Pacific Graphics 2016
E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

Supplementary Material for A Multilevel SPH Solver with Unified Solid Boundary Handling

1. Algorithms

For implementation of our method, we give algorithms for our incompressible SPH solver (Algorithm 1), the PPE solve (Algorithm 2), and V-cycle (Algorithm 3).

Algorithm 1 Incompressible SPH Solver

for all particle i **do**
  find neighbor fluid and solid particles
for all fluid particle i **do**
  compute density ρ_i
for all fluid particle i **do**
  predict intermediate velocity $\mathbf{u}_i^*$
for all fluid particle i **do**
  predict intermediate density ρ_i^*
Solve the PPE (Algorithm 2)
for all particle i **do**
  compute pressure force $\mathbf{F}_i^{\mathrm{p}}$
for all particle i **do**
  integrate velocity $\mathbf{u}_i$ and position $\mathbf{x}_i$

Algorithm 2 PPE Solve

for all fluid particle i **do**
  if i is Dirichlet or isolated particle **then**
    $p_i = 0$
  else
    for all neighbor solid particle s **do**
      compute $\frac{1}{\delta_s}\nabla W_{is}$
    compute $\alpha_i = \frac{\rho_0^2}{\rho_i^2}\|\sum_s \frac{1}{\delta_s}\nabla W_{is}\|^2$ and $b_i = \frac{\rho_i^* - \rho_0}{\Delta t^2}$
    if i is separated particle **then**
      $p_i = \max(0, \frac{b_i}{\alpha_i})$
    else if i is Poisson particle **then**
      for all neighbor fluid particle j **do**
        compute a_{ij}
      Assemble the matrix and source term (right hand side)
Solve the PPE with our MGCG solver
for all Poisson particle i **do**
  $p_i = \max(0, p_i)$

Algorithm 3 V_cycle(l)

//$\mathbf{r}$: residual
//k^{pre} and k^{post}: the number of pre- and post-smoothing.
if $l == 0$ **then**
  Solve $\mathbf{A}^0\mathbf{p}^0 = \mathbf{b}^0$
else
  for $k = 1$ to k^{pre} **do**
    PreSmooth($\mathbf{A}^l, \mathbf{b}^l, \mathbf{p}^l$)
  $\mathbf{r}^l = \mathbf{b}^l - \mathbf{A}^l\mathbf{p}^l$
  $\mathbf{b}^{l-1} = \mathrm{Restrict}(\mathbf{r}^l)$
  $\mathbf{p}^{l-1} = 0$
  V_cycle($l-1$)
  $\mathbf{p}^l = \mathbf{p}^l + \mathrm{Interporate}(\mathbf{p}^{l-1})$
  for $k = 1$ to k^{post} **do**
    PostSmooth($\mathbf{A}^l, \mathbf{b}^l, \mathbf{p}^l$)