On the road to Nonlinear Incompressible Parabolized Stability Equations

Shaun Harris

December 12, 2018

Abstract

Parabolized Stability Equations (PSE) is an approach to analyze the streamwise evolution of single or interacting modes in a weakly nonparallel flow [Herbert, 1994]. Much of the numerical aspects of coding linear incompressible PSE is shown in previous works [Herbert, 1994]. However, adding the additional nonlinear terms can be beneficial, but more complicated from a numerical perspective. It is a more advanced technique and allows the prediction of disturbance growth from linear to strong nonlinear stages [Schmid and Henningson, 2001]. This work will outline common methods to modeling linear and nonlinear terms in the PSE equations, and will apply a method to capture the linear growth of TS waves along a flat plate boundary layer as depicted in Fig. 1 shown below [Bertolotti et al., 1992]. The nonlinear outline will be presented, but further investigation is required to capture the nonlinear growth.

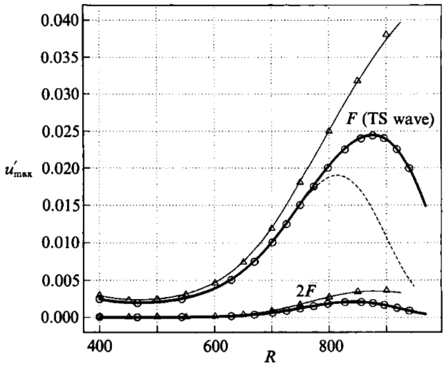


Figure 1: Amplitude of $u'_{\max}$ vs. Reynolds number R for TS waves at $F = 86$ and the first harmonic with $2F$ using initial TS amplitude of 0.25% (heavy lines) and 0.30% (thin lines) at $R = 400$. The dashed line shows the linear result. Symbols denote results of the direction Navier-Stokes simulation.

1 Blasius Boundary Layer Base Flow

The Blasius boundary layer can be found by integrating Eq. 1 as shown in [White, 2006].

$$\begin{aligned} f''' + ff'' &= 0 \\ f &= f(\eta) \\ \eta &= y\sqrt{\frac{U_\infty}{2\nu x}} \end{aligned} \tag{1}$$

With this solution, the base flow quantities and respective partial derivatives can be found in Eq. 2. These base flow values were used in both the Orr-Sommerfled spatial eigenvalue problem (OSS) and the Parabilized

Stability Equations (PSE) marching method.

$$\begin{array}{lll} U = U_\infty f', & U_x = U_\infty \left(\dfrac{-\eta}{2x}\right) f'', & U_y = \sqrt{\dfrac{U_\infty^3}{2\nu x}} f'', \\ & & U_{yy} = \dfrac{U_\infty^2}{2\nu x} f''', \\ V = \sqrt{\dfrac{\nu U_\infty}{2x}} (\eta f' - f), & V_x = \sqrt{\dfrac{\nu}{U}}_\infty 8x^3(-\eta f' + f - \eta^2 f''), & V_y = \dfrac{U_\infty}{2x} \eta f'' \end{array} \tag{2}$$

2 Orr-Sommerfeld Spatial Eigenvalue Problem

If we assume the Blasius profile is a parallel base flow, aligned with the x coordinate, then $U(y)$ and the governing equations for the fluctuating quantities are shown in Eq. 3.

$$\begin{aligned} \frac{\partial u}{\partial t} + U\frac{\partial u}{\partial x} + vU' &= -\frac{\partial P}{\partial x} + \frac{1}{Re}\nabla^2 u \\ \frac{\partial v}{\partial t} + U\frac{\partial v}{\partial x} &= -\frac{\partial P}{\partial y} + \frac{1}{Re}\nabla^2 v \\ \frac{\partial w}{\partial t} + U\frac{\partial w}{\partial x} &= -\frac{\partial P}{\partial z} + \frac{1}{Re}\nabla^2 w \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} &= 0 \end{aligned} \tag{3}$$

Introducing the normal mode assumptions $u(x, y, z, t) = \hat{u}(y, z)\exp(i(-\omega t + \alpha x))$ and rearranging we can find the spatial eigenvalue system shown in Eq. 4.

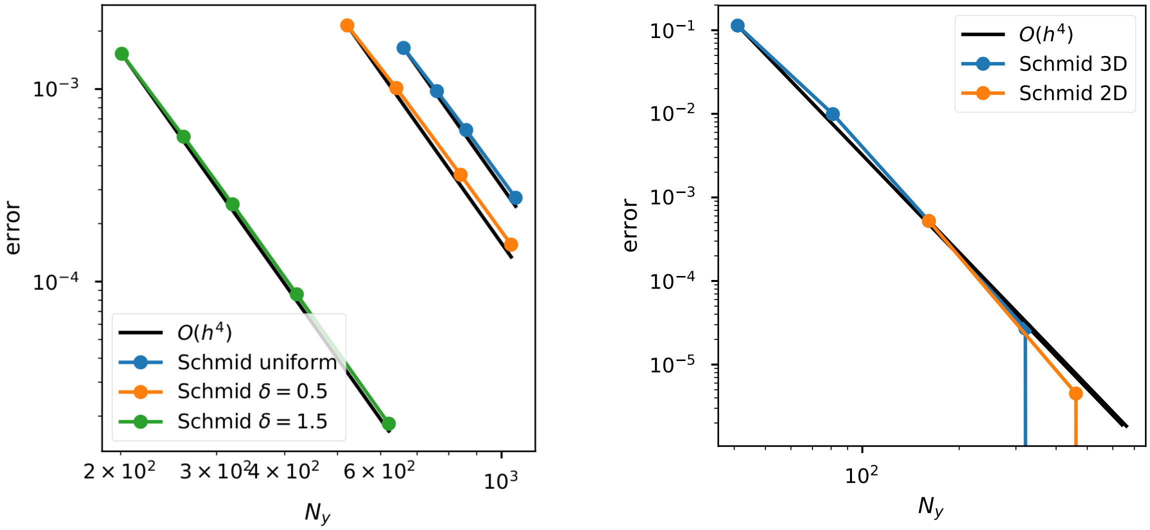
$$\begin{aligned} (-i\omega - \frac{1}{Re}(\partial_y^2 + \partial_z^2))\hat{u} + U'\hat{v} &= \alpha(-i\hat{P} - \frac{i}{Re}(\partial_y \hat{v} + \partial_z \hat{w}) - Ui\hat{u}) \\ (-i\omega - \frac{1}{Re}(\partial_y^2 + \partial_z^2))\hat{v} + \partial_y \hat{P} &= \alpha(-\frac{1}{Re}(\alpha\hat{v}) - Ui\hat{v}) \\ (\alpha\hat{v}) &= \alpha \cdot \hat{v} \\ (-i\omega - \frac{1}{Re}(\partial_y^2 + \partial_z^2))\hat{w} + \partial_z \hat{P} &= \alpha(-\frac{1}{Re}(\alpha\hat{w}) - Ui\hat{w}) \\ (\alpha\hat{w}) &= \alpha \cdot \hat{w} \\ \partial_y \hat{v} + \partial_z \hat{w} &= \alpha(-i\hat{u}) \end{aligned} \tag{4}$$

Then, we can introduce the state variables at each of the (y, z) locations and the system becomes a general eigenvalue problem shown in 5.

$$\mathbf{z}_i = \begin{bmatrix} \hat{u} \\ \alpha\hat{v} \\ \hat{v} \\ \alpha\hat{w} \\ \hat{w} \\ \hat{P} \end{bmatrix}_{z=z(i)} \quad \text{and } \hat{\mathbf{q}} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_{n_z-1} \\ \mathbf{z}_{n_z} \end{bmatrix} \tag{5}$$

$$\mathcal{A}\hat{\mathbf{q}} = \alpha\mathcal{B}\hat{\mathbf{q}}$$

It can be noted here, that we could alternatively taken advantage of common polynomial eigenvalue solvers, instead of introducing the inflation of the state vector as done here. However, this method was not used here. We can then solve this system shown in 4 for plane channel flow and for Blasius boundary layer and compare to the most unstable eigenvalues given by [Schmid and Henningson, 2001]. The derivative operators used finite difference on a stretched grid as outlined in Appendices A and B. This study used fourth order accurate derivative operators, and since light stretching was used in the grids, then the $\mathcal{O}(h^4)$ order of accuracy is retained in this study. The validation and convergence plots are shown in Fig. 2. The $\mathcal{O}(h^4)$ order of accuracy is retained as N_y number of points in the wall normal, y is increased. The 3D case was run, also, the 2D case with the spanwise wavenumber being set to zero $\beta = 0$.



(a) Blasius boundary layer validation case at $Re_\delta = 1000$ (Reynolds number based on displacement thickness) and $\omega = 0.26$ at various stretching ratios δ

(b) Channel flow validation case at $Re = 2000$ and $\omega = 0.3$.

Figure 2: Orr-Sommerfeld validation cases compared to [Schmid and Henningson, 2001]

This spatial eigenvalue problem is used for initial conditions into PSE, which is described in the next section.

3 Linear Parabolized Stability Equations

The parabolized stability equations (PSE) can be derived by starting from the governing equations. We decompose the Navier-Stokes equations into base flow and fluctuating quantities, then we subtract the base flow solution and obtain Eq. 6.

$$\begin{gathered}\frac{\partial \mathbf{u}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{U} + \frac{1}{\rho} \nabla p - \frac{1}{Re} \Delta \mathbf{u} = -\mathbf{u} \cdot \nabla \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0\end{gathered} \tag{6}$$

We then introduce the ansatz shown in Eq. 7 and retain only terms that are $\mathcal{O}(\frac{1}{Re})$ and greater. This leads to the Eq. 8. These have been derived in the primitive variables where $\mathbf{q}(x, y, z, t) = [u, v, w, p]^T$

$$\mathbf{q}(x, y, z, t) = \sum_{m=-M}^{M} \hat{\mathbf{q}}_m(x, y, z) e^{\left(i \int_{x_0}^{x} \alpha_m(\xi) d\xi - i m \omega t\right)} \tag{7}$$

$$\mathbf{A} \hat{\mathbf{q}}_m + \mathbf{B} \frac{\partial \hat{\mathbf{q}}_m}{\partial y} + \mathbf{C} \frac{\partial^2 \hat{\mathbf{q}}_m}{\partial y^2} + \mathbf{D} \frac{\partial \hat{\mathbf{q}}_m}{\partial x} + \mathbf{E} \frac{\partial \hat{\mathbf{q}}_m}{\partial z} + \mathbf{G} \frac{\partial^2 \hat{\mathbf{q}}_m}{\partial z^2} = \hat{\mathbf{F}}_m \tag{8}$$

Neglecting the nonlinear terms, $\hat{\mathbf{F}}_m = 0$, and using finite differences for the derivatives in y and z and using implicit Euler method for the x derivative, we obtain the resulting PSE system shown in Eq. 9.

$$(\mathbf{A} + \mathbf{B}\mathcal{D}_y + \mathbf{C}\mathcal{D}_{yy} + \frac{\mathbf{D}}{h_x} + \mathbf{E}\mathcal{D}_z + \mathbf{G}\mathcal{D}_{zz})_{i+1} \hat{\mathbf{q}}(x = x_{i+1}) = \frac{\mathbf{D}_i}{h_x} \hat{\mathbf{q}}(x = x_i) \tag{9}$$

Where all of the matrices are defined in Eq. 11. And for brevity, we can introduce $\mathbf{A}_{\text{solve}}$ and observe the linear system as shown in 10.

$$\mathbf{A}_{\text{solve}} = \mathbf{A} + \mathbf{B}\mathcal{D}_y + \mathbf{C}\mathcal{D}_{yy} + \mathbf{E}\mathcal{D}_z + \mathbf{G}\mathcal{D}_{zz}$$
$$\left(\mathbf{A}_{\text{solve}} + \frac{\mathbf{D}}{h_x}\right)_{i+1} \hat{\mathbf{q}}_{i+1} = \mathbf{b}_i \tag{10}$$

$$\mathbf{A} = \begin{bmatrix} r + U_x & U_y & 0 & i\alpha \\ 0 & r + V_y & 0 & 0 \\ 0 & 0 & r & 0 \\ i\alpha & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} V & 0 & 0 & 0 \\ 0 & V & 0 & 1 \\ 0 & 0 & V & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$
$$\mathbf{C} = \begin{bmatrix} \frac{-1}{Re} & 0 & 0 & 0 \\ 0 & \frac{-1}{Re} & 0 & 0 \\ 0 & 0 & \frac{-1}{Re} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} U & 0 & 0 & 1 \\ 0 & U & 0 & 0 \\ 0 & 0 & U & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$
$$\mathbf{E} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \frac{-1}{Re} & 0 & 0 & 0 \\ 0 & \frac{-1}{Re} & 0 & 0 \\ 0 & 0 & \frac{-1}{Re} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Where $r = -im\omega + i\alpha U + \frac{\alpha^2}{Re}$

It is noted here, that the α value used in the $\mathbf{A}_{\text{solve}}$ matrix uses a guess for α that is at the next x location. Since this is a guess, the correction for this α must satisfy the following closure equation and update shown in Eq. 12

$$\int_\Omega \hat{\mathbf{q}}^{\mathsf{H}} \hat{\mathbf{q}}_x d\Omega = 0$$

then iterative update becomes:

$$\alpha_{i+1}^{n+1} = \alpha_{i+1}^{n} - \frac{i}{\Delta x} \frac{\int_\Omega \hat{\mathbf{q}}_{i+1}^{\mathsf{H}} (\hat{\mathbf{q}}_{i+1} - \hat{\mathbf{q}}_j) d\Omega}{\int_\Omega |\hat{\mathbf{q}}_{i+1}|^2 d\Omega} \tag{12}$$

$$\text{until } \int_\Omega \hat{\mathbf{q}}^{\mathsf{H}} \hat{\mathbf{q}}_x d\Omega = 0 \text{ is satisfied}$$

We can therefor produce the linear solution procedure shown in Fig. 3.

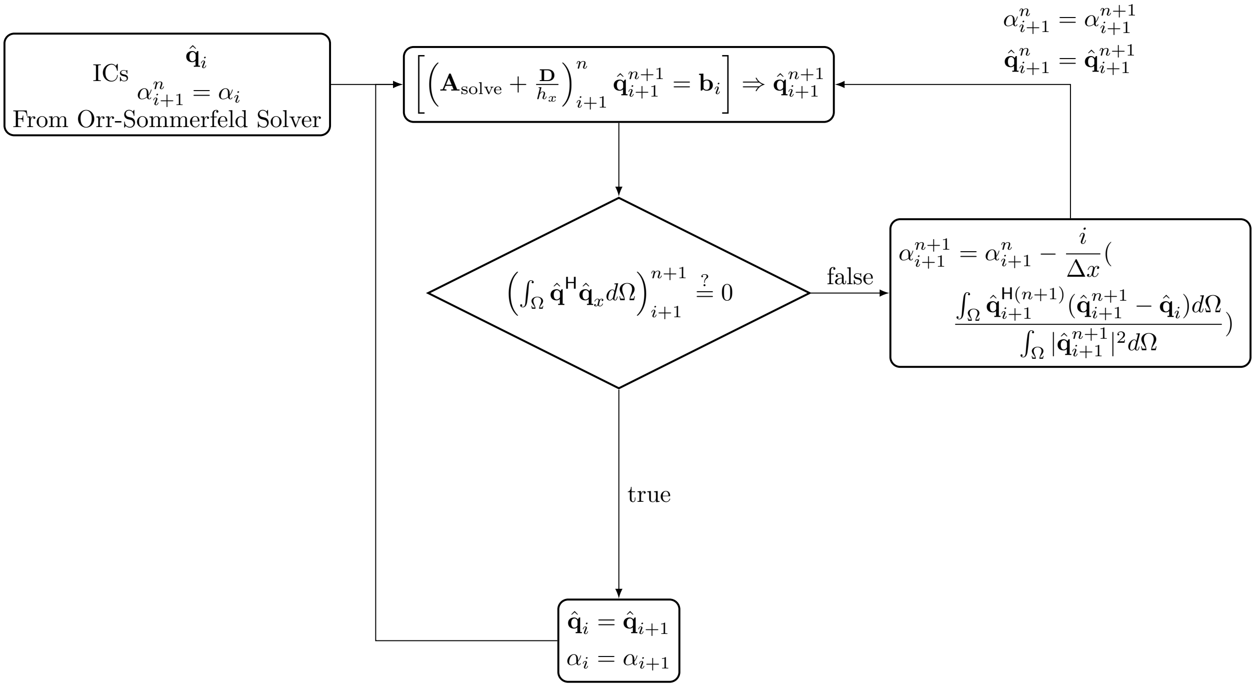

$\alpha^{n}_{i+1} = \alpha^{n+1}_{i+1}$
$\hat{\mathbf{q}}^{n}_{i+1} = \hat{\mathbf{q}}^{n+1}_{i+1}$
ICs
$\hat{\mathbf{q}}_i$
$\alpha^{n}_{i+1} = \alpha_i$
From Orr-Sommerfeld Solver
$\left[\left(\mathbf{A}_{\text{solve}} + \frac{\mathbf{D}}{h_x}\right)^{n}_{i+1} \hat{\mathbf{q}}^{n+1}_{i+1} = \mathbf{b}_i\right] \Rightarrow \hat{\mathbf{q}}^{n+1}_{i+1}$
$\left(\int_\Omega \hat{\mathbf{q}}^{\mathsf{H}} \hat{\mathbf{q}}_x d\Omega\right)^{n+1}_{i+1} \overset{?}{=} 0$
false
$\alpha^{n+1}_{i+1} = \alpha^{n}_{i+1} - \frac{i}{\Delta x}\left(\frac{\int_\Omega \hat{\mathbf{q}}^{\mathsf{H}(n+1)}_{i+1} (\hat{\mathbf{q}}^{n+1}_{i+1} - \hat{\mathbf{q}}_i) d\Omega}{\int_\Omega |\hat{\mathbf{q}}^{n+1}_{i+1}|^2 d\Omega} \right)$
true
$\hat{\mathbf{q}}_i = \hat{\mathbf{q}}_{i+1}$
$\alpha_i = \alpha_{i+1}$

Figure 3: Linear Parabolized Stability Equations Solution Procedure

We can then use this procedure to test two different cases. [Herbert, 1994] uses a parallel base flow eigenmode jump case where the initial $\omega = 0.0172$ and then the marching procedure uses $\omega = 0.0344$ with a steady base flow. It was shown that we can reproduce the results shown in [Herbert, 1994] shown in Fig. 4. We observe good agreement for this case.

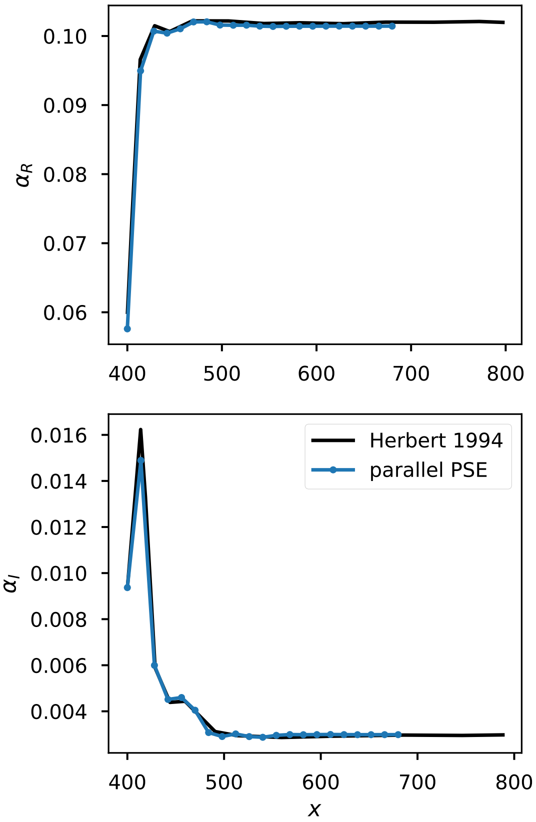


Figure 4: Eigenmode jump case compared to [Herbert, 1994] for $Re_0 = 400$ and $\omega = 0.0344$ with initial conditions with similar Re_0 but with $\omega = 0.0172$ from the OSS solver

A second case would be to replicate the growth and decay of a Tollmien-Schlichting wave along a flat plate boundary layer. Replicating and comparing to the case observed in [Herbert, 1994] we find good agreement for 2D and 3D case. We can also run the linear stability Orr-Sommerfeld solver (OSS) and see good agreement in that case as well.

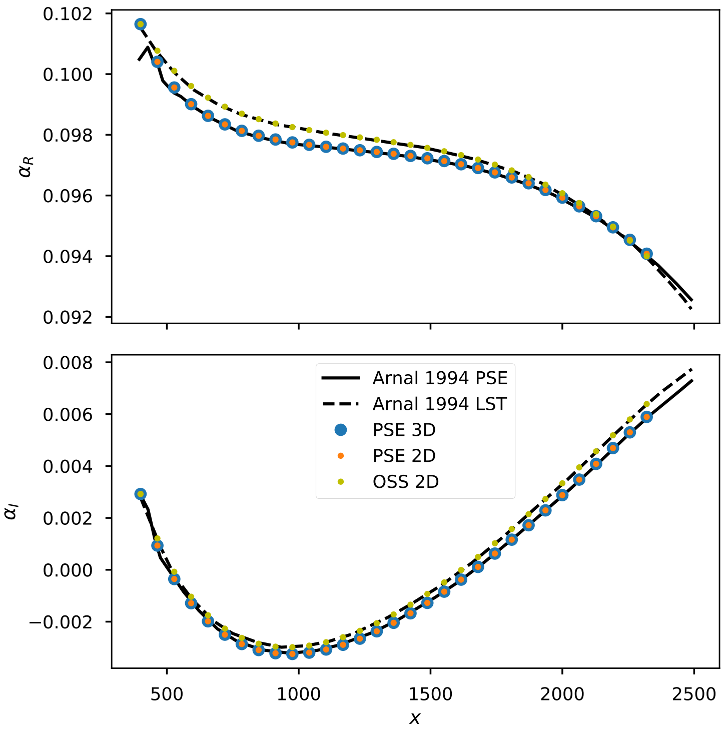


Figure 5: α values for TS wave for $Re_0 = 400$ and $F = 86$ or $\omega = 0.0344$

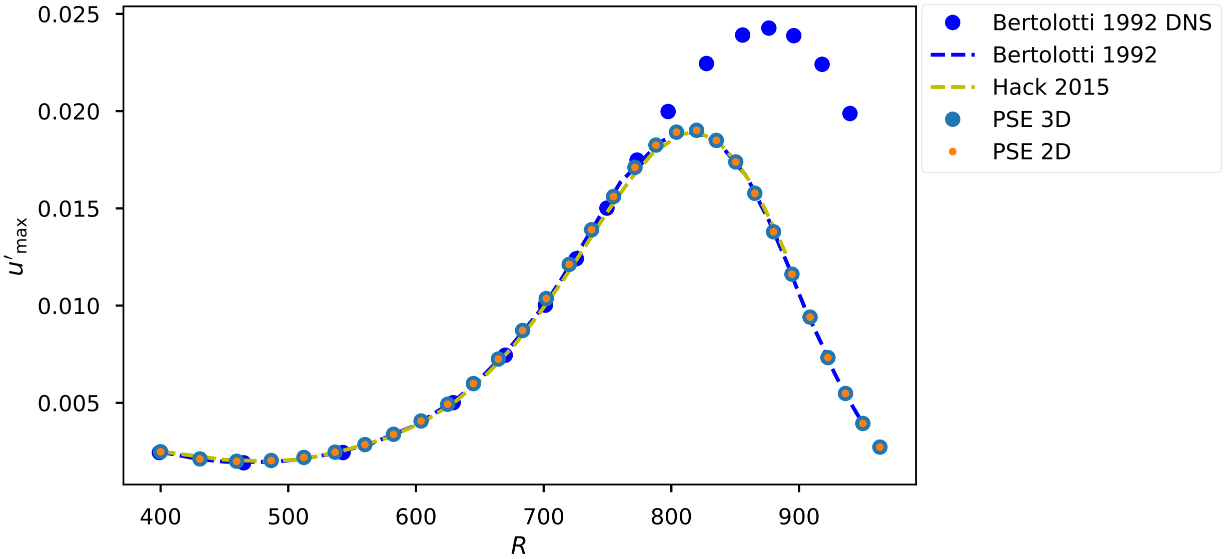


Figure 6: Maximum fluctuating growth and decay when initial amplitude is reduced to 0.25% of the normalized quantity. This is the same case as Fig. 5

A third case was run to test the three dimensional aspect of the algorithm. However, further investigation is required to resolve the discrepancy seen here in Fig. 7.

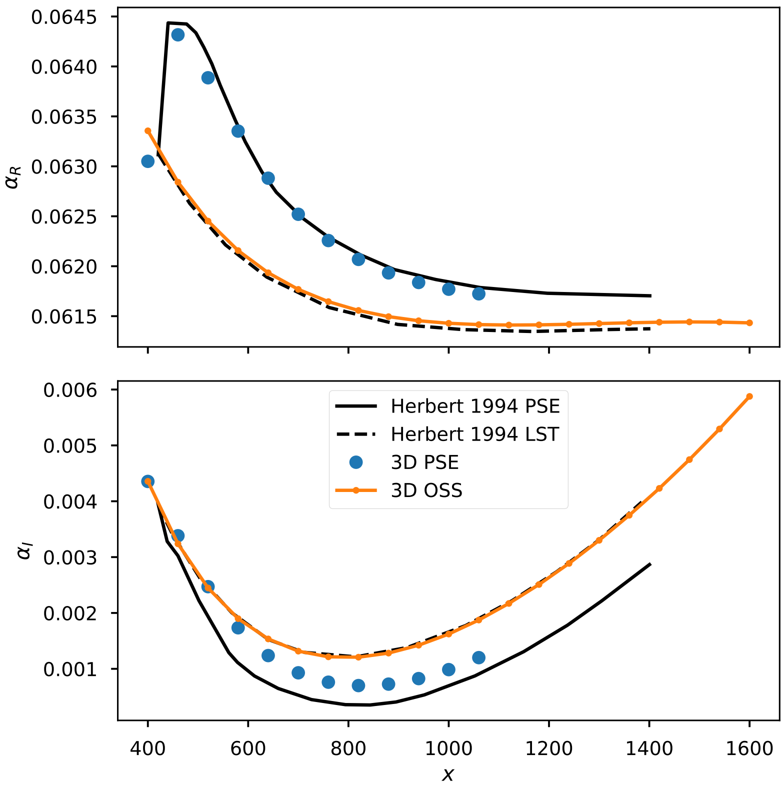


Figure 7: Case from [Herbert, 1994] to test the 3D case when $\beta = 0.14$, $\omega = 0.02604$ and $Re_0 = 400$. A discrepancy is observed here that must be resolved before further advancement can be done in the algorithm.

4 Nonlinear Parabolized Stability Equations

The nonlinear PSE system can be seen by keeping the $\hat{\mathbf{F}}_m$ term. This term is added to the right hand side as a forcing function, and the system then becomes dependant on the other ω modes. These additional modes must then all be marched together and an extra nonlinear loop is added to the solution procedure as seen in Fig. 8

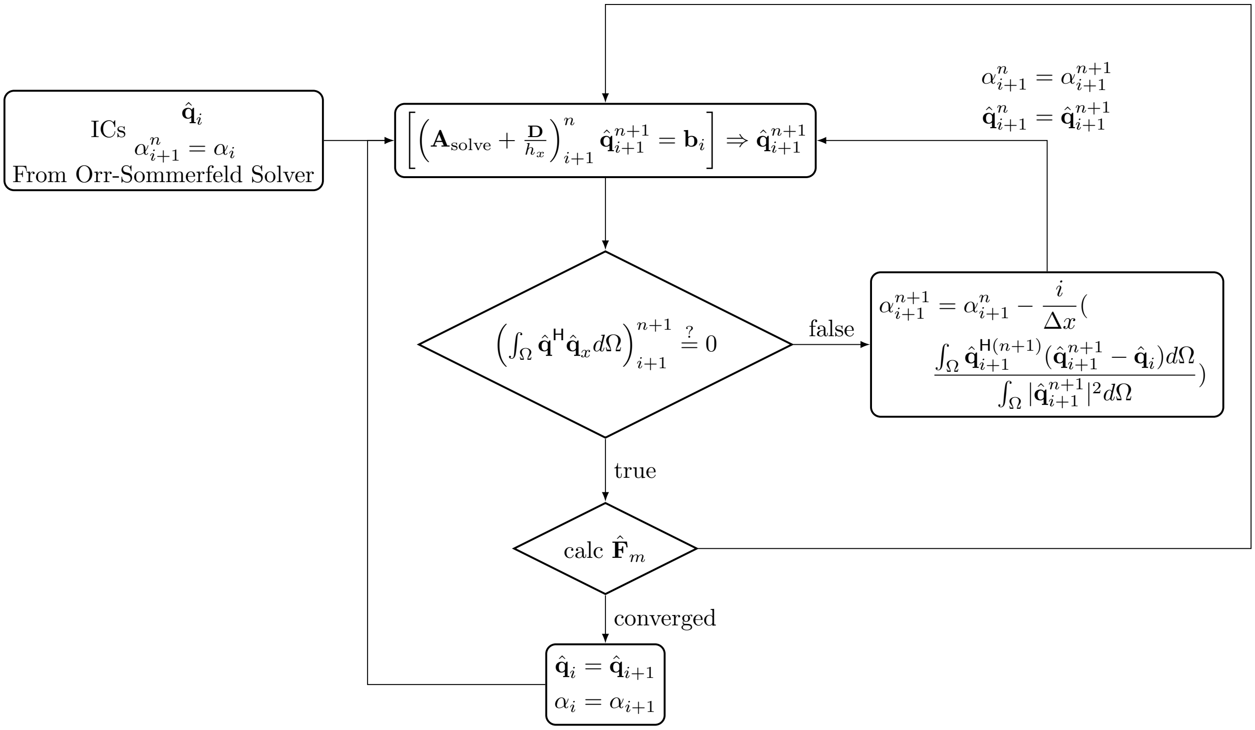

ICs $\hat{\mathbf{q}}_i$ $\alpha_{i+1}^{n} = \alpha_i$
From Orr-Sommerfeld Solver
$\left[\left(\mathbf{A}_{\text{solve}} + \frac{\mathbf{D}}{h_x}\right)_{i+1}^{n} \hat{\mathbf{q}}_{i+1}^{n+1} = \mathbf{b}_i\right] \Rightarrow \hat{\mathbf{q}}_{i+1}^{n+1}$
$\alpha_{i+1}^{n} = \alpha_{i+1}^{n+1}$
$\hat{\mathbf{q}}_{i+1}^{n} = \hat{\mathbf{q}}_{i+1}^{n+1}$
$\left(\int_\Omega \hat{\mathbf{q}}^{\mathsf{H}} \hat{\mathbf{q}}_x d\Omega\right)_{i+1}^{n+1} \overset{?}{=} 0$
false
$\alpha_{i+1}^{n+1} = \alpha_{i+1}^{n} - \frac{i}{\Delta x}\left(\frac{\int_\Omega \hat{\mathbf{q}}_{i+1}^{\mathsf{H}(n+1)}(\hat{\mathbf{q}}_{i+1}^{n+1} - \hat{\mathbf{q}}_i)d\Omega}{\int_\Omega |\hat{\mathbf{q}}_{i+1}^{n+1}|^2 d\Omega}\right)$
true
calc $\hat{\mathbf{F}}_m$
converged
$\hat{\mathbf{q}}_i = \hat{\mathbf{q}}_{i+1}$
$\alpha_i = \alpha_{i+1}$

Figure 8: Nonlinear Parabolized Stability Equation Solution Procedure

Where the nonlinear terms are added to the $\mathbf{b}_i$, where Eq. 13 shows the nonlinear terms.

$$\hat{\mathbf{F}}_m = \begin{bmatrix} -\sum_{m=-M}^{M}(\hat{u}_m(\hat{u}_x + \hat{u}i\alpha) + \hat{v}_m\hat{u}_y + \hat{w}_m\hat{u}_z)e^{i\Theta_m} \\ -\sum_{m=-M}^{M}(\hat{u}_m(\hat{v}_x + \hat{v}i\alpha) + \hat{v}_m\hat{v}_y + \hat{w}_m\hat{v}_z)e^{i\Theta_m} \\ -\sum_{m=-M}^{M}(\hat{u}_m(\hat{w}_x + \hat{w}i\alpha) + \hat{v}_m\hat{w}_y + \hat{w}_m\hat{w}_z)e^{i\Theta_m} \\ 0 \end{bmatrix} \tag{13}$$

The full nonlinear PSE equation set can be defined in Appendix C. Once these terms are added into the algorithm, then recreating the solution seen in Fig. 9. This calculation was not completed in this study, but will be investigated further.

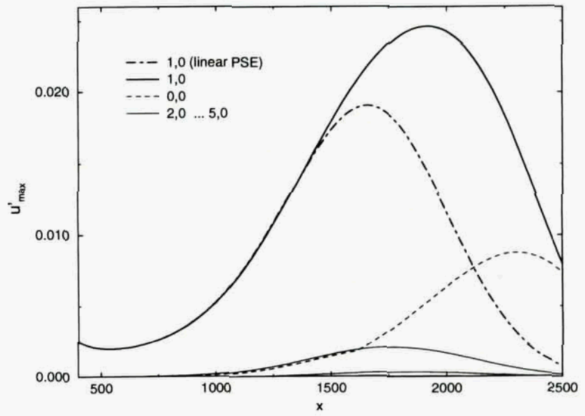


Figure 19: Linear and nonlinear evolution of a TS wave of frequency $\omega_1 = 0.0344$ with $u'_{max} = 0.0025$ at $x_s = Re_0 = 400$. The various modes are labeled n, m for their frequency ω_n and spanwise wavenumber β_m as in figure 18 (l is omitted if $L = 0$).

Figure 9: Nonlinear marching extracted from [Herbert, 1994]

5 Conclusion

A spatial eigenvalue problem was solved and validated against [Schmid and Henningson, 2001] eigenvalues for a plane channel flow and a flat plate boundary layer. This was used as an input to march the TS mode in the x coordinate. We observed growth and decay of a TS wave as seen in [Herbert, 1994]. The 3D aspect of this is not validated yet, and requires further investigation. The nonlinear system of equations for the PSE was presented, but not implemented.

References

[Bertolotti et al., 1992] Bertolotti, F. P., Herbert, T., and Spalart, P. R. (1992). Linear and nonlinear stability of the {Blasius} boundary layer. *Journal of Fluid Mechanics*, 242:441–474.

[Herbert, 1994] Herbert, T. (1994). *Parabolized Stability Equations.*

[Schmid and Henningson, 2001] Schmid, P. J. and Henningson, D. S. (2001). *Stability and Transition in Shear Flows (Textbook).* Springer.

[Vinokur, 1983] Vinokur, M. (1983). On one-dimensional stretching functions for finite-difference calculations. *Journal of Computational Physics*, 50(2):215–234.

[White, 2006] White, F. (2006). *Viscous Fluid Flow.* McGraw Hill.

Appendices

A Finite Difference

Finite difference were used for the uniform grid. These finite difference matrices can be approximated using a stencil. We can then represent these matrices (that act on the state vector) as shown in 14.

$$\begin{aligned}\text{uniform grid } \frac{\partial(\cdot)}{\partial y} &\approx \mathcal{D}_y \\ \frac{\partial^2(\cdot)}{\partial y^2} &\approx \mathcal{D}_{yy}\end{aligned} \tag{14}$$

However, when using a stretched grid, we need to account for that using the chain rule as shown in Eq. 15.

$$\begin{aligned}\text{non-uniform grid } \frac{\partial(\cdot)}{\partial y} &= \frac{\partial(\cdot)}{\partial \xi}\frac{\partial \xi}{\partial y} \text{ with: } \frac{\partial(\cdot)}{\partial \xi} \approx \mathcal{D}_\xi, \frac{\partial \xi}{\partial y} = \frac{1}{\frac{\partial y}{\partial \xi}} \approx \frac{1}{\mathcal{D}_\xi y} \\ &\boxed{\frac{\partial(\cdot)}{\partial y} \approx (\mathcal{D}_\xi(\cdot))\frac{1}{\mathcal{D}_\xi y}} \\ &\frac{\partial^2(\cdot)}{\partial y^2} = \frac{\partial^2(\cdot)}{\partial \xi^2}\left(\frac{\partial \xi}{\partial y}\right)^2 + \frac{\partial(\cdot)}{\partial \xi}\frac{\partial^2 \xi}{\partial y^2} \text{ with: } \frac{\partial^2 \xi}{\partial y^2} = -\frac{\partial^2 y}{\partial \xi^2}\left(\frac{\partial \xi}{\partial y}\right)^3 \approx -\mathcal{D}_{\xi\xi}\left(\frac{1}{\mathcal{D}_{\xi\xi} y}\right)^3 \\ &\boxed{\frac{\partial^2(\cdot)}{\partial y^2} \approx \mathcal{D}_{\xi\xi}(\cdot)\left(\frac{1}{\mathcal{D}_\xi y}\right)^2 + \mathcal{D}_\xi(\cdot)(-\mathcal{D}_{\xi\xi} y)\left(\frac{1}{\mathcal{D}_\xi y}\right)^3}\end{aligned} \tag{15}$$

B Stretched Grid in wall normal coordinate

The wall normal, y, coordinate used a stretched grid for the Blasius boundary layer flow. The stretched grid is defined in [Vinokur, 1983] and is shown in Eq. 16. Here, $s(y)$ is the stretched coordinate, y is the uniform grid, and δ is the stretching factor. As $\delta \to 0$ then we obtain the uniform grid.

$$\begin{aligned}s(y) &= \left(1 + \frac{\tanh[\delta(y/I - 1)]}{\tanh(\delta)}\right) \cdot I \\ y &\in [0, I]\end{aligned} \tag{16}$$

C Nonlinear PSE

The nonlinear parabolized stability equations are shown in Eq. 17. With all the coefficient matrices shown in Eq. 18. Implicit Euler is used for x derivative here as well.

$$\begin{aligned}\mathbf{A}\hat{\mathbf{q}}_m + \mathbf{B}\frac{\partial \hat{\mathbf{q}}_m}{\partial y} + \mathbf{C}\frac{\partial^2 \hat{\mathbf{q}}_m}{\partial y^2} + \mathbf{D}\frac{\partial \hat{\mathbf{q}}_m}{\partial x} + \mathbf{E}\frac{\partial \hat{\mathbf{q}}_m}{\partial z} + \mathbf{G}\frac{\partial^2 \hat{\mathbf{q}}_m}{\partial z^2} &= \hat{\mathbf{F}}_m \\ (\mathbf{A} + \mathbf{B}\mathcal{D}_y + \mathbf{C}\mathcal{D}_{yy} + \frac{\mathbf{D}}{h_x} + \mathbf{E}\mathcal{D}_z + \mathbf{G}\mathcal{D}_{zz})\hat{\mathbf{q}}(x = x_{i+1}) &= \frac{\mathbf{D}}{h_x}\hat{\mathbf{q}}(x = x_i) + \hat{\mathbf{F}}_m \\ \mathbf{A}_{\text{solve}} &= \mathbf{A} + \mathbf{B}\mathcal{D}_y + \mathbf{C}\mathcal{D}_{yy} + \mathbf{E}\mathcal{D}_z + \mathbf{G}\mathcal{D}_{zz} \\ \left(\mathbf{A}_{\text{solve}} + \frac{\mathbf{D}}{h_x}\right)_{i+1}\hat{\mathbf{q}}_{i+1} &= \mathbf{b}_i\end{aligned} \tag{17}$$

$$
\begin{aligned}
\mathbf{A} &= \begin{bmatrix} r + U_x & U_y & 0 & i\alpha \\ 0 & r + V_y & 0 & 0 \\ 0 & 0 & r & 0 \\ i\alpha & 0 & 0 & 0 \end{bmatrix} &
\mathbf{B} &= \begin{bmatrix} V & 0 & 0 & 0 \\ 0 & V & 0 & 1 \\ 0 & 0 & V & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\
\mathbf{C} &= \begin{bmatrix} \frac{-1}{Re} & 0 & 0 & 0 \\ 0 & \frac{-1}{Re} & 0 & 0 \\ 0 & 0 & \frac{-1}{Re} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} &
\mathbf{D} &= \begin{bmatrix} U & 0 & 0 & 1 \\ 0 & U & 0 & 0 \\ 0 & 0 & U & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\
\mathbf{E} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} &
\mathbf{G} &= \begin{bmatrix} \frac{-1}{Re} & 0 & 0 & 0 \\ 0 & \frac{-1}{Re} & 0 & 0 \\ 0 & 0 & \frac{-1}{Re} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\hat{\mathbf{F}}_m &= \begin{bmatrix} -\sum_{m=-M}^{M} (\hat{u}_m(\hat{u}_x + \hat{u} i\alpha) + \hat{v}_m \hat{u}_y + \hat{w}_m \hat{u}_z) e^{i\Theta_m} \\ -\sum_{m=-M}^{M} (\hat{u}_m(\hat{v}_x + \hat{v} i\alpha) + \hat{v}_m \hat{v}_y + \hat{w}_m \hat{v}_z) e^{i\Theta_m} \\ -\sum_{m=-M}^{M} (\hat{u}_m(\hat{w}_x + \hat{w} i\alpha) + \hat{v}_m \hat{w}_y + \hat{w}_m \hat{w}_z) e^{i\Theta_m} \\ 0 \end{bmatrix}
\end{aligned}
\tag{18}
$$

Where $r = -im\omega + i\alpha U + \frac{\alpha^2}{Re}$

D Numerical Code

The numerical algorithms used to solve this case is available to Stanford researchers on GitHub at `https://github.com/stanfordhpccenter/PSE`. This is a private repository, but the code documentation is available online at `http://stanford.edu/~srharris/PSE/`. This code is a work in progress, and is activley being updated at the time of this report.