



Best Practices in Software Engineering: Tools Available to Software Developers

Karla Morris

Sandia National Laboratories

Location:

Stanford University

26 February 2013

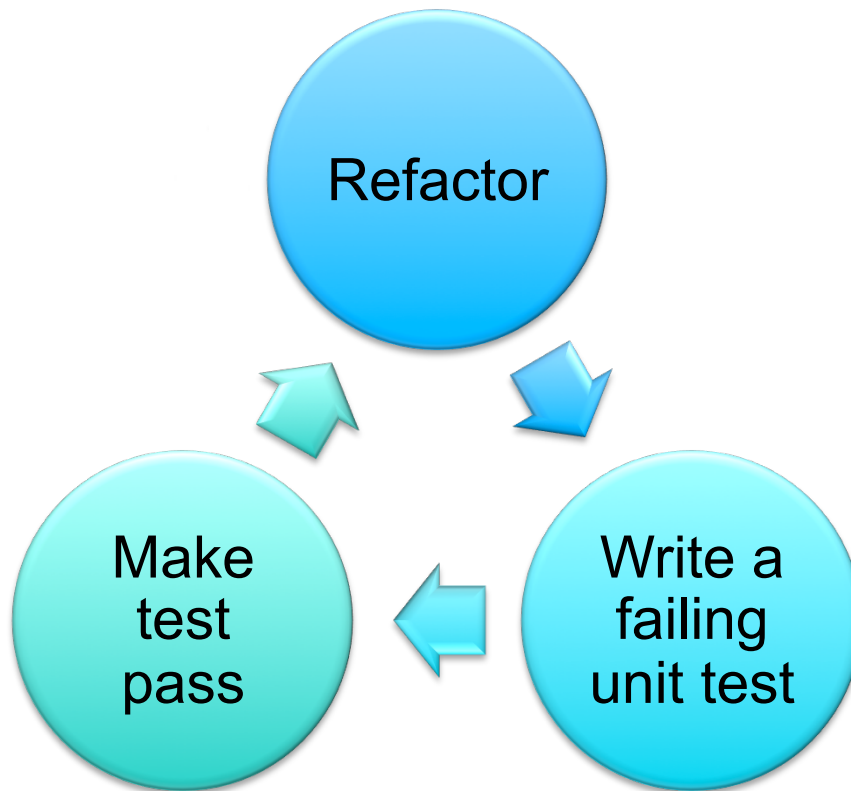


Agile Software Development

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change
- Processes and tools
- Comprehensive documentation
- Contract negotiation
- Following a plan



Test Driven Development



- Write a test for the new functionality and make design decisions regarding its possible implementation.
- Write the code required to successfully pass the test. Concentrate on current requirements.
- Refactor by doing any necessary modification to the code.



Supporting Infrastructure

- Version Control Repository:
 - GIT: <http://git-scm.com>
 - A distributed version control and source code management system
 - Master Repository
 - Multiple Local Repositories
 - Supports multiple branches in the same working directory



Supporting Infrastructure (cont.)

- Building System:
 - CMake: <http://www.cmake.org>
 - Cross-platform, open-source build system
 - Controls software compilation
 - Use platform independent configuration files
 - Use compiler independent configuration files
 - Automatically generated Makefiles



Supporting Infrastructure (cont.)

- Unit testing:
 - CTest: <http://www.cmake.org/cmake/help/v2.8.8/ctest.html>
 - Platform for verification and validation testing
 - Supports test-driven development
 - Must be used before all commits



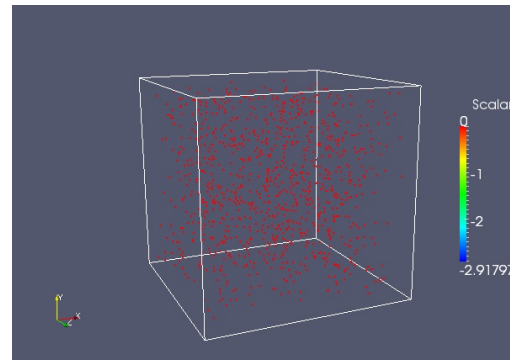
Supporting Infrastructure (cont.)

- Software documentation:
 - Doxygen: <http://www.stack.nl/~dimitri/doxygen/>
 - Tool for writing software reference documentation.
 - Documentation for all modules, objects and their functionality.
 - List of relevant publications or references of interest to the research application.



Stoke's Particles: Particle Tracking Application Prototype

- Particle dynamics can be modeled base on the following assumptions:
 - Particles are spherical (d_p)
 - Particles are small compare to the Kolmogorov scale ($d_p \ll L_k$)
 - Particles Reynolds number is small ($Re_p < 1$)
 - Density of the particles in larger than the density of the fluid ($\rho/\rho_p < 1$)
 - Particle collisions are negligible
 - Shear induced lift is neglected due to particle size





Stoke's Particles: Governing Equations

The Equation of Motion

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(t)$$

$$\frac{d\mathbf{v}}{dt} = \mathbf{g} - \frac{\tau_f}{\tau_p}(\mathbf{v} - \mathbf{u})$$

\mathbf{x} : Particle position

\mathbf{v} : Particle velocity

\mathbf{u} : Fluid velocity

\mathbf{g} : Gravitational acceleration

m : Particle mass

Euler Time Integration

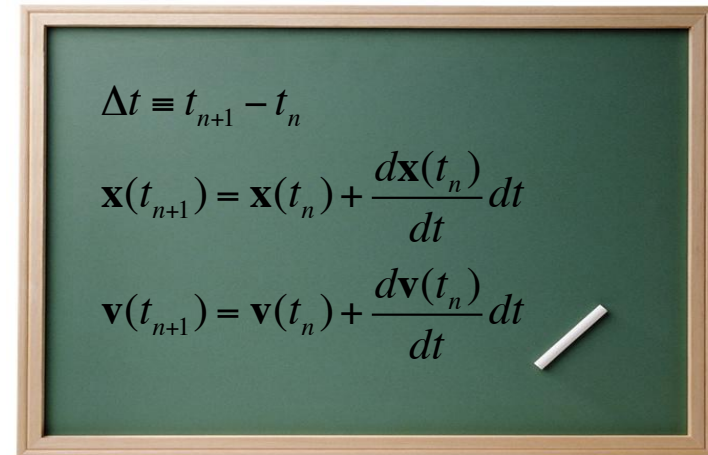
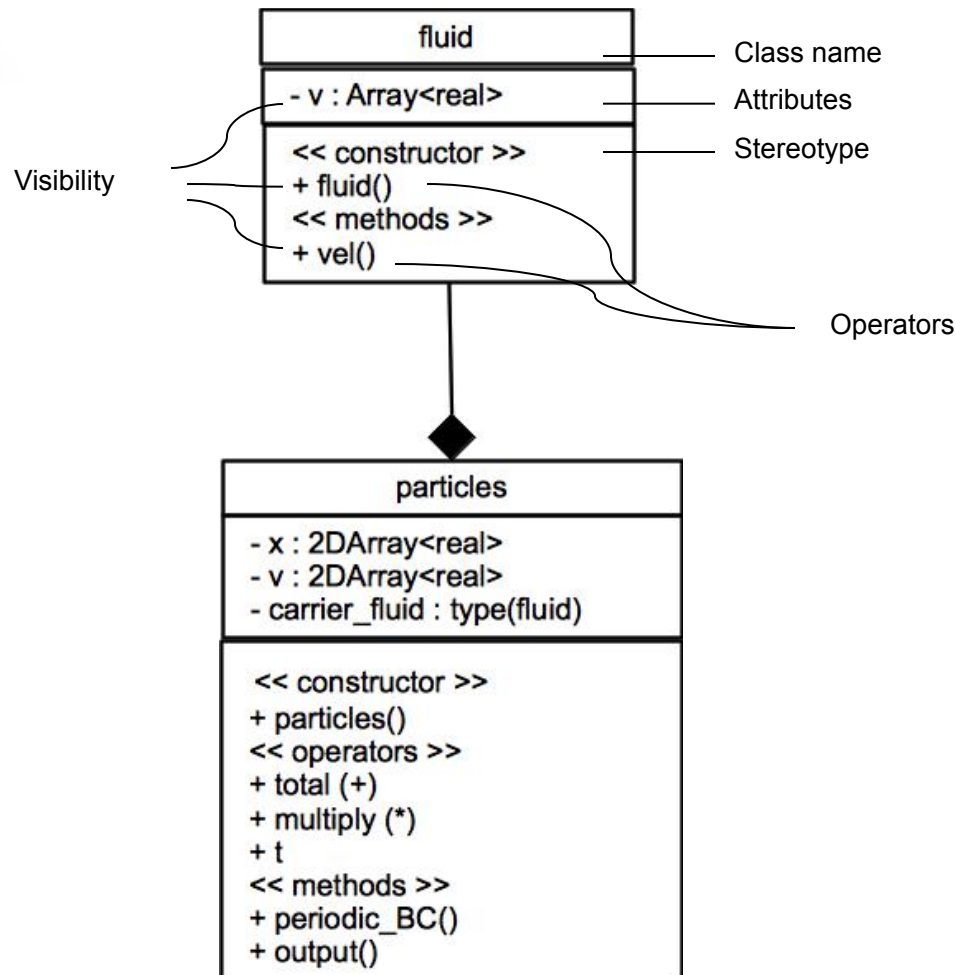
$$\Delta t \equiv t_{n+1} - t_n$$

$$\mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) + \frac{d\mathbf{x}(t_n)}{dt} dt$$

$$\mathbf{v}(t_{n+1}) = \mathbf{v}(t_n) + \frac{d\mathbf{v}(t_n)}{dt} dt$$



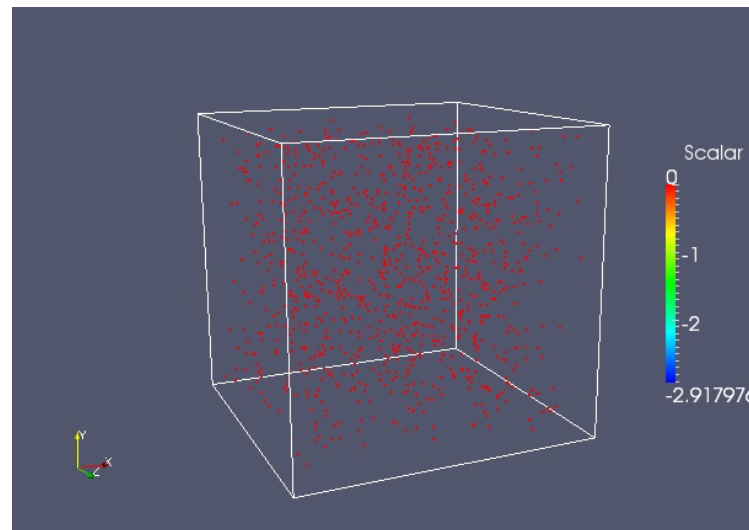
Software Design: Unified Model Language (UML) Diagram





Stoke's Particles: Applying Software Engineering Tools

- GIT repository
- CMake build
- CTest
- [Doxygen documentation](#)





Particle Tracking: Implementation

```
program main
  use math_constants, only: zero, one
  use fluid_implementation, only: fluid
  use particles_implementation, only : particles
  implicit none ! Prevents implicit declarations
  type(particles):: dust ! Particles and their properties
  type(fluid) :: gas ! Gas background fluid
  real::t=0.,t_final=10.0,dt=0.0005 ! Time,end time,time step
  gas = fluid([zero,-2.0,zero]) ! Object construction
  dust = particles(gas) ! Object construction
  do while (t<t_final)
    dust = dust + dust%t()*dt ! Advance position
    t=t+dt ! Advance time
    call dust%periodic_BC() ! Enforcing PBC
  end do
end
```



Questions and Answers

