 CMES, vol.79, no.4, pp.223-236, 2011

# A Numerical Technique Based on Integrated RBFs for the System Evolution in Molecular Dynamics

**N. Mai-Duy[1], T. Tran-Cong[1] and N. Phan-Thien[2]**

**Abstract:** This paper presents a new numerical technique for solving the evolution equations in molecular dynamics (MD). The variation of the MD system is represented by radial-basis-function (RBF) equations which are constructed using integrated multiquadric basis functions and point collocation. The proposed technique requires the evaluation of forces once per time step. Several examples are given to demonstrate the attractiveness of the present implementation.

**Keywords:** molecular dynamics, time evolution equation, numerical solver, radial basis functions.

## 1 Introduction

Based on the criterion of timescale, fluid simulations can be categorised into three regions: atomistic (of the order of nanoseconds), mesoscopic (microseconds) and macroscopic (seconds). On a macroscopic scale, a fluid can be modelled as a continuum and its motion is adequately described by the Navier-Stokes equations which can be solved by traditional discretisation techniques such as finite-difference and finite-volume methods (e.g. [Roache (1980)]). On a microscopic scale, one can employ molecular dynamics (MD) simulation methods, which allow all of the actual atoms to be represented explicitly, thus to have the ability to provide very detailed information about the behaviour of complex fluid systems (e.g. [Haile (1992); Leach (2001); Rapaport (2004)]). On a mesoscopic scale, simulations can be conducted using dissipative particle dynamics (DPD), where each particle is regarded as a group of molecules (e.g. [Espanol and Warren (1995)]).

In MD and DPD, a large number of particles are usually required. The time evolution of particles is described by the Newton's equations of motion which can

[1] Computational Engineering and Science Research Centre, Faculty of Engineering and Surveying, University of Southern Queensland, Toowoomba, QLD 4350, Australia.

[2] Department of Mechanical Engineering, Faculty of Engineering, National University of Singapore, Singapore.

be solved numerically. The essential idea of numerical methods is that the time domain is represented by a set of time steps over which the evolution equations are integrated. Considerable effort has been made to derive effective and efficient numerical schemes for the evolution of the particle system.

Radial basis function networks (RBFNs) have emerged as a powerful numerical tool for the solution of differential equations (e.g. [Fasshauer (2007)]). These approximators are able to work well (i.e. providing fast convergence) with gridded and scattered data points. The RBF approximations can be constructed through differentiation (DRBFNs) (e.g. [Kansa (1990)]) or integration (IRBFNs) (e.g. [Mai-Duy and Tran-Cong (2001); Mai-Duy and Tran-Cong (2003); Mai-Duy (2005)]). The latter has the ability to avoid the reduction in convergence rate caused by differentiation and to provide an effective way of implementing derivative boundary values.

This study is concerned with the application of IRBFNs for solving the evolution equations in MD. The paper is organised as follows. A brief review of traditional numerical solvers is given in Section 2. The proposed IRBF method is described in Section 3 and then verified numerically in Section 4. Section 5 concludes the paper.

## 2 Traditional numerical solvers

The equations of motion of particles can be simply written in the form

$$\ddot{\phi}(t) = F(\phi(t)) \tag{1}$$

where $\phi$ is the field variable representing a component of the position vector; $t$ the time; F a known function; and dots represent derivatives with respect to time. In solving (1), the evaluation of $F$ (force) is known to dominate the computational time. Any numerical solver for the MD equations should be designed to keep the number of force evaluations as small as possible. Verlet-based and predictor-corrector integration methods, which require the evaluation of forces once per time step, have been widely used in MD.

### *2.1 Verlet-based methods*

The original Verlet algorithm [Verlet (1967)] can be described as

$$\phi(t+h) = 2\phi(t) - \phi(t-h) + h^2 F(\phi(t)) \tag{2}$$

where $h$ denotes the size of the time step used for the numerical integration.

Implementation of (2) is straightforward. However, its RHS is the sum of a term of order $h^2$ and terms of order $h^0$. As a result, only a few significant figures of

$F(\phi(t))$ are actually utilised, which may lead to a loss of precision. There is also the lack of an explicit velocity term in the equation and one thus has to face difficulties in certain applications. The velocity can be evaluated using the following approximation

$$\dot{\phi}(t) = \frac{\phi(t+h) - \phi(t-h)}{2h} \tag{3}$$

which is only carried out after the position at the next time step is obtained. The following modified Verlet algorithms eliminate these types of problems.

**Leap-frog algorithm:** [Hockney (1970)]

$$\dot{\phi}(t+h/2) = \dot{\phi}(t-h/2) + hF(\phi(t)) \tag{4}$$
$$\phi(t+h) = \phi(t) + h\dot{\phi}(t+h/2) \tag{5}$$

**Velocity Verlet algorithm:** [Swope, Andersen, Berens, and Wilson (1982)]

$$\phi(t+h) = \phi(t) + h\dot{\phi}(t) + (h^2/2)F(\phi(t)) \tag{6}$$
$$\dot{\phi}(t+h) = \dot{\phi}(t) + (h/2)\left[F(\phi(t)) + F(\phi(t+h))\right] \tag{7}$$

### *2.2 Predictor-corrector integration methods*

The predictor-corrector methods typically involve three basic steps [Gear (1971)]. First, new positions, velocities, etc., are predicted using the values of the past

$$\phi(t+h) = \phi(t) + h\dot{\phi}(t) + h^2 \sum_{i=1}^{k-1} \alpha_i F(\phi(t+[1-i]h)) \tag{8}$$

$$h\dot{\phi}(t+h) = \phi(t+h) - \phi(t) + h^2 \sum_{i=1}^{k-1} \alpha_i' F(\phi(t+[1-i]h)) \tag{9}$$

Second, the forces are evaluated at the new positions to give accelerations $F(\phi(t+h))$. Third, by taking into account the difference between the predicted and computed accelerations, the positions, velocities, etc., are corrected through

$$\phi(t+h) = \phi(t) + h\dot{\phi}(t) + h^2 \sum_{i=1}^{k-1} \beta_i F(\phi(t+[2-i]h)) \tag{10}$$

$$h\dot{\phi}(t+h) = \phi(t+h) - \phi(t) + h^2 \sum_{i=1}^{k-1} \beta_i' F(\phi(t+[2-i]h)) \tag{11}$$

In (8)-(11), $\alpha_i$, $\alpha_i^{'}$, $\beta_i$ and $\beta_i^{'}$ are the coefficients that satisfy

$$\sum_{i=1}^{k-1}(1-i)^q\alpha_i = \frac{1}{(q+1)(q+2)}, \quad q=0,1,\cdots,k-2 \tag{12}$$

$$\sum_{i=1}^{k-1}(1-i)^q\alpha_i^{'} = \frac{1}{q+2} \tag{13}$$

$$\sum_{i=1}^{k-1}(2-i)^q\beta_i = \frac{1}{(q+1)(q+2)} \tag{14}$$

$$\sum_{i=1}^{k-1}(2-i)^q\beta_i^{'} = \frac{1}{q+2} \tag{15}$$

In the case of $k=4$, $\{\alpha_i\}_{i=1}^3 = 1/24\times\{19,-10,3\}$, $\{\alpha_i^{'}\}_{i=1}^3 = 1/24\times\{27,-22,7\}$, $\{\beta_i\}_{i=1}^3 = 1/24\times\{3,10,-1\}$ and $\{\beta_i^{'}\}_{i=1}^3 = 1/24\times\{7,6,-1\}$. It is noted that the force function $F$ is evaluated using the results of the predictor step rather than the corrected values. Variations of this method include applying the corrector more than once.

## 3 Proposed IRBF technique

In this section, a numerical procedure based on IRBFNs and point collocation for the solution of (1) on a set of points is presented. Integration constants arising from the RBF construction process allow more preceding information to be incorporated into the discrete system for the solution at the next time level.

### *3.1 Constructing the approximations*

Second-order derivative of $\phi$ is decomposed into RBFs

$$\ddot{\phi}(t) = \sum_{i=1}^{n} w_i g_i(t) \tag{16}$$

where $\{w_i\}_{i=1}^n$ is the set of network weights and $\{g_i(t)\}_{i=1}^n$ the set of RBFs. Expressions for the first-order derivative and function itself are then obtained through integration

$$\dot{\phi}(t) = \sum_{i=1}^{n} w_i H_i(t) + c_1 \tag{17}$$

$$\phi(t) = \sum_{i=1}^{n} w_i \bar{H}_i(t) + c_1 t + c_2 \tag{18}$$

where $H_i(t) = \int g_i(t)dt$, $\bar{H}_i(t) = \int H_i(t)dt$ and $(c_1, c_2)$ are the constants of integration.

A distinguishing feature here is that the present coefficient vector, $(w_1, \cdots, w_n, c_1, c_2)^T$, is larger than that, $(w_1, \cdots, w_n)^T$, of the conventional/differential approach.

### *3.2 Discretising the equations of motion*

We implement two IRBFN versions, namely 2-point and 4-point schemes, which correspond to, in terms of nodes, Verlet-based and predictor-corrector ($k = 4$) methods, respectively. It can be seen that the values of $\phi$, $\dot{\phi}$ and $\ddot{\phi}$ at $t$, $t-h$, $t-2h$, etc., are given. Our objective here is to find the values of $\phi$, $\dot{\phi}$ and $\ddot{\phi}$ at $t+h$.

#### *3.2.1 Two-point IRBFN scheme*

The IRBFN approximations are constructed over two points ($n = 2$), $t$ and $t+h$. Owing to the presence of the integration constants, one can incorporate not only $\phi(t)$ and $\ddot{\phi}(t)$ but also $\dot{\phi}(t)$ into the IRBFN system in an exact manner

$$\begin{pmatrix} \phi(t) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{pmatrix} = \mathscr{A}_1 \begin{pmatrix} w_1 \\ w_2 \\ c_1 \\ c_2 \end{pmatrix} \tag{19}$$

where $\mathscr{A}_1$ is defined as

$$\mathscr{A}_1 = \begin{bmatrix} \bar{H}_1(t), & \bar{H}_2(t), & t, & 1 \\ g_1(t), & g_2(t), & 0, & 0 \\ H_1(t), & H_2(t), & 1, & 0 \end{bmatrix} \tag{20}$$

Solving (19) gives

$$\begin{pmatrix} w_1 \\ w_2 \\ c_1 \\ c_2 \end{pmatrix} = \mathscr{A}_1^{-1} \begin{pmatrix} \phi(t) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{pmatrix} \tag{21}$$

The new position is then obtained

$$\phi(t+h) = [\bar{H}_1(t+h), \bar{H}_2(t+h), t+h, 1]\mathscr{A}_1^{-1} \begin{pmatrix} \phi(t) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{pmatrix} \tag{22}$$

from which one can calculate the new acceleration $\ddot{\phi}(t+h)$.

To compute $\dot{\phi}(t+h)$, an IRBFN system is constructed as follows

$$\begin{pmatrix} \phi(t+h) \\ \ddot{\phi}(t+h) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{pmatrix} = \mathscr{A}_2 \begin{pmatrix} w_1 \\ w_2 \\ c_1 \\ c_2 \end{pmatrix} \tag{23}$$

where $\mathscr{A}_2$ is defined as

$$\mathscr{A}_2 = \begin{bmatrix} \bar{H}_1(t+h), & \bar{H}_2(t+h), & t+h, & 1 \\ g_1(t+h), & g_2(t+h), & 0, & 0 \\ g_1(t), & g_2(t), & 0, & 0 \\ H_1(t), & H_2(t), & 1, & 0 \end{bmatrix} \tag{24}$$

In (23), there are four values included instead of the usual two. These values are imposed exactly owing to the presence of $c_1$ and $c_2$.

The value of the new velocity is then determined through

$$\dot{\phi}(t+h) = [H_1(t+h), H_2(t+h), 1, 0]\mathscr{A}_2^{-1} \begin{pmatrix} \phi(t+h) \\ \ddot{\phi}(t+h) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{pmatrix} \tag{25}$$

### *3.2.2 Four-point IRBFN scheme*

The IRBFN approximations are constructed over four points ($n = 4$), $\{t-2h, t-h, t, t+h\}$. The solution procedure is similar to that of the 2-point IRBFN scheme. The IRBFN approximations for the position and velocity are of the forms

$$\begin{pmatrix} \phi(t-2h) \\ \ddot{\phi}(t-2h) \\ \ddot{\phi}(t-h) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{pmatrix} = \mathscr{A}_1 \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ c_1 \\ c_2 \end{pmatrix} \tag{26}$$

$$\begin{pmatrix} \phi(t+h) \\ \ddot{\phi}(t+h) \\ \ddot{\phi}(t) \\ \ddot{\phi}(t-h) \\ \ddot{\phi}(t-2h) \\ \dot{\phi}(t) \end{pmatrix} = \mathscr{A}_2 \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ c_1 \\ c_2 \end{pmatrix} \tag{27}$$

where

$$\mathscr{A}_1 = \begin{bmatrix} \bar{H}_1(t-2h), & \bar{H}_2(t-2h), & \bar{H}_3(t-2h), & \bar{H}_4(t-2h), & t-2h, & 1 \\ g_1(t-2h), & g_2(t-2h), & g_3(t-2h), & g_4(t-2h), & 0, & 0 \\ g_1(t-h), & g_2(t-h), & g_3(t-h), & g_4(t-h), & 0, & 0 \\ g_1(t), & g_2(t), & g_3(t), & g_4(t), & 0, & 0 \\ H_1(t), & H_2(t), & H_3(t), & H_4(t), & 1, & 0 \end{bmatrix} \tag{28}$$

and

$$\mathscr{A}_2 = \begin{bmatrix} \bar{H}_1(t+h), & \bar{H}_2(t+h),, & \bar{H}_3(t+h), & \bar{H}_4(t+h) & t+h, & 1 \\ g_1(t+h), & g_2(t+h), & g_3(t+h), & g_4(t+h), & 0, & 0 \\ g_1(t), & g_2(t), & g_3(t), & g_4(t), & 0, & 0 \\ g_1(t-h), & g_2(t-h), & g_3(t-h), & g_4(t-h), & 0, & 0 \\ g_1(t-2h), & g_2(t-2h), & g_3(t-2h), & g_4(t-2h), & 0, & 0 \\ H_1(t), & H_2(t), & H_3(t), & H_4(t), & 1, & 0 \end{bmatrix} \tag{29}$$

The IRBFN formulas for computing the new position and velocity thus become

$$\phi(t+h) = [\bar{H}_1(t+h), \bar{H}_2(t+h), \bar{H}_3(t+h), \bar{H}_4(t+h), t+h, 1]\mathscr{A}_1^{-1} \begin{pmatrix} \phi(t-2h) \\ \ddot{\phi}(t-2h) \\ \ddot{\phi}(t-h) \\ \ddot{\phi}(t) \\ \dot{\phi}(t) \end{pmatrix} \tag{30}$$

$$\dot{\phi}(t+h) = [H_1(t+h), H_2(t+h), H_3(t+h), H_4(t+h), 1, 0]\mathscr{A}_2^{-1} \begin{pmatrix} \phi(t+h) \\ \ddot{\phi}(t+h) \\ \ddot{\phi}(t) \\ \ddot{\phi}(t-h) \\ \ddot{\phi}(t-2h) \\ \dot{\phi}(t) \end{pmatrix} \tag{31}$$

One common feature in the two present IRBFN schemes is that information about the governing equation (1) is used as much as possible. For example, in (23) and (27), (1) is collocated at every interpolation point.

With (18), (17) and (16), one is also able to observe the variation of $\phi$ and its derivatives between $t$ and $t+h$. In the proposed technique, the approximations are of high order and one is able to incorporate more preceding information into the discrete system in a proper manner. Unlike the predictor-corrector method, the present value of $\phi$ at $t+h$ is unique. Since the values of $\phi$, $\dot{\phi}$ and $\ddot{\phi}$ at $t+h$

are imposed at the same time ((23) and (27)), their relations, i.e. $\dot{\phi} = d\phi/dt$ and $\ddot{\phi} = d^2\phi/dt^2$, are satisfied exactly.

It can be seen that the process of constructing the IRBFN approximations needs be done only once. The resultant IRBF approximations can be used for every time step and for every particle in the system. At each time step, the present technique requires one force evaluation and the multiplication of matrices and vectors whose sizes are quite small.

## 4 Numerical Results

The present IRBFN schemes are implemented with the multiquadric (MQ) function whose form is

$$g_i(t) = \sqrt{(t-t_i)^2 + a_i^2} \tag{32}$$

where $t_i$ and $a_i$ are the centre and width of the $i$th MQ-RBF, respectively. The latter is simply chosen according to the relation

$$a_i = \beta h \tag{33}$$

where $\beta$ is a positive number. The RBF width can be used to influence the accuracy of a RBF solution. The best accuracy is often achieved when the value of $a_i$ is large. On the other hand, the matrix condition number grows as the number of RBF centres and the RBF width increase. This inter-dependence is known as the uncertainty or trade-off principle. Fortunately, the number of MQ centres in the present schemes are relatively low (only two and four) which allow larger values of $\beta$ to be used here.

### *4.1 Example 1*

Consider the following ODE of motion for a one-dimensional harmonic oscillator

$$\ddot{\phi}(t) = -(2\pi)^2 \phi(t) \tag{34}$$

This problem was suggested as a simple exact test for molecular dynamics algorithms [Venneri and Hoover (1987)]. The analytic solution can be verified to be $\phi_e(t) = \cos(2\pi t)$.

Calculations are carried out over $0 \leq t \leq 1$ with various time steps $\{1/10, 1/14, 1/18, 1/22, \cdots, 1/102\}$.

Tables 1 and 2 present the relative $L_2$ error, denoted by $N_e(\phi)$, of the computed solution obtained by different techniques.

Table 1: 1D Harmonic oscillator, two interpolation points: discrete relative $L_2$ error

| | $N_e(\phi)$ | | | |
|---|---|---|---|---|
| h | Verlet | Leap-frog | vVerlet | Present |
| 1.0000e-1 | 5.1834e-1 | 5.5036e-2 | 5.5036e-2 | 5.4507e-2 |
| 7.1429e-2 | 3.9505e-1 | 2.8318e-2 | 2.8318e-2 | 2.7777e-2 |
| 5.5556e-2 | 3.1659e-1 | 1.7274e-2 | 1.7274e-2 | 1.6725e-2 |
| 4.5455e-2 | 2.6374e-1 | 1.1638e-2 | 1.1638e-2 | 1.1085e-2 |
| 3.8462e-2 | 2.2590e-1 | 8.3738e-3 | 8.3738e-3 | 7.8166e-3 |
| 3.3333e-2 | 1.9753e-1 | 6.3139e-3 | 6.3139e-3 | 5.7541e-3 |
| 2.9412e-2 | 1.7548e-1 | 4.9307e-3 | 4.9307e-3 | 4.3690e-3 |
| 2.6316e-2 | 1.5785e-1 | 3.9572e-3 | 3.9572e-3 | 3.3938e-3 |
| 2.3810e-2 | 1.4344e-1 | 3.2461e-3 | 3.2461e-3 | 2.6814e-3 |
| 2.1739e-2 | 1.3143e-1 | 2.7108e-3 | 2.7108e-3 | 2.1451e-3 |
| 2.0000e-2 | 1.2129e-1 | 2.2979e-3 | 2.2979e-3 | 1.7312e-3 |
| 1.8519e-2 | 1.1259e-1 | 1.9726e-3 | 1.9726e-3 | 1.4051e-3 |
| 1.7241e-2 | 1.0506e-1 | 1.7118e-3 | 1.7118e-3 | 1.1437e-3 |
| 1.6129e-2 | 9.8473e-2 | 1.4995e-3 | 1.4995e-3 | 9.3081e-4 |
| 1.5152e-2 | 9.2663e-2 | 1.3244e-3 | 1.3244e-3 | 7.5518e-4 |
| 1.4286e-2 | 8.7501e-2 | 1.1783e-3 | 1.1783e-3 | 6.0860e-4 |
| 1.3514e-2 | 8.2883e-2 | 1.0551e-3 | 1.0551e-3 | 4.8498e-4 |
| 1.2821e-2 | 7.8729e-2 | 9.5026e-4 | 9.5026e-4 | 3.7976e-4 |
| 1.2195e-2 | 7.4971e-2 | 8.6030e-4 | 8.6030e-4 | 2.8946e-4 |
| 1.1628e-2 | 7.1555e-2 | 7.8254e-4 | 7.8254e-4 | 2.1139e-4 |
| 1.1111e-2 | 6.8437e-2 | 7.1487e-4 | 7.1487e-4 | 1.4344e-4 |
| 1.0638e-2 | 6.5580e-2 | 6.5561e-4 | 6.5561e-4 | 8.3918e-5 |
| 1.0204e-2 | 6.2951e-2 | 6.0343e-4 | 6.0343e-4 | 3.1496e-5 |
| 9.8039e-3 | 6.0526e-2 | 5.5724e-4 | 5.5724e-4 | 1.4917e-5 |
| | $O(h^{0.94})$ | $O(h^{1.98})$ | $O(h^{1.98})$ | $O(h^{3.01})$ |

Table 2: 1D Harmonic oscillator, four interpolation points: discrete relative $L_2$ error

| | $N_e(\phi)$ | |
|---|---|---|
| h | Predictor-corrector | Present |
| 1.0000e-1 | 2.7464e-2 | 1.0063e-2 |
| 7.1429e-2 | 9.3827e-3 | 2.2737e-3 |
| 5.5556e-2 | 4.1403e-3 | 7.8787e-4 |
| 4.5455e-2 | 2.1678e-3 | 3.4399e-4 |
| 3.8462e-2 | 1.2735e-3 | 1.7581e-4 |
| 3.3333e-2 | 8.1145e-4 | 1.0037e-4 |
| 2.9412e-2 | 5.4898e-4 | 6.2104e-5 |
| 2.6316e-2 | 3.8886e-4 | 4.0966e-5 |
| 2.3810e-2 | 2.8561e-4 | 2.8459e-5 |
| 2.1739e-2 | 2.1601e-4 | 2.0629e-5 |
| 2.0000e-2 | 1.6737e-4 | 1.5520e-5 |
| 1.8519e-2 | 1.3234e-4 | 1.1952e-5 |
| 1.7241e-2 | 1.0647e-4 | 9.4772e-6 |
| 1.6129e-2 | 8.6944e-5 | 7.6480e-6 |
| 1.5152e-2 | 7.1924e-5 | 6.4127e-6 |
| 1.4286e-2 | 6.0182e-5 | 5.4785e-6 |
| 1.3514e-2 | 5.0867e-5 | 4.5454e-6 |
| 1.2821e-2 | 4.3383e-5 | 4.1820e-6 |
| 1.2195e-2 | 3.7301e-5 | 3.3929e-6 |
| 1.1628e-2 | 3.2306e-5 | 3.1211e-6 |
| 1.1111e-2 | 2.8166e-5 | 2.8360e-6 |
| 1.0638e-2 | 2.4705e-5 | 2.4722e-6 |
| 1.0204e-2 | 2.1790e-5 | 2.4024e-6 |
| 9.8039e-3 | 1.9316e-5 | 2.2701e-6 |
| | $O(h^{3.11})$ | $O(h^{3.58})$ |

Results by the 2-point IRBFN scheme ($\beta = 16$) are compared with those predicted by the Verlet, leap-frog and velocity Verlet algorithms (Table 1).

Results by the 4-point IRBFN scheme ($\beta = 31$) are compared with those obtained by the predictor-corrector integration method with $k = 4$ (Table 2). Effect of $\beta$ on the solution accuracy is investigated in Figure 1, showing that the IRBFN results are not much different for various large values of $\beta$.

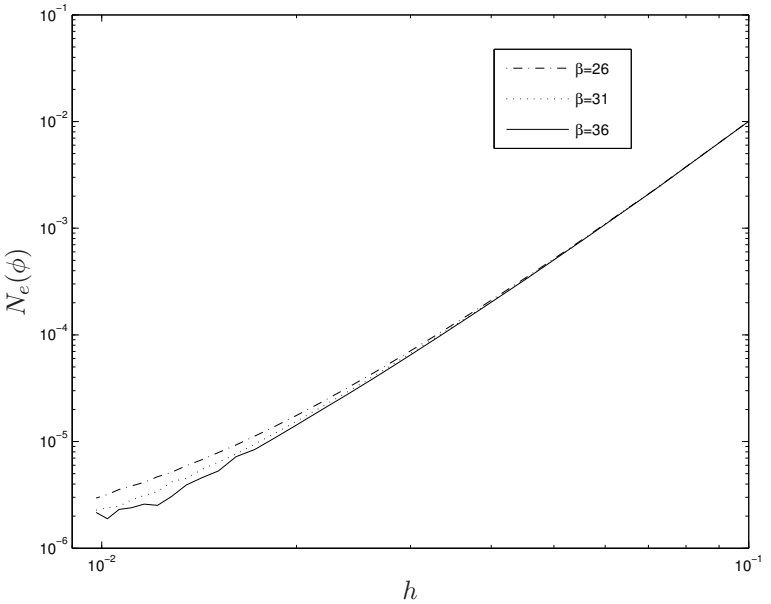


Figure 1: 1D Harmonic oscillator, four interpolation points: Effect of $\beta$ on the solution accuracy.

It can be seen that the present schemes gives superior accuracy and faster convergence than the traditional methods for the same number of interpolation points used.

### *4.2 Example 2*

Consider a three-dimensional system of MD particles which is allowed to evolve to equilibrium. We adopt the Weeks-Chandler-Anderson (WCA) potential, i.e. a modification of the Lennard-Jones potential,

$$u(r_{ij}) = 4\varepsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^{6} \right] + \varepsilon, \quad r_{ij} < r_c = 2^{1/6}\sigma \tag{35}$$

where $r_{ij} = \|\mathbf{r}_{ij}\|$, in which $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and $\mathbf{r}_i$ and $\mathbf{r}_j$ are the position vectors of particles $i$ and $j$, respectively, $\varepsilon$ is a parameter characterising the strength of the interaction, $\sigma$ the molecular length scale and $r_c$ the cut-off distance, i.e. $u(r_{ij}) = 0$ when $r_{ij} \geq r_c$. Parameters used here are the same as those in Problem 1 (Chapter 3) [Rapaport (2004)]. We employ an array of $5 \times 5 \times 5$ unit cells and the density of 0.8. The initial state is a simple cubic lattice so that the system involves 125 particles. Simulations are carried out with several time steps. For comparison purposes, the leap-frog and predictor-corrector ($k = 4$) methods are also employed.

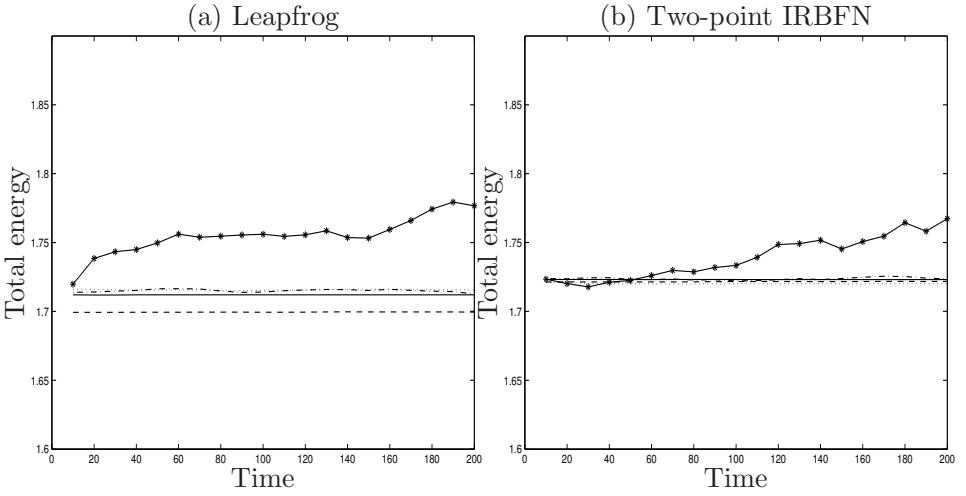


Figure 2: Energy drift for different time steps by the leap-frog (top) and two-point IRBFN (bottom) methods. It is noted that solid line represents the case of $h = 0.00125$, dashed $h = 0.0025$, dotted $h = 0.005$, dashdot $h = 0.01$ and * $h = 0.02$. Both plots have the same coordinate scaling.

Results concerning energy conservation are presented in Figures 2 and 3. A total run involves 200 time units and results are stored at every ten time units. In the case of two-point discretisations (Figure 2a), the two-point IRBFN scheme ($\beta = 16$) and the leap-frog method both perform well. The total energy is well conserved with time for a wide range of time step from 0.00125 to 0.01 and slightly increases for a large time step of 0.02. In the case of four-point discretisations (Figure 2b), for a given degree of energy conservation, large time steps can be used with the proposed scheme. For the predictor-corrector method, there are noticeable increases in the total energy with time for time steps of 0.0025 and 0.005. In contrast, the 4-point IRBFN scheme ($\beta = 31$) works well for all time steps of 0.00125, 0.0025, 0.005

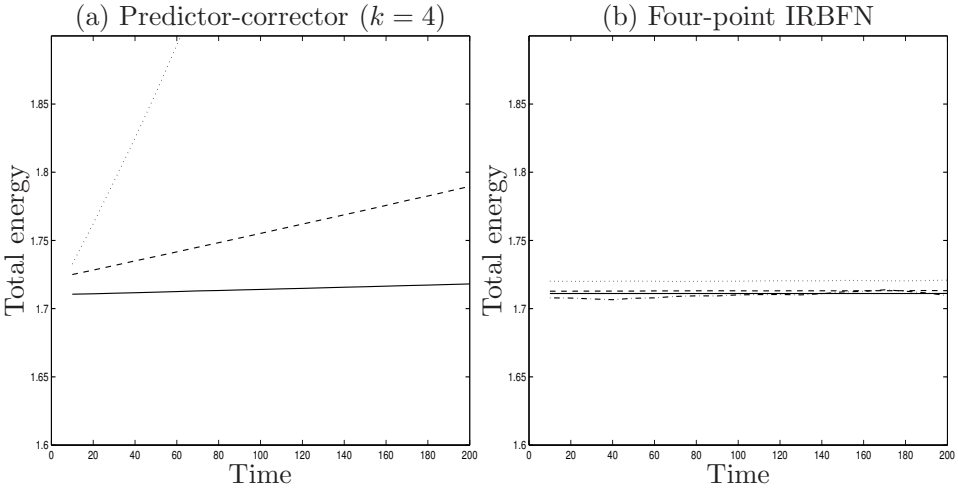


Figure 3: Energy drift for different time steps by the predictor-corrector ($k = 4$) (top) and four-point IRBFN (bottom) methods. It is noted that solid line represents the case of $h = 0.00125$, dashed $h = 0.0025$, dotted $h = 0.005$ and dashdot $h = 0.01$. Both plots have the same coordinate scaling.

and 0.01.

## 5 Concluding remarks

In this paper, a new numerical technique for solving the evolution equations in molecular dynamics is presented. The proposed technique is based on high-order integrated RBFs and point collocation. For a given number of interpolation points, the proposed technique has the ability to incorporate more preceding information owing to the presence of the constants of integration. Numerical examples show that the proposed technique is more accurate and stabler than traditional techniques.

**Acknowledgement:** This work is supported by the Australian Research Council.

## References

**Espanol, P.; Warren, P.** (1995): Statistical mechanics of dissipative particle dynamics. *Europhysics Letters*, vol. 30, no. 4, pp. 191–196.

**Fasshauer, G. E.** (2007): *Meshfree Approximation Methods with Matlab*. World Scientific Publishing.

**Gear, C. W.** (1971): *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall.

**Haile, J. M.** (1992): *Molecular Dynamics Simulation: Elementary Methods*. John Wiley & Sons.

**Hockney, R. W.** (1970): The potential calculation and some applications. *Methods in Computational Physics*, vol. 9, pp. 135–211.

**Kansa, E. J.** (1990): Multiquadrics- A scattered data approximation scheme with applications to computational fluid-dynamics-II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers and Mathematics with Applications*, vol. 19, no. 8/9, pp. 147–161.

**Leach, A. R.** (2001): *Molecular Modelling Principles and Applications*. Prentice Hall.

**Mai-Duy, N.** (2005): Solving high order ordinary differential equations with radial basis function networks. *International Journal for Numerical Methods in Engineering*, vol. 62, pp. 824–852.

**Mai-Duy, N.; Tran-Cong, T.** (2001): Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Networks*, vol. 14, no. 2, pp. 185–199.

**Mai-Duy, N.; Tran-Cong, T.** (2003): Approximation of function and its derivatives using radial basis function networks. *Applied Mathematical Modelling*, vol. 27, pp. 197–220.

**Rapaport, D. C.** (2004): *Art of Molecular Dynamics Simulation*. Cambridge University Press.

**Roache, P. J.** (1980): *Computational Fluid Dynamics*. Hermosa Publishers.

**Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R.** (1982): A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *Journal Chemical Physics*, vol. 76, no. 1, pp. 637–649.

**Venneri, G. D.; Hoover, W. G.** (1987): Simple exact test for well-known molecular dynamics algorithms. *Journal of Computational Physics*, vol. 73, no. 2, pp. 468–475.

**Verlet, L.** (1967): Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review*, vol. 159, no. 1, pp. 98–103.