A Matrix-Free Implicit Method for Flows at all Speeds

Thibaud Kloczko
CEA Saclay
DEN / DM2S / LTMF
91191 Gif-sur-Yvette Cedex
FRANCE
thibaud.kloczko@cea.fr

Christophe Corre
SINUMEF Lab. ENSAM
151 boulevard de l'hôpital
75013 Paris
FRANCE
corre@paris.ensam.fr

Alberto Beccantini
CEA Saclay
DEN / DM2S / LTMF
91191 Gif-sur-Yvette Cedex
FRANCE
beccantini@cea.fr

Objective of the study

• *Efficient* computation of approximation solutions for the Euler and Navier-Stokes equations

- ✓ for the whole compressible regime

  (supersonic, transonic, subsonic, $M \to 0$ (low-Mach))
- ✓ steady and unsteady flows

• Efficiency :

- ✓ reduced computational time
  - ✗ convergence to steady-state after a small number of iterations and / or for a small amount of time per iteration
- ✓ reduced memory storage
  - ✗ crucial for computations involving a large number of grid points

Context of the study

- fast convergence to steady state ensured by using **implicit** schemes

Efficiency $\Rightarrow$ Optimisation of the implicit stage solution both in terms of CPU time and memory requirements

- existing treatments for the compressible regime with emphasis on the reduction of memory storage :

  - ✓ diagonalization
    - ✗ approximate factorization (Pulliam & Chaussée, 1981)
    - ✗ relaxation (Corre & Lerat, 1998)
  - ✓ **matrix-free method** (Huo, Baum & Löhner, 1998, 2001)

Context of the study

• extension to *low-Mach number* flow

- through low-Mach number preconditioning

  $\Rightarrow$ need to optimize implicit solution for the preconditioned Euler and Navier-Stokes equations

• optimisation of implicit solution for preconditioned Euler and Navier-Stokes equations

- diagonal implicit stage $\Rightarrow$ see e.g. OVERFLOW (Pandya, Pulliam, Venkateswaran, 2003)
- **matrix-free method** $\Rightarrow$ present study

Outline of the presentation

1. Description of a matrix-free implicit treatment that is no longer matrix-free for the preconditioned Euler or Navier-Stokes equations

2. Optimisation of the implicit treatment taking advantage of the Turkel precondioning properties

3. Applications to low-Mach number flows on structured and unstructured grid / efficiency assessment

Outline

1. Description of a matrix-free implicit treatment that is no longer matrix-free for the preconditioned Euler or Navier-Stokes equations

2. Optimisation of the implicit treatment taking advantage of the Turkel precondioning properties

3. Applications to low-Mach number flows on structured and unstructured grid / efficiency assessment

Governing Equations

Time-accurate solution of the Navier-Stokes equations for flows at all speeds :

$$P^{-1}\cdot\frac{\partial w}{\partial \tau}+\frac{\partial w}{\partial t}+\frac{\partial f^{E}(w)}{\partial x}=\frac{\partial f^{V}(w,w_x)}{\partial x}$$

w : conservative variables

τ : pseudo-time or dual time

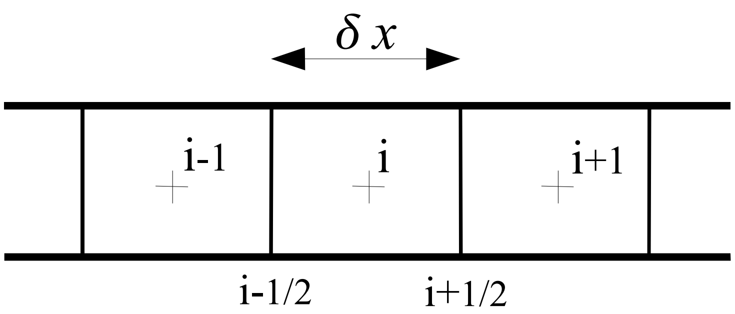
t : physical time

f^E: Euler flux

f^V: viscous flux

P : low-Mach preconditioning matrix for conservative variables

Discretization (1)



1- Time and space steps :

δx : mesh size

Δt : physical time step

$\Delta \tau$: pseudo time step

$$\sigma = \frac{\Delta \tau}{\delta x} \qquad \lambda = \frac{\Delta \tau}{\Delta t}$$

2- Difference and average operators over one grid cell:

$$\delta \Phi_{i+1/2} = \Phi_{i+1} - \Phi_i \qquad \mu \Phi_{i+1/2} = \frac{1}{2}\left(\Phi_{i+1} + \Phi_i\right)$$

Φ : function of mesh point $x = i \cdot \delta x$

Discretization (2)

$$P^{-1}\cdot \underbrace{\frac{w_i^{n,m+1}-w_i^{n,m}}{\Delta\tau}}_{\text{First order accuracy for pseudo-time}} + \underbrace{\frac{\frac{3}{2}\left(w_i^{n,m+1}-w_i^{n}\right)-\frac{1}{2}\left(w_i^{n}-w_i^{n-1}\right)}{\Delta t}}_{\text{Second order accuracy for physical-time}} + \underbrace{\frac{\delta h_i^{n,m}}{\delta x}}_{\text{Numerical flux}} = 0$$

m : pseudo-time counter

$\Delta w^{n,m} = w^{n,m+1} - w^{n,m}$

$\sigma = \dfrac{\Delta\tau}{\delta x}$

n : physical-time counter

$\Delta w^{n-1} = w^{n} - w^{n-1}$

$\lambda = \dfrac{\Delta\tau}{\Delta t}$

$h_{i+1/2}$: numerical flux approximating $(f^E - f^V)$ at midpoint

- Explicit stage :

$$\left(P^{-1}+\frac{3}{2}\lambda\, Id\right)\cdot\Delta w_i^{n,m} = \underbrace{-\sigma\,\delta h_i^{n,m} - \frac{3}{2}\lambda\left(w_i^{n,m}-w^{n}\right) + \frac{1}{2}\lambda\,\Delta w_i^{n-1}}_{RHS\, =\, \Delta w_i^{\exp}}$$

Numerical flux

• Upwind discretization for inviscid flux (Rusanov, Roe or AUSM+ scheme)

$$f^E(w) \rightarrow h^E_{i+1/2}$$

• High-resolution computation with MUSCL reconstruction of left and right states

$$h^E_{i+1/2} = h^E[w^L_{i+1/2}, w^R_{i+1/2}]$$

• Simple second-order centered discretization for viscous flux

$$f^V(w, w_x) \rightarrow \tilde{f}^V = f^V(\mu w, \frac{\delta w}{\delta x})$$

<u>Example</u> : preconditionned Roe scheme numerical flux

$$h_{i+1/2} = \frac{1}{2}(f^E(w^L_{i+1/2}) + f^E(w^R_{i+1/2})) - \tilde{f}^V_{i+1/2} - \frac{1}{2}P^{-1}|P \cdot A|(w^R_{i+1/2} - w^L_{i+1/2})$$

Implicit Stage Construction (1)

- Implicitation with respect to the dual time

$$P^{-1}\cdot\Delta w_i^{n,m} + \frac{3}{2}\lambda\,\Delta w_i^{n,m} + \sigma\,\delta\,\Delta h_i^{n,m} = \Delta w_i^{\exp}$$

- Details of the numerical fluxes implicitation

$$h^E = h^E(w_L, w_R) \Rightarrow (\Delta h^E)^{n,m} \simeq \left(\frac{\partial h^E}{\partial w_L}\right)^{n,m} \Delta w_L^{n,m} + \left(\frac{\partial h^E}{\partial w_R}\right)^{n,m} \Delta w_R^{n,n}$$

$$(\Delta \tilde{f}^V)^{n,m} = \left(\frac{\partial f^V}{\partial (w_x)}\right)^{n,m} \cdot \frac{\delta}{\delta x}(\Delta w^{n,m}) = \frac{(A_1^V)^{n,m}}{\delta x}\,\delta\,\Delta w^{n,m}$$

Implicit Stage Construction (2)

- First-order block-implicit stage

$$D\,\Delta w_i^{n,m} + C_1^{+}\,\Delta w_{i+1}^{n,m} + C_1^{-}\,\Delta w_{i-1}^{n,m} = \Delta w_i^{\exp}$$

$$\begin{cases} C_1^{+} = \quad \sigma \left[\left(\dfrac{\partial h^E}{\partial w_R} \right) - \dfrac{A_1^V}{\delta x} \right]_{i+1/2} \\[2ex] C_1^{-} = -\,\sigma \left[\left(\dfrac{\partial h^E}{\partial w_R} \right) + \dfrac{A_1^V}{\delta x} \right]_{i-1/2} \\[2ex] D = P^{-1} + \dfrac{3}{2}\lambda\, Id - C_1^{+} - C_1^{-} \end{cases}$$

Matrix-Free Simplifications

- Step 1 : $h^E = h^{RUSANOV}$ in implicit stage only

$$h^{RUSANOV} = \frac{1}{2}\left(f^E(w_L)+f^E(w_R)\right) - \frac{1}{2}\rho(P A^E) P^{-1}(w_R - w_L)$$

$$\Rightarrow \begin{cases} \left(\dfrac{\partial h^E}{\partial w_L}\right) = \dfrac{1}{2}\left(A^E + \rho(P A^E) P^{-1}\right) \\ \left(\dfrac{\partial h^E}{\partial w_R}\right) = \dfrac{1}{2}\left(A^E - \rho(P A^E) P^{-1}\right) \end{cases}$$

- Step 2 : spectral radius simplification $A_1^V \rightarrow \rho(A_1^V)\, Id$

$\Rightarrow$ Simplified implicit stage

Matrix-Free Simplifications

⇒ Simplified implicit stage

$$D\,\Delta w_i^{n,m} + C_1^+\,\Delta w_{i+1}^{n,m} + C_1^-\,\Delta w_{i-1}^{n,m} = \Delta w_i^{\exp} - \sigma\,\delta\,\Delta(f^E)_i^{n,m}$$

$$A^E\,\Delta w^{n,m} \approx \Delta(f^E)^{n,m}$$

$$C_1^+ = -\left[\frac{1}{2}\dot{\tilde{\rho}}^E P^{-1} + \dot{\rho}^V Id\right]_{i+1/2}$$

$$C_1^- = -\left[\frac{1}{2}\dot{\tilde{\rho}}^E P^{-1} + \dot{\rho}^V Id\right]_{i-1/2}$$

$$\begin{cases} \dot{\tilde{\rho}}^E = \sigma\,\rho(P A^E) \\ \dot{\rho}^V = \dfrac{\sigma}{\delta x}\rho(A_1^V) \end{cases}$$

$$D = \left(1 + \dot{\tilde{\rho}}^E_{i-1/2} + \dot{\tilde{\rho}}^E_{i+1/2}\right) P_i^{-1} + \left(\frac{3}{2}\lambda + \dot{\rho}^V_{i-1/2} + \dot{\rho}^V_{i+1/2}\right) Id$$

Matrix-Free Simplifications

- Step 3 : Point Jacobi treatment

$$\Delta w_i^{(l+1)} = D_i^{-1}\left[\Delta w_i^{\exp} - \sigma\,\delta\,\mu\,\Delta\left(f^{E}\right)_i^{(l)} + \dot{\rho}_{i+1/2}^{V}\,\Delta w_{i+1}^{(l)} + \dot{\rho}_{i-1/2}^{V}\,\Delta w_{i-1}^{(l)}\right]$$
$$+\, D_i^{-1}\cdot P_i^{-1}\left[\frac{\dot{\tilde{\rho}}_{i+1/2}}{2}\,\Delta w_{i+1}^{(l)} + \frac{\dot{\tilde{\rho}}_{i-1/2}}{2}\,\Delta w_{i-1}^{(l)}\right]$$

$$D^{-1} = \left(a\,P^{-1} + b\,Id\right)^{-1} \qquad \begin{cases} a = 1 + \dot{\tilde{\rho}}_{i-1/2}^{E} + \dot{\tilde{\rho}}_{i+1/2}^{E} \\ b = \dfrac{3}{2}\lambda + \dot{\rho}_{i-1/2}^{V} + \dot{\rho}_{i+1/2}^{V} \end{cases}$$

- No preconditioning : $P^{-1} = Id \Rightarrow D^{-1} = 1/(a+b)$
  - ✓ above treatment is truly matrix-free (*Löhner et al*, 1998, 2001)
  - ✓ extremely cheap
  - ✓ globally efficient if not intrinsically highly efficient

"Matrix-Free" method for low-Mach flows

• Low-Mach regime :

$$P = Id + (\beta^2 - 1)\, Q = Id + (\beta^2 - 1)\, \frac{\gamma - 1}{c^2} \begin{bmatrix} 1 \\ u \\ v \\ H \end{bmatrix} \cdot \left[q^2 \,; -u \,; -v \,; 1 \right]$$

$\Rightarrow$ D is a full matrix

$\Rightarrow$ Increase of storage requirement and operation count

$\Rightarrow$ Need for optimisation of the treatment

• Observations :

Matrix Q is idempotent for Turkel preconditioning

$$Q^2 = Q$$

Outline

1. Description of a matrix-free implicit treatment that is no longer matrix-free for the preconditioned Euler or Navier-Stokes equations

2. Optimisation of the implicit treatment taking advantage of the Turkel precondioning properties

3. Applications to low-Mach number flows on structured and unstructured grid / efficiency assessment

"Matrix-Free" method optimisation

- Property $Q^2 = Q$:

$\Rightarrow$ Simple explicit computations of D^{-1} and $D^{-1}P^{-1}$

$$D^{-1} = \frac{1}{a+b}\left[Id + \frac{a(\beta^2 - 1)}{a + b\beta^2} Q \right]$$

$$\begin{cases} a = 1 + \dot{\tilde{\rho}}^{E}_{i-1/2} + \dot{\tilde{\rho}}^{E}_{i+1/2} \\ b = \dfrac{3}{2}\lambda + \dot{\rho}^{V}_{i-1/2} + \dot{\rho}^{V}_{i+1/2} \end{cases}$$

$$D^{-1} \cdot P^{-1} = \frac{1}{a+b}\left[Id - \frac{b(\beta^2 - 1)}{a + b\beta^2} Q \right]$$

"Matrix-Free" method optimisation

- Summing things up : PJ treatment for low-Mach number flows

$$\Delta w_i^{(l+1)} = \frac{1}{a+b}\left(\Delta w_1^{(l)} + \Delta w_2^{(l)}\right) + \frac{\beta^2 - 1}{(a+b)(a+b\beta^2)} Q\left(a\,\Delta w_1^{(l)} - b\,\Delta w_2^{(l)}\right)$$

with
$$\begin{cases} \Delta w_1^{(l)} = \Delta w_i^{\exp} - \sigma\,\delta\mu\,\Delta\left(f^E\right)_i^{(l)} + \dot{\rho}_{i+1/2}^V\,\Delta w_{i+1}^{(l)} + \dot{\rho}_{i-1/2}^V\,\Delta w_{i-1}^{(l)} \\ \Delta w_2^{(l)} = \dfrac{\dot{\tilde{\rho}}_{i+1/2}^E}{2}\Delta w_{i+1}^{(l)} + \dfrac{\dot{\tilde{\rho}}_{i-1/2}^E}{2}\Delta w_{i-1}^{(l)} \end{cases}$$

- Optimisation of the remaining matrix-vector product (Turkel, 1999)

$$Q \cdot X = Q \cdot \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \frac{\gamma - 1}{c^2}\left(\frac{q^2}{2} X_1 - u X_2 - v X_3 + X_4\right) \cdot \begin{bmatrix} 1 \\ u \\ v \\ H \end{bmatrix}$$

Truly Matrix-Free Implicit Treatment

- Final PJ-MF treatment for all-speed flows

$$\Delta w_i^{(l+1)} = \underbrace{\frac{1}{a+b}\left(\Delta w_1^{(l)} + \Delta w_2^{(l)}\right)}_{\substack{\text{standard MF treatment} \\ \text{with } \dot{\tilde{\rho}}^E \text{ instead of } \dot{\rho}^E}} + \underbrace{\chi_i \cdot \begin{bmatrix} 1 \\ u \\ v \\ H \end{bmatrix}}_{\substack{\text{specific low-Mach treatment} \\ (= 0 \text{ if } \beta = 1)}}$$

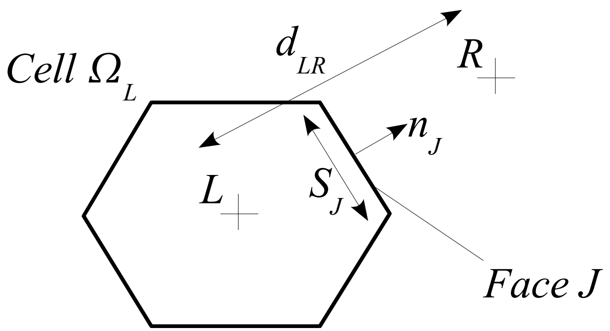
$$\chi_i = \frac{(\beta^2-1)(\gamma-1)}{c^2(a+b)(a+b\beta^2)}\left[a\left(\frac{q^2}{2}(\Delta w_1^{(1)})^{(l)} - u(\Delta w_1^{(2)})^{(l)} - v(\Delta w_1^{(3)})^{(l)} + (\Delta w_1^{(4)})^{(l)}\right) - b\left(\frac{q^2}{2}(\Delta w_2^{(1)})^{(l)} - u(\Delta w_2^{(2)})^{(l)} - v(\Delta w_2^{(3)})^{(l)} + (\Delta w_2^{(4)})^{(l)}\right)\right]$$

Truly Matrix-Free Implicit Treatment

- Immediate extension to unstructured grids with :

$$\begin{cases} \Delta w_1^{(l)} = \Delta w_L^{\exp} - \sum_J \left[\frac{\sigma}{2} \left(\Delta (f^E)_L^{(l)} + \Delta (f^E)_R^{(l)} \right) + \dot{\rho}_J^V \Delta w_R^{(l)} \right] \\ \Delta w_2^{(l)} = \sum_J \frac{\dot{\tilde{\rho}}_J^E}{2} \Delta w_R^{(l)} \end{cases}$$

where
$$\begin{cases} \sigma = \frac{\Delta t}{\Omega_L} \cdot S_J \\ \dot{\rho}_J^V = \frac{\sigma}{d_{LR}} \cdot \rho(A_J^V) \\ \dot{\tilde{\rho}}_J^E = \sigma \rho(P A_J^E) \end{cases}$$



Outline

1. Description of a matrix-free implicit treatment that is no longer matrix-free for the preconditioned Euler or Navier-Stokes equations

2. Optimisation of the implicit treatment taking advantage of the Turkel precondioning properties

3. Applications to low-Mach number flows on structured and unstructured grid / efficiency assessment

Fourier Analysis of the Matrix-Free Method

Purposes :

- Estimate the loss of intrinsic efficiency due to MF simplifications
- Compare the MF method with standard implicit block treatment

Analysis features :

- Roe scheme with MUSCL extrapolation
- Implicit Block treatment
- Implicit Matrix-Free treatment
- Direct Solver (*ie* the linear system is supposed to be solved exactly)
- Low-Mach preconditioning

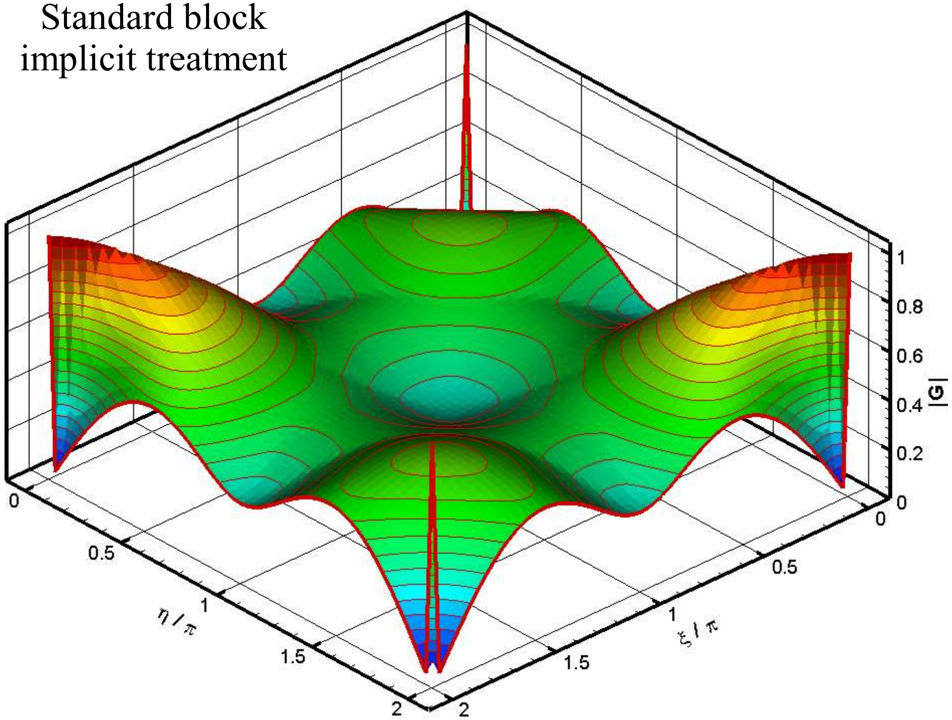
Euler Case Analysis

$$M = 10^{-1}$$
$$CFL = 10^{6}$$

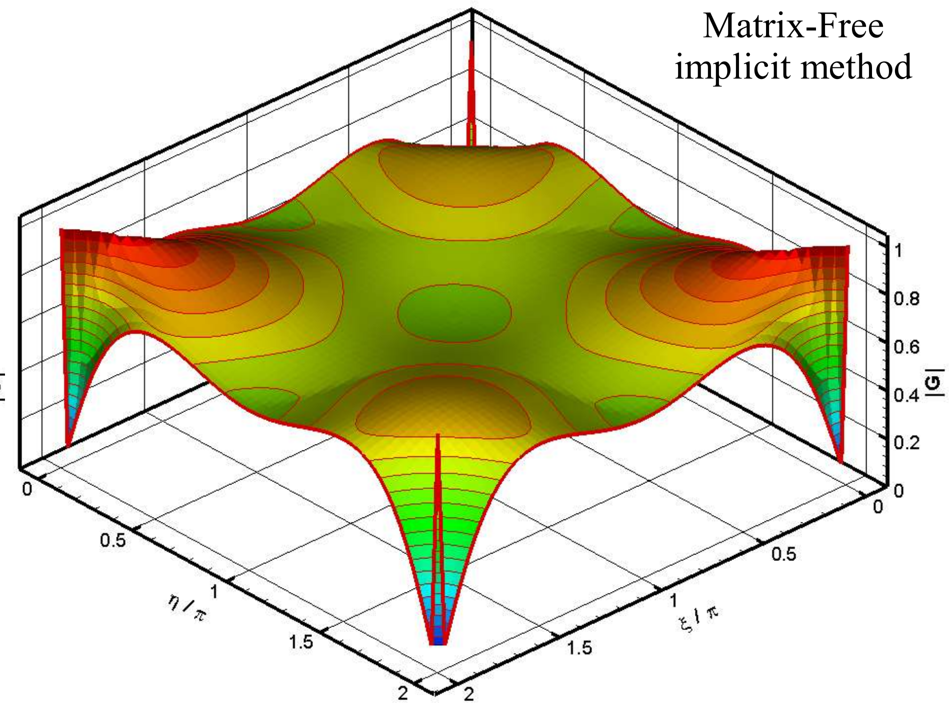
Roe-MUSCL scheme with Turkel preconditioning

Isovalues of the amplification factor in the reduced wave-numbers plane

Standard block implicit treatment



Matrix-Free implicit method



$\Rightarrow$ Loss of efficiency specially for wave-numbers close to π

Estimation of the loss of efficiency

Roe-MUSCL scheme
with Turkel preconditioning

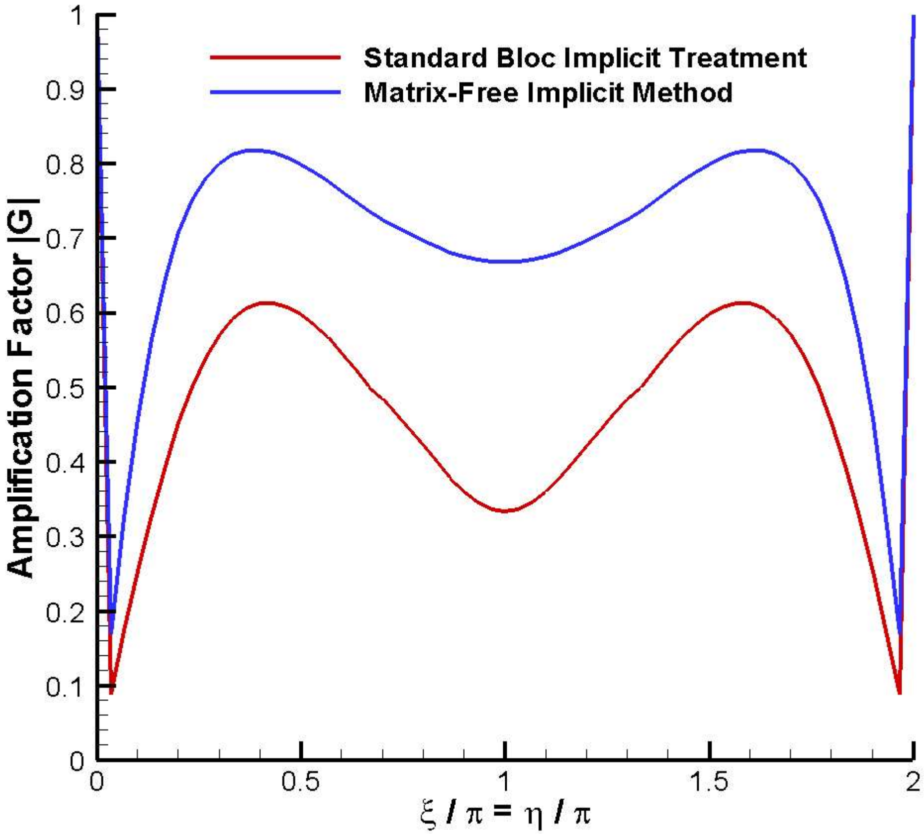
$$\text{Mach} = 10^{-1}$$

$$\text{CFL} = 10^{6}$$

$$\left|G^{BL}\right|_{ave} = 0.475$$
$$\left|G^{MF}\right|_{ave} = 0.704$$

$$\Rightarrow \quad \boxed{\frac{\log\left(\left|G^{BL}\right|\right)}{\log\left(\left|G^{MF}\right|\right)} = 2.12}$$

Amplification factor for equal wave-numbers



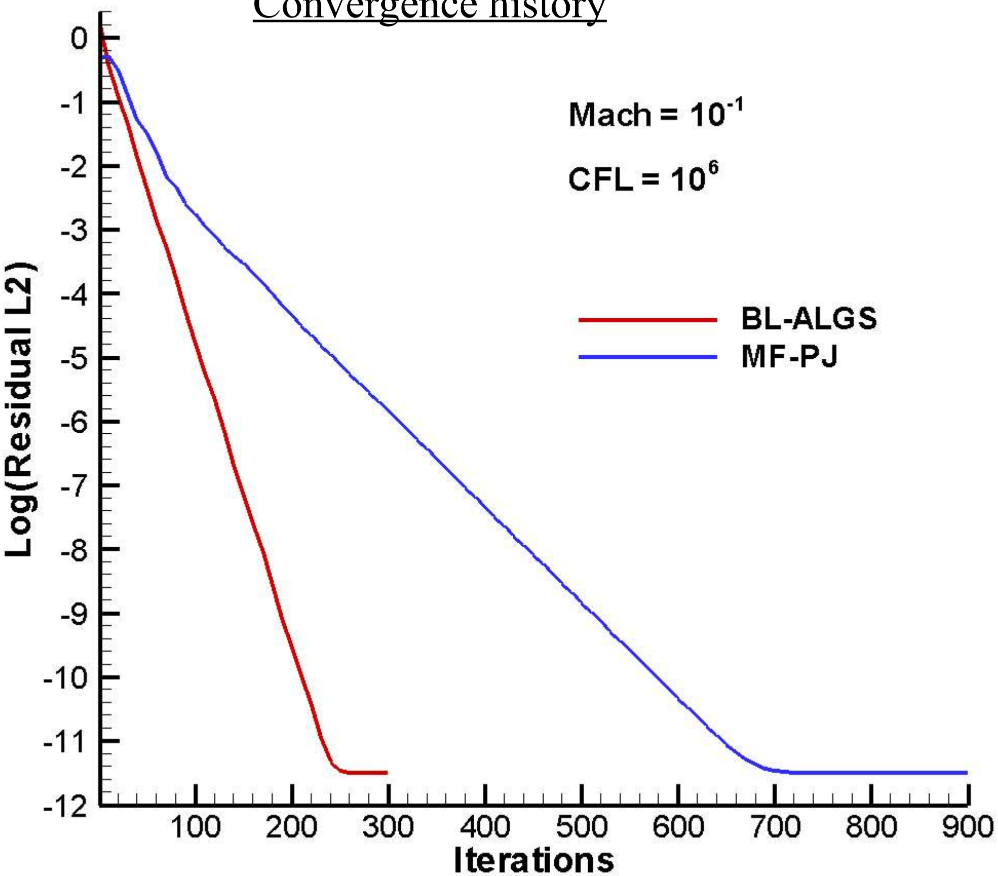
$\Rightarrow$ MF should need twice more iterations than Block to converge

Sine Bump Test Case

Roe-MUSCL scheme with Turkel preconditioning

- Regular mesh of 1280 elements
- Mach = 10^{-1} and CFL = 10^{6}

Convergence history



Number of iterations to converge :

$$N^{BL} = 250$$
$$N^{MF} = 690$$

Ratio between MF and Block :

$$\boxed{\frac{N^{MF}}{N^{BL}} = 2.76}$$

$\Downarrow$

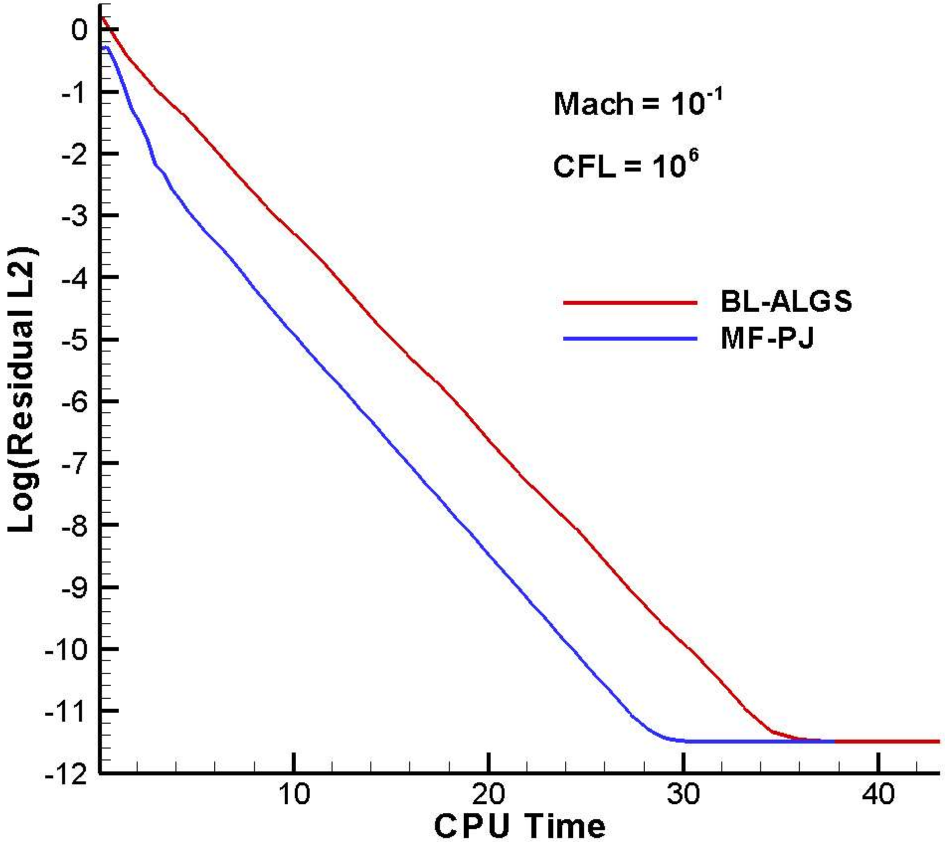
Good accordance with Fourier analysis

Global Cost of the Method

Nelts	Method	Iterations	CPU	CPI	CPPPI	Memory
1280	MF-PJ	690	29 s	0.039	3.1 E-5	2.88 Go
	BL-ALGS	250	36 s	0.146	11.4 E-5	3.21 Go

CPI : Cost per iteration
CPPPI : Cost per point per iteration

Convergence history in term of CPU time



Loss of intrinsic efficiency is balanced by a cheap cost per iteration and memory storage is significantly reduced

⇓

MF method seems to be a competitive alternative method to simulate flows at all speeds

Sine Bump : Unstructured Grids

- Irregular mesh of 1004, 4016 and 16064 elements
- Mach = 10^{-5}
- NK : Newton-Krylov algorithm (GMRES + ILU(0) preconditionner)
- MF-PJ : Matrix-Free Point Jacobi algorithm

Nelts	Method	Iterations	CPU
1004	MF-PJ	40	490 s
	NK	45	920 s
4016	MF-PJ	54	3700 s
	NK	55	5100 s
16064	MF-PJ	80	38000 s
	NK	65	32000 s

Observations :

MF method is also competitive on unstructured grids.

Although MF method requires more CPU time and iterations as the grid becomes finer it is still attractive for its low memory storage.

Conclusion

Matrix-Free implicit method for flows at all speeds :

- Truly matrix-free treatment derived for preconditionned NS equations
  - using specific properties of the widly spread Turkel preconditioning
- Easy to implement on structured and unstructured grids
- Low-memory storage
- Computationaly efficient for a wide range of external flows :
  - steady and unsteady inviscid flows (Sinebump, contact discontinuity...)
  - steady and unsteady viscous flows (Poiseuille, Stokes second problem...)

Future Work

- Further investigation and applications to internal flow problems
  - natural convection in cavity, vessel presurisation...
  - Formulation in viscous variables (p, u, v, T)
- Extension to multi-component flows

This is it !

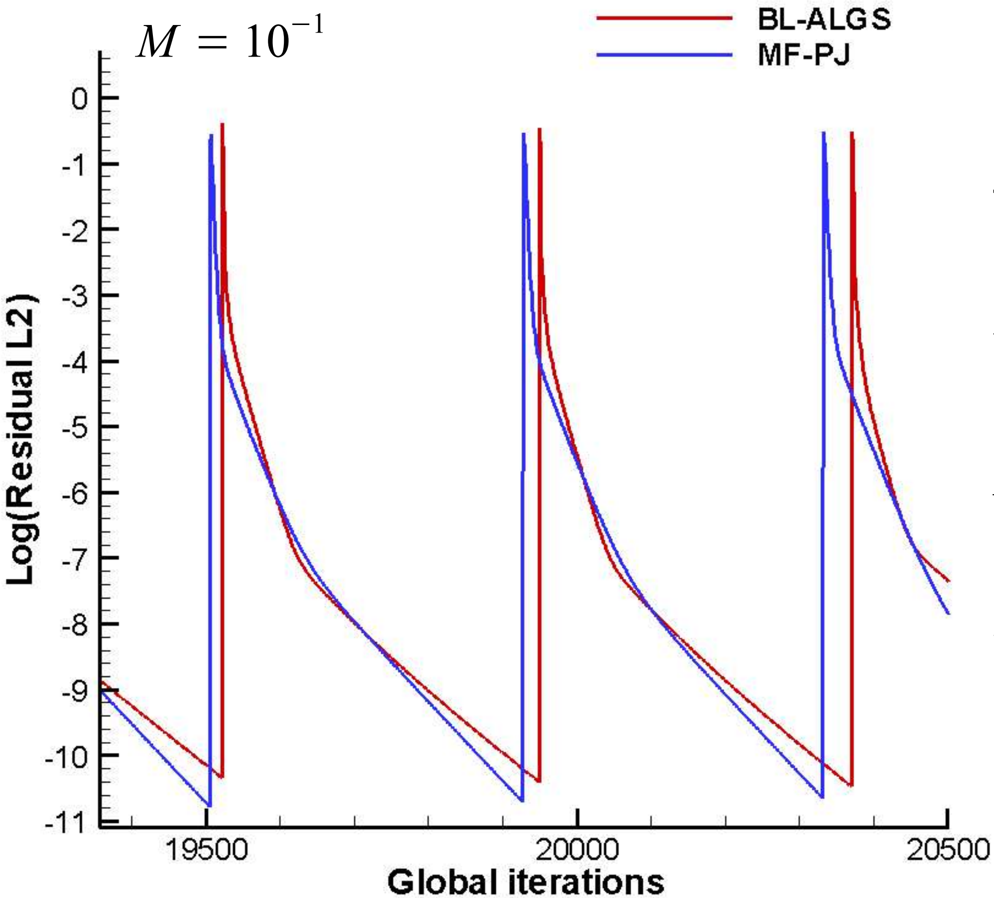
Thank you for your attention !

Steady Oscillations of a Plane below a Viscous Fluid

Nelts	Method	Iterations	CPU	CPPPI	Relative Memory
960	MF-PJ	65381	1265 s	2.0 E-5	1,51
	BL-ALGS	75935	6043 s	8.3 E-5	3,11

$R_e = 10$

$M = 10^{-1}$



<u>Observations</u> :

MF method is very competitive for unsteady flows computed through a dual-time fashion.

Not only MF method requires less CPU time and iterations but it is also very attractive for its low memory storage.