Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

International Journal of Solids and Structures 41 (2004) 6465–6483

INTERNATIONAL JOURNAL OF SOLIDS and STRUCTURES

www.elsevier.com/locate/ijsolstr

Remesh-free shape optimization using the wavelet-Galerkin method

Gang-Won Jang [a], Yoon Young Kim [a,*], Kyung K. Choi [b]

[a] *School of Mechanical and Aerospace Engineering and National Creative Research, Initiatives Center for Multiscale Design, Seoul National University, San 56-1, Shillim-dong, Kwanak-gu, Seoul 151-742, South Korea*

[b] *Department of Mechanical and Industrial Engineering and Center for Computer Aided Design, The University of Iowa, Iowa City, IA 52242, USA*

Received 19 August 2003; received in revised form 3 May 2004
Available online 11 June 2004

Abstract

The objective of this investigation is to develop a new remesh-free shape optimization method based on the adaptive wavelet-Galerkin analysis. To avoid cumbersome remeshing processes, we embed the original analysis design domain inside a simple fictitious domain and use the predetermined analysis grid points. For efficient adaptive analysis, we utilize the multiscale multiresolution characteristics of the hat interpolation wavelets that are associated with predetermined grid points. Furthermore, we approximate the design boundary curve by oblique lines that do not necessarily pass through the analysis grid points and develop an accurate stress evaluation technique based on the proposed boundary approximation. The validity of the proposed remesh-free shape optimization formulation based on the wavelet-Galerkin method is verified through benchmark shape optimization problems.

Keywords: Shape optimization; Remesh-free; Wavelet-Galerkin; Multiscale

1. Introduction

Excessive mesh distortion requiring cumbersome remeshing processes is still a major problem in finite element based structural shape optimization. This problem becomes severe when large shape changes occur during optimization. To resolve this, the meshfree method (Kim et al., 2002) or the fixed-grid finite element method (Garcia and Steven, 1998) has been suggested as an alternative analysis tool to the standard finite element method.

To develop efficient meshfree methods, significant efforts have been made to generate shape functions that are independent of mesh geometries (see, e.g., Belytschko et al. (1994) and Liu et al. (1995)). Nevertheless, the complexity of the meshfree method in formulating shape functions and imposing boundary

[*] Corresponding author. Tel.: +82-2-8807154; fax: +82-2-8831513.
E-mail addresses: gangwon@idealab.snu.ac.kr (G.-W. Jang), yykim@snu.ac.kr (Y.Y. Kim), kkchoi@ccad.uiowa.edu (K.K. Choi).

0020-7683/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijsolstr.2004.05.010

conditions still hinders its spread into non-expert researchers. The fixed-gird finite element method by Garcia and Steven (1998) is much easier to implement for shape optimization than the meshfree method. It uses well-structured uniform grids while embedding the domain of interest into an enclosing fictitious domain of a simple geometry. In this approach, the material properties of elements lying over the boundary are approximated by their area fractions. This method avoids remeshing since the predetermined analysis grid points are never changed, but the resulting poor boundary approximation demands excessively fine meshing for the accurate prediction of local performance measures such as boundary stresses. Haslinger et al. (2001a,b) proposed fictitious domain methods with distributed Lagrange multipliers. In their approach, the homogeneous Dirichlet boundary condition can be exactly imposed even with a uniformly-triangulated mesh.

The main objective of the investigation is to develop a new shape optimization approach having the capability of adaptive analysis node insertion and accurate boundary curve approximation. To give an overview of the proposed method, we compare in Fig. 1 how shape optimization processes proceed by the proposed and existing methods. In Fig. 1, $\mathbf{C}$ denotes the elasticity tensor of the original elastic body, and $\tilde{\mathbf{C}}$ is the elasticity tensor to account for the area fraction of the elastic body within an element. Both the fixed-

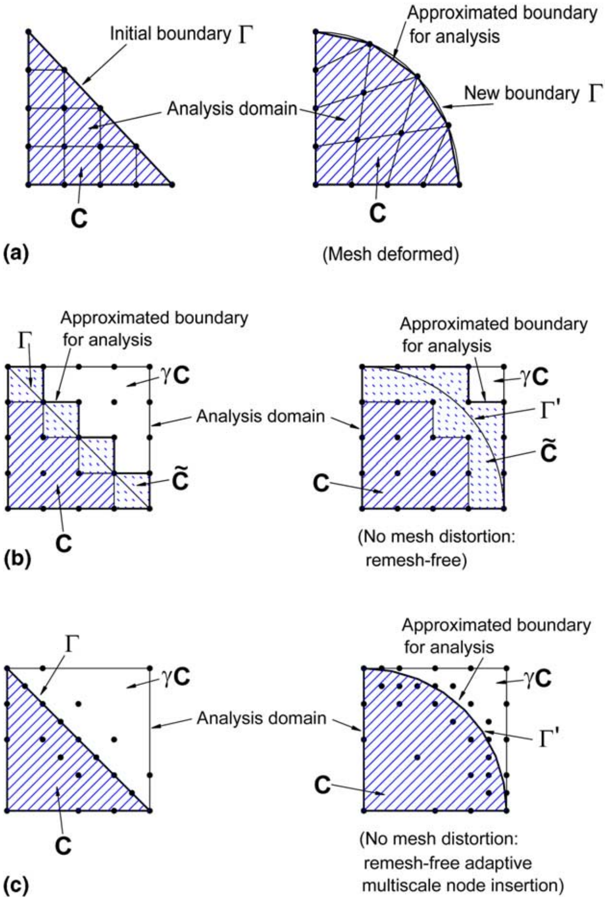

Initial boundary Γ
Analysis domain
C
(a)
Approximated boundary for analysis
New boundary Γ'
C
(Mesh deformed)
Approximated boundary for analysis
Γ
γC
Analysis domain
C̃
C
(b)
Approximated boundary for analysis
γC
Γ'
C̃
C
(No mesh distortion: remesh-free)
Γ
γC
Analysis domain
C
(c)
Approximated boundary for analysis
γC
Γ'
C
(No mesh distortion: remesh-free adaptive multiscale node insertion)

Fig. 1. Comparison of the characteristics of the proposed method with the existing methods. (The approximated boundaries for each method are illustrated with thick solid lines. Dots denote the locations of the nodes for analysis, and γ denotes a small positive number.) (a) Conventional finite element-based approach; (b) fixed-grid-based approach; (c) proposed approach.

grid method and the proposed method use the fictitious domain; so a small parameter γ is introduced to model the non-design domain as illustrated in Fig. 1(b) and (c).

In the sense that no remeshing is required and the analysis nodal locations are predetermined, the proposed method and the fixed-grid method are similar. However, the proposed method inserts analysis nodal points adaptively through the whole optimization process by introducing the adaptive multiresolution interpolation wavelet-Galerkin method. Based on the difference-checking nature of multiscale wavelets, the adaptive analysis will enhance the analysis efficiency considerably. Another important contribution of the proposed method is to approximate the boundary curve by piecewise oblique lines as illustrated in Fig. 1(c) and to compute the resulting system stiffness matrix without damaging the adaptive multiresolution property. As shall be seen later in numerical results, this greatly improves the accuracy in the stress calculation at or near the design boundary.

In implementing the wavelet-Galerkin method, either orthogonal/bi-orthogonal wavelets (Diaz, 1999) or interpolation wavelets (Kim and Jang, 2002) can be chosen as the basis functions. Orthogonal/bi-orthogonal wavelets have certain advantages such as vanishing moments, but they are difficult to use in handling general boundary curves. On the other hand, the hat interpolation wavelets have very simple functional forms and are easy to handle general boundary curves. In this work, we propose to employ the hat interpolation wavelet based method for shape optimization. To speed up the convergence of numerical analyses, one may incorporate a multiresolution adaptive strategy that is easy to utilize wavelet bases. To realize the multiresolution adaptive strategy, either the isoparametric mapping method (Christon and Roach, 2000) or the fictitious domain embedding method (Jang et al., 2004) may be utilized. The former approach allows multiresolution analysis only in an element level, so it is not appropriate for remesh-free shape optimization. On the other hand, the latter allows global-level multiresolution analysis, so the shape optimization can be achieved efficiently without remeshing.

Though the fictitious domain embedded wavelet-Galerkin method is efficient because of its excellent adaptivity, the fact that the analysis node locations are preset may decrease the solution efficiency. To overcome this bottleneck, we will approximate the design boundary curve by means of piecewise oblique lines. The oblique boundary curve does not usually pass through the analysis nodes, so a new scheme to evaluate the system stiffness matrix is needed. To this end, we propose a simple and effective scheme without damaging the remesh-free multiresolution adaptive property of the wavelet-Galerkin method. The use of this new scheme will slightly increase the computation time, but the improved accuracy as well as the overall computational speedup is considerable. The potential of this approach has been addressed by the authors (Jang et al., 2002, 2003). We also remark that there is no need to calculate the domain design velocity field in the proposed fictitious domain-embedded wavelet method, since the locations of node points are predetermined.

To parameterize domain boundaries, we use the Bezier or B-spline curves, so the control points of the curves are used as the design parameters. The advantages of using the control points, not the direct nodal coordinates, are addressed by Chang and Choi (1992). In this investigation, the sensitivity is calculated numerically by the finite difference. However, the advantages of the proposed approach are well demonstrated through benchmark problems.

2. Design parameterization

In this section, we will briefly describe the boundary parameterization method that will be used for the proposed method. Depending on how boundary curves are represented, the boundary nodal coordinates, polynomial coefficients, or the control points of the Bezier (or B-spline) curves can be used as the design parameters. The parameterization using boundary nodes results in a large number of design variables. This approach tends to result in non-smooth boundary curves with high computational cost. Moreover, it is not suitable for the proposed wavelet-Galerkin solver since the node locations for the analysis mesh remain unchanged throughout the optimization procedure. The use of polynomial coefficients as design parameters is reported to produce often oscillatory boundaries (Ding, 1986).

Based on these observations, we will mainly use the Bezier or B-spline curve to parameterize the design boundary. Fig. 2 illustrates how curves can be parameterized. If applicable, predefined geometric features such as the center point and the radius of a circle will be used as design variables as shown in Fig. 2(c). When a Bezier curve having $k+1$ control points is used to parameterize a boundary curve, a generic point $\mathbf{p}=\{p^x,p^y\}^{\mathrm{T}}$ can be expressed as

$$\mathbf{p}(u)=\sum_{i=0}^{k}B_{i,k}(u)\mathbf{P}_i \quad (0\leqslant u\leqslant 1) \tag{1}$$

where $\mathbf{P}_i=\{P_i^x,P_i^y\}^{\mathrm{T}}$ is the control point and $B_{i,k}(u)$ is the blending function of order k, which is defined as (Lee, 1999)

$$B_{i,k}(u)=\binom{k}{i}u^i(1-u)^{k-i}. \tag{2}$$

In case of $k=3$ which will be used in this work, Eqs. (1) and (2) can put into the following form:

$$\mathbf{p}=\mathbf{PAU} \tag{3a}$$

where

$$\mathbf{U}=\begin{Bmatrix}u^3\\u^2\\u\\1\end{Bmatrix},\quad \mathbf{A}=\begin{bmatrix}-1&3&-3&1\\3&-6&3&0\\-3&3&0&0\\1&0&0&0\end{bmatrix}\quad\text{and}\quad \mathbf{P}=\begin{bmatrix}P_0^x&P_1^x&P_2^x&P_3^x\\P_0^y&P_1^y&P_2^y&P_3^y\end{bmatrix}. \tag{3b}$$

Sometimes, it may be convenient to parameterize the boundary curve with two end points and their tangent vectors. In this case, the design variables are the end points $\mathbf{P}_A$ and $\mathbf{P}_B$ of the curve and their tangent vectors $\mathbf{P}'_A$ and $\mathbf{P}'_B$:

$$\mathbf{p}=\mathbf{QBU} \tag{4a}$$

where

$$\mathbf{B}=\begin{bmatrix}2&1&-2&1\\-3&-2&3&-1\\0&1&0&0\\1&0&0&0\end{bmatrix}\quad\text{and}\quad \mathbf{Q}=\begin{bmatrix}P_A^x&P_A'^x&P_B^x&P_B'^x\\P_A^y&P_A'^y&P_B^y&P_B'^y\end{bmatrix}. \tag{4b}$$

A boundary can be also parameterized using a B-spline curve. While the order of a Bezier curve is determined by the number of control points as in Eqs. (1) and (2), the order of a B-spline curve can be

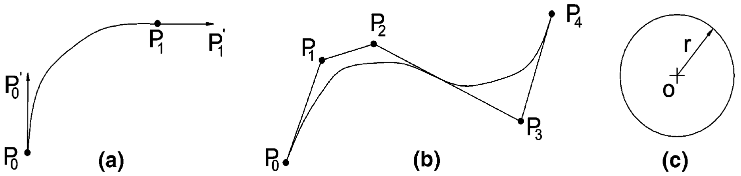


Fig. 2. Examples of the design parameterization: (a) end points and their tangent vectors of the Bezier curve, (b) control points of the B-spline curve and (c) predefined geometric features.

independently given. Moreover, a B-spline curve has the local modification property that moving a control point does not affect the shape of the entire curve. A B-spline curve with $(k-1)$-th order blending functions and $n+1$ control points is expressed as

$$\mathbf{p}(u) = \sum_{i=0}^{n} N_{i,k}(u)\mathbf{P}_i, \quad (0 \leqslant u \leqslant 1) \tag{5}$$

where the blending functions $N_{i,k}$ are defined by the following recursive formula:

$$N_{i,k}(u) = \frac{(u-t_i)N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}} \tag{6a}$$

$$N_{i,1}(u) = \begin{cases} 1 & t_i \leqslant u \leqslant t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{6b}$$

The symbol t_i in Eqs. (6) denotes the knot value.

3. Remesh-free Wavelet-Galerkin formulation

3.1. Comparison with the finite element analysis

To avoid remeshing during shape optimization and carry out the numerical analysis adaptively, we employ the multiscale wavelet-Galerkin method. By formulating the wavelet-Galerkin method in fictitious domain setting, and presetting the locations of analysis nodes, remeshing can be avoided. To explain the major difference between the standard finite element analysis and the wavelet-Galerkin analysis using the preset grids, we consider the classical problem of the torque arm design shown in Fig. 3. The boundary $\overline{AB}$ is parameterized by the Bezier curve, and the domain shape will be varied symmetrically. The end points of the curve are used as design variables with their tangent vectors fixed.

Consider the case where only the design variable P_A^y is changed. The effects of the change on the design boundary and the analysis mesh are shown in Fig. 4 for the finite element analysis and for the "single-scale" wavelet-Galerkin analysis using the preset analysis nodes. (We will actually use an adaptive multiscale wavelet-Galerkin method in this investigation, but the non-adaptive, single-scale version is illustrated just for the comparison.) As clearly seen in Fig. 4(b), there is no need for remeshing even for large shape changes as the analysis nodes are fixed regardless of the location of the domain boundary. In the multiscale wavelet method, the analysis nodes are adaptively located at some of the predetermined node locations.

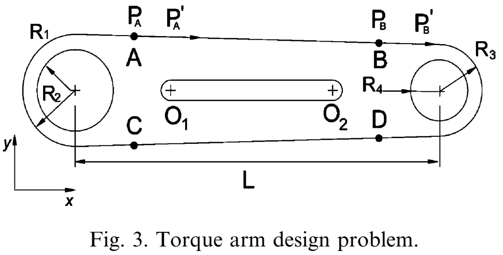


Fig. 3. Torque arm design problem.

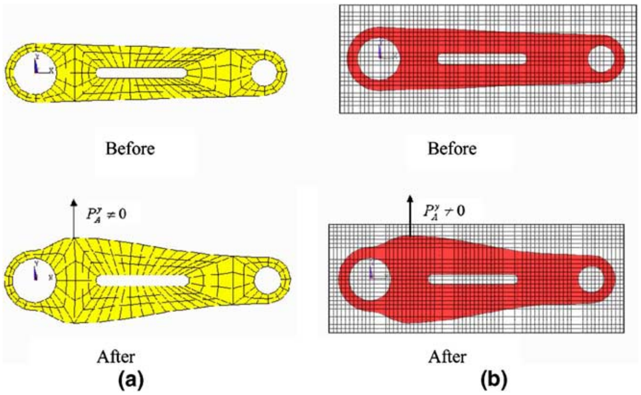


Fig. 4. The effects of the change of P_A^y (other variables unchanged) on the design boundary and the analysis mesh when (a) the finite element analysis and (b) the single-scale wavelet-Galerkin analysis using the fixed grids are used.

3.2. Multiscale wavelet-Galerkin approach

We begin with the single-scale Galerkin approximation and transform it into the multiscale Galerkin method using the hat interpolation wavelet (See Kim et al. (2003)). The weak formulation for a general plane elasticity problem is expressed as

Find the displacement $\mathbf{u} \in S_\omega$ for all $\mathbf{v} \in V_\omega$

$$\int_\omega \boldsymbol{\varepsilon}(\mathbf{v}) : \mathbf{C} : \boldsymbol{\varepsilon}(\mathbf{u})\,\mathrm{d}\omega = \int_\omega \mathbf{f} \cdot \mathbf{v}\,\mathrm{d}\omega + \int_{\Gamma_\omega^h} \mathbf{t} \cdot \mathbf{v}\,\mathrm{d}\Gamma \tag{7}$$

with

$$S_\omega = \{u_i \in H^1(\omega) | u_i = g_i \text{ on } \Gamma_\omega^g, \quad i = 1, 2\} \tag{8}$$

$$V_\omega = \{v_i \in H_0^1(\omega) | v_i = 0 \text{ on } \Gamma_\omega^g, \quad i = 1, 2\} \tag{9}$$

where $\mathbf{C}$ and H^1 denotes the elasticity tensor and the Sobolev space of degree 1, respectively. The strain tensor $\boldsymbol{\varepsilon}$ is the symmetric part $\nabla_{\text{sym}}\mathbf{u}$ of the displacement gradient $\nabla\mathbf{u}$. A surface traction $\mathbf{t}$ is applied on the section of the boundary Γ_ω^h and the displacement $\mathbf{u} = \mathbf{g}$ is prescribed on the boundary Γ_ω^g.

In the fictitious domain approach, the original domain of interest or the package domain ω is embraced by a fictitious domain Ω of a simple geometry. This is also illustrated in Fig. 5 and we will assume

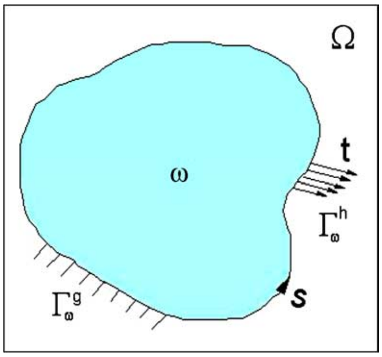


Fig. 5. The typical plane elasticity problem with a package domain ω embedded into a fictitious domain Ω.

$\Omega = [0,1] \times [0,1]$ without loss of generality. The fictitious domain should be determined large enough to embrace the maximum change of the package domain in the middle of the optimization process.

In the augmented domain Ω, the elasticity tensor $\mathbf{C}_\Omega$ and the load vector $\mathbf{f}_\Omega$ are now defined as

$$\mathbf{C}_\Omega = \begin{cases} \mathbf{C} & \text{in } \omega \\ \gamma\mathbf{C} & \text{in } \Omega \setminus \omega \end{cases} \tag{10}$$

and

$$\mathbf{f}_\Omega = \begin{cases} \mathbf{f} & \text{in } \omega \\ \mathbf{0} & \text{in } \Omega \setminus \omega \end{cases} \tag{11}$$

where γ is a small parameter.

The weak formulation (7) is now approximated by the problem to find $\mathbf{u}_\Omega \in S_\Omega$ satisfying

$$\int_\Omega \boldsymbol{\varepsilon}(\mathbf{v}_\Omega) : \mathbf{C}_\Omega : \boldsymbol{\varepsilon}(\mathbf{u}_\Omega)\,\mathrm{d}\Omega = \int_\Omega \mathbf{f}_\Omega \cdot \mathbf{v}_\Omega\,\mathrm{d}\Omega + \int_{\Gamma^h_\omega} \mathbf{t} \cdot \mathbf{v}_\Omega\,\mathrm{d}\Gamma \tag{12}$$

for all $\mathbf{v}_\Omega \in V_\Omega$. The definition of the spaces S_Ω and V_Ω is similar to that of S_ω and V_ω:

$$S_\Omega = \{u_{\Omega i} \in H^1(\Omega) | u_{\Omega i} = g_i \text{ on } \Gamma^g_\omega,\ i = 1,2\} \tag{13}$$

$$V_\Omega = \{v_{\Omega i} \in H^1(\Omega) | v_{\Omega i} = 0 \text{ on } \Gamma^g_\omega,\ i = 1,2\} \tag{14}$$

Obviously, the value of γ affects the solution accuracy and convergence. Jang et al. (2004) showed that if $\gamma \leqslant 0.001$, the difference between the solution of Eq. (12) and the solution of Eq. (7) is almost negligible. Too small values of γ, however, could cause the ill-conditioning of the stiffness matrix. For all numerical examples considered in this work, $\gamma = 0.001$ was used.

Now consider the approximation $\mathbf{u}^J_\Omega$ of $\mathbf{u}_\Omega$ at the resolution level of J. The resolution symbol J corresponds to the mesh divided by $2^J + 1$ by $2^J + 1$ analysis nodes for the case of a square region. For a rectangular region, J should represent the mesh division for the shorter edge of Ω. When the hat interpolation scaling function $\phi_{J,k,l}(x,y)$ to approximate the space S_Ω is used, we write

$$\mathbf{u}^J_\Omega = \sum_{k,l} \mathbf{s}^u_{J,k,l}\phi_{J,k,l}(x,y)$$

or

$$\begin{Bmatrix} u^J_{x\Omega} \\ u^J_{y\Omega} \end{Bmatrix} = \sum_{k,l} \begin{Bmatrix} s^u_{x;J,k,l} \\ s^u_{y;J,k,l} \end{Bmatrix} \phi_{J,k,l}(x,y) \tag{15}$$

where $(k,l) \in \{0,1,2,\ldots,2^J\}$. The symbol $\widehat{\mathbf{U}}_J$ will be used to denote the $\mathbf{s}^u_{J,k,l}(k,l \in \{0,1,2,\ldots,2^J\})$. The scaling function $\phi_{j,k,l}(x,y)$ is defined as

$$\phi_{j,k,l}(x,y) = 2^j\phi(2^jx - k, 2^jy - l) \tag{16}$$

where

$$\phi(x,y) = \hat{\phi}(x)\hat{\phi}(y) \tag{17}$$

with

$$\hat{\phi}(x) = \begin{cases} x+1 & \text{for } -1 \leqslant x \leqslant 0 \\ 1-x & \text{for } 0 \leqslant x \leqslant 1 \\ 0 & \text{else} \end{cases} \tag{18}$$

If $\Omega = [0,1] \times [0,1]$, for instance, the representation (17) is exactly the same as the bilinear basis function representation used in the finite element analysis.

By using the same approximation for $\mathbf{v}_\Omega^J$ and substituting the approximations for $\mathbf{u}_\Omega^J$ and $\mathbf{v}_\Omega^J$ into (12), we obtain the following system of equations in single-scale form:

$$\widehat{\mathbf{K}}_J \widehat{\mathbf{U}}_J = \widehat{\mathbf{F}}_J. \tag{19}$$

In (19), the load term $\widehat{\mathbf{F}}_J$ has components $\mathbf{s}_{J,k,l}^f$ which is defined as

$$\mathbf{s}_{J,k,l}^f = \int_\Omega \phi_{J,k,l} \mathbf{f}_\Omega \, d\Omega + \int_{\Gamma_\omega^h} \phi_{J,k,l} \mathbf{t} \, d\Gamma. \tag{20}$$

The system stiffness matrix $\widehat{\mathbf{K}}_J$ consists of

$$\widehat{K}_{J,k,l,k',l'} = \int_\Omega \nabla_{\text{sym}} \phi_{J,k,l} : \mathbf{C}_\Omega : \nabla_{\text{sym}} \phi_{J,k',l'} \, d\Omega. \tag{21}$$

In actual implementation, the double-index notation (k,l) will be converted to a single-index notation, so $\widehat{\mathbf{U}}_J$ and $\widehat{\mathbf{F}}_J$ are column arrays and $\widehat{\mathbf{K}}_J$ is a two-dimensional array.

To carry out the multiscale wavelet-Galerkin analysis, we must express $\mathbf{u}_\Omega^J$ in the following multiscale form using the wavelet basis functions:

$$\mathbf{u}_\Omega^J = \sum_{k,l} \mathbf{s}_{j_0,k,l}^u \phi_{j_0,k,l}(x,y) + \sum_{j=j_0}^{J-1} \sum_{m=1}^{3} \sum_{k,l} \mathbf{d}_{j,k,l}^{u,m} \psi_{j,k,l}^m(x,y) \tag{22}$$

where $\psi_{j,k,l}^m$ is the hat interpolation wavelet defined as

$$\psi_{j,k,l}^m(x,y) = 2^j \psi^m(2^j x - k, 2^j y - l) \quad (m = 1,2,3). \tag{23}$$

Two-dimensional wavelets ψ^1, ψ^2 and ψ^3 are constructed by the tensor product of the one-dimensional wavelet $\hat{\psi}$ and scaling function $\hat{\phi}$:

$$\psi^1(x,y) = \hat{\psi}(x)\hat{\phi}(y) \tag{24a}$$

$$\psi^2(x,y) = \hat{\phi}(x)\hat{\psi}(y) \tag{24b}$$

$$\psi^3(x,y) = \hat{\psi}(x)\hat{\psi}(y). \tag{24c}$$

The one-dimensional interpolation wavelet $\hat{\psi}(x)$ is defined as

$$\hat{\psi}(x) = \hat{\phi}(2x-1) = \begin{cases} 2x & \text{for } 0 \leqslant x \leqslant 1/2 \\ 2-2x & \text{for } 1/2 \leqslant x \leqslant 1 \\ 0 & \text{else} \end{cases} \tag{25}$$

Since ψ^1, ψ^2 and ψ^3 represent the differences in the horizontal, vertical and diagonal directions, they are often called the horizontal, vertical and diagonal wavelets, respectively.

In Eq. (22), the scale index j for wavelets ranges from j_0 to $J-1$ ($j_0 < J$, j_0 is usually set to be 1). When $j_0 = 1, J = 2$ and $\Omega = [0,1] \times [0,1]$, there are nine interpolation scaling functions $\phi_{1,k,l}(x,y)|_{(k,l) \in (0,1,2)}$ and 16 interpolation wavelets consisting of six horizontal wavelets ($\psi_{1,k,l}^1(x,y)$ for $k \in (0,1)$, $l \in (0,1,2)$), six vertical wavelets ($\psi_{1,k,l}^2(x,y)$ for $k \in (0,1,2)$, $l \in (0,1)$) and four diagonal wavelets ($\psi_{1,k,l}^3(x,y)$ for $k,l \in (0,1)$). In Fig. 6, we plot $\phi_{1,1,1}$, $\psi_{1,0,1}^1$, $\psi_{1,1,0}^2$ and $\psi_{1,0,0}^3$.

The direct approach to construct the discretized multiscale system equation requires somewhat involved analysis. The easiest way to derive the multiscale system equation is to transform the single-scaled equation (19) into the multiscale system equation (see Jang et al. (2004)):

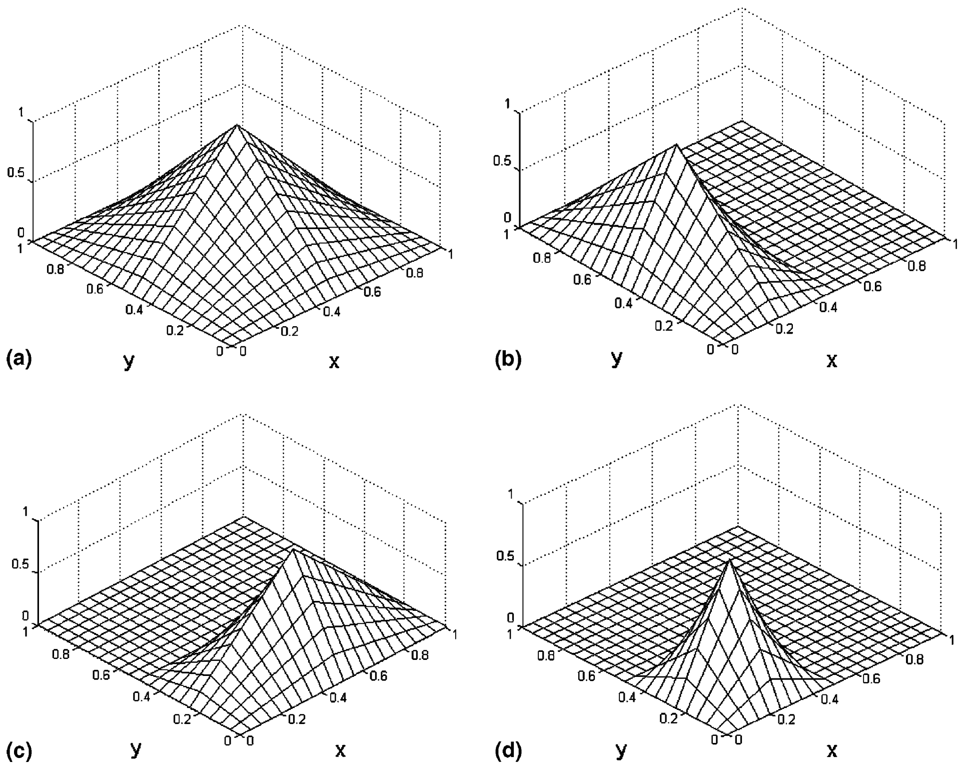


Fig. 6. Illustration of the hat interpolation wavelet system defined on $\Omega = [0,1] \times [0,1]$. (a) $\phi_{1,1,1}$; (b) $\psi^1_{1,0,1}$; (c) $\psi^2_{1,1,0}$; (d) $\psi^3_{1,0,0}$.

$$\mathbf{K}_J \mathbf{U}_J = \mathbf{F}_J \tag{26}$$

through the following transform

$$\widehat{\mathbf{U}}_J = \mathbf{T}_{\text{system}} \mathbf{U}_J \tag{27}$$

$$\mathbf{F}_J = \mathbf{T}^{\mathrm{T}}_{\text{system}} \widehat{\mathbf{F}}_J \tag{28}$$

$$\mathbf{K}_J = \mathbf{T}^{\mathrm{T}}_{\text{system}} \widehat{\mathbf{K}}_J \mathbf{T}_{\text{system}}. \tag{29}$$

In Eqs. (27)–(29), $\mathbf{T}_{\text{system}}$ is the transformation matrix to convert single-scale variables into multiscale variables. The procedure to construct $\mathbf{T}_{\text{system}}$ is given in Jang et al. (2004), so the explicit expression of $\mathbf{T}_{\text{system}}$ will not be given here. The column array $\mathbf{U}_J$ is defined as

$$\mathbf{U}_J = \left\{ \mathbf{s}^u_{j_0}, \mathbf{d}^u_{j_0}, \mathbf{d}^u_{j_0+1}, \ldots, \mathbf{d}^u_{J-1} \right\}^{\mathrm{T}}$$

where

$$\mathbf{s}^u_{j_0} = \{ \mathbf{s}^u_{j_0,k,l} \}^{\mathrm{T}}, \quad (k,l) \in \{0,1,2,\ldots,2^{j_0}\}$$

$$\mathbf{d}^u_j = \{ {}^1\mathbf{d}^u_j, {}^2\mathbf{d}^u_j, {}^3\mathbf{d}^u_j \}^{\mathrm{T}}$$

with

$$ {}^{1}\mathbf{d}_j^u = \{{}^{1}\mathbf{d}_{j,k,l}^u\}^{\mathrm{T}}, \quad k \in \{0,1,\ldots,2^j-1\}, \quad l \in \{0,1,\ldots,2^j\} $$

$$ {}^{2}\mathbf{d}_j^u = \{{}^{2}\mathbf{d}_{j,k,l}^u\}^{\mathrm{T}}, \quad k \in \{0,1,\ldots,2^j\}, \quad l \in \{0,1,\ldots,2^j-1\} $$

$$ {}^{3}\mathbf{d}_j^u = \{{}^{3}\mathbf{d}_{j,k,l}^u\}^{\mathrm{T}}, \quad k \in \{0,1,\ldots,2^j-1\}, \quad l \in \{0,1,\ldots,2^j-1\} $$

The column array $\mathbf{F}_J$ is similarly defined.

The biggest advantage of using multiscale wavelets as the basis functions is the efficiency in adaptive analysis; since the wavelet coefficients represent the difference between the solutions at different resolution levels, they can be directly used as error estimators for adaptive analysis. By using an adaptive scheme, the additional numerical expense caused by the introduction of a fictitious domain can be substantially reduced. Based on the work by Cohen and Masson (1999), we employ the traditional concept that utilizes the lower and upper threshold parameters δ_j^{up} and $\delta_j^{\mathrm{low}}(\delta_j^{\mathrm{up}} > \delta_j^{\mathrm{low}} > 0)$.

The adaptive scheme:

- Exclude wavelet $\psi_{j,k,l}^m$ from the basis set if $|d_{j,k,l}^m| < \delta_j^{\mathrm{low}}$.
- Preserve $\psi_{j,k,l}^m$ in the basis set if $\delta_j^{\mathrm{low}} \leqslant |d_{j,k,l}^m| \leqslant \delta_j^{\mathrm{up}}$.
- Add children wavelets of $\psi_{j,k,l}^m$ into the basis set if $|d_{j,k,l}^m| > \delta_j^{\mathrm{up}}$.

The children wavelets of $\psi_{j,k,l}^m$ represent the wavelets appearing in the next higher resolution $j+1$ with their support centers just around the support center of $\psi_{j,k,l}^m$. The analysis procedure is depicted in Fig. 7.

The Dirichlet condition on Γ_ω^g is prescribed along the grids in Ω nearest to Γ_ω^g. Following the work by Jang et al. (2004), the constrained degrees of freedom are condensed out from $\widehat{\mathbf{K}}_J$ before the multiscale

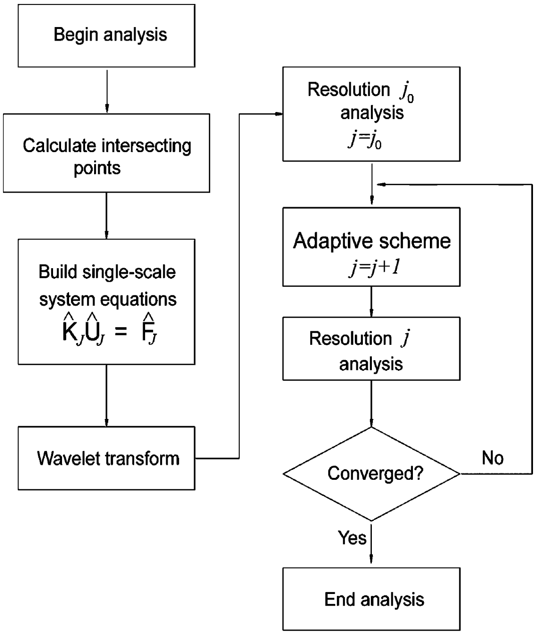

Begin analysis
Calculate intersecting points
Build single-scale system equations
$\hat{K}_J\hat{U}_J = \hat{F}_J$
Wavelet transform
Resolution j_0 analysis
$j=j_0$
Adaptive scheme
$j=j+1$
Resolution j analysis
Converged?
No
Yes
End analysis

Fig. 7. Flowchart for the multiscale adaptive analysis.

transform in Eqs. (28) and (29) is performed. In this work, to circumvent the labor to resize $\widehat{\mathbf{K}}_J$, we simply used the penalty method by adding very large numbers to the corresponding components of the stiffness matrix. For example, the imposition of $\widehat{U}_J(i) = \overline{U}$ is replaced by the following penalization:

$$\widehat{K}_J(i,i) \leftarrow \widehat{K}_J(i,i) + \alpha \quad \text{and} \quad \widehat{F}_J(i) \leftarrow \widehat{F}_J(i) + \alpha\overline{U} \quad (\alpha \gg \widehat{K}_J(i,i)).$$

4. Stress evaluation issue

When stresses are used as the performance measure in shape optimization, the accurate stress evaluation is critical in speeding up the solution convergence and improving the quality of the final design. Though the existing multiscale wavelet-Galerkin method was successful in topology optimization applications (Kim et al., 2003), its direct application to shape optimization problems will not be so effective. The main reason is that the existing multiscale wavelet-Galerkin method works with the domain boundary approximation modeled by stair-step representation. For example, consider a package domain ω bounded by a circle shown in Fig. 8(a). Since we use the wavelet-Galerkin method using pre-determined node locations, as depicted in Fig. 8(a), the circular boundary of Fig. 8(a) is modeled by an approximation in Fig. 8(b). Because the approximation in Fig. 8(b) is a poor approximation to the curve in Fig. 8(a), there arise two major problems. Unless the target point for the stress measurement lies at the exact nodal locations, it is difficult to get accurate stress values when the approximation in Fig. 8(b) is used. Furthermore, the stair-step approximation yields considerably inaccurate stiffness values especially near the domain boundary.

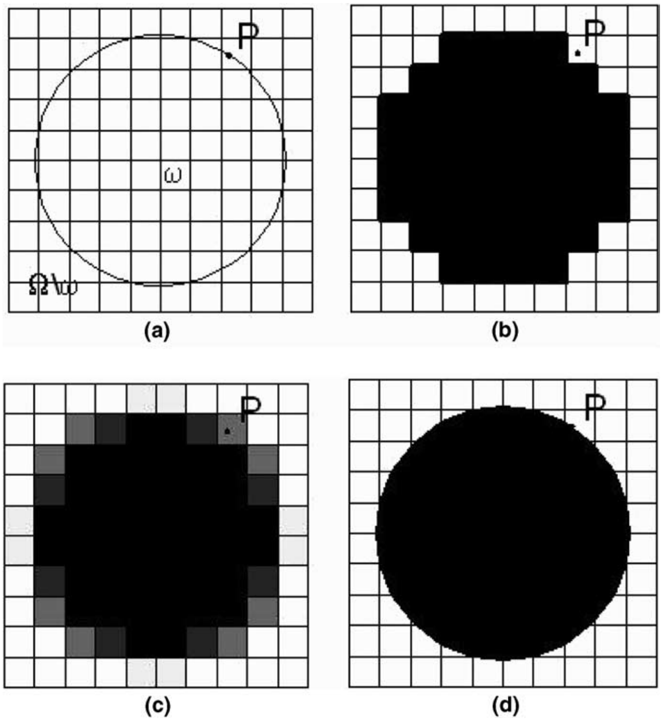


Fig. 8. The approximation of a circle: (a) the original domain of the circle and its approximated domains using, (b) stair-step representation, (c) area-fraction-based method and (d) piecewise oblique lines.

Though all the analyses are carried out in multiscales, we will consider the boundary approximation issue and the stress evaluation issue in the single scale; once the strategy for the single-scale analysis is established, we can convert the result by using the transform rules such as Eqs. (27)–(29). In the single-scale fixed grid method, exactly the same problem occurred. Garcia and Steven (1998) proposed an area-fraction-based stiffness evaluation method to improve the solution accuracy. Their idea is to assign an area-fraction-based elasticity tensor $\mathbf{C}^{\mathrm{e}}$ for elements lying on the domain boundary as:

$$\mathbf{C}^{\mathrm{e}} = \alpha^{\mathrm{e}}\mathbf{C} + (1-\alpha^{\mathrm{e}})\gamma\mathbf{C} \tag{30}$$

where

$$\alpha^{\mathrm{e}} = \frac{A^{\mathrm{e}}_{\omega}}{A^{\mathrm{e}}}. \tag{31}$$

In Eq. (31), A^{e} and A^{e}_{ω} stand for the boundary element area and the portion of A^{e} lying inside ω, respectively. How the area-fraction-based method modifies the stiffness value for elements lying on the boundary is depicted in Fig. 8(c). The gray level in the figure corresponds to the magnitude of the element stiffness depending on the area fraction of the original elastic body within the element.

To improve the solution accuracy substantially, we approximate the boundary curve by piecewise oblique lines, as shown in Fig. 8(d) and develop a new scheme to calculate accurately the stiffness matrix of the corresponding element. To this end, we first determine the intersection points of the domain boundary and the element boundaries, e.g., points 2 and 3 in Fig. 9, and connect these points to form the approximate boundary line. Note that the oblique boundary usually crosses the element domain without passing through the analysis nodes.

Now we propose a new strategy to evaluate the stiffness. Here, we simply work with one element denoted by e in Fig. 9. First, we express the single-scale stiffness matrix $\mathbf{k}^{\mathrm{e}}$ as

$$\mathbf{k}^{\mathrm{e}} = \int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^{\mathrm{T}}(\xi,\eta)\widehat{C}_{\Omega}(\xi,\eta)\mathbf{B}(\xi,\eta)|\mathbf{J}|\,\mathrm{d}\xi\,\mathrm{d}\eta \tag{32}$$

where (ξ,η) are the element local coordinates, and $\mathbf{B}$ denotes the matrix relating strains to nodal displacements. The matrix version of the elasticity tensor $\mathbf{C}_{\Omega}$ is denoted by $\widehat{C}_{\Omega}$. The Jacobian $|\mathbf{J}|$ is simply one

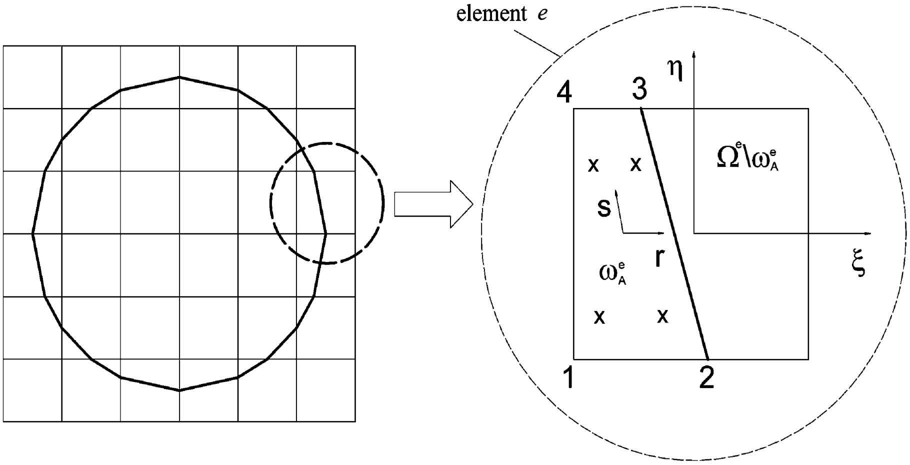


Fig. 9. The circular boundary approximated by piecewise oblique lines. The element local coordinates (ξ,η) and material coordinates (r,s) are also shown.

quarter of the element area. To integrate (32), we introduce the material coordinates (r,s) that maps a normalized rectangular domain $[-1,1]\times[-1,1]$ to the region bounded by 1–2–3–4 in Fig. 9. In Fig. 9, ω_A^{e} and $\Omega^{\mathrm{e}}\setminus\omega_A^{\mathrm{e}}$ are the approximations of ω and $\Omega\setminus\omega$ within the element, respectively. As in the standard bilinear finite element, the element local coordinates are expressed as

$$\begin{pmatrix}\xi\\ \eta\end{pmatrix}=\sum_{i=1}^{4}N_i(r,s)\begin{pmatrix}\xi_i\\ \eta_i\end{pmatrix} \tag{33}$$

where (ξ_i,η_i) are the coordinates of points 1, 2, 3 and 4 in Fig. 9 and $N_i(r,s)$ represent standard bilinear functions. If ω_A^{e} becomes a triangular region, it can be also treated by (33) as a degenerate case.

Using the transformation (33), the element stiffness in (32) can be written as

$$\begin{aligned}\mathbf{k}^{\mathrm{e}}&=\gamma\int_{-1}^{1}\int_{-1}^{1}\mathbf{B}^{\mathrm{T}}(\xi,\eta)\overset{\frown}{\mathbf{C}}\mathbf{B}(\xi,\eta)|\mathbf{J}|\,\mathrm{d}\xi\,\mathrm{d}\eta+(1-\gamma)\int_{\omega_A^{\mathrm{e}}}\mathbf{B}^{\mathrm{T}}(\xi,\eta)\overset{\frown}{\mathbf{C}}\mathbf{B}(\xi,\eta)|\mathbf{J}|\,\mathrm{d}\xi\,\mathrm{d}\eta\\&=\gamma\int_{-1}^{1}\int_{-1}^{1}\mathbf{B}^{\mathrm{T}}(\xi,\eta)\overset{\frown}{\mathbf{C}}\mathbf{B}(\xi,\eta)|\mathbf{J}|\,\mathrm{d}\xi\,\mathrm{d}\eta+(1-\gamma)\int_{-1}^{1}\int_{-1}^{1}\mathbf{B}^{\mathrm{T}}(\xi(r,s),\eta(r,s))\overset{\frown}{\mathbf{C}}\mathbf{B}(\xi(r,s),\eta(r,s))|\mathbf{J}||\mathbf{J}_1|\,\mathrm{d}r\,\mathrm{d}s\end{aligned} \tag{34}$$

where $|\mathbf{J}_1|$ is the Jacobian relating (r,s) and (ξ,η). When γ is small as in the proposed problem, Eq. (34) can be simplified to

$$\mathbf{k}^{\mathrm{e}}\approx\int_{-1}^{1}\int_{-1}^{1}\mathbf{B}^{\mathrm{T}}(\xi(r,s),\eta(r,s))\overset{\frown}{\mathbf{C}}\mathbf{B}(\xi(r,s),\eta(r,s))|\mathbf{J}||\mathbf{J}_1|\,\mathrm{d}r\,\mathrm{d}s \tag{35}$$

When $\Omega^{\mathrm{e}}\setminus\omega_A^{\mathrm{e}}$ is a triangle, ω_A^{e} becomes a pentagon. In this case, Eqs. (34) and (35) are not applicable and we use the following equation:

$$\mathbf{k}^{\mathrm{e}}=\int_{-1}^{1}\int_{-1}^{1}\mathbf{B}^{\mathrm{T}}(\xi,\eta)\overset{\frown}{\mathbf{C}}\mathbf{B}(\xi,\eta)|\mathbf{J}|\,\mathrm{d}\xi\,\mathrm{d}\eta-(1-\gamma)\int_{-1}^{1}\int_{-1}^{1}\mathbf{B}^{\mathrm{T}}(\xi(r,s),\eta(r,s))\overset{\frown}{\mathbf{C}}\mathbf{B}(\xi(r,s),\eta(r,s))|\mathbf{J}||\mathbf{J}_2|\,\mathrm{d}r\,\mathrm{d}s \tag{36}$$

where $|\mathbf{J}_2|$ results from the mapping between a normalized rectangle and $\Omega^{\mathrm{e}}\setminus\omega_A^{\mathrm{e}}$. To integrate the second term in Eq. (36), the integration scheme in the $r-s$ coordinates will be degenerated to handle a triangular region.

We emphasize that the stiffness evaluation above is applied only to elements lying on the boundary of ω. For elements completely inside or outside of the boundary, the element stiffness evaluation is exact in the proposed wavelet-Galerkin method. Therefore, the main advantages of the fixed grid-type method such as fast meshing and efficient stiffness matrix formulation are still preserved. It only requires small additional expenses to calculate the intersection points and element stiffness matrices along the boundary of the domain.

5. Numerical examples

In this section, we will consider two benchmark problems for which the design performance measures consist of the total mass of a structure and the von Mises stress. Unlike mass, stress represents local solution behavior and the extreme value of it will appear only on the boundary when the body force is absent. For the accurate and stable estimation of the boundary stress, we use the least square approximation. The stresses at the Gauss points of the neighboring elements connected to the stress calculation point are used to approximate the stress on the desired point. However, we exclude the elements lying

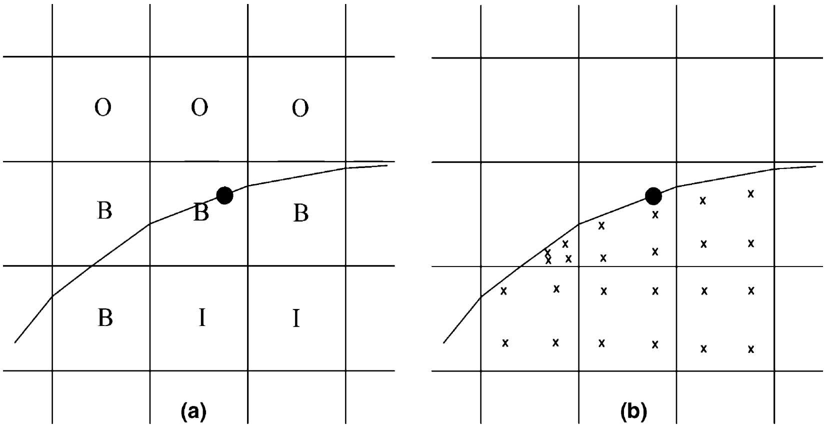


Fig. 10. (a) The patch used to calculate the stress on a boundary point and (b) the Gauss points considered in the least square approximation. (I, O and B denote the elements inside the package domain, outside the package domain and on the boundary, respectively.)

outside the package domain as they are artificial. Fig. 10 shows the neighboring elements and their Gauss points that are used to calculate the stress at the desired point.

The gradient-based shape optimization requires the sensitivity analysis. The analytic sensitivity for the proposed wavelet-Galerkin method based on the fictitious domain method has not yet been established, so we will carry out the analysis numerically by the finite difference method. Even without the analytic sensitivity analysis, however, we can demonstrate all of the key features of the proposed method.

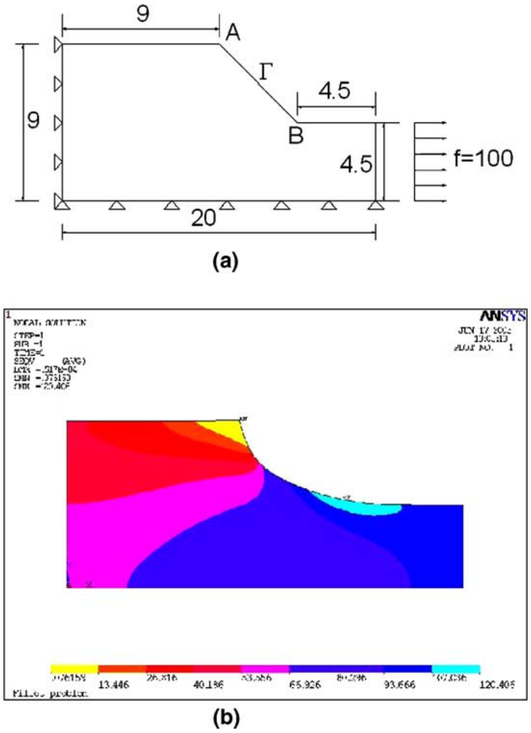
5.1. Fillet optimization

We consider the shape optimization of a fillet in a tension bar shown in Fig. 11(a). The design boundary Γ connecting points A and B is parameterized by the B-spline curve of order 4 with six control points. The control points corresponding to the two end points are fixed, so eight parameters among the control points $\mathbf{P}_i$ in Eq. (5) are used as the design parameters. The optimization problem is given as

$$\begin{aligned} &\underset{\mathbf{P}=\{P_1^x,P_1^y,\ldots,P_4^x,P_4^y\}}{\text{minimize}} \quad f = \text{total mass} \\ &\text{subject to} \quad \sigma_i^{VM} \leqslant \sigma^a \quad (i = 1, 2, \ldots, 20) \end{aligned}$$

where σ_i^{VM} denotes the von Mises stress of a point under stress constraint and σ^a denotes the allowable von Mises stress. The geometry and the material properties are given in Fig. 11(a).

The optimized result by the proposed method is illustrated in Fig. 11(b), where the stress distribution is also shown. We used SQP (Sequential Quadratic Programming) in DOT (Vanderplaats, 1997), and the optimization was completed in nine iterations. Fig. 12 shows how adaptively the hat interpolation wavelet points are added for the analyses of the initial and final optimized fillet configurations. Note that the wavelets are mainly positioned around the fillet, which is the region of high stress. Therefore, the efficiency of the wavelet-based adaptive method is greatly enhanced for problems having highly localized solutions. The convergence and the solution accuracy can be affected by the values of the threshold parameters. We used the threshold values of δ_j^{up} and δ_j^{low} as 1% and 0.01% of the maximum of the solution at the lowest resolution level. Note also that the node locations for the horizontal and vertical displacements are not necessarily the same in the proposed wavelet-based adaptive method.



(a)

(b)

Fig. 11. Fillet shape optimization problem: (a) problem description (Young's modulus $E = 3 \times 10^7$, Poisson's ratio $\nu = 0.3$, allowable von Mises stress $\sigma^a = 120$), (b) the optimized result with the stress distribution. (The stress constraint satisfaction is checked.)

To examine the effect of the boundary approximation on the solution, the stress for the final optimized fillet shape in Fig. 11 is calculated again by the area-fraction-based method. Table 1 compares the stress values along the fillet obtained by different approaches. The results are also compared against the refined ANSYS finite element results. Table 1 shows that the results of the proposed method agree with the ANSYS results within 2%, while the area-fraction-based method results are not in good agreement with the ANSYS results.

5.2. Torque arm optimization

The torque arm problem (Bennett and Botkin, 1985) in Fig. 3 is considered as the second example. The dimensions and material properties of the torque arm are given as $L = 42$, $R_1 = 6$, $R_2 = 4$, $R_3 = 4.5$, $R_4 = 2.5$, $E = 2074$, $\nu = 0.3$ and $\sigma^a = 2.4$. As considered in Section 2, the third-order Bezier curve with four control points in Eq. (4) is used to parameterize the curve between A and B. The parameters for the curve CD are symmetrically linked to the parameters of the curve AB. We choose the position of the control points A and B as design variables with their tangent vectors fixed. The radii of the half circle O_1 and O_2 and their positions are also set as design variables, so the total number of the design variables is eight.

The final optimized result shown in Fig. 13 agrees well with the existing result (Bennett and Botkin, 1985). If the finite element analysis were used for this example, severe mesh distortion would occur due to the large shape change of the model (see Bennett and Botkin (1985)). Fig. 14 shows how the total mass decreases as the proposed design optimization proceeds. Table 2 demonstrates how accurately the stress is

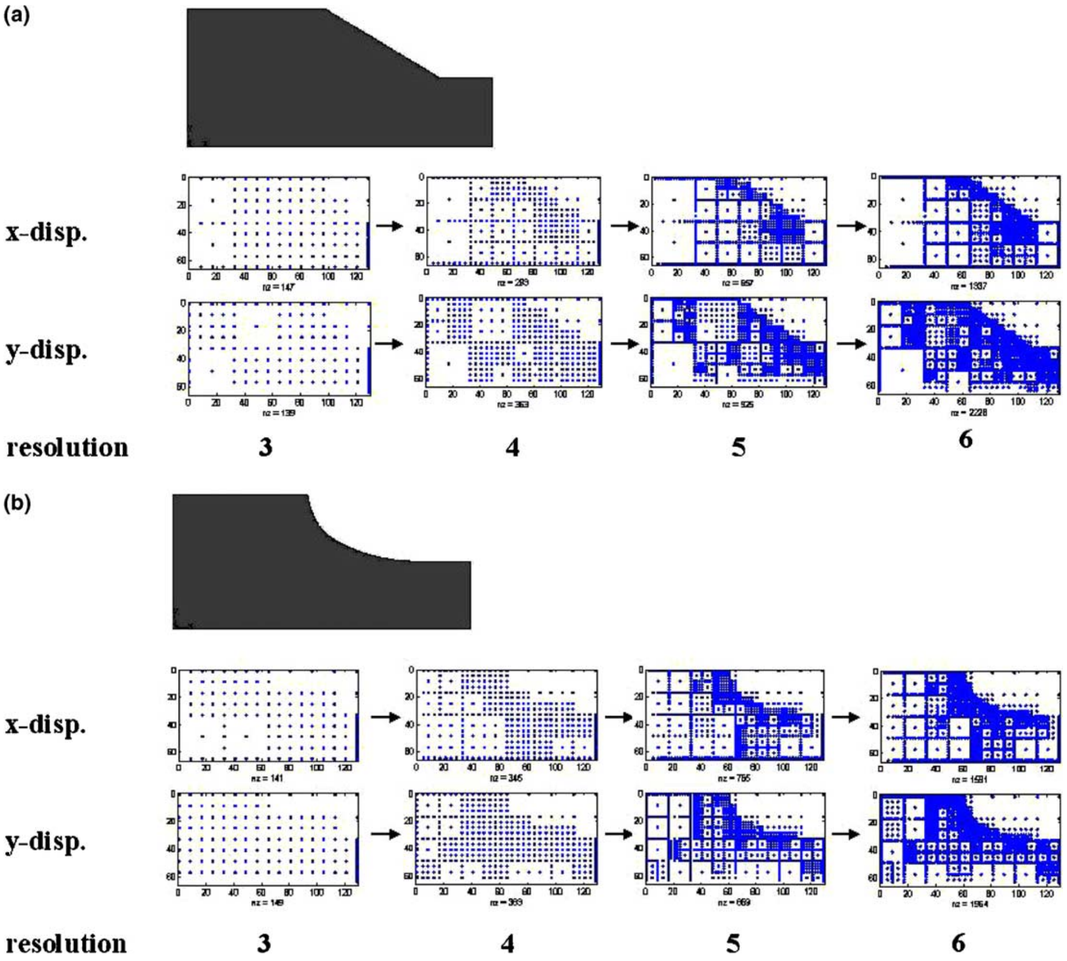


Fig. 12. The distribution of adaptively-added wavelet nodes for (a) the initial and (b) the final fillet configurations.

Table 1
The comparison of the von Mises stresses on the boundary of the fillet obtained by ANSYS (3329 eight-node plane elements) and the wavelet-Galerkin method with two different boundary approximation methods

x	y	Area-fraction-based method		Proposed method		ANSYS
		Value	Agreement (%)	Value	Agreement (%)	
12.67	5.14	103.7401	92.78	111.8565	100.04	111.8142
13.01	5.02	102.9263	89.11	116.9224	101.23	115.5072
13.37	4.92	108.7474	91.96	118.3333	100.07	118.2497
13.72	4.83	119.9064	99.99	119.9563	100.03	119.9157
14.09	4.75	115.2029	95.74	120.2007	99.90	120.3253
14.46	4.69	118.0062	98.74	118.7356	99.35	119.5149
14.83	4.63	115.0632	97.10	117.7819	99.39	118.4992
15.20	4.59	114.1644	97.01	115.6407	98.26	117.6858
15.54	4.56	115.0972	98.21	115.3933	98.46	117.1977

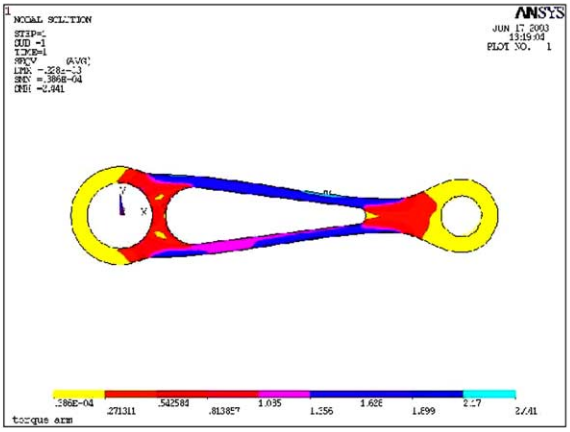


Fig. 13. Optimized shape of the torque arm. (Stress constraint is verified using ANSYS.)

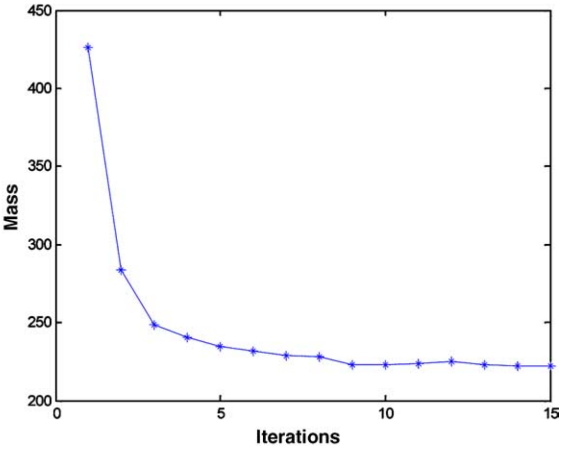


Fig. 14. Iteration history of the objective value in the torque arm problem.

Table 2
The comparison of the von Mises stresses on the boundary of the torque arm between the wavelet-Galerkin method and ANSYS (5535 elements of PLANE82 (eight-node element))

x	y	WGM	ANSYS	Agreement (%)
6.08	4.69	1.9794	1.9821	99.86
7.72	4.66	2.2929	2.2958	99.87
8.54	4.63	2.3919	2.4045	99.48
10.19	4.54	2.0533	2.0566	99.84
11.83	4.4	1.7105	1.7337	98.66
13.47	4.23	1.5898	1.5846	100.33
25.8	2.51	2.3763	2.3551	100.90
26.62	2.4	2.3948	2.3677	101.14
30.73	1.96	2.3902	2.3812	100.38

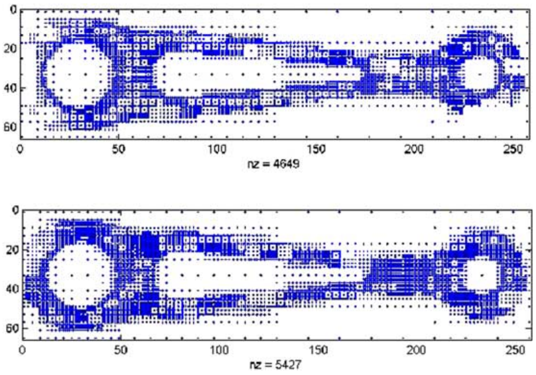


Fig. 15. The distribution of the adaptively-added wavelet bases for the torque arm example (upper: for the *x*-directional displacement, lower: for the *y*-directional displacement).

calculated by the proposed method. The stresses in Table 2 are calculated at the final iteration. The adaptively-located analysis nodes for the final optimized torque arm are also plotted in Fig. 15.

6. Conclusions

The interpolation-based wavelet-Galerkin method was successfully incorporated as the analysis tool for remesh-free shape optimization. The difference-checking nature of the interpolation wavelets allowed efficient adaptive analysis while the implementation of the wavelet method in a fictitious domain effectively avoided remeshing for numerical analysis. The idea of approximating the boundary curve by oblique lines that do not necessarily pass through the interpolation nodes was shown to improve the solution accuracy greatly and thus expedited the entire optimization process.

References

Bennett, J.A., Botkin, M.E., 1985. Structural shape optimization adaptive mesh refinement. AIAA J. 23, 458–464.

Belytschko, T., Lu, Y.Y., Gu, L., 1994. Element free Galerkin method. Int. J. Numer. Meth. Eng. 37, 229–256.

Chang, K.H., Choi, K.K., 1992. A geometry-based parameterization method for shape design of elastic solids. Mech. Struct. Mach. 20 (2), 215–252.

Christon, M.A., Roach, D.W., 2000. The numerical performance of wavelets for PDEs: the multi-scale finite element. Comput. Mech. 25, 230–244.

Cohen, A., Masson, R., 1999. Wavelet methods for second-order elliptic problems, preconditioning, and adaptivity. SIAM J. Sci. Comput. 21 (3), 1006–1026.

Diaz, A.R., 1999. A wavelet Galerkin scheme for analysis of large scale problems on simple domains. Int. J. Numer. Mech. Eng. 44, 1599–1616.

Ding, Y., 1986. Shape optimization of structures–a literature survey. Comput. Struct. 24 (6), 985–1004.

Garcia, M.J., Steven, G.P., 1998. Fixed grid finite elements in elasticity problems. Eng. Comput. 16 (2), 154–164.

Haslinger, J., Maitre, J.-F., Tomas, L., 2001a. Fictitious domains methods with distributed Lagrange multipliers. Part I: application to the solution of elliptic state problems. Math. Models Meth. App. Sci. 11 (3), 521–547.

Haslinger, J., Maitre, J.-F., Tomas, L., 2001b. Fictitious domains methods with distributed Lagrange multipliers. Part II: application to the solution of shape optimization problems. Math. Models Meth. Appl. Sci. 11 (3), 549–563.

Jang, G.W., Kim, J.E., Kim, Y.Y., 2004. Multiscale Galerkin method using interpolation wavelets for two-dimensional elliptic problems in general domains. Int. J. Numer. Meth. Eng. 59, 225–253.

Jang, G.W., Kim, Y.Y., Choi, K.K., 2002. Shape optimization using the wavelet-Galerkin method. The Second China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Busan, Korea, pp. 865–870.

Jang, G.W., Kim, Y.Y., Choi, K.K., 2003. Remesh-free shape optimization using the wavelet-Galerkin method and its applications to MEMS actuator design. WCSMO-5 Lido de Jesolo., pp. 121–122.

Kim, N.H., Choi, K.K., Botkin, M.E., 2002. Numerical method for shape optimization using meshfree method. Struct. Multidisciplinary Optimization 24 (6), 418–429.

Kim, Y.Y., Jang, G.W., 2002. Interpolation wavelet-based multi-scale Galerkin method for thin-walled box beam analysis. Int. J. Numer. Mech. Eng. 53, 1579–1592.

Kim, J.E., Jang, G.W., Kim, Y.Y., 2003. Adaptive multiscale wavelet-Galerkin analysis for plane elasticity problems and its applications to multiscale topology design optimization. Int. J. Solids Struct. 40, 6473–6496.

Lee, K., 1999. Principles of CAD/CAM/CAE Systems. Addison Wesley.

Liu, W.K., Jun, S., Zhang, Y.F., 1995. Reproducing kernel particle methods. Int. J. Numer. Meth. Fluids 20, 1081–1106.

Vanderplaats, G.N., 1997. DOT User's Manual. VMA Corp., Colorado Springs, CO.