An effective improved algorithm for Finite Particle Method

†Yang Yang Xu Fei Li Xiao-ting Wang Lu

School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

†Corresponding author: nwpu_yang@mail.nwpu.edu.cn

Abstract: Finite Particle Method (FPM) is a significant improvement to the traditional SPH method, which can greatly improve the computational accuracy for boundary particles. However, in the iteration process, long computing time and potential numerical instability are the key factors restricting the application of FPM. By conducting matrix decomposition and structural analysis on the basic equations of FPM, an improved FPM method (IFPM) is proposed, which can not only maintain the high computational accuracy of FPM for boundary particles, but also avoid the restriction on the invertibility of the coefficient matrix in traditional FPM and greatly reduce the computing time. Finally, some simulation results show that IFPM is an effective improvement for traditional FPM.

Key words: Finite Particle Method; SPH; matrix decomposition; stability; accuracy

Traditional Smoothed Particle Hydrodynamics method (SPH) is the most important meshfree particle methods [Lucy. (1997); Gingold and Monaghan. (1997); Liu and Liu (2003)], which has been widely applied in Fluid Dynamics [Yang et al. (2014) ;Feng et al. (2013)], Continuum Elasticity [Liu et al. (2011)], Solid Mechanics [Libersky et al. (1993)] and so on. However, there are still some inherent defects for SPH, in which the low accuracy near the boundary or the interface is most remarkable.

In 2005, M B Liu, G R Liu and G M Zhang, R C Batra proposed a new-type SPH method based on Taylor series expansion respectively[Liu et al. (2005); Zhang and Batra (2004)], named Finite Particle Method (FPM). Compared with SPH, FPM has the advantages of free selection on the basis function, high accuracy near the boundary, and it is also not sensitive to the smooth length and the irregular distribution of particles. In addition, FPM could get the function value and derivative value simultaneously, which avoids the error propagation when the low-order derivative is used in the calculation of the high-order derivative in SPH.

However, there are still two disadvantages for FPM, which are the long computation time and computational instability. The former is caused by the large amount of calculation on solving the linear equations for each particle in the computational domain, and the latter is because the invertibility of the coefficient matrix in the linear equations cannot be well satisfied all the time. Therefore, in this paper, an improved algorithm for FPM is proposed, which is abbreviated to IFPM and is proven to cover the shortage in FPM effectively.

1. Original FPM

In 1D case, considering a Taylor series expansion at x_i up to the first-order derivative,

$$f(x) = f(x_i) + (x - x_i) f_x(x_i) \tag{1}$$

Multiplying both sides of (1) with the basis function $\varphi_1(x)$ and $\varphi_2(x)$ respectively, and integrating in the computational domain,

$$\begin{aligned} \int f(x)\varphi_1(x)\mathrm{d}x &= f(x_i)\int \varphi_1(x)\mathrm{d}x + f_x(x_i)\int (x - x_i)\varphi_1(x)\mathrm{d}x \\ \int f(x)\varphi_2(x)\mathrm{d}x &= f(x_i)\int \varphi_2(x)\mathrm{d}x + f_x(x_i)\int (x - x_i)\varphi_2(x)\mathrm{d}x \end{aligned} \tag{2}$$

Expression (2) could be seemed as linear equations, and expressed as the following matrix form,

$$\begin{bmatrix} \int \varphi_1(x)\mathrm{d}x & \int (x - x_i)\varphi_1(x)\mathrm{d}x \\ \int \varphi_2(x)\mathrm{d}x & \int (x - x_i)\varphi_2(x)\mathrm{d}x \end{bmatrix} \begin{bmatrix} f(x_i) \\ f_x(x_i) \end{bmatrix} = \begin{bmatrix} \int f(x)\varphi_1(x)\mathrm{d}x \\ \int f(x)\varphi_2(x)\mathrm{d}x \end{bmatrix} \tag{3}$$

Further, the particle form of the above equations (3) can be obtained as follows,

$$\begin{bmatrix} \sum_{j=1}^{N} \varphi_1(x_j)\Delta d_j & \sum_{j=1}^{N} (x_j - x_i)\varphi_1(x_j)\Delta d_j \\ \sum_{j=1}^{N} \varphi_2(x_j)\Delta d_j & \sum_{j=1}^{N} (x_j - x_i)\varphi_2(x_j)\Delta d_j \end{bmatrix} \begin{bmatrix} f(x_i) \\ f_x(x_i) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N} f(x_j)\varphi_1(x_j)\Delta d_j \\ \sum_{j=1}^{N} f(x_j)\varphi_2(x_j)\Delta d_j \end{bmatrix} \tag{4}$$

where Δd_j is the particle size. Equation (4) is the basic expression of FPM in 1D case.

Similarly, the basic expression of FPM in 2D case could be derived as follows,

$$\begin{bmatrix} \sum_{j=1}^{N} \varphi_1(x_j, y_j)\Delta S_j & \sum_{j=1}^{N} (x_j - x_i)\varphi_1(x_j, y_j)\Delta S_j & \sum_{j=1}^{N} (y - y_i)\varphi_1(x_j, y_j)\Delta S_j \\ \sum_{j=1}^{N} \varphi_2(x_j, y_j)\Delta S_j & \sum_{j=1}^{N} (x_j - x_i)\varphi_2(x_j, y_j)\Delta S_j & \sum_{j=1}^{N} (y - y_i)\varphi_2(x_j, y_j)\Delta S_j \\ \sum_{j=1}^{N} \varphi_3(x_j, y_j)\Delta S_j & \sum_{j=1}^{N} (x_j - x_i)\varphi_3(x_j, y_j)\Delta S_j & \sum_{j=1}^{N} (y - y_i)\varphi_3(x_j, y_j)\Delta S_j \end{bmatrix} \begin{bmatrix} f(x_i, y_i) \\ f_x(x_i, y_i) \\ f_y(x_i, y_i) \end{bmatrix} = \\ \begin{bmatrix} \sum_{j=1}^{N} f(x_j, y_j)\varphi_1(x_j, y_j)\Delta S_j \\ \sum_{j=1}^{N} f(x_j, y_j)\varphi_2(x_j, y_j)\Delta S_j \\ \sum_{j=1}^{N} f(x_j, y_j)\varphi_3(x_j, y_j)\Delta S_j \end{bmatrix} \tag{5}$$

where ΔS_j is the particle size.

2. IFPM

2.1 1D case

In 1D case, IFPM could be derived based on the matrix decomposition on the coefficient matrix and the constant term in the basic FPM equation (4).

First, the coefficient matrix decomposition:

$$A=\begin{bmatrix}\sum_{j=1}^{N}\varphi_1(x_j)\Delta d_j & \sum_{j=1}^{N}(x_j-x_i)\varphi_1(x_j)\Delta d_j\\ \sum_{j=1}^{N}\varphi_2(x_j)\Delta d_j & \sum_{j=1}^{N}(x_j-x_i)\varphi_2(x_j)\Delta d_j\end{bmatrix}= \begin{bmatrix}\varphi_1(x_1) & \varphi_1(x_2) & \cdots & \varphi_1(x_N)\\ \varphi_2(x_1) & \varphi_2(x_2) & \cdots & \varphi_2(x_N)\end{bmatrix}\begin{bmatrix}\Delta d_1 & & & \\ & \Delta d_2 & & \\ & & \ddots & \\ & & & \Delta d_N\end{bmatrix}\begin{bmatrix}1 & x_1-x_i\\ 1 & x_2-x_i\\ \vdots & \vdots\\ 1 & x_N-x_i\end{bmatrix}\triangleq K\cdot D\cdot C \tag{6}$$

Second, the constant term decomposition:

$$B=\begin{bmatrix}\sum_{j=1}^{N}f(x_j)\varphi_1(x_j)\Delta d_j\\ \sum_{j=1}^{N}f(x_j)\varphi_2(x_j)\Delta d_j\end{bmatrix}= \begin{bmatrix}\varphi_1(x_1) & \varphi_1(x_2) & \cdots & \varphi_1(x_N)\\ \varphi_2(x_1) & \varphi_2(x_2) & \cdots & \varphi_2(x_N)\end{bmatrix}\begin{bmatrix}\Delta d_1 & & & \\ & \Delta d_2 & & \\ & & \ddots & \\ & & & \Delta d_N\end{bmatrix}\begin{bmatrix}f(x_1)\\ f(x_2)\\ \vdots\\ f(x_N)\end{bmatrix}\triangleq K\cdot D\cdot F \tag{7}$$

Based on (6) and (7), the basic FPM equation (4) could be expressed as follows,

$$KDCf = KDF \tag{8}$$

where $f=\begin{bmatrix}f(x_i) & f_x(x_i)\end{bmatrix}^T$, N is the number of the particles in the support of x_i.

Solving equations (8) is equivalent to solve the following equations,

$$KD(Cf-F)=0 \tag{9}$$

Since FPM is free on the selection of the basis function, we could just consider the case *rank*(K) = 2, i.e. K is a row full-rank matrix. Specially, as shown in Figure 1, if just two nearest particles to x_i in its support (i.e. N = 2) are chosen and introduced into the approximate calculation equations (9) of x_i, the matrix K, C and F have the following reduced forms,

$$K=\begin{bmatrix}\varphi_1(x_1) & \varphi_1(x_2)\\ \varphi_2(x_1) & \varphi_2(x_2)\end{bmatrix},\quad D=\begin{bmatrix}\Delta d_1 & 0\\ 0 & \Delta d_2\end{bmatrix},\quad C=\begin{bmatrix}1 & x_1-x_i\\ 1 & x_2-x_i\end{bmatrix},\quad F=\begin{bmatrix}f(x_1)\\ f(x_2)\end{bmatrix} \tag{10}$$

Here, the matrices K and D are invertible, and thus the equations (8) is equivalent to the following reduced equations (11),

$$Cf-F=0 \tag{11}$$

Equations (11) are the basic equations in IFPM. It can be clearly found that the coefficient matrix of equations (11) is always invertible, and thus 1D-IFPM is always stable.

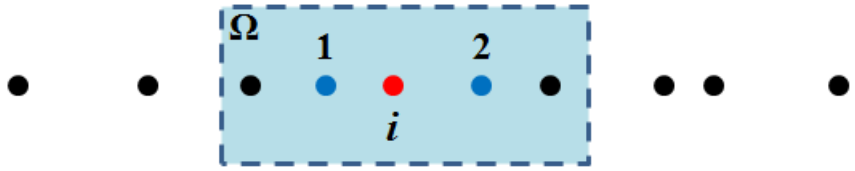

Figure 1. The schema on the selection of computational particles in 1D-IFPM

2.2 2D case

Similar to section 2.1, 2D-IFPM equation could be obtained after the decomposition and deformation to equation (5). The difference is that, as shown in Figure 2, three nearest particles to x_i in its support (i.e. N = 3) should be chosen and introduced into the approximate calculation equations in order to keep the matrix K is a square matrix. The basic equation of 2D-IFPM are shown as follows (12),

$$\begin{bmatrix}1 & x_1-x_i & y_1-y_i\\ 1 & x_2-x_i & y_2-y_i\\ 1 & x_3-x_i & y_3-y_i\end{bmatrix}\begin{bmatrix}f(x_i,y_i)\\ f_x(x_i,y_i)\\ f_y(x_i,y_i)\end{bmatrix}=\begin{bmatrix}f(x_1,y_1)\\ f(x_2,y_2)\\ f(x_3,y_3)\end{bmatrix} \tag{12}$$

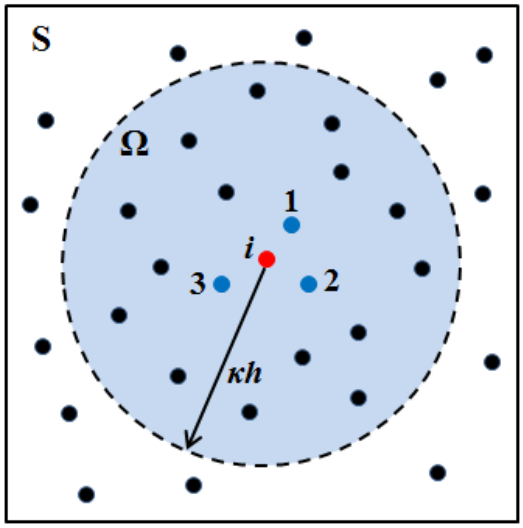

Figure 2. The schema on the selection of computational particles in 2D-IFPM

It can be also clearly found that the coefficient matrix of equations (12) is always invertible, and thus 2D-IFPM is always stable.

Meanwhile, it must be mentioned that the uniform distribution of particles is actually a special case of random distribution, thus the above IFPM method is also suitable for uniformly distributed particles.

3. Numerical tests and analysis

3.1 Particle approximation accuracy

Generally speaking, if one method could reproduce a k-order polynomial, this method could be called with C^k continuity. In order to verify the accuracy of IFPM, the following examples are tested. For 1D case, considering the function $f(x)=1$ and $f(x)=x$ in [0,10] respectively, 11 particles are randomly distributed and their coordinates are shown as follows,

(R): 0.3770, 2.3160, 3.9550, 4.8890, 6.2410, 6.7910, 7.9620, 8.8520, 9.1330, 9.8800

The smooth length h=1, and the cubic B-spline function and its derivative function are chosen as the kernel function for SPH and the basic functions for FPM [Monaghan. (1992)]. The numerical results are shown in the Table 1~2, where *Error* is defined as the difference between the reproduced value and the exact value, i.e. $Error=\hat{f}-f$. It can be seen that both IFPM and FPM have distinct accuracy compared with SPH method. They could reproduce both the interior particles and the particles near the boundary accurately, which means they could be called with C^1 continuity.

Table 1. Reproduced results for $f(x)=1$

Exact f	Reproduced results and error					
	SPH		FPM		IFPM	
	$\hat{f}$	*Error*	$\hat{f}$	*Error*/$\times10^{-15}$	$\hat{f}$	*Error*/$\times10^{-15}$
1	0.8977	-0.1023	1.0000	0	1.0000	-0.2220
1	1.2028	0.2028	1.0000	0	1.0000	0
1	1.1022	0.1022	1.0000	0	1.0000	0
1	1.0648	0.0648	1.0000	-0.1110	1.0000	0
1	1.0745	0.0745	1.0000	-0.2220	1.0000	0
1	1.0971	0.0971	1.0000	0	1.0000	0
1	0.9539	-0.0461	1.0000	0	1.0000	0
1	1.0076	0.0076	1.0000	0	1.0000	0
1	0.9476	-0.0524	1.0000	-0.1110	1.0000	0
1	0.5817	-0.4183	1.0000	0	1.0000	0
1	0.8977	-0.1023	1.0000	0	1.0000	-0.2220

Table 2. Reproduced results for $f(x)=x$

Exact f	Reproduced results and error					
	SPH		FPM		IFPM	
	$\hat{f}$	*Error*	$\hat{f}$	*Error*/$\times 10^{-14}$	$\hat{f}$	*Error*/$\times 10^{-15}$
0.3770	0.3386	-0.0384	0.3770	-0.0056	0.3770	-0.2220
2.3160	2.8021	0.4861	2.3160	0	2.3160	0
3.9550	4.5517	0.5967	3.9550	0.0888	3.9550	0
4.8890	5.0218	0.1328	4.8890	-0.0888	4.8890	0
6.2410	6.8541	0.6131	6.2410	-0.0888	6.2410	0
6.7910	7.3309	0.5399	6.7910	0	6.7910	0
7.9620	7.6692	-0.2928	7.9620	0	7.9620	0
8.8520	8.8754	0.0234	8.8520	0	8.8520	0
9.1330	8.5581	-0.5749	9.1330	-0.1776	9.1330	0
9.8800	5.5330	-4.3470	9.8800	0	9.8800	0
0.3770	0.3386	-0.0384	0.3770	-0.0056	0.3770	-0.2220

For 2D case, considering the function $f(x,y)=1$ and $f(x,y)=x+y$ in $[0,100]\times[0,100]$ respectively, 5664 particles are randomly distributed. The numerical results are shown in the Table 3, where *MSE* represents the Mean Square Error. It can be seen that IFPM could also keep C^1 continuity in 2D case.

Table 3. Reproduced results in 2D case

Function type	*MSE*		
	SPH	FPM	IFPM
$f(x,y)=1$	0.0357	1.5834×10^{-32}	7.4650×10^{-31}
$f(x,y)=x+y$	427.7285	3.8521×10^{-28}	2.1792×10^{-27}

3.2 Computation time analysis

In order to compare the computation time among the SPH, FPM and IFPM, we consider the function $f(x)=x$ in the interval [0, 100]. The computation time of three methods with increasing of the number of particles in the computational domain from 11 to 10001 are shown in Figure 3.

It is shown that with increasing of the particle number, the computation time of FPM increases rapidly, and SPH method increases steadily. While the computation time of IFPM proposed in this paper has no obvious change. When the number of particles increases from 11 to 10001, the computation time just increases from 0.0002s to 0.1176s. Therefore, IFPM method can greatly reduce the computation time compared with FPM.

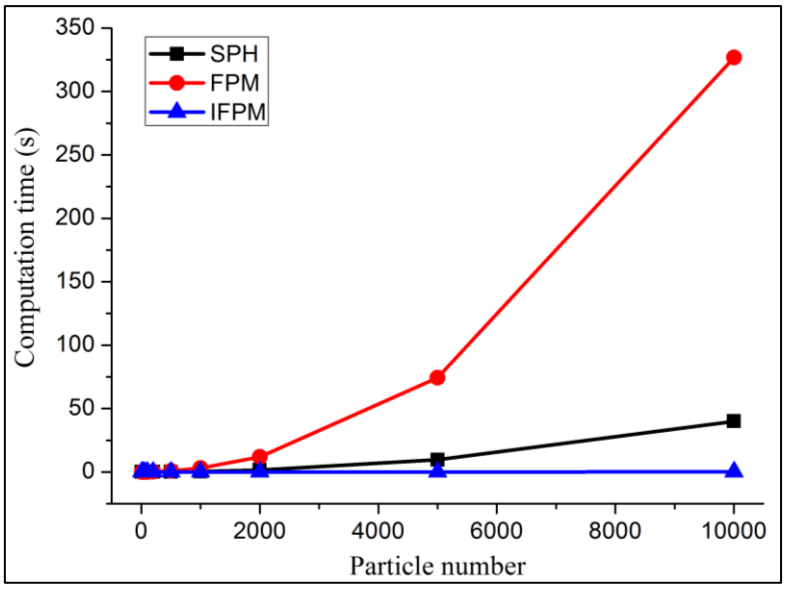

Figure 3. Comparison on the computation time among the three methods

3.3 Calculation of high-order functions

Table 4 shows the *MSE* among the SPH, FPM and IFPM for the function $f(x)=1$, $f(x)=x$ and $f(x)=x^2$ in the interval [0, 100] with the number of particles 11 and 10001. It can be found that the accuracy of IFPM clearly decreases when the function order is bigger than 1.

Table 4. *MSE* of reproduced results by using different methods

Particle number	Function	SPH	FPM	IFPM
11	$f(x)=1$	0.0089	2.2411×10^{-33}	0
	$f(x)=x$	54.6845	9.7532×10^{-29}	9.9513×10^{-33}
	$f(x)=x^2$	6.3848×10^{5}	1.8257×10^{3}	1.5455×10^{4}
10001	$f(x)=1$	1.3038×10^{-5}	4.9299×10^{-36}	0
	$f(x)=x$	0.0598	1.3449×10^{-28}	3.7050×10^{-30}
	$f(x)=x^2$	554.8744	2.2677×10^{-9}	1.0068×10^{-8}

However, it is also shown in Table 4 that the accuracy of IFPM could be improved by increasing the total number of particles in the computational domain. Figure 4 shows the *MSE* of reproduced results for $f(x)=x^2$ with increasing of the number of particles by using IFPM, where the data in Figure 4 is the denary logarithm of original results. It is shown that the *MSE* will keep linear decrease with the increased particle number, which means the accuracy of IFPM is controllable.

4. Conclusions

Based on the numerical tests above, it can be found that the proposed IFPM

method could not only keep the high accuracy of FPM in both the interior area and the boundary area for the constant function and the linear function, but also modify the deficiencies of the long computation time and computational instability in traditional FPM. For the high-order functions, the accuracy of IFPM could be improved by increasing the number of particles. Therefore, IFPM is an effective improvement for traditional FPM.

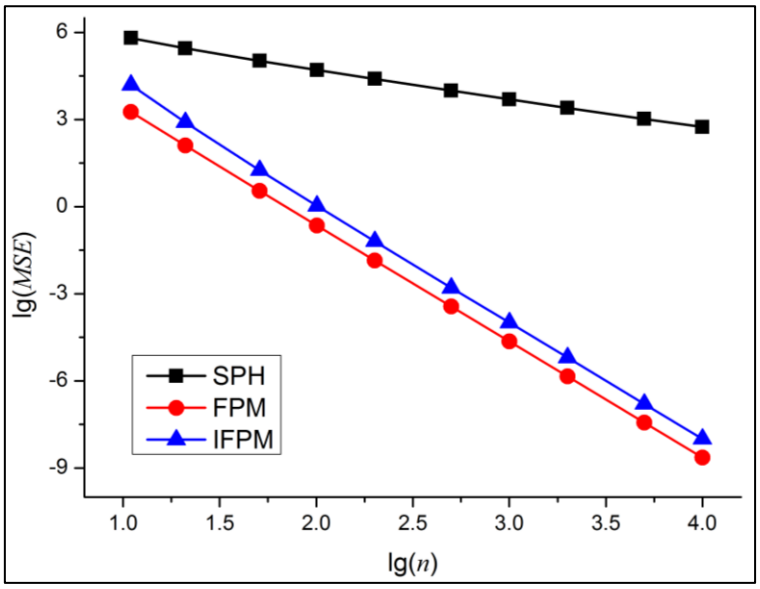

Figure 4. *MSE* of reproduced results with the number of particles

Acknowledgements

This paper is supported by the National Natural Science Foundation of China (Grant No. 11272266) and the graduate starting seed fund of Northwestern Polytechnical University(Grant No. Z2014083).

References

Lucy L B. (1977) Numerical approach to testing the fission hypothesis, *The Astronomical Journal* **82**, 1013-1024.

Gingold R A, Monaghan J J. (1977) Smoothed particle hydrodynamics: Theory and application to non-spherical stars, *Monthly Notices of the Royal Astronomical Society* **181**, 375–389.

Liu, G. R. and Liu, M. B. (2003) *Smoothed Particle Hydrodynamics — A Meshfree Particle Method*, World Scientific, Singapore.

Yang X F, Peng S L, Liu M B. (2014) A new kernel function for SPH with applications to free surface flows, *Applied Mathematical Modelling* **38**, 3822-3833.

Feng D L, Liu M B, Li H Q, et al. (2013) Smoothed particle hydrodynamics modeling of linear shaped charge with jet formation and penetration effects, *Computers & Fluids* **86**, 77-85.

Liu M B, Zong Z, Chang J Z. (2011) Developments and applications of Smoothed Particle Hydrodynamics, *Advances in Mechanic* **41**, 219-236. (in Chinese)

Libersky L D, Petschek A G, Carney T C, et al. (1993) High strain Lagrangian hydrodynamics - a three dimensional SPH code for dynamic material response, *Journal of Computational Physics* **109**,

67-75.

Liu M B, Xie W P, Liu G R. (2005) Modeling incompressible flows using a finite particle method, *Applied Mathematical Modelling* **29**, 1252-1270.

Zhang G M, Batra R C. (2004) Modified smoothed particle hydrodynamics method and its application to transient problems, *Computational Mechanics* **34**, 137-146.

Monaghan J J. (1992) Smoothed particle hydrodynamics, *Annual Review of Astronomy and Astrophysics* **30**, 543-574.