Massively Parallel Vortex Particle Simulations of Aircraft Wakes

Philippe Chatelain

with: M. Bergdorf, D. Rossinelli, P. Koumoutsakos and A. Curioni (IBM Research)
Outline

• Motivation
• Vortex Particle Method
• Medium Wavelength Instability
• Optimization of Wake Decay
• Conclusions
Motivation
Motivation

- **COMPUTATION**
  - Massively Parallel Simulations Using Particles
    - IBM/BG: Distributed memory (10⁴ - 10⁵ CPUS), with low RAM per node
    - Achieve sustained $O(\text{Tflops})$ performance
Motivation

- **COMPUTATION**
  - Massively Parallel Simulations Using Particles
    - IBM/BG: Distributed memory ($10^4 - 10^5$ CPUS), with low RAM per node
    - Achieve sustained $O$(Tflops) performance

- **PHYSICS**
  - Optimization of aircraft wake decay
    - aircraft & lift devices design
  - High Reynolds DNS
    - Capture accurate decay of vortical flows
    - Physical insight - Basis for turbulent models
Vortex Particle Method
Vortex Particle Method

• GOVERNING EQUATIONS
  • Navier-Stokes, incompressible

\[
\nabla \cdot \mathbf{u} = 0 \\
\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla p + \nu \Delta \mathbf{u}
\]
Vortex Particle Method

- GOVERNING EQUATIONS
  - Navier-Stokes, incompressible
    \[
    \nabla \cdot \mathbf{u} = 0
    \]
    \[
    \frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u}
    \]
  - Vorticity form
    \[
    \nabla \cdot \mathbf{u} = 0
    \]
    \[
    \frac{D\omega}{Dt} = (\omega \cdot \nabla)\mathbf{u} + \nu \Delta \omega
    \]
Vortex Particle Method

• **GOVERNING EQUATIONS**
  
  • Navier-Stokes, incompressible
  
  • Vorticity form
  
  • with velocity field

\[
\begin{align*}
\nabla \cdot \mathbf{u} &= 0 \\
\frac{D\mathbf{u}}{Dt} &= -\frac{1}{\rho} \nabla p + \nu \Delta \mathbf{u} \\
\nabla \cdot \mathbf{u} &= 0 \\
\frac{D\omega}{Dt} &= (\omega \cdot \nabla)\mathbf{u} + \nu \Delta \omega \\
\mathbf{u} &= \nabla \times \psi \\
\Delta \psi &= -\omega \\
\nabla \cdot \psi &= 0
\end{align*}
\]
Vortex Particle Method

- **GOVERNING EQUATIONS**
  - Navier-Stokes, incompressible
  - Vorticity form
  - with velocity field

- **DISCRETIZATION**

\[
\begin{align*}
\nabla \cdot \mathbf{u} &= 0 \\
\frac{D\mathbf{u}}{Dt} &= -\frac{1}{\rho} \nabla p + \nu \Delta \mathbf{u} \\
\nabla \cdot \mathbf{u} &= 0 \\
\frac{D\omega}{Dt} &= (\omega \cdot \nabla) \mathbf{u} + \nu \Delta \omega \\
\mathbf{u} &= \nabla \times \psi \\
\Delta \psi &= -\omega \\
\nabla \cdot \psi &= 0
\end{align*}
\]
Vortex Particle Method

• **GOVERNING EQUATIONS**
  
  - Navier-Stokes, incompressible
  
  - Vorticity form
  
  - with velocity field

• **DISCRETIZATION**
  
  - Particles: position $x_p$ and strength $\alpha_p = \int_{V_p} \omega \, dV \simeq \omega_p \, V_p$

\[
\begin{align*}
\nabla \cdot \mathbf{u} &= 0 \\
\frac{Du}{Dt} &= -\frac{1}{\rho} \nabla p + \nu \Delta \mathbf{u} \\
\nabla \cdot \mathbf{u} &= 0 \\
\frac{D\omega}{Dt} &= (\omega \cdot \nabla)\mathbf{u} + \nu \Delta \omega \\
\mathbf{u} &= \nabla \times \psi \\
\Delta \psi &= -\omega \\
\nabla \cdot \psi &= 0
\end{align*}
\]
Vortex Particle Method

- **GOVERNING EQUATIONS**
  - Navier-Stokes, incompressible
    \[
    \nabla \cdot u = 0 \\
    \frac{Du}{Dt} = -\frac{1}{\rho} \nabla p + \nu \Delta u
    \]
  - Vorticity form
    \[
    \nabla \cdot u = 0 \\
    \frac{D\omega}{Dt} = (\omega \cdot \nabla) u + \nu \Delta \omega
    \]
  - with velocity field
    \[
    u = \nabla \times \psi \\
    \Delta \psi = -\omega \\
    \nabla \cdot \psi = 0
    \]

- **DISCRETIZATION**
  - Particles: position \( x_p \) and strength \( \alpha_p = \int_{V_p} \omega \ dV \approx \omega_p \ V_p \)

- **EVOLUTION EQUATIONS**
  \[
  \frac{dx_p}{dt} = u(x_p) \\
  \frac{d\alpha_p}{dt} = ((\omega \cdot \nabla) u(x_p) + \nu \nabla^2 \omega(x_p)) \ V_p
  \]
Vortex Particle Method
Vortex Particle Method

- Particle methods **ARE NOT MESH-FREE**
Vortex Particle Method

- **ACCURATE** Particle methods ARE NOT MESH-FREE

  ➡ Particle distortion = loss of convergence

\[
\frac{D\Gamma_p}{Dt} = 0
\]

Euler Equations (2D : \(u-w\)) for an incompressible evolution of an axi-symmetric vortex patch.
Remeshed Particle Methods

\[ Q_p^{\text{new}} = \sum_{p'} Q_{p'} M(j h - x_{p'}) \]

Interpolation Kernel \( M(x) \)
- Moment conserving
- Tensorial Product of 1D kernels
Remesh Particle Methods

- Remesh: reinitialize particles onto regular locations

\[ Q_p^{\text{new}} = \sum_{p'} Q_{p'} M(jh - x_{p'}) \]

Interpolation Kernel \( M(x) \)
- Moment conserving
- Tensorial Product of 1D kernels
Remeshed Particle Methods

- **Remesh**: reinitialize particles onto regular locations

\[ Q_p^{\text{new}} = \sum_{p'} Q_{p'} M(j \ h - x_{p'}) \]

**Interpolation Kernel** \( M(x) \)

- **Moment** conserving
- **Tensorial Product of 1D kernels**
Remeshed Particle Methods

- **Remesh**: reinitialize particles onto regular locations

\[ Q_{p}^{\text{new}} = \sum_{p'} Q_{p'} M(j h - x_{p'}) \]

**Interpolation Kernel** \( M(x) \)
- Moment conserving
- Tensorial Product of 1D kernels

- Mesh also used for the efficient computation of **Right-Hand Side**
Vortex particle method
Vortex particle method

- Hybrid scheme
  - Mesh: RHS evaluations
Vortex particle method

- Hybrid scheme
  - Mesh: **RHS** evaluations
    - Differential operators (F.D.)
    - Fast Poisson solver (Fourier)
Vortex particle method

- Hybrid scheme
  - Mesh: **RHS** evaluations
    - Differential operators (F.D.)
    - Fast Poisson solver (Fourier)
  - Particles only handle **advection**

\[
\frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + \nabla \cdot (\omega u)
\]
Vortex particle method

- Hybrid scheme
  - Mesh: **RHS** evaluations
    - Differential operators (F.D.)
    - Fast Poisson solver (Fourier)
  - Particles only handle **advection**
    \[
    \frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + \nabla \cdot (\omega u)
    \]
  - Particles and Mesh communicate through interpolation

\[
\alpha_p \rightarrow \omega_{ij}
\]
Vortex particle method

- **Hybrid scheme**
- **Mesh:** RHS evaluations
  - Differential operators (F.D.)
  - Fast Poisson solver (Fourier)
- **Particles only handle advection**
  \[
  \frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + \nabla \cdot (\omega u)
  \]
- **Particles and Mesh communicate through interpolation**
Implementation

- Code is client of the PPM library
- Parallel Particle Mesh Library (Fortran 90, MPI)
- Client and library tuned for BG/L

Weak eff.: constant size per CPU

Strong eff.: constant total size


Sbalzarini et al, JCP 2006
Chatelain et al, CMAME 2008

IBM T. J. Watson Center, Yorktown Heights, NJ
IBM Zurich Research Laboratory
Implementation

- Code is client of the **PPM** library
- **Parallel Particle Mesh** Library (Fortran 90, MPI)
- Client and library *tuned* for BG/L

**Weak eff.: constant size per CPU**

**Strong eff.: constant total size**

**PPMers:** I. Sbalzarini, J. Walther, M. Bergdorf, P. Chatelain, S. Hieber, E. Kotsalis, P. Koumoutsakos, F. Milde, M. Quack, B. Hejazi Alhosseini

---

**IBMT J. Watson Center, Yorktown Heights, NJ**

**IBM Zurich Research Laboratory**

---

**Sbalzarini et al, JCP 2006**

**Chatelain et al, CMAME 2008**
Implementation

- Code is client of the **PPM** library
- **Parallel Particle Mesh** Library (Fortran 90, MPI)
- Client and library *tuned* for BG/L

**Weak eff.: constant size per CPU**

- Without Poisson solver
- ~ 400000 particles per CPU

**Strong eff.: constant total size**

- Per-CPU problem size

**First prod. run** $10^{10}$

**PPMers:** I. Sbalzarini, J. Walther, M. Bergdorf, P. Chatelain, S. Hieber, E. Kotsalis, P. Koumoutsakos, F. Milde, M. Quack, B. Hejazi Alhosseini

**IBM T. J. Watson Center, Yorktown Heights, NJ**
**IBM Zurich Research Laboratory**

Sbalzarini et al, JCP 2006
Chatelain et al, CMAME 2008
Medium Wavelength Instability
Medium Wavelength Instability

- Long domain
- Periodic in all directions
- Initiation by ambient noise

Ortega, J.M. et al., JFM 2003
Durston D.A. et al., J Aircraft 2005
Medium Wavelength Instability

- Long domain
- Periodic in all directions
- Initiation by ambient noise

**Physical parameters**
- $\text{Re} = 6,000$
- Circulation ratio $\Gamma_2/\Gamma_1 = -0.35$
- Span ratio $b_2/b_1 = 0.5$
- Domain $L_x = 10b_1$
- Time = 0.35
- Ambient white noise $u_{\text{RMS}} = 0.5\%$

**Numerical parameters**
- $\sim 1.6 \times 10^9$ particles
- $1024 \times 768 \times 2,048$ grid
- 10,000 time steps
- RK3 Low-storage
- 4th order FD

**CPU parameters**
- IBM BlueGene/L
- 4096 CPUs
- 100 hours

---

Ortega, J.M. et al., JFM 2003
Durston D.A. et al., J Aircraft 2005
Medium Wavelength Instability

- Spectra
  - Mode $\lambda/b_1 = 0.86 - 0.943$
  - Experimental $\lambda/b_1 = 0.9 - 1.3$
  - Bursts when $\Omega$-loop feet come together
- Cross-flow energy
  - Visualization: volume rendering of $|\omega|$

Rossinelli D. et al., SIGGRAPH08
Instability: Linear phase $t=0.21$
Instability: **Linear phase** $t=0.21$
Instability: \textbf{Reconnections} $t=0.25$
Instability: **Reconnections** $t=0.25$
Instability: **Propagation** $t=0.27$
Instability: Propagation $t=0.27$
Instability: Decay $t=0.35$
Instability: **Decay** $t=0.35$

- Large Vortex Pair
- Series of Vortex Rings
Optimization of Vortex Decay

• Find fastest decaying wake configuration
  • in terms of global end-result: energy, induced rolling moment,...
Optimization of Vortex Decay

• Find fastest decaying wake configuration
  • in terms of global end-result: energy, induced rolling moment,...

• Given, e.g.
  • a range in trim
  • active device perturbations
Optimization of Vortex Decay
Optimization of Vortex Decay

- Configuration
  - Co-rotating pairs
  - Re = 2500

Crouch, JFM 1997
Optimization of Vortex Decay

- Configuration
  - Co-rotating pairs
  - Re = 2500

- Problem: minimize objective function

\[ f_{\text{decay}} = \frac{\int_0^T E(t) \, dt}{E(0)T} \quad (T = 4) \]
Optimization of Vortex Decay

- Configuration
  - Co-rotating pairs
  - Re = 2500
- Problem: minimize objective function
  \[ f_{\text{decay}} = \frac{\int_0^T E(t) \, dt}{E(0)T} \quad (T = 4) \]
- Challenges
  - \( f \) can be multi-modal, non-convex, etc.
  - \( \nabla f \) not readily available
  - evaluation of \( f \) is expensive

Crouch., JFM 1997
Optimization of Vortex Decay

• Configuration
  • Co-rotating pairs
  • Re = 2500

• Problem: minimize objective function
  \[ f_{\text{decay}} = \frac{\int_0^T E(t) \, dt}{E(0)T} \quad (T = 4) \]

• Challenges
  • \( f \) can be multi-modal, non-convex, etc.
  • \( \nabla f \) not readily available
  • evaluation of \( f \) is expensive

• Approach
  • Evolutionary Optimization
  • Prior knowledge encoded in parametrization

Crouch, JFM 1997
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)
- **iterative methods** operating with **populations** of candidate solutions
- Here: Covariance Matrix Adaptation - ES


```
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)
- iterative methods operating with populations of candidate solutions
- Here: Covariance Matrix Adaptation - ES

```
Evolutionary Algorithms (EAs)

- **iterative methods** operating with **populations** of candidate solutions

- Here: Covariance Matrix Adaptation - ES

Optimization of Vortex Decay

Evolutionary Algorithms (EAs)
- iterative methods operating with populations of candidate solutions
- Here: Covariance Matrix Adaptation - ES

Optimization of Vortex Decay

Evolutionary Algorithms (EAs)
- **iterative methods** operating with populations of candidate solutions
- Here: Covariance Matrix Adaptation - ES

Evolutionary Algorithms (EAs)
- **iterative methods** operating with **populations** of candidate solutions
- Here: Covariance Matrix Adaptation - ES


Optimization of Vortex Decay

Selection

Evaluation

Parents $\{x_k\}_k^\mu$

Recombination

Mutation

Offspring $\{x_k\}_k^\lambda$

Stopping criteria fulfilled?

yes

no

$g = g + 1$

$x_1$

$x_2$

start with initial offspring population

$g = 0$

Initial offspring population

Optimization of Vortex Decay

Evolutionary Algorithms (EAs)
- **iterative methods** operating with **populations** of candidate solutions
- Here: Covariance Matrix Adaptation - ES

![Evolutionary Algorithm Diagram]

Optimization of Vortex Decay

Evolutionary Algorithms (EAs)
- **iterative methods** operating with **populations** of candidate solutions
- Here: Covariance Matrix Adaptation - ES

![Evolutionary Algorithm Diagram](http://www.cse-lab.ethz.ch/)

Evolutionary Algorithms (EAs)
- **iterative methods** operating with populations of candidate solutions
- Here: Covariance Matrix Adaptation - ES

Optimization of Vortex Decay

**Evolutionary Algorithms (EAs)**
- **iterative methods** operating with **populations** of candidate solutions
- Here: Covariance Matrix Adaptation - ES

- popular because of their flexibility and **robustness**

---

**Diagram: Evolutionary Algorithm Flowchart**

- **Selection**
- **Evaluation**
- **Recombination**
- **Mutation**
- **Offspring**

Starting with initial offspring population $\{x_k\}_{k=1}^{\lambda}$, the process evolves through the following steps:

1. **Stopping Criteria**
   - If fulfilled, stop.
   - If not, go to the next step.

2. **Parents** $\{x_k\}_{k=1}^{\mu}$
3. **Recombination**
4. **Mutation**
   - $x_k \sim P(\theta^{(g)})$

5. **Offspring** $\{x_k\}_{k=1}^{\lambda}$

The diagram also indicates:
- Probability distribution
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)
- **iterative methods** operating with **populations** of candidate solutions
- Here: Covariance Matrix Adaptation - ES

- popular because of their flexibility and **robustness**
- main **disadvantage**: Need **large number of objective function evaluations**

Optimization of Vortex Decay

\[ f \]
Optimization of Vortex Decay

- PRESENT
- After sweep toward large $\lambda$, moves toward smaller values
- Convergence
Optimization of Vortex Decay

- **PRESENT**
  - After sweep toward large $\lambda$, moves toward smaller values
  - Convergence
Optimization of Vortex Decay

- **PRESENT**
  - After sweep toward large $\lambda$, moves toward smaller values
  - Convergence
Optimization of Vortex Decay

- **PRESENT**
  - After sweep toward large $\lambda$, moves toward smaller values
  - Convergence
Optimization of Vortex Decay

- **PRESENT**
  - After sweep toward large $\lambda$, moves toward smaller values
- **Convergence**
- **PROBLEMS**
  - Resolution of Optimized Flows?
  - Synchronize Optimization and Numerics
Optimization of Vortex Decay

- FUNCTION LANDSCAPE
- in \((\lambda, \delta, \Gamma)\) hyperplane
- Multi-modality
- Effect of numerics?
Optimization of Vortex Decay

- FUNCTION LANDSCAPE
- in $(\lambda, \delta, \Gamma)$ hyperplane
- Multi-modality
- Effect of numerics?
Ongoing work

• Development

• **Bigger, faster:**
  Mixed MPI/SMP implementation of library
  • multicore machines: BlueGene/P

• Smarter:
  Unbounded boundary conditions

• More physical:
  Non-periodic streamwise boundary conditions
  • spatially developing wake
Conclusions

- Implementation of Vortex Particle Method on massively parallel architecture
  - Scalability
- Large-scale High Re DNS
- Coupling of DNS code with Evolutionary Strategy: Optimization of wake decay
Acknowledgements

M. Bergdorf
D. Rossinelli
S. Kern
P. Koumoutsakos
M. Gazzola
M. Quack
A. Curioni

IBM Zurich Research Center
IBM T. J. Watson Center
Swiss Super-Computing Center