INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1909–1928
Published online 1 August 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1571

Collocated discrete least-squares (CDLS) meshless method: Error estimate and adaptive refinement

M. H. Afshar*, †, ‡ and M. Lashckarbolok§

Civil Engineering Department, Iran University of Science and Technology, Tehran, Islamic Republic of Iran

SUMMARY

Meshless methods are new approaches for solving partial differential equations. The main characteristic of all these methods is that they do not require the traditional mesh to construct a numerical formulation. They require node generation instead of mesh generation. In other words, there is no pre-specified connectivity or relationships among the nodes. This characteristic make these methods powerful. For example, an adaptive process which requires high computational effort in mesh-dependent methods can be very economically solved with meshless methods. In this paper, *a posteriori* error estimate and adaptive refinement strategy is developed in conjunction with the collocated discrete least-squares (CDLS) meshless method. For this, an error estimate is first developed for a CDLS meshless method. The proposed error estimator is shown to be naturally related to the least-squares functional, providing a suitable posterior measure of the error in the solution. A mesh moving strategy is then used to displace the nodal points such that the errors are evenly distributed in the solution domain. Efficiency and effectiveness of the proposed error estimator and adaptive refinement process are tested against two hyperbolic benchmark problems, one with shocked and the other with low gradient smooth solutions. These experiments show that the proposed adaptive process is capable of producing stable and accurate results for the difficult problems considered. Copyright © 2007 John Wiley & Sons, Ltd.

Received 1 January 2007; Revised 12 June 2007; Accepted 13 June 2007

KEY WORDS: meshless; adaptive refinement; error estimate; hyperbolic problems; collocated discrete least-squares (CDLS) method

1. INTRODUCTION

The use of a mesh is a basic characteristic of most numerical approaches for the solution of partial differential equations. In these approaches, assumptions are made for the local approximation of the primitive variables, which require meshes to support them. In recent years, a considerable

*Correspondence to: M. H. Afshar, Civil Engineering Department, Iran University of Science and Technology, Tehran, Islamic Republic of Iran.
†E-mail: mhafshar@iust.ac.ir
‡Assistant Professor.
§M.Sc. Student.

attempt has been made to the development of so-called mesh-free (meshless) methods. The aim of this type of approach is to get rid of at least the structure of the mesh and to approximate the solution entirely using nodes inside and/or on the boundary of the domain. Many meshless (or particle) methods have been reported in the literature, such as element-free Galerkin method [1], diffuse element method [2], reproducing kernel particle method [3], smooth particle hydrodynamics method [4], particle finite element method (FEM) [5], meshless local Petrov–Galerkin (MLPG) method [6], etc. Afshar and Arzani [7] developed discrete least-squares (DLS) meshless method for the solution of convection-dominated problems. In this method, a fully least-squares approach is used in both function approximation and discretization of the governing differential equations. Their method has the additional advantage of producing symmetric, positive-definite matrices even for nonself-adjoint operators as encountered in fluid flow problems.

In an adaptive analysis, there are two fundamental issues, error estimation and domain refinement. The first requires an error estimate to measure the local and global errors in the approximation, whereby an adaptive procedure determines whether a refinement is required and if true, where to refine a domain. The second is performed based on the error information provided by the first. The effectiveness and efficiency of these two aspects are critical to the performance of an adaptive procedure. To conduct *a posteriori* error estimation, two values of a quantity—a computed value and a reference value—are usually required. The first is the raw value given by direct computations, while the second is derived from the first *via* postprocessing (e.g. smoothing or projection). In FEM, the computed quantities which are usually taken as gradients (or stresses) do not possess inter-element continuity and have a low accuracy along element boundaries. The improved values are, therefore, obtained *via* smoothing procedures. The difference between the raw and improved values constitutes a basis for error estimation in FEM [8]. For detailed descriptions of these approaches, the readers may refer to FEM literature [9, 10].

The use of adaptive refinement to improve the accuracy of numerical solution is very recent in meshless methods. In meshless methods, there is no inter-element discontinuity and the resulting solution field is already very smooth over the entire problem domain. As a result, error estimates based on stress smoothing techniques introduced in the context of FEM cannot be used for error estimation in meshless methods. There is a need, therefore, to develop different error estimates for adaptive analysis in meshless methods [8]. Some error estimators have been developed for meshless methods using different techniques. Orkisz [11] presented an adaptive multigrid meshless finite difference method. Laouar and Villon [12] presented a technique of resolution using the diffuse element method with an adaptive set of nodes. Nodes are generated by a quadtree-type of decomposition of the area and the adjustment is made with the help of *a posteriori* knowledge of the error estimate. Duarte and Oden [13] used the partition of unity concept in a very general manner by constructing it from a moving least-squares shape function. The major advantage of the Duarte and Oden formulation is that it enables the extrinsic basis to vary from node to node, thus facilitating hp-adaptivity.

Liu and Tu [8] developed an adaptive procedure based on a background mesh using moving least-squares (MLS) method. It comprises a cell energy error estimate and a local domain refinement technique. The error estimate differs from conventional pointwise approaches in that it evaluates error based on individual cells instead of points. Gavete *et al.* [14] proposed an error indicator for the element-free Galerkin (EFG) method following the same line used in FEM. The method calculated the gradient values at the Gauss integration point. A moving least-squares approach was then used to construct a continuous approximation of the gradient which was subsequently used to obtain an indication of the '*a posteriori*' error for EFG method.

DOI: 10.1002/fld

In this paper, *a posteriori* error estimate and adaptive refinement strategy is developed for the collocated discrete least-squares (CDLS) meshless method. It is shown that the least-squares functional defined by the weighted sum of the squared residuals at collocation points is a suitable posterior measure of the error in the solution. A mesh moving strategy is then used to displace the nodal points such that the errors are evenly distributed in the solution domain. A projection mechanism is also introduced and used before the mesh moving step as a means to control the maximum and minimum nodal spacing in the refined mesh. This is shown to be very effective for problems with high gradient solutions. Efficiency and effectiveness of the proposed error estimators and adaptive refinement process are tested against two hyperbolic benchmark problems, one with shocked and the other with low gradient smooth solutions. The steady nonlinear Burgers equation and steady shallow-water equation simulating the flow of water over an ogee spillway are used. These experiments show that the proposed adaptive process is capable of producing stable and accurate results for the difficult problems considered.

2. MOVING LEAST-SQUARES (MLS) METHOD

Several techniques have been developed to construct shape functions in the development of the meshless methods. The most widely used of these methods is the MLS approximation by Lancaster and Salkauskas [15], the radial point interpolation method (RPIM) by Liu and Gu [16] and the Kriging interpolation by Gu [17]. Among these methods, the method of MLS has been widely used for function approximation by the meshless community. The advantages of MLS are threefold: first, there is no need for explicit meshes in the construction of MLS shape functions. Second, high-order continuity of shape functions so constructed eliminates the necessity of using weak forms of governing equations, as required in FEM, which uses standard shape functions. In addition, higher-order continuity, if required, is not introduced at the expense of increasing the unknown parameters as usually practiced in FEM. Third; the availability of smooth derivatives eliminates the need for costly procedure of gradient recovery which is usually required by standard FEM.

In MLS, the function to be approximated is represented by

$$u^h(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) a_i(\mathbf{x}) \equiv p^{\mathrm{T}}(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{1}$$

Here, $\mathbf{p}^{\mathrm{T}}(\mathbf{x})$ is a set of linearly independent polynomial basis and $\mathbf{a}(\mathbf{x})$ represents the unknown coefficients to be determined by the fitting algorithm. The polynomial bases of order m in one and two dimensions are given by

$$\mathbf{p}^{\mathrm{T}}(x) = [1, x, x^2, \ldots, x^m] \tag{2}$$

$$\mathbf{p}^{\mathrm{T}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(x, y) = [1, x, y, x^2, xy, y^2, \ldots, x^m, \ldots, xy^{m-1}, y^m] \tag{3}$$

In MLS approximation, at each point $\mathbf{x}$, $\mathbf{a}(\mathbf{x})$ is chosen to minimize the sum of weighted squared residuals defined by

$$J = \frac{1}{2} \sum_{I=1}^{n} w(|\mathbf{x} - \mathbf{x}_I|)[\mathbf{p}^{\mathrm{T}}(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - u_I]^2 \tag{4}$$

DOI: 10.1002/fld

In Equation (4), u_I is the nodal value of the function to be approximated, n is the number of nodes and $w(|\mathbf{x}-\mathbf{x}_I|)$ is the weight function defined to have compact support. The weight functions are chosen to have the following properties:

$$(1)\ w(|\mathbf{x}-\mathbf{x}_I|)>0 \quad \text{on a sub domain} \tag{5}$$

$$(2)\ w(|\mathbf{x}-\mathbf{x}_I|)=0 \quad \text{outside the sub-domain} \tag{6}$$

$$(3)\ \int_\Omega w(|\mathbf{x}-\mathbf{x}_I|)\,\mathrm{d}\Omega=1 \quad \text{a normality property} \tag{7}$$

$$(4)\ w(|\mathbf{x}-\mathbf{x}_I|) \quad \text{is a monotonically decreasing function} \tag{8}$$

$$(5)\ w(|\mathbf{x}-\mathbf{x}_I|)\to\delta(s) \text{ as } |\mathbf{x}-\mathbf{x}_I|=h\to 0 \text{ where } \delta(s) \text{ is the Dirac delta function} \tag{9}$$

Many weight functions are established and used by different researchers. In this paper, an exponential weight function is used as follows:

$$w(r)=\begin{cases}\dfrac{1}{(50)^r}, & r\leqslant 1\\ 0, & r>1\end{cases} \tag{10}$$

In which $r=s/s_{\max}$, $s=\|\mathbf{x}-\mathbf{x}_I\|$ and $s_{\max}$ is the radius of the support. Equation (4) can be written in the matrix form as

$$\mathbf{J}=(\mathbf{Pa}-\mathbf{u})^{\mathrm{T}}\mathbf{W}(\mathbf{Pa}-\mathbf{u}) \tag{11}$$

where

$$\mathbf{u}^{\mathrm{T}}=(u_1,u_2,\ldots,u_n) \tag{12}$$

$$\mathbf{P}=\begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \ldots & p_m(\mathbf{x}_1)\\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \ldots & p_m(\mathbf{x}_2)\\ \vdots & \vdots & \vdots & \vdots\\ p_1(\mathbf{x}_n) & p_2(\mathbf{x}_n) & \cdots & p_m(\mathbf{x}_n)\end{bmatrix} \tag{13}$$

and

$$\mathbf{W}(\mathbf{x})=\begin{bmatrix} w(|\mathbf{x}-\mathbf{x}_1|) & 0 & \ldots & 0\\ 0 & w(|\mathbf{x}-\mathbf{x}_2|) & \ldots & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \ldots & w(|\mathbf{x}-\mathbf{x}_n|)\end{bmatrix} \tag{14}$$

The coefficients $\mathbf{a}(\mathbf{x})$ are found by minimizing $\mathbf{J}$ with respect to these coefficients. Carrying out the differentiation

$$\frac{\partial\mathbf{J}}{\partial\mathbf{a}}=\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x})-\mathbf{B}(\mathbf{x})\mathbf{u}=0 \tag{15}$$

DOI: 10.1002/fld

where

$$\mathbf{A} = \mathbf{P}^{\mathrm{T}}\mathbf{W}(\mathbf{x})\mathbf{P} \tag{16}$$

$$\mathbf{B} = \mathbf{P}^{\mathrm{T}}\mathbf{W}(\mathbf{x}) \tag{17}$$

Solving the above equation for the unknown parameters $\mathbf{a}(\mathbf{x})$ results in:

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} \tag{18}$$

The approximation of the unknown function can now be written in the familiar form of:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{I=1}^{n} \mathbf{N}_I(\mathbf{x})\mathbf{u}_I \tag{19}$$

where $\mathbf{N}_I(\mathbf{x})$ denote the shape function of node I defined as

$$\mathbf{N} = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}) \tag{20}$$

In this case, $\mathbf{u}_I \neq \mathbf{u}^h(x_I)$, so the parameters $\mathbf{u}_I$ cannot be treated like nodal values of the unknown function. The shape functions are not strict interpolants since they do not pass through the data. The shape functions do not satisfy the Kronecker delta condition:

$$N_i(\mathbf{x}_j) \neq \delta_{ij} = \begin{Bmatrix} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{Bmatrix} \tag{21}$$

Here, $N_i(\mathbf{x}_j)$ is the shape function of node i evaluated at node j and δ_{ij} is the Kronecker delta. Generally, it is necessary to obtain the shape function derivatives. The spatial derivatives of the shape functions are obtained as

$$\frac{\mathrm{d}\mathbf{N}(\mathbf{x})}{\mathrm{d}x} = \frac{\mathrm{d}\mathbf{P}}{\mathrm{d}x}\mathbf{A}^{-1}\mathbf{B} + \mathbf{P}\frac{\mathrm{d}(\mathbf{A}^{-1})}{\mathrm{d}x}\mathbf{B} + \mathbf{P}\mathbf{A}^{-1}\frac{\mathrm{d}\mathbf{B}}{\mathrm{d}x} \tag{22}$$

3. COLLOCATED DISCRETE LEAST-SQUARES (CDLS) MESHLESS METHOD

Consider the following differential equation:

$$L(\mathbf{u}) = \mathbf{f} \quad \text{in } \Omega \tag{23}$$

$$B(\mathbf{u}) = \mathbf{g} \quad \text{on } \Gamma_{\mathrm{t}} \tag{24}$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_{\mathrm{u}} \tag{25}$$

where L and B are some proper differential operators defined on the problem domain Ω and its Neumann boundary Γ_{t}, respectively. The philosophy of least squares is to find an approximate solution that minimizes the least-squares functional to be defined later. As shown in Figure 1, the problem domain and boundaries are discretized by n number of nodal points. Besides the nodal points, the collocation points are used in the problem domain and on its boundaries to construct the least-squares functional. The total number of collocation points is M comprised of M_{d} internal

DOI: 10.1002/fld

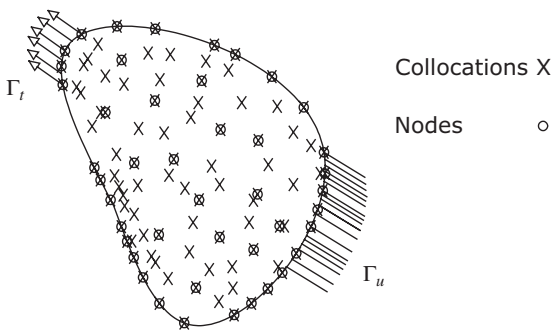

Figure 1. The problem domain discretized by nodal points and collocation points.

collocation points, M_{u} collocation points on the Dirichlet boundary, and M_{t} collocation points on the Neumann boundary, i.e.

$$M = M_{\mathrm{d}} + M_{\mathrm{u}} + M_{\mathrm{t}} \tag{26}$$

The approximate value of the function u at a collocation point x_k can be obtained through interpolation

$$u(x_k) = \sum_{i=1}^{n} N_i(x_k) \cdot u_i \tag{27}$$

where n is the number of nodal points having x_k in their support domain. Substituting Equation (27) into Equations (23)–(25), leads to the differential equation residual R^{d}, the Neumann boundary condition residual R^{t} and the Dirichlet boundary condition residual R^{u} defined as

$$R_k^{(\mathrm{d})} = L(u_k) - f(x_k) = \sum_{j=1}^{n} L(N_j(x_k))u_j - f(x_k) \quad (k = 1\text{–}M) \tag{28}$$

$$R_k^{(\mathrm{t})} = B(u_k) - g(x_k) = \sum_{j=1}^{n} B(N_j(x_k))u_j - g(x_k) \quad (k = 1\text{–}M_{\mathrm{t}}) \tag{29}$$

$$R_k^{(\mathrm{u})} = u_k - \overline{u} = \sum_{j=1}^{n} (N_j(x_k))u_j - \overline{u} \quad (k = 1\text{–}M_{\mathrm{u}}) \tag{30}$$

Now, the following least-squares functional of all residuals at all collocation points can be constructed as

$$J = \frac{1}{2}\left(\sum_{k=1}^{M_{\mathrm{d}}} [R_k^{(\mathrm{d})}]^2 + \alpha \cdot \sum_{k=1}^{M_{\mathrm{t}}} [R_k^{(\mathrm{t})}]^2 + \beta \cdot \sum_{k=1}^{M_{\mathrm{u}}} [R_k^{(\mathrm{u})}]^2 \right) \tag{31}$$

The factors α and β in the above equation are meant to represent the relative weights of the boundary residuals with respect to the interior residual.

Minimization of Equation (31) with respect to the nodal parameters u_i leads to

$$\frac{\partial \mathbf{J}}{\partial \mathbf{u_i}} = \sum_{k=1}^{M_{\mathrm{d}}} \frac{\partial \mathbf{R_k^{(d)}}}{\partial \mathbf{u}_i} [\mathbf{R_k^{(d)}}] + \alpha \sum_{k=1}^{M_{\mathrm{t}}} \frac{\partial \mathbf{R_k^{(t)}}}{\partial \mathbf{u}_i} [\mathbf{R_k^{(t)}}] + \beta \sum_{k=1}^{M_{\mathrm{u}}} \frac{\partial \mathbf{R_k^{(u)}}}{\partial \mathbf{u}_i} [\mathbf{R_k^{(u)}}] = 0 \tag{32}$$

DOI: 10.1002/fld

Substituting Equations (28)–(30) into Equation (32) yields the final system of equations:

$$\mathbf{KU} = \mathbf{F} \tag{33}$$

The typical components of the matrix $\mathbf{K}$ and the right-hand side vector $\mathbf{F}$ are defined as

$$K_{lm} = \sum_{i=1}^{M_d} [L(N_l)]_i^T [L(N_m)]_i + \alpha \sum_{i=1}^{M_t} [B(N_l)]_i^T [B(N_m)]_i + \beta \sum_{i=1}^{M_u} [(N_l)]_i^T [(N_m)]_i$$

$$l, m = 1, \ldots, n \tag{34}$$

$$F_l = -\sum_{i=1}^{M_d} [L(N_l)]_i^T \mathbf{f}_i + \sum_{i=1}^{M_t} [B(N_l)]_i^T \mathbf{g}_i + \sum_{i=1}^{M_u} [(N_l)]_i^T (u - \overline{u}) \quad l = 1, \ldots, n \tag{35}$$

The stiffness matrix $\mathbf{K}$ in Equation (34) can be observed to be symmetric and positive definite. Therefore, the final system of equations can be solved by using efficient iterative procedures such as conjugate gradient methods.

Two points to be noted here regarding the number of collocation points, M, and the values of the penalty parameters α and β. In this paper, a uniformly distributed collocation points is used for all the problems considered. The number of collocation points, M, can be arbitrary as long as it is large enough to make Equation (33) nonsingular, i.e. $M \geqslant n$. Theoretically, the more the collocation points used, the more accurate the solution obtained. However, as M increases, the computational cost of the method will increase. Hence, a reasonable M should be selected. No theoretical basis for choosing the optimum value of M exists at this time. The proper value of M can only be obtained *via* the numerical experimentation [18]. A comprehensive study on the effect of the number of collocation points on the accuracy of the DLSM method for elliptic problems is carried out elsewhere [19].

Penalty parameters α and β used in DLSM method are used for the implementation of the essential and natural boundary conditions. The satisfaction of essential boundary condition is not straightforward since the unknown nodal values in MLS approaching the shape function are not physical values. The value of α, β must be chosen in such a way that the residual on the boundaries with known boundary conditions is more important than the residual in the problem domain. In this paper, $\alpha = 200$ is chosen for all the problems considered.

4. ERROR ESTIMATOR

The importance of a suitable error estimator for adaptive refinement cannot be overemphasized. Since the variational statement for the self-adjoint boundary value problem is normally based on a weighted residual formulation, it is not surprising that the element residual has been used by some researchers [20–22] as the basic error indicator for the Galerkin finite element method. However, in this case the calculation of the element residual may not be straightforward. For example, if the finite element approximation is piecewise linear on the element, the contribution of second-order derivatives to the residual is identically zero everywhere, except on the element boundary where it is undefined [23]. Hence, the inter-element jump discontinuities must be used to calculate these higher derivatives in this case.

DOI: 10.1002/fld

Since the value of residuals represents the extent to which the numerical solution satisfies the governing differential equation and its boundary condition, the least-squares functional defined as the squares residual at an arbitrary point can be considered as a measure of the error of the numerical solution. This concept can be easily implemented for adaptive refinement and applied to linear and nonlinear problems. The method is particularly efficient since the least-squares calculations are already available from the solution procedure in this formulation. Hence, the least-squares functional provides a suitable error indicator for the CDLS meshless method used here. The error indicator at an arbitrary point k which may or may not be a nodal or collocation point can, therefore, be defined as the sum of squared residuals:

$$J(k) = \tfrac{1}{2}([R_k^{(\mathrm{d})}]^2 + \alpha \cdot [R_k^{(\mathrm{t})}]^2 + \beta \cdot [R_k^{(\mathrm{u})}]^2) \tag{36}$$

5. ADAPTIVE REFINEMENT: MESH MOVING STRATEGY

In general, an analyst has no *a priori* knowledge of the location of the areas of the solution domain in which large gradients will occur. An ideal computational algorithm would then require the ability to automatically refine the nodes in zones of high gradients, as the computation proceeds. Adaptive refinements have been extensively used in finite element computations. For finite elements, three methods, namely mesh movement, mesh enrichment and remeshing have been suggested for adaptive refinement.

In mesh movement, the total number of nodes remains constant, but the location of the nodes is changed in order to achieve a better overall distribution of the error. Full details of the strategy adopted for moving the mesh and handling some of the subsequent problems which may arise have been given in details elsewhere [24]. In mesh-enrichment method, the original mesh is held fixed and hierarchical elements [25] or simply more elements are added. In the remeshing approach, a completely new mesh is constructed using the information acquired from the previous computation [26]. It is obvious that mesh movement approach is more suitable than mesh enrichment because the scale of the problem remains constant when a mesh movement strategy is used. Mesh movement approach, however, would encounter some serious problems such as element distortion in the finite element context. This is the main reason why other methods of refinements have been favoured in finite element computation. Mesh movement strategy, however, could be easily and efficiently used with meshless methods since no element distortion is associated with the method. It should be noted that the mesh movement technique can be used in conjunction with CDLS meshless method to adaptively adjust both nodal and collocation points to improve the quality of the solution obtained with a pre-specified number of nodes and collocation points. Here, only a nodal refinement procedure is used.

When a node refinement is required, springs of prescribed stiffness are placed between each pair of nodes belonging to the same sub-domain and the nodes are then moved until the spring system is in equilibrium. For this, consider two nodes i and j belonging to the same sub-domain, as shown in Figure 2. The force f_{ij} induced in the spring ij connecting these two nodes is taken to be a function of the distance between nodes

$$f_{ij} = c_{ij}(X_i - X_j) \tag{37}$$

where c_{ij} is the stiffness of the spring and X_i and X_j are the coordinates of nodes i and j, respectively. The stiffness of each spring is assumed to be some function of the solution error. In

DOI: 10.1002/fld

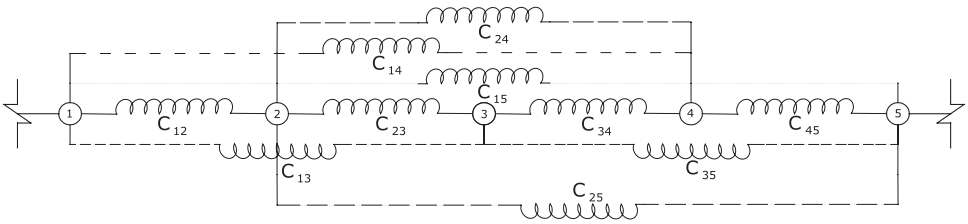

Figure 2. System of springs in a sample sub-domain with five nodes.

FEM, the solution error is obtained using approximation theory. Here, c_{ij} is calculated using the values of the error indicator at nodes i and j as follows:

$$c_{ij} = (J(i) + J(j))/|\overline{X}_i - \overline{X}_j| \tag{38}$$

Here, $\overline{X}_i$ and $\overline{X}_j$ are the initial coordinates of nodes i and j on the initial unrefined mesh of nodes, respectively. This means that the stiffness of each spring is a function of initial nodal coordinates. The spring ij exerts two forces of the same magnitude but opposite direction on nodes i and j defined as

$$f_i^{ij} = c_{ij}(X_i - X_j) \tag{39}$$

$$f_j^{ij} = -c_{ij}(X_i - X_j) \tag{40}$$

where f_i^{ij} and f_j^{ij} are used to represent the forces exerted by spring ij on nodes i and j, respectively. These equations can be written in a matrix form to define the stiffness relation for each spring ij in the system:

$$\begin{bmatrix} f_i \\ f_j \end{bmatrix}^{ij} = \begin{bmatrix} c_{ij} & -c_{ij} \\ -c_{ij} & c_{ij} \end{bmatrix} \begin{bmatrix} X_i \\ X_j \end{bmatrix} \tag{41}$$

Now, one can allow the nodes to move to an equilibrium position. This means that the vector $\mathbf{F}$ of assembled nodal forces should be equal to zero. This requirement leads to the following system of algebraic equation to be solved for the unknown vector of nodal position $\mathbf{X}$:

$$\mathbf{F} = \mathbf{CX} = \mathbf{0} \tag{42}$$

where $\mathbf{C}$ is the stiffness matrix of the system calculated by assembling the stiffnesses of all the springs defined in the system. This process is very similar to the assembly process used in the FEM to derive stiffness of a system using the element stiffness matrices. It should be noted that the resulting system of equations is clearly linear as c_{ij} for each spring is considered as a function of the nodal positioning of the initial nonadapted mesh and, therefore, is not updated during the solution of the system of equations. It should also be noted that the equation system defined by (42) is singular before any boundary condition is considered. The boundary condition used here to solve this system of equations is defined by the requirement that the boundary nodes, 2 for 1-D problems used here, should not be allowed to move. The sub-domains used to construct the spring system may or may not correspond to the sub-domains used to construct the meshless solution.

DOI: 10.1002/fld

Here, the most simple of spring systems is used in which each node is only connected to two nodes on two sides of the node.

Application of this procedure leads to a nodal positioning which is expected to yield a uniform distribution of the error indicator defined by Equation (36). This could be useful for problems of elliptic nature with smooth solutions in which the error indicator calculated *via* Equation (36) is bounded. Application of this error estimator, however, to problems with high gradient solution could result in a nodal positioning with very high concentration of nodes around zones of high gradient solutions. As a remedy to this problem a mapping is proposed and used here to ensure that a smooth distribution of nodes is obtained. Here, a simple projection of the power-law type

$$e = J^b \tag{43}$$

is used where e is the projected error estimator, J is the value of the raw error indicator defined by Equation (36) and b is a constant parameter calculated based on the requirement that the ratio of the maximum and minimum nodal spacing in the final refined mesh is equal to a pre-specified ratio defined as

$$\frac{h_{\max}}{h_{\min}} = \frac{\overline{h}_{\max}}{\overline{h}_{\min}} \tag{44}$$

Here, $h_{\max}$ and $h_{\min}$ are the maximum and minimum values of the nodal spacing in the adapted mesh, while $\overline{h}_{\max}$ and $\overline{h}_{\min}$ denote the required maximum and minimum nodal spacings in the refined mesh of nodes, set by the user. Assuming a linear relationship between nodal spacing of the refined mesh and the projected error estimator:

$$\frac{h_{\max}}{h_{\min}} = \frac{e_{\max}}{e_{\min}} \tag{45}$$

Requirement of Equation (44) leads to the calculation of projection parameter as follows:

$$b = \ln\left(\frac{\overline{h}_{\max}}{\overline{h}_{\min}}\right) \Big/ \ln\left(\frac{J_{\max}}{J_{\min}}\right) \tag{46}$$

where $J_{\max}$ and $J_{\min}$ are the maximum and minimum values of the raw error indictor calculated at nodal points. The proposed projection mechanism is meant to lead to an adapted distribution of nodes with nodal spacing in proportion to $\overline{h}_{\max}$ and $\overline{h}_{\min}$ at the location of lowest and highest error indicator defined by Equation (36). Early experiments, however, showed that this goal could not be achieved when Equation (46) is used to calculate the projection parameter. This can be attributed to the complex nonlinear relationship between adapted nodal spacing and projected error indicator implied by Equation (43). An iterative procedure is, therefore, devised in which the value of the projection parameter is updated at each iteration as follows:

$$b^{k+1} = b^k \ln\left(\frac{\overline{h}_{\max}}{\overline{h}_{\min}}\right) \Big/ \ln\left(\frac{h^k_{\max}}{h^k_{\min}}\right) \tag{47}$$

DOI: 10.1002/fld

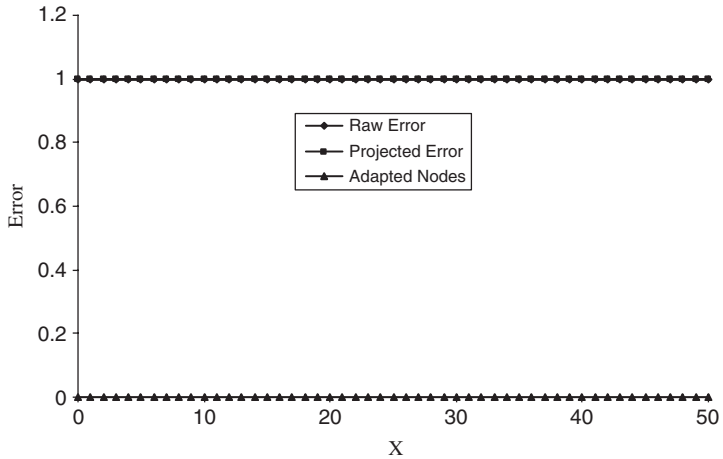

Figure 3. Distribution of projected error and the resulting mesh of nodes for constant raw error estimator.

where $h^k_{\max}$ and $h^k_{\min}$ are the maximum and minimum values of the nodal spacing of the adapted mesh at iteration k using the value of b^k. This procedure will make sure that the ratio of nodal spacing in the refined mesh is exactly the same as the required ratio. Our experiments indicate that usually a few iterations are required by the above procedure to converge. It should be remarked that when the distribution of the raw error indicator is uniform, $J_{\max} = J_{\min}$, the above procedure is bypassed assuming a value of unity for the projection parameter b.

It should be emphasized that the proposed error estimator and adaptive refinement is a one-step process and not an iterative procedure. To exemplify the projection method and mesh movement strategy, four simple examples are included in this section assuming known distributions of the raw error indicator. These examples assume three different simple distributions of raw error indicator, namely uniform, linear and quadratic which should theoretically lead to uniform, linearly varying and quadratically varying mesh of nodes. A fourth sinusoidal distribution of error is also included to test the performance of the method for the problems with zones of high gradient solutions. Figures 3–6 show the distributions of assumed raw error estimator, calculated projected error and refined mesh of nodes for four cases of uniform, linear, quadratic and sinusoidal variations of raw error estimator, respectively. The processes of projection and refinement are carried out on an initial mesh of 51 uniformly distributed nodes, assuming a value of 5 and 0.5 for $\overline{h}_{\max}$ and $\overline{h}_{\min}$, respectively. It is clearly seen that in the first example, Figure 3, the refined mesh coincides with the initial uniform mesh due to the fact that both of the raw and projected errors are uniform throughout the mesh which indicates no need for mesh refinement. This situation is well recognized by the projection method and the proposed refinement strategy. For linear distribution of raw error estimator, shown in Figure 4, the resulting nodal spacing varies linearly from about 4.6 to 0.46 unit of length in the refined mesh. It is clearly seen that the ratio of the maximum and minimum nodal spacing of the refined mesh is exactly the same as the one set by the user through the definition of $\overline{h}_{\max}$ and $\overline{h}_{\min}$. Figure 5 shows the results for the third case where a quadratic distribution of estimated error is assumed. Here, the refined nodes are distributed more closely around the centre of the domain as expected. The maximum and minimum nodal spacings in this case are calculated to be 5.4 and 0.54. Finally, Figure 6 shows the results of the mesh refinement strategy for the sinusoidal distribution

DOI: 10.1002/fld

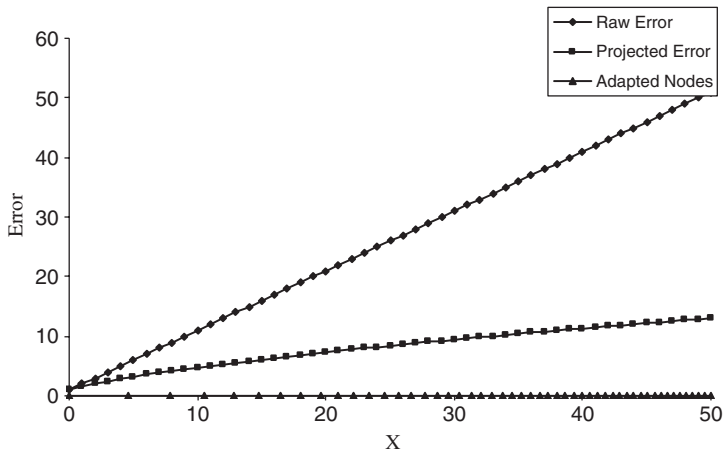

Figure 4. Distribution of projected error and the resulting mesh of nodes for linear variation of raw error estimator.

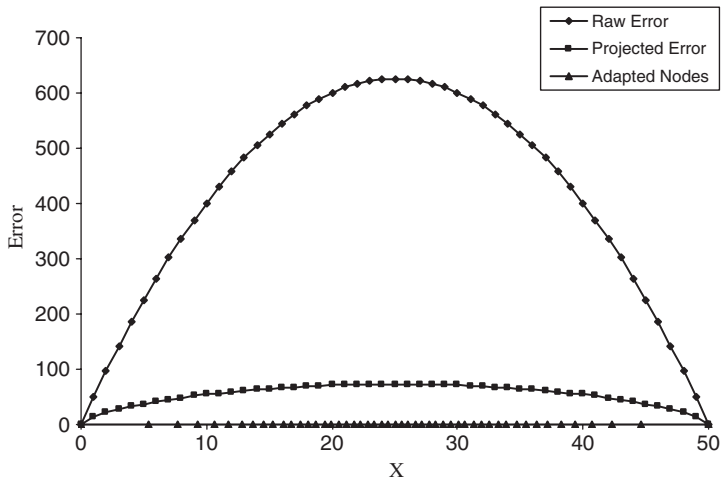

Figure 5. Distribution of projected error and the resulting mesh of nodes for quadratic variation of raw error estimator.

of error estimator with maximum and minimum nodal spacings of 4.6 and 0.46, respectively. The last example shows the capability of the proposed projection method and the refinement strategy to recognize the zones of larger error and act accordingly by moving the mesh of nodes on and around these areas. The maximum number of updating iterations required in these examples was equal to 6.

The efficiency of the proposed error estimator, projection method and mesh refinement strategy on engineering problems will be further verified in the next section when considering two numerical examples of nonlinear burgers and shallow-water equations.

DOI: 10.1002/fld

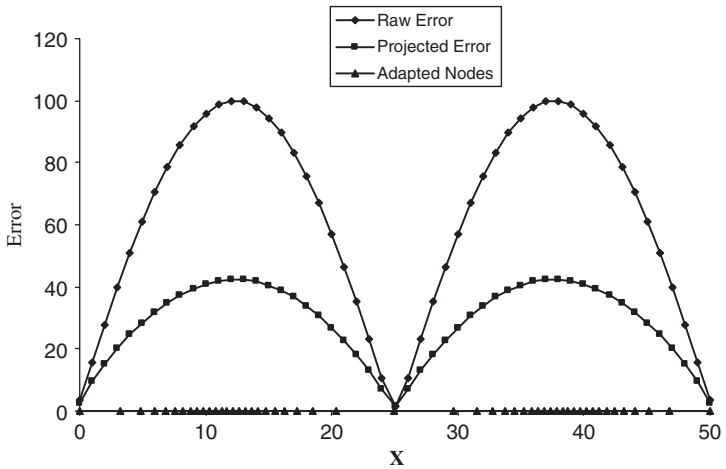

Figure 6. Distribution of projected error and the resulting mesh of nodes for sinusoidal raw error estimator.

6. SOLUTION STRATEGY

In this paper, a time marching method is used for the solution of steady-state hyperbolic problems. For this, the transient form of equations governing the underlying problems are discretized in time and solved *via* a time marching method towards steady-state solution. Consider the general form of differential equations governing the transient hyperbolic problems written in the matrix form as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}(\mathbf{u})\frac{\partial \mathbf{u}}{\partial x} = \mathbf{Q}(\mathbf{u}) \quad \text{on } \Omega \tag{48}$$

subject to appropriate Dirichlet boundary condition:

$$\mathbf{u} = \overline{\mathbf{u}} \quad \text{on } \Gamma_{\mathrm{u}} \tag{49}$$

Here, $\mathbf{u}$ denotes the unknown problem vector, $\mathbf{A}$ is the Jacobian matrix which is generally a function of the unknown vector $\mathbf{u}$ and $\mathbf{Q}$ is the source term. A semi-discretization is first carried out using the θ method in time

$$\mathbf{u}^{n+1} - \mathbf{u}^n + \Delta t\theta\left[\mathbf{A}^{n+1}\frac{\partial \mathbf{u}^{n+1}}{\partial x} - \mathbf{Q}^{n+1}\right] + \Delta t(1-\theta)\left[\mathbf{A}^n\frac{\partial \mathbf{u}^n}{\partial x} - \mathbf{Q}^n\right] = 0 \tag{50}$$

with $\frac{1}{2} \leqslant \theta \leqslant 1$. Assuming $\mathbf{Q} = \mathbf{S}\mathbf{u}$, the linearized residuals in the problem domain and its boundaries are now defined as

$$\mathbf{R}_{\Omega}^{n+1} = \mathbf{u}^{n+1} - \mathbf{u}^n + \Delta t\theta\left[\mathbf{A}^n\frac{\partial \mathbf{u}^{n+1}}{\partial x} - \mathbf{S}^n\mathbf{u}^{n+1}\right] + \Delta t(1-\theta)\left[\mathbf{A}^n\frac{\partial \mathbf{u}^n}{\partial x} - \mathbf{Q}^n\right] \quad \text{on } \Omega \tag{51}$$

$$\mathbf{R}_{\Gamma_{\mathrm{u}}}^{n+1} = \mathbf{u}^{n+1} - \overline{\mathbf{u}} \quad \text{on } \Gamma_{\mathrm{u}} \tag{52}$$

DOI: 10.1002/fld

To recast the problem into proper form defined by Equations (23) and (25), the differential operators can now be defined as

$$L(\cdot) = (\cdot) + \Delta t\theta \left[\mathbf{A}_j^n \frac{\partial(\cdot)}{\partial x_j} - \mathbf{S}^n(\cdot) \right] \tag{53}$$

$$\mathbf{f} = \mathbf{u}^n - \Delta t(1-\theta) \left[\mathbf{A}_j^n \frac{\partial \mathbf{u}^n}{\partial x_j} - \mathbf{Q}^n \right] \tag{54}$$

$$\mathbf{B}(\cdot) = 0.0 \tag{55}$$

The system of equations can now be formed and solved at each time step and the required solution produced in a time marching manner until a steady-state solution is reached. It should be noted here that the proposed method is stable for any time and space step sizes due to the implicit nature of the method.

7. NUMERICAL EXAMPLES

In this section, the application of proposed adaptive CDLS meshless method to solve two examples from fluid dynamics discipline, namely nonlinear Burger's equation and steady flow over an ogee spillway governed by the one-dimensional shallow-water equation is considered. All the solutions presented are obtained using the definition of $s_{\max}$ in a way that at least two nodes are chosen in the local interpolation domain. A polynomial basis of order zero ($P = [1]$) is used in all problems in order to construct MLS shape functions.

7.1. Steady Burger's equation

This is a problem governed by the inviscid Burger's equation defined by the following parameters of Equation (48)

$$\mathbf{A} = u, \quad \mathbf{Q} = 0$$

The problem is solved on the domain $0 \leqslant x \leqslant 1$ with the following initial and boundary conditions:

$$u(0) = 1 - 2x, \quad 0 \leqslant x \leqslant 1$$

$$u(t) = 1, \quad x = 0.0$$

$$u(t) = -1, \quad x = 1$$

The exact solution to this problem is represented by a discontinuity located at the centre of the domain. First, this problem is solved on a mesh of 51 equally distributed nodes using a time step size of 0.01. The solution of the problem is attempted by using 501 equally distributed collocation points, 51 of which coincide with the nodal points. The solution to this problem is shown in Figure 7 where the discontinuity is captured with only three nodal points. By neglecting the small over- and under-shoots, the solution can be considered as a numerically exact solution.

The raw estimated error of this solution is shown in Figure 8, while the projected error along with the adapted mesh of nodes is shown in Figure 9. Figure 10 illustrates the high-quality solution obtained on the adaptively generated mesh which is the exact numerical solution to this problem.

DOI: 10.1002/fld

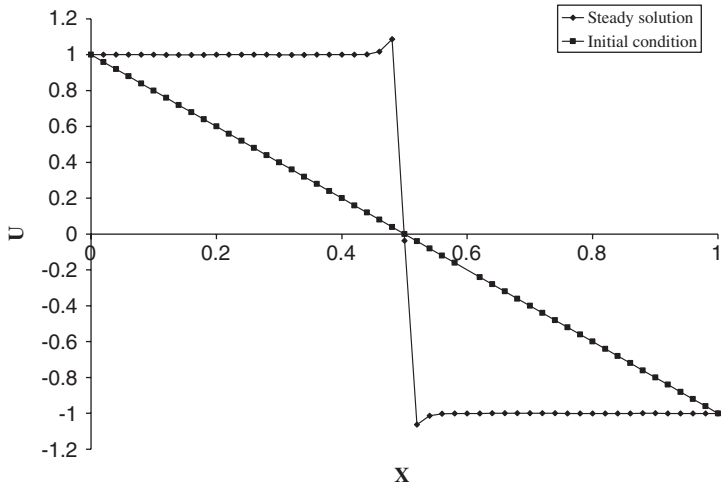

Figure 7. Steady solution on a uniform mesh of 51 nodes (Burger's equation).

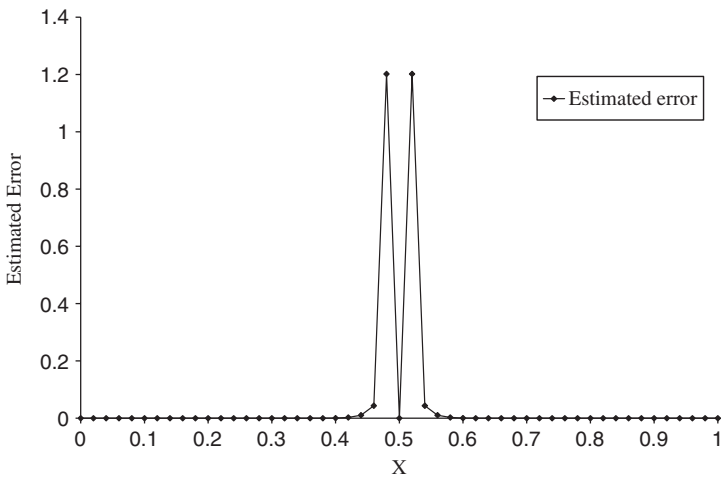

Figure 8. Estimated error of the solution obtained on a uniform mesh of 51 nodes (Burger's equation).

It could be useful to note that the projected error is obtained using the values of $\overline{h}_{\max} = 0.15$ and $\overline{h}_{\min} = 0.001$. The maximum and minimum values of raw error estimator are calculated to be $J_{\max} = 1.20$ and $J_{\min} = 6.0\text{E}-13$, leading to a final value of $b = 0.2$ and, consequently, a smoother distribution of the projected error between $e_{\max} = 1.04$ and $e_{\min} = 0.0036$. It is the small value of $b = 0.2$ which is responsible for scaling down the peak value of the error indicator from 1.2 to 1.04, while scaling up the minimum value of the error from $6.0\text{E}-13$ to 0.0036 when projected. The resulting mesh of nodes shown in Figure 9, has maximum and minimum nodal spacings equal to $h_{\max} = 0.95$ and $h_{\min} = 0.000635$.

DOI: 10.1002/fld

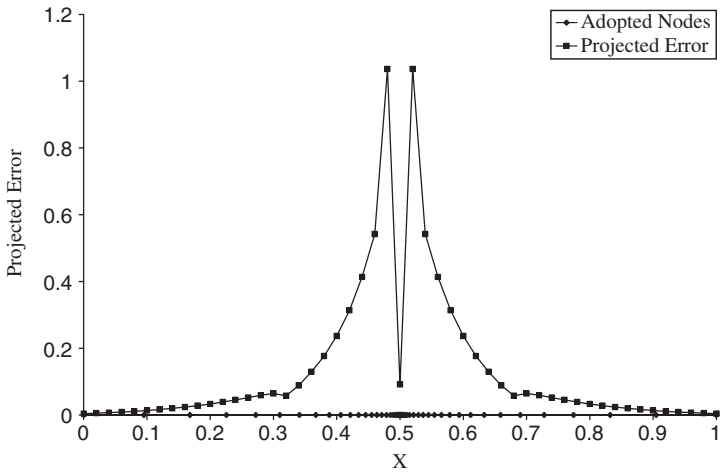

Figure 9. Projected error of the solution obtained on a uniform mesh of 51 nodes (Burger's equation).

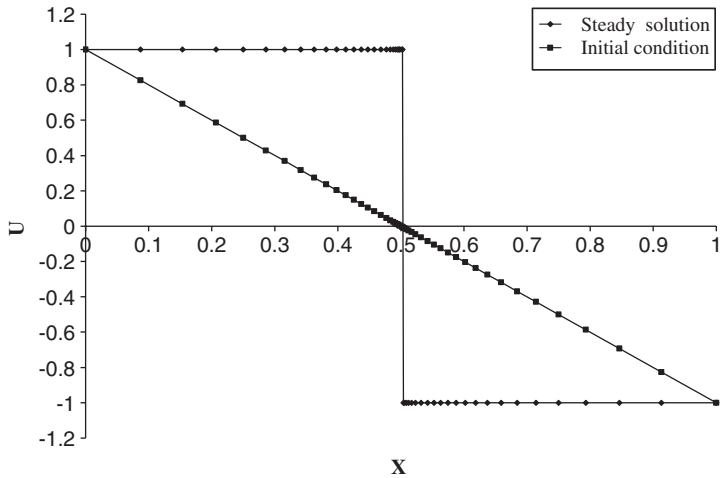

Figure 10. Final solution obtained on the adapted mesh of 51 nodes (Burger's equation).

7.2. *Inviscid flow an ogee spillway*

The flow of inviscid water over an ogee spillway is considered here to be governed by the nonlinear shallow-water equations in one dimension defined by the following parameters of Equation (48):

$$\mathbf{u}=\begin{bmatrix} H+\eta \\ (H+\eta)u \end{bmatrix}, \quad \mathbf{A}=\begin{bmatrix} 0 & 1 \\ -u^2+g(H+\eta) & 2u \end{bmatrix}, \quad \mathbf{Q}=\begin{bmatrix} 0 \\ g(H+\eta)\mathrm{d}H/\mathrm{d}x \end{bmatrix}$$

Here, H is the flow depth, η is the surface elevation, u is the average velocity, $g=9.81$ is the acceleration due to gravity and $\mathrm{d}H/\mathrm{d}x$ is the bed slope.

DOI: 10.1002/fld

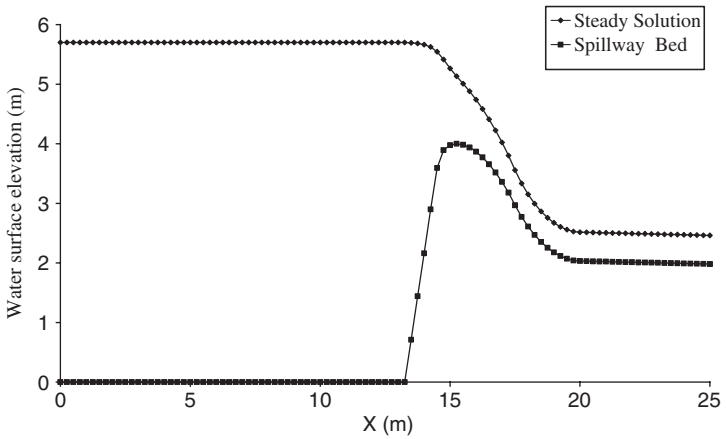

Figure 11. Steady solution obtained on a uniform mesh of 101 nodes (flow over an ogee spillway).

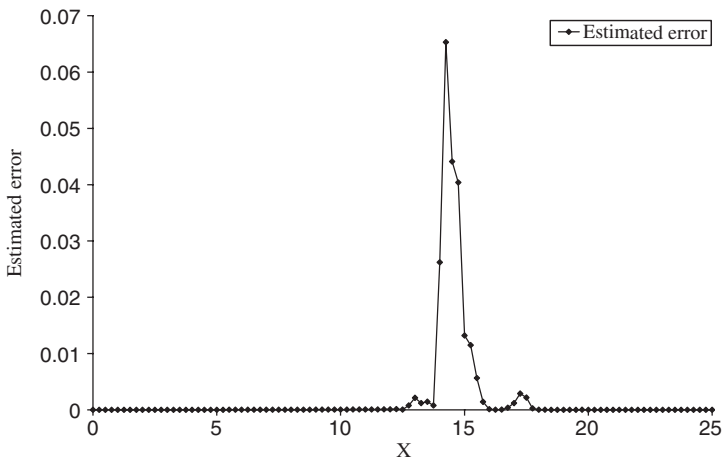

Figure 12. Estimated error obtained on a uniform mesh of 101 nodes (flow over an ogee spillway).

This problem is first solved on a mesh of 101 equally distributed nodes using a time step size of 0.2. The solution of the problem is attempted using 501 equally distributed collocation points, 101 of which coincide with the nodal points. The numerical results representing the water elevation are shown in Figure 11.

The estimated error for the above solution is shown in Figure 12, while the projected error along with the position of generated nodes is shown in Figure 13. Figure 14 illustrates the numerical water surface elevation along the spillway obtained on the adapted mesh of nodes. The improvement made in the final solution is not as apparent as in the solution of Burger's equation due to the fact that the exact solution of this problem is smooth enough to be properly represented on a uniform mesh. It could be again useful to note that the projected error is obtained in this example using the values of $\overline{h}_{\max} = 1.0$ and $\overline{h}_{\min} = 0.0025$. The maximum and minimum values of raw

DOI: 10.1002/fld

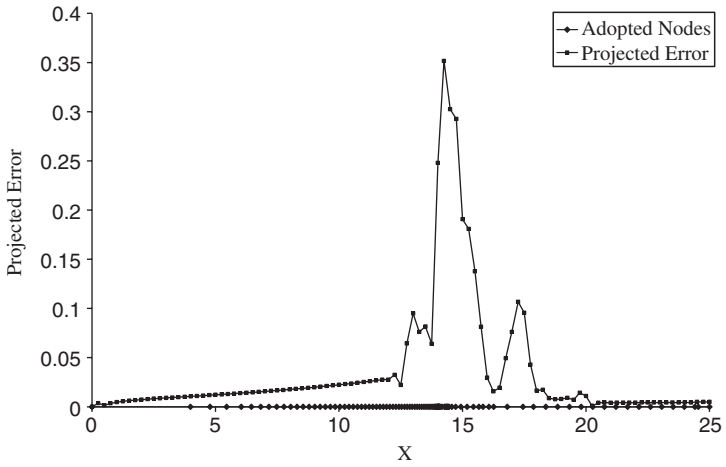

Figure 13. Projected error obtained on a uniform mesh of 101 nodes (flow over an ogee spillway).

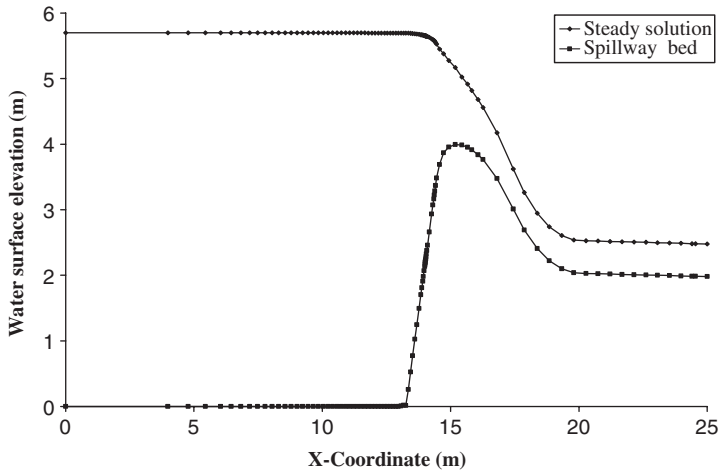

Figure 14. Steady solution obtained on an adapted mesh of 101 nodes (flow over an ogee spillway).

error estimator are calculated to be $J_{\max} = 0.0653$ and $J_{\min} = 1.04\text{E}{-}08$, leading to a value of $b = 0.383$ and, consequently, a smoother distribution of the projected error between $e_{\max} = 0.352$ and $e_{\min} = 0.0876$. Again, here the larger range of the raw error indicator is shrunk into a smaller range due to the fact that the projection parameter is smaller than one. The resulting mesh of nodes shown in Figure 11, has maximum and minimum nodal spacings equal to $h_{\max} = 4.0$ and $h_{\min} = 0.01$.

A future note should be added regarding the difference between the raw and projected error in the examples considered. The peak value of the raw error indicator is much larger in the first example due to the fact that the exact solution to this problem contains a discontinuity located at the centre of the domain, while the exact solution of the second example contains a low gradient

DOI: 10.1002/fld

area located at the spillway crest. This difference is relaxed *via* projection as observed from the results shown in Figures 9 and 13. The minimum value of the raw error indicator, on the other hand, is very small in the first example compared with the second one. This is because the exact solution to the Burger's equation contains two constant fields which can be accurately represented by most of the numerical algorithms leading to a very small error.

8. CONCLUDING REMARKS

An error estimator and adaptive refinement procedure is developed in this paper to be used in conjunction with the CDLS meshless method for the solution of hyperbolic problem with high gradient solution. The squared residual is proposed as the error estimator which is naturally related to the least-squares functional used for the construction of the meshless formulation. A mesh moving strategy is then used to redistribute the nodes such that a uniform distribution of error indicator is achieved on the adapted mesh. A single-parameter projection strategy was also used to limit the nodal spacing of the adapted mesh within a predefined minimum and maximum values set by the user. The efficiency and effectiveness of the proposed error estimator and mesh moving strategy was successfully tested against two benchmark examples, namely nonlinear Burger's equation with a shocked solution and nonlinear shallow-water equation with a low gradient smooth solution.

REFERENCES

1. Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
2. Nayroles B, Touzot G, Villon P. Generalizing the finite element method diffuse approximation and diffuse element. *Computational Mechanics* 1992; **10**:307–318.
3. Liu WK, Jun S, Zhang Y. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering* 1995; **20**:1081–1106.
4. Gingold RA, Moraghan JJ. Smooth particle hydrodynamics: theory and application to non spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**:375–389.
5. Onate E, Idelsohn SR, Del Pin F, Aubry R. The particle finite element method: an overview. *International Journal of Computational Methods* 2004; **12**:267–308.
6. Atluri SN, Zhu T. A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* 1998; **22**:117–127.
7. Afshar MH, Arzani H. Solving Poisson's equations by the discrete least square meshless method. *WIT Transactions on Modelling and Simulation* 2005; **42**:23–32.
8. Liu GR, Tu ZH. An adaptive procedure based on background cells for meshless methods. *Computer Methods Applied Mechanics and Engineering* 2002; **191**:1923–1943.
9. Zienkiewicz OC, Taylor RL. *The Finite Element Method* (4th edn). McGraw-Hill: New York, 1989.
10. Babuska I, Miller A. A feedback finite element method with a-posteriori error estimation: part I. The finite element method and some basic properties of the a posteriori error estimator. *Computer Methods in Applied Mechanics and Engineering* 1987; **61**:1–40.
11. Orkisz J. Meshless finite difference method II. Adaptative approach. In *Computation Mechanics*, Idelsohn S, Oñate E, Dvorkin E (eds). CIMNE: Barcelona, Spain, 1998.
12. Laouar T, Villon P. Adaptative analysis for the diffuse element method. In *Computational Mechanics*, Idelhson S, Oñate E, Dvorkin E (eds). CIMNE: Barcelona, Spain, 1998.
13. Duarte A, Oden JT. An *h–p* adaptive method using clouds. *TICAM Report 96–97*, 1996.
14. Gavete L, Falcón S, Ruiz A. An error indicator for the element free Galerkin method. *European Journal of Mechanics A/Solids* 2001; **20**:327–341.
15. Lancaster P, Salkauskas K. *Curve and Surface Fitting Introduction*. Academic Press: New York, 1986.

DOI: 10.1002/fld

16. Liu GR, Gu YT. A local radial point interpolation method (LR-PIM) for free vibration analyses of 2-D solids. *Journal of Sound and Vibration* 2001; **246**(1):29–46.
17. Lei G. Moving Kriging interpolation and element-free Galerkin method. *International Journal for Numerical Methods in Engineering* 2003; **56**:1–11.
18. Wang QX, Li H, Lam KY. Development of a new meshless—point weighted least-squares (PWLS) method for computational mechanics. *Computational Mechanics* 2005; **35**:170–181.
19. Firoozjaee AR, Afshar MH. Discrete least square meshless method (DLSM) with sampling points for the solution of elliptic partial differential equations. *International Journal of Engineering Analysis with Boundary Elements* 2007, under review.
20. Babuska I. The self adaptive approach in the finite element method. In *Mathematics of Finite Elements and Applications*, Whiteman JR (ed.). Academic Press: London, 1975; 125–142.
21. Babuska I, Rheinboldt WC. Reliable error estimation and mesh adaptation for the finite element method. In *Computational Methods in Nonlinear Mechanics*, Oden JT (ed.). North-Holland: Amsterdam, 1980; 67–108.
22. Carey GF, Humphrey DL. Mesh refinement and iterative solution methods for finite element computations. *International Journal for Numerical Methods in Engineering* 1981; **17**:1717–1734.
23. Jiang B-N. Least squares finite element methods with element-by-element solution including adaptive refinement. *Ph.D. Thesis*, University of Texas at Austin, 1986.
24. Loner R, Morgan K, Zienkiewicz OC. Adaptive grid refinement for the Euler and compressible Navier–Stokes equations. *Proceedings of the Conference on Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Lisbon, 1984.
25. Zienkiewicz OC, Morgan K. *Finite Element and Approximation*. Wiley: New York, 1983.
26. Zienkiewicz OC, Huang HC, Pastor M. Localization problems in plasticity using the finite elements with adaptive remeshing. *International Journal for Numerical Analytical Methods in Geomechanics* 1995; **19**:48–127.

DOI: 10.1002/fld