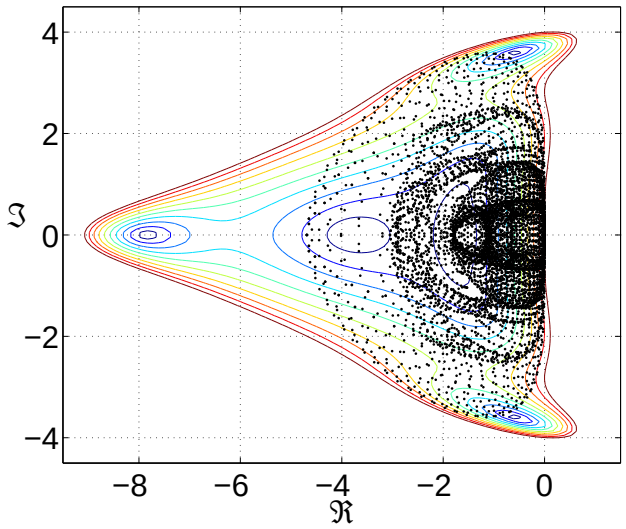
Squared Preconditioning and Optimized Multistaging for Explicit Multistage Flow Solvers



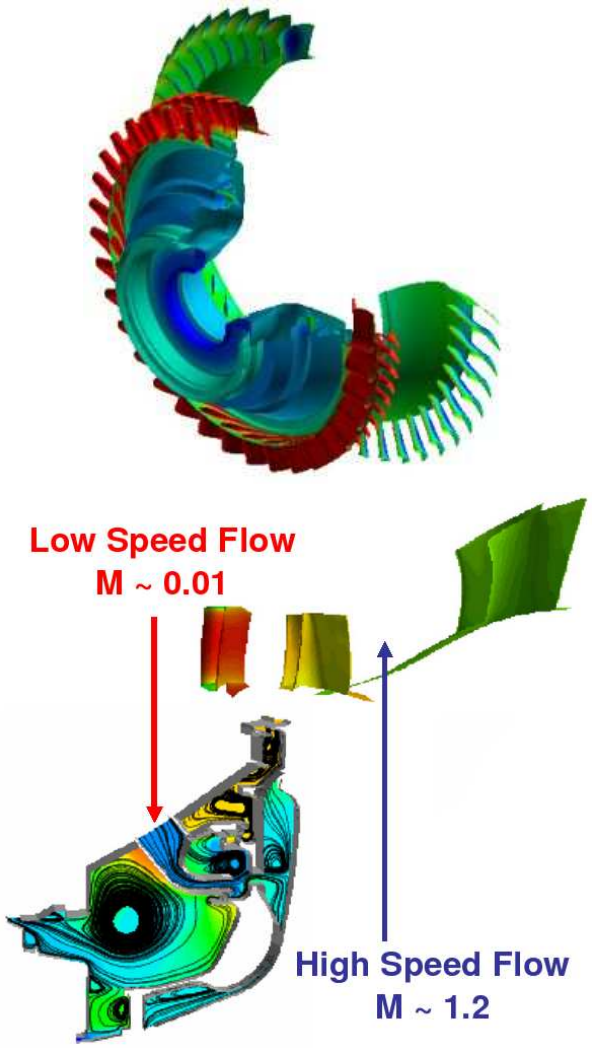
Kaveh Hosseini
Department of Aeronautics and Astronautics
Stanford University

California Institute of Technology
Pasadena, CA
September 28, 2004

Outline

- Motivation and introduction
- Preconditioning alternatives
- Robust and efficient implementation challenges
- 2D flow solver test-bed
- Parameter and Coefficient Optimization
- 3D parallel flow solver results
- Conclusions & Future Work

Motivation



- Explicit multistage multigrid flow solvers widely used in CFD
- Degraded convergence caused by numerical stiffness, directional decoupling, etc
- Growing need for compressible flow solvers to tackle full Mach range
- Typical flow solvers lose accuracy and convergence rate for low Mach numbers
- Preconditioning can address these problems to a great extent

Introduction

- 2 families of preconditioners: Block-Jacobi and low-speed
- Block-Jacobi: addresses principally discrete stiffness and is robust, but useless for low-Mach
- Low-speed: mainly improves condition number for all Mach and restores low-Mach accuracy, but is plagued by robustness issues especially for stagnation points
- Possibility to combine both types: preconditioning squared
- Proper and cost effective implementation not straightforward
- Full potential of preconditioning only achievable with fine-tuned dissipation scheme and adapted multistage scheme

Convergence characteristics of explicit multigrid schemes

- Advantages of explicit schemes
  - Low operation count and storage requirements
  - Good parallel scalability
- Convergence in pseudo-time by elimination of error modes using
  - Propagation (convection) out of the computational domain
  - Damping
- **Multigrid** and **multistage** techniques used together to achieve high damping and propagative efficiencies.
  - Multigrid transforms low freq. errors into high freq. errors by moving them to coarser meshes
  - Explicit multistage schemes are efficient in eliminating high freq. errors

Fourier analysis

- Multistage schemes are known for having extended **stability domains** compared to single stage schemes

- Fourier analysis can be used to plot the **amplification factor** associated with a given discretization stencil on the stability domain.

- The Fourier analysis of the **scalar 1D model wave equation** can be used as a powerful tool to illustrate convergence characteristics in terms of propagation and damping:

$$\begin{aligned} w_t &+ a w_x + \mu_a \Delta x^3 w_{xxxx} &= 0 \\ w_t &+ R(w) &= 0 \end{aligned} \tag{1}$$

Discrete scalar 1D wave equation

- Apply central-differencing to discretize the residual

$$\begin{aligned} \Delta t R_j &= \frac{\sigma}{2}\left(w_{j+1} - w_{j-1}\right) \\ &+ \sigma\mu_a\left(w_{j+2} - 4w_{j+1} + 6w_j - 4w_{j-1} + w_{j-2}\right), \end{aligned} \tag{2}$$

- Define the CFL number $\sigma = \frac{a\Delta t}{\Delta x} = \frac{\Delta t}{\Delta x}$.
- Consider a Fourier mode $\hat{w} = e^{ipx}$ with $x_j = j\Delta x$ and set $\xi = p\Delta x$
- Define Fourier **residual operator** z **and amplification factor** $g(z)$

$$\Delta t \frac{d\hat{w}}{dt} = z\hat{w}, \tag{3}$$

$$\hat{w}^{n+1} = g(z)\hat{w}^n. \tag{4}$$

Fourier residual operator and amplification factor

- For central-differencing and third order dissipation

$$z(\xi) = -\sigma i \sin\xi - 4\sigma\mu_a(1-\cos\xi)^2. \tag{5}$$

- If we use first order dissipation instead

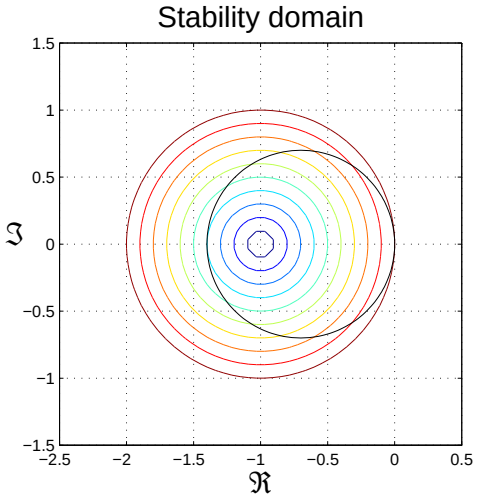
$$z(\xi) = -\sigma i \sin\xi - 2\sigma\mu_a(1-\cos\xi). \tag{6}$$

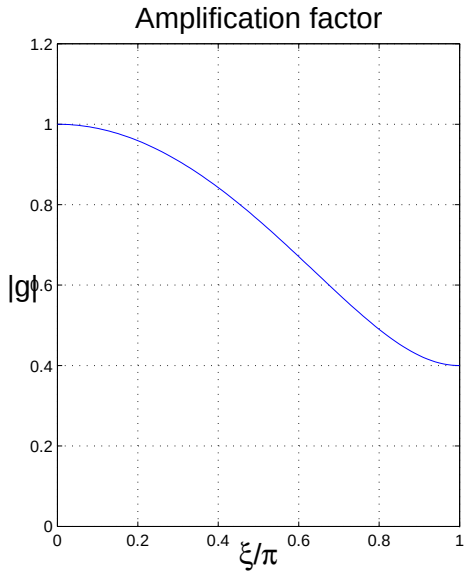
- For an m-stage **classical Runge-Kutta** multistage scheme the amplification factor is a complex number of the form

$$g(z) = 1 + z + \alpha_{m-1}z^2 + \ldots + \prod_{i=1}^{l}\alpha_{m-i+1}z^l + \ldots + \prod_{i=1}^{m}\alpha_{m-i+1}z^m. \tag{7}$$

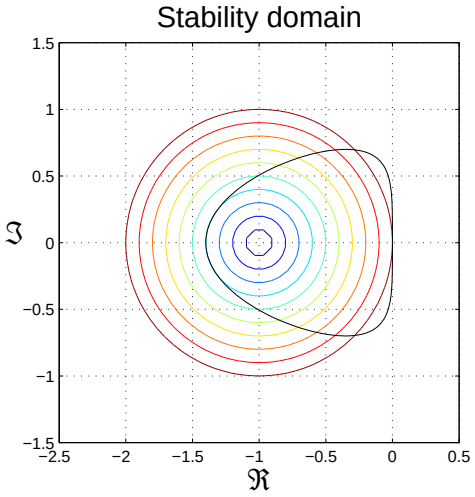
Single-stage scheme

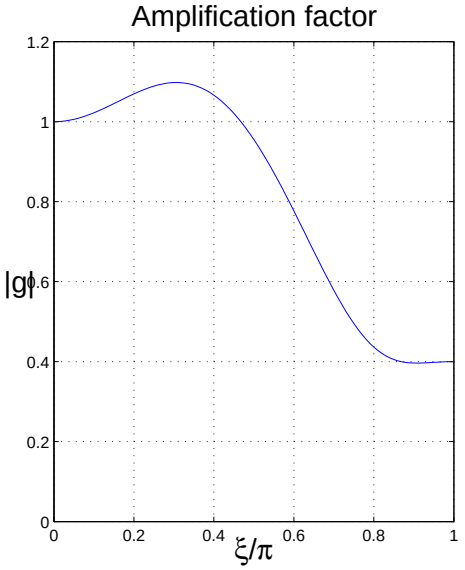
First order scheme with $\sigma = 0.7$ and $\mu_a = 0.5$





Third order scheme with $\sigma = 0.7$ and $\mu_a = 0.125$





Modified Runge-Kutta scheme

- The modified Runge-Kutta uses two sets of coefficients α_l and β_l resulting in larger stability domains than the classical Runge-Kutta.

- Computational savings achieved by taking every other β_l to be equal to zero.

$$
\begin{aligned}
W^{(n+1,0)} &= W^n \\
W^{(n+1,1)} &= W^{(0)} - \alpha_1 \Delta t R^{(0)} \\
&\vdots \\
W^{(n+1,l)} &= W^{(0)} - \alpha_l \Delta t R^{(l-1)} \\
&\vdots \\
W^{(n+1,m)} &= W^{(0)} - \quad \Delta t R^{(m-1)} \\
W^{n+1} &= W^{(n+1,m)}
\end{aligned}
$$

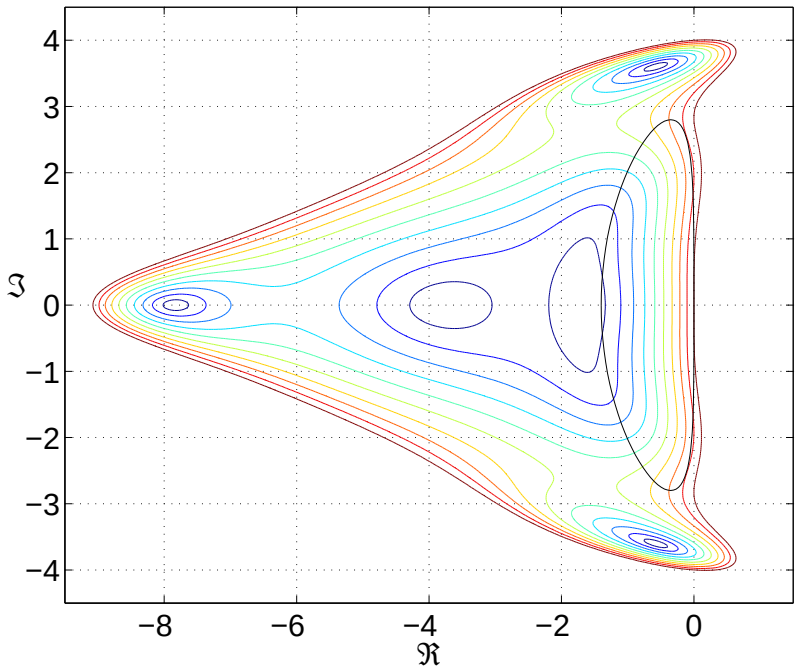
$$
\begin{aligned}
R^{(l)} &= Q^{(l)} + D^{(l)} \\
Q^{(0)} &= Q(W^n) \\
D^{(0)} &= D(W^n) \\
Q^{(l)} &= Q(W^{(n+1,l)}) \\
D^{(l)} &= \beta_l D(W^{(n+1,l)}) \\
&+ (1-\beta_l) D(W^{(n+1,l-1)})
\end{aligned}
$$

Martinelli-Jameson (MJ) Multistage Coefficients

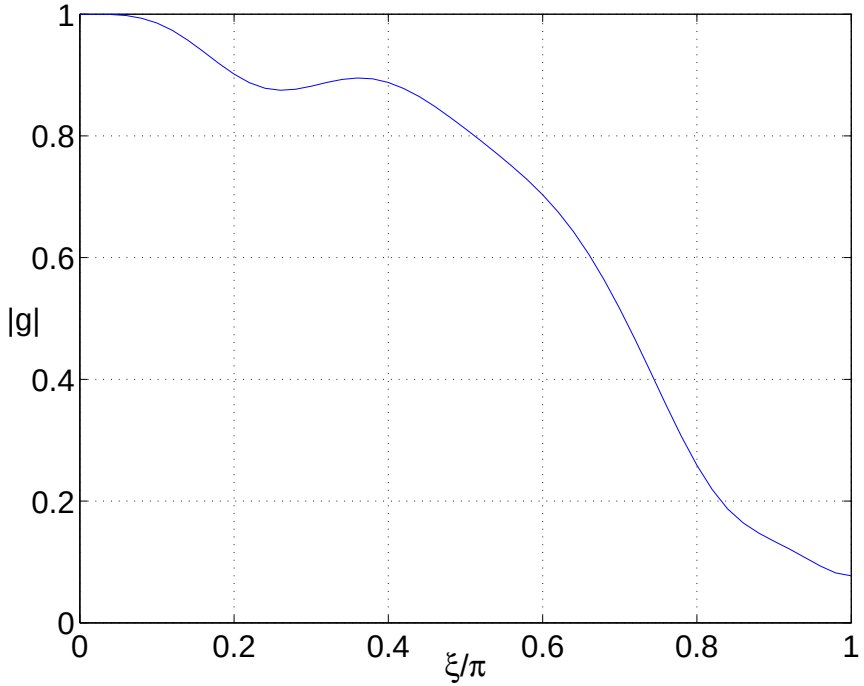
- Used as baseline/reference
- Developed via trial and error to have large stability regions in both the real and imaginary directions

$$
\begin{aligned}
\alpha_1 &= 1/4 & \beta_1 &= 1 \\
\alpha_2 &= 1/6 & \beta_2 &= 0 \\
\alpha_3 &= 3/8 & \beta_3 &= 14/25 \\
\alpha_4 &= 1/2 & \beta_4 &= 0 \\
\alpha_5 &= 1 & \beta_5 &= 11/25.
\end{aligned}
\tag{8}
$$

Stability domain for the MJ scheme



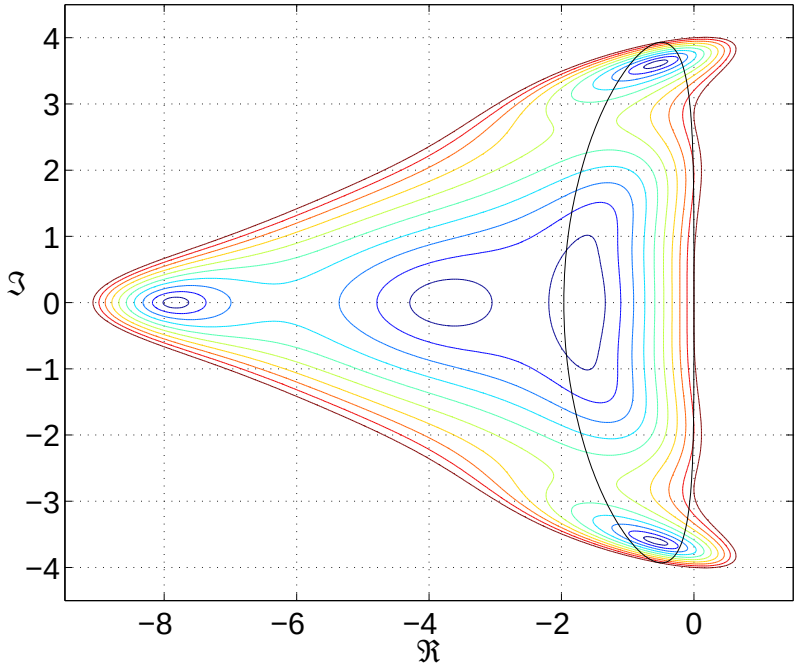
Stability domain and locus of z for $\sigma = 2.8$ and $\mu_a = \frac{1}{32}$



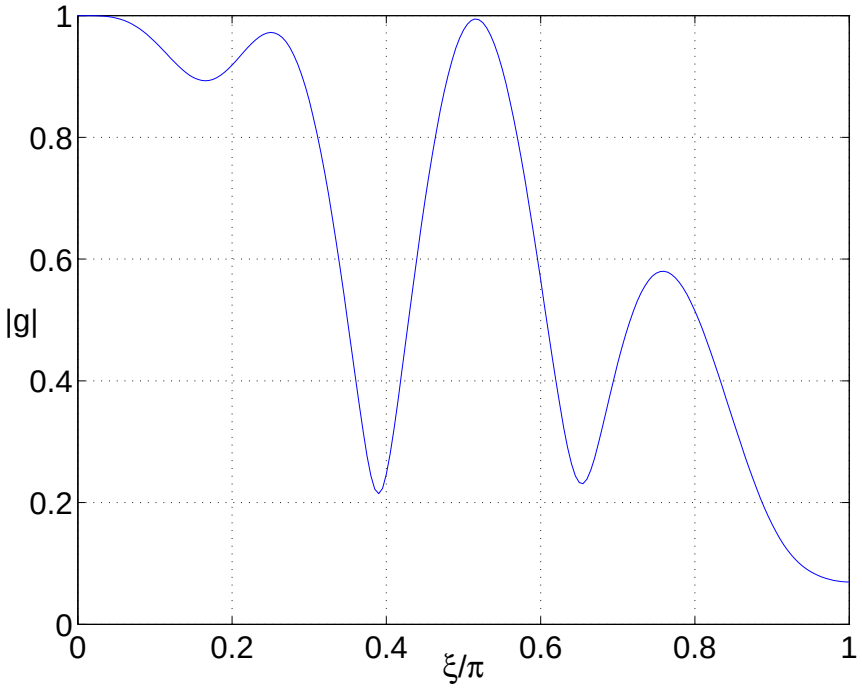
Amplification factor $|g|$ for $\sigma = 2.8$ and $\mu_a = \frac{1}{32}$

Extended stability domain allows larger values for σ. The dissipation coefficient $\mu_a = \mu_{a_0} = \frac{1}{32}$ is representative of a typical value for scalar dissipation.

Increasing σ



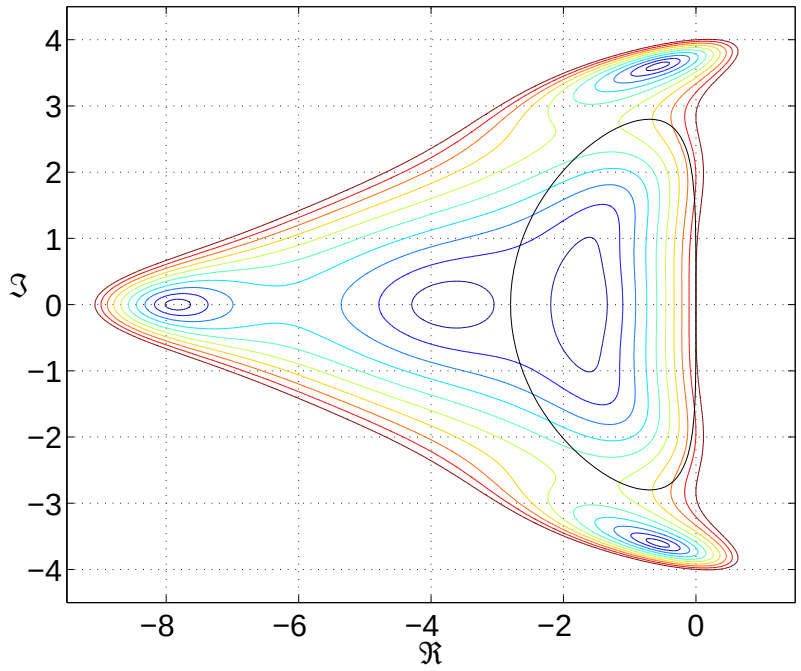
Stability domain and locus of z for $\sigma = 3.93$ and $\mu_a = \frac{1}{32}$



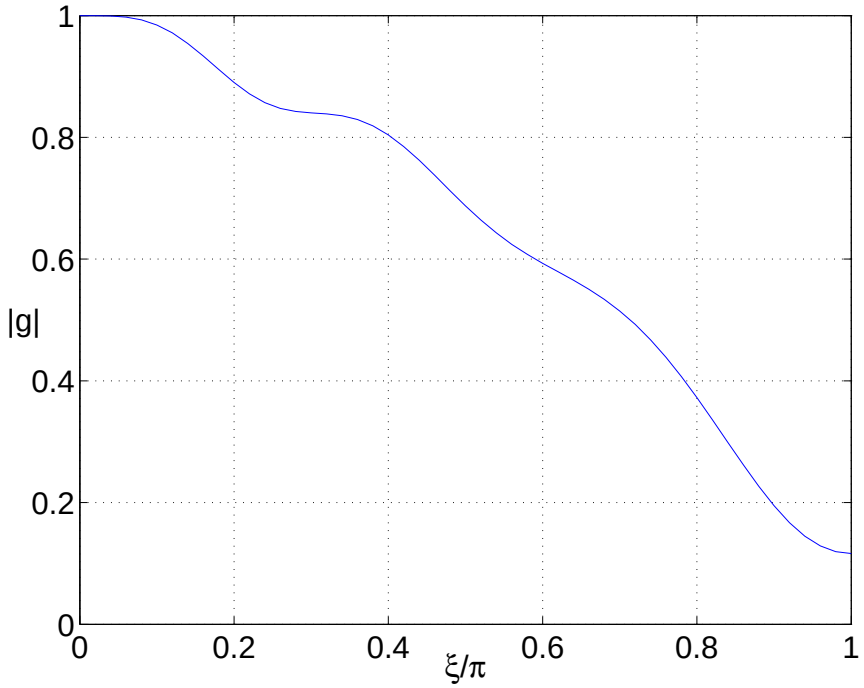
Amplification factor $|g|$ for $\sigma = 3.93$ and $\mu_a = \frac{1}{32}$

Stability domain and amplification factor for the MJ coefficients with maximum σ improves propagation.

Increasing μ_a



Stability domain and locus of z for $\sigma = 2.8$ and $\mu_a = \frac{1}{16}$



Amplification factor $|g|$ for $\sigma = 2.8$ and $\mu_a = \frac{1}{16}$

Larger values of μ_a improves damping.

Two-dimensional Euler equations

- Using third order **scalar** dissipation (more dissipative)

$$\frac{\partial W}{\partial t}+\left[A_x\frac{\partial W}{\partial x}+A_y\frac{\partial W}{\partial y}\right]-\mu_a\left[\Delta x^3\rho(A_x)I\frac{\partial^4 W}{\partial x^4}+\Delta y^3\rho(A_y)I\frac{\partial^4 W}{\partial y^4}\right]=0.$$

  where $\rho(A)$ is the spectral radius of A.

- Using third order **matrix** dissipation (less dissipative, requires larger μ_a)

$$\frac{\partial W}{\partial t}+\left[A_x\frac{\partial W}{\partial x}+A_y\frac{\partial W}{\partial y}\right]-\mu_a\left[\Delta x^3\left|A_x\right|\frac{\partial^4 W}{\partial x^4}+\Delta y^3\left|A_y\right|\frac{\partial^4 W}{\partial y^4}\right]=0.$$

  where $|A|=R_A\left|\Lambda_A\right|L_A$, $|\Lambda_A|$ is the diagonal matrix of the absolute values of the eigenvalues of A, and R_A and L_A are the matrices of the right and left eigenvectors of A.

Fourier analysis for the 2D Euler equations

- Consider a Fourier mode $\hat{W} = e^{ipx}e^{iry}$. Set $\xi = p\Delta x$ and $\eta = r\Delta y$

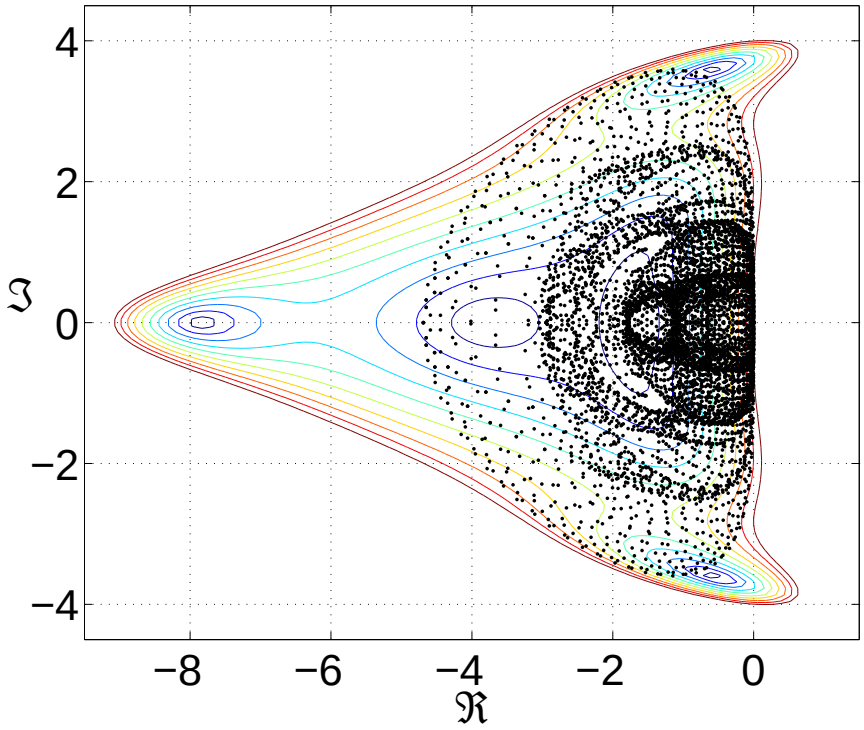
$$\Delta t \frac{d\hat{W}}{dt} = Z\hat{W}. \tag{9}$$

- Fourier matrix residual operator for third order matrix dissipation

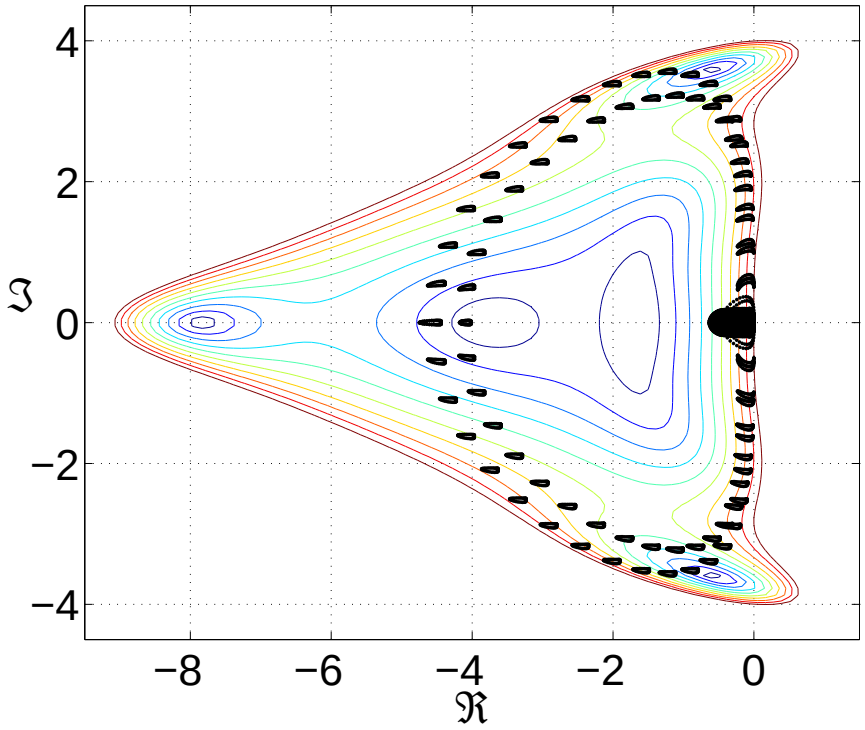
$$\begin{aligned} Z(\xi,\eta) = &-\frac{\Delta t}{\Delta x}\left[iA\sin\xi + 4\mu_a\left|A\right|\left(1-\cos\xi\right)^2\right] \\ &-\frac{\Delta t}{\Delta y}\left[iB\sin\eta + 4\mu_a\left|B\right|\left(1-\cos\eta\right)^2\right]. \end{aligned} \tag{10}$$

- The eigenvalues of Z will be uniquely defined as functions of ξ and η for a given computational cell where we know the Mach number $M = \frac{\sqrt{u^2+v^2}}{c}$,the flow angle $\varphi = \arctan\left(\frac{v}{u}\right)$, and the aspect ratio $\mathcal{AR} = \frac{\Delta x}{\Delta y}$.

Loci of the eigenvalues of Z on the MJ stability domain



$M = 0.7$, $\mathcal{R} = 1$, and $\varphi = 30°$

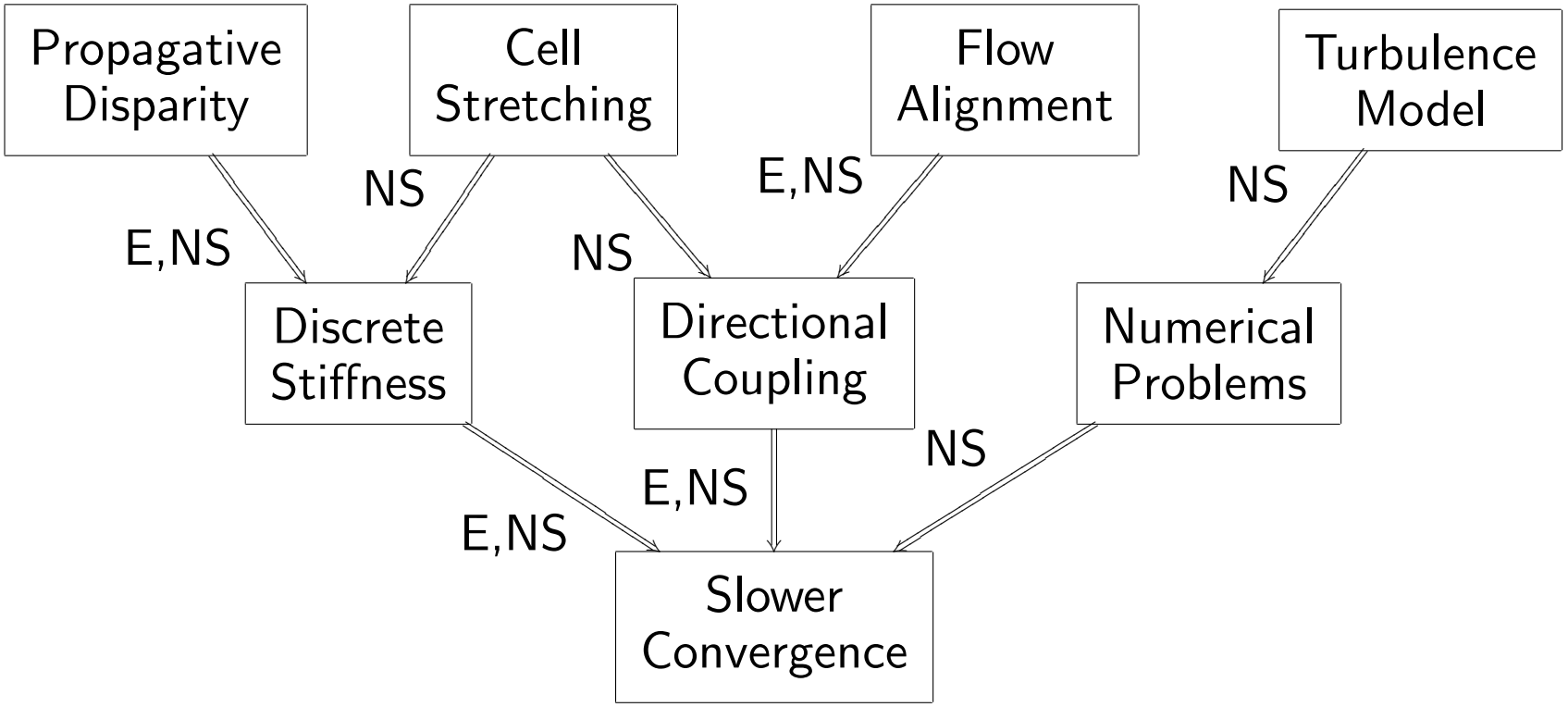


$M = 0.1$, $\mathcal{R} = 10$, and $\varphi = 30°$

Examples of stability domains for $\sigma = 3.6$ and $\mu_a = \frac{2.5}{32}$

The CFL number is now defined as $\sigma = \Delta t \left(\frac{u\Delta y + v\Delta x + c\sqrt{\Delta x^2 + \Delta y^2}}{\Delta x \Delta y} \right)$

Diagnosis of multigrid breakdown (Pierce 97)



- Pierce used Allmaras' Block-Jacobi preconditioner
- Best results obtained when combined with semi-coarsening
- Semi-coarsening not necessarily possible for complex geometries

Preconditioning principle

- Let us consider the Euler equations in their non-conservative form

$$\frac{\partial W}{\partial t} + A_x \frac{\partial W}{\partial x} + A_y \frac{\partial W}{\partial y} = 0, \tag{11}$$

  and replace them by

$$\frac{\partial W}{\partial t} + P \left(A_x \frac{\partial W}{\partial x} + A_y \frac{\partial W}{\partial y} \right) = 0, \tag{12}$$

  where P is the preconditioning matrix.

- When $\frac{\partial W}{\partial t} = 0$, eqs. 11 and 12 are obviously equivalent.

- For an appropriate choice of the matrix P, the iterative process can be altered in a way that offers a faster path to a zero residual.

Preconditioning alternatives

- Block-Jacobi preconditioning (Allmaras, Pierce, . . .) mostly based on the Fourier analysis of the **discrete** set of equations.

- Family of low-speed preconditioners (Van Leer, Turkel, . . .) mostly based on the analysis of the condition number of the **continuous** set of equations.

  – Improvement in convergence speed greatest in the low-Mach limit.
  – Additionally, accuracy of compressible flow solvers is preserved in the incompressible limit.

- Squared preconditioning (Turkel, Darmofal, . . .) combines and complements the advantages of both approaches.

Block-Jacobi Preconditioning

- Using traditional **local time-stepping** (rather than global time stepping) can be thought of as scalar preconditioning

$$\Delta t^{-1} = {P_S}^{-1} = \frac{1}{\sigma}\left(\frac{\rho\left(A_x\right)}{\Delta x} + \frac{\rho\left(A_y\right)}{\Delta y}\right), \tag{13}$$

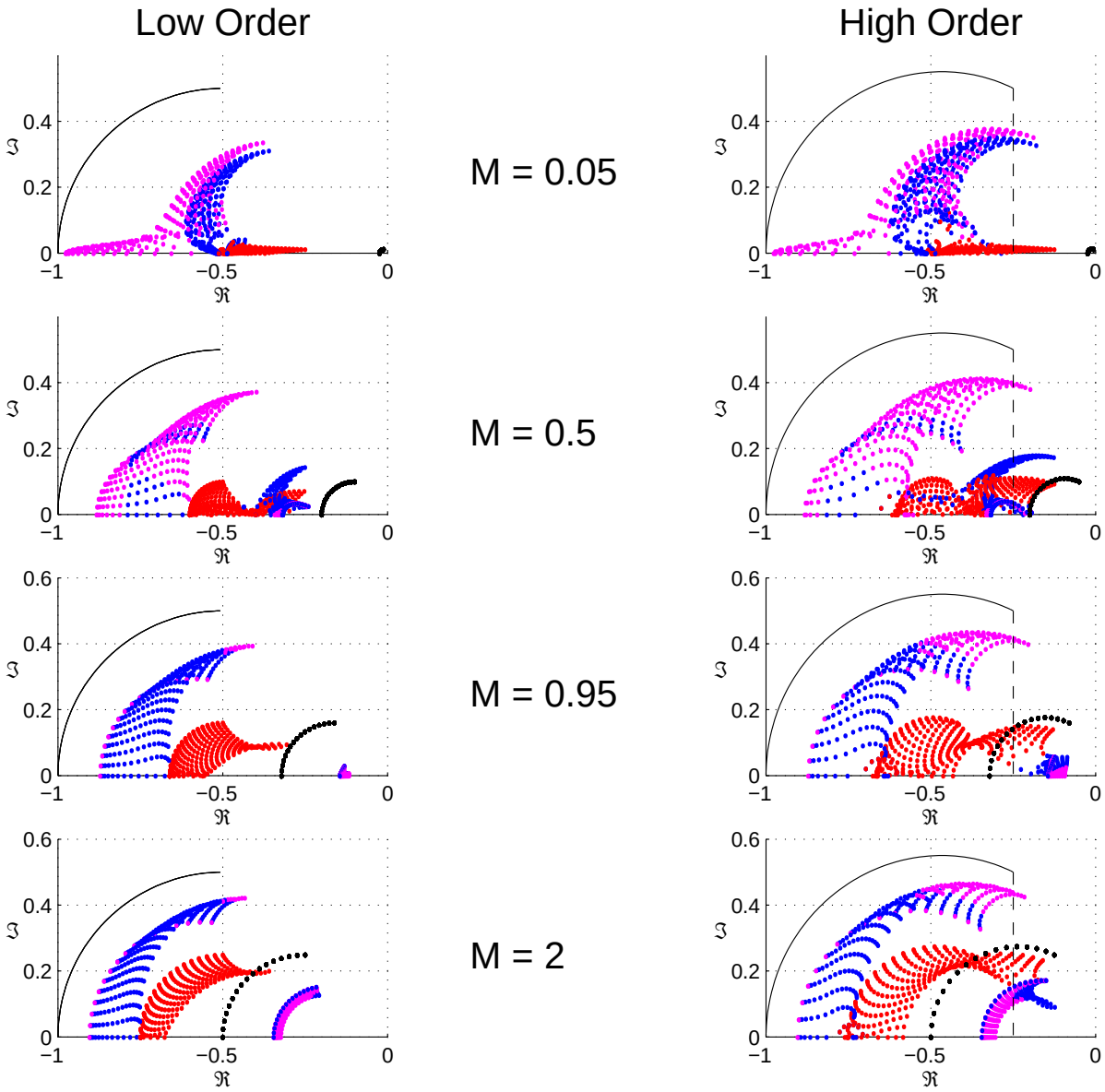
  where $\rho\left(A\right)$ is the spectral radius of A.

- The **Block-Jacobi** preconditioner, on the other hand, can be thought of as a matrix time-step defined as

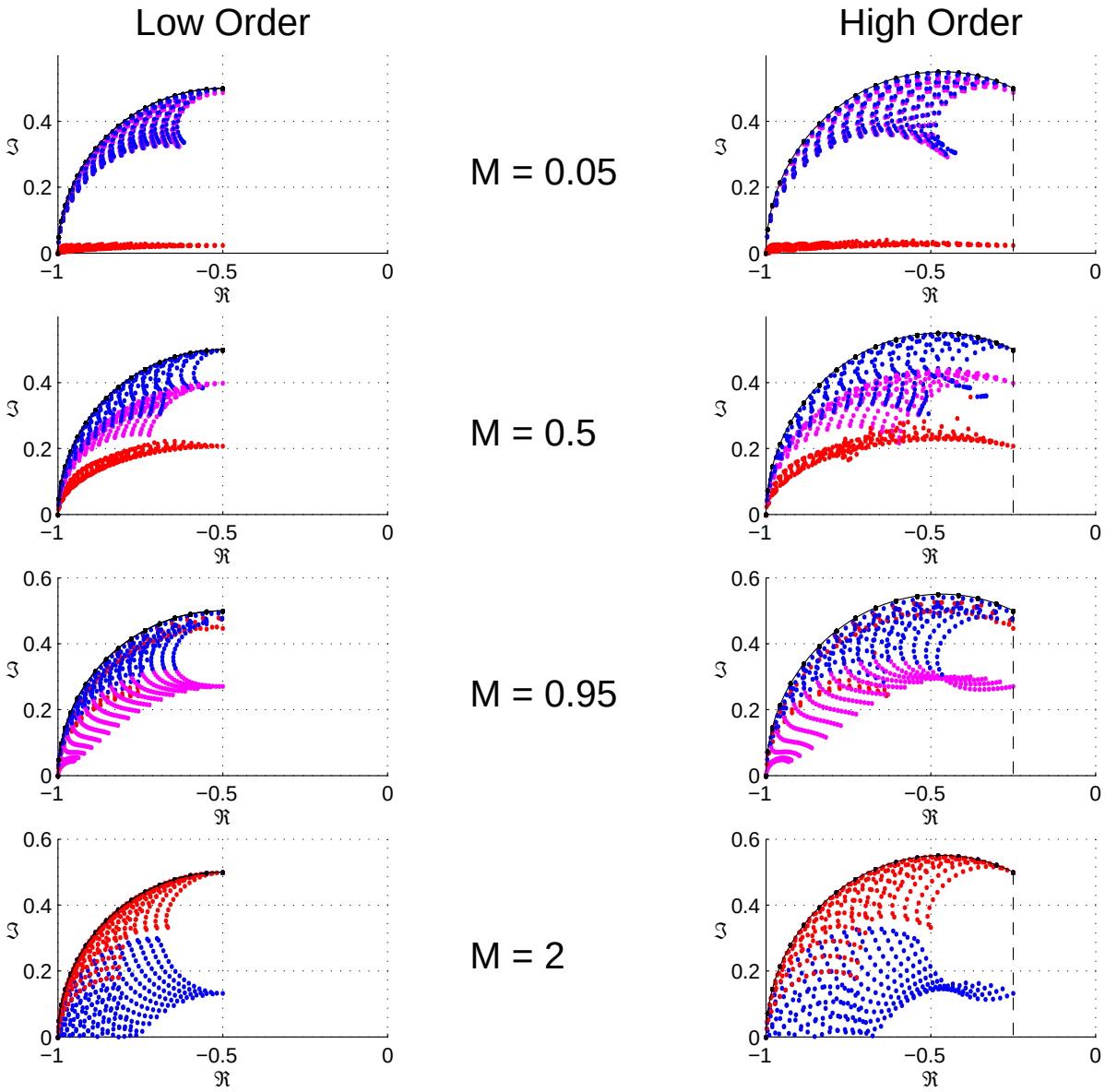
$$\left[\Delta t\right]^{-1} = {P_{BJ}}^{-1} = \frac{1}{\sigma}\left(\frac{\left|A_x\right|}{\Delta x} + \frac{\left|A_y\right|}{\Delta y}\right), \tag{14}$$

- Allmaras has produced single-stage results for un-preconditioned and preconditioned using first and third order matrix dissipations and maximum σ.

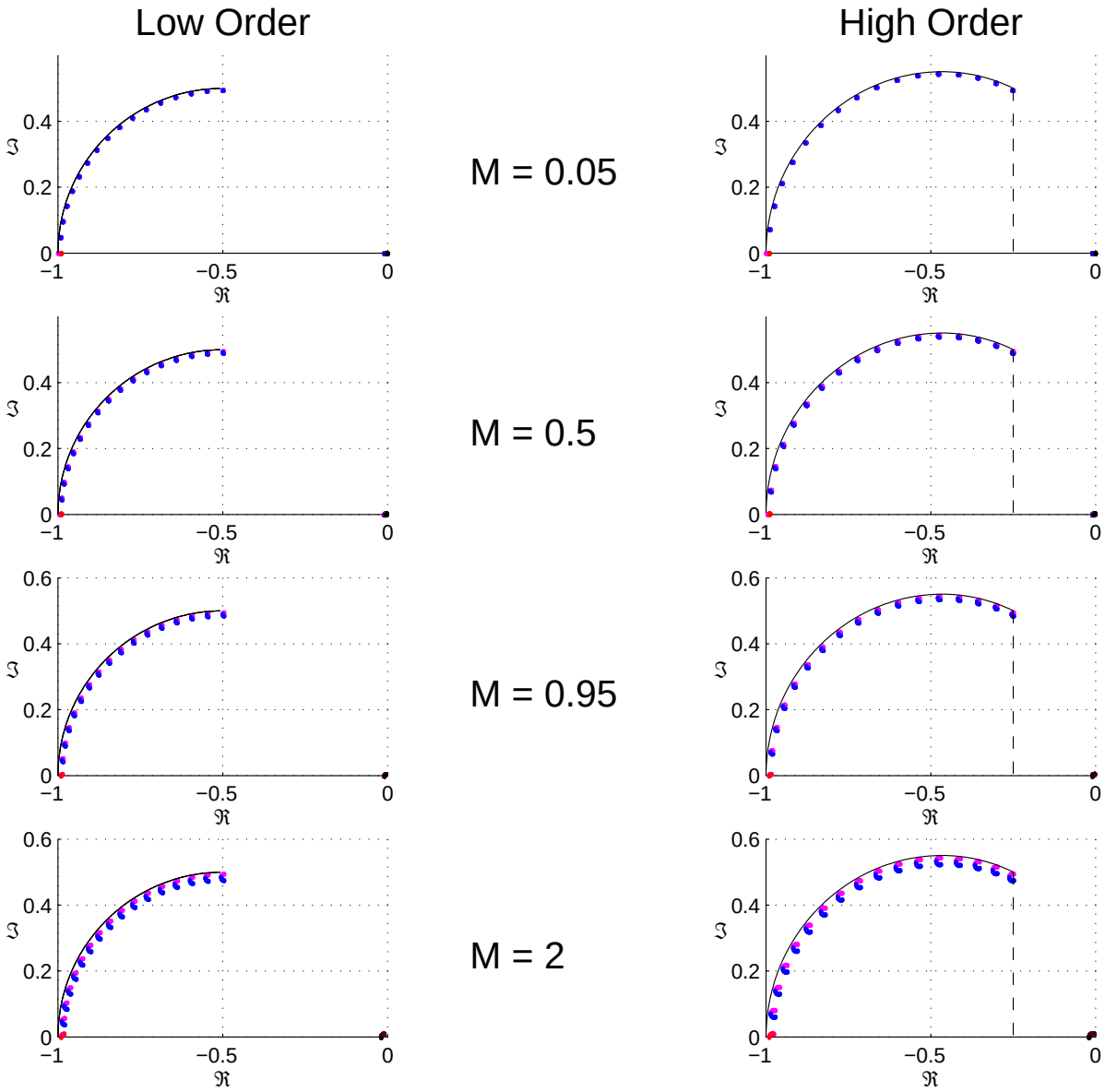
Un-preconditioned Euler for $\varphi = 0$ and $\mathscr{R} = 1$



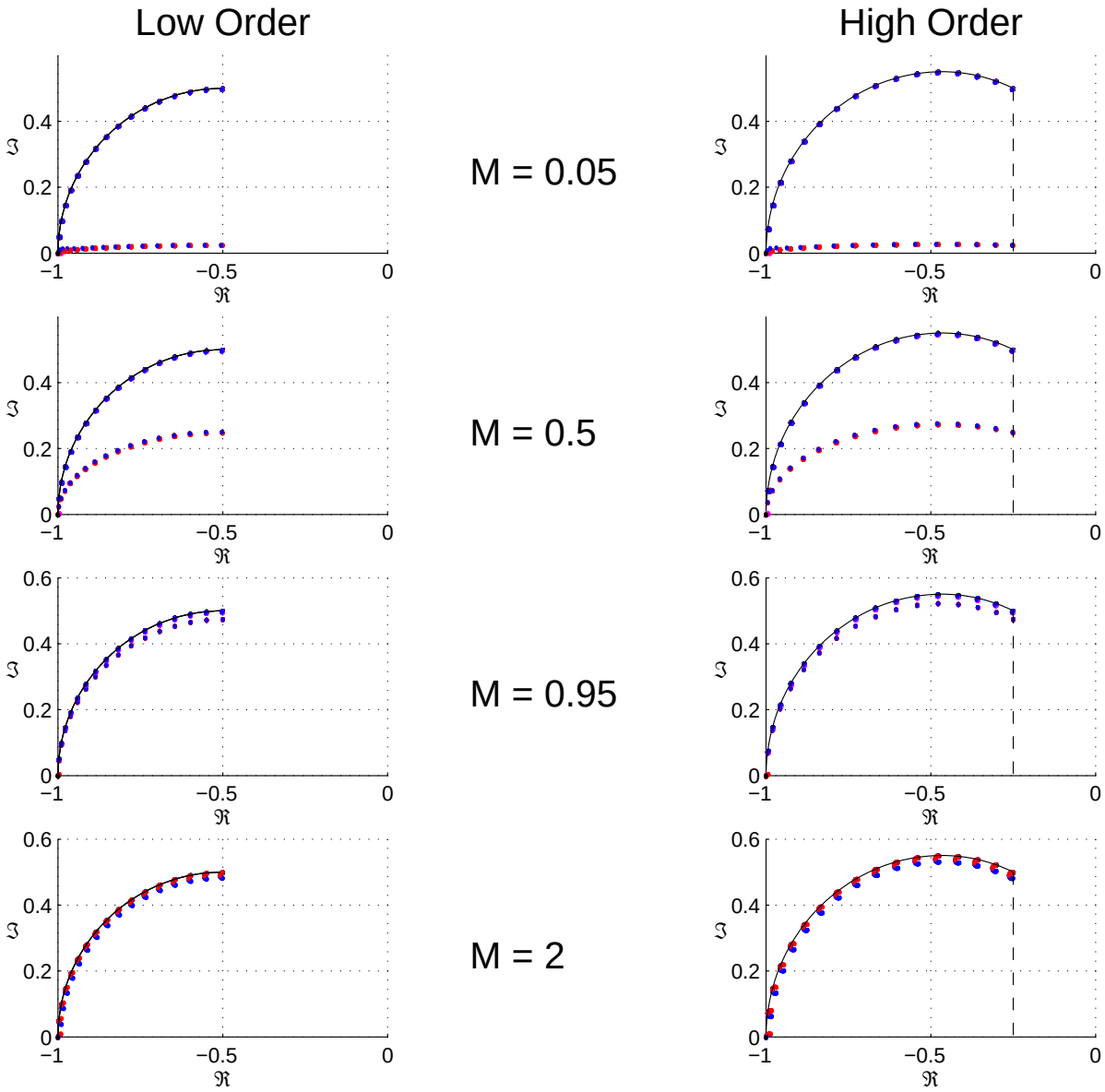
Block-Jacobi Euler for $\varphi = 0$ and $\mathscr{R} = 1$



Un-preconditioned Euler for $\varphi = 0$ and $\mathcal{R} = 100$



Block-Jacobi Euler for $\varphi = 0$ and $\mathscr{R} = 100$



Block-Jacobi benefits and shortcomings

- Fourier analysis shows that the Block-Jacobi preconditioner clusters the eigenvalues of the discrete system in regions of high damping and propagative efficiencies (Allmaras, 1993; Pierce, 1997).

- The Block-Jacobi preconditioner has also been shown to be numerically very **robust**.

- However, it cannot address the problems encountered in the limit of low Mach numbers
  - Convergence deteriorates as the Mach number diminishes due to a large **condition number** of the continuous system of equations.
  - **Accuracy** is lost due to a poor scaling of the artificial dissipation terms.

- There are several families of low-speed preconditioners (Choi-Merkle, Turkel, Van Leer-Lee-Roe, Weiss-Smith, . . .) that can address these problems to a great extent.

Implementation difficulties with conservative variables

- Conservative variables are popularly used in the implementation of CFD codes

$$W^c = \{ \ \rho, \quad \rho u, \quad \rho v, \quad \rho E \ \}^t. \tag{15}$$

- The convective flux Jacobians A_x^c and A_y^c in conservative variables have rather complicated expressions.

$$A_x^c = \begin{pmatrix} 0 & 1 & 0 & 0 \\ (\gamma-1)\frac{q^2}{2} - u^2 & (3-\gamma)u & (1-\gamma)v & \gamma - 1 \\ -uv & v & u & 0 \\ u[(\gamma-1)\frac{q^2}{2} - H] & H - (\gamma-1)u^2 & (1-\gamma)uv & \gamma u \end{pmatrix} \tag{16}$$

- Other complicated terms such as $|A_x|$ and $|A_y|$, and their modified forms through entropy fix $|A_x|^*$ and $|A_y|^*$ need to be computed for preconditioners and matrix dissipation terms.

Simplifications through entropy variables

- For the development and implementation of preconditioners it is advantageous to use entropy variables

$$dW^e = \left\{ \begin{array}{cccc} \frac{dp}{\rho c}, & du, & dv, & dp - c^2 d\rho \end{array} \right\}^t . \tag{17}$$

- The flux Jacobians in entropy variables can be obtained using the transformation

$$A^e = T^{ec} A^c T^{ce}, \tag{18}$$

where T^{ec} and T^{ce} are the transformation matrices between the conservative and entropy variables to obtain the symmetric flux Jacobians A_x^e and A_y^e with many zero terms

$$A_x^e = \begin{pmatrix} u & c & 0 & 0 \\ c & u & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix}, A_y^e = \begin{pmatrix} v & 0 & c & 0 \\ 0 & v & 0 & 0 \\ c & 0 & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix}. \tag{19}$$

Van Leer-Lee-Roe Preconditioner

- The VLR preconditioner (van Leer-Lee-Roe, 1991) is a symmetric preconditioner, sometimes referred to as optimal preconditioner.

- Its entropy variable form in a **flow-aligned** coordinate system is given by

$$P^e_{VLR} = \begin{pmatrix} \frac{\tau}{\beta^2}M^2 & \frac{\tau}{\beta^2}M & 0 & 0 \\ \frac{\tau}{\beta^2}M & \frac{\tau}{\beta^2}+1 & 0 & 0 \\ 0 & 0 & \tau & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{20}$$

with

$$\beta = \sqrt{|1-M^2|}, \tau = \min\left(\beta, \frac{\beta}{M}\right). \tag{21}$$

- Although theoretically it produces an optimal reduction in the condition number for all Mach numbers, in practice it is not robust (especially at stagnation points) mostly because it requires a well-defined flow angle.

Turkel Preconditioner

- The Turkel preconditioner is not theoretically as optimal as the VLR preconditioner.

- It also suffers from stagnation point related problems, but it is not sensitive to flow angle problems and, in that sense, it is generally considered more robust than the VLR preconditioner.

- Its entropy variable form is

$$P_T^e = \begin{pmatrix} \beta_T^2 & 0 & 0 & 0 \\ -\alpha_T \frac{u}{c} & 1 & 0 & 0 \\ -\alpha_T \frac{v}{c} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{22}$$

  where α_T and β_T are free parameters.

Turkel Preconditioner in flow aligned coordinates

- The entropy variable form of the Turkel preconditioner, expressed in flow-aligned coordinates further reduces further to

$$P_T^e = \begin{pmatrix} \beta_T^2 & 0 & 0 & 0 \\ -\alpha_T M & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{23}$$

- For optimal condition number reduction in subsonic flow we must have

$$\beta_T^2 = \frac{M^2}{1-M^2}, \alpha_T = 1 + \beta_T^2. \tag{24}$$

Squared Preconditioner

- It is possible to combine the advantages of the Block-Jacobi preconditioner with those of a low-speed preconditioner in a method introduced by Turkel (1997) as preconditioning-squared.

- One very robust implementation of such a method was proposed by Darmofal (1999) where the stagnation point problem has been addressed very efficiently.

- The low-speed preconditioner chosen for this task was a Weiss-Smith preconditioner which belongs to the Turkel family of preconditioners.

- The squared preconditioner proposed by Darmofal has essentially the following form

$$P_{SQ}{}^{-1} = \frac{P_{WS}{}^{-1}}{\sigma}\left(\frac{|P_{WS}A_x|}{\Delta x} + \frac{|P_{WS}A_y|}{\Delta y}\right), \tag{25}$$

Darmofal's limiter for the Weiss-Smith preconditioner

- The entropy variables form of P_{WS} is

$$P^e_{WS} = \begin{pmatrix} \epsilon & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{26}$$

- Darmofal proposed to express ϵ such that

$$\epsilon = \left\{ \begin{array}{ll} \frac{M^2}{1-3M^2} & M < 0.5 \\ 1 & M \geq 0.5 \end{array} \right\}. \tag{27}$$

- Furthermore he proposes to prevent ϵ from going to zero using a pressure based cutoff

$$\epsilon_{\text{lim}} = \frac{\bar{p}_{\text{Right}} - \bar{p}_{\text{Left}}}{\hat{\rho}\hat{c}^2}. \tag{28}$$

From un-preconditioned to Block-Jacobi

Un-preconditioned Euler equations

$$W^{n+1} - W^n + \left[\frac{1}{\sigma}\left(\frac{\rho\left(A_x\right)}{\Delta x} + \frac{\rho\left(A_y\right)}{\Delta y}\right)\right]^{-1}\left[A_x\frac{\partial W}{\partial x} + A_y\frac{\partial W}{\partial y} - \mu_a\left(\Delta x^3\left|A_x\right|\frac{\partial^4 W}{\partial x^4} + \Delta y^3\left|A_y\right|\frac{\partial^4 W}{\partial y^4}\right)\right] = 0. \quad (29)$$

Block-Jacobi preconditioner

$$W^{n+1} - W^n + \left[\frac{1}{\sigma}\left(\frac{|A_x|}{\Delta x} + \frac{|A_y|}{\Delta y}\right)\right]^{-1}\left[A_x\frac{\partial W}{\partial x} + A_y\frac{\partial W}{\partial y} - \mu_a\left(\Delta x^3\left|A_x\right|\frac{\partial^4 W}{\partial x^4} + \Delta y^3\left|A_y\right|\frac{\partial^4 W}{\partial y^4}\right)\right] = 0. \quad (30)$$

From low-speed to squared preconditioning

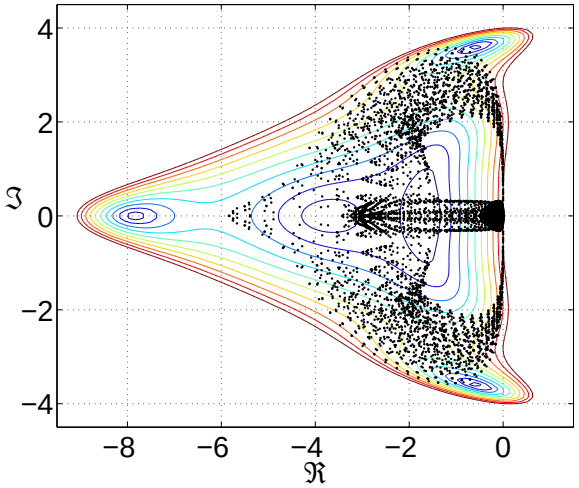
Any low-speed preconditioner P

$$W^{n+1} - W^n + P\left[A_x\frac{\partial W}{\partial x} + A_y\frac{\partial W}{\partial y} - \mu_a\left(\Delta x^3 P^{-1}\left|PA_x\right|\frac{\partial^4 W}{\partial x^4} + \Delta y^3 P^{-1}\left|PA_y\right|\frac{\partial^4 W}{\partial y^4}\right)\right] = 0. \quad (31)$$

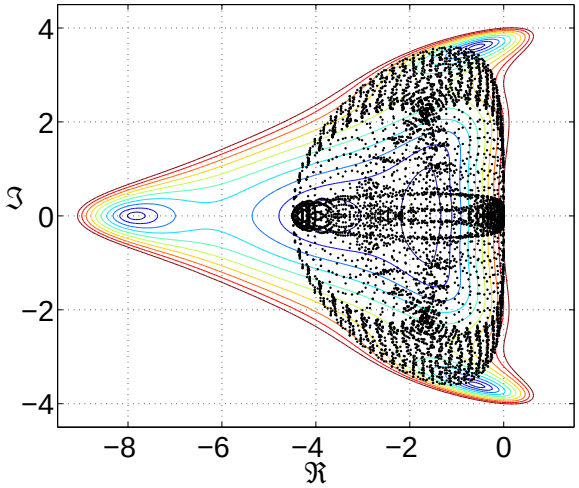
Squared preconditioners

$$W^{n+1} - W^n + \left[\frac{P^{-1}}{\sigma}\left(\frac{|PA_x|}{\Delta x} + \frac{|PA_y|}{\Delta y}\right)\right]^{-1}\left[A_x\frac{\partial W}{\partial x} + A_y\frac{\partial W}{\partial y} - \mu_a\left(\Delta x^3 P^{-1}\left|PA_x\right|\frac{\partial^4 W}{\partial x^4} + \Delta y^3 P^{-1}\left|PA_y\right|\frac{\partial^4 W}{\partial y^4}\right)\right] = 0. \quad (32)$$

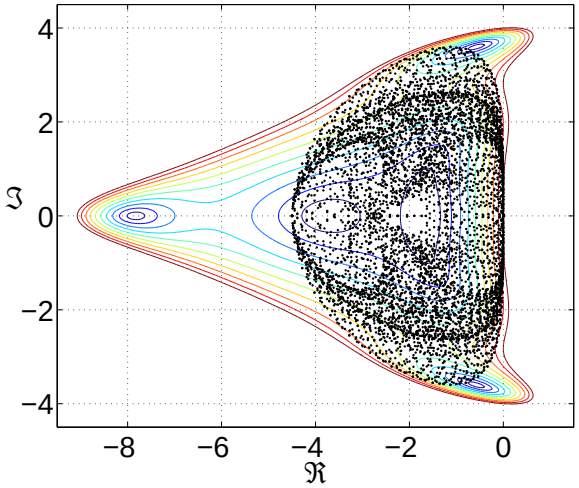
Stability Domains



No preconditioning



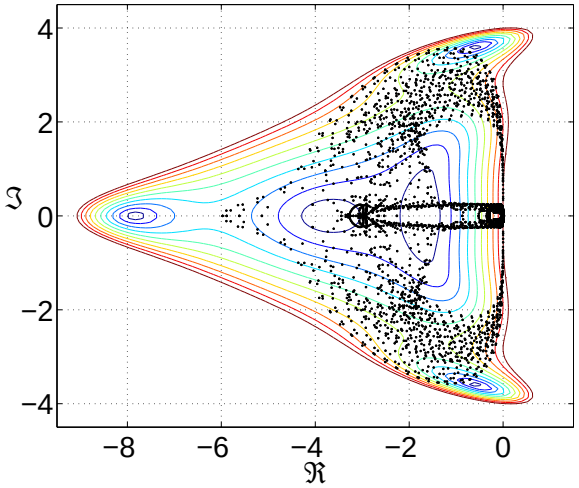
Block-Jacobi preconditioning



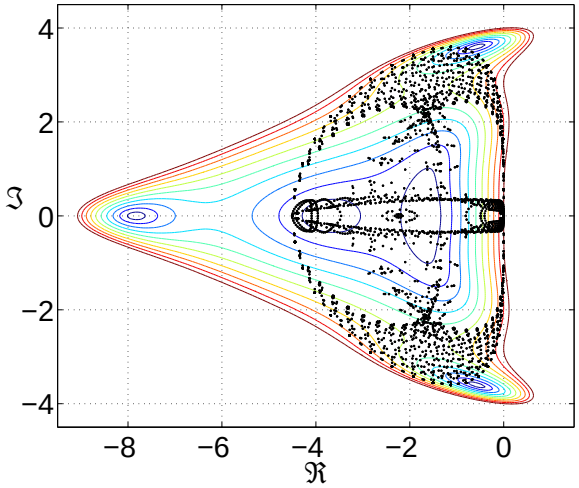
Squared preconditioning

Stability domains for $\sigma = 3.6$, $\mu = \frac{2.5}{32}$, $M = 0.1$, $\mathcal{R} = 1$, **and** $\varphi = 30^\circ$

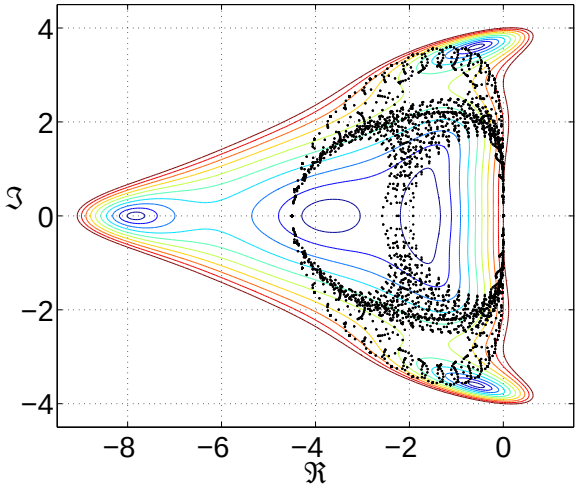
Stability Domains



No preconditioning



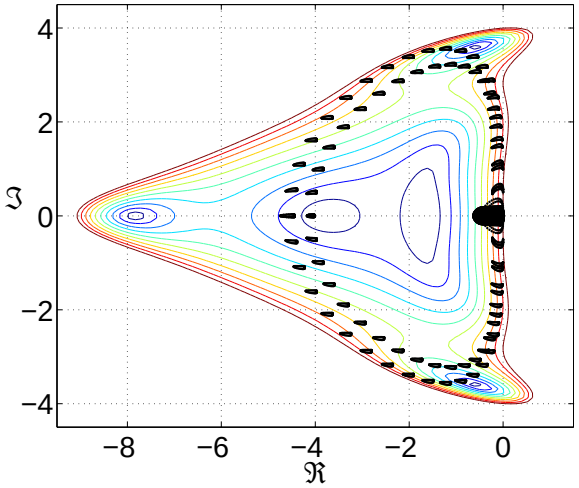
Block-Jacobi preconditioning



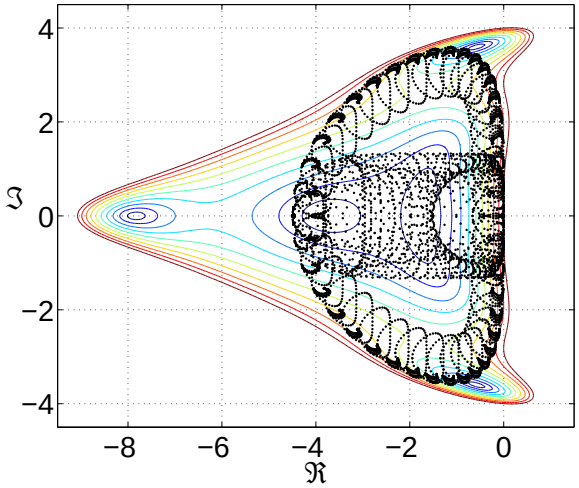
Squared preconditioning

Stability domains for $\sigma = 3.6$, $\mu = \frac{2.5}{32}$, $M = 0.1$, $\mathit{R} = 1$, **and** $\varphi = 0°$

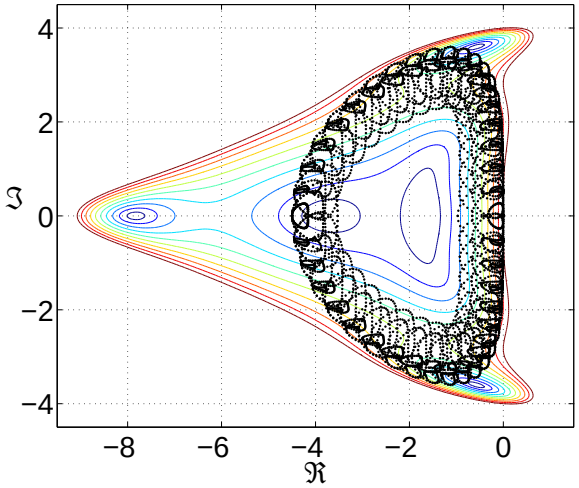
Stability Domains



No preconditioning



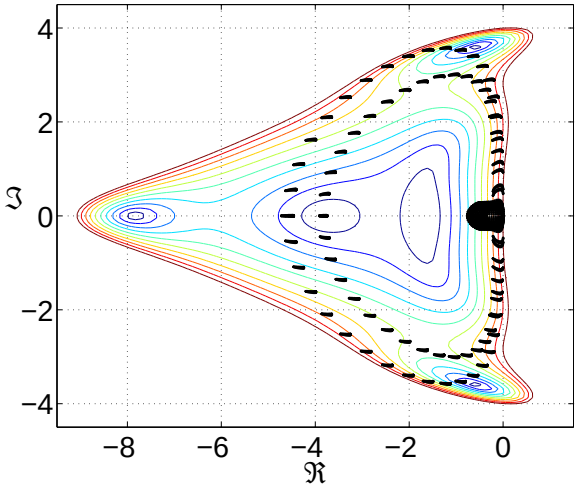
Block-Jacobi preconditioning



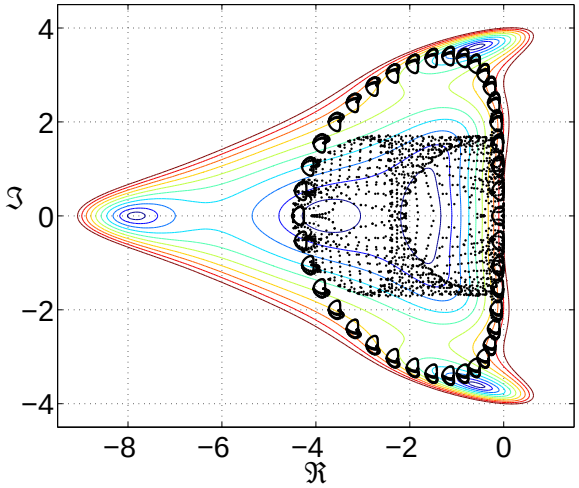
Squared preconditioning

Stability domains for $\sigma = 3.6$, $\mu = \frac{2.5}{32}$, $M = 0.1$, $\mathit{R} = 10$, **and** $\varphi = 30°$

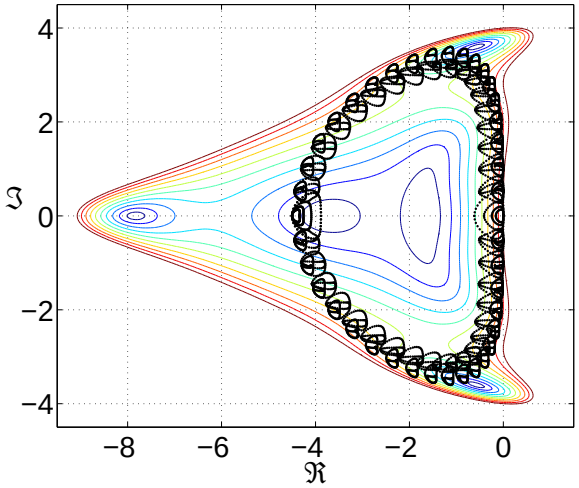
Stability Domains



No preconditioning



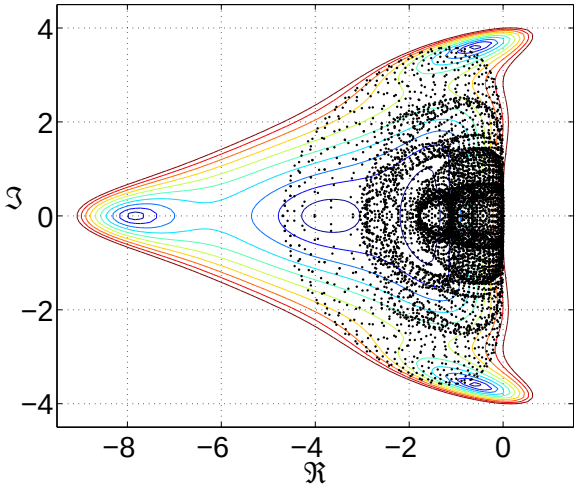
Block-Jacobi preconditioning



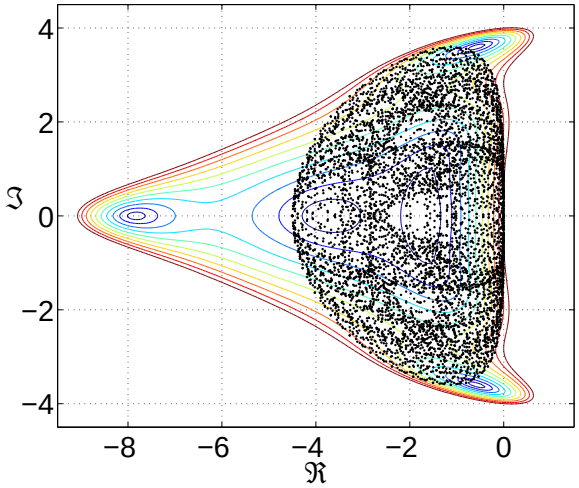
Squared preconditioning

Stability domains for $\sigma = 3.6$, $\mu = \frac{2.5}{32}$, $M = 0.1$, $\mathit{Æ} = 0.1$, **and** $\varphi = 30°$

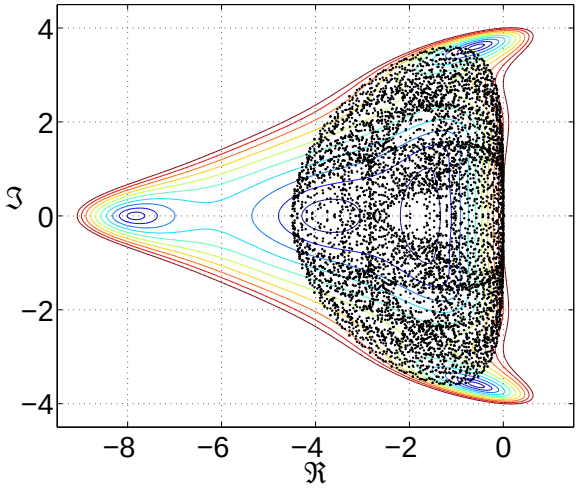
Stability Domains



No preconditioning



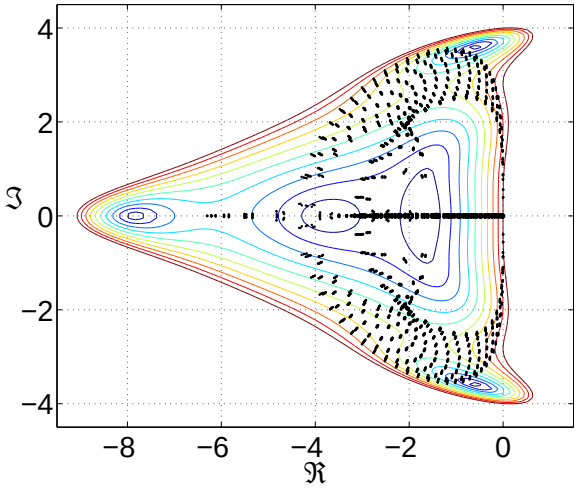
Block-Jacobi preconditioning



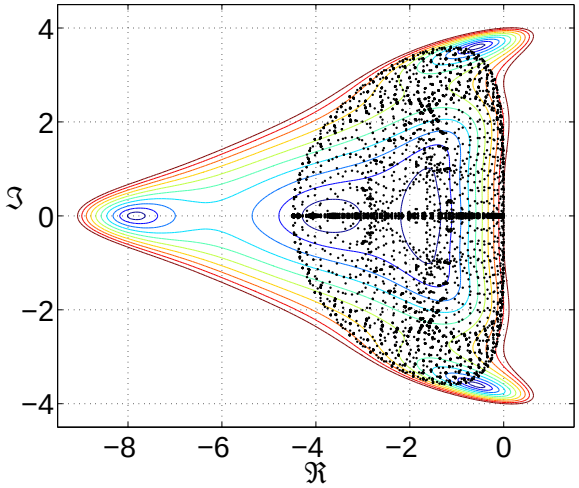
Squared preconditioning

Stability domains for $\sigma = 3.6$, $\mu = \frac{2.5}{32}$, $M = 0.7$, $\mathcal{R} = 1$, **and** $\varphi = 30^\circ$

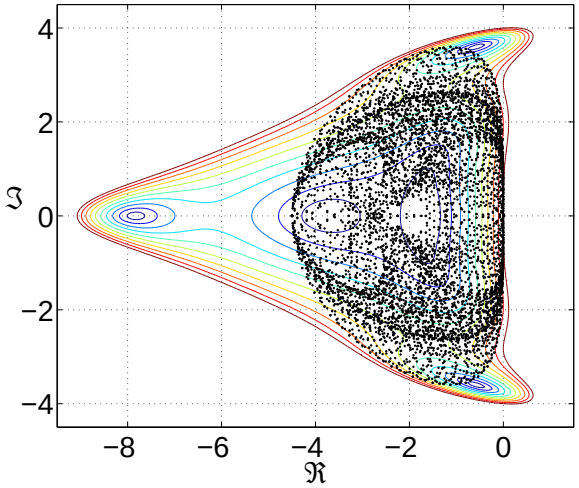
Stability Domains



No preconditioning



Block-Jacobi preconditioning



Squared preconditioning

Stability domains for $\sigma = 3.6$**,** $\mu = \frac{2.5}{32}$**,** $M = 0.01$**,** $\mathcal{R} = 1$**, and** $\varphi = 30\circ$

2D test-bed: FLO103

- Two-dimensional Euler and Navier-Stokes solver by Jameson and Martinelli (1987-1988).
- Uses multigrid, local time stepping, residual smoothing, etc. for convergence acceleration.
- The original FLO103 uses a Jameson-Schmidt-Turkel (JST) switched scalar artificial dissipation scheme. We have modified it to use matrix dissipation versions of JST for compatibility with Block-Jacobi and squared preconditioning.
- We intentionally turned off residual smoothing in order to isolate the effects of preconditioners, dissipation schemes, and multistage coefficients on convergence.

The problem of entropy Fix

- The eigenvalues of $|PA|$ must be prevented from going to zero for two reasons
  - The inversion of the preconditioning matrix requires the avoidance of singular matrices
  - Artificial dissipation terms must not become too small in order to insure stability.

- Therefore we replace

$$|PA| = R_{PA} \, |\Lambda_{PA}| \, L_{PA}. \tag{33}$$

by

$$|PA|^* = R_{PA} \, |\Lambda_{PA}|^* \, L_{PA}. \tag{34}$$

where the $|\lambda_i|$ diagonal terms have been replaced by cut off $|\lambda_i|^*$ terms.

Implementing an efficient entropy Fix

- Darmofal, Pierce and many others suggest

$$|\lambda_i|^* = \left\{ \begin{array}{ll} \Delta\lambda_i, & |\lambda_i| < \Delta\lambda_i \\ |\lambda_i|, & |\lambda_i| \geq \Delta\lambda_i \end{array} \right\}, \tag{35}$$

where

$$\Delta\lambda_i = 2\left|\lambda_{i_{\text{Left}}} - \lambda_{i_{\text{Right}}}\right|. \tag{36}$$

- We observed substantial stability and convergence improvements using the alternative forms in practice

$$|\lambda_i|^* = \left\{ \begin{array}{ll} \frac{1}{2}(\delta c + \frac{\lambda^2}{\delta c}), & |\lambda_i| < \delta c \\ |\lambda_i|, & |\lambda_i| \geq \delta c \end{array} \right\}, \tag{37}$$

and

$$|\lambda_i|^* = \left\{ \begin{array}{ll} \frac{1}{2}(\delta c + \frac{\lambda^2}{\delta c}), & |\lambda_i| < \delta \max(|\lambda_i|) \\ |\lambda_i|, & |\lambda_i| \geq \delta \max(|\lambda_i|) \end{array} \right\}, \tag{38}$$

Implementation of preconditioning and diffusive terms

If we choose $P = P_{WS}$, it is possible to write a compact analytic and symmetric form for the matrix $P^{e-1}\left|P^e A^e\right|^*$ that is used in both preconditioning and diffusive terms

$$P^{e-1}\left|P^e A^e\right|^* = \begin{pmatrix} \frac{c(s+|\lambda_2|^*)-q_n r(1-\epsilon)}{c\epsilon} & rn_x & rn_y & 0 \\ rn_x & sn_x^2+|\lambda_2|^* & sn_x n_y & 0 \\ rn_y & sn_x n_y & sn_y^2+|\lambda_2|^* & 0 \\ 0 & 0 & 0 & |\lambda_2|^* \end{pmatrix}. \tag{39}$$

with

$$q_n = un_x + vn_y \qquad \tau = \sqrt{(1-\epsilon)^2 q_n^2 + 4\epsilon c^2}$$

$$\lambda_1 = \frac{1}{2}\left[(1+\epsilon)q_n - \tau\right] \qquad \lambda_2 = \lambda_3 = q_n \qquad \lambda_4 = \frac{1}{2}\left[(1+\epsilon)q_n + \tau\right]$$

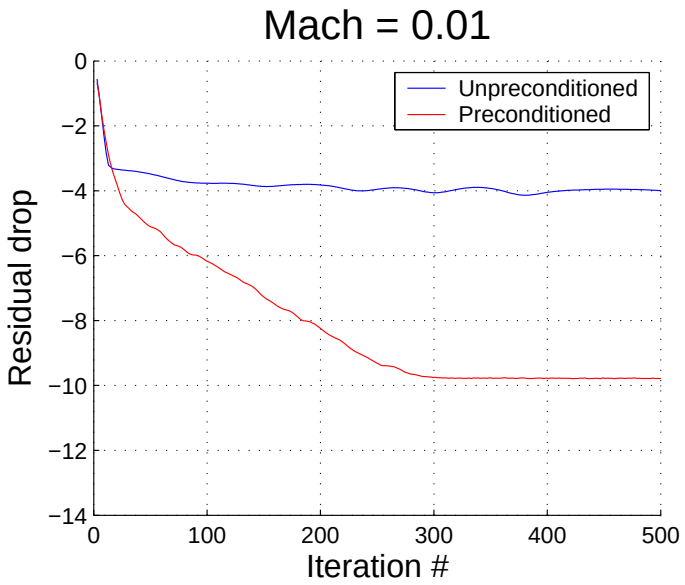
$$r = \frac{c\left(|\lambda_4|^* - |\lambda_1|^*\right)}{\tau} \qquad s = \frac{|\lambda_1|^* - 2|\lambda_2|^* + |\lambda_4|^*}{2} + \frac{q_n(1-\epsilon)r}{2c}$$

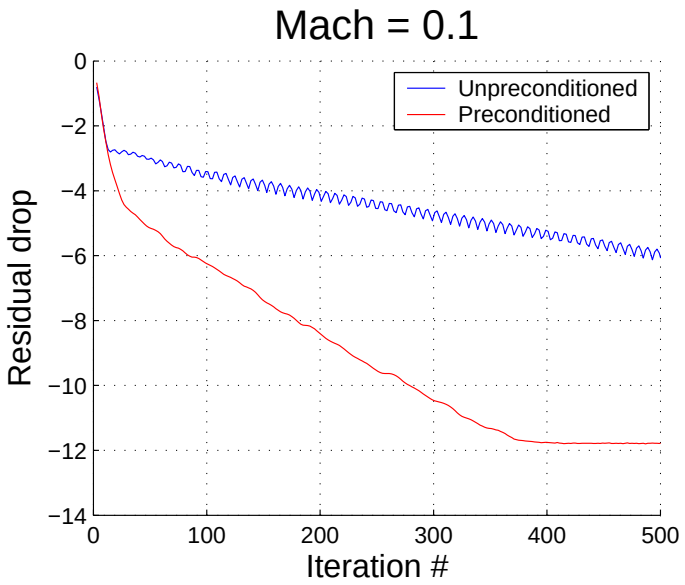
2D Euler test case

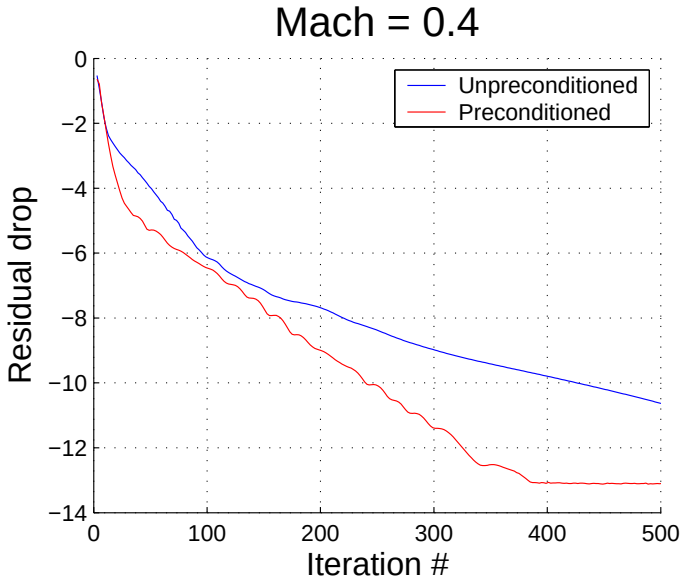
- 32x160 NACA0012 airfoil at 2.25° and $M = \{0.01, 0.1, 0.4, 0.8\}$.
- 5-level multigrid using the MJ coefficients for multistaging.
- CFL number $\sigma = 3$ on both fine and coarse meshes.
- Artificial dissipation:
  - $\mu_a = \frac{5}{64}$ on fine meshes using high order JST matrix scheme
  - $\mu_a = \frac{5}{32}$ on coarse meshes using low order matrix scheme
- In every test case, the squared preconditioner converged faster.
- The un-preconditioned scheme was unable to properly converge for low Mach numbers.

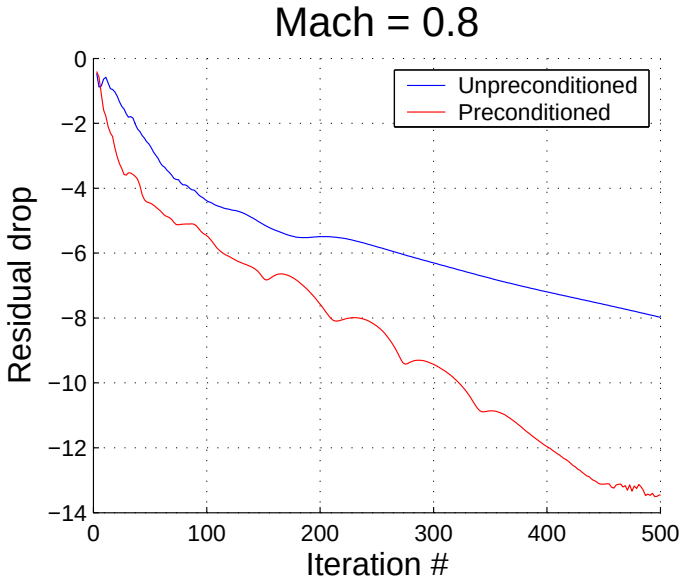
	C_L	$1000C_D$	C_L	$1000C_D$	C_L	$1000C_D$	C_L	$1000C_D$
Mach	0.01		0.1		0.4		0.8	
Un-prec.	0.254	158	0.273	151	0.302	5	0.651	524
Squared Prec.	0.274	-3	0.276	-3	0.303	0	0.652	522

Convergence history









Convective terms in the limit of low Mach numbers

Using entropy variables it can be shown (Gustafsson & Stoor 91) that in the limit of low Mach numbers

$$\begin{aligned} dW^e &= \left\{ \frac{dp}{\rho c}, \quad du, \quad dv, \quad dp - c^2 d\rho \right\}^t \\ &= \{O(M), O(1), O(1), O(1)\}^t , \end{aligned} \tag{40}$$

$$A_x^e = \begin{pmatrix} u & c & 0 & 0 \\ c & u & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix} = \begin{pmatrix} O(1) & O\left(\frac{1}{M}\right) & 0 & 0 \\ O\left(\frac{1}{M}\right) & O(1) & 0 & 0 \\ 0 & 0 & O(1) & 0 \\ 0 & 0 & 0 & O(1) \end{pmatrix}, \tag{41}$$

$$A_y^e = \begin{pmatrix} v & 0 & c & 0 \\ 0 & v & 0 & 0 \\ c & 0 & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix} = \begin{pmatrix} O(1) & 0 & O\left(\frac{1}{M}\right) & 0 \\ 0 & O(1) & 0 & 0 \\ O\left(\frac{1}{M}\right) & 0 & O(1) & 0 \\ 0 & 0 & 0 & O(1) \end{pmatrix}, \tag{42}$$

$$A_x^e \frac{\partial W^e}{\partial x} + A_y^e \frac{\partial W^e}{\partial y} = \left\{ O\left(\frac{1}{M}\right), O(1), O(1), O(1) \right\}^t . \tag{43}$$

Dissipative terms in the limit of low Mach numbers

$$|A_x^e| = \begin{pmatrix} c & u & 0 & 0 \\ u & c & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix} = \begin{pmatrix} O\left(\frac{1}{M}\right) & O(1) & 0 & 0 \\ O(1) & O\left(\frac{1}{M}\right) & 0 & 0 \\ 0 & 0 & O(1) & 0 \\ 0 & 0 & 0 & O(1) \end{pmatrix}, \quad (44)$$

$$|A_y^e| = \begin{pmatrix} c & 0 & v & 0 \\ 0 & v & 0 & 0 \\ v & 0 & c & 0 \\ 0 & 0 & 0 & v \end{pmatrix} = \begin{pmatrix} O\left(\frac{1}{M}\right) & 0 & O(1) & 0 \\ 0 & O(1) & 0 & 0 \\ O(1) & 0 & O\left(\frac{1}{M}\right) & 0 \\ 0 & 0 & 0 & O(1) \end{pmatrix}. \quad (45)$$

Therefore the dissipative terms

$$\Delta x \left|A_x^e\right| \frac{\partial^2 W^e}{\partial x^2} + \Delta y \left|A_y^e\right| \frac{\partial^2 W^e}{\partial y^2} = \left\{O(1), O\left(\frac{1}{M}\right), O\left(\frac{1}{M}\right), O(1)\right\}^t \quad (46)$$

do not scale properly with the convective terms.

Proper scaling of dissipation terms

Choosing any low speed preconditioner (Turkel, VLR, Weiss-Smith, . . .) we have

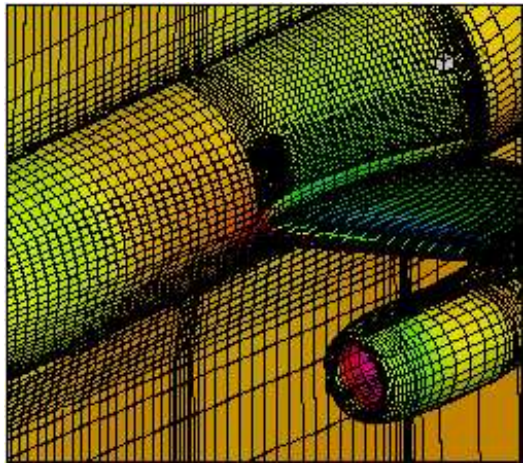
$$P^{e-1} = P^{e-1}\left|P^e {A_x}^e\right| = P^{e-1}\left|P^e {A_y}^e\right| = \begin{pmatrix} O\left(\frac{1}{M^2}\right) & O\left(\frac{1}{M}\right) & O\left(\frac{1}{M}\right) & O\left(\frac{1}{M}\right) \\ O\left(\frac{1}{M}\right) & O(1) & O(1) & O(1) \\ O\left(\frac{1}{M}\right) & O(1) & O(1) & O(1) \\ O\left(\frac{1}{M}\right) & O(1) & O(1) & O(1) \end{pmatrix}. \tag{47}$$

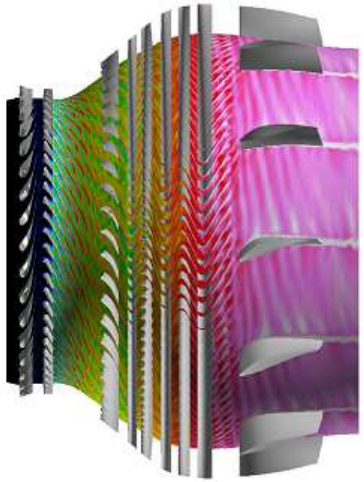
Therefore

$$\Delta x P^{e-1}\left|P^e {A_x}^e\right| \frac{\partial^2 W^e}{\partial x^2} + \Delta y P^{e-1}\left|P^e {A_y}^e\right| \frac{\partial^2 W^e}{\partial y^2} = \left\{O\left(\frac{1}{M}\right), O(1), O(1), O(1)\right\}^t. \tag{48}$$

TFLO2000 solver

- One of the two solvers developed at Stanford's ASC Center.
- Massively parallel URANS, MPI-based flow solver.
- Multiple turbulence models including Durbin's $v^2 - f$.
- Conservative sliding mesh interface implementation.
- Parallel Python interface to other fluid and structural solvers.
- Steady-state, time-accurate, and time-spectral approaches.

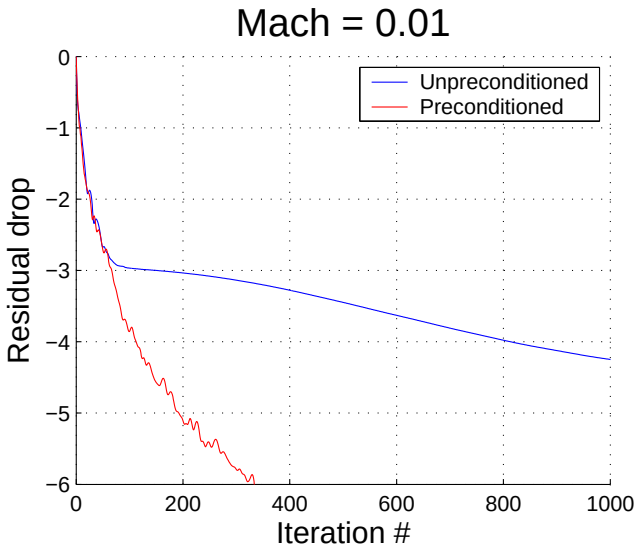


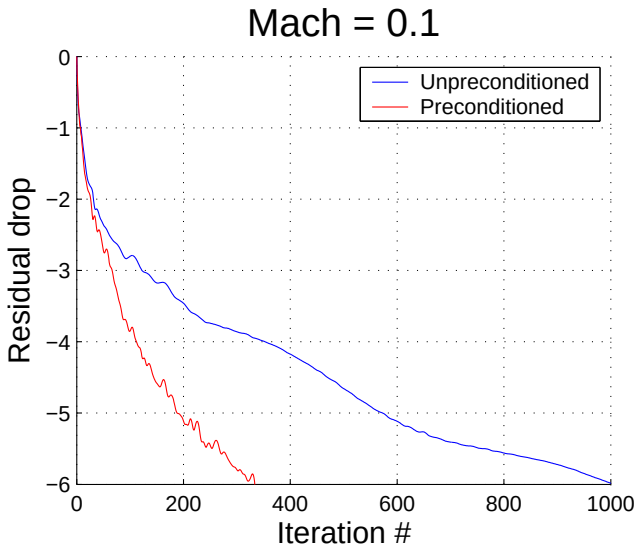


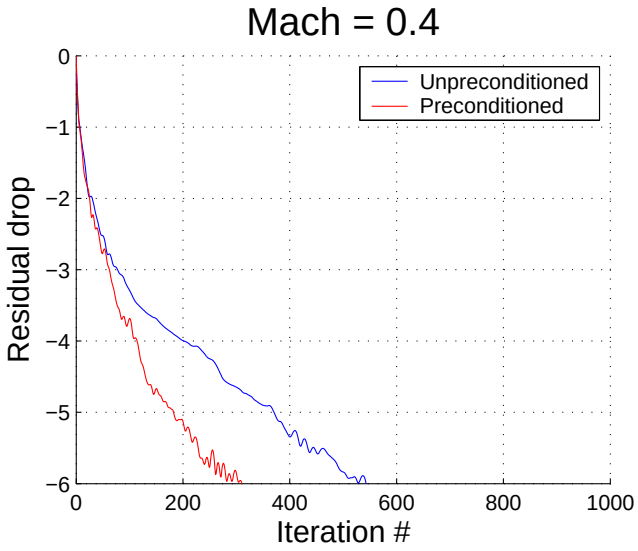
3D Euler test case

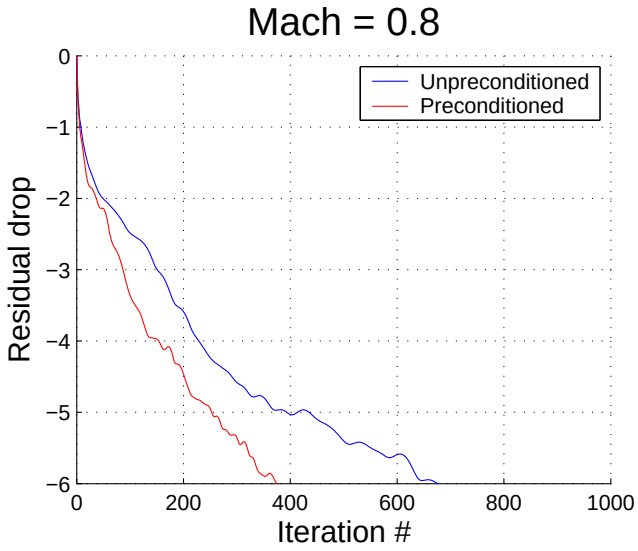
- TFLO2000 used on a 48-block mesh.
- $192 \times 32 \times 48$ ONERA M6 wing at $3.06°$.
- 4-level multigrid using the MJ coefficients for multistaging.
- Mach numbers 0.01, 0.1, 0.4, and 0.8.
- CFL number $\sigma = 2$ on both fine and coarse meshes.
- Artificial dissipation:
  - $\mu_a = \frac{1}{8}$ on fine meshes using high order JST matrix scheme.
  - $\mu_a = \frac{1}{2}$ on coarse meshes using low order matrix scheme.
- In a similar fashion than the 2D case, for every Mach number the squared preconditioner converged faster than the un-preconditioned scheme.

Convergence history for the 3D Euler ONERA M6 wing









Block-Jacobi for the Navier-Stokes equations

The Navier-Stokes equations are

$$\begin{aligned} \frac{\partial W}{\partial t} + \left[A_x \frac{\partial W}{\partial x} + A_y \frac{\partial W}{\partial y} \right] \\ -\mu_a \left[\Delta x^3 \left| A_x \right| \frac{\partial^4 W}{\partial x^4} + \Delta y^3 \left| A_y \right| \frac{\partial^4 W}{\partial y^4} \right] \\ -B_{xx} \frac{\partial^2 W}{\partial x^2} - B_{yy} \frac{\partial^2 W}{\partial y^2} - B_{xy} \frac{\partial^2 W}{\partial x \partial y} \quad = \quad 0. \end{aligned} \tag{49}$$

Therefore the matrix time-step becomes

$$[\Delta t]^{-1} = {P_{BJ}}^{-1} = \frac{1}{\sigma} \left(\frac{|A_x|}{\Delta x} + \frac{|A_y|}{\Delta y} + \frac{2B_{xx}}{\Delta x^2} + \frac{2B_{yy}}{\Delta y^2} \right). \tag{50}$$

Viscous Jacobians in entropy variables

The viscous Jacobians are not symmetric in entropy variables. Moreover the inversion of the Block-Jacobi preconditioner now requires a full matrix inversion.

$$B^e_{xx} = \begin{pmatrix} \frac{\gamma\mu\left(p-\rho c^2\right)}{\rho c^2 \mathrm{Pr}(\gamma-1)} & \frac{(\gamma-1)\mu u}{c} & \frac{4(\gamma-1)\mu v}{3c} & -\frac{\gamma p\mu}{\rho^2 c^3 \mathrm{Pr}(\gamma-1)} \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \frac{4\mu}{3} & 0 \\ \frac{\gamma\mu\left(p-\rho c^2\right)}{c\mathrm{Pr}(\gamma-1)} & \rho\left(\gamma-1\right)\mu u & \frac{4\rho(\gamma-1)\mu v}{3} & -\frac{\gamma p\mu}{\rho c^2 \mathrm{Pr}(\gamma-1)} \end{pmatrix} \tag{51}$$

and

$$B^e_{yy} = \begin{pmatrix} \frac{\gamma\mu\left(p-\rho c^2\right)}{\rho c^2 \mathrm{Pr}(\gamma-1)} & \frac{4(\gamma-1)\mu u}{3c} & \frac{(\gamma-1)\mu v}{c} & -\frac{\gamma p\mu}{\rho^2 c^3 \mathrm{Pr}(\gamma-1)} \\ 0 & \frac{4\mu}{3} & 0 & 0 \\ 0 & 0 & \mu & 0 \\ \frac{\gamma\mu\left(p-\rho c^2\right)}{c\mathrm{Pr}(\gamma-1)} & \frac{4\rho(\gamma-1)\mu u}{3} & \rho\left(\gamma-1\right)\mu v & -\frac{\gamma p\mu}{\rho c^2 \mathrm{Pr}(\gamma-1)} \end{pmatrix}. \tag{52}$$

Parabolic symmetrizing variables

It is possible to symmetrize both the inviscid and viscous Jacobians (Abarbanel & Gottlieb 80), nevertheless the Block-Jacobi matrix will be full except for two zero terms.

Using

$$dW^s = \left\{ \frac{cd\rho}{\sqrt{\gamma}\rho}, \quad du, \quad dv, \quad \frac{cdT}{\sqrt{\gamma(\gamma-1)T}} \right\}^t. \tag{53}$$

the inviscid Jacobians become

$$A_x^s = \begin{pmatrix} u & \frac{c}{\sqrt{\gamma}} & 0 & 0 \\ \frac{c}{\sqrt{\gamma}} & u & 0 & c\sqrt{\frac{\gamma-1}{\gamma}} \\ 0 & 0 & u & 0 \\ 0 & c\sqrt{\frac{\gamma-1}{\gamma}} & 0 & u \end{pmatrix}, A_y^s = \begin{pmatrix} v & 0 & \frac{c}{\sqrt{\gamma}} & 0 \\ 0 & v & 0 & 0 \\ \frac{c}{\sqrt{\gamma}} & 0 & v & c\sqrt{\frac{\gamma-1}{\gamma}} \\ 0 & 0 & c\sqrt{\frac{\gamma-1}{\gamma}} & v \end{pmatrix}, \tag{54}$$

and the viscous Jacobians become

$$B_{xx}^s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{4\nu}{3} & 0 & 0 \\ 0 & 0 & \nu & 0 \\ 0 & 0 & 0 & \frac{\gamma\nu}{Pr} \end{pmatrix}, B_{xy}^s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\nu}{3} & 0 \\ 0 & \frac{\nu}{3} & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, B_{yy}^s = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \nu & 0 & 0 \\ 0 & 0 & \frac{4\nu}{3} & 0 \\ 0 & 0 & 0 & \frac{\gamma\nu}{Pr} \end{pmatrix}. \tag{55}$$

Fine-tuning of convergence acceleration

- Many coefficients such as the CFL number, the artificial dissipation coefficients, and the multistage coefficients can be optimized to achieve maximum convergence.

- In our optimizations we have used gradient-based methods such as the FORTRAN package SNOPT (Murray, Gill, Saunders) and the MATLAB function `fmincon`, as well as the genetic algorithm MicroGA (Carroll).

- One simple approach bases the objective function on an analytical model such as Fourier analysis.

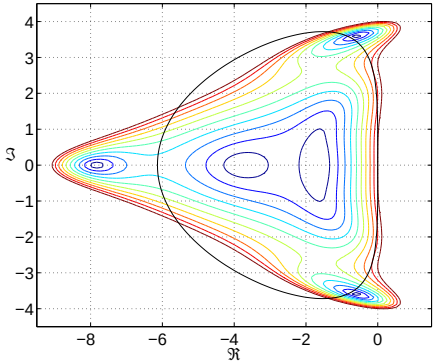
- Another approach links an optimizer directly to a flow solver and minimizes the residual.

Four-Variable Optimization Results

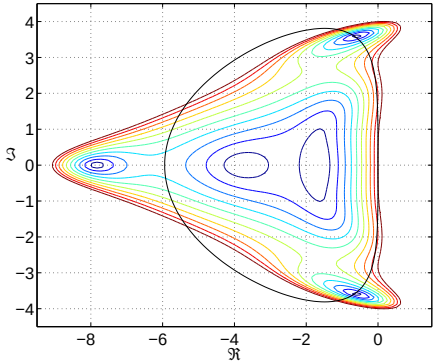
- Gradient-based optimizer SNOPT linked to FLO103 to improve convergence criterion of residual drop after 100 iterations, $R_D = \log\left(\frac{R_1}{R_{100}}\right)$.
- Direct numerical optimization, not based on any model.
- *Design* parameters: σ, δ, $\mu_{a_{\text{fine}}}$, and $\mu_{a_{\text{coarse}}}$.

	Un-preconditioned				Block-Jacobi				Squared			
Case	4UP0	4UP1	4UP4	4UP8	4BJ0	4BJ1	4BJ4	4BJ8	4SQ0	4SQ1	4SQ4	4SQ8
M	0.01	0.1	0.4	0.8	0.01	0.1	0.4	0.8	0.01	0.1	0.4	0.8
σ	3.88	3.46	3.49	3.62	3.80	3.29	3.54	3.14	3.72	3.81	3.54	3.14
δ	1.00	1.00	1.00	1.00	0.80	0.97	0.29	0.43	0.02	0.21	0.29	0.43
$\frac{\mu_c}{\mu_{c_0}}$	3.37	5.05	2.04	1.51	2.68	5.87	1.86	1.00	1.20	2.10	1.86	1.00
$\frac{\mu_f}{\mu_{f_0}}$	1.65	3.16	3.19	3.07	2.03	3.36	2.81	3.63	3.30	3.12	2.81	3.63
C_L	0.241	0.273	0.301	0.656	0.234	0.272	0.302	0.647	0.274	0.275	0.303	0.651
D_C	172	15	9	531	181	24	6	524	-1	-1	0	522
R_D	4.00	4.42	6.83	4.91	4.02	5.04	7.19	5.84	6.54	6.72	7.19	5.84

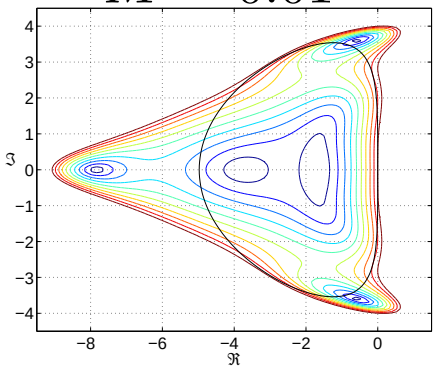
Stability domains for optimized variables



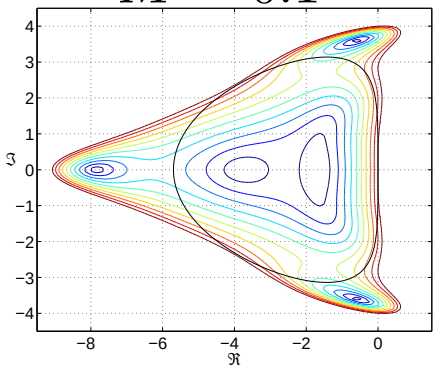
$M = 0.01$



$M = 0.1$



$M = 0.4$



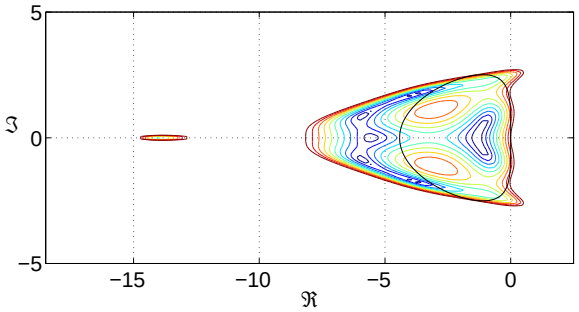
$M = 0.8$

Stability domains for the optimized coefficients for four-variable optimizations for the Squared preconditioner

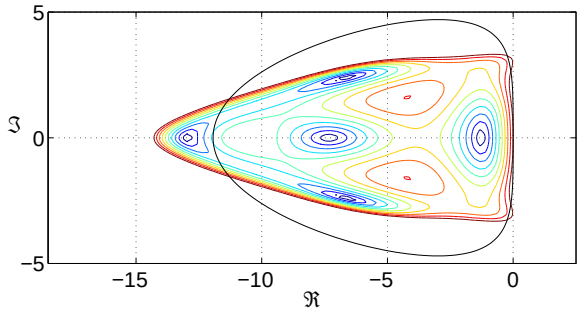
Ten-Variable Optimization Results

Additional parameters are α_{1-4}, β_3, and β_5.

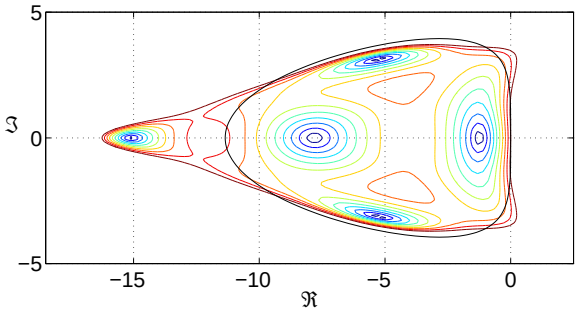
M	σ	δ	$\frac{\mu_c}{\mu_{cr}}$	$\frac{\mu_f}{\mu_{f_r}}$	α_1	α_2	α_3	α_4	β_3	β_5	R_D
0.01	2.51	0.00218	1.36	3.52	0.00278	0.524	0.220	0.856	0.280	0.0765	6.45
0.1	4.70	0.291	2.30	5.08	0.0694	0.210	0.279	0.471	0.293	0.286	7.48
0.4	3.95	0.773	2.41	5.75	0.165	0.173	0.348	0.454	0.314	0.277	8.40
0.8	2.16	1.000	1.64	6.70	0.100	0.240	0.641	0.412	0.527	0.117	5.68



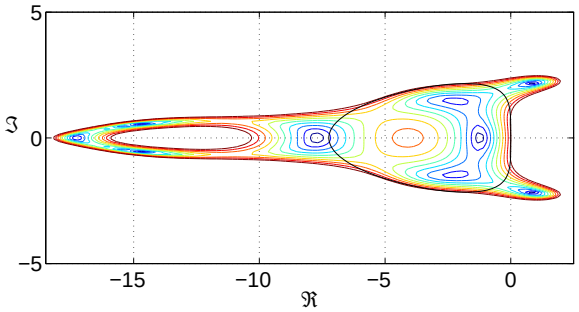
$M = 0.01$



$M = 0.1$

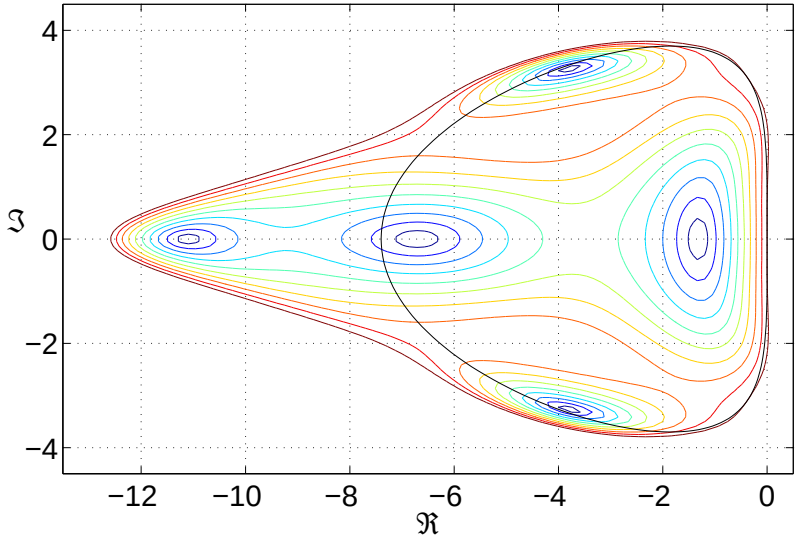


$M = 0.4$

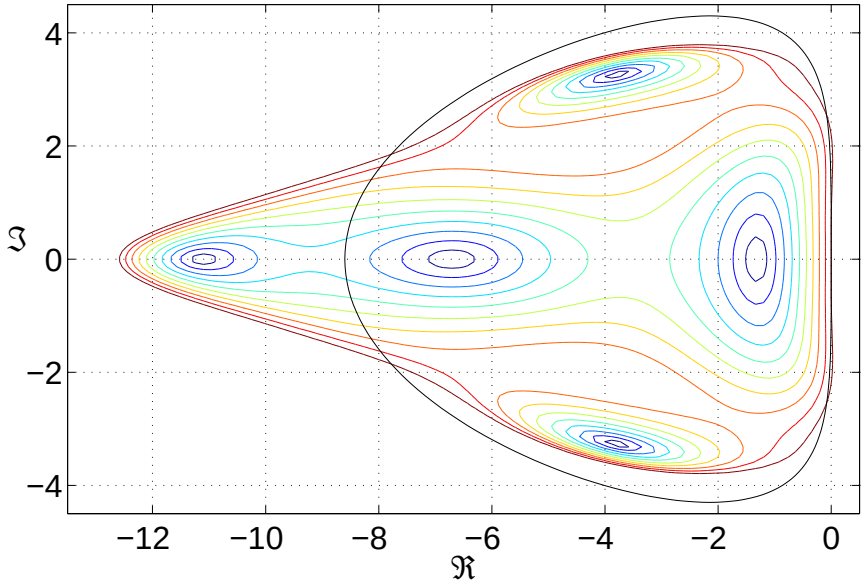


$M = 0.8$

Objective function based on Fourier analysis



$\sigma = 3.7$ used for $M = 0.8$



$\sigma = 4.3$ used for $M = 0.01$

Stability domains for the analytically optimized coefficients

Conclusions

- Summary of important aspects of preconditioning methods in explicit, multistage, multigrid flow solvers.
- Advantages of squared preconditioning shown both analytically and numerically.
- Specific aspects of implementation to achieve good convergence highlighted.
- Fine-tuning of parameters and coefficients of the method performed using optimization based on both numerical and analytical models.
- Showed correlations and limitations of each method.
- Future work will finish efficient implementation for NS and will explore hybrid optimization approaches for both Euler and NS.