International Journal of Mechanical Sciences 70 (2013) 26–38

Contents lists available at SciVerse ScienceDirect

International Journal of Mechanical Sciences

journal homepage: www.elsevier.com/locate/ijmecsci

Upper bound limit analysis using radial point interpolation meshless method and nonlinear programming

Fengtao Liu *, Jidong Zhao [1]

Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China

ARTICLE INFO

Article history:
Received 17 August 2011
Received in revised form
5 September 2012
Accepted 14 January 2013
Available online 17 February 2013

Keywords:
Upper bound limit analysis
RPIM method
Nonlinear programming
Cartesian Transformation Method (CTM)

ABSTRACT

This paper presents a numerical upper bound limit analysis using radial point interpolation method (RPIM) and a direct iterative method with nonlinear programming. By expressing the internal plastic dissipation power with a kinematically admissible velocity field obtained through RPIM interpolation, the upper bound problem is formulated mathematically as a nonlinear programming subjected to single equality constraint which is solved by a direct iterative method. To evaluate the integration of internal power dissipation rate without any background integral cell, a new meshless integration technique based on Cartesian Transformation Method (CTM) is employed to transform the domain integration first as boundary integration and then one-dimensional integration. The effectiveness and accuracy of the proposed approach are demonstrated by two classical limit analysis problems. Further discussion is devoted to optimal selection of relevant parameters for the computation.

1. Introduction

As a proved direct and efficient approach to estimate the ultimate bearing capacity for structures, limit analysis has long been used in the design of a wide range of applications in civil and geotechnical engineering, such as shells, plates, foundations, retaining walls and slopes. Relying frequently on hand calculation in early days, limit analysis has now been dominated by numerical solutions with the aid of modern computers. Almost all engineering structures, no matter how complex their shapes and/or loading conditions might be, can now be conveniently discretized by numerical methods, such as Finite Element Method (FEM). Both the lower bound and the upper bound theorem can be reformulated as numerical optimization problems, and be applied to the discretized physical domain in sought for limit loads.

Finite elements and linear programming have commonly been used for numerical limit analysis for long (see, e.g., [51,1,20,6,17,63,64,71,62]). With the recent progress in the theory of nonlinear programming (hereafter shortened as NLP), a wide variety of advanced numerical techniques have been developed in limit analysis and more rigorous solutions can be sought. Typical examples include the constrained nonlinear optimization formulation based on mixed finite elements developed by Zouain et al. [75] for cohesive materials, and its recent generalization to frictional-cohesive materials by [49,50]. Recently, more advanced nonlinear programming techniques such as those based on the primal-dual interior point method [32,58–61] and those based on the second-order cone programming (SOCP) [52–55,56,34,19,37,38] have also been successfully applied to the limit analysis involving different materials.

Upper bound limit analysis has traditionally been based on finite element method for both purely cohesive materials and cohesive-frictional materials. The plastic incompressible condition in the analysis can be typically treated by such techniques as discontinuous velocity field [62,49,50], penalty function method [47,43], mixed formulations [18,2,9] and variational principle [68]. High-order elements [71,53] and cell-based smoothing finite element method [45,37] have also been developed to overcome the issue of volumetric locking when the penalty function method is used. In treating non-differential plastic dissipation function in numerical upper bound limit analysis, a wide range approaches including viscous plastic regularization [30,10], smoothed method ([2,3,23,69,29], etc.) and direct iterative method based on distinguished rigid and plastic regions [72,73,47,8,40–43] have been employed.

More recently, meshless methods have received much attention in applications relevant to numerical limit analysis. For example, Chen et al. [13] and Le et al. [36] have developed a lower bound approach using element-free Galerkin (EFG) approach with moving least squares (MLS) method to construct the self-equilibrium stress basis vectors and the static admissible stress field. Le et al. [35] have also developed an upper bound limit analysis approach based on EFG method, in which the MLS

* Corresponding author. Tel./fax: +852 23581534.
E-mail addresses: celiuft@gmail.com (F. Liu), jzhao@ust.hk (J. Zhao).
[1] Tel.: +852 23588481.

0020-7403/$ - see front matter © 2013 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.ijmecsci.2013.01.017

approximation is employed to construct the kinematically admissible velocity field. The upper bound limit analysis is then formulated as a SOCP problem and solved by a primal-dual interior point method originally proposed by Andersen et al. [3]. Moreover, as an extension of limit analysis, shakedown analysis of structures and solids with repeated loads can also be performed based on EFG [14] and meshless local Petrov-Galerkin (MLPG) method [15]. Without requiring the discretization of physical domain into meshes, meshless methods have been proved to offer improved computational efficiency over FEM with reasonable accuracy.

There are two key issues deciding whether or not an element-free Galerkin method can be successfully applied to this upper bound limit analysis. The first is pertinent to how the essential boundary conditions can be effectively reinforced, and the second is related to accurate numerical integration of the plastic dissipation power. First, it is well-known that in EFG method the shape function $\Phi_i(\mathbf{x}_j)$ lacks the property of Kronecker delta function, i.e. $\Phi_i(\mathbf{x}_j) \neq \delta_{ij}$, where δ_{ij} is the Kronecker delta function. It is hence difficult to ensure that the approximation of nodal displacement $u^h(\mathbf{x}_i)$ is exactly equal to the fictious nodal values $\hat{u}_i$ at node $\mathbf{x}_i$, i.e., $u^h(\mathbf{x}_i)=\sum \Phi_i(\mathbf{x}_j)\hat{u}_i \neq \hat{u}_i$. Consequently, the displacement boundary conditions cannot be directly enforced, i.e., $\hat{u}_b \neq \bar{u}$, where $\hat{u}_b$ is the fictious nodal value at boundary node $\mathbf{x}_b$ and $\bar{u}$ is the prescribed displacement. We notice that Le et al. [35] have adopted a collocation method proposed by Zhu and Atluri [74] to treat the boundary conditions. This method, however, may lead to increasing constraints for the NLP problem. Second, numerical integration of dissipation function has traditionally been performed by using either nodal integration method (see, e.g. [4]), or the Gauss quadrature based on an integral background cell (see, [16]). Chen et al. [12,11] have also developed a stabilized conforming nodal integration (SCNI) which proves to be robust but needs a voronoi cell. Various issues regarding accuracy and efficiency still need to be tackled with the various methods.

This paper presents a study using EFG method for limit analysis, in an attempt to improve its performance in the above two aspects. A novel numerical procedure will be proposed for upper bound limit analysis. We shall employ a radial point interpolation method (RPIM) originally proposed by Wang and Liu [67] to construct the kinematically admissible velocity field. With the built-in property of Kronecker delta function in the shape function of RPIM, it is expected to resolve the first issue concerning the enforcement of boundary conditions. Meanwhile, we shall employ a novel meshless integration technique based on the Cartesian Transformation Method (CTM) developed by Khosravifard and Hematiyan [31]. By using this technique, a domain integration can be sequentially first transformed into a boundary integration and then a one-dimensional (1D) integration, such that no integral background cells are required. A direct iterative method will be used to solve the NLP upper bound problem.

2. Numerical formulation of upper bound approach based on RPIM

2.1. Mathematical description of the upper bound theorem

Under the assumption of small deformation, consider a rigid-perfectly plastic solid V with a boundary S subjected to body forces $\mathbf{g}$ and tractions $\mathbf{t}$ at part of the surface, S_σ. The remaining part of the surface is supposed to be S_u, and $S_\sigma \cup S_u = S$, $S_\sigma \cap S_u = \emptyset$. The upper bound theorem states that the solid will collapse if there exists a kinematically admissible velocity field $\dot{\mathbf{u}} \in U$, such that the rate of external forces work equals to the rate of internal power dissipation. Mathematically, the collapse load multiplier λ can be determined by the following optimization problem:

$$\lambda = \min_{\dot{\mathbf{u}} \in U} \int_V D(\dot{\boldsymbol{\varepsilon}})dV$$
$$\text{s.t.} \quad W_{\text{ext}}(\dot{\mathbf{u}}) = \int_V \mathbf{g}^T\dot{\mathbf{u}}dV + \int_{S_\sigma} \mathbf{t}^T\dot{\mathbf{u}}\ dS = 1 \tag{1}$$

where $\dot{\boldsymbol{\varepsilon}} = \nabla\dot{\mathbf{u}}$ is the plastic admissible strain rate with ∇ being the linear differential operator. U is a set of kinematically admissible velocity field defined below:

$$U = \{\dot{\mathbf{u}} = \dot{\mathbf{u}}_0 \text{ on } S_u,\ W_{ext}(\dot{\mathbf{u}}) > 0\} \tag{2}$$

In addition, associated flow rule is assumed such that the plastic admissible strain rates can be expressed as follows:

$$\dot{\boldsymbol{\varepsilon}} = \dot{\mu}\frac{\partial f}{\partial \boldsymbol{\sigma}} \tag{3}$$

where $\dot{\mu}$ denotes a non-negative plastic multiplier. In Eq. (1), $D(\dot{\boldsymbol{\varepsilon}})$ denotes the plastic dissipation function which may be defined as follows:

$$D(\dot{\boldsymbol{\varepsilon}}) = \max_{\boldsymbol{\sigma} \in K}\{\boldsymbol{\sigma}:\dot{\boldsymbol{\varepsilon}}\} \equiv \boldsymbol{\sigma}_\varepsilon:\dot{\boldsymbol{\varepsilon}} \tag{4}$$

where $\boldsymbol{\sigma}$ is the admissible stresses constrained by the convex yield surface, and $\boldsymbol{\sigma}_\varepsilon$ is the stresses on the yield surface associated with any strain rates $\boldsymbol{\varepsilon}$ through the associated flow rule. K denotes a set of plastic admissible stresses which can be expressed as follows [53]:

$$K = \{\boldsymbol{\sigma}: f\{\boldsymbol{\sigma}\} \leq 0\} \tag{5}$$

Evidently, the mathematical optimization problem in (1) is solvable only if a yield function is appropriately specified. The dissipation function in Eq. (4) can be reformulated as follows [7]:

$$D = \sqrt{\dot{\boldsymbol{\varepsilon}}^T\boldsymbol{\Theta}\dot{\boldsymbol{\varepsilon}}} \tag{6}$$

For a plane strain problem, the stress matrix $\boldsymbol{\Theta}$ can be expressed as

$$\boldsymbol{\Theta} = \begin{bmatrix} \sigma_s^2 & -\sigma_s^2 & 0 \\ -\sigma_s^2 & \sigma_s^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7}$$

whilst for plane stress problem, it can be expressed as

$$\boldsymbol{\Theta} = \frac{1}{3}\begin{bmatrix} 4\sigma_s^2 & -2\sigma_s^2 & 0 \\ -2\sigma_s^2 & 4\sigma_s^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{bmatrix} \tag{8}$$

In both Eqs. (7) and (8) σ_s is the yield stress. Consequently, limit analysis by the upper bound theorem can be recast into the following generalized nonlinear optimization problem:

$$\lambda = \min_{\dot{\mathbf{u}} \in U}\int_V [\sqrt{\dot{\boldsymbol{\varepsilon}}^T\boldsymbol{\Theta}\dot{\boldsymbol{\varepsilon}}}]dV$$
$$\text{s.t.} \quad \int_V \mathbf{f}^T\dot{\mathbf{u}}dV + \int_{S_\sigma} \mathbf{g}^T\dot{\mathbf{u}}dS = 1 \tag{9}$$

For a practical problem with finite domain, the above mathematical optimization problem may be solved by discretization techniques with such numerical methods as finite element method or mesh-free method. In the following subsection, a NLP scheme in conjunction with mesh-free method will be developed for this purpose.

2.2. Nonlinear programming based on radial point interpolation method

2.2.1. Radial point interpolation method

A radial point interpolation method (RPIM) originally proposed by Wang and Liu [67] will be employed to construct a displacement field for the mesh-free method. RPIM is based on local supporting nodes and includes polynomial reproduction in the

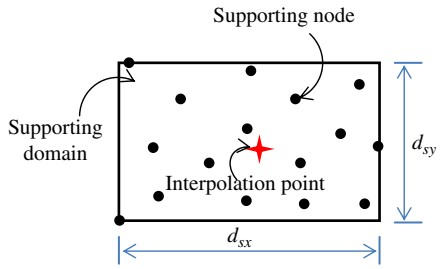

Fig. 1. Illustration of interpolation point and its supporting nodes in a rectangular supporting domain.

Table 1
Two typical radial basis functions used in EFG method.

Item	Name	Expression	Parameters
1	Multiquadric (MQ)	$R_i(\mathbf{x})=(r_i^2+(\alpha_c d_c)^2)^q$	α_c, q
2	Gaussian (EXP)	$R_i(\mathbf{x})=\exp(-\alpha_c(r_i/d_c)^2)$	α_c

radial basis function to restore the consistency of RPIM shape function and to ensure the reproduction of the linear field. Consider a rectangular supporting domain with a set of arbitrary scattered points $\mathbf{x}_i$ ($i=1, 2, \ldots, n$) where n is the number of supporting nodes in the domain (Fig. 1), the approximation of trial function $u(\mathbf{x})$ at any interpolation point $\mathbf{x}$ within the local supporting domain can be expressed as

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_{i=1}^{n} R_i(\mathbf{x})a_i + \sum_{j=1}^{m} p_j(\mathbf{x})b_j = \mathbf{R}^T(\mathbf{x})\mathbf{a} + \mathbf{p}^T(\mathbf{x})\mathbf{b} \tag{10}$$

where a_i is the coefficient for the radial basis $R_i(\mathbf{x})$, b_j is the coefficient for polynomial basis $p_j(\mathbf{x})$.

There are two typical radial basis functions we can choose shape parameters for the construction of RPIM shape function (Table 1). One presents a general form of the original multiquadric (MQ) function proposed by Hardy [26]. This general form of MQ-radial basis function (hereafter denoted as MQ-RBF) has two parameters α_c and q. The other is called the Gaussian radial function, or EXP, which has only one shape parameter α_c. In both cases, the extra parameter d_c denotes a characteristic length which is related to the nodal spacing in the local supporting domain. It can be determined by the following expression [46]:

$$d_c = \frac{\sqrt{A_s}}{\sqrt{n_{A_s}}-1} \tag{11}$$

where A_s is the area of the supporting domain; n_{A_s} is the number of nodes contained in the supporting domain. Take the case of MQ-RBF as an example. The two shape parameters α_c and q in MQ-RBF control the shape of the radial basis function. Fig. 2 shows how the two parameters α_c and q affect the shape of one-dimensional MQ-RBF at a point wherein the parameters α_c and q vary within the range of $\alpha_c \in [0,10]$ and $q \in [0,20]$. In this study, the performance of both RBFs for upper bound limit analysis will be investigated.

The polynomial basis function in Eq. (10) for a two-dimensional problem can be expressed as follows:

$$\mathbf{p}^T(\mathbf{x}) = \begin{bmatrix} 1 & x & y & xy & x^2 & y^2 & \cdots \end{bmatrix} \tag{12}$$

The coefficient a_i and b_j in Eq. (10) can be determined by imposing the interpolation passes through all n field nodes within the local supporting domain of the interpolation point $\mathbf{x}$. This condition leads to n linear equations. In these n linear equations, however, there are a total of $(n+m)$ variables including n coefficients a_i ($i=1,n$) and m coefficients b_j ($j=1,m$). To determine a_i and b_j, another unique approximation constraint needs to be

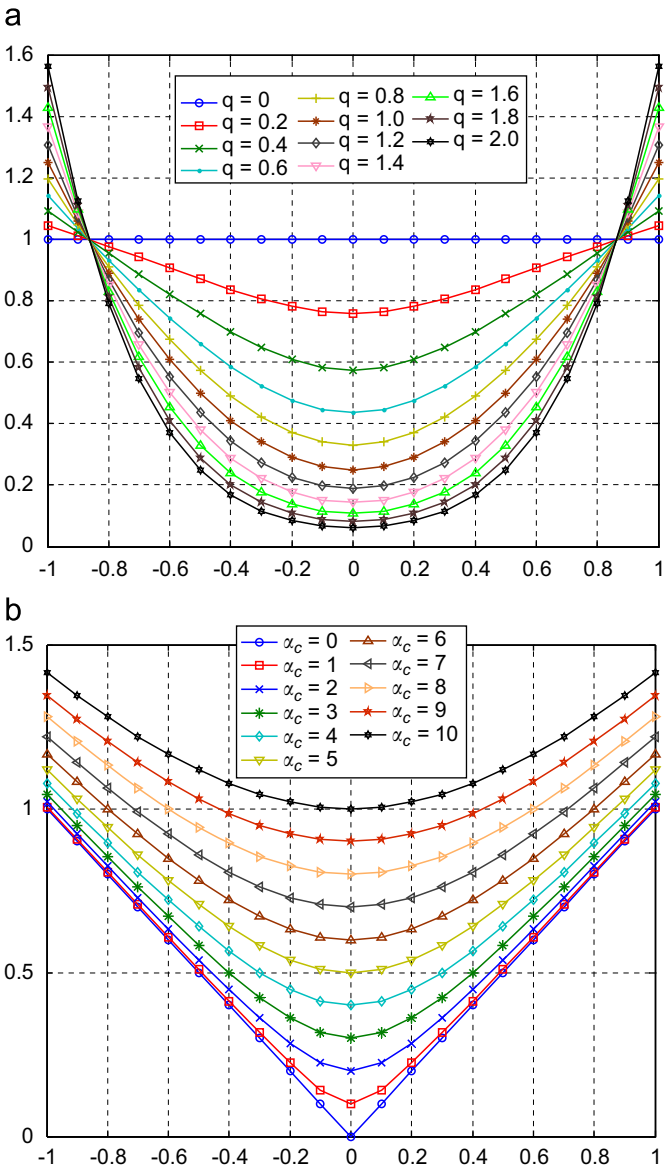

Fig. 2. Variation of the shape of MQ radial basis function with q and α_c: (a) fixing $\alpha_c=4$ and varing q from 0 to 2.0; (b) fixing $q=0.5$ and varying α_c from 0 to 10.

satisfied for the polynomial term [25], which eventually results in the following $(n+m)$ linear equations [46]:

$$\begin{bmatrix} \mathbf{R}_Q & \mathbf{P}_m \\ \mathbf{P}_m^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_s \\ \mathbf{0} \end{bmatrix} \tag{13}$$

where $\mathbf{U}_s = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}^T$ is the vector of function values at all n field nodes within the local supporting domain. $\boldsymbol{R}_Q$ is the moment matrix of RBF given by

$$\mathbf{R}_Q = \begin{bmatrix} R_1(\mathbf{x}_1) & R_2(\mathbf{x}_1) & \cdots & R_n(\mathbf{x}_1) \\ R_1(\mathbf{x}_2) & R_2(\mathbf{x}_2) & \cdots & R_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ R_1(\mathbf{x}_n) & R_2(\mathbf{x}_n) & \cdots & R_n(\mathbf{x}_n) \end{bmatrix} \tag{14}$$

and $\mathbf{P}_m$ is the moment matrix of polynomial basis function defined by

$$\mathbf{P}_m = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_n) & p_2(\mathbf{x}_n) & \cdots & p_m(\mathbf{x}_n) \end{bmatrix} \tag{15}$$

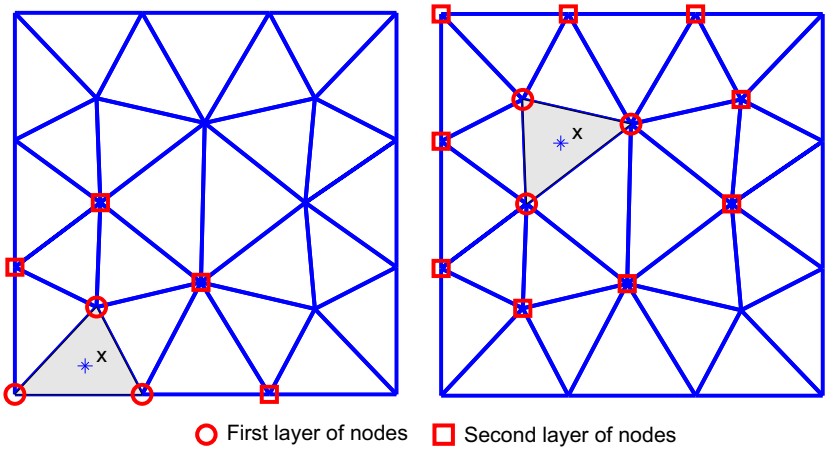

Fig. 3. Illustration of the scheme on selecting nodes for a point x based on triangular cell (after [46]).

Solving Eq. (13), we have

$$\mathbf{a}=\mathbf{S}_a\mathbf{U}_s;\quad \mathbf{b}=\mathbf{S}_b\mathbf{U}_s \tag{16}$$

where

$$\mathbf{S}_a=\mathbf{R}_Q^{-1}-\mathbf{R}_Q^{-1}\mathbf{P}_m\mathbf{S}_b;\quad \mathbf{S}_b=[\mathbf{P}_m^T\mathbf{R}_Q^{-1}\mathbf{P}_m]^{-1}\mathbf{P}_m^T\mathbf{R}_Q^{-1} \tag{17}$$

The interpolation Eq. (10) can then be expressed as

$$u^h(\mathbf{x})=[\mathbf{R}^T(\mathbf{x})\mathbf{S}_a+\mathbf{p}^T(\mathbf{x})\mathbf{S}_b]\mathbf{U}_s=\boldsymbol{\varphi}(\mathbf{x})\mathbf{U}_s \tag{18}$$

where $\boldsymbol{\varphi}(\boldsymbol{x})$ is the RPIM shape functions corresponding to the n field nodes which is given by

$$\boldsymbol{\varphi}(\mathbf{x})=\begin{bmatrix}\phi_1(\mathbf{x}) & \phi_2(\mathbf{x}) & \cdots & \phi_n(\mathbf{x})\end{bmatrix} \tag{19}$$

wherein

$$\phi_k(\mathbf{x})=\sum_{i=1}^{n}R_i(\mathbf{x})S_{ik}^a+\sum_{j=1}^{m}p_j(\mathbf{x})S_{jk}^b \tag{20}$$

where S_{ik}^a denotes the (i, k) element of matrix $\boldsymbol{S}_a$, S_{jk}^b denotes the (j, k)-th element of matrix $\mathbf{S}_b$. $\boldsymbol{R}_Q$ is symmetric, so is the transformed moment matrix $\mathbf{P}_m^T\mathbf{R}_Q^{-1}\mathbf{P}_m$. If the columns in $\boldsymbol{P}_m$ are independent (with a rank of m), the transformed moment matrix $\mathbf{P}_m^T\mathbf{R}_Q^{-1}\mathbf{P}_m$ is invertible by simply invoking the full rank property of $\boldsymbol{R}_Q$. To ensure the columns in $\boldsymbol{P}_m$ are independent, the number of supporting nodes must be equal or greater than that of the columns in $\boldsymbol{P}_m$, i.e., $n\geq m$. To this end, the T2L-scheme demonstrated by Liu [46] is adopted in this paper to select local supporting nodes. The basic idea of this scheme is to select supporting node based on triangulation cell. As shown in Fig. 3, the first layer of nodes (in circle) refers the three nodes of the home cell (shaded triangle), and the second layer contains those nodes (in square) which are directly connected to the three nodes of the first layer.

In addition, because of using T2L-scheme, the polynomial basis functions are linearly independent. And because the transformed moment matrix is symmetric, invertible and full rank, the RPIM shape functions are also linearly independent. Therefore, the n-dimensional vector should be uniquely produced by linear combination of these n shape functions. Letting

$$\mathbf{U}_s=\begin{bmatrix}0 & 0 & \cdots & u_i & \cdots & 0 & 0\end{bmatrix}^T$$

and substituting the above equation into Eq. (18), we have at $\boldsymbol{x}=\boldsymbol{x}_j$

$$u^h(\mathbf{x}_j)=\boldsymbol{\varphi}(\mathbf{x}_j)\mathbf{U}_s=\phi_i(\mathbf{x}_j)u_i$$

when $i=j$, we obtain

$$u_i=\phi_i(\mathbf{x}_i)u_i$$

which leads to

$$\phi_i(\mathbf{x}_i)=1$$

when $i\neq j$, we have

$$u_j=0=\phi_i(\mathbf{x}_j)u_i$$

which requires

$$\phi_i(\mathbf{x}_j)=0$$

Therefore, we obtain the following property of the shape functions:

$$\phi_i(\mathbf{x}_j)=\begin{cases}1, & i=j\\ 0, & i\neq j\end{cases}$$

This proves that the RPIM shape functions possess the Kronecker delta function property.

2.2.2. Discrete nonlinear programming problem based on RPIM

The RPIM has been used with mesh-free method to solve a wide range of practical problems. There have been very few studies, however, in applying RPIM and EFG to limit analysis. Indeed, the kinematically admissible velocity field in a typical upper bound limit analysis can be conveniently constructed by the interpolation using RPIM shape function. We shall employ RPIM in conjunction with EFG in this paper for the upper bound limit analysis. For a two-dimensional domain discretized into a set of nodes $\mathbf{x}_I$ $(I=1,\ldots,n)$, suppose the unknown nodal displacement velocity is

$$\dot{\mathbf{u}}_I=\begin{bmatrix}\dot{u}_{Ix} & \dot{u}_{Iy}\end{bmatrix}^T \tag{21}$$

According to Eq. (18), the approximation $\dot{\mathbf{u}}^h(\mathbf{x})=[\dot{u}_x^h,\dot{u}_y^h]^T$ of velocity field $\dot{\mathbf{u}}(\mathbf{x})=[\dot{u}_x,\dot{u}_y]^T$ at the interpolation point $\boldsymbol{x}=[x,y]^T$ can be formulated as follows:

$$\dot{\mathbf{u}}(\mathbf{x})\approx\dot{\mathbf{u}}^h(\mathbf{x})=\boldsymbol{\Phi}(\mathbf{x})\dot{\mathbf{u}}_I \tag{22}$$

where

$$\mathbf{u}_I=[u_{1x},u_{1y},u_{2x},u_{2y},\cdots u_{nx},u_{ny}]^T;\ \boldsymbol{\Phi}=\begin{bmatrix}\phi_1 & 0 & \phi_2 & 0 & \cdots & \phi_n & 0\\ 0 & \phi_1 & 0 & \phi_2 & \cdots & 0 & \phi_n\end{bmatrix}$$

Here, $\phi_I(\boldsymbol{x})$ is the so-called shape function at the I-th node in Eq. (20). Substituting the strain–displacement equation into Eq. (22), the strain rate field $\dot{\boldsymbol{\varepsilon}}$ can be formulated as follows:

$$\dot{\boldsymbol{\varepsilon}}=\mathbf{D}\cdot\dot{\mathbf{u}}_I \tag{23}$$

where the strain matrix is

$$\mathbf{D}=\begin{bmatrix}\phi_{1,x} & 0 & \phi_{2,x} & 0 & \cdots & \phi_{n,x} & 0\\ 0 & \phi_{1,y} & 0 & \phi_{2,y} & \cdots & 0 & \phi_{n,y}\\ \phi_{1,y} & \phi_{1,x} & \phi_{2,y} & \phi_{2,x} & \cdots & \phi_{n,y} & \phi_{n,x}\end{bmatrix} \tag{24}$$

and $\phi_{k,i}$ is the derivative of shape function, it can be calculated by using the following equations:

$$\begin{cases}\frac{\partial\phi_k}{\partial x}=\sum_{i=1}^{n}\frac{\partial R_i}{\partial x}S_{ik}^a+\sum_{j=1}^{m}\frac{\partial p_j}{\partial x}\\ S_{jk}^b\frac{\partial\phi_k}{\partial y}=\sum_{i=1}^{n}\frac{\partial R_i}{\partial y}S_{ik}^a+\sum_{j=1}^{m}\frac{\partial p_j}{\partial y}S_{jk}^b\end{cases} \tag{25}$$

Using Eq. (23) in Eq. (9) leads to the following expression of the optimization problem:

$$\lambda=\min\int_V\sqrt{\dot{\mathbf{u}}_I^T\mathbf{D}^T\boldsymbol{\Theta}\mathbf{D}\dot{\mathbf{u}}_I}dV \text{ s.t. } \quad \int_V\mathbf{g}^T\boldsymbol{\Phi}\cdot\dot{\mathbf{u}}_I dV+\int_{S_\sigma}\mathbf{t}^T\boldsymbol{\Phi}\cdot\dot{\mathbf{u}}_I dS=1$$
$$\dot{\mathbf{u}}=\mathbf{0} \quad \text{on } S_u \tag{26}$$

Since the integrand functions of the objective function in Eq. (26) are numerically expressed based on the discretized field nodes, the integrations can be evaluated numerically based on these nodes. We shall employ the Cartesian Transformation Method (CTM) recently proposed Khosravifard and Hematiyan [31] to perform the numerical integration.

2.2.3. Cartesian transformation method for meshless integration

The main advantage of Cartesian Transformation Method (CTM) is that the domain integration can be exactly evaluated without any background integral cell. The CTM is based on the Green's theorem, which states that a domain integration $I=\int_\Omega f(x,y)\,d\Omega$ defined over a domain Ω can be transformed first into a boundary integration along the boundary Γ_R of the auxiliary domain Ω_R, and then a 1D integration defined over the domain Ω_R [31], i.e.,

$$I=\int_\Omega f(x,y)d\Omega=\int_{\Omega_R}g(x,y)d\Omega=\int_{\Gamma_R}\left(\int_c^x g(\xi,y)d\xi\right)d\Omega \tag{27}$$

where $g(x,y)$ is the auxiliary function which is defined as follows:

$$g(x,y)=\begin{cases}f(x,y), & (x,y)\in\Omega\\ 0, & (x,y)\notin\Omega\end{cases}$$

If a rectangular auxiliary domain is adopted, the formulation can be much simplified. Take the case of Fig. 4 for example. As for the shown original integration domain Ω and the corresponding rectangular auxiliary domain Ω_R, the boundary Γ_R of Ω_R consists of four line segments:

$$\Gamma_R=\Gamma_1\cup\Gamma_2\cup\Gamma_3\cup\Gamma_4 \tag{28}$$

where

$$\Gamma_1:\ y=y_1;\quad \Gamma_2:\ x=b;\quad \Gamma_3:\ y=y_2;\quad \Gamma_4:\ x=a$$

The integration in Eq. (27) can hence be expressed as follows:

$$I=\int_{y_1}^{y_2}\left(\int_a^b g(x,y)dx\right)dy \tag{29}$$

The two level 1D-integrations in Eq. (29) can be evaluated numerically using Gaussian quadrature method. To evaluate the outer boundary integration along the boundary Γ_2, Γ_2 is first divided into n_y intervals (the i-th interval $[y_i, y_{i+1}]$ is shown in Fig. 4 as an example). m_y-point Gaussian quadrature is then applied to each interval. As a result, the outer boundary integration can be expressed as follows:

$$I=\int_{y_1}^{y_2}h(y)dy=\sum_{l=1}^{n_y}\sum_{i=1}^{m_y}J_{il}^y w_i^y h(y(\eta_i)) \tag{30}$$

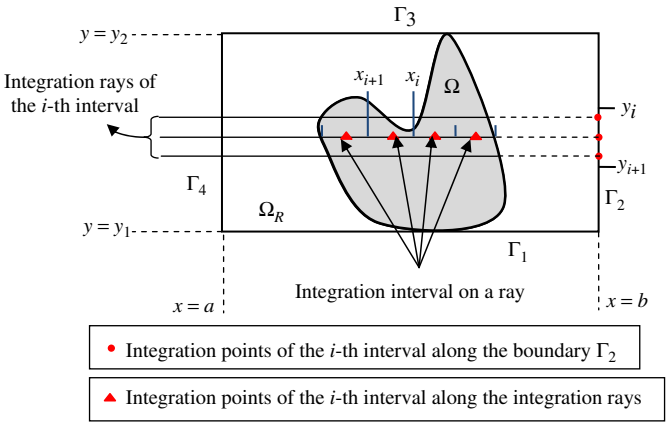

Fig. 4. Rectangular auxiliary integration domain and the integration rays and intervals.

where

$$h(y(\eta_i))=\int_a^b g(x,y(\eta_i))dx \tag{31}$$

and J_{il}^y is the Jacobian of the transformation from the l-th interval $[y_i, y_{i+1}]$ to $[-1, 1]$; η_i and w_i^y are the Gaussian points and weights, respectively. We need to evaluate $h(y)$ first in order to compute the integration in Eq. (30). Notice that $h(y)$ is also a 1D-integration which can also be evaluated by the Gaussian quadrature method. In Eq. (30), the function $y=y(\eta_i)$ denotes a horizontal line, which is called integration ray (Fig. 4). Suppose that an integration ray intersects with the boundary of a closed domain at an even number of times, i.e., $t=2l$ times, where t is the number of intersection points, and l is a positive integer. The intersections split the ray into $t-1$ segments. Accordingly, the integration in Eq. (31) can be expressed as

$$h(y(\eta_i))=\sum_{k=1}^{l}\int_{x_{2k-1}}^{x_{2k}}f(x,y(\eta_i))dx \tag{32}$$

The Gaussian quadrature can then be applied to the evaluation of Eq. (32) which has n_x interval and m_x integration points in each interval for a integration ray, i.e.,

$$\int_{x_{2k-1}}^{x_{2k}}f(x,y(\eta_i))dx=\sum_{r=1}^{n_x}\sum_{s=1}^{m_x}J_{sr}^x w_s^x f(x(\xi_s),y_i) \tag{33}$$

where J_{sr}^x is the Jacobian of the transformation from the r-th interval $[x_i, x_{i+1}]$ to [-1, 1]; ξ_s and w_s^x are the Gaussian points and weights, respectively.

In considering Eqs. (30) and (33), it is readily seen that the integration can be calculated in terms of the following formulation:

$$I=\mathbf{W}^{2D}\cdot\mathbf{F}=\sum_{i=1}^{n_G}W^{2D}(\mathbf{x}_i)\cdot f(\mathbf{x}_i) \tag{34}$$

where

$$\mathbf{F}=\begin{bmatrix}f_1(\mathbf{x}_i) & f_2(\mathbf{x}_i) & \cdots & f_{n_G}(\mathbf{x}_i)\end{bmatrix}^T \tag{35}$$

$$\mathbf{W}^{2D}=\begin{bmatrix}W^{2D}(\mathbf{x}_1) & W^{2D}(\mathbf{x}_2) & \cdots & W^{2D}(\mathbf{x}_{n_G})\end{bmatrix}^T \tag{36}$$

$$W^{2D}(\mathbf{x}_i)=J_i^x\cdot J_i^y\cdot w_i^x\cdot w_i^y \tag{37}$$

where $\boldsymbol{x}_i=[x_i,y_i]$ are the Gaussian points, $i=1,\ldots,n_G$, and n_G is the number of the Gaussian points in the solution domain. For the case in Fig. 4, $n_G=n_x\cdot m_x\cdot n_y\cdot m_y$. J_i^x and J_i^y are the corresponding Jacobian, respectively, for the intervals along x- and y-direction.

w_i^x and w_i^y are weights associated with the integration points of the intervals.

According to the CTM integration formulation (34), the objective function in Eq. (26) can be expressed as

$$\int_{\Omega_p} \sqrt{\mathbf{U}^T\mathbf{K}_i\mathbf{U}}d\Omega = \sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\sqrt{\mathbf{U}^T\mathbf{K}_i\mathbf{U}} \tag{38}$$

where $\mathbf{U}=\begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_N\end{bmatrix}^T$, N is the number of nodes, $\mathbf{K}_i=\mathbf{C}_e^T\mathbf{D}^T\boldsymbol{\Theta}\mathbf{D}\mathbf{C}_e$, $\boldsymbol{C}_e$ is the transformation matrix from the nodal velocity vector $\boldsymbol{u}_I$ to the global nodal velocity vector $\boldsymbol{U}$, i.e.

$$\mathbf{u}_I=\mathbf{C}_e\cdot\mathbf{U} \tag{39}$$

On the other hand, the integration appearing in the constraints of Eq. (26) can be calculated by using 1-D Gaussian quadrature method, and the nodal traction force vector $\boldsymbol{T}$ can be expressed as follows (the gravity is omitted here):

$$\mathbf{T}^T=\sum_{l}^{n_{ct}}\sum_{i=1}^{n_{Gt}} J_{il}w(\mathbf{x}_i)\cdot\mathbf{t}(\mathbf{x}_i)^T\boldsymbol{\Phi}\cdot\mathbf{C}_e \tag{40}$$

where n_{ct} is the number of integral interval on the traction boundary; J_{il} is the Jacobian of the l-th integral interval; $w(\boldsymbol{x}_i)$ is the Gauss weight for the i-th Gaussian point. Consequently, the finally numerical formulation of upper bound approach based on RPIM and CTM can be recast as

$$\begin{aligned}&\lambda=\min\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\sqrt{\mathbf{U}^T\mathbf{K}_i\mathbf{U}}\\ &\text{s.t.}\quad \mathbf{T}^T\cdot\mathbf{U}=1\\ &\quad\mathbf{u}=0\quad \text{on } S_u\end{aligned} \tag{41}$$

Notably, however, the plastically incompressible condition has not been enforced in the above numerical formulation which may result in a non-optimal load factor to be found for a plain strain problem. The incompressible condition needs therefore to be carefully considered.

2.2.4. Plastically incompressible condition

Nagtegaal et al. [57] are among the first who recognized the important effect of plastic incompressibility condition on the overall solution procedure. In order for a limit load to exist for the discretized model of an elastic–plastic problem, it is necessary that the volumetric strain rate $\dot{\varepsilon}_v=\dot{\varepsilon}_{kk}=0$ pointwise throughout the entire plastic zones. Otherwise, no limit load can be found. In limit analysis, the penalty function method is commonly used to enforce the incompressibility conditions [47]:

$$\frac{1}{2}\alpha\int_v \dot{\varepsilon}_v^2 dV=\frac{1}{2}\alpha\int_v \dot{\boldsymbol{\varepsilon}}^T\boldsymbol{\Lambda}\dot{\boldsymbol{\varepsilon}}dV \tag{42}$$

By discretizing the domain into a set of suitable field nodes, the integration in Eq. (42) can be calculated in terms of CTM, i.e.,

$$\frac{1}{2}\alpha\int_v \dot{\varepsilon}_v^2 dV=\frac{1}{2}\alpha\int_v \dot{\mathbf{u}}_I^T\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}\dot{\mathbf{u}}_I dV=\frac{1}{2}\alpha\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\dot{\mathbf{u}}_I^T\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}\dot{\mathbf{u}}_I \tag{43}$$

By considering the transformation matrix $\boldsymbol{C}_e$, the penalty function item can reformulated as

$$\begin{aligned}\frac{1}{2}\alpha\int_v \dot{\varepsilon}_v^2 dV&=\frac{1}{2}\alpha\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\mathbf{U}^T\mathbf{C}_e^T\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}\mathbf{C}_e\mathbf{U}\\ &=\frac{1}{2}\alpha\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\mathbf{U}^T(\mathbf{K}_v)_i\mathbf{U}\end{aligned} \tag{44}$$

where

$$(\mathbf{K}_v)_i=\mathbf{C}_e^T\mathbf{D}^T\boldsymbol{\Lambda}\mathbf{D}\mathbf{C}_e$$

As a result, Eq. (26) can be finally expressed as follows:

$$\begin{aligned}&\lambda=\min\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\sqrt{\mathbf{U}^T\mathbf{K}_i\mathbf{U}}\\ &\text{s.t.}\quad \mathbf{T}^T\cdot\mathbf{U}=1\\ &\quad\mathbf{U}^T(\mathbf{K}_v)_i\mathbf{U}=0\quad i=1,\ldots,n_G\\ &\quad\mathbf{u}=0\quad \text{on}S_u\end{aligned} \tag{45}$$

In Eq. (45), the second constrained equation denotes the penalty term for the impressibility condition, and the third constrained equation denotes the displacement velocity boundary condition which is equivalent to the essential boundary condition of linear elastic analysis. In consideration of the property of Kronecker delta function in the shape function of RPIM, the third constrained equation will be enforced in a direct way similarly as in FEM instead of invoking special treatments. In essence, the original matrix $\mathbf{K}_i$ and $(\mathbf{K}_v)_i$ for the i-th component of the prescribed velocity can be modified as below:

$$\mathbf{K}_i(\text{or }(\mathbf{K}_v)_i)=\begin{bmatrix} K_{11} & \cdots & K_{1(i-1)} & 0 & K_{1(i+1)} & \cdots & K_{1(2N)}\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ K_{(i-1)1} & \cdots & K_{(i-1)(i-1)} & 0 & K_{(i-1)(i+1)} & \cdots & K_{(i-1)(2N)}\\ 0 & \vdots & \vdots & 0 & 0 & \vdots & 0\\ K_{(i+1)1} & \cdots & K_{(i+1)(i-1)} & 0 & K_{(i+1)(i+1)} & \cdots & K_{(i+1)(2N)}\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ K_{(2N)1} & \cdots & K_{(2N)(i-1)} & 0 & K_{(2N)(i+1)} & \cdots & K_{(2N)(2N)}\end{bmatrix} \tag{46}$$

Consequently, Eq. (45) can be expressed as a discretized NLP problem subjected to two equality constrains, and will be solved by the direct iterative method discussed subsequently.

3. Direct iterative method

It is noticed that there is a term of square root in the constrained optimization problem (45) which could render the objective function unsmooth and non-differentiable when the term $\mathbf{U}^T\mathbf{K}_i\mathbf{U}$ at some Gaussian integration points is equal to zero. This may cause numerical difficulties in solving the NLP problem. To resolve this issue, we employ here the direct iterative method widely used in recent studies (see, [72,47]; [7]; [42,43]). This method solves the optimization problem iteratively. In each iteration step, the rigid regions in the domain are distinguished from the plastic regions and the objective and constraint equations are revised accordingly. Since the objective function is non-differentiable because the plastic strain rate is vanishing within the rigid region, it is treated as a constraint and is taken into accounted in the optimization problem based on penalty method. Eq. (45) can then be transformed a differentiable optimization problem which may be solved by Lagrangian multiplier and penalty function method. An iterative control parameter ω is defined below to judge which Gaussian integration point is in the plastic or rigid region:

$$\omega=\sqrt{\mathbf{U}^T\mathbf{K}_i\mathbf{U}} \tag{47}$$

If $\omega=0$ at a Gaussian integration point, this point is regarded in the rigid region; Otherwise, it is located in the plastic region. Accordingly, the NLP problem (45) can be reformulated as

$$\begin{aligned}&\lambda=\min\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\frac{\mathbf{U}^T\mathbf{K}_i\mathbf{U}}{\omega}\\ &\text{s.t.}\quad \mathbf{T}^T\cdot\mathbf{U}=1\\ &\quad\mathbf{U}^T(\mathbf{K}_v)_i\mathbf{U}=0,\quad i\in S\end{aligned} \tag{48}$$

Prior to an iteration, an initial velocity field needs to be specified. However, according to Liu et al. [47] and Li and Yu [42,43], the initial trial solution will not affect the convergence of iteration.

For simplicity, we start iteration by setting $\omega=1$ in this paper. The initial velocity field $\boldsymbol{U}_0$ is computed by solving the following Lagrangian optimization problem:

$$L(\mathbf{U},\mu)=\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\mathbf{U}^T\mathbf{K}_i\mathbf{U}+\alpha\sum_{i=1}^{n_G}\mathbf{U}^T(\mathbf{K}_\nu)_i\mathbf{U}+2\mu(1-\mathbf{T}^T\cdot\mathbf{U}) \quad (49)$$

where μ is the Lagrangian multiplier. Using the minimum optimization conditions, $\partial L/\partial\mathbf{U}=\mathbf{0}$, $\partial L/\partial\mu=0$, the following system of linear equations can be obtained:

$$\begin{cases}\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\mathbf{K}_i\mathbf{U}_0+\sum_{i=1}^{n_G}\alpha_0(\mathbf{K}_\nu)_i\mathbf{U}_0=\mu_0\mathbf{T}\\ \mathbf{T}^T\mathbf{U}_0=1\end{cases} \quad (50)$$

where μ_0 is the initial Lagrangian multiplier and α_0 is the initial penalty factor. The initial nodal displacement velocity $\mathbf{U}_0$ can then be determined, by which the initial load multiplier λ_0 can be calculated:

$$\lambda_0=\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\sqrt{\mathbf{U}_0^T\mathbf{K}_i\mathbf{U}_0} \quad (51)$$

Based on the initial load multiplier λ_0 and the initial nodal displacement velocity $\mathbf{U}_0$, the non-differentiable areas can be distinguished, and the objective function can then be evaluated iteratively.

Based on the results at Step k (k=0, 1, 2,…), the value of ω needs to be calculated at each Gaussian integration point. The Gaussian integration point set, S, will then be subdivided into two subsets: a subset S_r^{k+1} where the object function is not differentiable, and the other subset S_p^{k+1} where the object function is, i.e.

$$S_r^{k+1}=\{i\in S,\quad \omega=0\};\quad S_p^{k+1}=\{i\in S,\quad \omega\neq 0\} \quad (52)$$

The purpose of distinguishing the two subsets S_r^{k+1} and S_p^{k+1} is to remove those in the rigid region from all the integration points so as to ensure that the next iterative step can be carried out smoothly. For $\omega=0$, the constraint condition is imposed on the rigid region Gaussian points as follows:

$$\mathbf{U}_{k+1}^T\mathbf{K}_i\mathbf{U}_{k+1}=0\quad (i\in S_r^{k+1}) \quad (53)$$

At points in S_r^{k+1}, the plastically incompressible condition can be satisfied naturally, while at points in S_p^{k+1}, the following constraint needs to be imposed to satisfy the incompressible condition:

$$\mathbf{U}_{k+1}^T(\mathbf{K}_\nu)_i\mathbf{U}_{k+1}=0\quad (i\in S_p^{k+1}) \quad (54)$$

Therefore, at Step $k+1$, the original optimization problem can be transformed as follows:

$$\begin{aligned}&\min_{\mathbf{U}}\sum_{i\in S_p^{k+1}} W^{2D}(\mathbf{x}_i)\cdot\frac{\mathbf{U}_{k+1}^T\mathbf{K}_i\mathbf{U}_{k+1}}{\sqrt{\mathbf{U}_k^T\mathbf{K}_i\mathbf{U}_k}}\\ &\text{s.t. }\mathbf{T}^T\mathbf{U}_{k+1}=1\\ &\mathbf{U}_{k+1}^T(\mathbf{K}_\nu)_i\mathbf{U}_{k+1}=0\quad (i\in S_p^{k+1})\\ &\mathbf{U}_{k+1}^T\mathbf{K}_i\mathbf{U}_{k+1}=0\quad (i\in S_r^{k+1})\end{aligned} \quad (55)$$

To determine the velocity field and the limit load multiplier at Step $k+1$, the revised optimization problem (55) can also be solved by using Lagrangian multiplier method and penalty function method for the following system of linear equations:

$$\begin{cases}\sum_{i\in S_p^{k+1}} W^{2D}(\mathbf{x}_i)\cdot\frac{\mathbf{K}_i\mathbf{U}_{k+1}}{\sqrt{\mathbf{U}_k^T\mathbf{K}_i\mathbf{U}_k}}+\sum_{i\in S_r^{k+1}}\beta_{k+1}\mathbf{K}_i\mathbf{U}_{k+1}+\sum_{i\in S_p^{k+1}}\alpha_{k+1}(\mathbf{K}_\nu)_i\mathbf{U}_{k+1}=\mu_{k+1}\mathbf{T}\\ \mathbf{T}^T\mathbf{U}_{k+1}=1\end{cases} \quad (56)$$

where α_{k+1} and β_{k+1} are the penalty factors at the $(k+1)$-th step, and μ_{k+1} is the Lagrangian multiplier at the $(k+1)$-th step.

Table 2
Comparison of the size of K_i and K_{vi} for different discretizing methods.

Matrixes	RPIM	FEM	
		Linear triangular element (three nodes)	Quadratic triangular element (six nodes)
$\mathbf{K}_i$	$2N\times 2N$	$6m\times 6m$	$12m\times 12m$
$\mathbf{K}_{vi}$	$2N\times 2N$	$6m\times 6m$	$12m\times 12m$

Consequently, the nodal velocity $\boldsymbol{U}_{k+1}$ and the limit load multiplier at this step can be obtained:

$$\lambda_{k+1}=\sum_{i=1}^{n_G} W^{2D}(\mathbf{x}_i)\cdot\sqrt{\mathbf{U}_{k+1}^T\mathbf{K}_i\mathbf{U}_{k+1}} \quad (57)$$

The above iterative process is repeated until the following convergence criteria are satisfied:

$$\frac{|\lambda_{k+1}-\lambda_k|}{\lambda_{k+1}}\le\eta_1 \text{ and } \frac{||\mathbf{U}_{k+1}-\mathbf{U}_k||}{\mathbf{U}_{k+1}}\le\eta_2 \quad (58)$$

where η_1 and η_2 are prescribed error tolerances. According to our numerical experience, it generally works well for both to be chosen in the range $\eta_1,\eta_2\in[10^{-3}, 10^{-4}]$. It is noteworthy that the convergence of this direct iterative algorithm for von Mises yield criterion has been proved by Zhang et al. [72], with some further applications with finite element (see, [73,47,43]).

At each iterative step, the current limit load multiplier and velocity field can be found by solving a system of linear equations. The computational costs of these direct iterative methods are thus proportional to the sizes of matrix $\mathbf{K}_i$ and $\mathbf{K}_{vi}$ in the system of linear Eq. (56), while the size of both matrices is related to the number of nodes N for mesh-free methods or the number of elements m for finite element method, which is shown in Table 2.

To compare the different size of matrix for $\mathbf{K}_i$ and $\mathbf{K}_{vi}$ using RPIM and FEM, we assumed that the solution domain is discretized by a given numbers of nodes for RPIM, and the mesh of triangulation for FEM is then generated based on these nodes. As a result, there is a topological relationship between the number of nodes N in RPIM and that of elements m in FEM. And this relationship can be described by Euler's formula, i.e.,

$$N-N_e+N_f=2 \quad (59)$$

where N denotes the number of nodes in triangulation mesh; $N_e=(3m+N_k)/2$ denotes the total number of edges; and N_f denotes the number of faces of triangulation. Based on the Eq. (59), we can derive the following relationship between the number of elements and nodes:

$$m=2N-2-N_k \quad (60)$$

where N_k is the number of nodes that lie on the boundary of the mesh. For large scale problems with $N\gg N_k$, Eq. (60) can be simplified as

$$m\approx 2N \quad (61)$$

Evidently, from Eq. (61) and Table 2, the size of matrix $\mathbf{K}_i$ (or $\mathbf{K}_{vi}$) in FEM is six times (for linear triangular element) or twelve times (for quadratic triangular element) of the size of matrix in RPIM. Though not exactly proportional, for a fixed memory of CPU, it is generally true that solving a larger size matrix will cost more computational effort. Nevertheless, it indicates that the computational cost using the direct iterative method can be reduced by using RPIM mesh-free method.

4. Examples

In this section, the performance of the present numerical procedure of upper bound limit analysis will be illustrated with two classical problems. All implementation was performed on a personal computer with an Intel (R) Core (TM) 2 Quad CPU 2.83 GHz processor.

4.1. Thick-walled cylinder subjected to internal pressure

Structural components in engineering such as pipelines or pressure vessels can generally be simplified as a plain strain problem of thick-walled cylinder subjected to internal pressure. To determine the carrying capacity of these structural components, we wish to find the limit internal pressure the thick-walled cylinder can sustain. If the cylinder material can be assumed to perfect plastic materials following von Mises yield criterion, an exact limit solution of the problem is available given by the following expression (see, e.g., [28]):

$$\lambda_{\lim} = \frac{p_s}{\sigma_s} = \frac{2}{\sqrt{3}} \ln r \tag{62}$$

where $r = R_2/R_1$ is the ratio between the internal radius and the external radius, σ_s denotes the yield stress. p_s is the limit pressure load and $\lambda_{\lim}$ is the limit load multiplier. We shall try to solve the same problem using the proposed method described in the foregoing sections. Specifically, the upper bound optimization problem in Eq. (45) is evaluated to find the limit load multiplier $\lambda_{\lim}$ of this problem.

To solve the optimization problem (45), the domain is first discretized by a set of field nodes. Due to symmetry, only the quadrant of the cylinder shown in Fig. 5a is considered. The field nodes can be generated randomly within the considered quadrant, while the Gaussian integration points need to be generated by satisfying the demand of CTM integral method (Fig. 5b). No background cell is needed any more for the numerical integration. After the generation of field nodes and Gaussian integration points, the kinematically admissible velocity at each integration point can be interpolated by using RPIM shape function. In choosing the radial basis function, both types of radial basis functions in Table 1 will be used here to construct the RPIM shape function for the purpose of comparison. For the MQ function we adopt a value of 4 for α_c and 0.5 for q; and for the EXP function, α_c is set to be 4. The limit load multipliers are then calculated according the proposed methods.

We first compare the efficiency using the two radial basis functions mentioned in Section 2. Presented in Fig. 6 are the convergence sequence of computation of the limit load by using MQ and EXP functions for two cases of r (a) $r=2$ and (b) $r=4$. It is worth noting that there is no rigid region in the whole cylinder when the internal pressure reaches the limit load. Consequently, in our method, no rigid regions can be distinguished from the plastic regions during the iteration process. The convergence criteria (58) should in theory be satisfied easily. Due to numerical

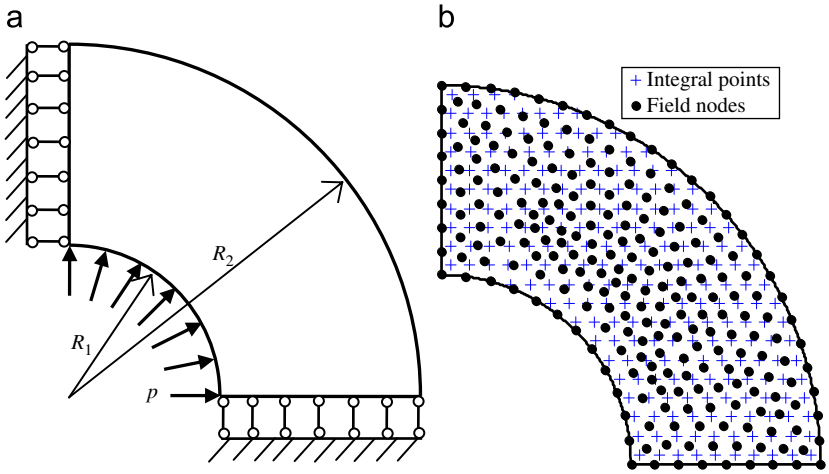

Fig. 5. Illustration of a quadrant of thick-walled cylinder subjected to a uniform internal pressure: (a) the geometrical model; (b) a typical discretization of nodes and integration points.

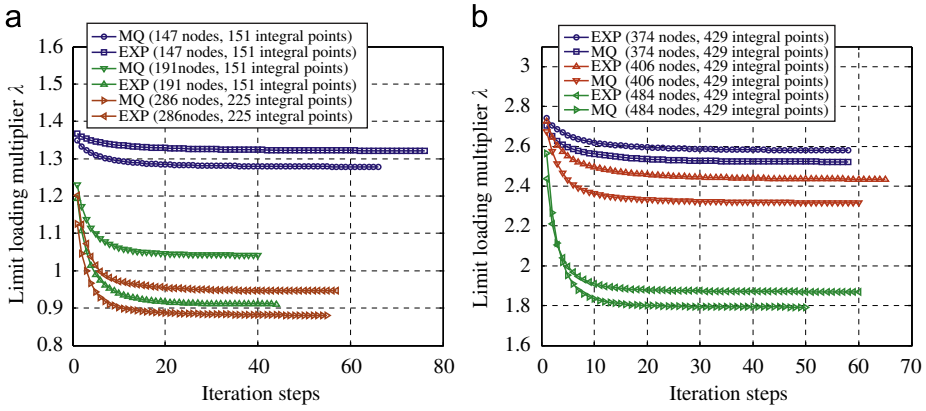

Fig. 6. Convergence of limit load multiplier by the presented method for different ratio: (a) $r=2$; (b) $r=4$.

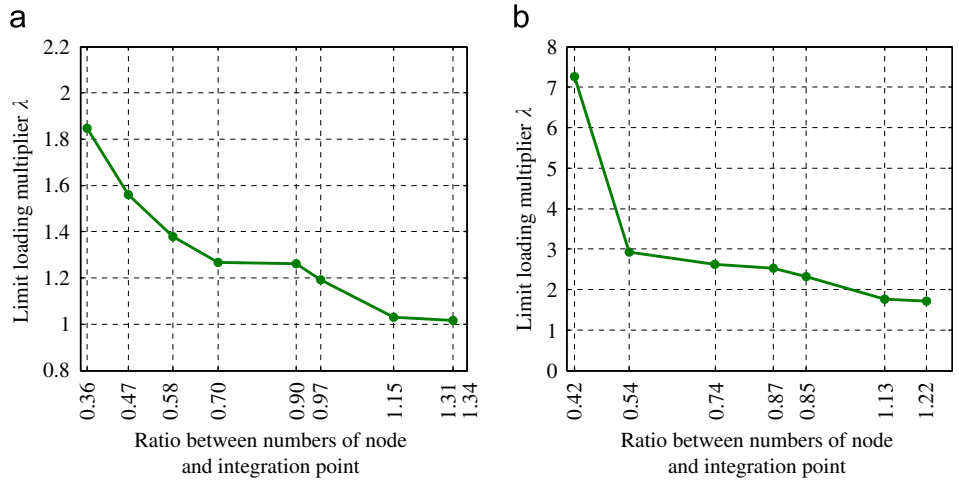

Fig. 7. Limit load multipliers with different numbers of node and integration point for different ratio: (a) $r=2$; (b) $r=4$.

errors in practical calculations, however, it still requires several iteration steps to satisfy the conditions of (58). Indeed, our computations indicate that the iteration will normally converge within 3 steps if only the first condition of (58) is considered. However, if both convergence conditions of (58) are considered, significantly more iterations (normally around 20) are needed to achieve convergence. Note that for the two cases of cylinder, the exact solutions of limit load multiplier, according to Eq. (62), are 0.8 for case (a) and 1.601 for case (b), respectively. We find for both types of radial basis function, using more nodes will generally lead to improved upper bounds (the influence of integration point number will be discussed later). Meanwhile, with same number of nodes and integration points, the limit load multiplier obtained by MQ-RBF appears to be generally smaller than that by the EXP-RBF and thus closer to the exact solution. Use of MQ-RBF is likely to offer higher accuracy than EXP-RBF in our method.

It is evident form Fig. 6 that the number of field nodes and integration points has a significant effect on the numerical solutions. A further study on this effect is made and the results are presented in Fig. 7. As is shown, with the increase of node number, the numerical limit solution consistently converges to a value close to the true one. However, this depends crucially on a properly chosen number of integration points. At certain number of nodes, it is not always true that the more integration points are used, the better solution is obtained. There is an optimal ratio between the number of nodes and that of the integration points which leads to more accurate solutions. For the present problem, our numerical experience indicates that this ratio should be around 1.2∼1.3, i.e., $a=n/n_g=1.2\sim1.3$.

However, it is observable from Figs. 6 and 7 the numerical solutions are still different considerably from the exact solutions. Further improvement on the accuracy can be achieved by fine-tuning the shape parameters. Take MQ-RBF as an example. By adjusting the value of α_c and q in pair within the range of [0, 10] and [0.1, 2.1] respectively, we have generated a contour of relative error for the limit load multiplier compared to the exact solution for the case of $r=2$. The results are presented in Fig. 8. As can be seen from Fig. 8, when $\alpha_c=8$ and $q=0.7$, the numerical upper bound reaches a relative error of 0.03% with the exact solution. We hereby choose this pair of values for α_c and q as optimal ones for the case of $r=2$. We have also investigated two other cases of r. It is found minor adjustment ones α_c and q around their optimal values may lead to satisfactory solutions. Summarized in Table 3 are the limit load multipliers calculated by the proposed methods for $r=2,3,4$. Comparison has also been

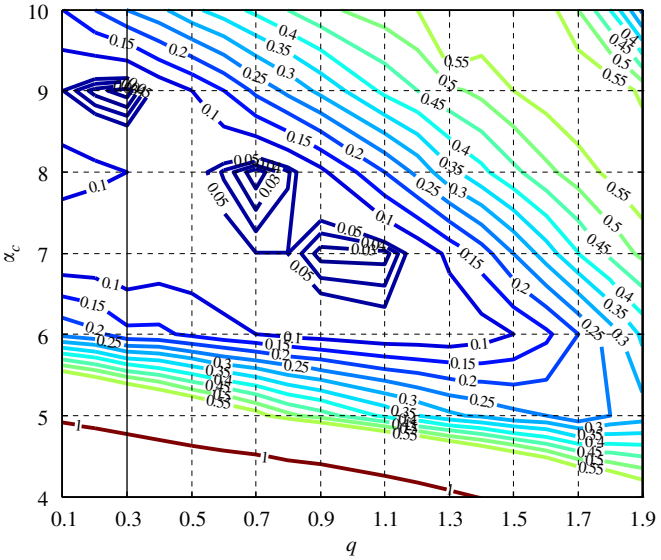

Fig. 8. Use of contour of relative error of limit load multiplier to select optimal value for α_c and q in MQ-RBF ($r=2$).

Table 3
Summary of limit loads of the extension pressure for the thick-walled cylinder using different methods.

$r=R_2/R_1$	Lower bound solution by Chen et al. [13,14]	Analytical solution	Present upper bound solution	Relative error[a] (%)
2	0.798	0.800	0.8002	0.03
3	1.265	1.269	1.270	0.08
4	1.596	1.601	1.602	0.06

[a] The present upper bound solution relative to the analytical solution.

made to the analytical solutions as well as the lower bound solutions obtained by Chen [16]. From the relative error of the present upper bound solution with respect to the analytical solution, the proposed upper bound has approach appears to offer relatively good accuracy.

4.2. Thin square plates subjected to uniaxial tension

Using the proposed method, we have also investigated the limit load of thin square plate with a square or circular hole subjected to uniaxial tension (see Fig. 9). The plate is simplified to

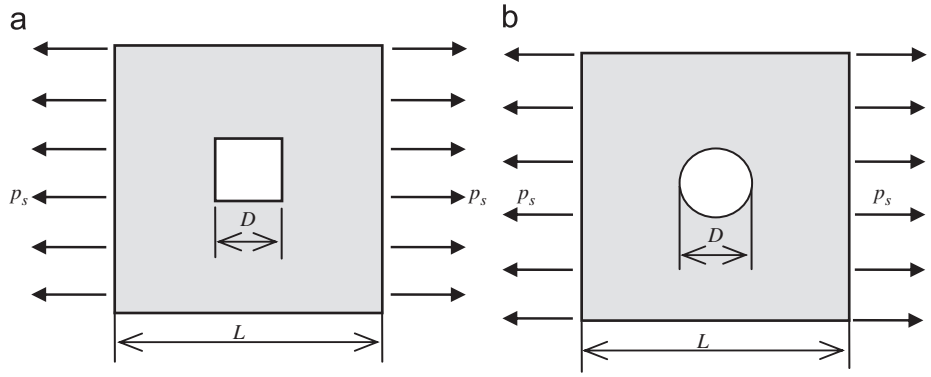

Fig. 9. A plate with a hole subject to uniaxial tension: (a) a square hole; (b) a circular hole.

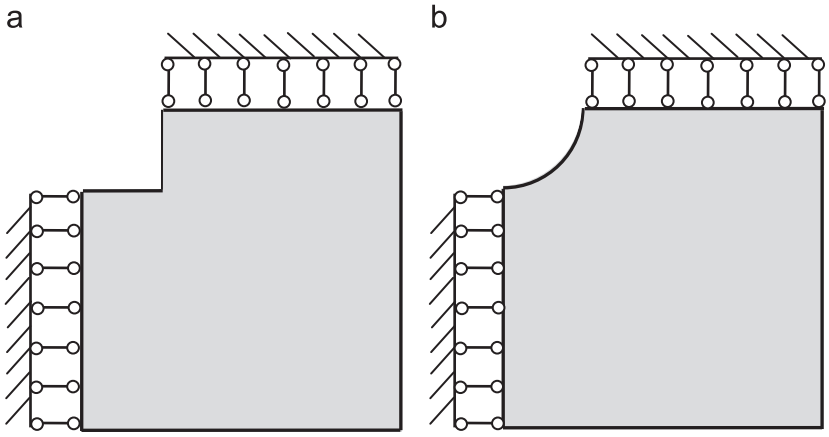

Fig. 10. Considered domain and prescribed boundary conditions for the plate (a) with square hole; (b) with circular hole.

be a plane-stress problem and the material is assumed to follow von Mises yield condition. Due to symmetry, only the bottom right quadrant part of the plate is considered. See Fig. 10 for the considered domain and the prescribed boundary conditions. Fig. 11 depicts the discritization of nodes and integration points in the considered domain for the two cases.

For the circular hole case of plate subjected to tension stress as shown in Fig. 9(b), there is an analytical solution available for the limit load if the material is assumed to be von Mises type [24]. For a ratio of D/L ranged from 0 to 0.2, the exact solution can be calculated by

$$\lambda_{\lim} = \frac{p_s}{\sigma_s} = \left(1 - \frac{D}{L}\right)$$

Readily we see the exact limit load multiplier is 0.8 when $D/L=0.2$. while for the square hole case, we compare our results with the upper bound solution obtained by Zhang et al. [72] for a case of $D/L=0.25$.

When the present upper bound method is applied to the plate problem, we still need to select optimal MQ-RBF parameters and the optimal number ratio between nodes and integration points. For case (a) with a square hole and ($D/L=0.25$), we found $\alpha_c=17$ and $q=2.5$ appear to offer good accuracy and convergence of our computation, and for case (b) with a circular hole ($D/L=0.2$), the optimal parameters are $\alpha_c=8$ and $q=2.7$. Refer to the contour of relative error in Fig. 12, where the relative error for the square hole case has been calculated relative to the solution by Zhang et al. [72], and the circular case relative to the solution by Gaydon and McCrum [24]. Note that $q=1$, 2 and 3 happen to lead to singular solutions for the RPIM and should therefore avoided to be chosen for q [44]. As for the ratio of nodes/integration points, we found from Fig. 13 that for both cases of plate, the optimal ratio is around 1.2.

We summarize in Table 4 the limit load calculated by the present method for the circular hole plate cases when $D/L=0.2$ and further compare it to results obtained by other numerical/analytical approach including lower bound limit analysis. The present upper bound approach appears to be accurate enough to lead to a limit load very close to the analytical solution among the various results. It should also be mentioned that all our computations for the upper bound limit can be completed within 3 min on a general personal computer, which shows that it is fairly efficient as well.

5. Conclusions

We have presented an upper bound limit analysis based on the meshless method. By using RPIM to construct the kinematically admissible velocity field, the key issue on the enforcement of essential boundary conditions in meshless method is resolved. Meanwhile, to avoid the use of any background cell for integration, a new integration method based on Cartesian transformation method (CTM) has been employed to compute the internal dissipation power. The numerical optimization problem of finding the upper bound limit load is then solved by an efficient direct iterative method with nonlinear programming. The proposed method has been applied to the analysis of two classical upper bound problems. A number of conclusions can be drawn from the study:

(1) The RPIM method can be used to construct the kinematically admissible velocity field to achieve relatively high accuracy, and the essential boundary conditions can be directly

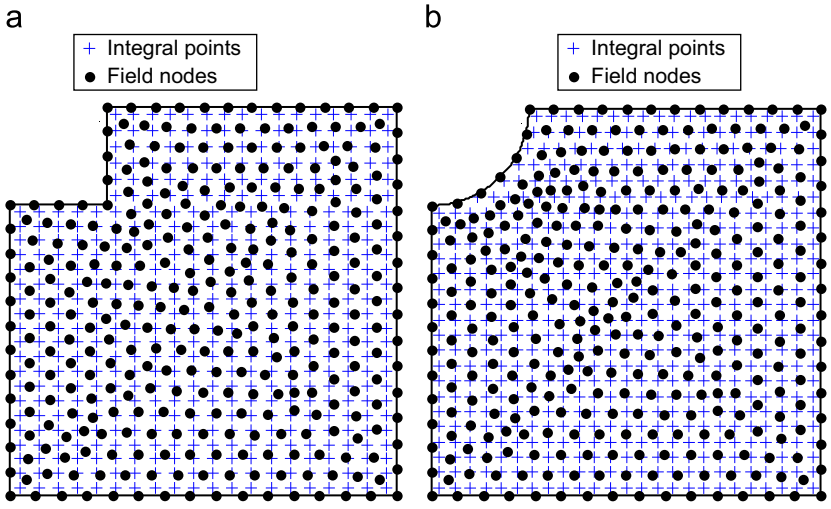

Fig. 11. Discretization in the considered domain of the plate: (a) the square hole case; (b) the circular hole case.

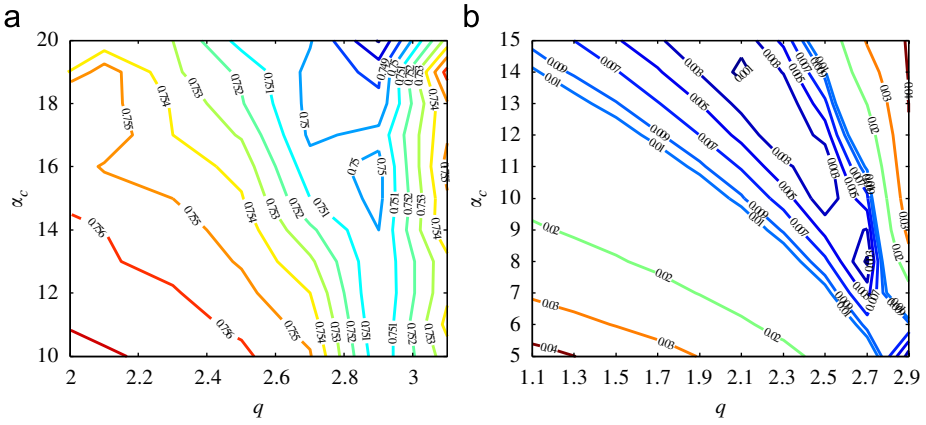

Fig. 12. Contour of relative error for limit load multiplier with different MQ-RBF parameters α_c and q: (a) the square hole case; (b) the circular hole case.

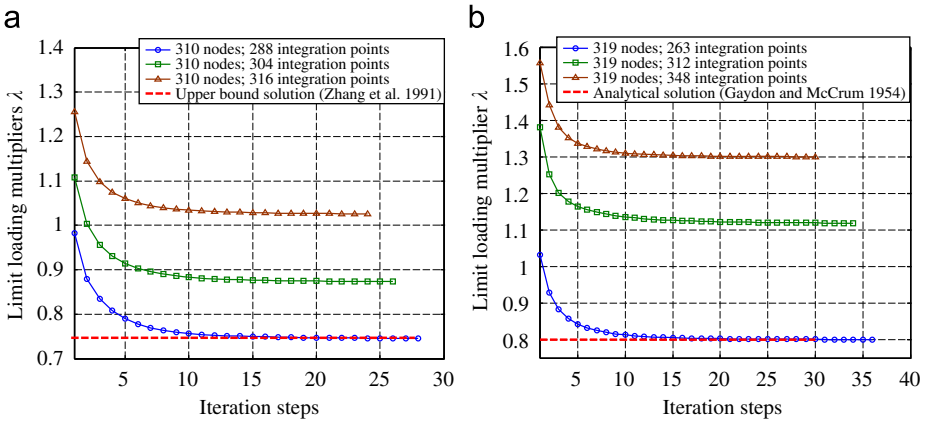

Fig. 13. Convergence of limit load multipliers and determination of optimal ratio between node number and integration point number for (a) the square hole case; (b) the circular hole case.

enforced similarly as that in finite element method. Meanwhile, the computational cost can be also reduced by using the RPIM method.

(2) Employment of Cartesian transformation method (CTM) in the meshless method for upper bound limit analysis can conveniently transform the domain integration into boundary integration and 1D integration, and facilitate the computation of integration. However, the number of integration points is not independent of the number of field nodes. An optimal ratio between the number of nodes and that of the integration points exist which may potentially lead to improved solutions.

(3) On the selection of radial basis function, the MQ method appears to provide higher accuracy than the EXP method. On choosing the shape parameter(s), using contour of relative error may be a feasible way.

While the present paper has assumed a small deformation analysis, upper bound limit analysis may likely lead to plastic

Table 4
Limit loads obtained by different methods for the circular-holed plate ($D/L=0.2$).

Methods	Specific approach	$\lambda_{\lim}$
Upper bound approach	Present upper bound method	0.801
	Yan et al. [69]	0.802
	Vicente da Silva and Antão [66]	0.807
	Le et al. [37]	0.801
Lower bound approach	Belytschko and Hodge (1970)	0.793
	Tin-Loi and Ngo [65]	0.803
	Liu et al. [48]	0.795
	Chen et al. [16]	0.798
Analytical solution	Gaydon and McCrum [24]	0.800

collapse problems where large deformation is important. Accordingly, the inherent problem of particle dislocation in a meshfree approach needs to be carefully addressed. To deal with large deformation problems associated with meshfree methods, an updating process named "Sequential Limit Analysis" can be employed to form a sequence leading to the solution of a large deformation problem (see e.g. [70,21,39]). This will be pursued in a future work.

Acknowledgment

The work was supported by Research Grants Council of Hong Kong (under grant No. 623609) and HKUST Postdoctoral Fellowship Matching Fund.

Reference

[1] Anderheggen E, Knöpfel H. Finite element limit analysis using linear programming. Int J Solids Struct 1972;8:1413–31.
[2] Andersen KD, Christiansen E, Overton ML. Computing limit loads by minimizing a sum of norms. SIAM J Sci Comput 1998;19(3):1046–52.
[3] Andersen KD, Christiansen E, Conn AR, Overton ML. An efficient primal-dual interior-point method for minimizing a sum of Euclidean norms. SIAM J Sci Comput 2000;22(1):243–62.
[4] Beissel S, Belytschko T. Nodal integration of the element-free Galerkin method. Comput Methods Appl Mech Eng 1996;139:49–74.
[6] Bottero A, Negre R, Pastor J, Turgeman S. Finite element method and limit analysis theory for soil mechanics problems. Comput Meth Appl Mech Eng 1980;22:131–49.
[7] Capsoni A, Corradi L. A finite element formulation of the rigid-plastic limit analysis problem. Int J Numer Methods Eng 1997;43:2063–86.
[8] Capsoni A, Corradi L, Vena P. Limit analysis of orthotropic structures based on Hill's yield condtion. Int J Solids Struct 2001;38:3945–63.
[9] Capsoni A. A mixed finite element model for plane strain limit analysis computations. Commun Numer Meth Eng 1999;15:101–12.
[10] Chaaba A, Bousshine L, De Saxce G. Kinematic limit analysis modeling by a regularization approach and finite element method. Int J Numer Meth Eng 2003;57:1899–922.
[11] Chen JS, Yoon S, Wu CT. Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. Int J Numer Meth Eng 2002;53:2587–615.
[12] Chen JS, Wu CT, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin mesh-free method. Int J Numer Meth Eng 2001;50:435–66.
[13] Chen S, Liu Y, Cen Z. Lower-bound limit analysis by using the EFG method and non-linear programming. Int J Numer Mech Eng 2008;74:391–415.
[14] Chen S, Liu Y, Cen Z. Lower-bound shakedown analysis by using the element free Galerkin method and nonlinear programming. Comput Meth Appl Mech Eng 2008;197:3911–21.
[15] Chen S, Liu Y, Li J, Cen Z. Performance of the MLPG method for static shakedown analysis for bounded kinematic hardening structure. Eur J Mech-A/Solids 2011;30:183–94.
[16] Chen WF. Limit Analysis and Soil Plasticity. New York: Elsevier Scientific Publishing Co.; 1975.
[17] Christiansen E. Computation of limit loads. Int J Numer Meth Eng 1981;17:1547–70.
[18] Christiansen E, Andersen KD. Computation of collapse states with von Mises type yield condition. Int J Numer Meth Eng 1999;46:1185–202.
[19] Ciria H, Peraire J, Bonet J. Mesh adaptive computation of upper and lower bounds in limit analysis. Int J Numer Meth Eng 2008;75:899–944.
[20] Cohn MZ, Maier G. Engineering plasticity by mathematical programming. In: Proceedings of the NATO advanced study INSTITUTE, Ontario, Canada, 1977.
[21] Corradi L, Panzeri N. A triangular finite element for sequential limit analysis of shells. Adv Eng Softw 2004;35:633–43.
[23] Gaudrat VFA. Newton type algorithm for plastic limit analysis. Comput Meth Appl Mech Eng 1991;88:207–24.
[24] Gaydon FA, McCrum AW. A theoretical investigation of the yield-point loading of a square plate with a central circular hole. J Mech Phys Solids 1954;2:156–69.
[25] Golberg MA, Chen CS. Some recent results and proposals for the use of radial basis functions in the BEM. Eng Anal Bound Element 1999;23:285–96.
[26] Hardy RL. Theory and applications of the multiquadrics – Biharmonic method (20 years of discovery 1968-1988). Comput Math Appl 1990;19(8/9):163–208.
[28] Hill R. The mathematical theory of plasticity. Oxford University Press; 1950 97-127.
[29] Huh H, Yang WH. A general algorithm for limit solutions of plane stress problems. Int J Solids Struct 1991;28(6):727–38.
[30] Jiang GL. Non-linear finite element formulation of kinematic limit analysis. Int J Numer Meth Eng 1995;38:2775–807.
[31] Khosravifard A, Hematiyan MR. A new method for meshless integration in 2D and 3D Galerkin meshfree methods. Eng Anal Bound Elements 2010;34:30–40.
[32] Krabbenhoft K, Damkilde L. A general non-linear optimization algorithm for lower bound limit anlaysis. Int J Numer Meth Eng 2003;56:165–84.
[34] Krabbenhoft K, Lyamin AV, Sloan SW. Three-dimensional Mohr-Coulomb limit analysis using semidefinite programming. Commun Numer. Meth Eng 2008;24:1107–19.
[35] Le CV, Gilbert M, Askes H. Limit analysis of plates using the EFG method and second-order cone programming. Int J Numer Mech Eng 2009;78:1532–52.
[36] Le CV, Gilbert M, Askes H. Limit analysis of plates using meshless equilibrium formulation. Int J Numer Mech Eng 2010;83:1793 1758.
[37] Le CV, Nguyen-Xuan H, Askes H, Bordas SPA, Rabczuk T, Nguyen-Vinh H. A cell-based smoothed finite element method for kinematic limit analysis. Int J Numer Mech Eng 2010;83:1651–74.
[38] Le CV, Nguyen-Xuan H, Nguyen-Dang H. Upper and lower bound limit analysis of plates using FEM and second-order cone programming. Comput Struct 2010;88:65–73.
[39] Leu SY. Static and kinematic limit analysis of orthotropic strain-hardening pressure vessels involving large deformation. Int J Mech Sci 2009;51:508–14.
[40] Li HX, Liu YH, Feng XQ, Cen ZZ. Mrico/marcomechanical plastic limit analyses of composite materials and structures. Acta Mech Solida Sin 2001;14:323–33.
[41] Li HX, Liu YH, Feng XQ, Cen ZZ. Limit analysis of ductile composites based on homogenization. Proc R Soc London Ser A 2003;459:659–75.
[42] Li HX, Yu HS. Kinematic limit analysis of frictional materials using nonlinear programming. Int J Solids Struct 2005;42:4058–76.
[43] Li HX, Yu HS. Limit analysis of 2-D and 3-D structures based on an ellipsoid yield criterion. Acta Geotech 2006;1:179–93.
[44] Liu GR, Gu YT. An introduction to meshfree methods and their programming. Spring; 2005.
[45] Liu GR, Dai KY, Nguyen TT. A smoothed finite element method for mechanics problems. Comput Mech 2007;39:859–77.
[46] Liu GR. Meshfree method: moving beyond the finite element method. 2nd ed. Boca Raton, USA: CRC press; 2010.
[47] Liu YH, Cen ZZ, Xu BY. A numerical method for plastic limit analysis of 3-D structures. Int J Solids Struct 1995;32:1645–58.
[48] Liu YH, Zhang X, Cen Z. Numerical determination of limit loads of three-dimensional structures using boundary element method. Eur J Mech – A/Solids 2004;23:127–38.
[49] Lyamin AV, Sloan SW. Upper bound limit analysis using linear finite elements and non-linear programming. Int J Numer Anal Meth Geomech 2002;26:181–216.
[50] Lyamin AV, Sloan SW. Lower bound limit analysis using non-linear programming. Int J Numer Meth Eng 2002;55:573–611.
[51] Lysmer J. Limit analysis of plane problems in soil mechanics. J Soil Mech Found Div 1970;96(4):1311–34.
[52] Makrodimopoulos A, Martin CM. Lower bound limit analysis of cohesive-frictional materials using second-order cone programming. Int J Numer Meth Eng 2006;66:604–34.
[53] Makrodimopoulos A, Martin CM. Upper bound limit analysis using simplex strain elements and second-order cone programming. Int J Numer Anal Mech Geomech 2007;31:835–65.
[54] Makrodimopoulos A, Martin CM. Finite-element limit analysis of Mohr-Coulomb materials in 3D using semidefinite programming. J Eng Mech ASCE 2008;134(4):339–47.
[55] Makrodimopoulos A, Martin CM. Upper bound limit analysis using discontinuous quadratic displacement fields. Commun Numer Meth Eng 2008;24:911–27.
[56] Makrodimopoulos A. Remarks on some properties of conic yield restrictions in limit analysis. Int J Numer Meth Biomed Eng 2010;26:1449–61.
[57] Nagtegaal JC, Parks DM, Rice JR. On numerically accurate finite element solutions in the fully plastic range. Comput Meth Appl Mech Eng 1974;4:153–77.
[58] Pastor F, Loute E. Solving limit analysis problems: an interior-point method. Commun Numer Methods Eng 2005;21:631–42.
[59] Pastor F, Trillat M, Pastor J, Loute E. Stress-based upper-bound method and convex optimization case of the Gurson material. C R Mec 2006;334:213–9.

[60] Pastor F, Thore Ph, Loute E, Pastor J, Trillat M. Convex optimization and limit analysis: Application to Gurson and porous Druckter–Prager materials. Eng Fract Mech 2008;75:1367–83.
[61] Pastor F, Loute E, Pastor J, Trillat M. Mixed method and convex optimization for limit analysis of homogeneous Gurson materials: a kinematic approach. Eur J Mech A/solids 2009;28:25–35.
[62] Sloan SW, Kleeman PW. Upper bound limit analysis using discontinuous velocity fields. Comput Appl Mech Eng 1995;127:293–314.
[63] Sloan SW. Lower bound limit analysis using finite elements and linear programming. Int J Numer Anal Meth Geomech 1988;12:61–77.
[64] Sloan SW. Upper bound limit analysis using finite elements and linear programming. Int J Numer Anal Meth Geomech 1989;13:263–82.
[65] Tin-Loi F, Ngo NS. Performance of the *p*-version finite element method for limit analysis. Int J Mech Sci 2003;45:1149–66.
[66] Vicente da Silva M, Antão AN. A non-linear programming method approach for upper bound limit analysis. Int J Numer Meth Eng 2007;72:1192–218.
[67] Wang JG, Liu GR. A point interpolation meshless method based on radial basis functions. Int J Numer Meth Eng 2002;54(11):1623–48.
[68] Yan AM, Nguyen-Dang H. Limit analysis of cracked structures by mathematical programming and finite element technique. Comput Mech 1999;24:319–33.
[69] Yang WH. A variational principle and an algorithm for limit analysis of beams and plates. Comput Meth Appl Mech Eng 1982;33:575–82.
[70] Yang WH. Large deformation of structures by sequential limit analysis. Int J Solids Struct 1993;30(7):1001–13.
[71] Yu HS, Sloan SW, Kleeman PW. A quadratic element for upper bound limit analysis. Eng Comput 1994;11:195–212.
[72] Zhang PX, Lu MW, Hwang K. A mathematical programming algorithm for limit analysis. Acta Mech Sin 1991;7:267–74.
[73] Zhang YG, Lu MW. An algorithm for plastic limit analysis. Comput Meth Appl Mech Engrgy 1995;126:333–41.
[74] Zhu T, Atluri SN. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. Comput Mech 1998;21:211–22.
[75] Zouain N, Herskovits J, Borges LA, Feijóo RA. An iterative algorithm for limit analysis with nonlinear yield functions. Int J Solids Struct 1993;30:1397–417.