Particle methods
Part 3

The Finite Volume Particle Method (FVPM)

Nathan Quinlan

Mechanical Engineering

NUI Galway
OÉ Gaillimh
New CFD should do everything classical CFD can

- **mathematically** accurate and robust
- **practically** accurate and robust
- walls
- inlets and outlets
- non-uniform resolution
- adaptive resolution
- consistency
- stability
- convergence
- conservation
- validation
Problems with basic SPH

- Low order of numerical convergence
- Sensitivity to particle distribution
- Computational cost
- No natural treatment of boundary conditions
# The lineage of FVPM

<table>
<thead>
<tr>
<th>Authors</th>
<th>Year(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hietel, Steiner, Struckmeier</td>
<td>2000</td>
<td><em>A finite-volume particle method for compressible flows</em> 2D, 1ˢᵗ-order</td>
</tr>
<tr>
<td>Junk</td>
<td>2001</td>
<td><em>Do finite volume methods need a mesh?</em></td>
</tr>
<tr>
<td>Ismagilov</td>
<td>2005</td>
<td>Smooth volume integral method 1D with MUSCL</td>
</tr>
<tr>
<td>Keck, Hietel</td>
<td>2005</td>
<td>Incompressible flow</td>
</tr>
<tr>
<td>Nestor et al.</td>
<td>2008, 2009</td>
<td>2D with MUSCL, higher order (?), viscous flow, moving body</td>
</tr>
<tr>
<td>Quinlan et al.</td>
<td>2011, 2014</td>
<td>Fast exact evaluation of particle volume and area 2D</td>
</tr>
<tr>
<td>Jahanbakhsh et al.</td>
<td>2016, 2017</td>
<td>Fast exact evaluation of particle volume and area in 3D</td>
</tr>
</tbody>
</table>
The finite volume particle method

Conservation law:
\[
\frac{dU}{dt} + \nabla \cdot \mathbf{F}(U) = 0
\]

Introduce a compactly supported test function \( \psi_i(x) \):

\[
\int_{\Omega} \psi_i \frac{dU}{dt} \, dx + \int_{\Omega} \psi_i \nabla \cdot \mathbf{F}(U) \, dx = 0
\]

Weak form:
\[
\int_{\Omega} \psi_i \frac{dU}{dt} \, dx - \int_{\Omega} \nabla \psi_i \cdot \mathbf{F}(U) \, dx = 0
\]
Choice of test function and support volume

\[
\int_{\Omega} \psi_i \frac{dU}{dt} dx - \int_{\Omega} \nabla \psi_i \cdot F(U) dx = 0
\]

\[
\psi_i(x) = \begin{cases} 
1 & x \in \Omega_i, \\
0 & \text{otherwise}
\end{cases}
\]

→ finite volume method
Choice of test function and support volume

\[ \int_{\Omega} \psi_i \frac{dU}{dt} \, dx - \int_{\Omega} \nabla \psi_i \cdot F(U) \, dx = 0 \]

\[ \psi_i(x) = \begin{cases} \frac{W_i(x)}{\sum_k W_k(x)} & x \in \Omega_i \\ 0 & \text{otherwise} \end{cases} \]

\[ \rightarrow \text{finite volume particle method} \]
Interpretation in terms of pair interactions

\[ \int_{\Omega} \psi_i \frac{dU}{dt} \, dx - \int_{\Omega} \nabla \psi_i \cdot F(U) \, dx = 0 \]

\[ \sum_j \frac{W_i(x) \nabla W_j(x) - W_j(x) \nabla W_i(x)}{\left( \sum_k W_k(x) \right)^2} = 0 \]

\[ \int_{\Omega} \psi_i \frac{dU}{dt} \, dx - \sum_j \int_{\Omega} (\gamma_{ij} - \gamma_{ji}) \cdot F(U) \, dx = 0 \]
Particle interactions

\[
\int_{\Omega} \frac{dU}{dt} \psi_i \, dx - \int_{\Omega} \nabla \psi_i \cdot F(U) \, dx - \int_{\Omega} \nabla \psi_i \cdot F(U) \, d\tau = 0
\]

\[
\sum_{j} \frac{W_i(x) \nabla W_j(x) - W_j(x) \nabla W_i(x)}{\left( \sum_{k} W_k(x) \right)^2}
\]

\[
\int_{\Omega} \frac{dU}{dt} \psi_i \, dx - \sum_{j} \int_{\Omega} (\gamma_{ij} - \gamma_{ji}) \cdot F(U) \, dx = 0
\]
3 numerical approximations

\[
\frac{d}{dt} \int_{\Omega} \psi_i U d\mathbf{x} - \sum_j \int_{\Omega} (\gamma_{ij} - \gamma_{ji}) \cdot F(U) d\mathbf{x} - \int_{\Omega} \frac{d\psi_i}{dt} U d\mathbf{x} = 0
\]

1. Replace the weighted volume average \( U \) with a "particle" value

2. Represent \( F(U(x)) \) with a single value for the overlap region

where

\[
V_i = \int_{\Omega} \psi_i d\mathbf{x}
\]

3. Reconstruct \( U_i, U_j \) at interface for Riemann problem

\[
\rightarrow F_{ij} = F(U_i, U_j)
\]
Boundary conditions

Particle support is truncated at boundary.

Compute boundary interaction vector directly...

\[ \beta_i^b = \int \frac{W_i}{\sum_j W_j(x)} n d\eta \]

...or by enforcing

\[ \sum_j \beta_{ij} + \beta_i^b = 0 \]
FVM

Classical mesh-based finite volume method: discrete cells

\[ \frac{d}{dt}(V \phi)_i \approx \sum_j F_{ij} A_{ij} \]

Exact conservation in cell-cell exchanges
FVPM vs FVM

Classical mesh-based finite volume method: discrete cells

\[ \frac{d(V\phi)_i}{dt} \approx \sum_j F_{ij} A_{ij} \]

Exact conservation in cell-cell exchanges.

\[ \frac{d(V\phi)_i}{dt} \approx \sum_j F_{ij} \beta_{ij} \]

Finite volume particle method: overlapping particles.

The classical finite volume method is a special case of FVPM.

FVPM vs FVM

Classical mesh-based finite volume method: discrete cells

\[
\frac{d(V\phi)_i}{dt} \approx \sum_j F_{ij} A_{ij}
\]

Exact conservation in cell-cell exchanges

\[
\frac{d(V\phi)_i}{dt} \approx \sum_j F_{ij} \beta_{ij}
\]

Finite volume particle method: overlapping particles.

The classical finite volume method is a special case of FVPM.

Flux functions, exactness, order of consistency

- Requirements for choice of a flux function – $G(U, U) = F(U)$ and so on
FVPM vs SPH variants

Le Touzé and NextMuSE consortium, SPHERIC Workshop (2012)
Flow around a body with prescribed motion
Test case: vortex-induced vibration
Test case: vortex-induced vibration

Nestor (2010)
Test case: vortex-induced vibration

- Amplitude of oscillations vs. vortex-shedding frequency
- Spring-mass natural frequency

Graph showing the relationship between amplitude of oscillations and vortex-shedding frequency, with data points labeled FVPM: from rest, FVPM: increasing $U_r$, and Singh & Mittal (2005): increasing $U_r$.
Test case: vortex-induced vibration
Smooth kernel functions

We evaluate $\beta$ by numerical integration (Gaussian quadrature), and it’s slow.
Top-hat kernel functions

\[ \beta_{ij} = \int \frac{W_i \nabla W_j - W_j \nabla W_i}{\left( \sum_k W_k(x) \right)^2} \, dx \]

We only have to integrate on edges (fast), and we can do it exactly!
Kernels for FVPM: summary
Top-hat kernels work

\[ L^{2} \text{norm error} \]

\[ \Delta x^{2} \text{top-hat smooth numerical error} \]

\[ \Delta x^{3} \text{particle size} \]

speedup ratio 3× to 8×
Onwards to 3D!!

<table>
<thead>
<tr>
<th>2D/3D</th>
<th>integration method</th>
<th>$h/\Delta x$</th>
<th>number of neighbours</th>
<th>number of particles</th>
<th>total time</th>
<th>time per particle</th>
<th>time per particle per neighbour</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>fast</td>
<td>0.26</td>
<td>9</td>
<td>1156</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2D</td>
<td>fast</td>
<td>0.51</td>
<td>25</td>
<td>1296</td>
<td>2.7</td>
<td>2.4</td>
<td>0.87</td>
</tr>
<tr>
<td>3D</td>
<td>fast</td>
<td>0.26</td>
<td>27</td>
<td>1728</td>
<td>62</td>
<td>41</td>
<td>14</td>
</tr>
<tr>
<td>3D</td>
<td>fast</td>
<td>0.51</td>
<td>125</td>
<td>2744</td>
<td>2050</td>
<td>864</td>
<td>173</td>
</tr>
<tr>
<td>3D</td>
<td>numerical</td>
<td>0.46</td>
<td>27</td>
<td>2744</td>
<td>187</td>
<td>79</td>
<td>26</td>
</tr>
<tr>
<td>3D</td>
<td>numerical</td>
<td>0.76</td>
<td>123</td>
<td>4096</td>
<td>3433</td>
<td>969</td>
<td>197</td>
</tr>
</tbody>
</table>

...or maybe not. At first attempt (c. 2011), 3D “fast”, on cubic particles, is 2-3 orders of magnitude slower than 2D, and not much faster than quadrature.
EPFL to the rescue!

Novel algorithms from Jahanbakhsh, Avellan et al. - exact area and volume computations extended to 3D for cubes and spheres, in practically reasonable CPU time

Jahanbakhsh et al. (2016, 2017)
EPFL to the rescue!

Jahanbakhsh et al. (2016, 2017)
Mechanical heart valve opening
Mechanical heart valve: vortex shedding

Experiment:
Bellofiore et al., Expts in Fluids (2011)
Mechanical heart valve: closing
References and further reading


