CFD ACCELERATION TECHNIQUES: IMPLICIT RUNGE-KUTTA

Matthew Fischels
Aerospace Engineering Department
Major Professor : Dr. R. Ganesh Rajagopalan

CFD Acceleration Techniques

- Goal
  - Decrease run-time of Computational Fluid Dynamics (CFD) software
- Methods
  - Implicit Runge-Kutta
  - Multigrid
  - GPU Acceleration

Implicit Runge-Kutta Methods

- CFD - Solving Navier-Stokes Equations
  - Coupled Non-linear PDEs

Momentum Equations - Newton's 2nd Law

$$\frac{\partial(\rho\vec{u})}{\partial t} + \rho\vec{u} \cdot \nabla\vec{u} = -\nabla P + \nabla \cdot (\mu\nabla\vec{u}) + S_u$$

Continuity - Mass Conservation

$$\frac{\partial\rho}{\partial t} + \nabla \cdot (\rho\vec{u}) = 0$$

- After spatial discretization of Momentum Equation
  - Non-Linear ODE

$$\frac{\partial u}{\partial t} = F\big(u(t)\big)$$

Implicit Runge-Kutta Methods

- General Form of Runge-Kutta

$$\frac{\partial u}{\partial t} = F(t, u)$$

$$\Delta t = t_{n+1} - t_n$$

$$s = \text{number of stages}$$

$$u_i = u_n + \Delta t \sum_{j=1}^{s} a_{i,j} F(t_n + c_j \Delta t, u_j)$$

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^{s} b_i F(t_n + c_i \Delta t, u_i)$$

$$\begin{array}{c|cccccc} c_1 & a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_i & a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{is} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{sj} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_j & \cdots & b_s \end{array}$$

Butcher Tableau

Implicit Runge-Kutta Methods

- Different Classifications of Runge-Kutta
  - Explicit (ERK)

$$u_i = u_n + \Delta t \sum_{j=1}^{i-1} a_{i,j} F\left(t_n + c_j \Delta t, u_j\right)$$

u_i has no dependence on u_i , u_{i+1} , . . .

$$\begin{array}{c|cccccc}
0 & 0 & & & & \\
c_2 & a_{21} & 0 & & & \\
c_3 & a_{31} & a_{32} & 0 & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Butcher Tableau

Implicit Runge-Kutta Methods

- Different Classifications of Runge-Kutta
  - Diagonally-Implicit (DIRK)

$$u_i = u_n + \Delta t \sum_{j=1}^{i} a_{i,j} F\left(t_n + c_j \Delta t, u_j\right)$$

$$= u_n + a_{i,i} \Delta t F\left(t_n + c_i \Delta t, u_i\right) + \Delta t \sum_{j=1}^{i-1} a_{i,j} F\left(t_n + c_j \Delta t, u_j\right))$$

$$\begin{array}{c|ccccc} c_1 & a_{11} & & & \\ c_2 & a_{21} & a_{22} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

Butcher Tableau

u_i is dependent on u_i , but not u_{i+1} , u_{i+2} , . . .

Implicit Runge-Kutta Methods

- Different Classifications of Runge-Kutta
  - Singly-Diagonal Implicit (SDIRK)

$$u_i = u_n + \Delta t \sum_{j=1}^{i} a_{i,j} F\big(t_n + c_j \Delta t, u_j\big)$$

$$= u_n + \gamma \Delta t F\big(t_n + c_i \Delta t, u_i\big) + \Delta t \sum_{j=1}^{i-1} a_{i,j} F\big(t_n + c_j \Delta t, u_j\big)\big)$$

$$\begin{array}{c|ccccc} c_1 & \gamma & & & \\ c_2 & a_{21} & \gamma & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & \gamma \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

Butcher Tableau

The implicit coefficients do not change from stage to stage. Can take advantage with factorization (ex. LU).

Implicit Runge-Kutta Methods

- Different Classifications of Runge-Kutta
    - Fully-Implicit (FIRK)
    - Explicit (ERK)
    - Diagonally-Implicit (DIRK)
    - Singly-Diagonal Implicit (SDIRK)
    - Explicit first stage, Diagonally-Implicit (EDIRK)
    - Explicit first stage, Singly-Diagonal Implicit (ESDIRK)
    - Implicit-Explicit (IMEX)
  - Focused on EDIRK and ESDIRK

Implicit Runge-Kutta Methods

- SIMPLER Algorithm for Incompressible Flows
  - Linearize and decouple equations

1) Guess velocity field
2) Solve pressure equation to find pressure
3) Solve momentum equations to find velocity field (approx)
4) Solve pressure correction equation and correct velocities
5) Iterate steps 2-4 until converged
6) Advance to the next time step using the current velocity field as the guessed velocity field for the next step

Implicit Runge-Kutta Methods

- Runge-Kutta SIMPLER Algorithm

1) Guess velocity field
2) Solve pressure equation to find pressure
3) Use Runge-Kutta to update the velocities
4) Advance to the next time step using the current velocity field as the guessed velocity field for the next step

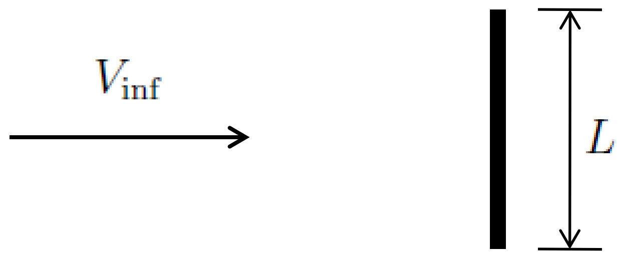
Removes iterations by updating the velocities in a more accurate way and removing the need for corrections

Implicit Runge-Kutta Methods

- Implementation
  - 2-D incompressible structured solver
  - Testing different implicit Runge-Kutta methods
    - EDIRK (2, 3, & 4 stage)
    - ESDIRK with Approximate Factorization & LU Factorization (2, 3, 4, & 5 stage)
    - Baseline comparisons are SIMPLER with C-N and 4-stage Explicit Runge-Kutta SIMPLER

Implicit Runge-Kutta Methods

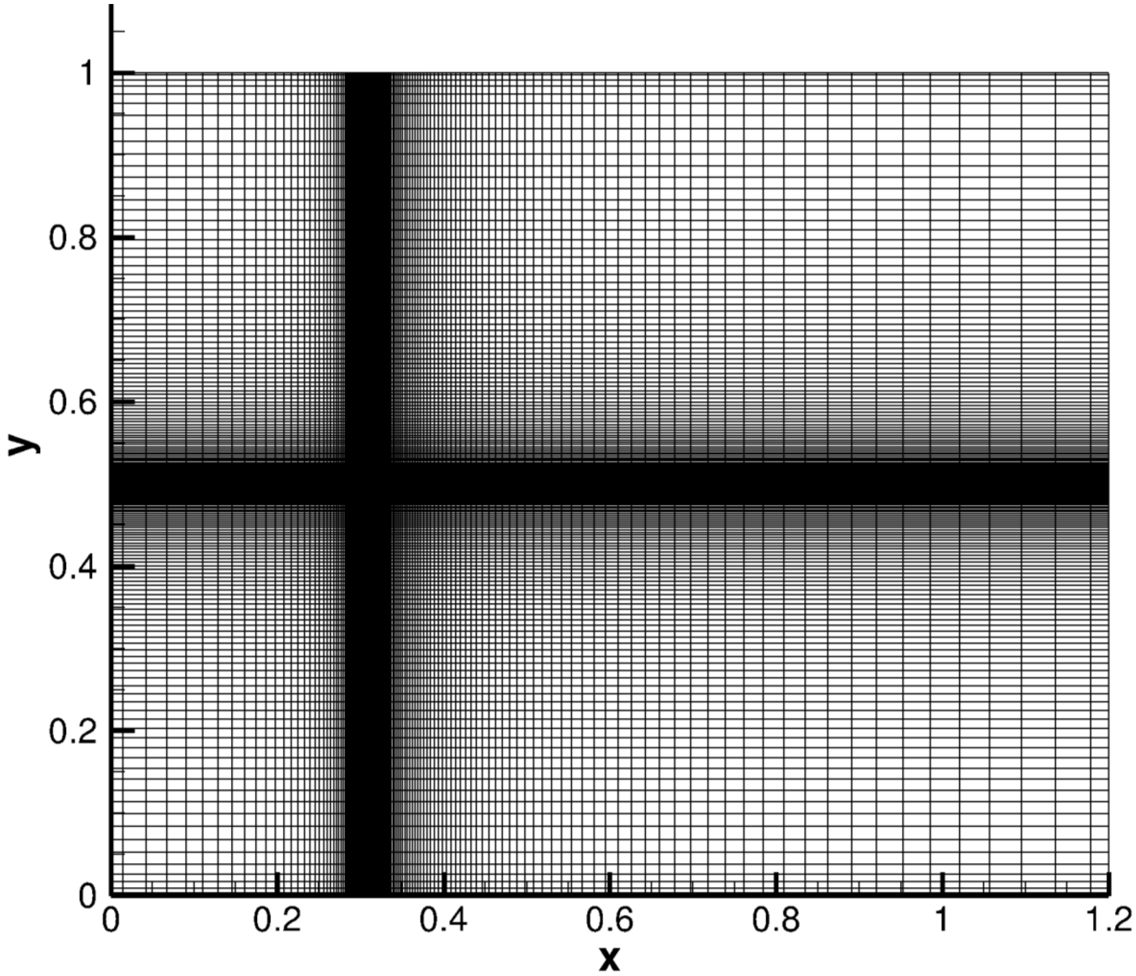
- Test Case
  - Flat Plate at 90 degrees to the flow

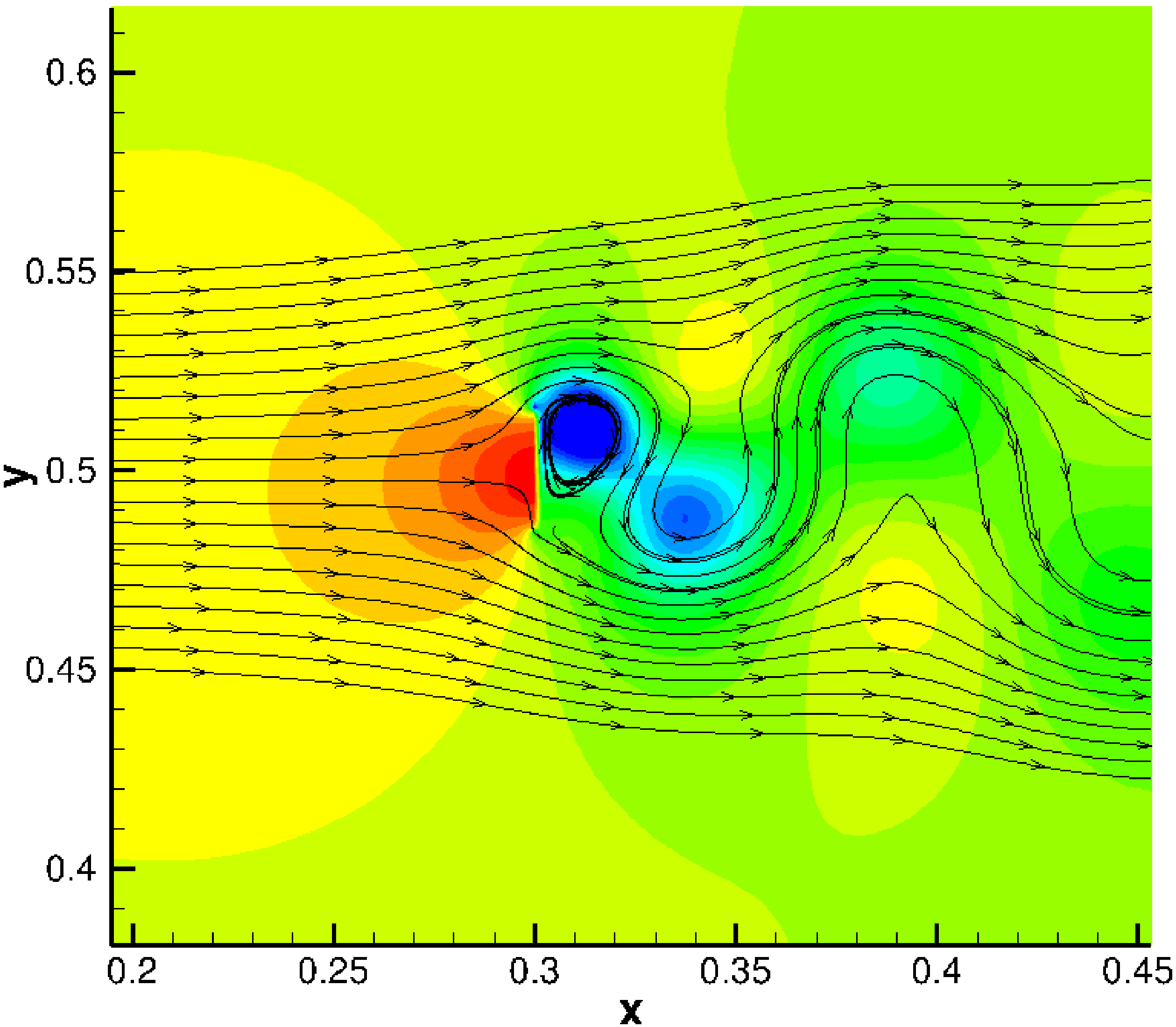


$$Re = \frac{\rho V_{\text{inf}} L}{\mu}$$

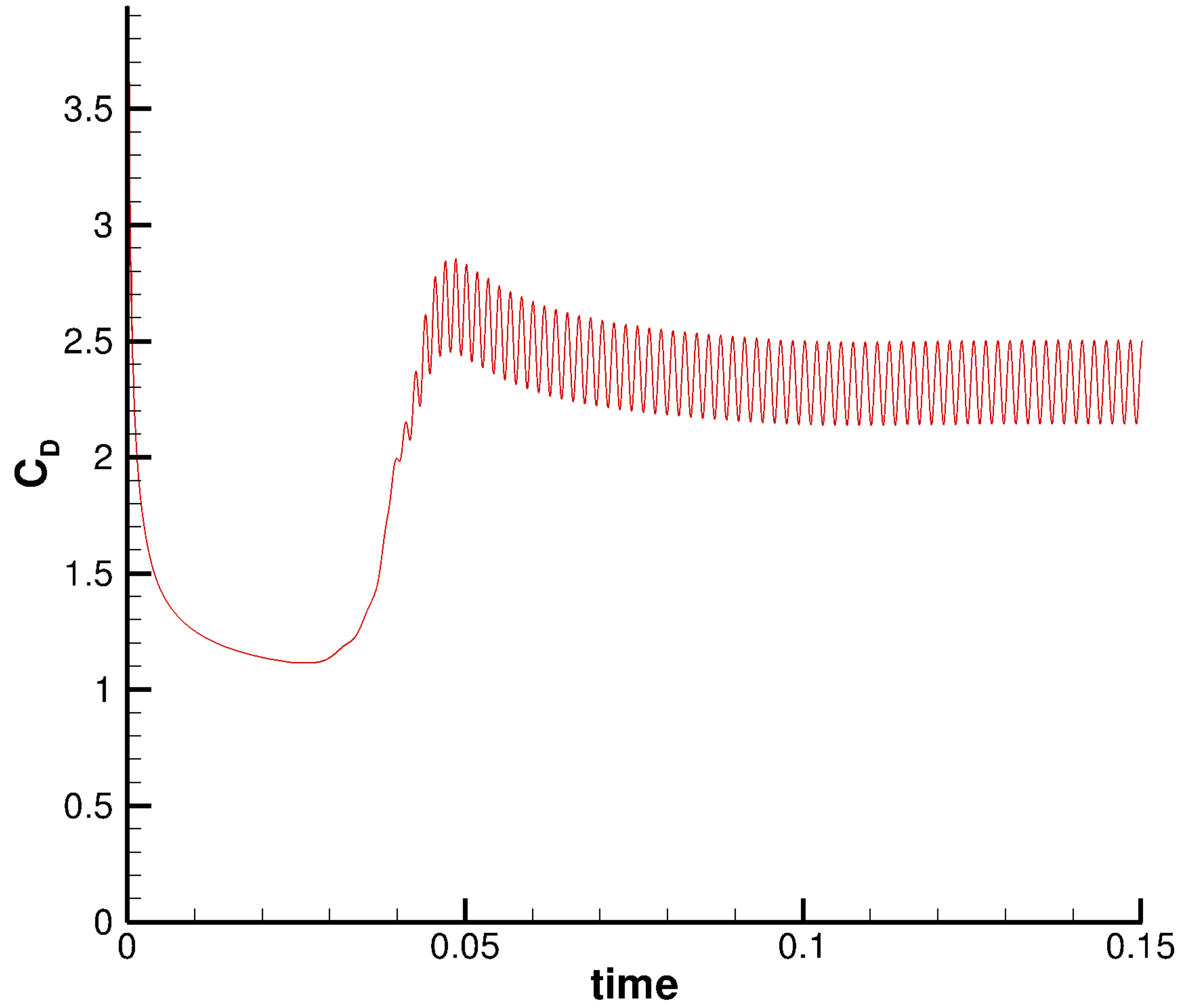
  - At high Reynolds numbers vortices are shed in a cyclic pattern

- 172 x 172 Structured Grid





0.6
0.55
0.5
0.45
0.4
y
0.2
0.25
0.3
0.35
0.4
0.45
x



3.5
3
2.5
2
1.5
1
0.5
0
C_D
0
0.05
0.1
0.15
time

Baseline Results

	SIMPLER C-N	ERK – 4
Max. time step	1.00 E-05	2.10 E-06
Run Time	5105 sec.	954 sec.
	~ 1hr. 15 min.	~ 16 min.
Decrease in Run time		81%
Period (1/s)	0.0036	0.0036
Avg. C_D	2.3	2.3

EDIRK Methods

	EDIRK – 4	EDIRK – 3	EDIRK – 2
Max. time step	6.53 E-06	6.41 E-06	6.50 E-6
time step / ERK time step	3.11	3.05	3.10
Run Time	1013 sec.	778 sec.	507 sec.
	~ 17 min.	~ 13 min.	~ 8.5 min.
Decrease in Run time vs. CN	80%	85%	90%
Decrease in Run time vs. ERK	-6%	18%	47%
Period (1/s)	0.0036	0.0036	0.0036
Avg. C_D	2.3	2.3	2.3

ESDIRK w/ AF & LU Fact. Methods

	ESDIRK – 5	ESDIRK – 4	ESDIRK – 3	ESDIRK – 2
Max. time step	6.53 E-06	6.55 E-06	6.46 E-6	6.67 E-6
time step / ERK time step	3.11	3.12	3.08	3.18
Run Time	564 sec.	506 sec.	438 sec.	365 sec.
	~ 9.5 min.	~ 8.5 min.	~ 7.5 min	~6 min.
Decrease in Run time vs. CN	89%	90%	91%	93%
Decrease in Run time vs. ERK	41%	47%	54%	62%
Period (1/s)	0.0036	0.0036	0.0036	0.0036
Avg. C_D	2.3	2.3	2.3	2.3

Conclusions

– Implicit Runge-Kutta can reduce run-time
– Fewer stages
    - Reduce number of computations
    - No loss in accuracy
– ESDIRK faster the EDIRK
    - Due to factorization
– EDIRK - 10% run-time of SIMPLER, 53% of ERK
– ESDIRK – 7% run-time of SIMPLER, 38% of ERK

Future Work

- Acceleration Techniques
  - Extend current implicit Runge-Kutta methods into 3D
  - Include rotor model in 3D and test with wind turbine
  - Multigrid
- Application
  - Solve wind energy problem with the use of CFD

Questions?