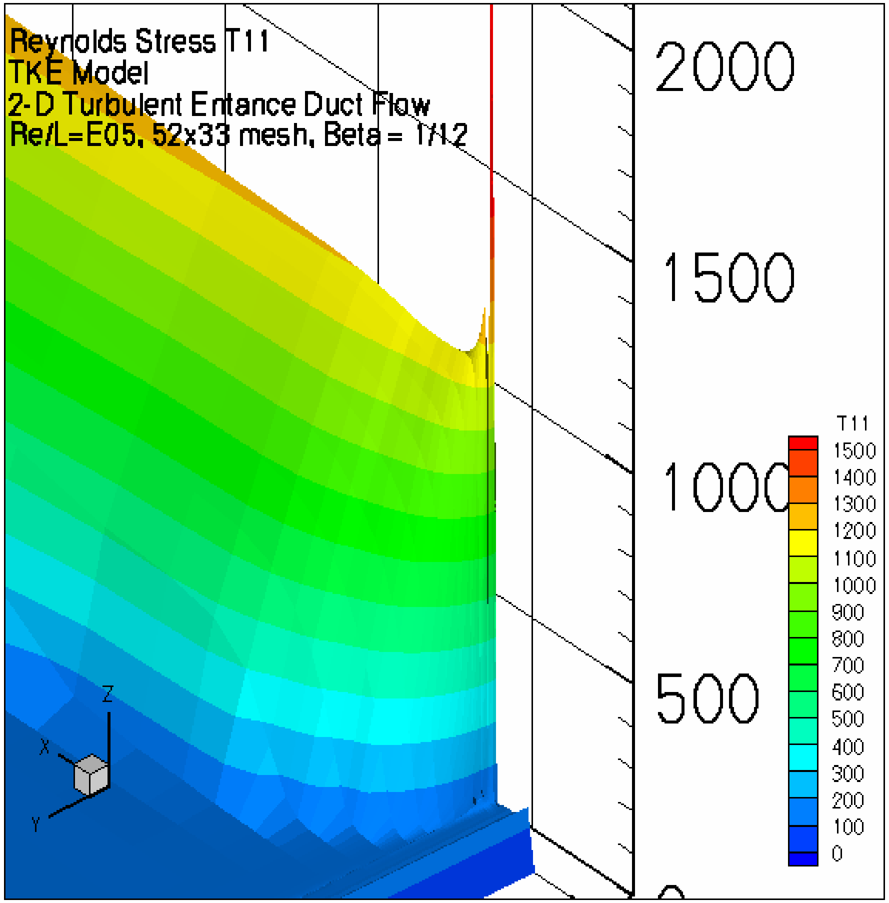
RaNS + TKE turbulent entrance duct TWS^h solution

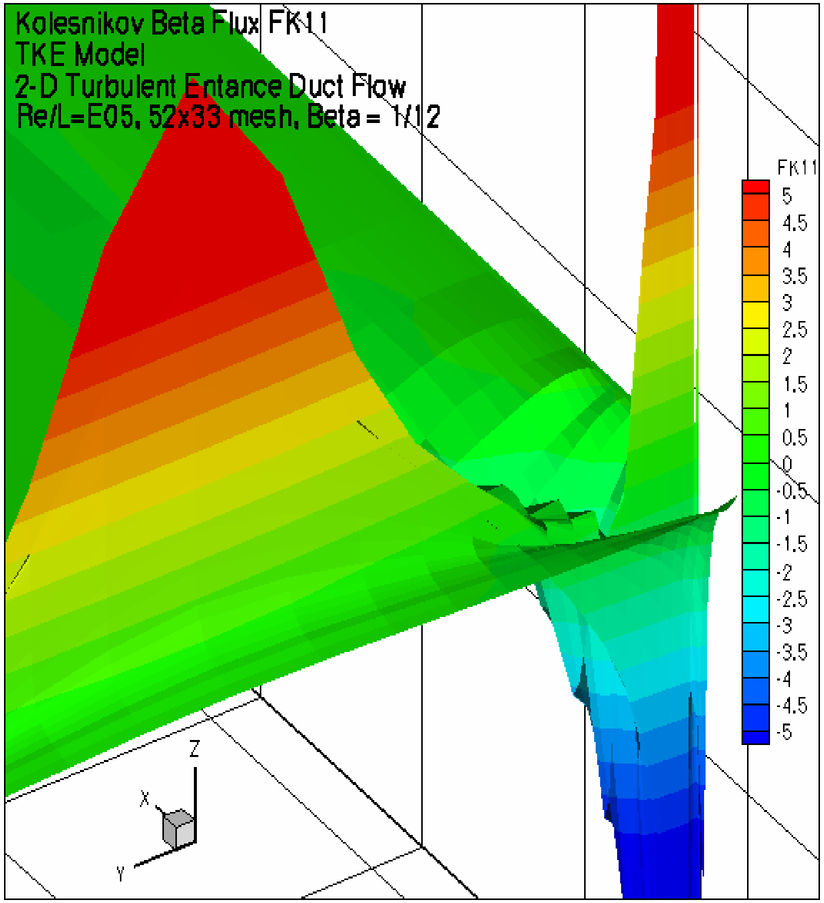




PPNS.18C RaNS Dissipative Flux Vector Distributions

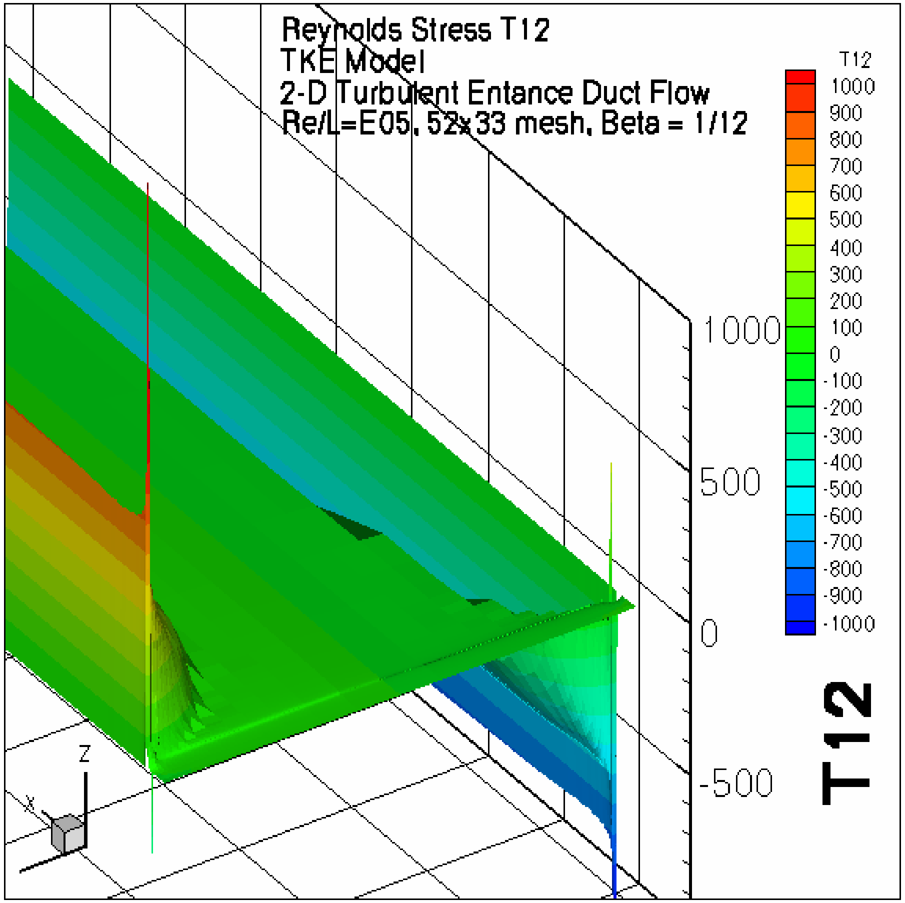
RaNS + TKE turbulent entrance duct TWS^h solution

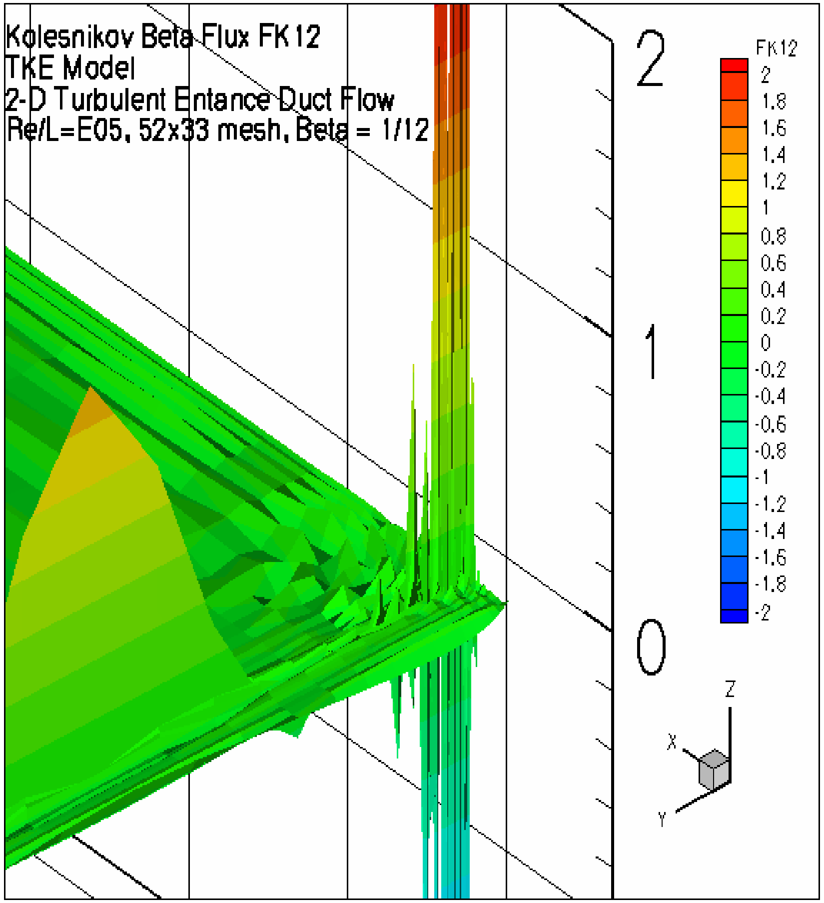




PPNS.18D RaNS Dissipative Flux Vector Distributions

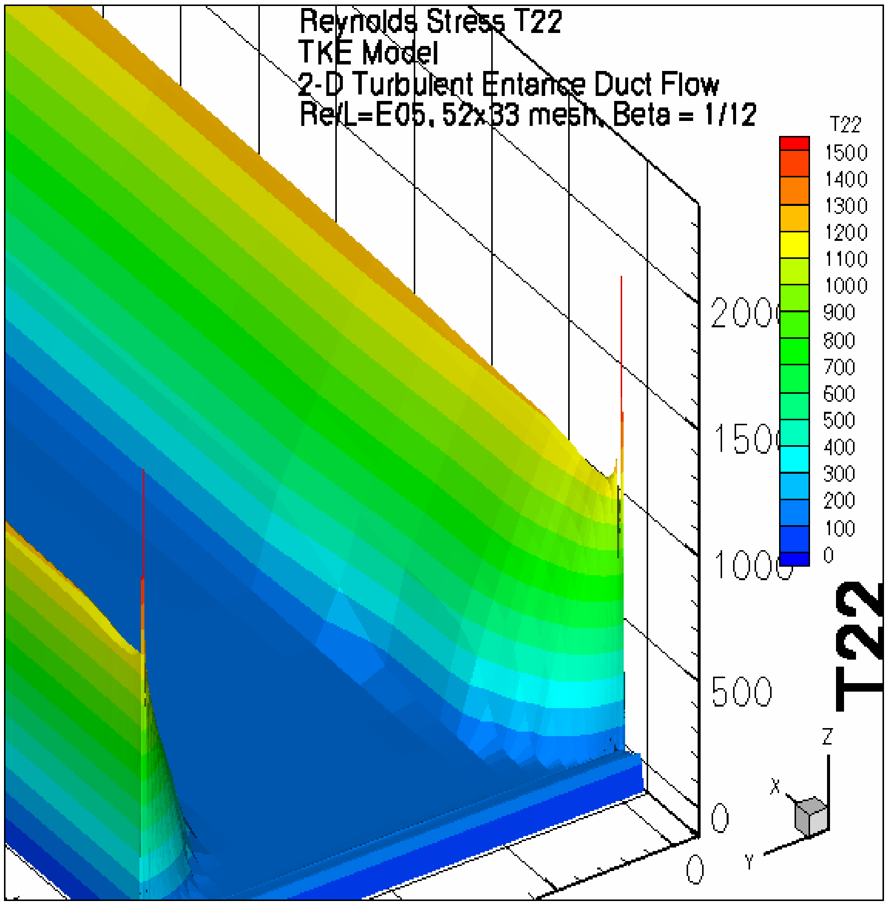
RaNS + TKE turbulent entrance duct TWS^h solution

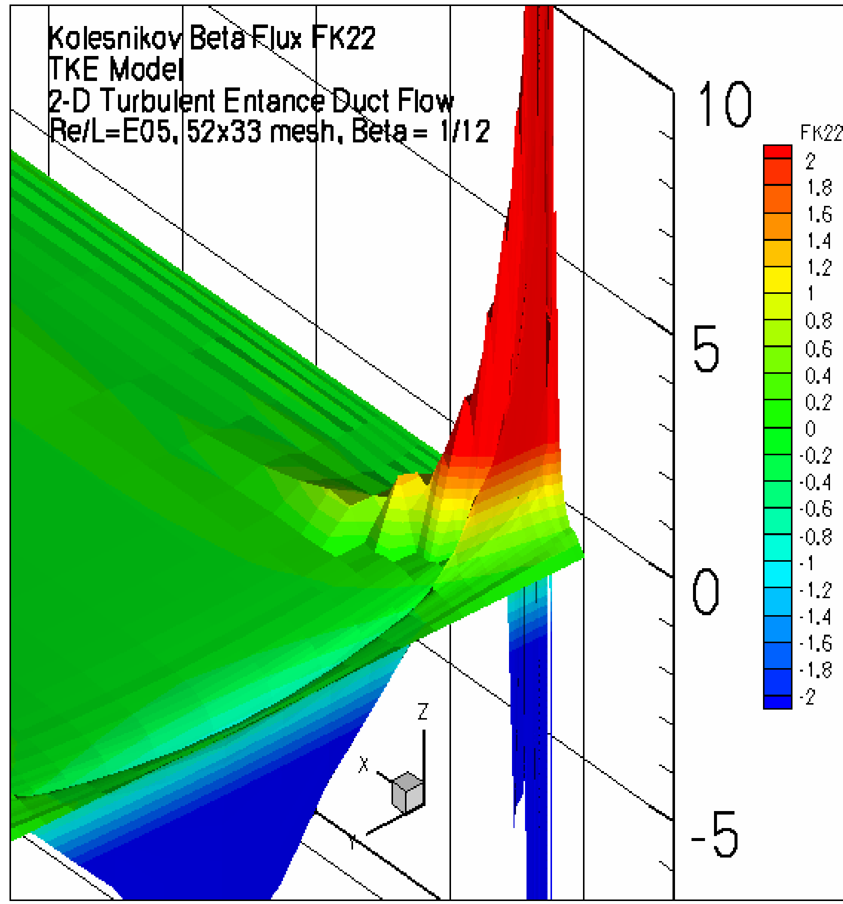




PPNS.18E RaNS Dissipative Flux Vector Distributions

RaNS + TKE turbulent entrance duct TWS^h solution





PPNS.19 Summary: Pressure Projection RaNS Algorithms

TWS^h + θTS PPNS iteration algorithm applicable to RaNS

key formulation issues: $\mathrm{D}\mathbf{P}^h = f\,(\mathrm{DM}^h \text{ via } P^*_{n+1})$

$\mathrm{D}M^h = f\,(\nabla^h \cdot \mathbf{u}^h, \phi^h \text{ at iteration } p+1)$

$P^*_{n+1} = \sum \phi + (\theta \Delta t)^{-1} \sum_{\alpha=0}^{p} \delta\varphi_{n+1}^{\alpha+1}$

$P_{n+1} = \mathrm{GWS}^h(\mathrm{L}\,(P), \left|\varphi^{P+1}\right|_E < \varepsilon)_{n+1}$

solution initiation: ICs for q^h are never (!) available

Algorithm performance fully resolvable

flux vector distribution solutions highly informative

- turbulence model phenomena detailed
- numerical dissipation clearly visualized
- meshing adequacy predictable via energy norms

⇒ a robust CFD basis