NICOGRAPH International 2013
Session 5: Physics-based Simulation

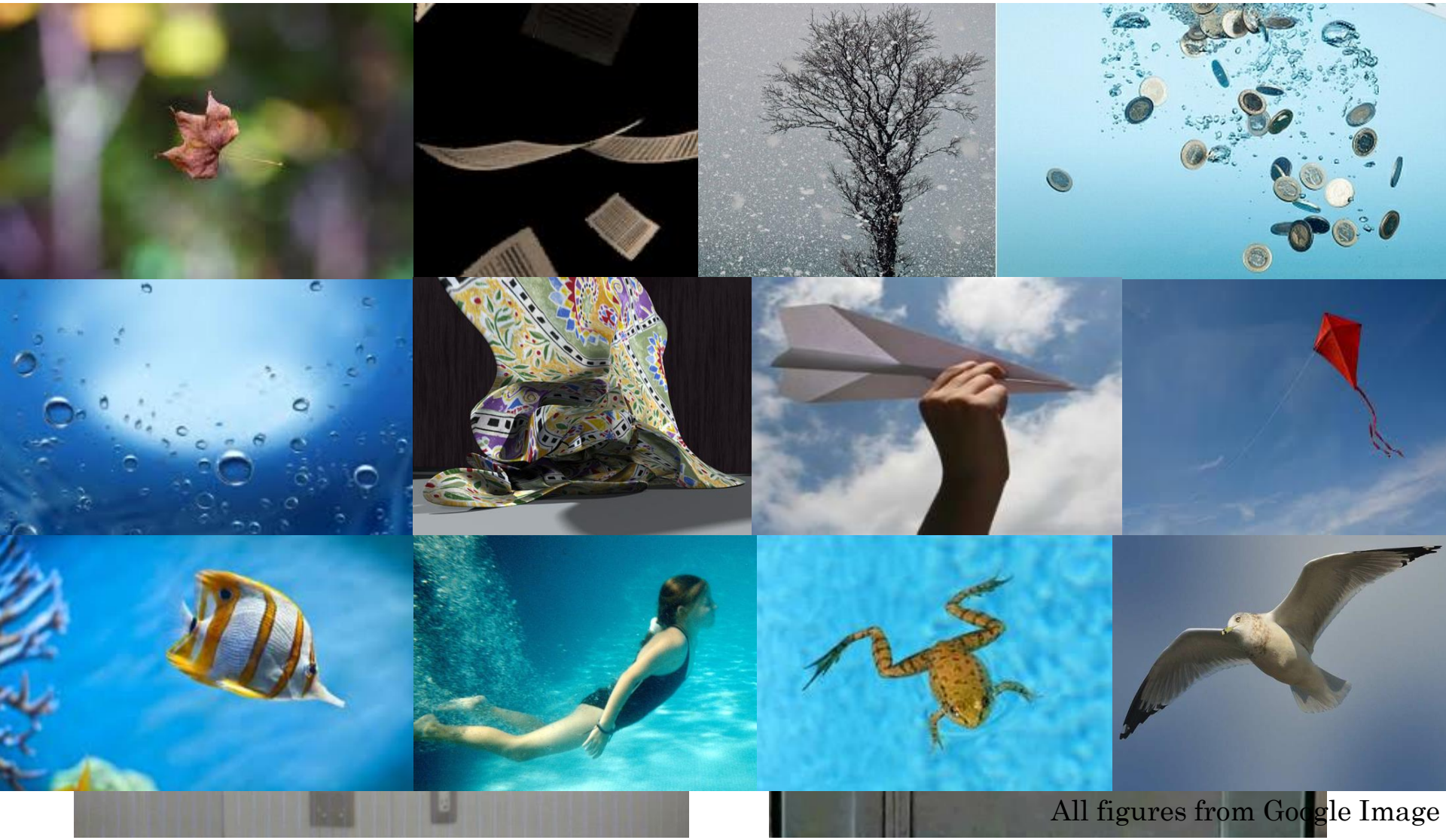
Langevin Rigid:

Animating Immersed Rigid Bodies in Real-time

Haoran Xie
Kazunori Miyata

Japan Advanced Institute of Science and Technology

Immersed Bodies

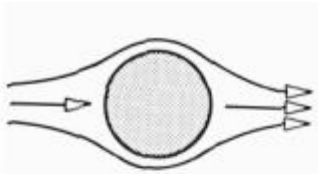


All figures from Google Image

All of them **cannot** be animated or
animated realistically by current simulation techniques in CG…

Reynolds Number<>Turbulence

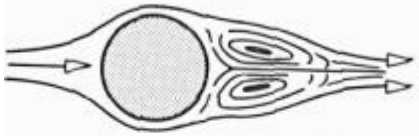
Re<5



Steady flow

$$Re = \frac{U_0 d}{\nu}$$

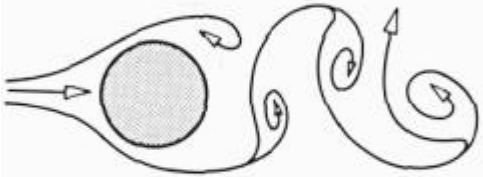
5<Re<40



Symmetric vortices

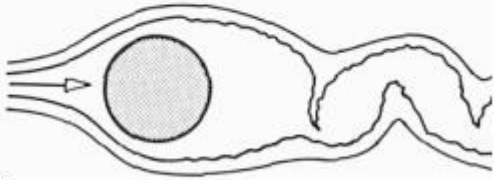
$$U_0 = \sqrt{(\bar{\rho} - 1) g b}$$

40<Re<200



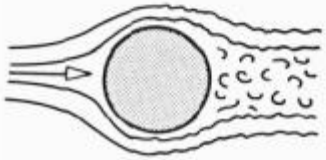
Laminar vortex sheet

200<Re<200K



Turbulent wake

200K<Re

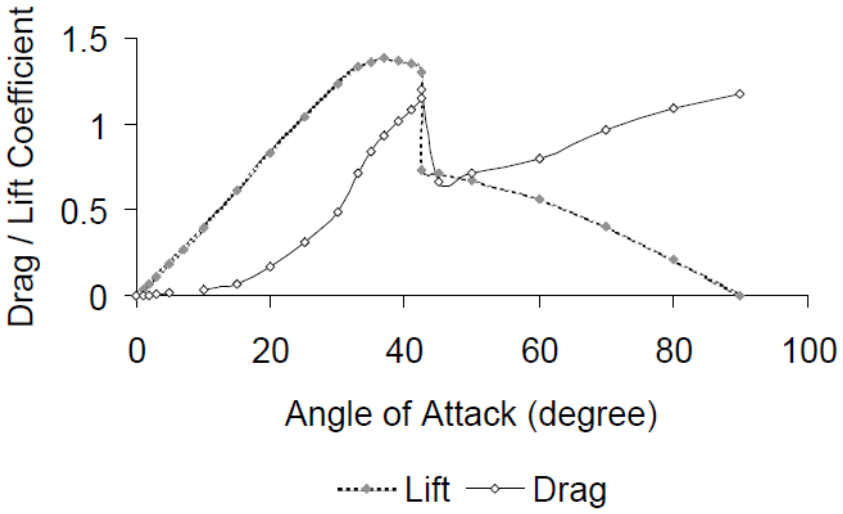


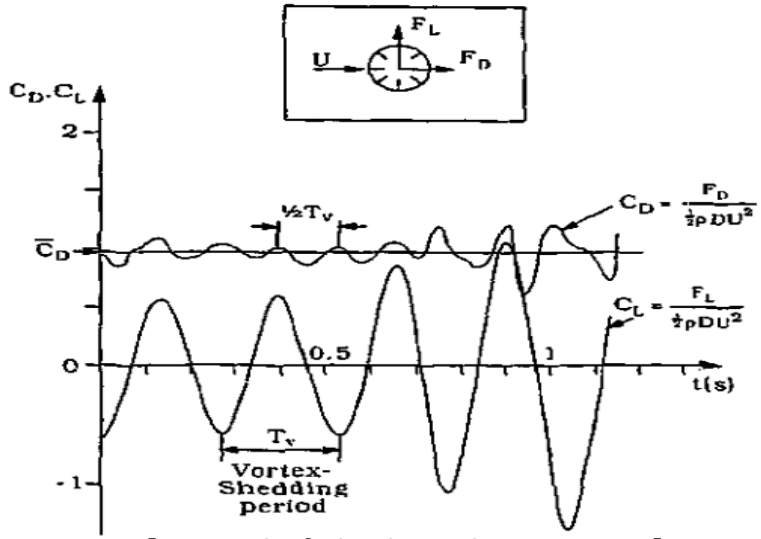
Fully turbulence

[Journal of Fluid Mechanics 1991]

Coupling between Objects and Turbulence

- One-way coupling
  - Newtonian Dynamics + Kutta-Joukowski theorem
    - Swimming [SCA04, TVCG11, SIGGRAPH11]
    - Flying [SIGGRAPH03, 09, SCA03]

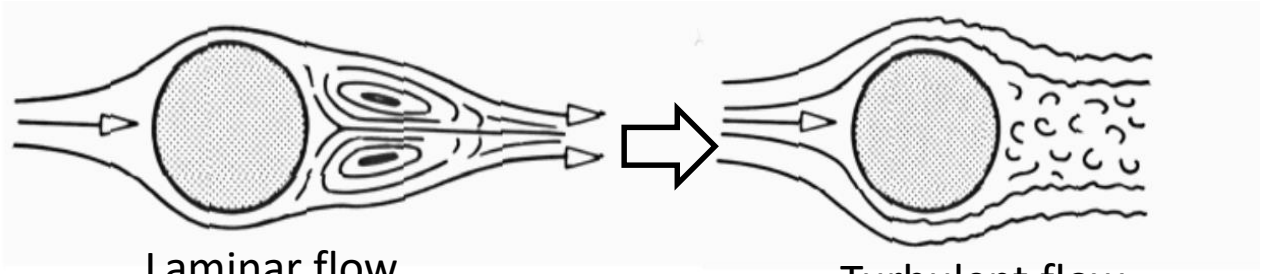




[Journal of Fluid Mechanics 1991]

- Two-way coupling
  - Newtonian Dynamics + Navier-Stokes Equations
    - Euler formulation and rigid bodies in Lagrangian formation [TOG04, 05, 06, 07, 11,SCA06, 08,]
    - Fully Langrangian meshless method [SCA05, TVCG09, SIGGRPAH12, CGF12, SA12]

Reynolds-Averaged Navier-Stokes
[SA08, SIGGRPAH09,10 Eurographics12]



Laminar flow

Turbulent flow

[Applied computational fluid dynamics,2001]

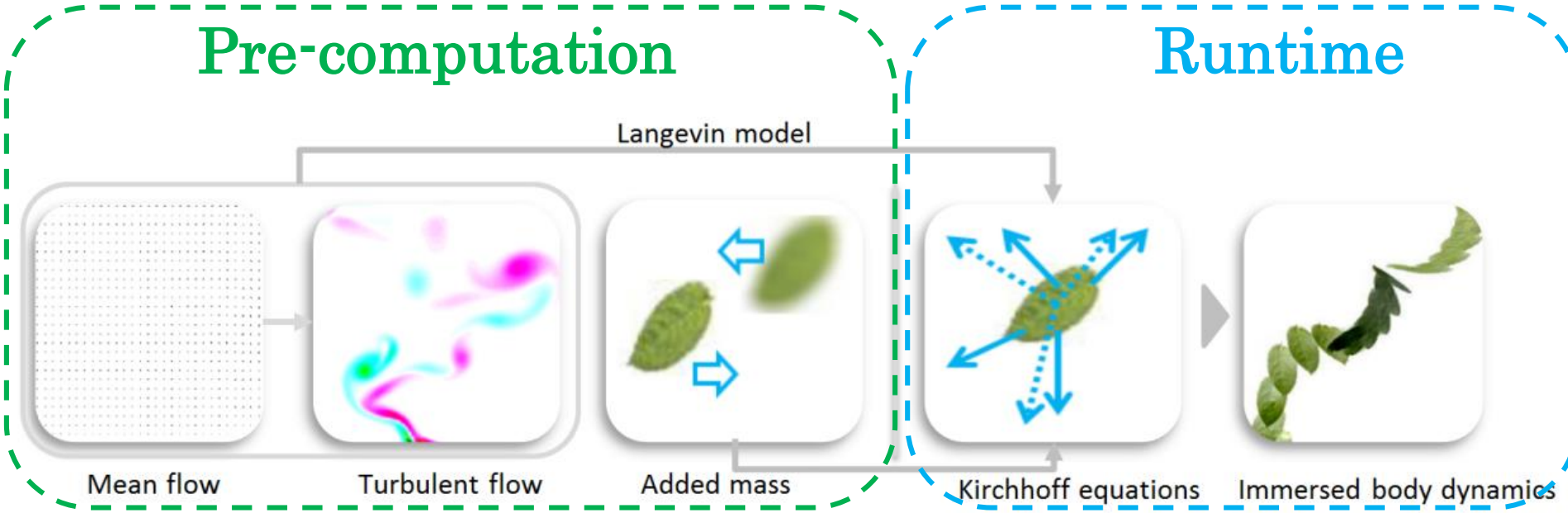
- Other approaches
  - Underwater rigid, cloth[TOG2010, SIGGRAPH2012]
    - Basset Force
    - Kirchhoff tensor

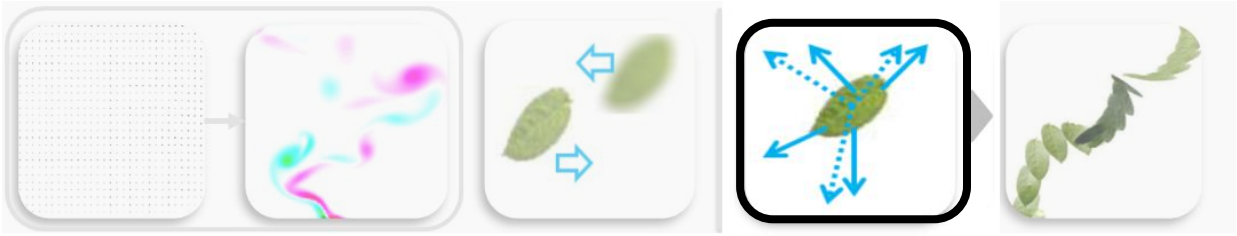
Introduction

- Real-time simulation of immersed rigid bodies
- A *Lagevin Rigid* approach
  - ◆ Coupling turbulence
    - ✓ Inertial effect ←added-mass tensors
    - ✓ Viscous effect

      ←Langevin model + Turbulence model
- New dynamical representation
  - Generalized Kirchhoff equations

Overview


Pre-computation
Runtime
Langevin model
Mean flow
Turbulent flow
Added mass
Kirchhoff equations
Immersed body dynamics

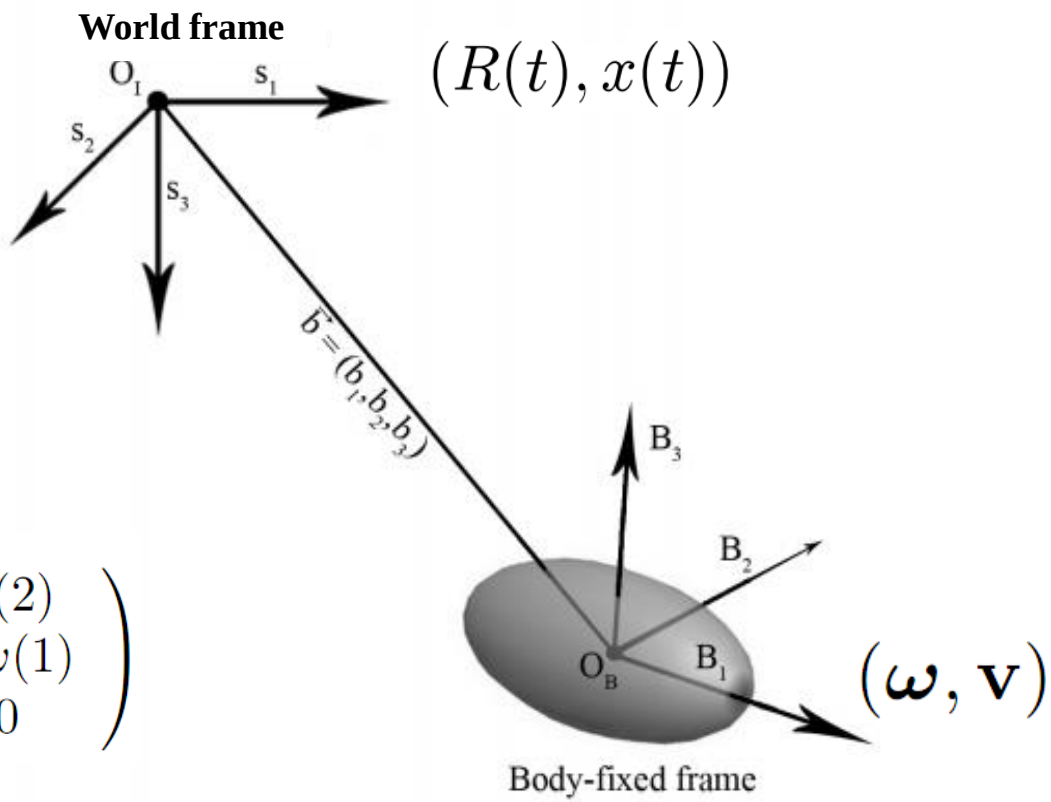


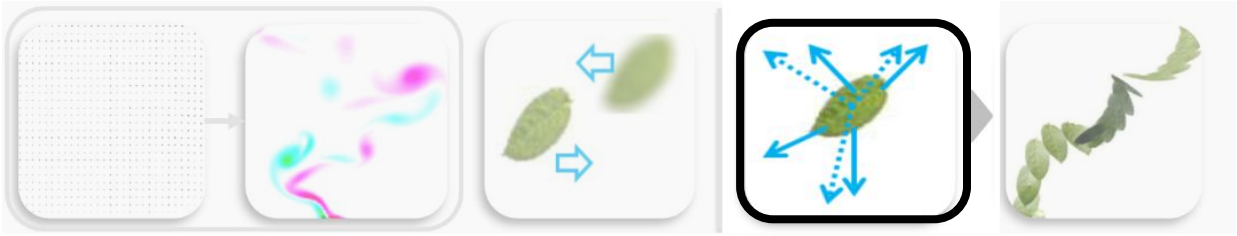
Equations of Motion

Kinematic Equations:

$$\dot{R} = R\hat{\omega},$$
$$\dot{x} = Rv$$

$$\hat{\omega} = \begin{pmatrix} 0 & -\omega(3) & \omega(2) \\ \omega(3) & 0 & -\omega(1) \\ -\omega(2) & \omega(1) & 0 \end{pmatrix}$$





Equations of Motion

Dynamic Equations:

Newton's Equations

$$M \cdot \dot{v} = F_g$$
$$J \cdot \dot{\omega} = \Gamma_g$$

Kirchhoff's Equations

$$M \cdot \dot{v} + v \times (M \cdot \omega) = 0$$
$$J \cdot \dot{\omega} + \omega \times (J \cdot \omega) + v \times (M_f \cdot v) = 0$$

Generalized Kirchhoff's Equations

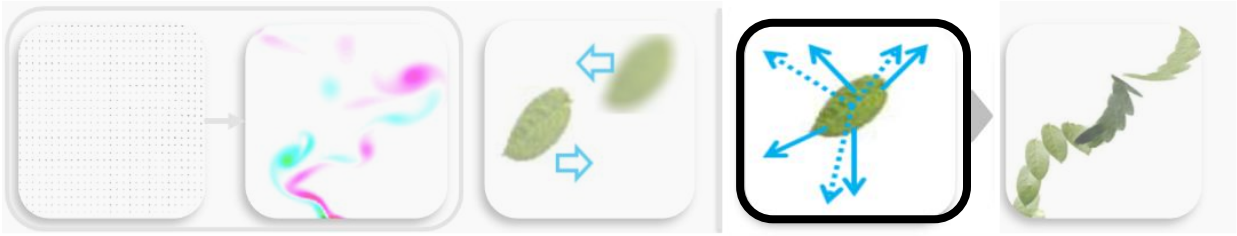
$$M \cdot \dot{v} + v \times (M \cdot \omega) = F_t + F_g$$
$$J \cdot \dot{\omega} + \omega \times (J \cdot \omega) + v \times (M_f \cdot v) = \Gamma_t + \Gamma_g$$

buoyancy-corrected gravity

Viscous effect

$$M = mI + M_f$$
$$J = J_0 + J_f$$

Inertia effect

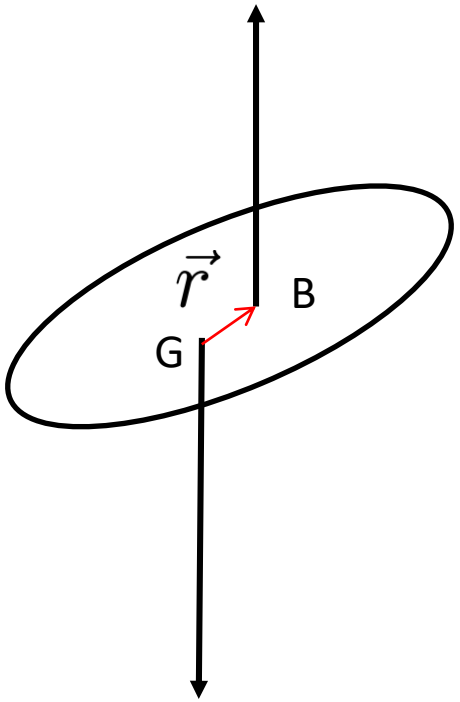


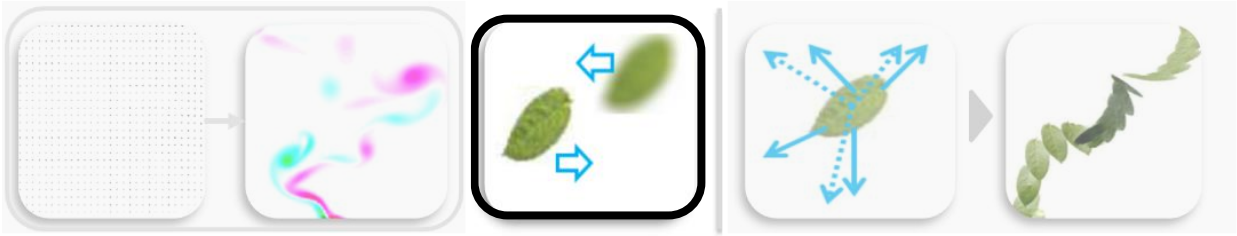
Equations of Motion

Buoyancy-corrected gravity

$$F_g = R^T(m - \rho_f V)g$$

$$\Gamma_g = \rho_f V \vec{r} \times R^T g$$





Added-mass Tensors[SIGGRAPH2012]

BD $\dfrac{\partial(\nabla\phi(p_i) - (\omega \times p_i + v))}{\partial n} = 0$

Potential $\phi(p_i) = \sum_j \dfrac{\sigma_j}{\|p_i - s_j\|}, p_i \in \partial B$

Velocity $\nabla\phi(p_i) = -\sum_j \sigma_j \dfrac{p_i - s_j}{\|p_i - s_j\|^3}$

One point quadrature

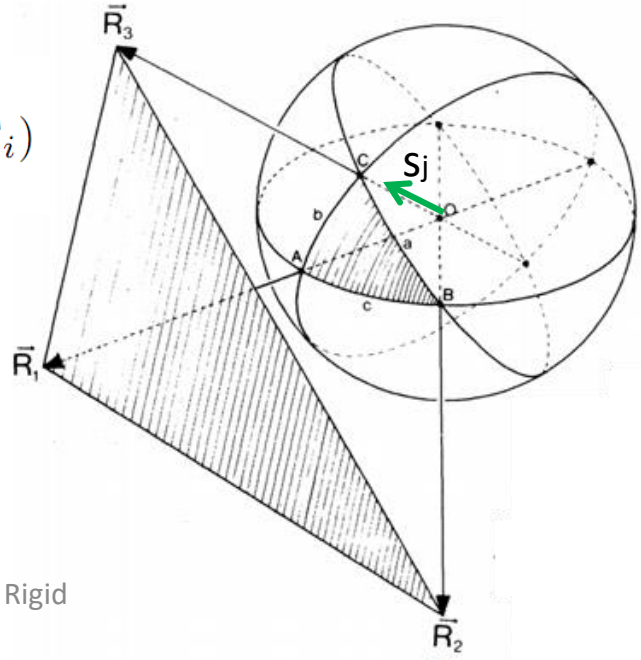
$\int_{\partial B} f(\mathbf{z})\, dA \approx \sum_i f(\mathbf{z}_i)\, \mathrm{vol}(\Gamma_i)$

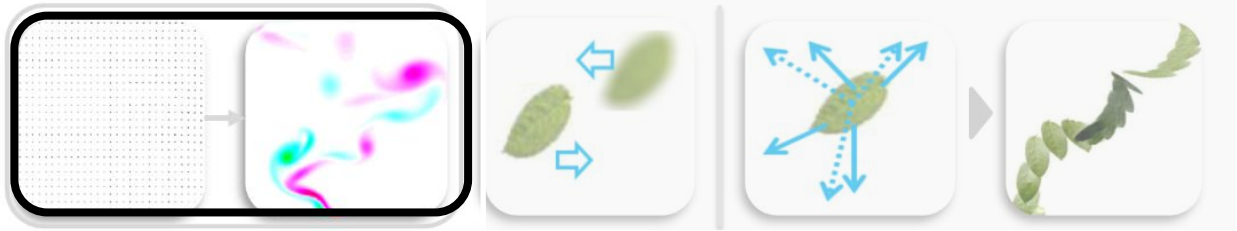
$$\begin{pmatrix} J_f \\ M_f \end{pmatrix} = \rho_f \sum_i \sum_j \frac{\sigma_j}{\|p_i - s_j\|} \begin{pmatrix} p_i \times n(p_i) \\ n(p_i) \end{pmatrix} vol(\Gamma_i)$$

where

$$\sigma = M^{-1} \sum_i \langle \omega \times p_i + v, n(p_i) \rangle vol(\Gamma_i)$$

Normal flux





Turbulent model

Navier-Stokes $\xrightarrow{(u = \langle u \rangle + u')}$ Reynolds-Averaged Navier-Stokes

$$\begin{cases} D_t\langle u\rangle = \frac{1}{\rho}\nabla P + \nabla\cdot(\nu + v_T)(\nabla\langle u\rangle + \nabla^T\langle u\rangle) \\ \nabla\cdot\langle u\rangle = 0 \end{cases}$$

Turbulent viscosity

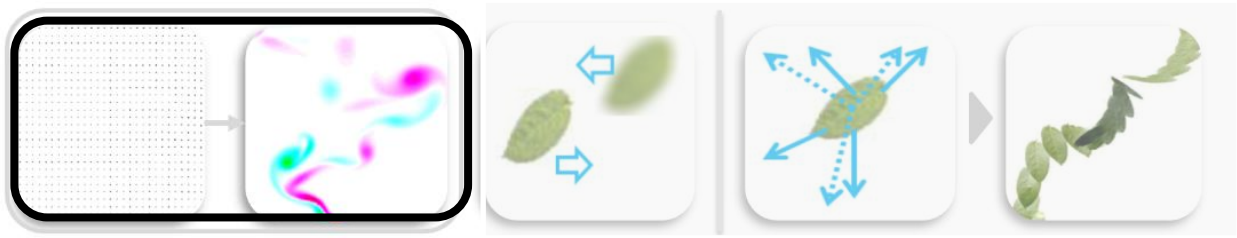
Energy transport equations

$$\begin{cases} D_t k = \nabla\cdot((\nu + \frac{v_T}{\sigma_k})\nabla k) + G - \varepsilon \\ D_t\varepsilon = \nabla\cdot((\nu + \frac{v_T}{\sigma_\varepsilon})\nabla\varepsilon) + \chi(C_1 G - C_2\varepsilon) \end{cases}$$

$$\sigma_k = 1.0,\ \sigma_\varepsilon = 1.3,\ C_1 = 1.44 \text{ and } C_2 = 1.92$$

$$v_T = C_\mu \frac{k^2}{\varepsilon}$$

$$C_\mu = 0.09$$



Turbulent model in Implementation

$$G = 2v_T \sum_{ij} S_{ij}^2$$

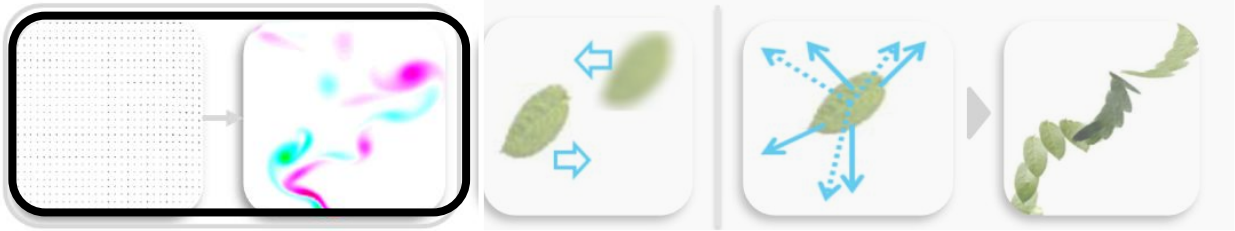
where

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \langle u \rangle_i}{\partial x_j} + \frac{\partial \langle u \rangle_j}{\partial x_i}\right)$$

$$\begin{cases} D_t k = G - \varepsilon \\ D_t \varepsilon = \chi(C_1 G - C_2 \varepsilon) \\ \chi = \varepsilon / k \end{cases}$$

Initial Conditions:

$$k_0 = \frac{3}{2} U_0^2; \qquad \varepsilon_0 = \frac{C_\mu^{\frac{3}{4}} k_0^{\frac{3}{2}}}{l}$$



Turbulent model in Implementation

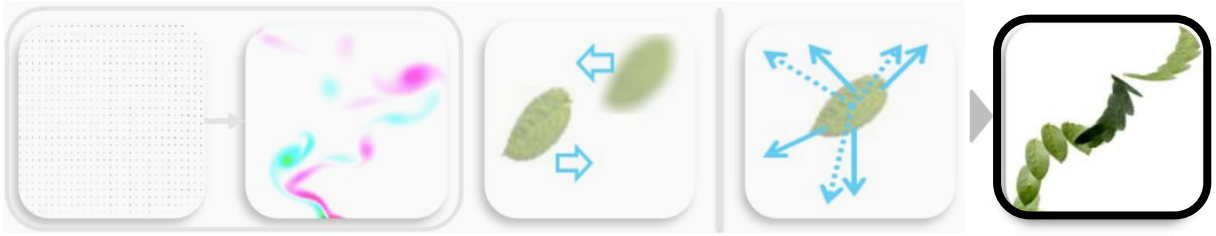
Algorithm 1 Pseudo-code for pre-generated turbulent model.

```
Boundary conditions
Timestep t = 0
while not stopped do
    // solve the mean flow ⟨u⟩
    Convection by semi-Langrangian
    Pressure projection by Poisson solver

    // Energy transport
    Compute turbulent viscosity v_T
    Compute strain tensor term G
    Integrate turbulent energy k
    Integrate dissipation rate ε

    t = t + Δt
end while
Output: (χ, ε)
```

→ Data File



Langevin Rigid

Generalized Langevin Equations

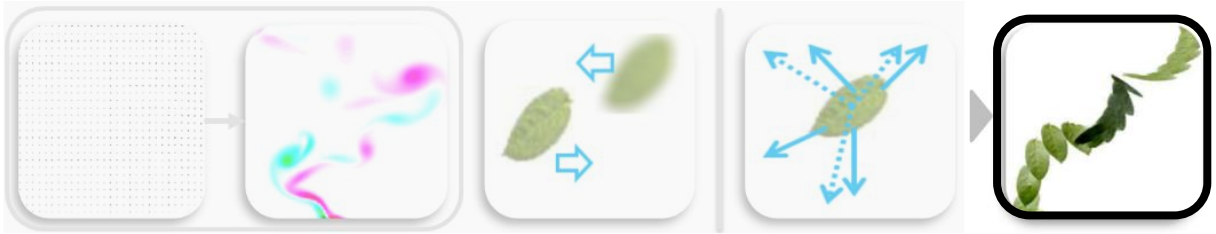
$$\begin{cases} du(t) = -\alpha u(t)dt + \beta dW \\ d\omega(t) = \beta dW \end{cases}$$

Wiener process

where

$$\alpha = (\frac{1}{2} + \frac{3}{4}C_0)\frac{\varepsilon}{k}, \qquad \beta = (C_0\varepsilon)^{\frac{1}{2}}$$

$$C_0(Re) = 6.5(1 + 140Re^{-\frac{4}{3}})^{-\frac{3}{4}}$$



Langevin Rigid

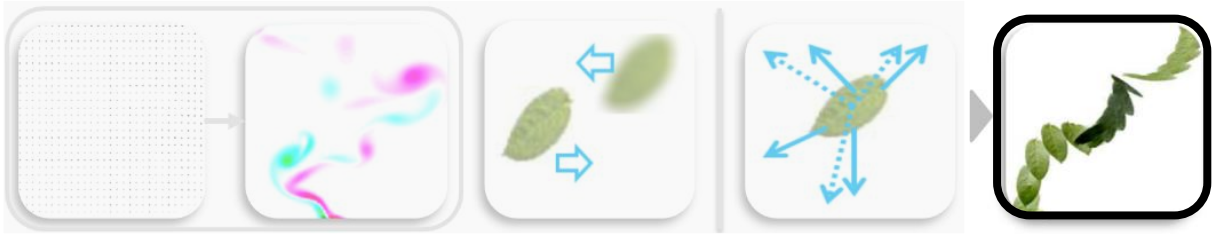
Kirchhoff + Langevin Equations

$$v(t+\Delta t) - v(t) = M^{-1}(-v(t) \times (M\omega(t))\Delta t - \chi(\frac{1}{2} + \frac{3}{4}C_0)\Delta t + (C_0\varepsilon\Delta t)^{\frac{1}{2}}\vec{\xi_1} + F_g(t)\Delta t)$$

$$\omega(t+\Delta t) - \omega(t) = J^{-1}(-\omega(t) \times (J\omega(t))\Delta t - v(t) \times (M_f v(t))\Delta t + (C_0\varepsilon\Delta t)^{\frac{1}{2}}\vec{\xi_2} + \Gamma_g(t)\Delta t)$$

$$Norm(0,1)$$

→*Standard Runge-Kutta Solver*



Langevin Rigid

Algorithm 2 Pseudo-code for the runtime computation.

```
Precompute added mass tensors
Initialization of rigid body
Timestep t = 0
while not arrive ground do
    Calculate buoyancy force and torque
    Query χ_t and ε_t (Algorithm 1)
    Compute translational velocity v
    Compute angular velocity ω
    Integrate (R, x)
    Render data
    t = t + Δt
end while
```

Simulation results

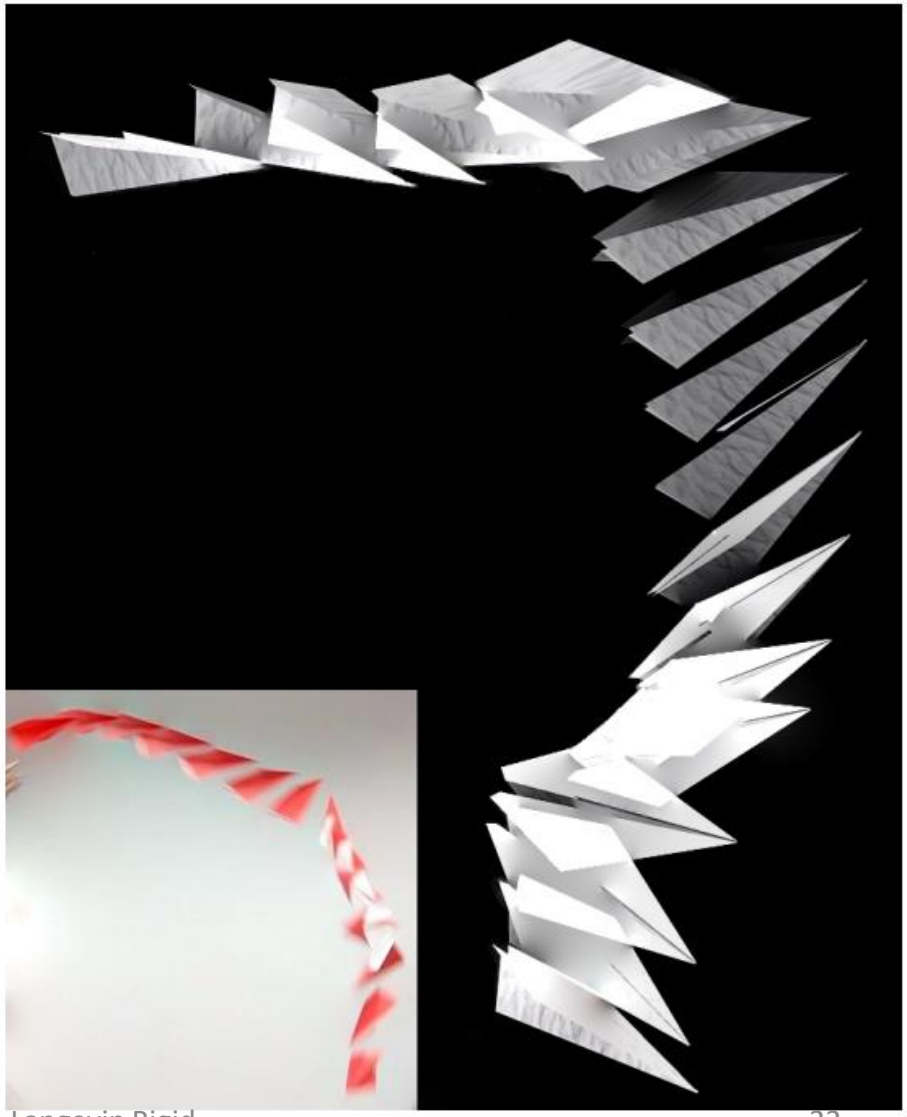
Falling paper
Elliptical area 4.0×1.0 cm
Thickness 0.01cm

Simulation results

- Falling paper

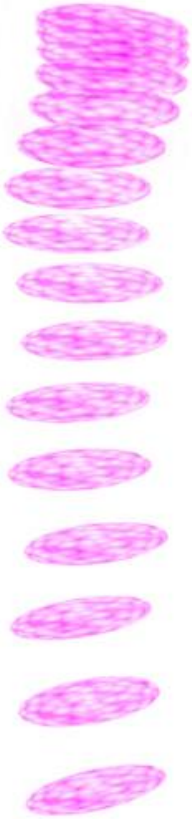
Simulation results

- Flying paper-airplane



Simulation results

- Falling rubber ellipsoid in water



[SIGGRAPH2012]

Our work

Ground truth

Computation costs

Rigid bodies	Meshes	Timestep	Average cost
Ellipsoid 1	320	1 ms	1.59 ms
Ellipsoid 1	320	5 ms	1.63 ms
Ellipsoid 1	320	10 ms	1.65 ms
Ellipsoid 2	1280	5 ms	1.71 ms
Piece of paper	1024	5 ms	1.73 ms
Paper airplane	288	5 ms	1.86 ms

Intel Core i7 CPU, 3.20 GHz, 12.0 GB RAM

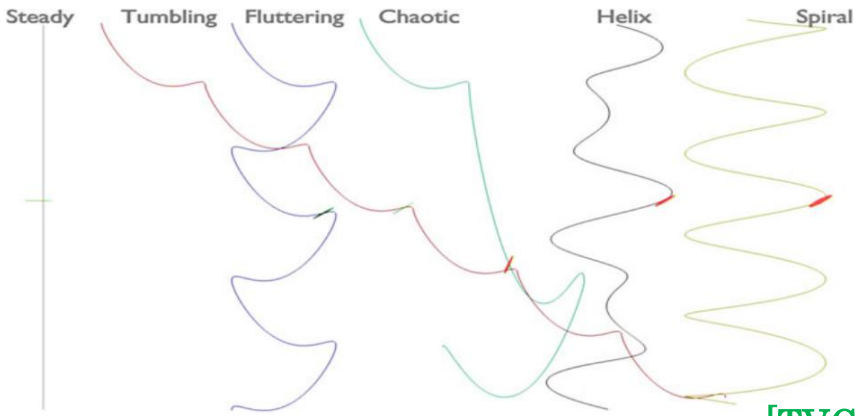
Conclusion

- **Real-time** and **realistic** immersed rigid body animation
- Pre-computation of **inertial** and **viscous** effects
- Translational and rotational **Langevin** model
- Dynamic equations in **Kirchhoff** form

Limitation and Future Work

- Hard to capture all characteristic motions

  →Hybrid method with data-driven approaches

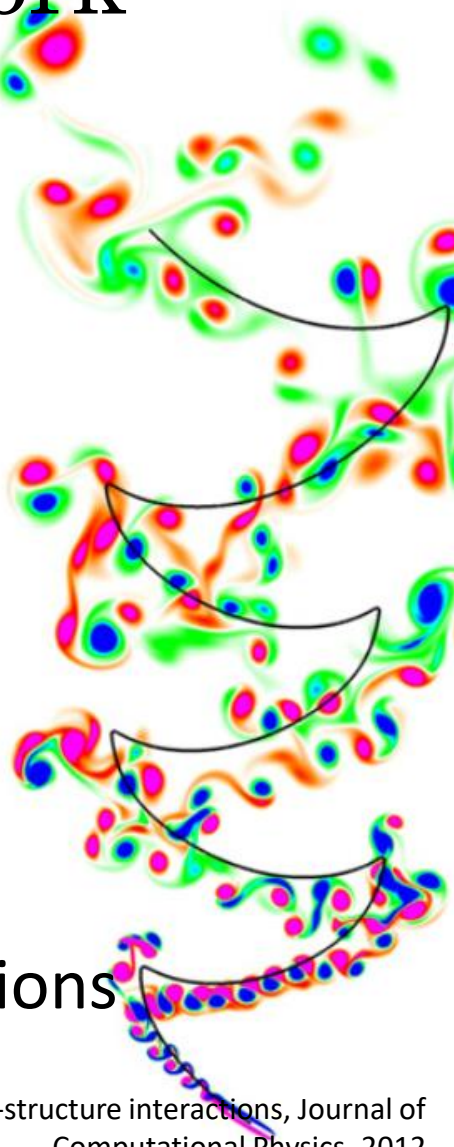


[TVC 2013]

- Sensitive to control

  →Toward a controllable animation

- Immersed deformable & character animations



※J. Yang, F., A simple and efficient direct forcing immersed boundary framework for fluid-structure interactions, Journal of Computational Physics, 2012

LANGEVIN RIGID:
ANIMATING IMMERSED RIGID BODIES IN REAL-TIME

Start from here～

THANK YOU!