Massively Parallel Vortex Particle Simulations of Aircraft Wakes

Philippe Chatelain

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

CSE Lab
Computational Science & Engineering Laboratory
http://www.icos.ethz.ch/cse

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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 (10^4 - 10^5 CPUS), with low RAM per node
    - Achieve sustained *O*(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

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{D\mathbf{u}}{Dt} &= -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} \end{aligned}$$

Vortex Particle Method

- **GOVERNING EQUATIONS**
  - Navier-Stokes, incompressible

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{D\mathbf{u}}{Dt} &= -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} \end{aligned}$$

  - Vorticity form

$$\boxed{\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \\ &\frac{D\omega}{Dt} = (\omega \cdot \nabla)\mathbf{u} + \nu \Delta \omega \end{aligned}}$$

Vortex Particle Method

- **GOVERNING EQUATIONS**
  - Navier-Stokes, incompressible

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{D\mathbf{u}}{Dt} &= -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} \end{aligned}$$

  - Vorticity form

$$\boxed{\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \\ &\frac{D\omega}{Dt} = (\omega \cdot \nabla)\mathbf{u} + \nu \Delta \omega \end{aligned}}$$

  - with velocity field

$$\boxed{\mathbf{u} = \nabla \times \psi \qquad \begin{aligned} \Delta \psi &= -\omega \\ \nabla \cdot \psi &= 0 \end{aligned}}$$

Vortex Particle Method

- **GOVERNING EQUATIONS**
  - Navier-Stokes, incompressible

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{D\mathbf{u}}{Dt} &= -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} \end{aligned}$$

  - Vorticity form

$$\boxed{\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \\ &\frac{D\omega}{Dt} = (\omega \cdot \nabla)\mathbf{u} + \nu \Delta \omega \end{aligned}}$$

  - with velocity field

$$\boxed{\mathbf{u} = \nabla \times \psi \qquad \begin{aligned} \Delta \psi &= -\omega \\ \nabla \cdot \psi &= 0 \end{aligned}}$$

- **DISCRETIZATION**

Vortex Particle Method

- **GOVERNING EQUATIONS**
  - Navier-Stokes, incompressible

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{D\mathbf{u}}{Dt} &= -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} \end{aligned}$$

  - Vorticity form

$$\boxed{\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \\ &\frac{D\omega}{Dt} = (\omega \cdot \nabla)\mathbf{u} + \nu \Delta \omega \end{aligned}}$$

  - with velocity field

$$\boxed{\mathbf{u} = \nabla \times \psi \qquad \begin{aligned} \Delta \psi &= -\omega \\ \nabla \cdot \psi &= 0 \end{aligned}}$$

- **DISCRETIZATION**
  - Particles: position $\boldsymbol{x}_p$ and strength $\boldsymbol{\alpha}_p = \int_{V_p} \boldsymbol{\omega}\, dV \simeq \boldsymbol{\omega}_p\, V_p$

Vortex Particle Method

- **GOVERNING EQUATIONS**
  - Navier-Stokes, incompressible

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{D\mathbf{u}}{Dt} &= -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} \end{aligned}$$

  - Vorticity form

$$\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \\ &\frac{D\omega}{Dt} = (\omega \cdot \nabla)\mathbf{u} + \nu \Delta \omega \end{aligned}$$

  - with velocity field

$$\mathbf{u} = \nabla \times \psi \qquad \begin{aligned} \Delta \psi &= -\omega \\ \nabla \cdot \psi &= 0 \end{aligned}$$

- **DISCRETIZATION**
  - Particles: position $\boldsymbol{x}_p$ and strength $\boldsymbol{\alpha}_p = \int_{V_p} \boldsymbol{\omega}\, dV \simeq \boldsymbol{\omega}_p\, V_p$
- **EVOLUTION EQUATIONS**

$$\begin{aligned} \frac{d\mathbf{x}_p}{dt} &= \mathbf{u}(\mathbf{x}_p) \\ \frac{d\boldsymbol{\alpha}_p}{dt} &= \left((\boldsymbol{\omega} \cdot \nabla)\mathbf{u}(\mathbf{x}_p) + \nu \nabla^2 \boldsymbol{\omega}(\mathbf{x}_p)\right) V_p \end{aligned}$$

Vortex Particle Method

Vortex Particle Method

- Particle methods ARE NOT MESH-FREE

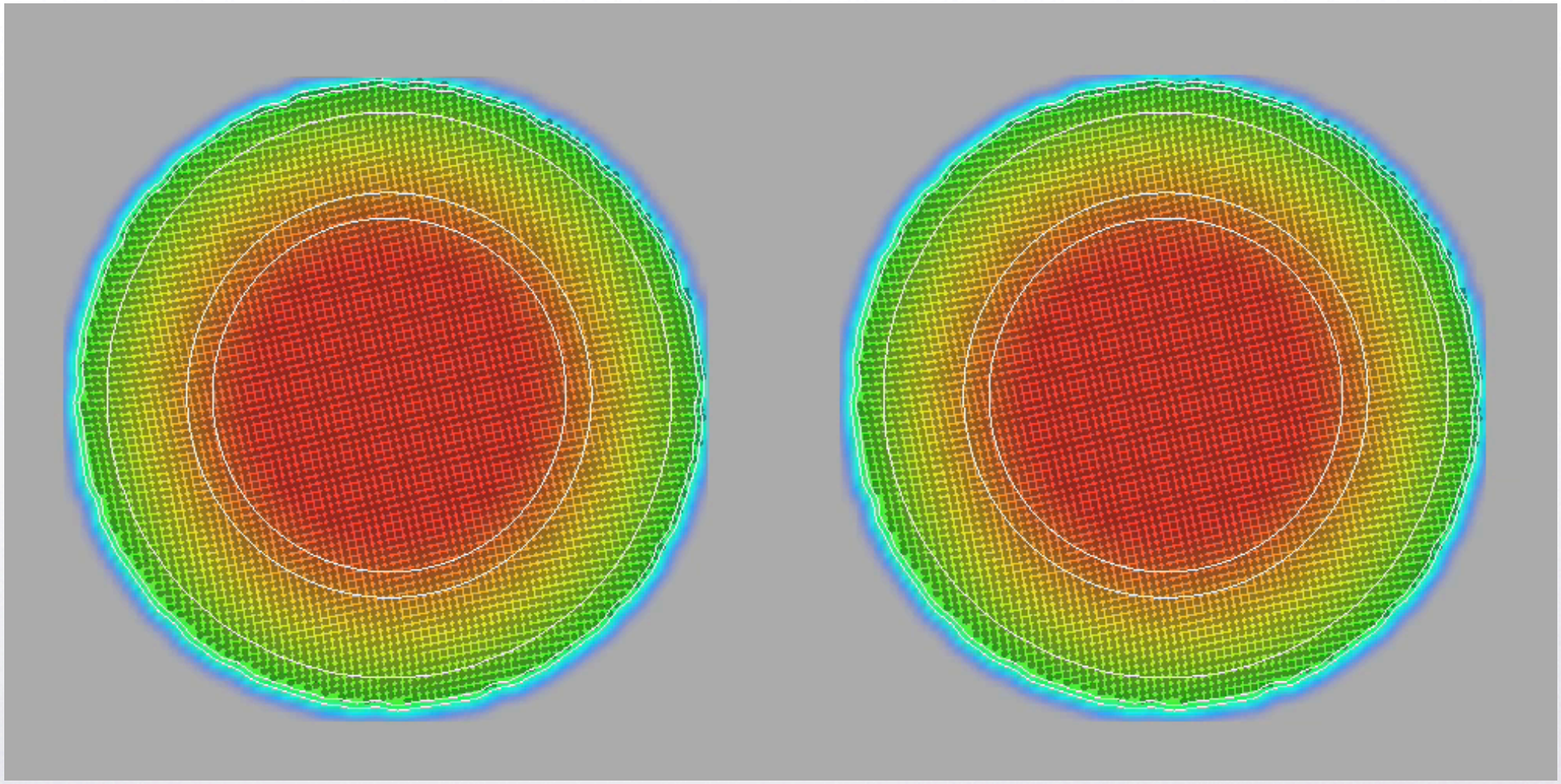
Vortex Particle Method

- 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

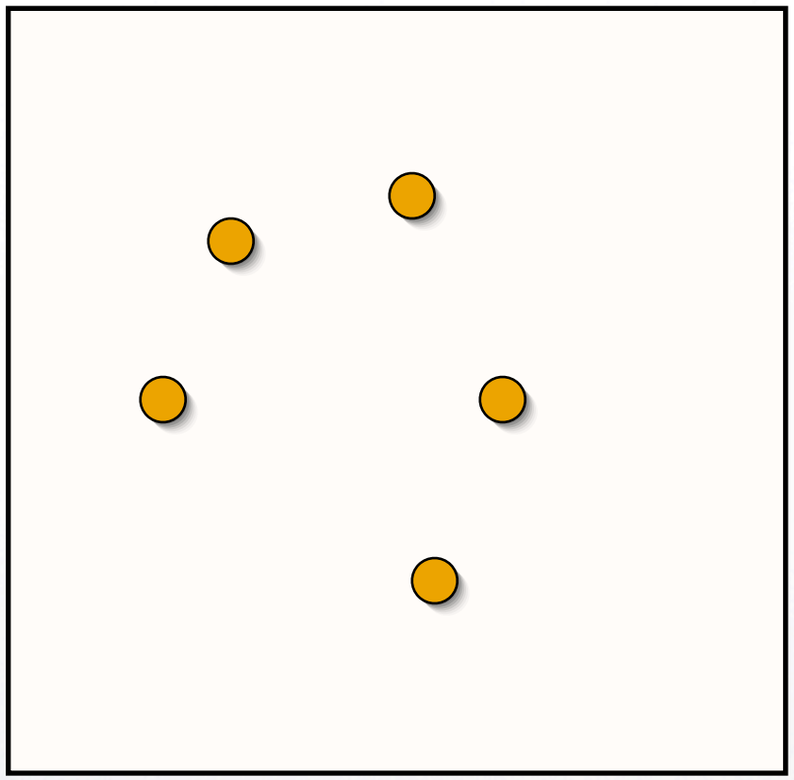
$$Q_p^{\text{new}} = \sum_{p'} Q_{p'} M(j\,h - x_{p'})$$

Interpolation Kernel $M(x)$

- **Moment** conserving
- Tensorial Product of 1D kernels

Remeshed Particle Methods

- *Remesh* : reinitialize particles onto regular locations



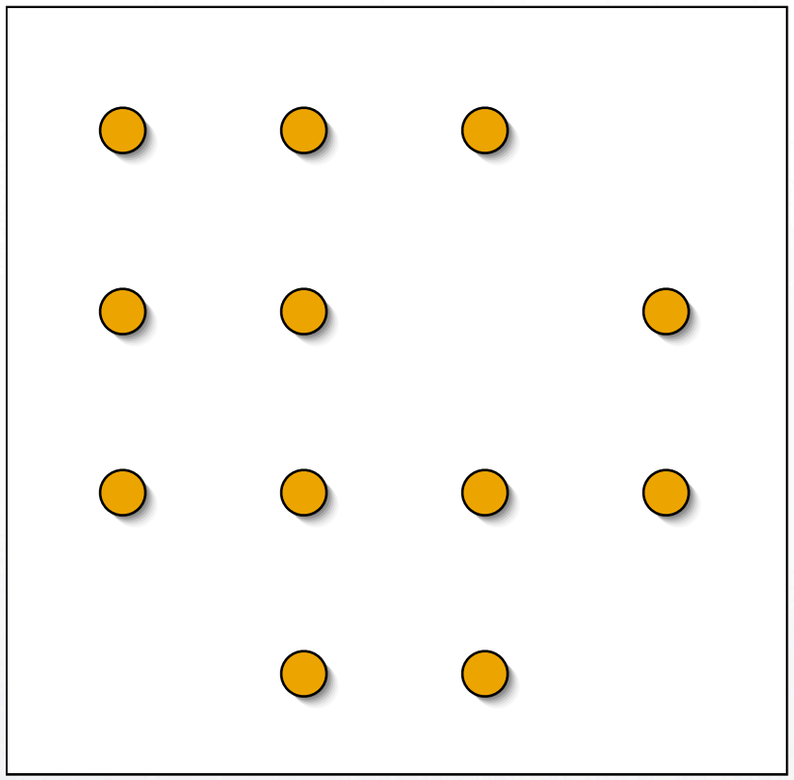
$$Q_p^{\text{new}} = \sum_{p'} Q_{p'} M(j\,h - x_{p'})$$

Interpolation Kernel $M(x)$

- **Moment** conserving
- Tensorial Product of 1D kernels

Remeshed Particle Methods

- *Remesh* : reinitialize particles onto regular locations



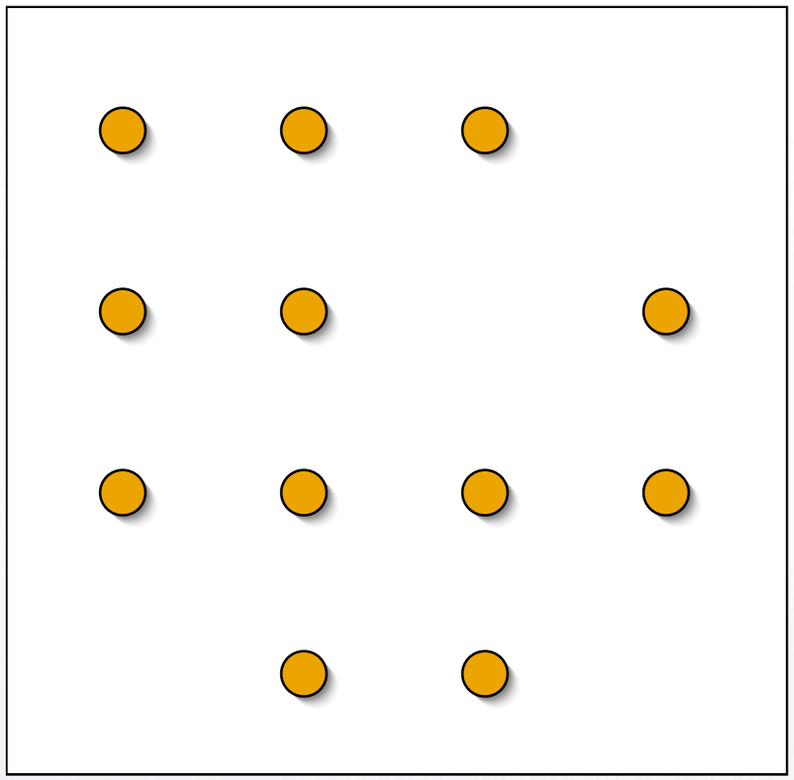
$$Q_p^{\text{new}} = \sum_{p'} Q_{p'} M(j\,h - x_{p'})$$

Interpolation Kernel $M(x)$

- **Moment** conserving
- Tensorial Product of 1D kernels

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**

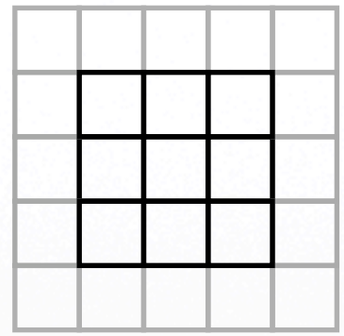
Vortex particle method

Vortex particle method

- Hybrid scheme
  - Mesh: **RHS** evaluations

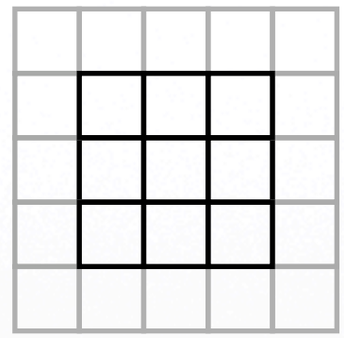
Vortex particle method

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



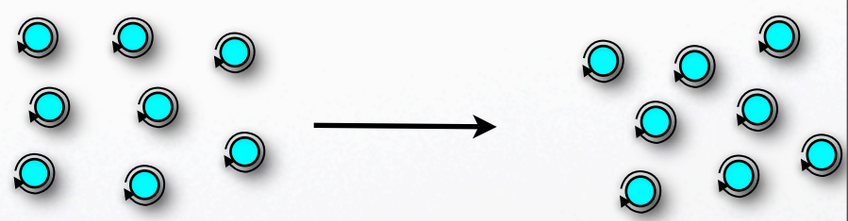
Vortex particle method

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



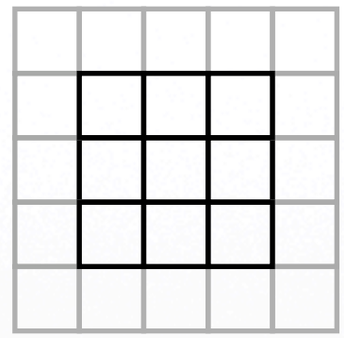
  - Particles only handle **advection**

$$\frac{\mathcal{D}\boldsymbol{\omega}}{\mathcal{D}t} = \frac{\partial\boldsymbol{\omega}}{\partial t} + \nabla\cdot(\boldsymbol{\omega}\mathbf{u})$$



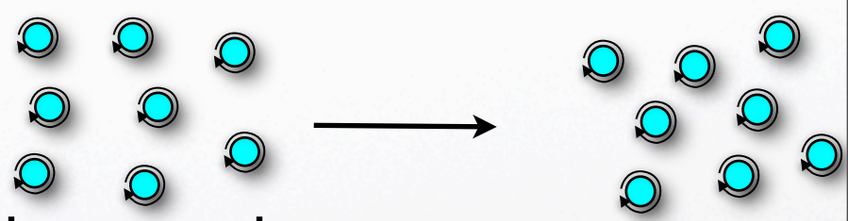
Vortex particle method

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

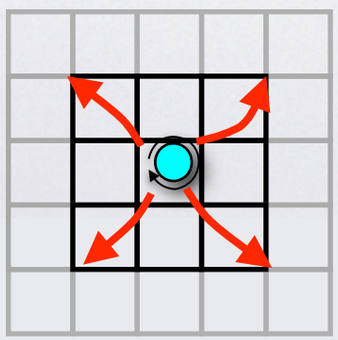


  - Particles only handle **advection**

$$\frac{\mathcal{D}\boldsymbol{\omega}}{\mathcal{D}t} = \frac{\partial\boldsymbol{\omega}}{\partial t} + \nabla\cdot(\boldsymbol{\omega}\mathbf{u})$$



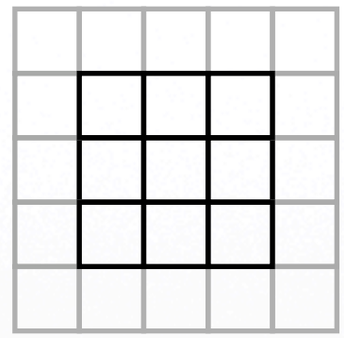
  - Particles and Mesh communicate through **interpolation**



$$\boldsymbol{\alpha}_p \rightarrow \boldsymbol{\omega}_{ij}$$

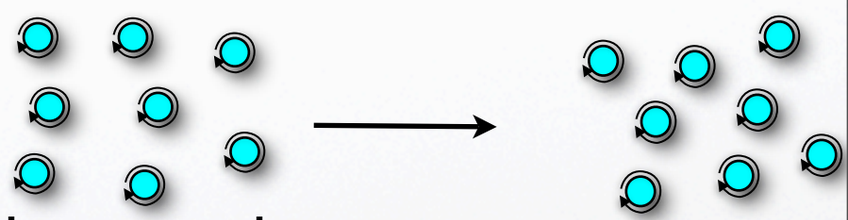
Vortex particle method

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

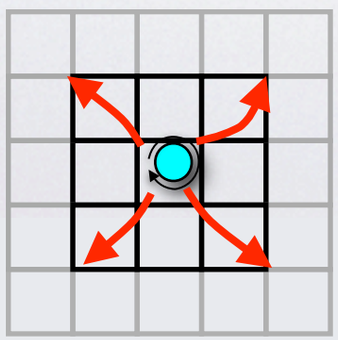


  - Particles only handle **advection**

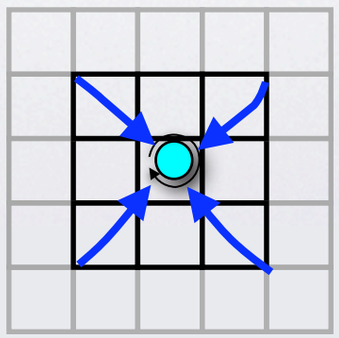
$$\frac{\mathcal{D}\boldsymbol{\omega}}{\mathcal{D}t} = \frac{\partial\boldsymbol{\omega}}{\partial t} + \nabla\cdot(\boldsymbol{\omega}\mathbf{u})$$



  - Particles and Mesh communicate through **interpolation**



$\boldsymbol{\alpha}_p \rightarrow \boldsymbol{\omega}_{ij}$



$$\begin{matrix}\mathbf{u}_{ij}\\ (\boldsymbol{\omega}\cdot\nabla)\mathbf{u}_{ij}\\ \Delta\boldsymbol{\omega}_{ij}\end{matrix} \rightarrow \begin{matrix}\mathbf{u}_{p}\\ (\boldsymbol{\omega}\cdot\nabla)\mathbf{u}_{p}\\ \Delta\boldsymbol{\omega}_{p}\end{matrix}$$

Implementation

www.cse-lab.ethz.ch/software.html

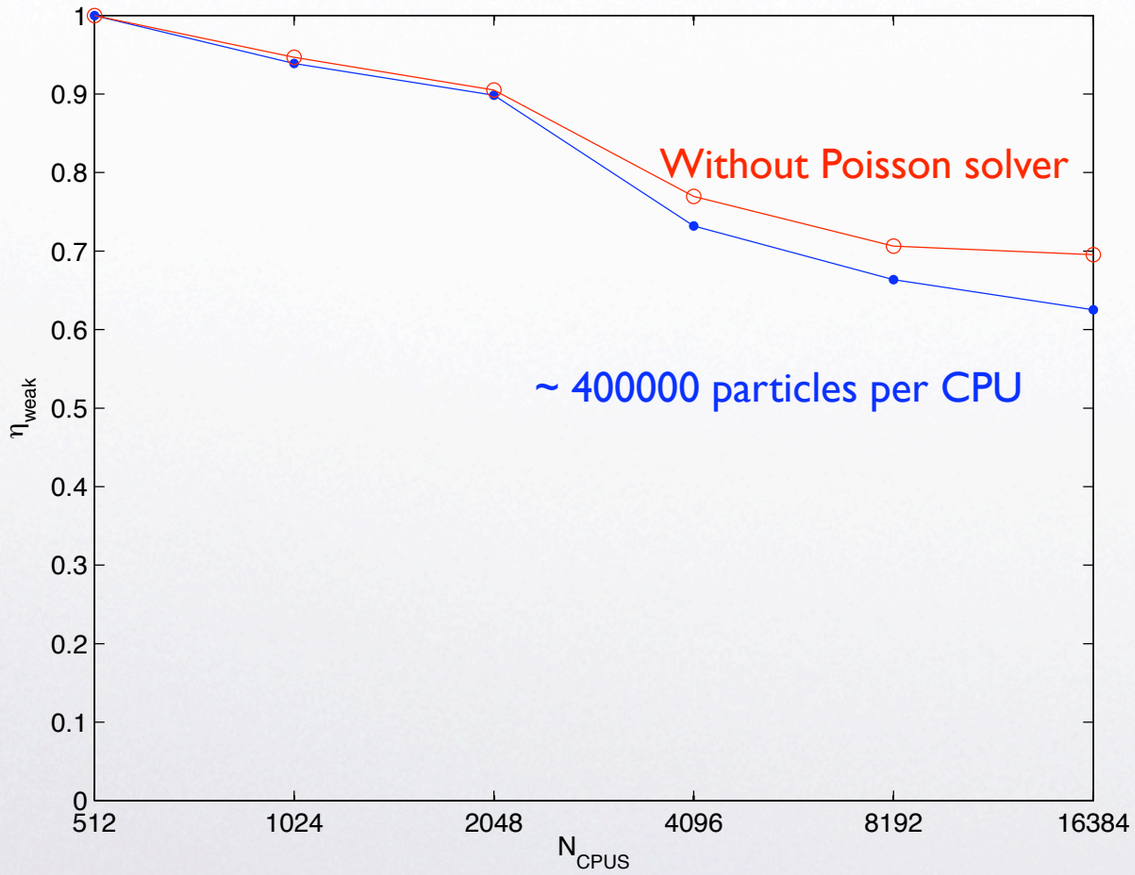
- Code is client of the **PPM** library

Sbalzarini et al, JCP 2006

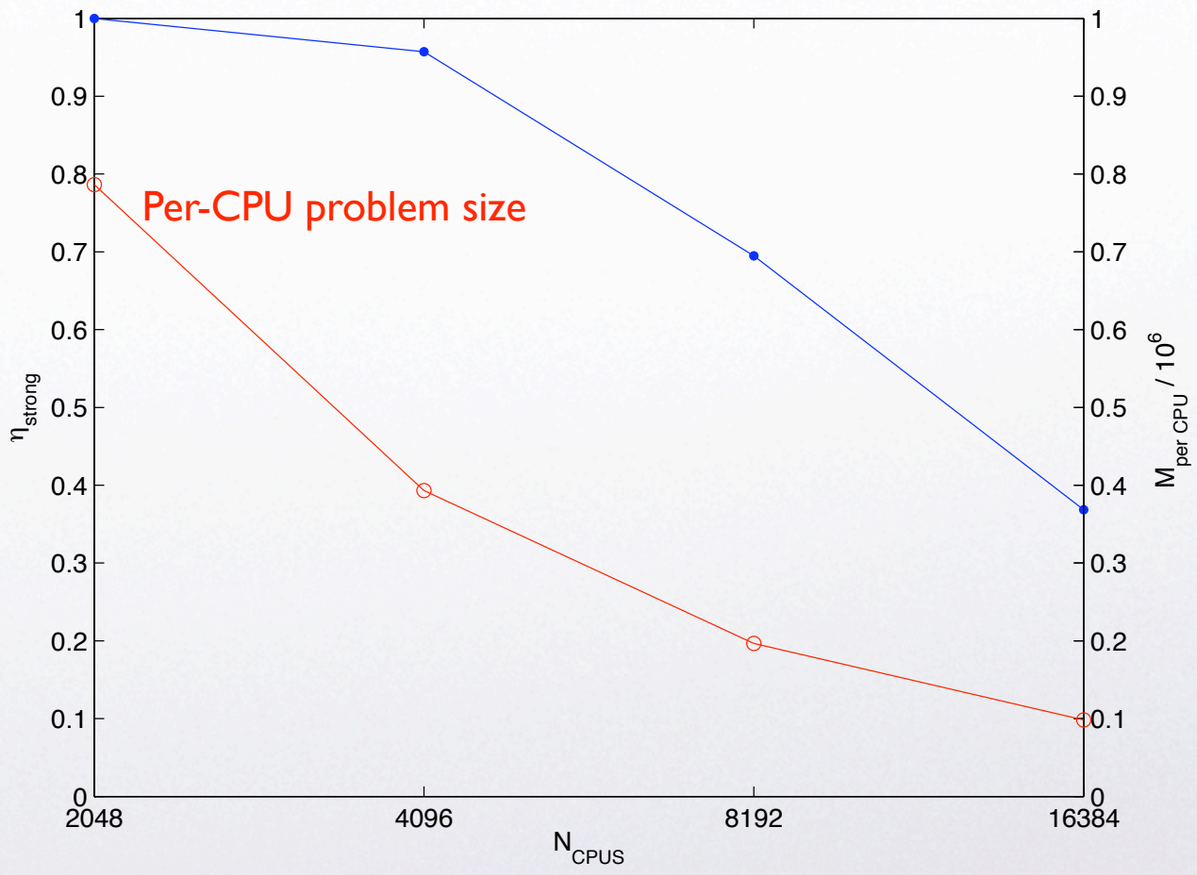
- **P**arallel **P**article **M**esh Library (Fortran 90, MPI)
- Client and library *tuned* for BG/L

Chatelain et al, CMAME 2008

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

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

Implementation

www.cse-lab.ethz.ch/software.html

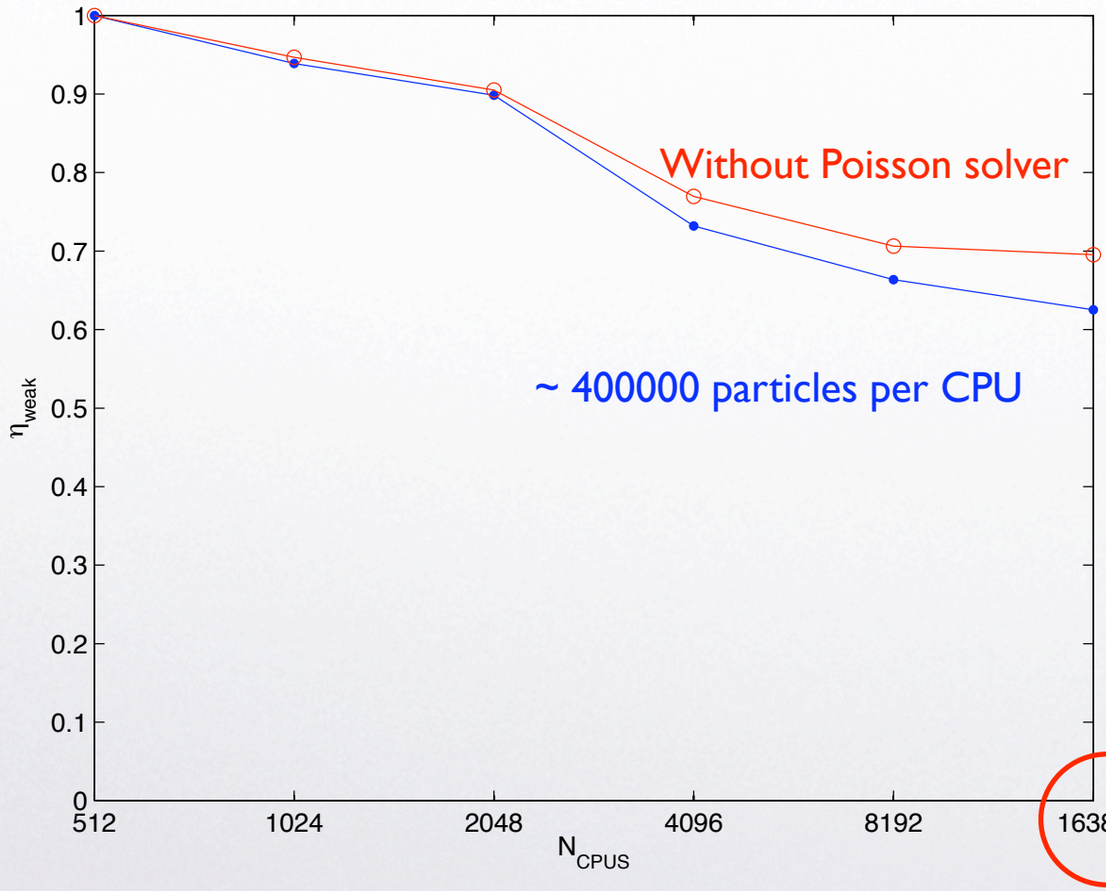
- Code is client of the **PPM** library

Sbalzarini et al, JCP 2006

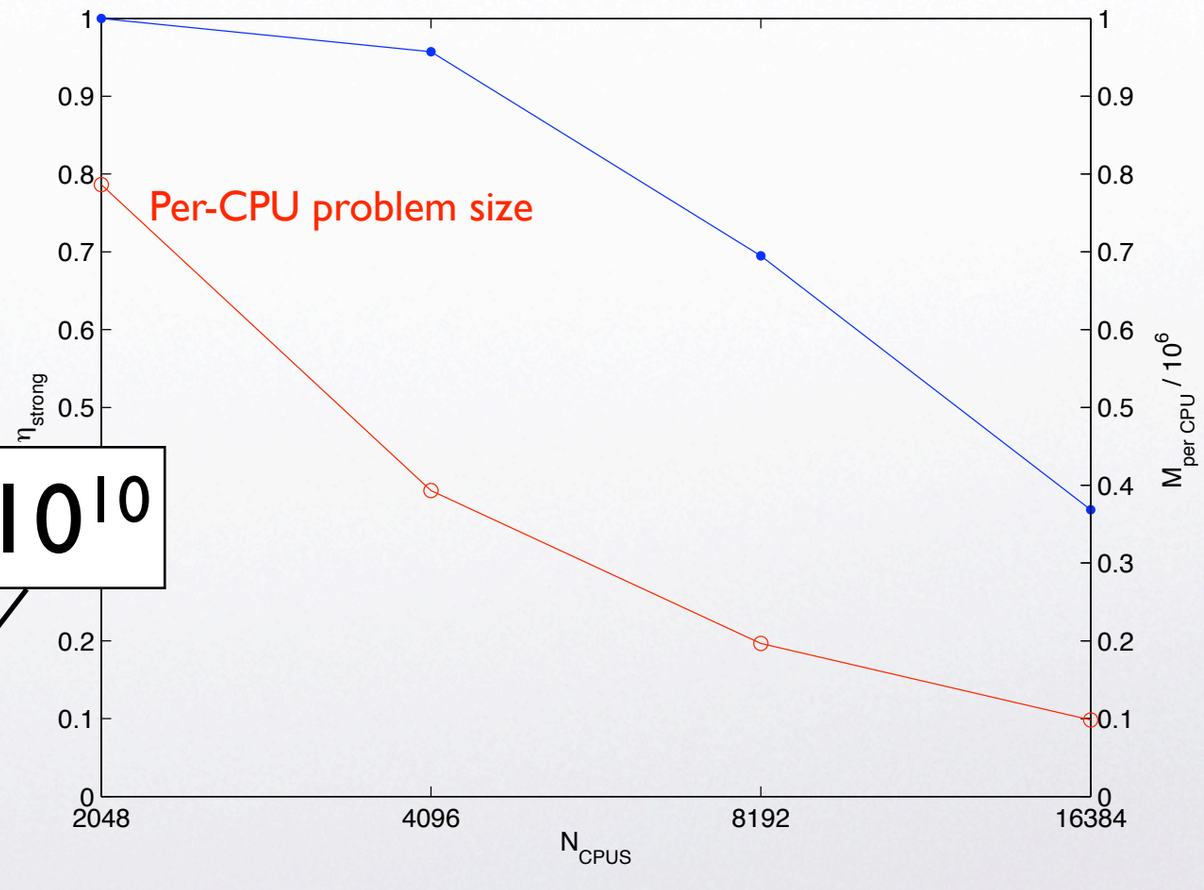
- **P**arallel **P**article **M**esh Library (Fortran 90, MPI)
- Client and library *tuned* for BG/L

Chatelain et al, CMAME 2008

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

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

Implementation

www.cse-lab.ethz.ch/software.html

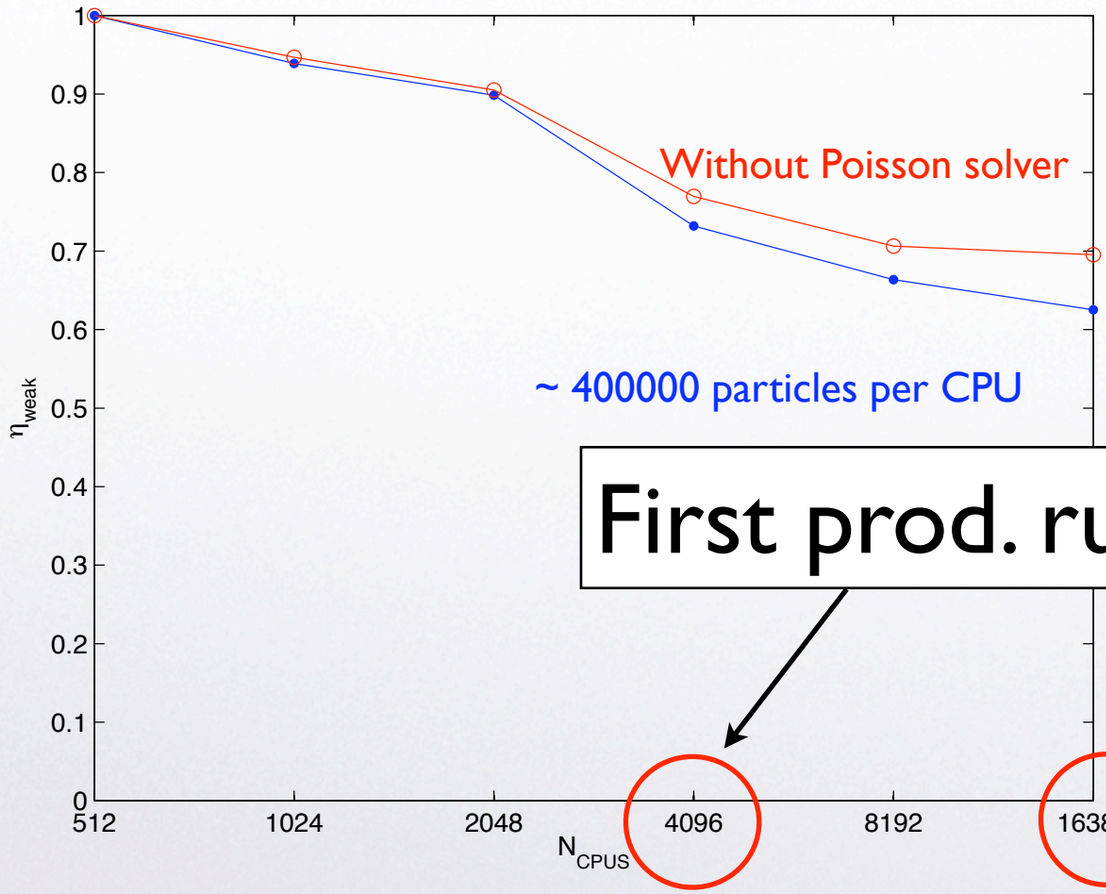
- Code is client of the **PPM** library

Sbalzarini et al, JCP 2006

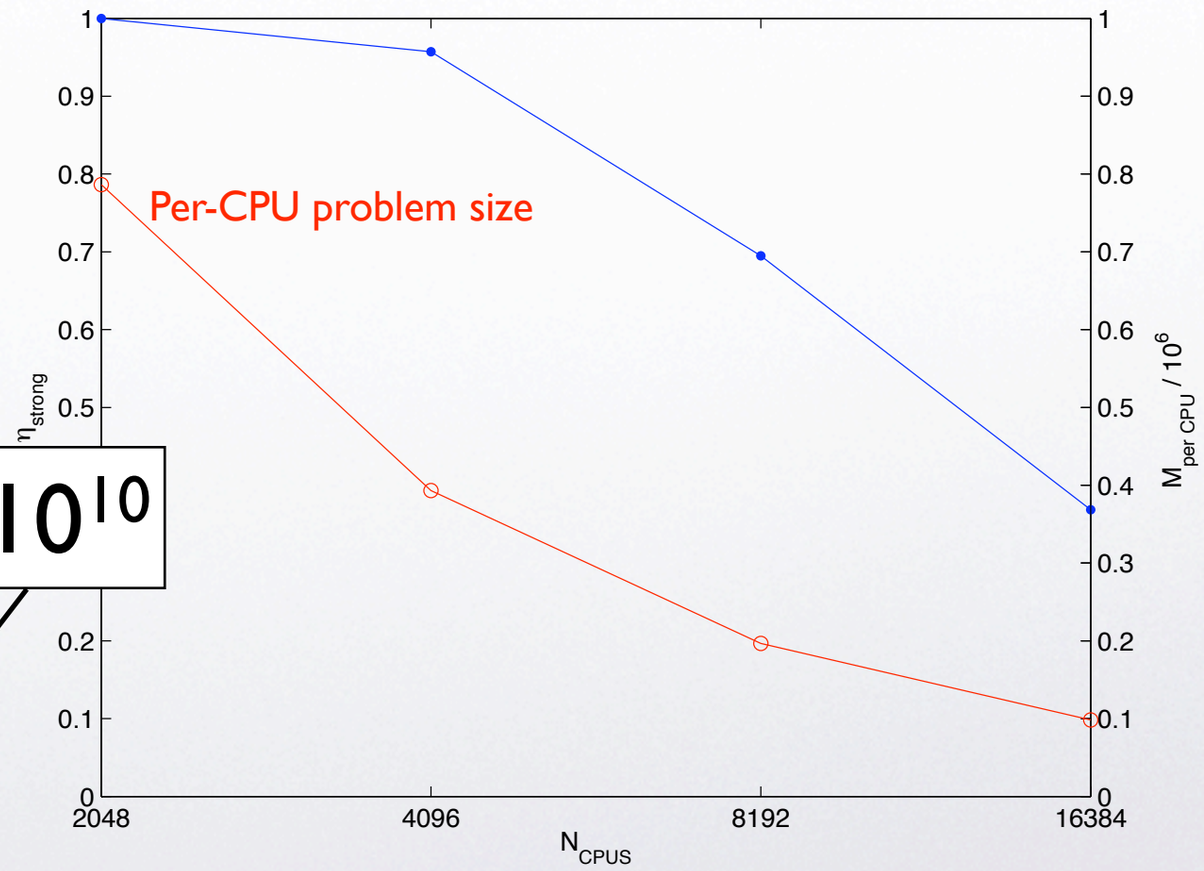
- **P**arallel **P**article **M**esh Library (Fortran 90, MPI)
- Client and library *tuned* for BG/L

Chatelain et al, CMAME 2008

Weak eff.: constant size per CPU



Strong eff.: constant total size



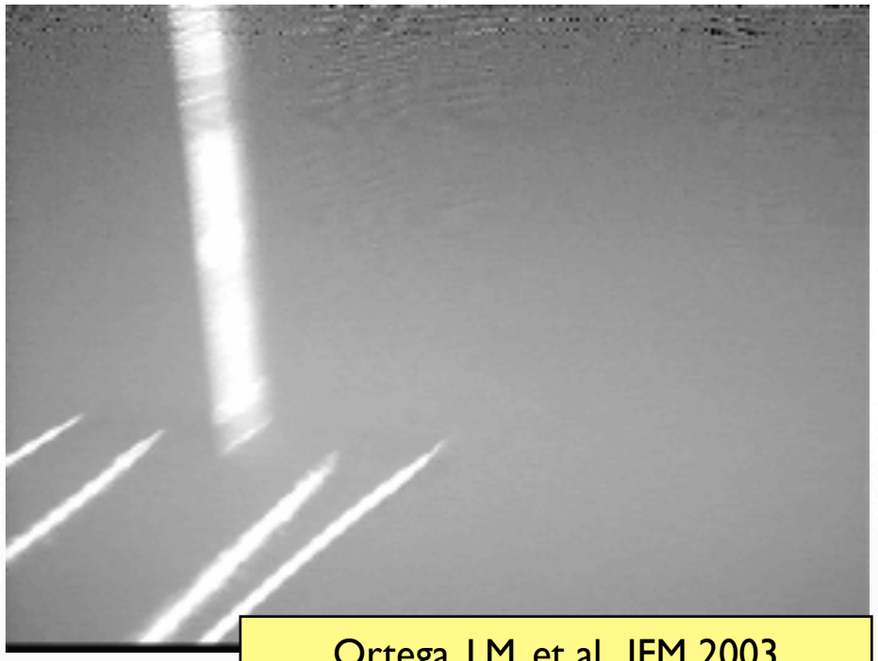
PPMers: I. Sbalzarini, J. Walther, M.Bergdorf, P. Chatelain, S. Hieber, E. Kotsalis, P. Koumoutsakos, F. Milde, M. Quack, B. Hejazi Alhosseini

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

Medium Wavelength Instability

Medium Wavelength Instability

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

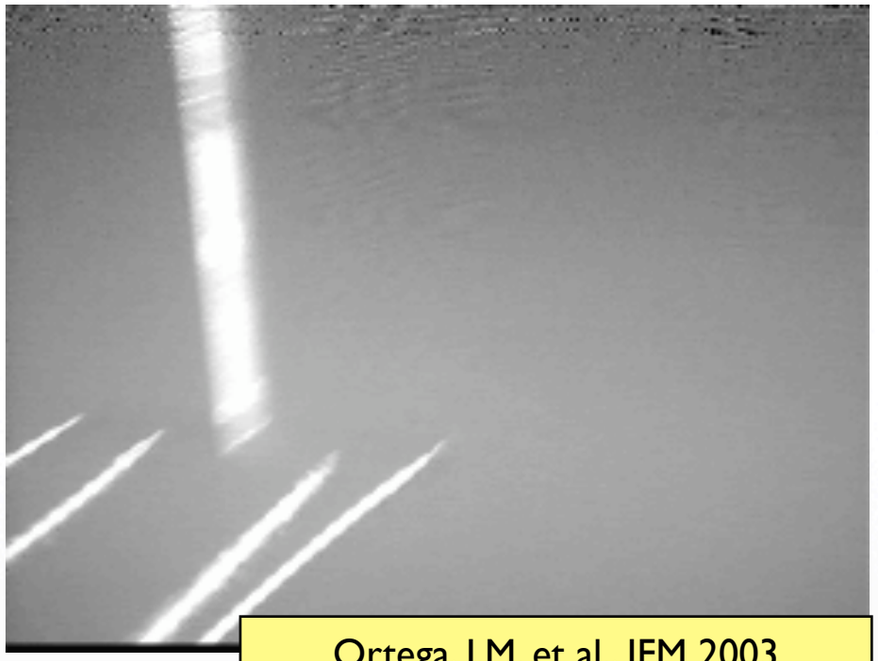


Ortega, J.M. et al., JFM 2003

Durston D.A. et al., J Aircraft 2005

Medium Wavelength Instability

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



Ortega, J.M. et al., JFM 2003

Durston D.A. et al., J Aircraft 2005

Physical parameters

- Re_Γ = 6,000
- Circulation ratio Γ_2/Γ_1 = -0.35
- Span ratio b_2/b_1 = 0.5
- Domain $L_x = 10\ b_1$
- Time = 0.35
- Ambient white noise u_{RMS}= 0.5%

Numerical parameters

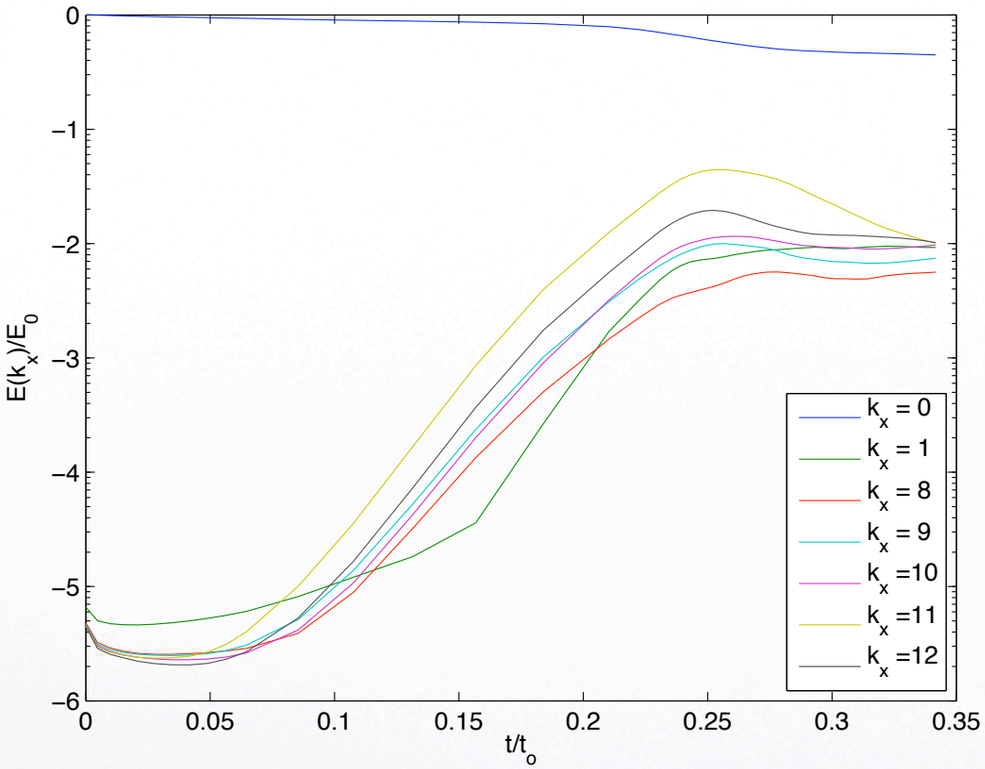
- ~1.6 10^9 particles
- 1024 x 768 x 2,048 grid
- 10,000 time steps
- RK3 Low-storage
- 4th order FD

CPU parameters

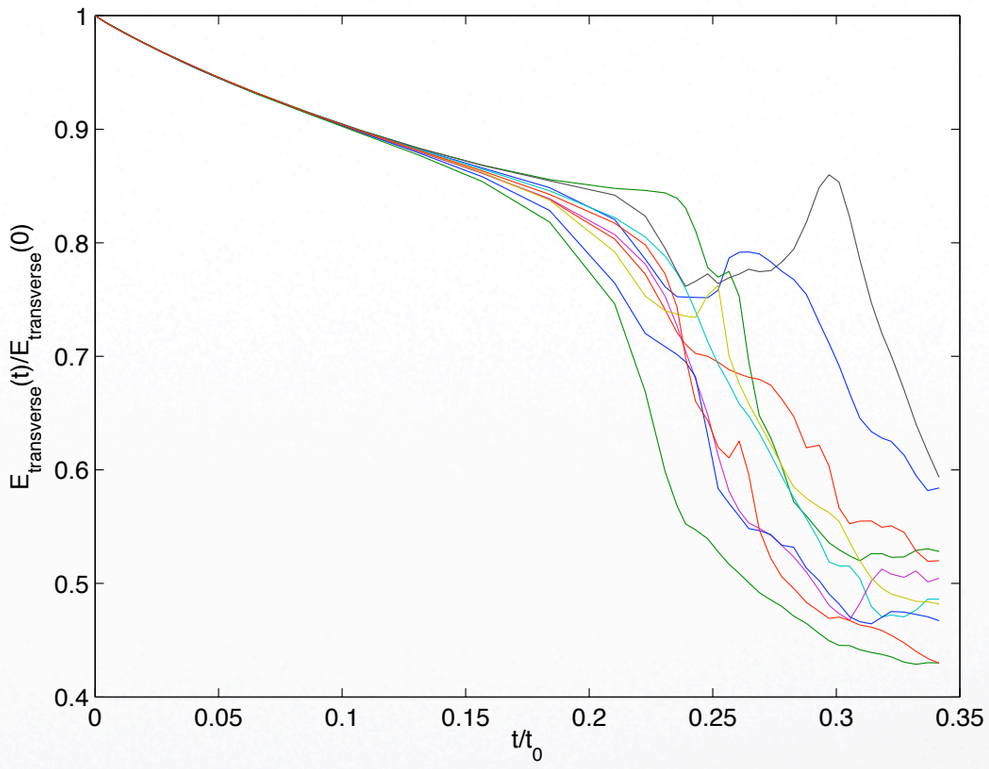
- IBM BlueGene/L
- 4096 CPUS
- 100 hours

Medium Wavelength Instability

- Spectra



- Cross-flow energy

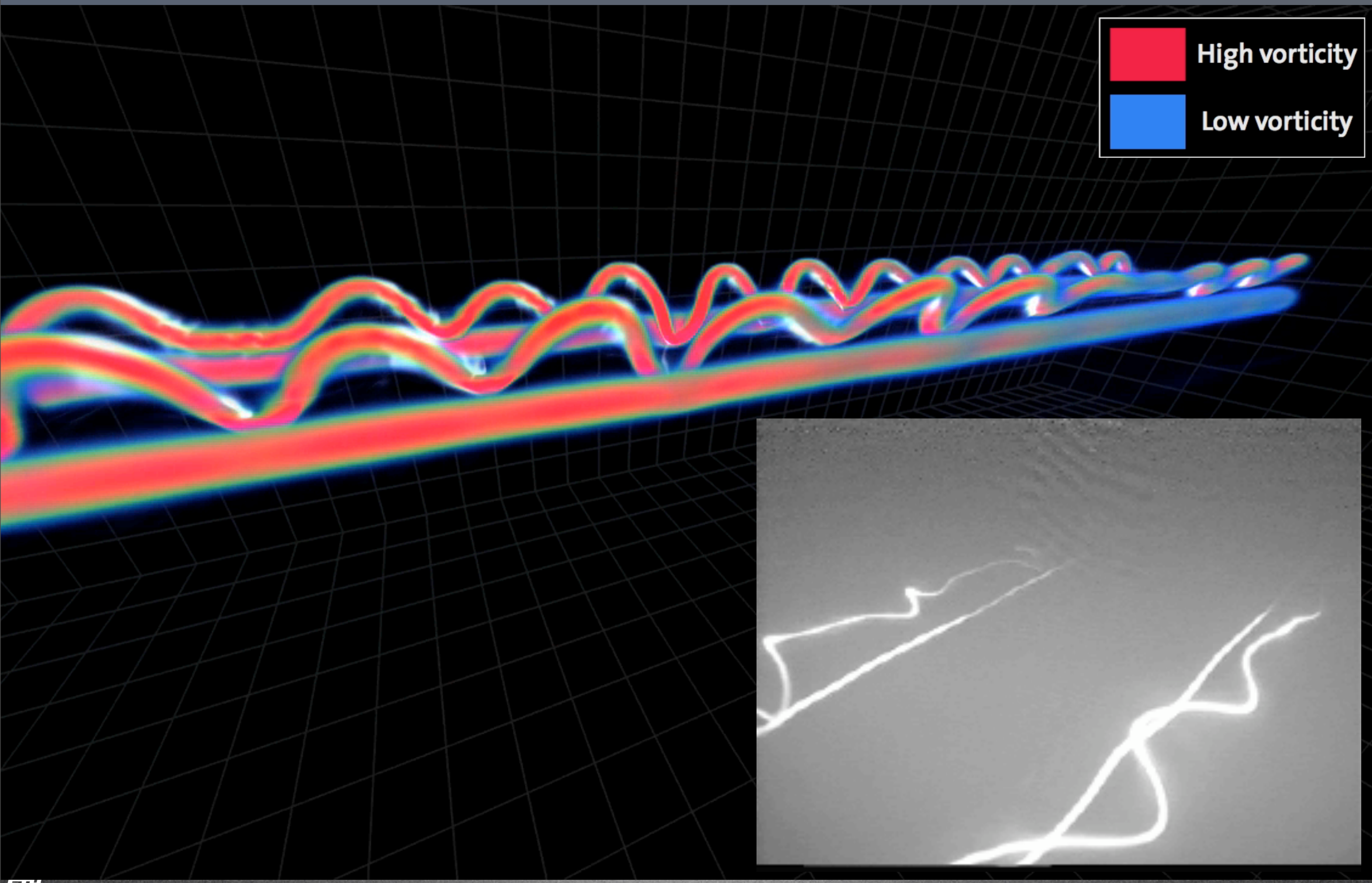


  - Mode $\lambda/b_1 = 0.86 - 0.943$
    - Experimental $\lambda/b_1 = 0.9 - 1.3$
  - Bursts when Ω-loop feet come together
- Visualization: volume rendering of $|\omega|$

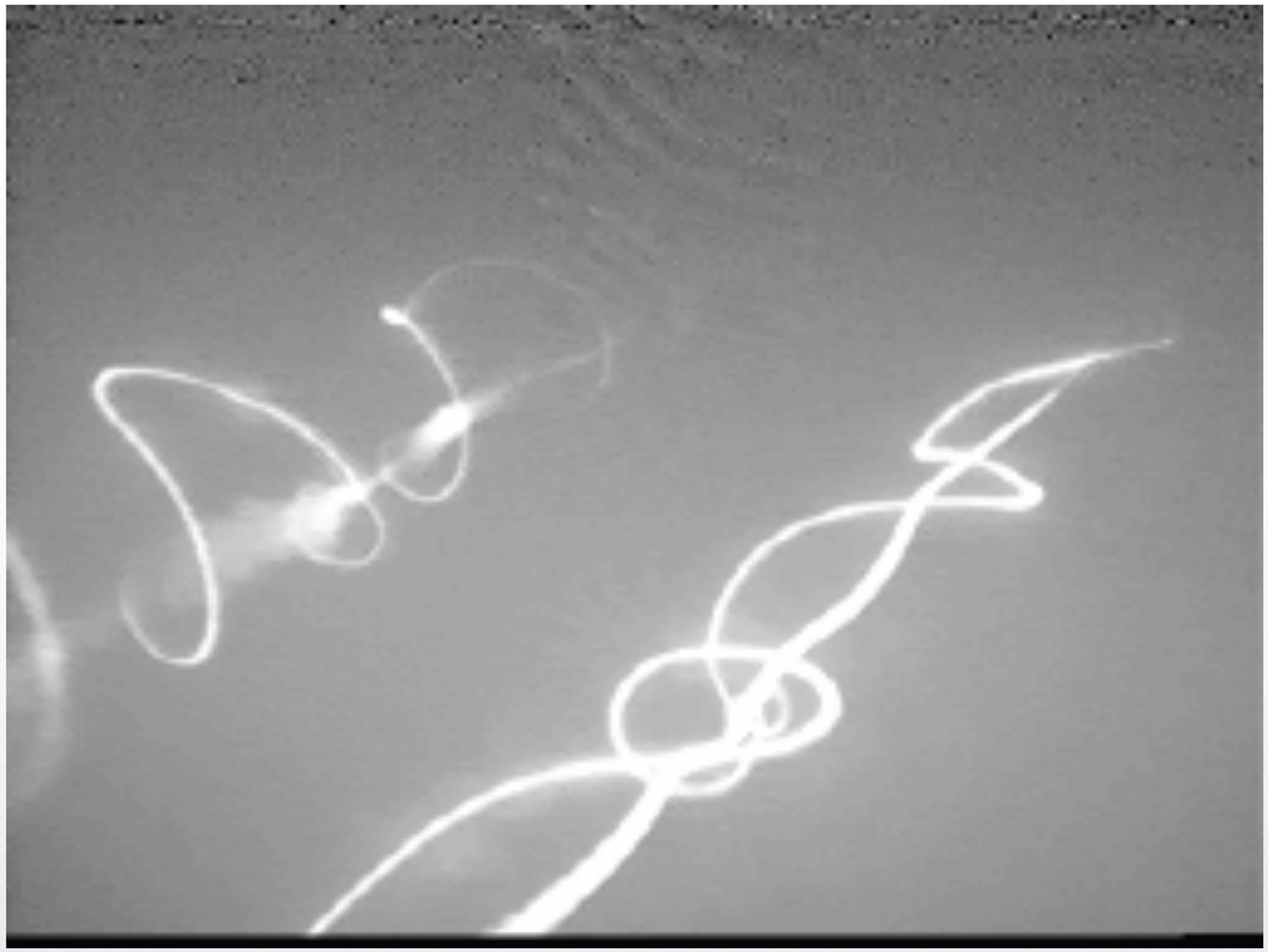
Rossinelli D. et al., SIGGRAPH08

Instability : **Linear phase** t=0.21

Instability : **Linear phase** t=0.21

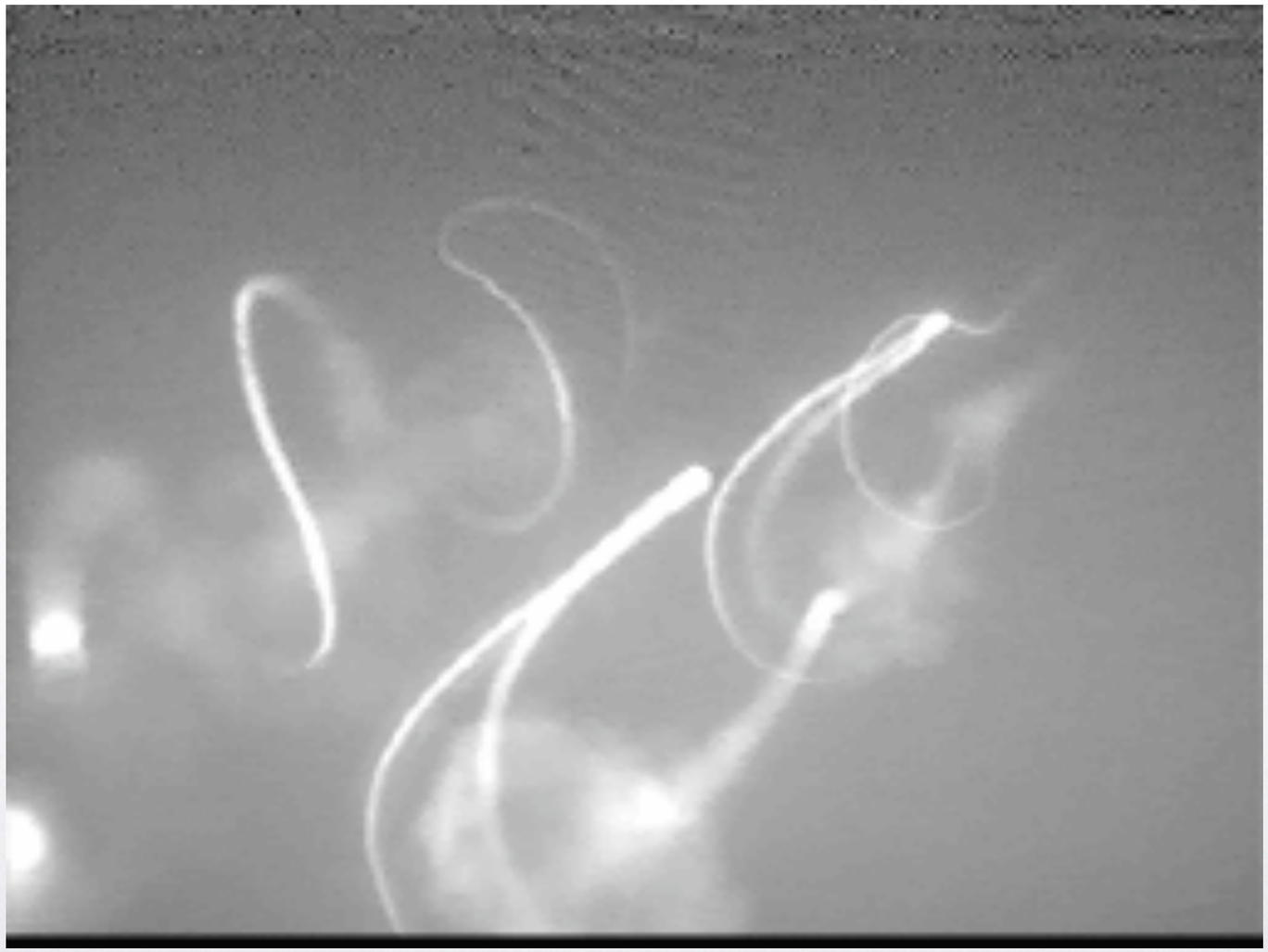


Instability : **Reconnections** t=0.25



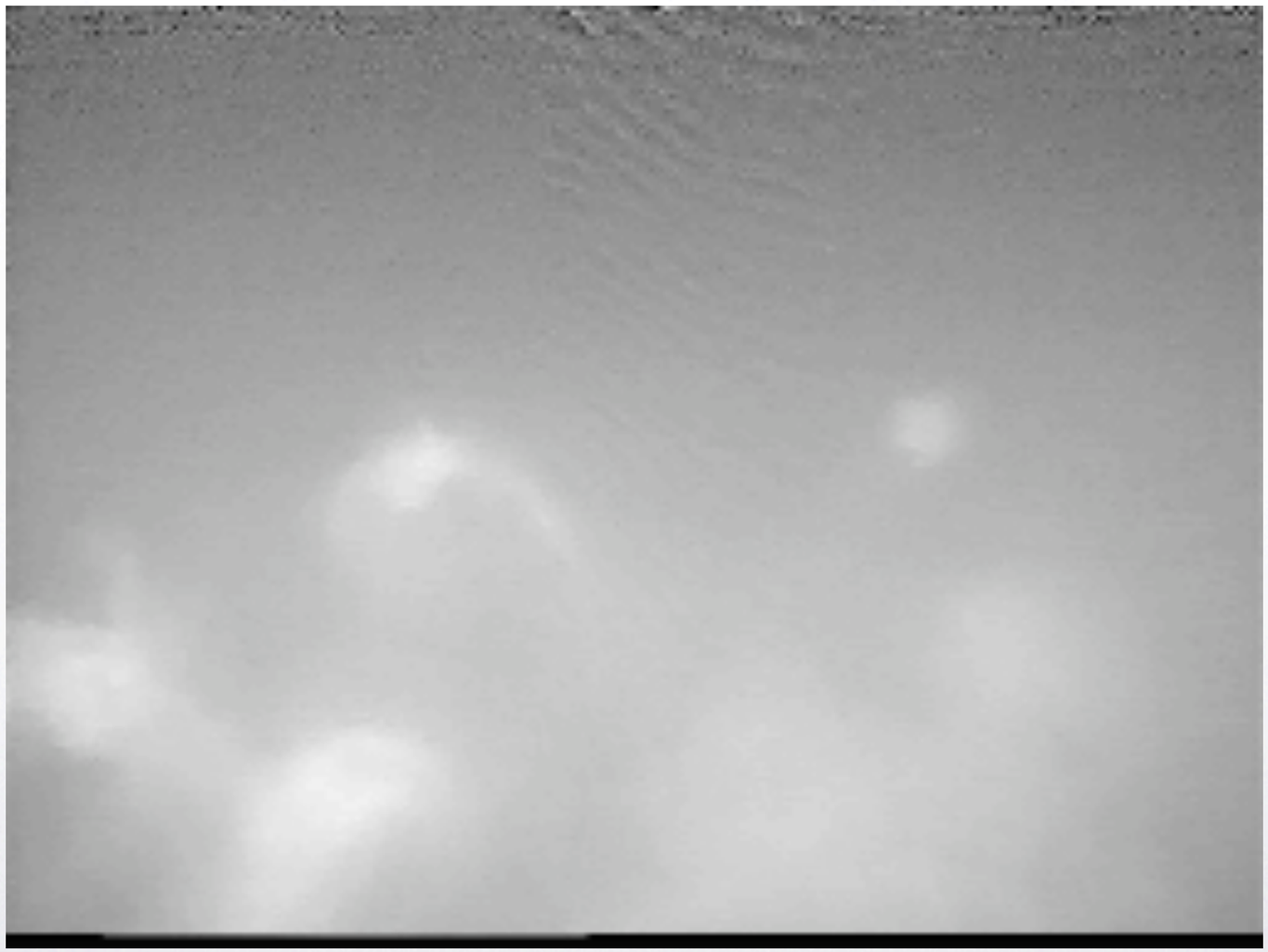
Instability : Reconnections t=0.25

Instability: **Propagation** t=0.27

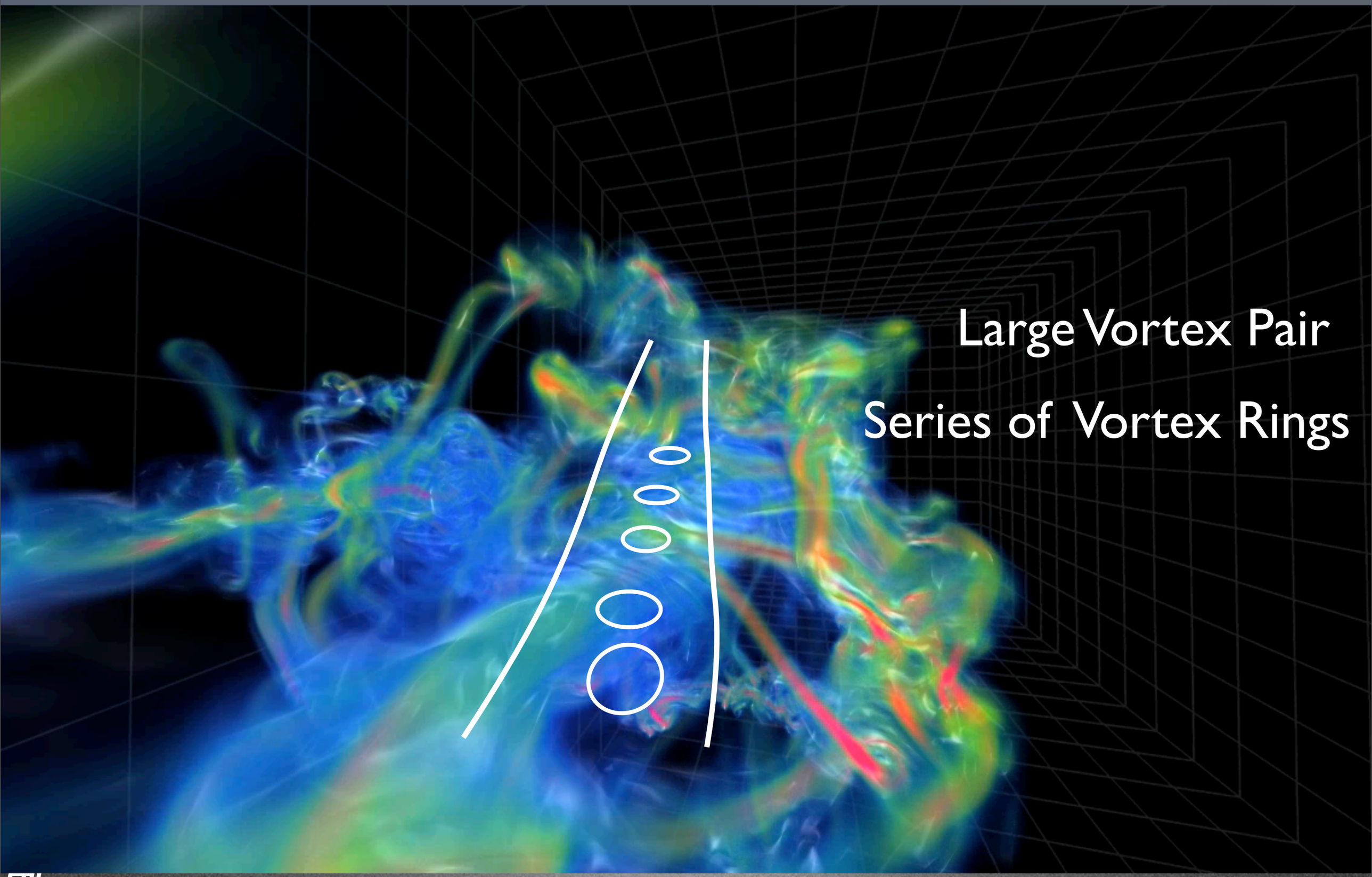


Instability: **Propagation** t=0.27

Instability: Decay t=0.35



Instability: **Decay** t=0.35



Optimization of Vortex Decay

Optimization of Vortex Decay

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

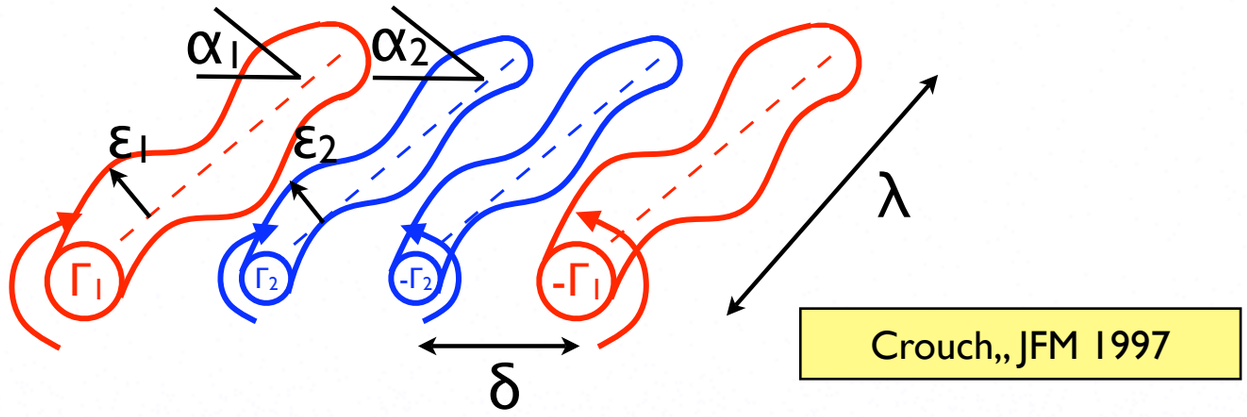
Optimization of Vortex Decay

- 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

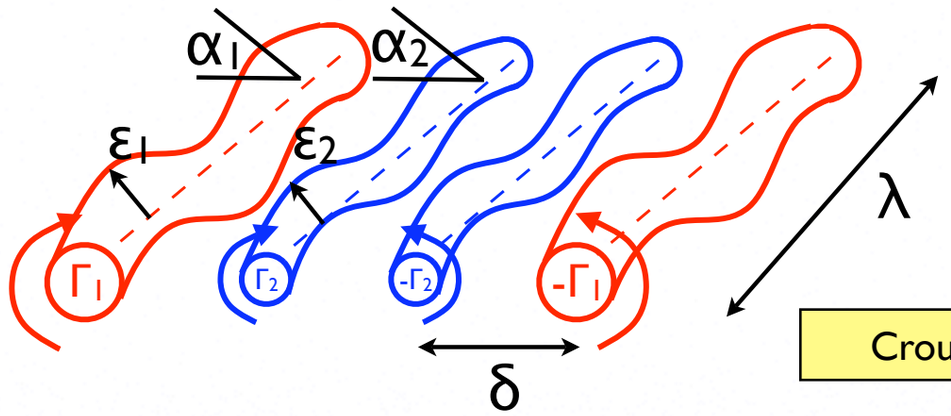
- Configuration
  - Co-rotating pairs
  - Re = 2500



Crouch,, JFM 1997

Optimization of Vortex Decay

- Configuration
  - Co-rotating pairs
  - Re = 2500



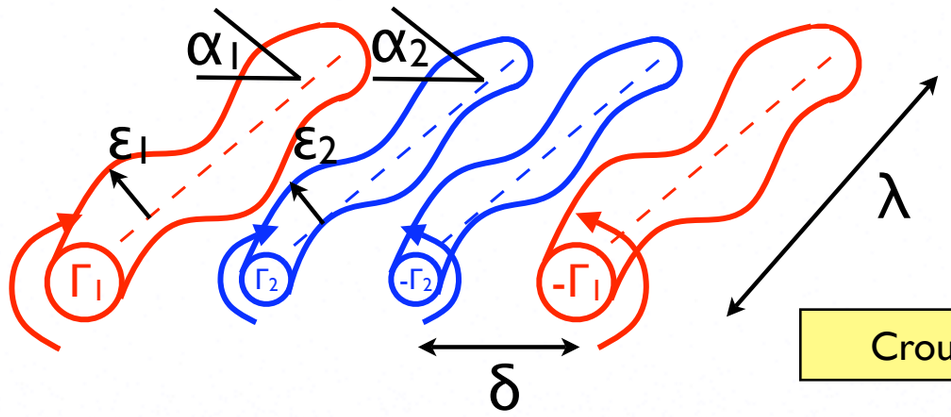
Crouch,, JFM 1997

- Problem: minimize **objective function**

$$f_{\text{decay}} = \frac{\int_0^T E(t)\,dt}{E(0)T} \qquad (\mathsf{T} = 4)$$

Optimization of Vortex Decay

- Configuration
  - Co-rotating pairs
  - Re = 2500

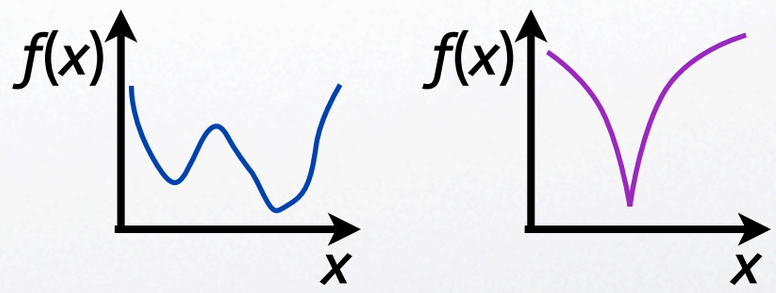


Crouch,, JFM 1997

- Problem: minimize **objective function**

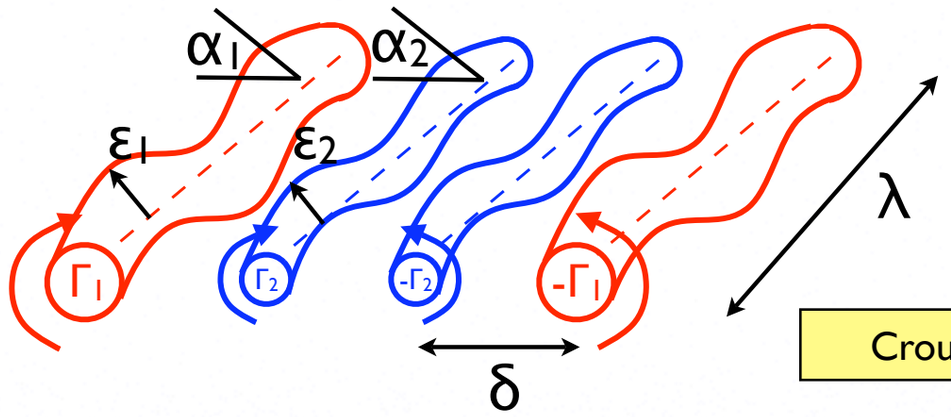
$$f_{\text{decay}} = \frac{\int_0^T E(t)\,dt}{E(0)T} \qquad (\mathsf{T} = 4)$$

- Challenges
  - f can be multi-modal, non-convex, etc.
  - ∇f not readily available
  - evaluation of f is **expensive**



Optimization of Vortex Decay

- Configuration
  - Co-rotating pairs
  - Re = 2500

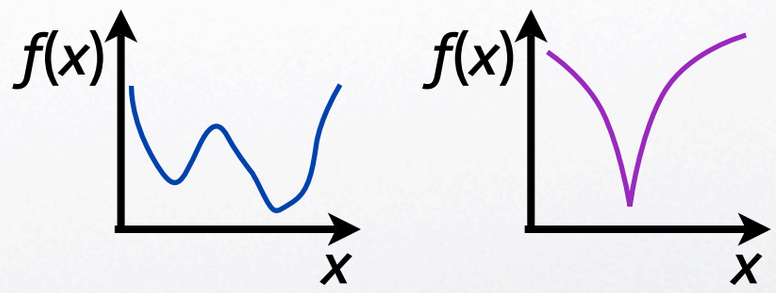


Crouch,, JFM 1997

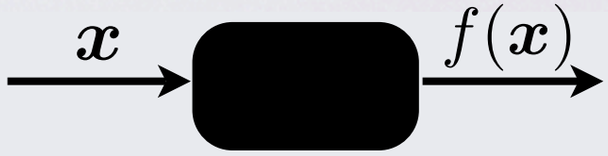
- Problem: minimize **objective function**

$$f_{\text{decay}} = \frac{\int_0^T E(t)\,dt}{E(0)T} \qquad (\mathsf{T} = 4)$$

- Challenges
  - f can be multi-modal, non-convex, etc.
  - ∇f not readily available
  - evaluation of f is **expensive**



- Approach
  - Evolutionary Optimization
  - Prior knowledge encoded in parametrization

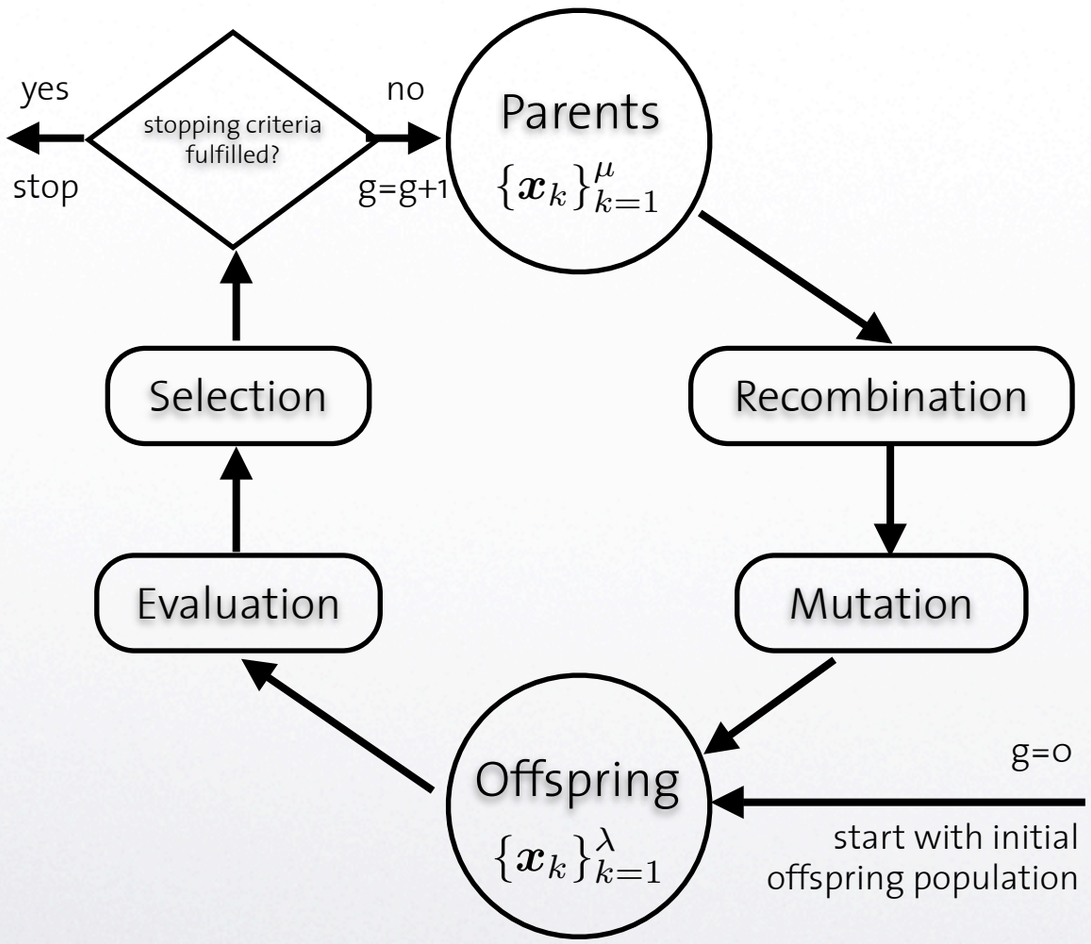


Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003

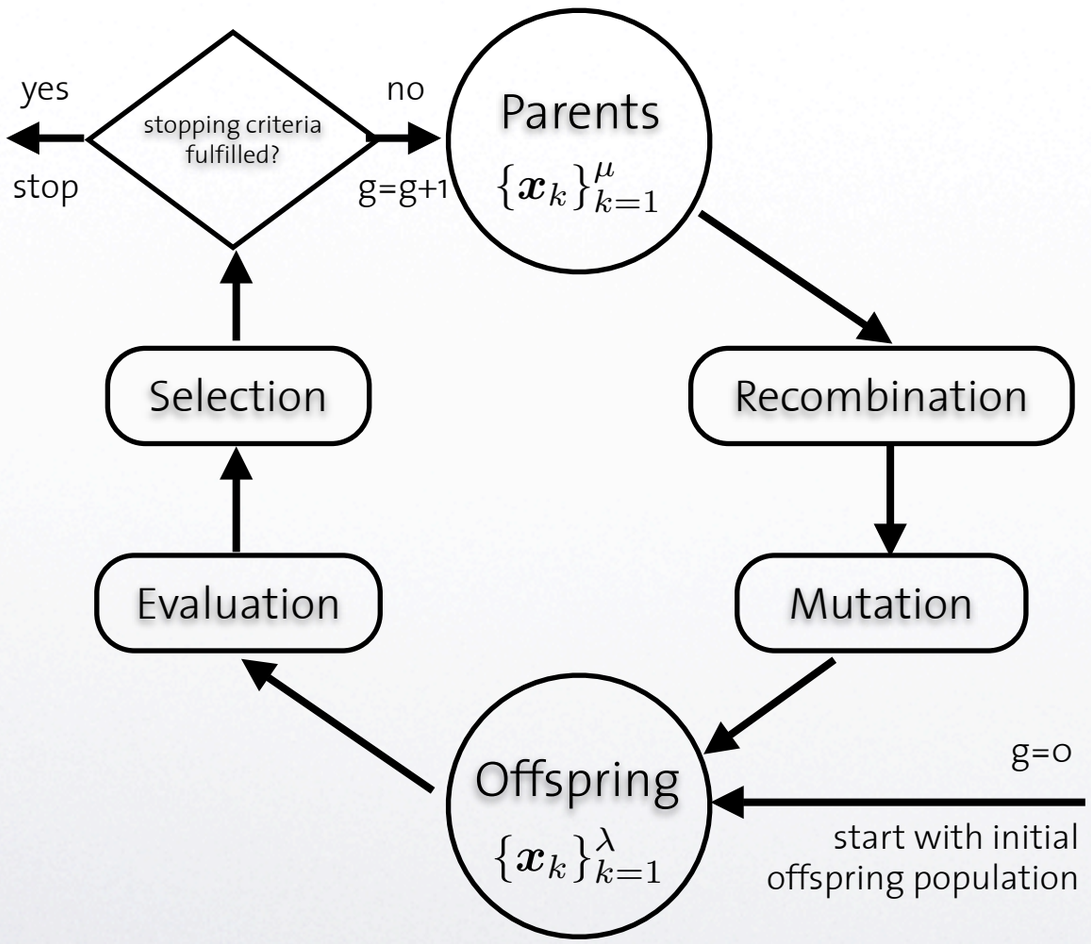


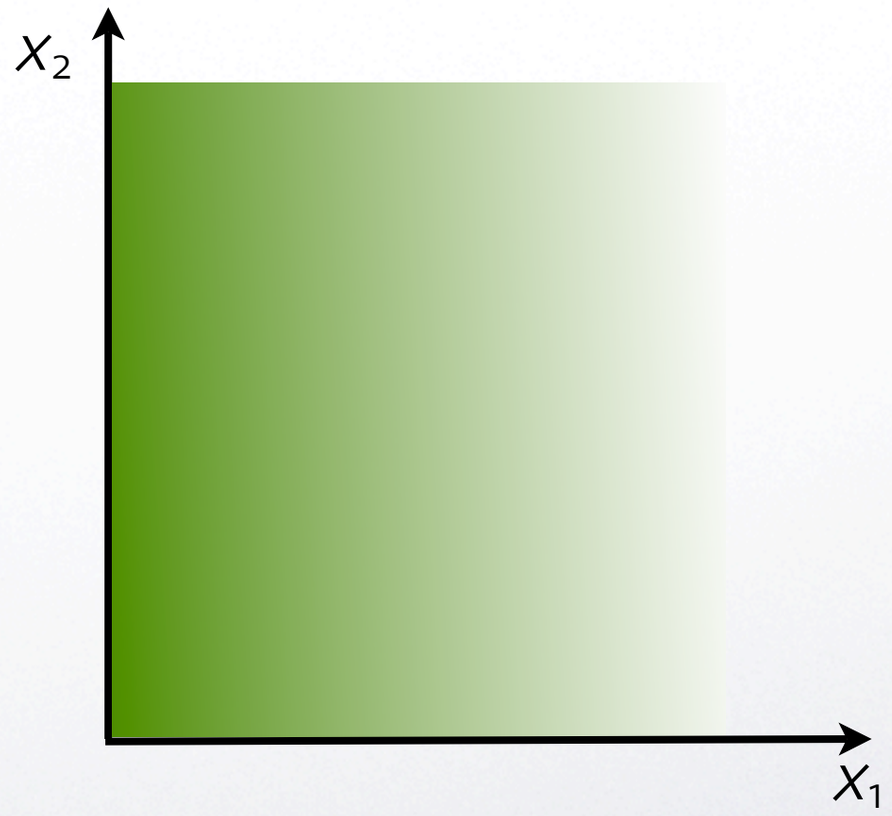
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



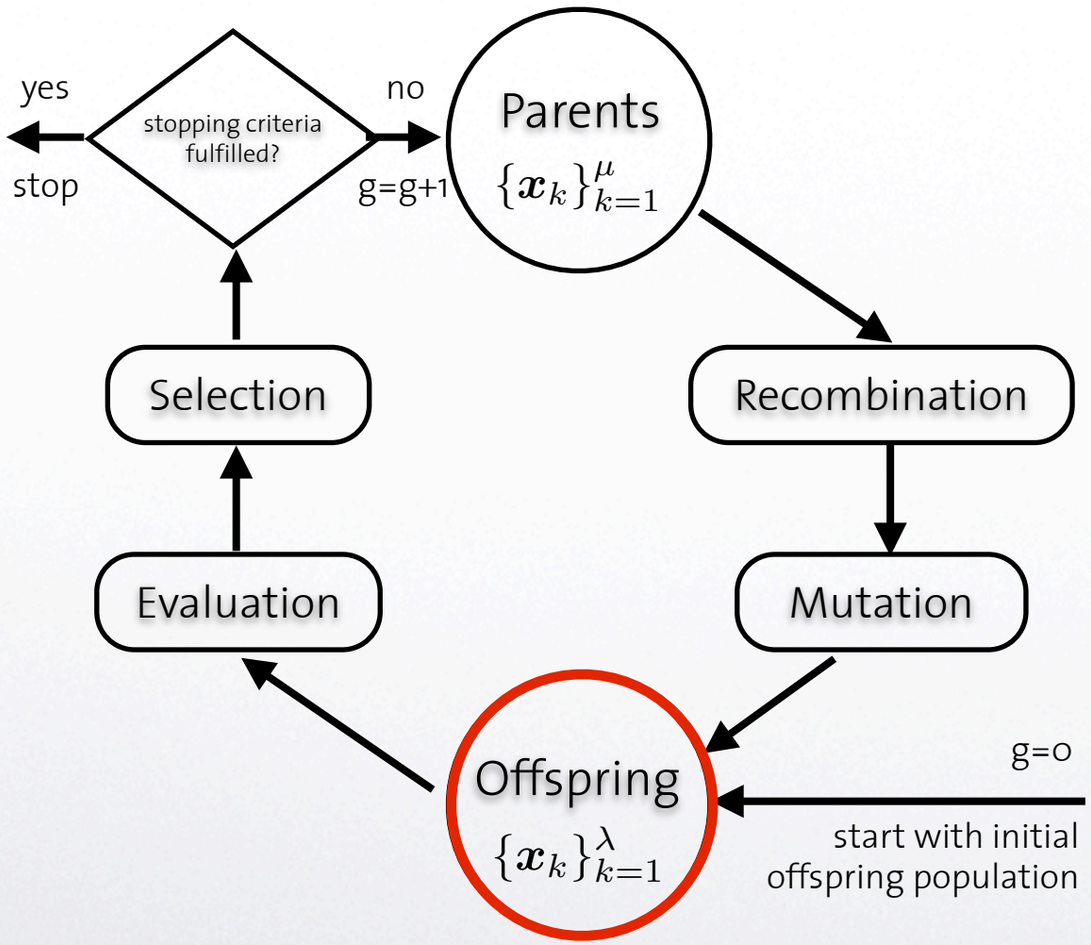


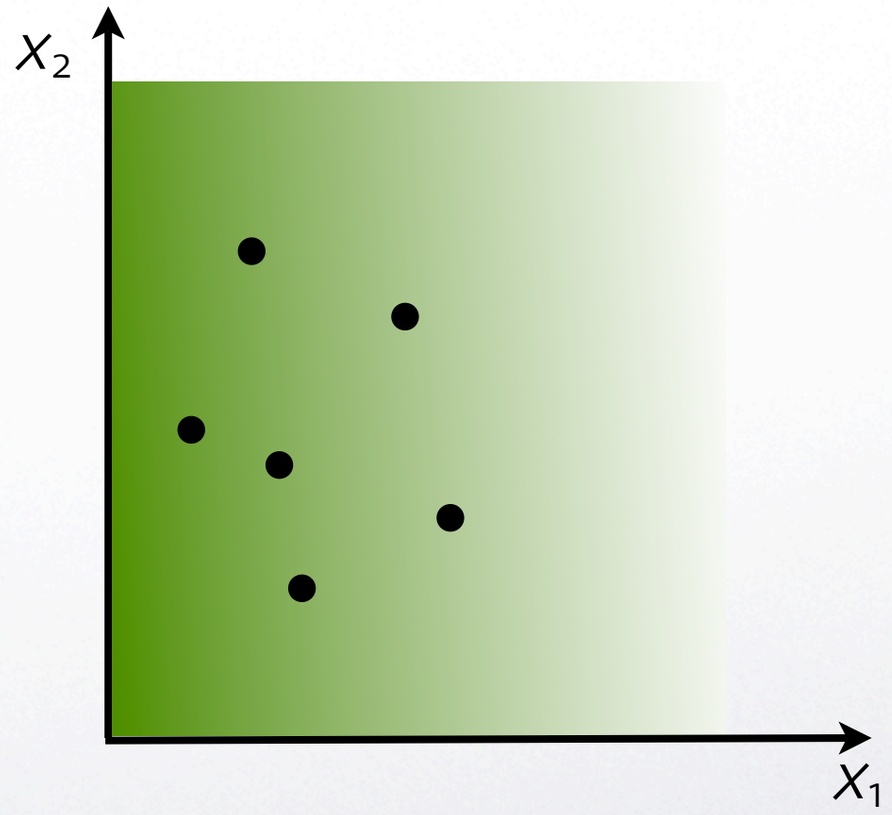
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



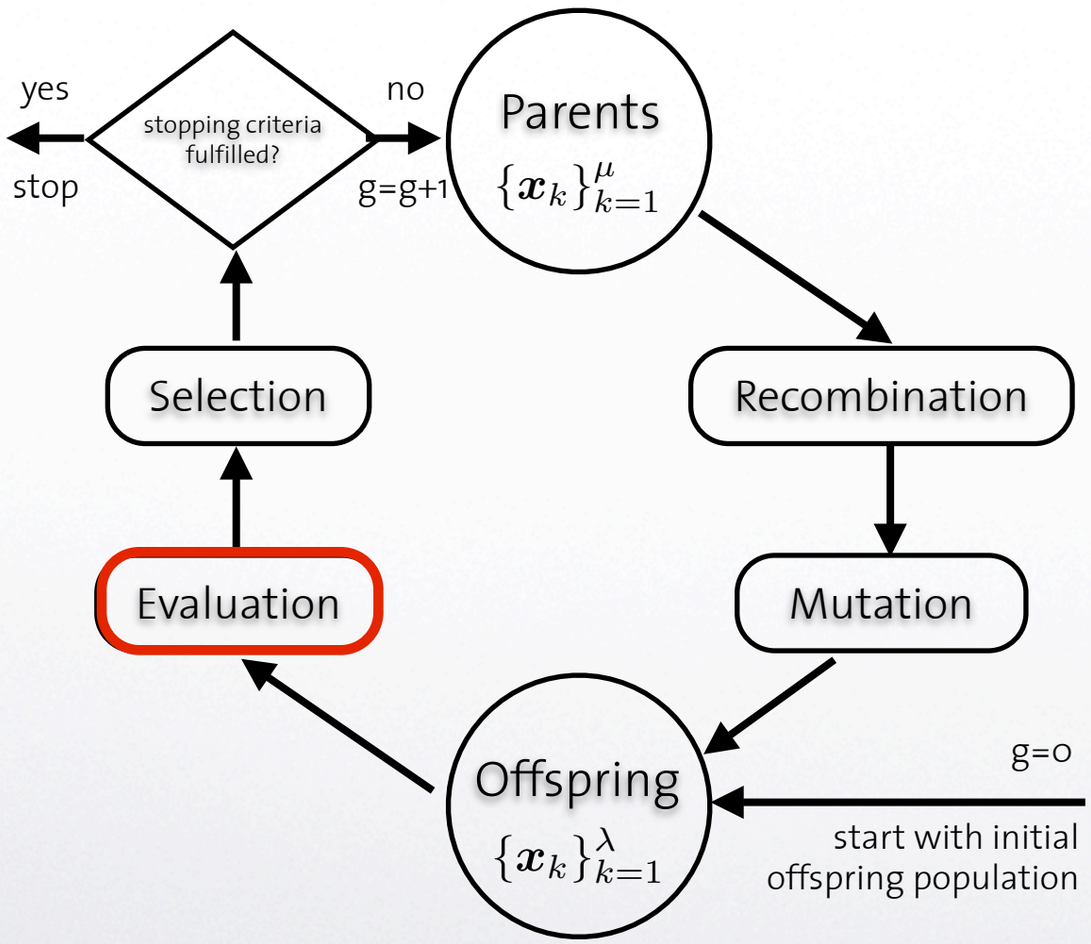


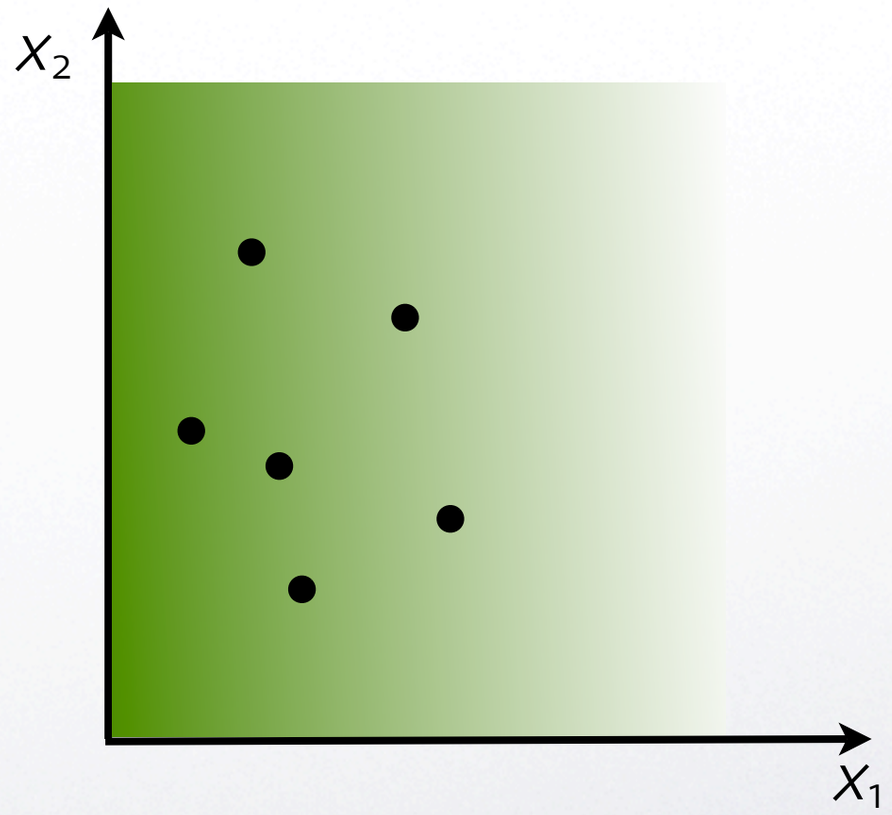
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



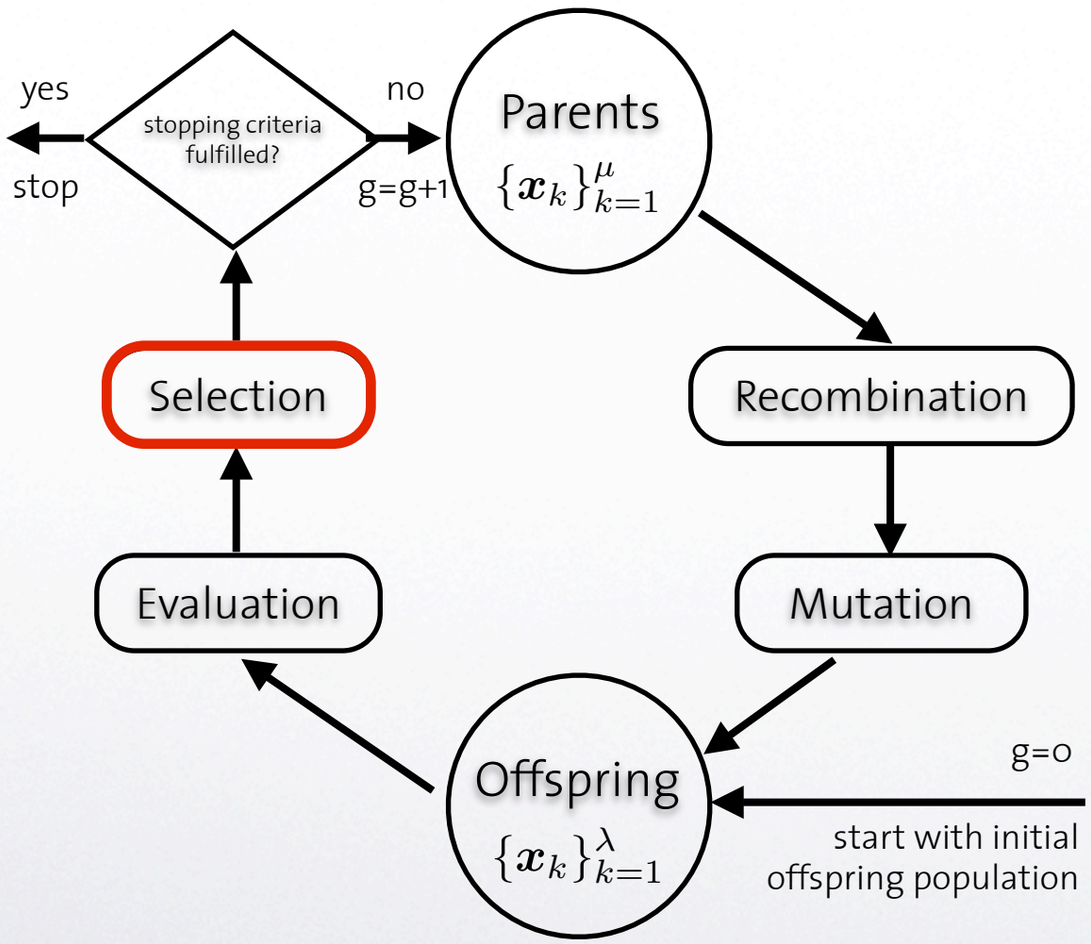


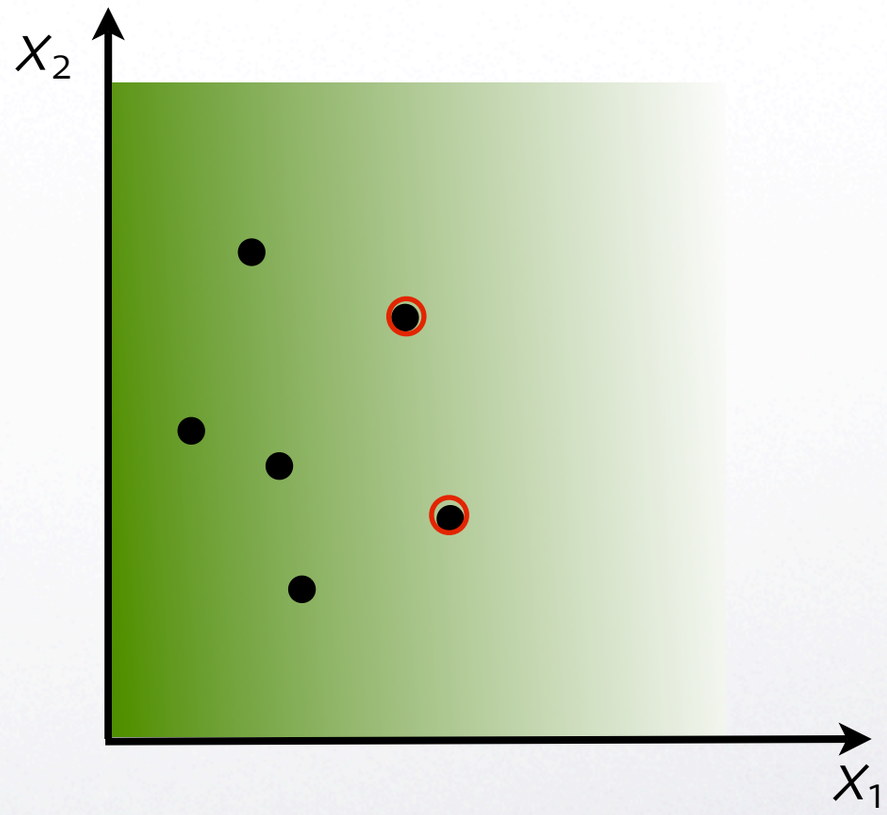
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



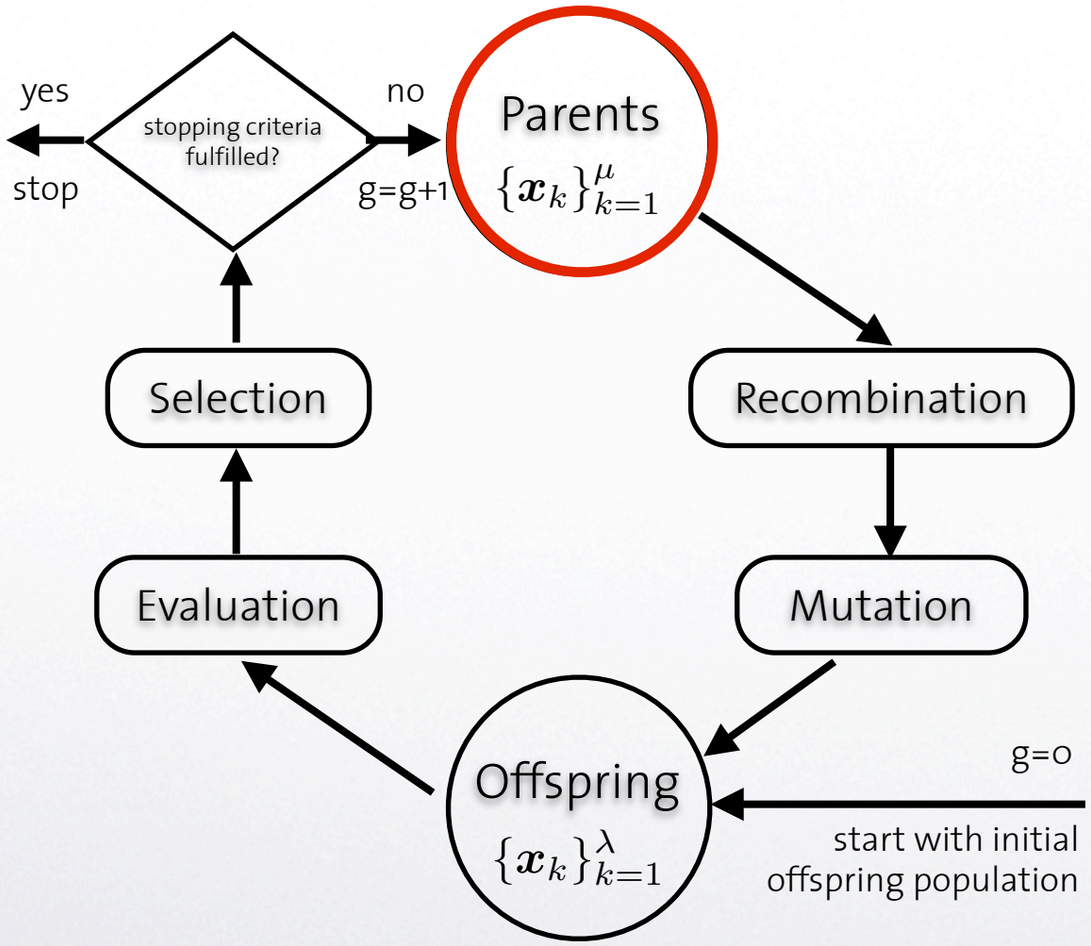


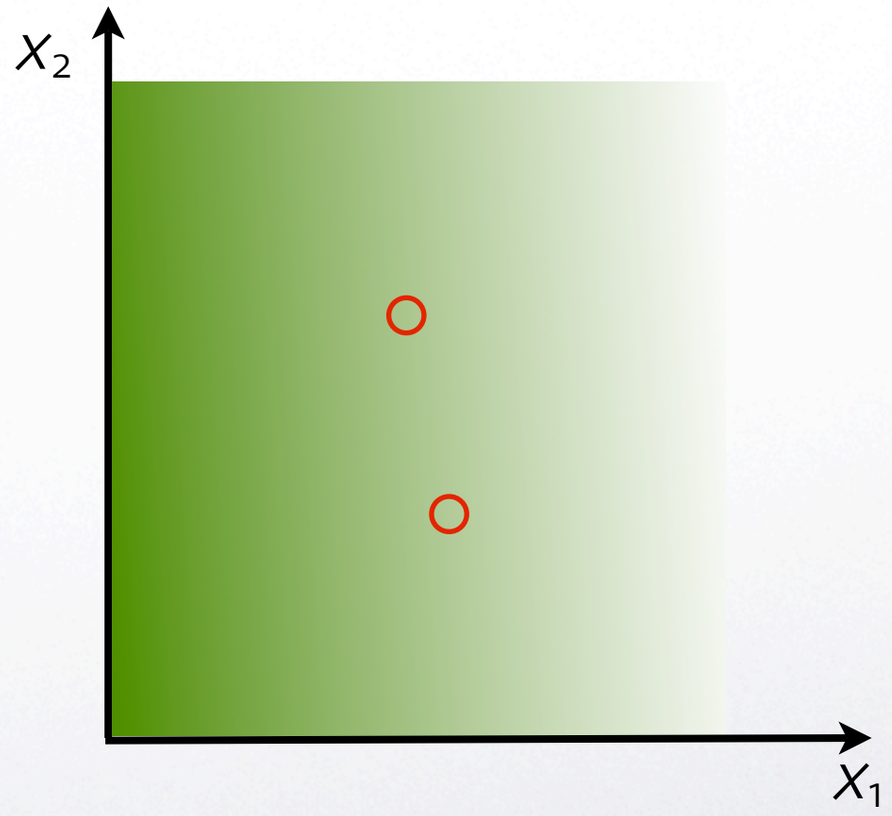
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



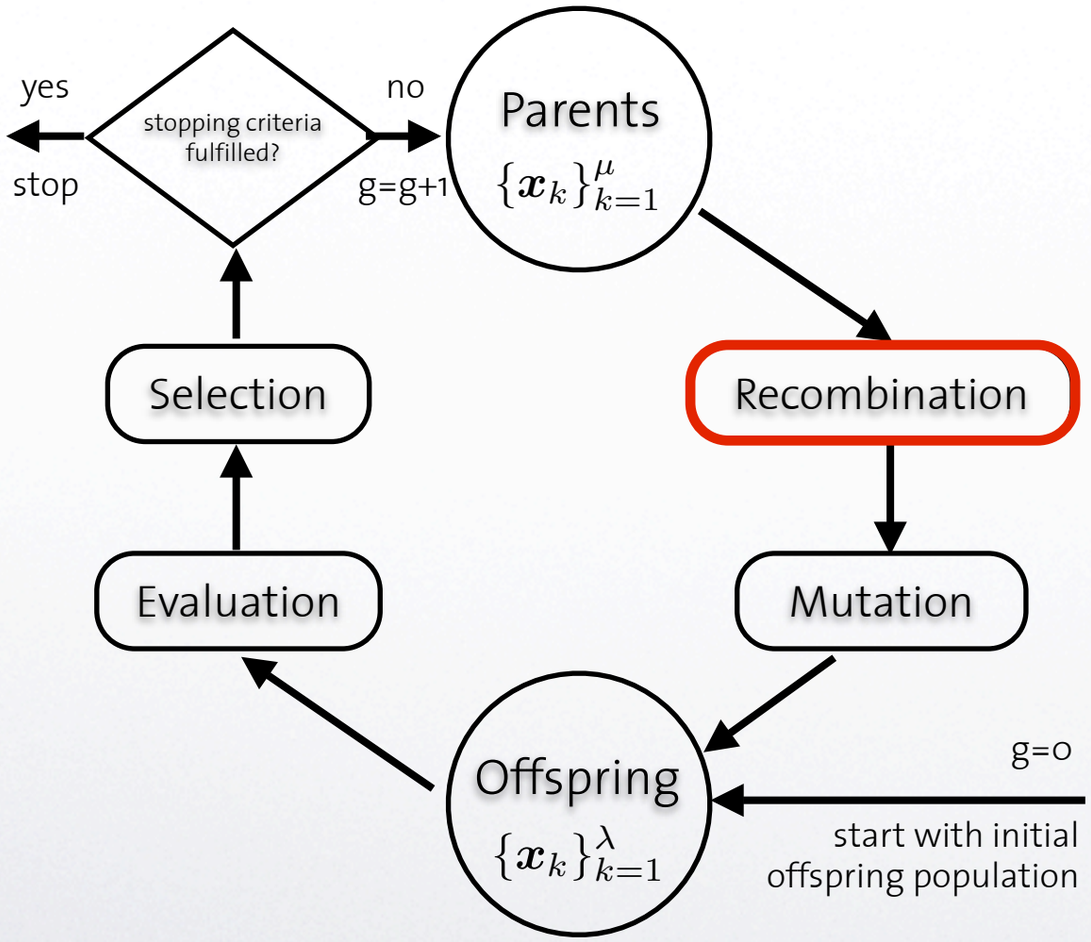


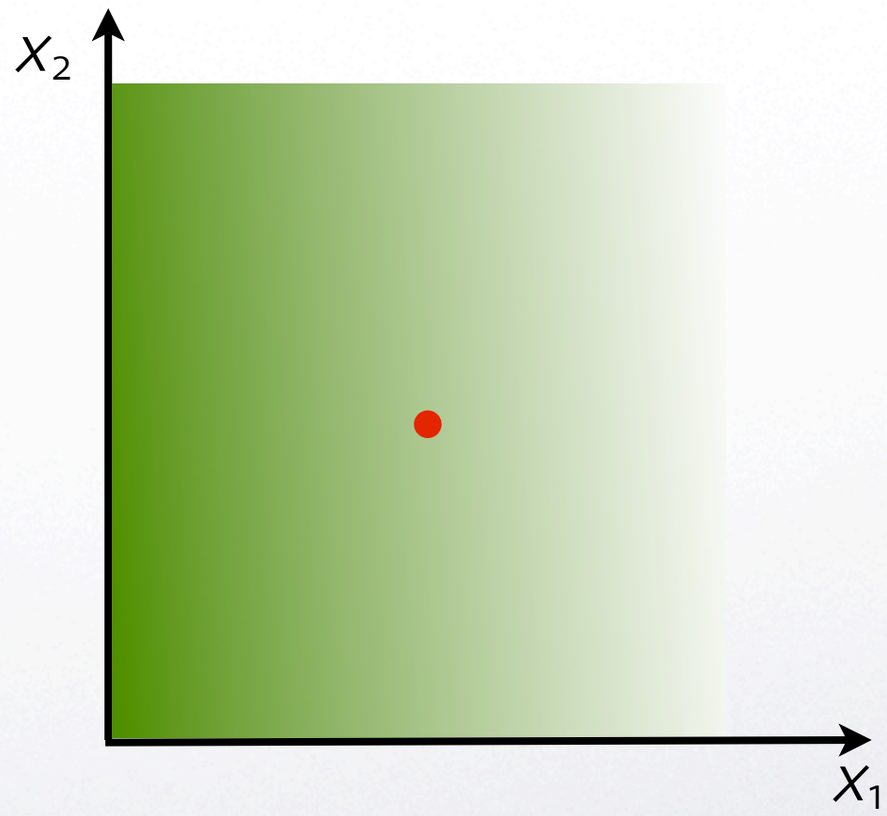
Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



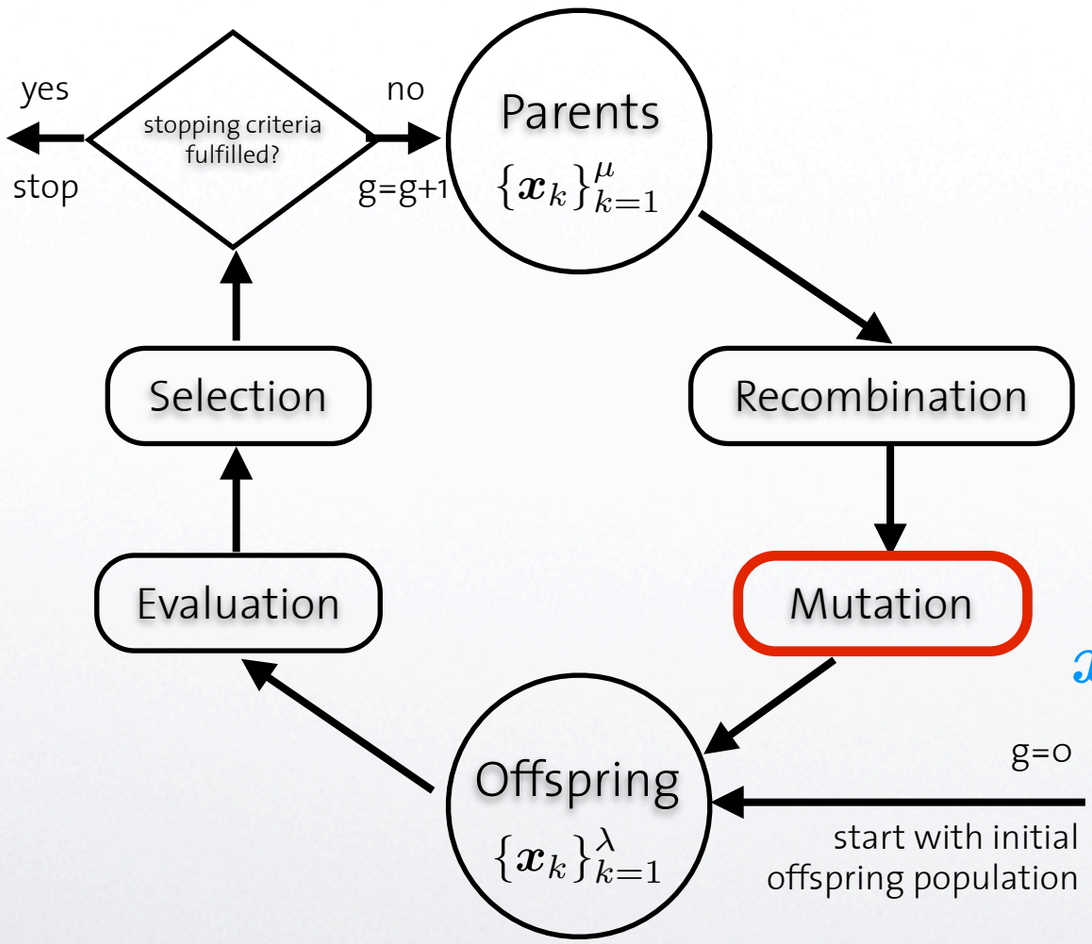


Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

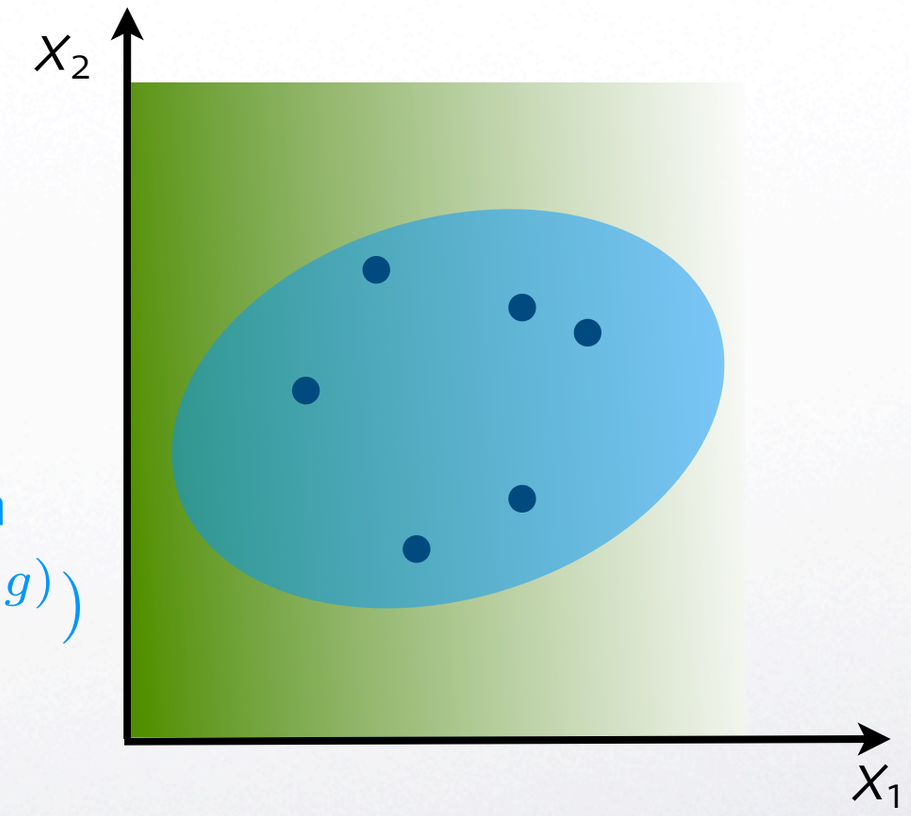
- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



Probability distribution

$$\boldsymbol{x}_k \sim P(\boldsymbol{\theta}^{(g)})$$

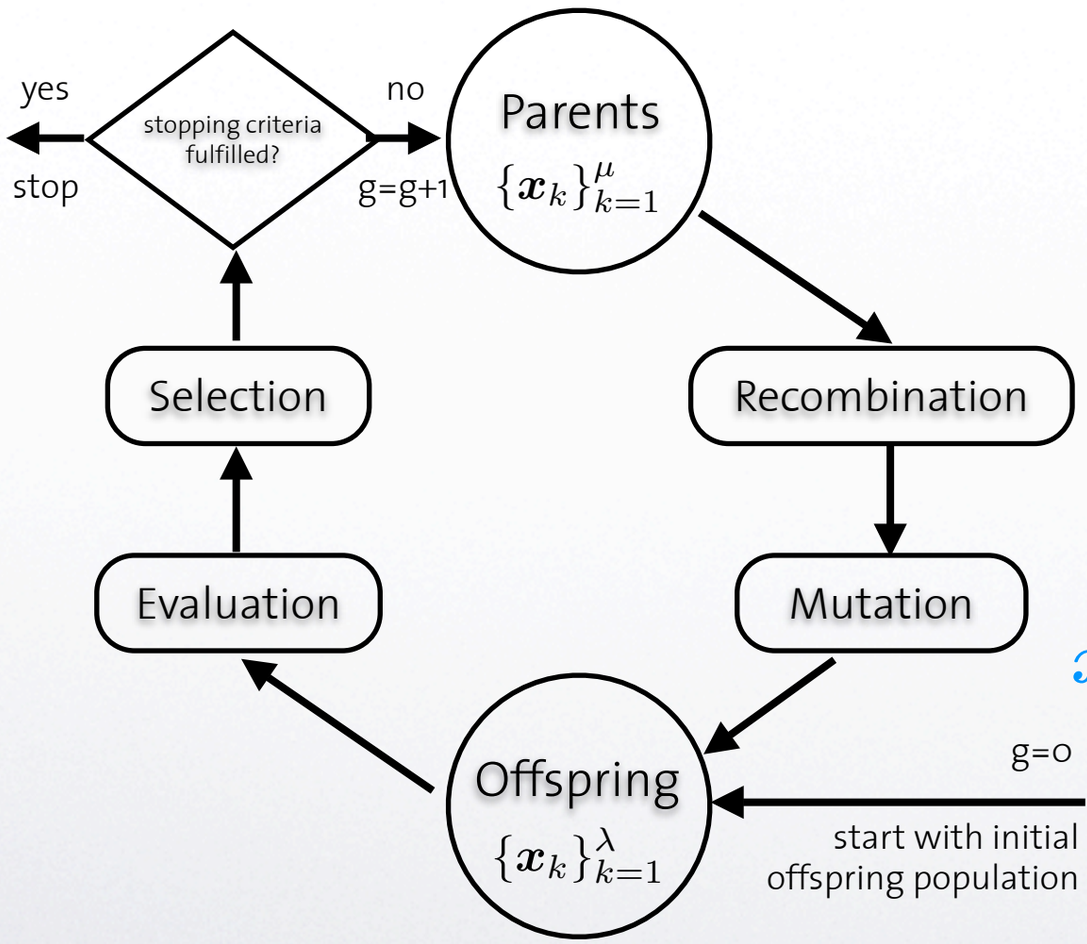


Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

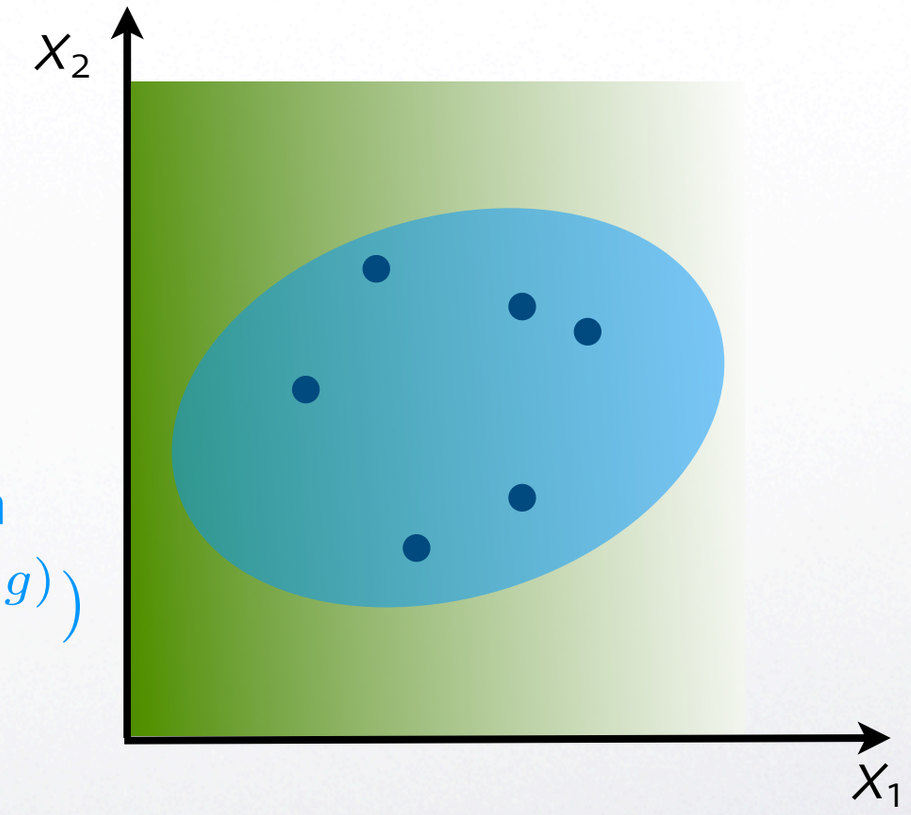
- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



Probability distribution

$$\boldsymbol{x}_k \sim P(\boldsymbol{\theta}^{(g)})$$



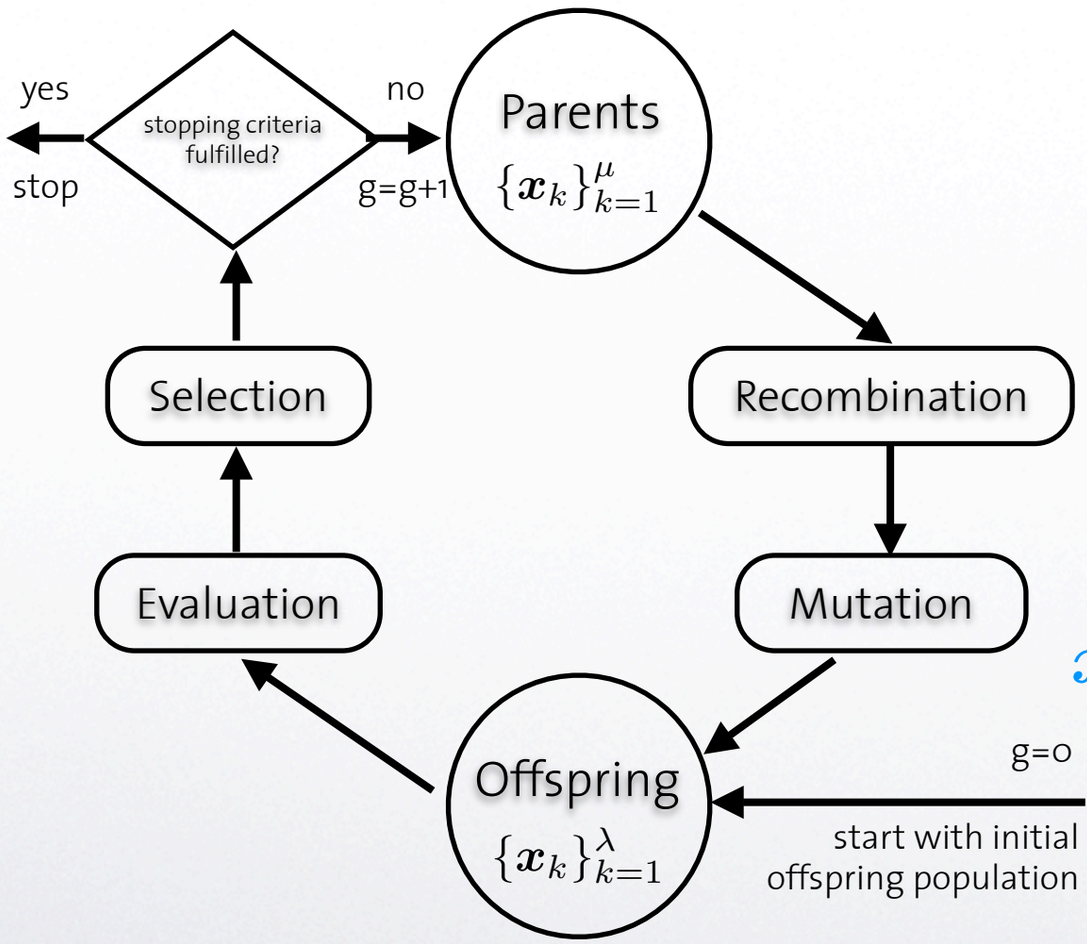
- popular because of their flexibility and **robustness**

Optimization of Vortex Decay

Evolutionary Algorithms (EAs)

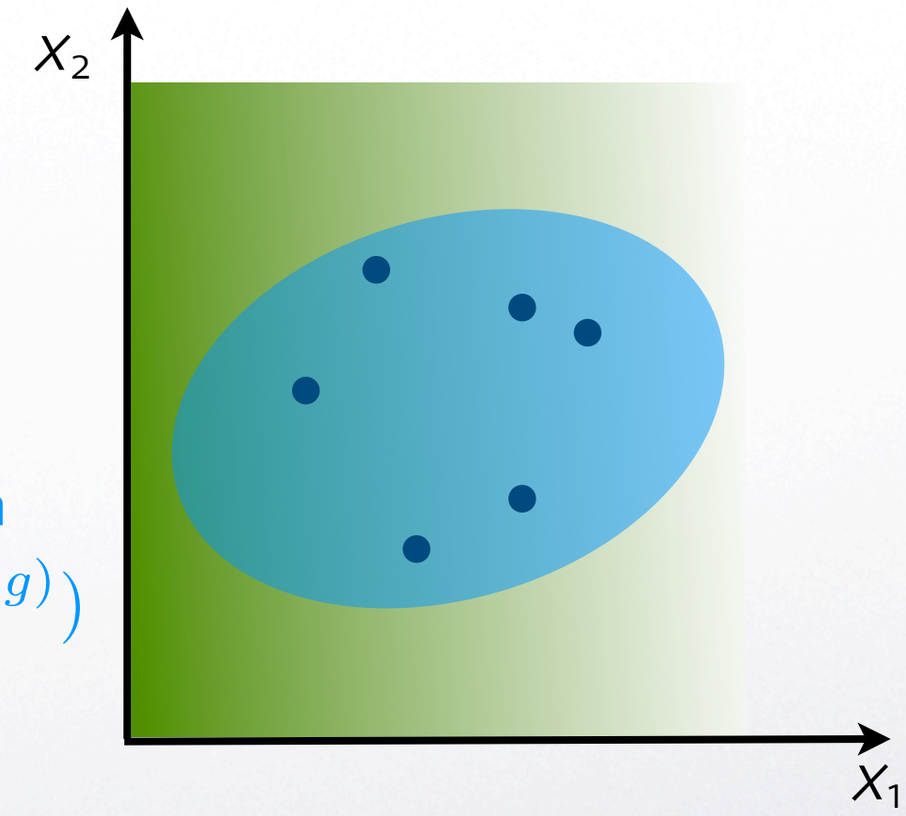
- **iterative methods** operating with **populations** of candidate solutions
- Here : Covariance Matrix Adaptation - ES

Hansen et al., Evol. Comput. 2003



Probability distribution

$$\boldsymbol{x}_k \sim P(\boldsymbol{\theta}^{(g)})$$



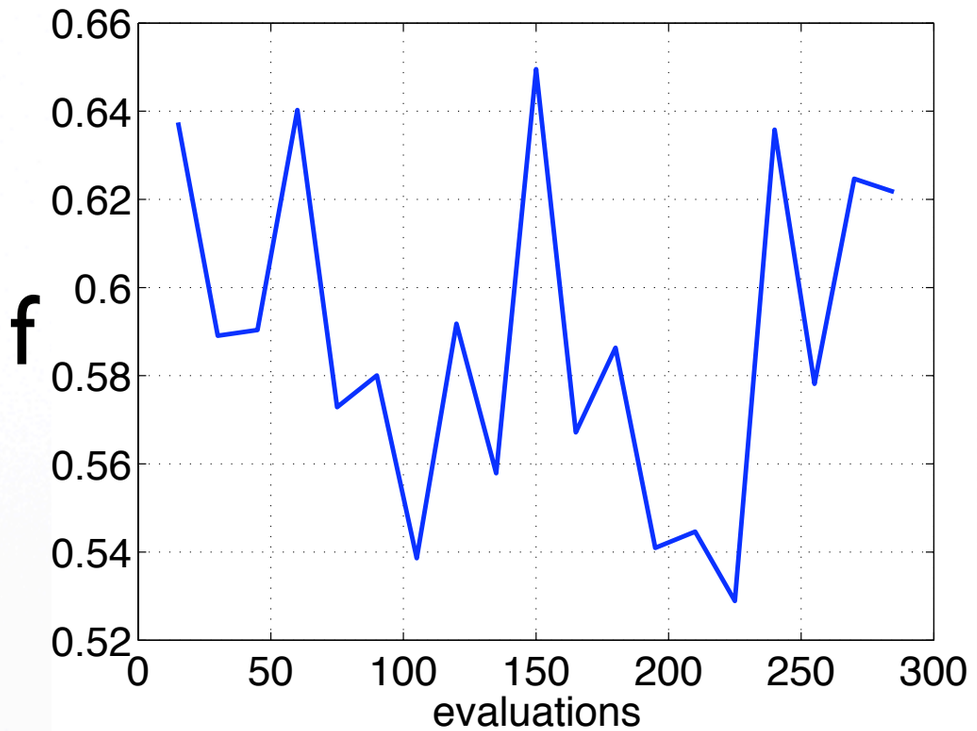
- popular because of their flexibility and **robustness**
- main **disadvantage**: Need **large number of objective function evaluations**

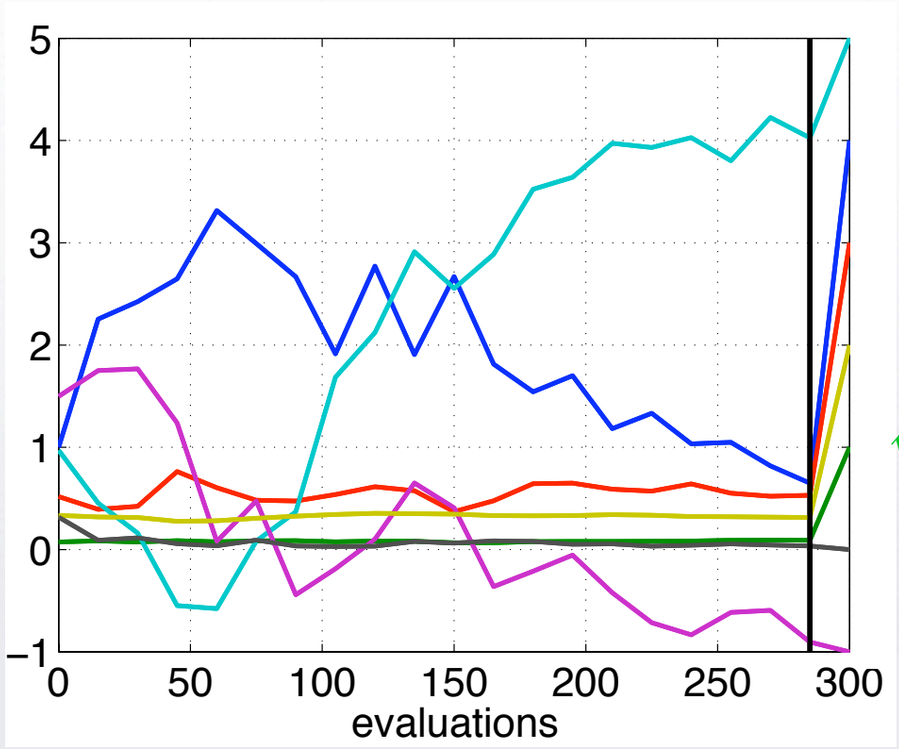
Optimization of Vortex Decay

$$f$$

Optimization of Vortex Decay

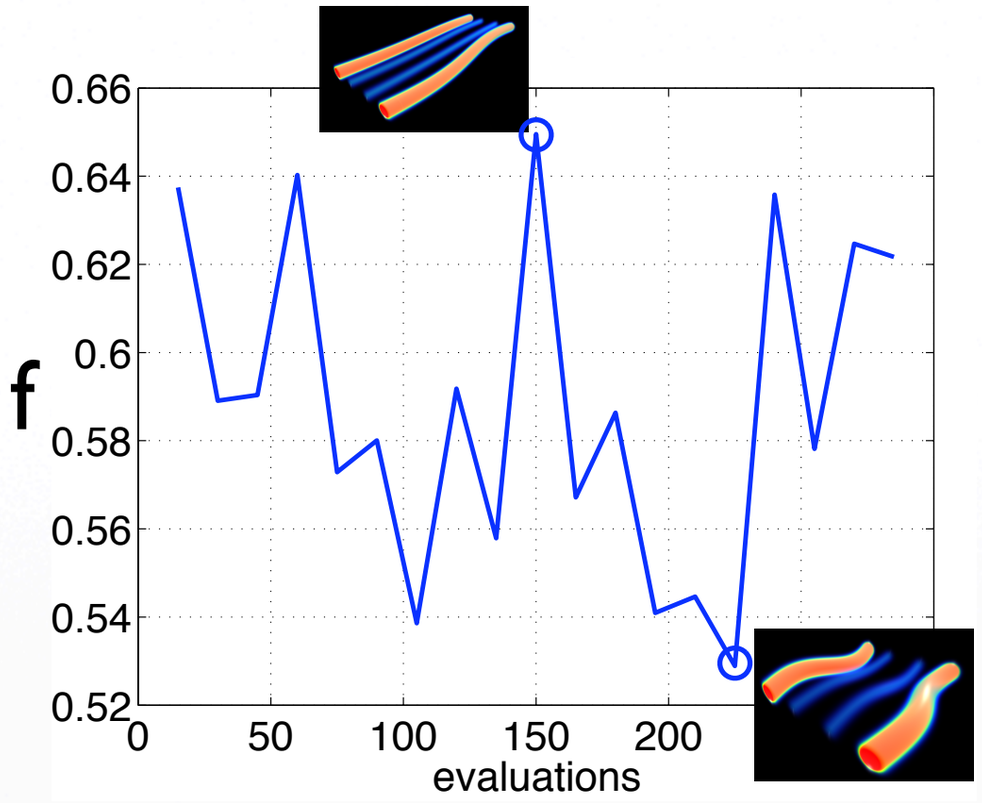
- PRESENT
  - After sweep toward large λ, moves toward smaller values
  - Convergence

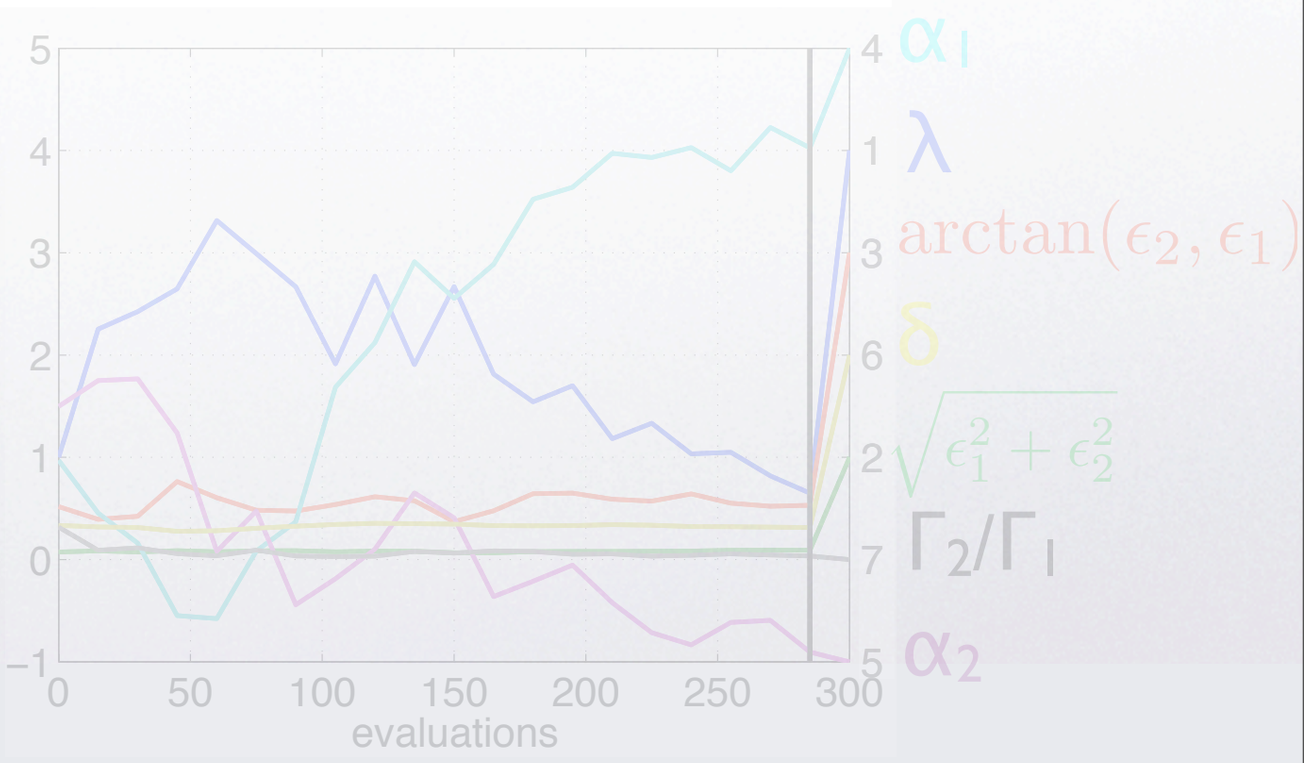




Optimization of Vortex Decay

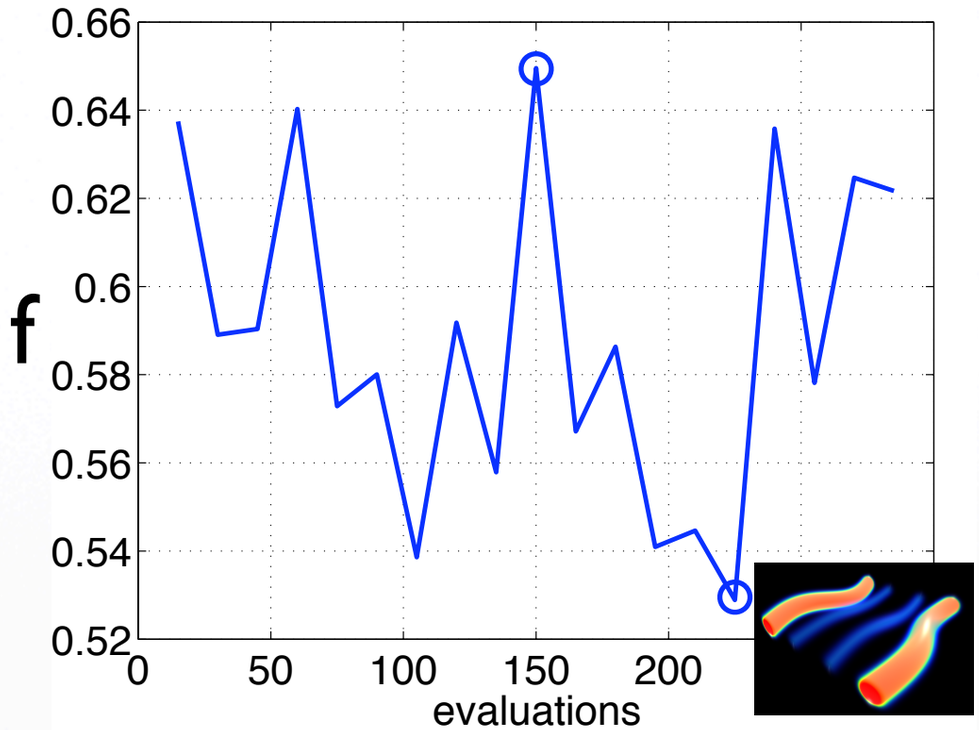
- PRESENT
  - After sweep toward large λ, moves toward smaller values
  - Convergence

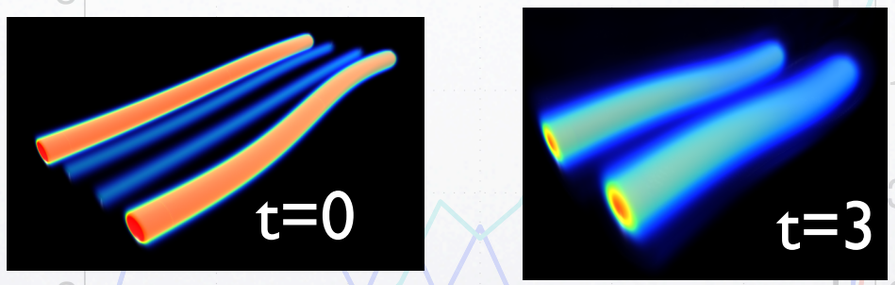




Optimization of Vortex Decay

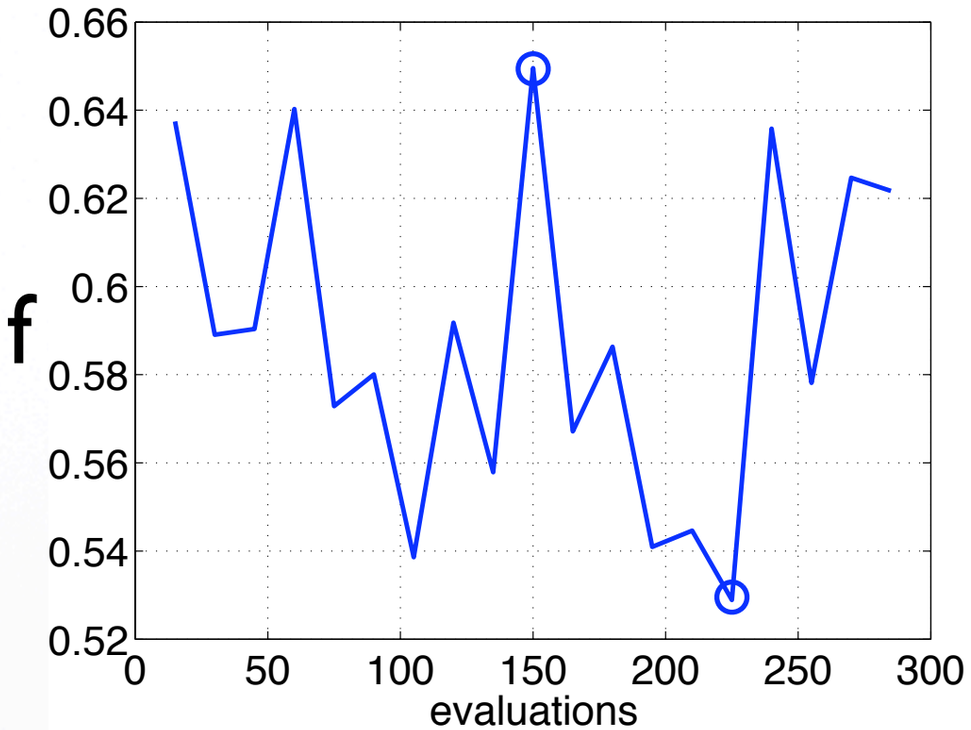
- PRESENT
  - After sweep toward large λ, moves toward smaller values
  - Convergence

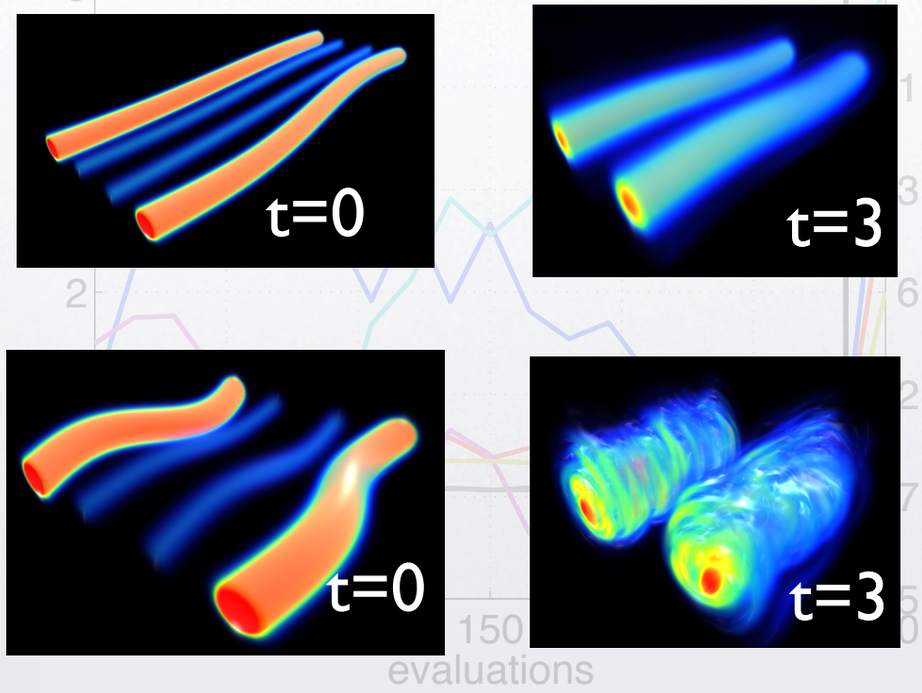




Optimization of Vortex Decay

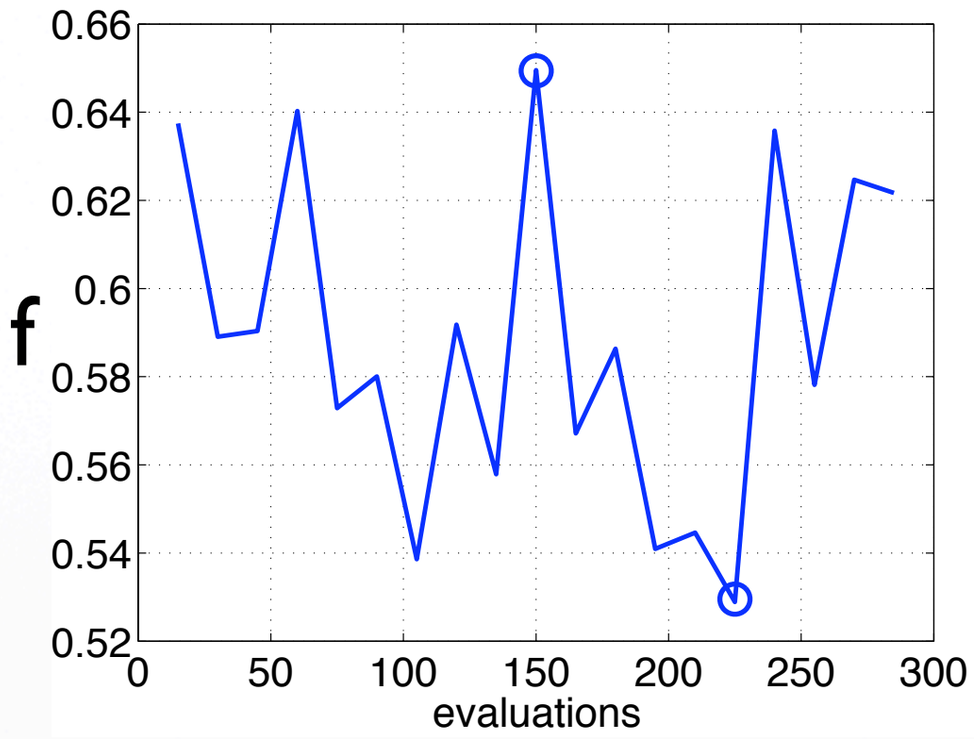
- PRESENT
  - After sweep toward large λ, moves toward smaller values
  - Convergence

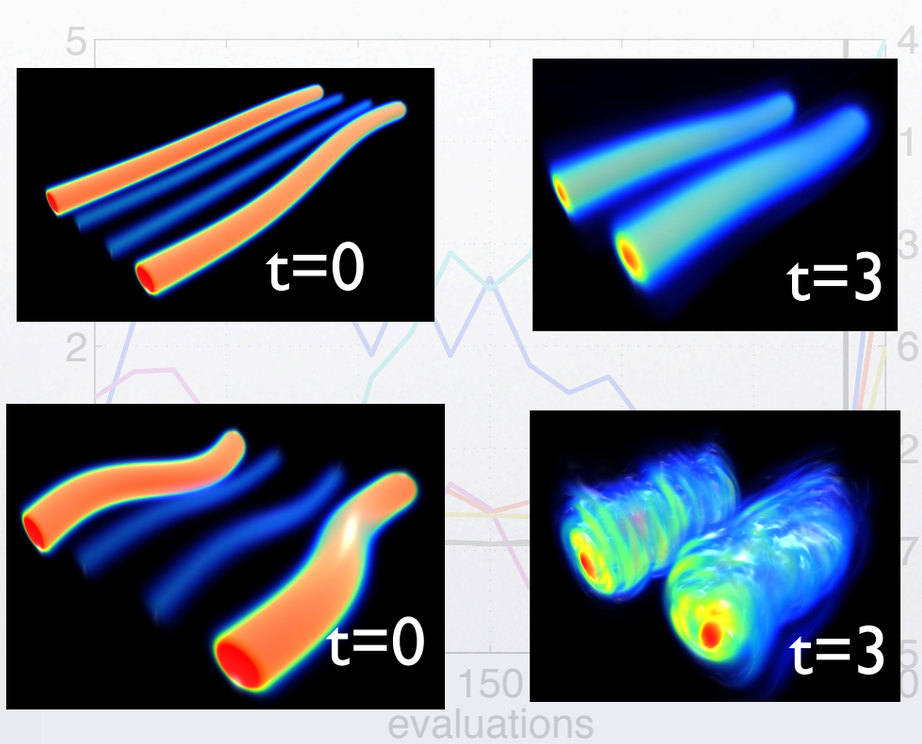




Optimization of Vortex Decay

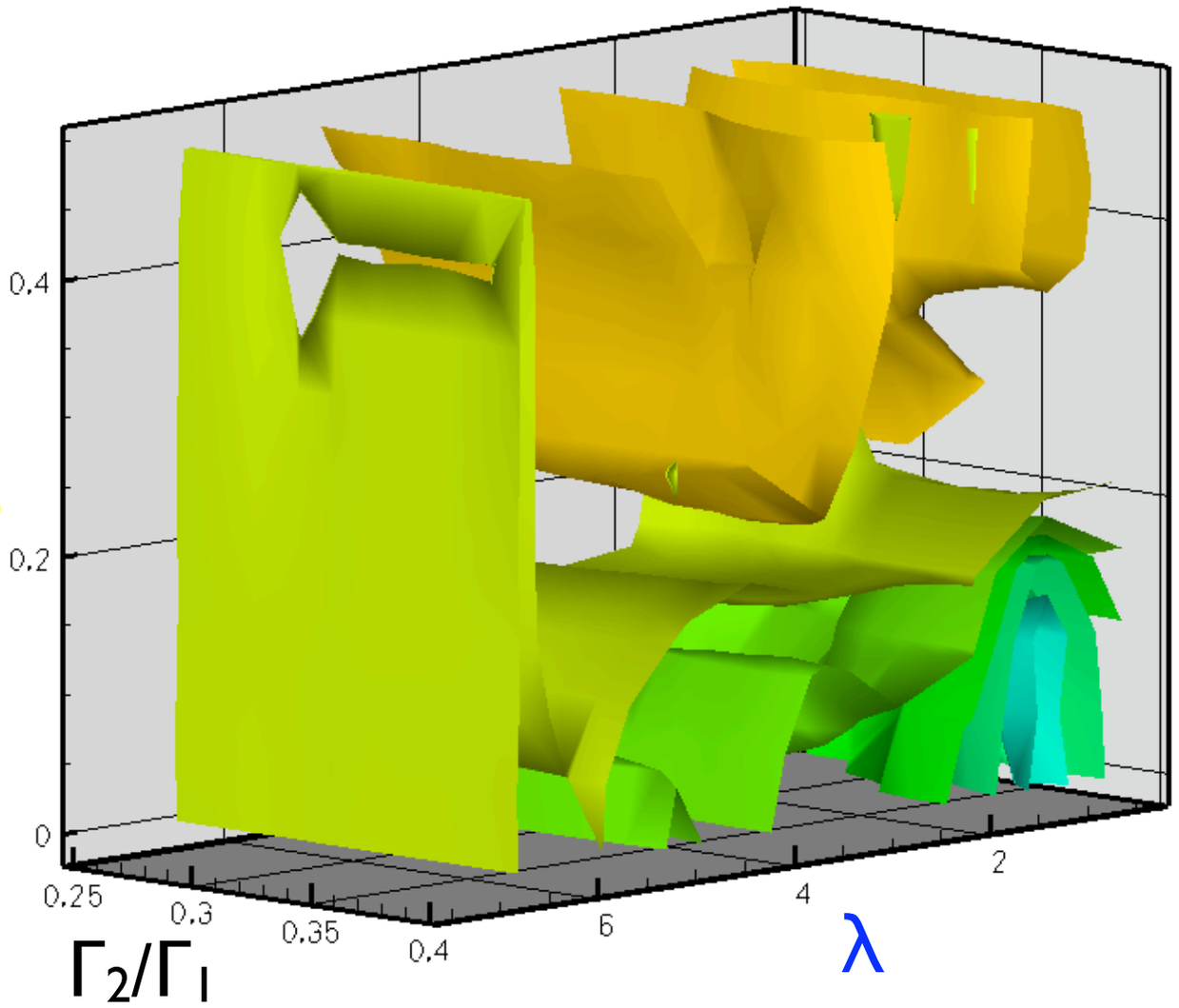
- PRESENT
  - After sweep toward large λ, moves toward smaller values
  - Convergence
- PROBLEMS
  - Resolution of Optimized Flows ?
  - Synchronize Optimization and Numerics





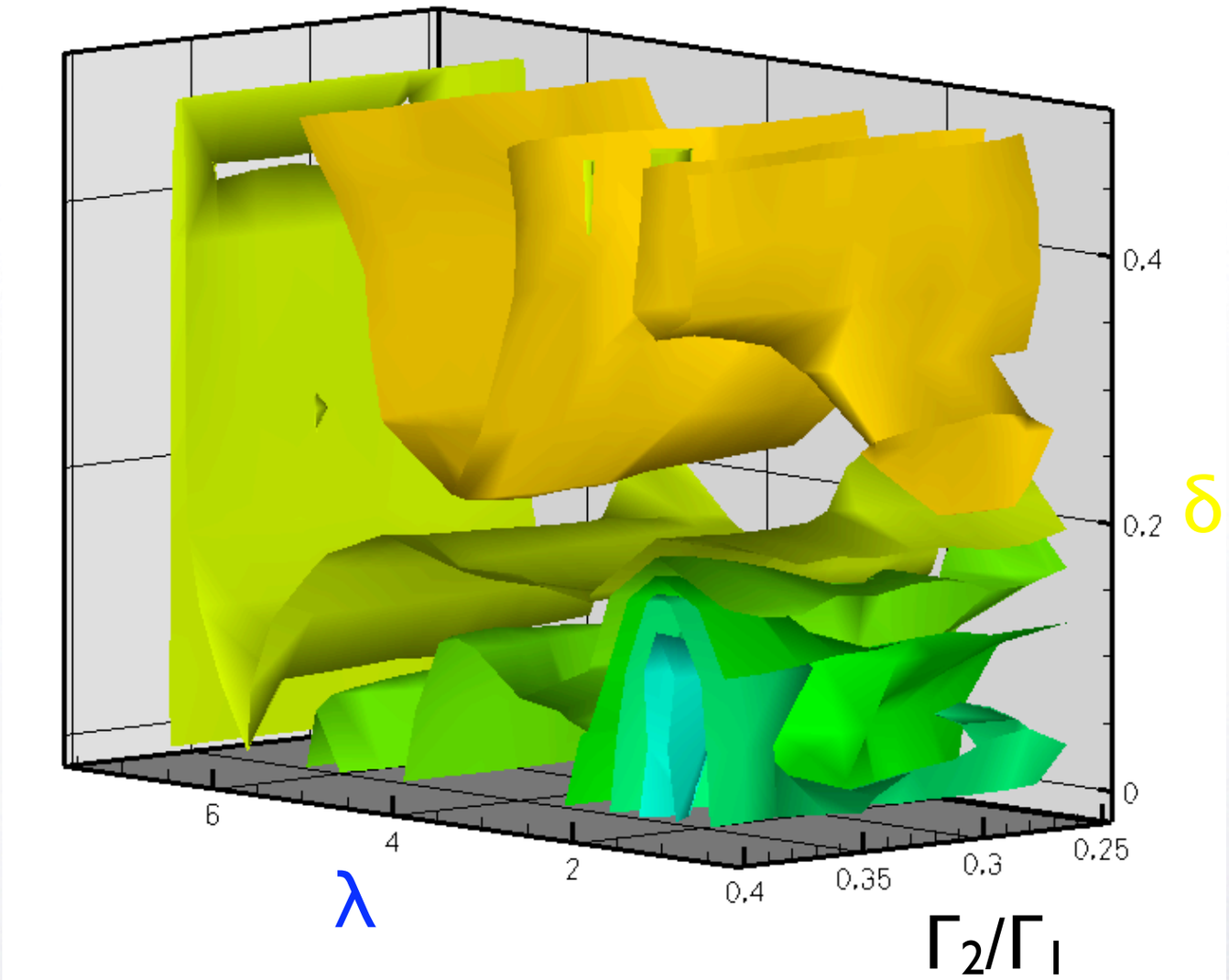
Optimization of Vortex Decay

- FUNCTION LANDSCAPE
  - in (λ,δ,Γ) hyperplane
  - Multi-modality
  - Effect of numerics?



Optimization of Vortex Decay

- FUNCTION LANDSCAPE
  - in (λ,δ,Γ) 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