

# 一种裂纹扩展的递推分析方法及其 Matlab 实现

朱旭程\*, 侯志强

(海军航空工程学院 飞行器工程系, 烟台 264001)

**摘要:** 利用映射函数法将裂纹扩展动域变分问题化为定域变分问题, 从最小势能原理出发推出裂纹扩展的递推积分方程, 进一步用有限元法求得递推代数方程, 用 Matlab 语言实现了裂纹扩展仿真算法, 该算法不必在每次裂纹扩展后中进行裂纹物体的重构和网格的重分, 有一定的实用价值。

**关键词:** 断裂力学; 区域变分; 有限元法; 裂纹扩展; Matlab  
**中图分类号:** O346.1      **文献标识码:** A

## 1 引言

计算机仿真已成为裂纹扩展分析的一种重要方法, 目前流行的几种有限元软件(如 ANSYS、FRANC 等)在分析这类问题时都是根据积分区域固定条件下的变分原理。由于裂纹扩展导致物体边界不断变化, 用这些软件分析时, 需要在每次裂纹变化之后进行物体的重绘和网格的重分, 难以满足一些特殊场合的要求。裂纹自动扩展问题实际上是一种积分区域变动条件下的变分问题。本文从最小势能原理出发, 利用动域变分原理<sup>[1]</sup>, 通过构造一坐标映射函数, 将扰动域积分变换为固定域积分, 化动域变分问题为固定域变分问题, 推出裂纹扩展的递推积分方程, 用有限元法离散后对其进行迭代计算即可进行裂纹扩展过程的仿真。由于采用动域变分方法, 在各轮迭代过程中不必进行裂纹物体的重绘和分网, 没有单元的生死变化, 因而简化了处理步骤。算法用 Matlab 语言实现, 易于根据具体情况变化使用。

## 2 数学模型的建立

### 2.1 物理量的坐标变换

考虑一带有初始裂纹  $a_0$  的弹性固体系统  $V_0$ , 在外载荷的作用下发生裂纹扩展, 扰动后物体系统的形状为  $V_1$ 。假设裂纹的扩展过程为一准静态平衡过程, 裂纹扩展量和物体形状的变化都是小扰动

量。为把扰动后区域上的物理量变换到扰动前区域中, 将动域积分泛函化为固定区域的积分, 构造一个坐标映射函数  $x_1 = F(x_0; a)$ , 此函数可将物体扰动前区域内的任意一点唯一地映射为扰动后区域上的一点, 进而将扰动后区域上的平衡方程变换为先前物体区域上的方程。

设映射函数  $x_1 = F(x_0; a)$  的具体形式为

$$x_1 = x_0 + a v(x_0) \quad (1)$$

式中

$$v(x_0) = \frac{\partial F}{\partial a}$$

式中  $x_0$  为扰动前区域内任意一点的坐标,  $x_1$  为扰动后区域内一点的坐标,  $a$  为裂纹扩展量, 扰动前后物体边界仅在裂纹尖端有  $a$  的微小变分。  $v(x_0)$  是定义在扰动前区域上的一个构造函数, 在量纲上可以看作一种位移场。函数  $v(x_0)$  的构造要求是: 函数的方向即裂纹方向, 函数在裂纹尖端处其范数是 1, 在物体边界(不包含裂纹)上其范数是 0, 在物体内部有连续的空间导数。此函数可采用多项式插值函数或其他锥形函数的形式, 在扰动前区域内以裂纹尖端为中心的一个适当大小(半径大于裂纹扩展增量)的圆形或多边形子域内构造。函数的构造具有任意性, 但采用简单的函数会简化计算, 提高计算速度。

设  $J$  为  $F$  的雅可比矩阵  $J = \nabla F$ , 对式(1)求导可得如下公式:

$$J = I + a \nabla v$$

$$J^{-1} = I + \sum_{n=1}^{\infty} (-1)^n a^n (\nabla v)^n \quad I - a \nabla v$$

$$|J| = 1 + a(\nabla \cdot v) + a^2 |\nabla v|^2 \quad |J|^{-1} = 1 - a(\nabla \cdot v) \quad (2)$$

收稿日期: 2007-03-01; 修改稿收到日期: 2007-09-04.

作者简介: 朱旭程\*(1973-), 男, 博士, 讲师

(E-mail: zhuxucheng1973@yahoo.com)

侯志强(1959-), 男, 博士, 教授.

根据弹性力学理论,扰动前后物体的应力、应变和位移应满足几何方程和本构方程<sup>[2]</sup>,则有下列关系成立:

$$\sigma_0 = \nabla u_0, \quad \sigma_1 = E \epsilon_0, \quad \epsilon_1 = \nabla u_1, \quad \epsilon_1 = E \epsilon_1 \quad (3)$$

式中  $\sigma_0, \epsilon_0$  和  $u_0$  为扰动前物体应力、应变和位移,  $\sigma_1, \epsilon_1$  和  $u_1$  为扰动后物体应力、应变和位移,  $E$  为材料的弹性张量。

利用式(1)中的映射关系,可将式(3)中扰动后区域中的物理量变换到扰动前区域中,得

$$\begin{aligned} u &= u_1(x_1) = u_1, \quad F(x_0) = u(x_0) \\ &= E \nabla u J^{-1} \\ &= E(\nabla u - a \nabla u_0 \nabla v) \end{aligned} \quad (4)$$

式中  $\sigma, u$  为  $\sigma_1, u_1$  经过坐标映射变换到扰动前空间中得到的应力和位移,  $\sigma$  和  $u$  为扰动前后应力和位移变分,其定义为

$$\sigma = \sigma_0 + \sigma, \quad u = u_0 + u$$

### 2.2 位移变分的计算

在准静态平衡假设条件下扰动前后物体区域中的位移都满足弹性力学中的平衡条件,弹性体在内部和边界的平衡条件可以用最小势能原理来表示,即势能变分取驻值:

$$\begin{aligned} \delta \Pi &= \int_{V_0} E \nabla u_0 : \nabla u dV - \int_{V_0} f_0 \cdot u dV - \int_{S_1} p_0 \cdot u ds = 0 \\ \delta \Pi &= \int_{V_1} E \nabla u_1 : \nabla u_1 dV - \int_{V_1} f_1 \cdot u_1 dV - \int_{S_1} p_1 \cdot u_1 ds = 0 \end{aligned} \quad (5)$$

式中  $f_i$  为作用在物体上的体积力,  $p_i$  为边界载荷,  $S_1$  为载荷作用边界,  $S_2$  为位移约束边界,设位移为固定约束。用式(1)对扰动后物体的能量变分进行积分域变换,将式(2)和式(3)代入上式,减去式(5),再利用式(4)消去应力变分,可得

$$\begin{aligned} \delta \Pi &= \int_{V_0} E \nabla u : \nabla u dV - \int_{V_0} f \cdot u dV - \int_{V_0} (\nabla \cdot v) f_0 \cdot u dV + \int_{S_1} p \cdot u ds = 0 \end{aligned} \quad (6)$$

式中

$$r_0 = E \left[ \nabla u_0 \nabla v^T - \nabla u_0 \nabla \cdot v + \frac{\nabla u_0 \nabla v + (\nabla u_0 \nabla v)^T}{2} \right]$$

式中  $p$  为边界载荷变化量,  $f$  为体载荷变化量。利用积分方程:

$$\int_{V_0} E \nabla u : \nabla u dV = \int_{V_0} f \cdot u dV + \int_{S_1} p \cdot u ds + \int_{V_0} r_0 : \nabla u dV + \int_{V_0} (\nabla \cdot v) f_0 \cdot u dV \quad (7)$$

当给定裂纹扩展量后,即可计算出位移变分  $u$ 。

### 2.3 积分方程的离散

将整个固体系统进行网格划分,离散后的系统由有限个单元体组合而成。采用分片函数的形式构造待解位移函数,待解位移可写成以节点位移为参数的插值函数形式:

$$u = N_e U_e, \quad u = N_c U_c$$

式中  $N_e$  为单元试探函数矩阵,  $U_e$  为单元节点位移向量,  $u$  为连续的位移函数。

将上式代入离散后的积分方程,采用伽辽金法<sup>[4]</sup>在每个单元体上积分,可得单元方程:

$$[K^e] \{U^e\} = \{F^e\} - a \{L^e\} \quad (8)$$

式中

$$\begin{aligned} K^e &= \int_{V_e} \nabla N_e^T E \nabla N_e dV \\ L^e &= - \int_{V_e} \nabla N_e^T r_0 dV - \int_{V_0} (\nabla \cdot v) N_e^T f_0 dV \\ F^e &= \int_{S_e} N_e^T p ds + \int_{V_e} N_e^T f dV \end{aligned}$$

将各单元方程按节点顺序进行装配,形成整体方程:

$$[K] \{U\} = \{F\} - a \{L\} \quad (9)$$

式中

$$K = \int_{V_e} K^e, \quad L = \int_{V_e} L^e, \quad F = \int_{V_e} F^e$$

设在第  $k$  步求解上式得到的位移增量为  $U(k)$ , 上一步得到的位移量  $U(k-1)$ , 则可递推计算本步的位移量:

$$U(k) = U(k-1) + U(k)$$

## 3 求解算法的 Matlab 实现

现以一个双悬臂梁试样为例,分析并验证上述方法的可行性。试样为一个中间带裂纹的矩形梁,其高  $h = 1.2 \text{ mm}$ , 宽度  $w = 4 \text{ mm}$ , 初始裂纹长度  $a = 1.5 \text{ mm}$ , 材料的弹性模量  $E = 200 \text{ GPa}$ , 泊松比  $\nu = 0.3$ , 载荷施加在左端小圆(半径  $0.1 \text{ mm}$ )上, 初始载荷  $p_0 = 100 \text{ N/mm}$ , 载荷变化量  $p = -5$

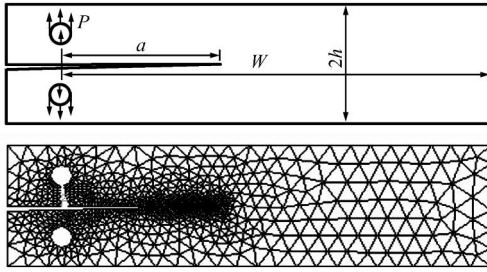


图 1 双悬臂梁试样外形及网格划分  
Fig. 1 Geometric configuration & mesh of DCB

N/mm, 其几何外形及网格划分如图 1 所示。利用式 (9) 采用 Matlab 编制有限元算法, 可进行连续的裂纹扩展仿真, 程序步骤大致如下。

### 3.1 分析裂纹物体的初始节点位移

采用绘图软件绘制含有初始裂纹物体的几何模型, 采用三角形单元进行网格划分, 施加载荷和位移约束, 设置弹性模量  $E$ 、泊松比等偏分方程系数, 设置求解器参数并求解初始节点位移场  $U_0$ , 图 2 显示了试样初始裂纹长度为 1.5 mm 时的应力分布。

保存得到的单元体和节点坐标数据。其中, 文件 elements3.dat 保存每个单元的顶点节点号, 每行格式为 (element # node1 node2 node3)。coordinates.dat 保存每个节点的坐标, 每行格式为 (node # coordinatex coordinatex)。neumann.dat 保存每个力边界边元的端点节点号, 每行格式为 (Nedge # node1 node2)。dirichlet.dat 保存每个位移边界边元的端点节点号, 每行格式为 (Dedge # node1 node2)。U.dat 保存每个节点的初始位移, 每行格式为 (node # Variable # Value)。算法的初始化部分从数据文件读入所得的数据文件信息, 实现程序如下:

```
% 初始化变量:
load coordinates.dat; coordinates(:,1) = [];
load elements3.dat; elements3(:,1) = [];
load u.dat; u(:, [1 2]) = [];
load neumann.dat; neumann(:,1) = [];
load dirichlet.dat; dirichlet(:,1) = [];
w = 4; a = 1.5; p = 100; E = 2E5; Nu = 0.3;
D = E / (1 - Nu * Nu) * [1 Nu 0 0; Nu 1 0 0; 0 0
(1 - Nu) / 2 0; 0 0 0 (1 - Nu) / 2];
FreeNodes = setdiff(1 : size(coordinates, 1),
unique (dirichlet));
% 递推方程迭代:
```

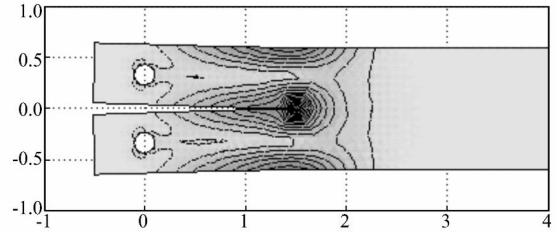


图 2 裂纹长度为 1.5 mm 时的应力分布  
Fig. 2 Stress contour when  $a = 1.5$  mm

While  $a < w$

```
K = sparse(2 * size(coordinates, 1),
2 * size(coordinates, 1));
b = sparse(2 * size(coordinates, 1), 1);
```

### 3.2 计算单元刚度矩阵

单元刚度矩阵计算公式为

$$K^e = \int_{V_e} \nabla N_e^T E \nabla N_e dV = \frac{L T L}{2} [B]^T [D] [B] \quad (10)$$

式中

$$T = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix}$$

$$b_i = y_{i+1} - y_{i+2}, \quad c_i = x_{i+2} - x_{i+1}$$

$$\nabla N_e = [B] = \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix}$$

$$E = [D] = \frac{E}{1 - \frac{\nu^2}{2}} \begin{bmatrix} 1 & 0 & 0 \\ & 1 & 0 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} & 0 \\ 0 & 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix}$$

矩阵  $N_e$  由单元顶点  $(x_i, y_i)$  的坐标确定, 采用 Function Ngradient = DN. e(vertices) 计算。

### 3.3 计算裂纹扩展量

根据断裂力学中裂纹扩展的 Griffith 能量准则, 当能量释放率达到裂纹阻力 (即  $G = G_c$ ) 时发生裂纹扩展, 裂纹阻力是一个与材料有关的常数, 表示了形成单位面积裂纹需要的断裂能。文献 [3] 中阐述了双悬臂梁试样的能量释放率和裂纹大小的理论关系, 并用实验进行了验证:

$$G_c = \frac{12 P^2 a^2}{b^3 h^3 E} = > a = - a \frac{P}{p} \quad (11)$$

式中  $p/P$  为载荷变化率。裂纹扩展量的计算程序如下:

```
%Crack increment
```

```
Da = [0 - a] * D. p (sum(coordinates
    (neumann(1, :, :))/2)/p;
p = p + [0 1] * D. p (sum(coordinates
    (neumann(1, :, :))/2); a = a + Da;
```

### 3.4 计算右端向量

映射函数的构造采用一个二次圆锥曲面作为母函数:

当  $0 < x < 0.5$  时,  $V(x) = -2x^2 + 1$ ;  
 当  $0.5 < x < 1$  时,  $V(x) = 2(x - 1)^2$ ;  
 其他情况下,  $V(x) = 0$ 。

对母函数缩放和平移使函数中心与裂纹尖端重合,使函数的非零域不超出物体边界,得到映射函数  $V(x)$ 。函数  $V(x)$  由 function  $Vdisplace = V(x)$  给出。

裂纹扩展向量的计算公式为

$$\int_{V_e} \nabla N^T r_0 dV = - [B]^T [D] t_0(x_s, y_s)$$

$$\int_{V_0} N^T s_0 dV = - \frac{[T]}{6} s_0(x_s, y_s) \quad (12)$$

式中

$$\{ \nabla u_0 \} = [B] \{ U_0 \}$$

$$t_0 = \nabla u_0 \nabla v^T - \nabla u_0 \nabla \cdot v + \frac{\nabla u_0 \nabla v + (\nabla u_0 \nabla v)^T}{2}$$

$$\{ s_0 \} = (\text{div} v) \{ f_0 \}$$

式中  $(x_s, y_s)$  为单元体中心坐标。函数  $t_0$  由 function  $T0Strain = T_0(x)$  给出,函数  $s_0$  由 function  $S0Force = S_0(x)$  给出。

体积力向量的计算公式为

$$\int_{V_e} N^T f dV = \frac{[T]}{6} f(x_s, y_s) \quad (13)$$

体积力变分由 function  $DForce = D \cdot f(x)$  给出。

边界力向量的计算公式为

$$\int_{s_e} N^T p ds = \frac{[S_e]}{2} p(x_m, y_m) \quad (14)$$

式中  $[s_e]$  为边元长度,  $(x_m, y_m)$  为边元中心坐标。边界力由 function  $DForce = D \cdot p(x)$  给出。

### 3.5 计算与装配矩阵和向量

采用循环语句计算与装配各单元体的刚度矩阵、裂纹扩展向量、体积力向量和边界力向量,形成整体矩阵和向量,具体程序如下:

```
% 装配矩阵和向量
for j = 1 : size(elements3, 1)
indE = [2 * elements3(j, :) - 1,
    2 * elements3(j, :)];
vertices = coordinates(elements3(j, :), :);
```

```
T = [ones(1, 3); vertices'];
N = [eye(2), eye(2), eye(2)];
DN = DN. e(vertices);
% 装配刚度矩阵
K(indE, indE) = K(indE, indE) + det(T) *
    DN' * D * DN / 2;
% 装配裂纹扩展向量
b(indE) = b(indE) + Da * det(T) * N' * S. 0
    (sum(vertices)/3)/6 + Da * DN' * D * T. 0
    (sum(vertices)/3)/6;
% 装配体积力向量 Volume Forces
b(indE) = b(indE) + det(T) * N' * D. f(sum
    (vertices)/3)/6;
end
% 装配力边界向量
for j = 1 : size(neumann, 1)
indN = [2 * neumann(j, :), 2 * neumann(j, :)
    + 1];
b(indN) = b(indN) + norm(coordinates
    (neumann(j, 1), :) - coordinates
    (neumann(j, 2), :)) * [eye(2); eye(2)] *
    D. p (sum(coordinates
    (neumann(j, :), :))/2) / 2;
end
```

### 3.6 处理位移约束条件和求解方程

总体方程  $[K] \{ U \} = \{ F \} - a \{ L \}$  可写成如下的分块形式:

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} Du \\ Du_D \end{Bmatrix} = \begin{Bmatrix} b \\ b_D \end{Bmatrix} \quad (15)$$

式中  $Du$  为自由节点的位移,  $Du_D$  为位移边界上的节点的位移,对于固定边界  $Du_D$  为 0。舍去第二块方程,得到约简后的方程为

$$K_{11} \cdot Du = b \quad (16)$$

将自由节点对应的元素从总体方程中抽取出来得第一块方程,采用左逆运算符 \ 求解。

```
Du = sparse(2 * size(coordinates, 1), 1);
% 位移约束条件和求解方程
FreeN = [2 * FreeNodes - 1, 2 * FreeNodes];
Du(FreeN) = K(FreeN, FreeN) \ b(FreeN);
```

### 3.7 计算扰动后的位移场

利用  $x_1 = x_0 + aV(x_0)$  映射关系把各节点的当前坐标变换为扰动后的节点坐标,由  $U = U_0 + U$  计算出扰动后物体各节点的位移场;返回 3.2 开始

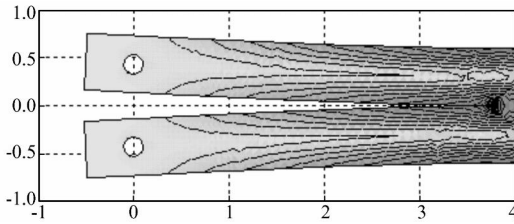


图 3 裂纹长度为 3.75 mm 时的应力分布  
Fig.3 Stress contour when  $a = 3.75$  mm

下一轮裂纹扩展分析。图 3 显示了试样裂纹长度为 3.75 mm 时的应力分布。

% 后处理

$u = u + Du;$

$coordinates(:,1) = coordinates(:,1) + Da * V(coordinates(:,1));$

end

### 3.8 实例验证

为了验证上述程序的计算结果,在有限元软件 ANSYS10.0 中对实例进行了对比计算。每次用常规方法在 ANSYS10.0 中绘出裂纹样本,采用平面应变三角型单元 PLANE2 进行分网,求解后提取所得的最大节点位移  $U_{y\max}$ ,与前文 Matlab 程序的计算结果对比分析如图 4 所示。可见,两种计算结果很接近。而统计分析估计,ANSYS 划分网所花时间占整个计算时间的 60% 左右。

## 4 结 语

本文利用映射函数法和动域变分原理推导了一种裂纹扩展过程计算的递推方程。由于采用了

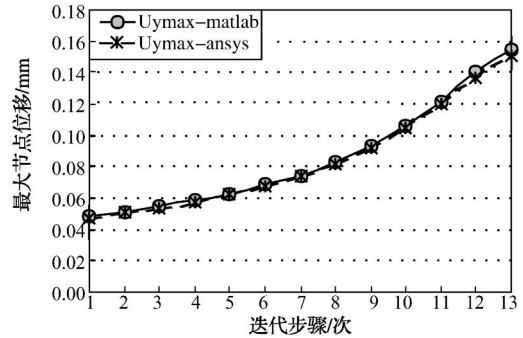


图 4 迭代过程中的最大节点位移  
Fig.4 Maximum node displacement during iterations

映射函数,裂纹扩展变化之后物体不需要再进行网格的重新划分,只需通过连续迭代递推方程即可仿真裂纹的扩展过程,简化了处理步骤,因而算法有一定的实用价值。

## 参考文献(References):

- [1] WOLFGANG A. Uniform convergence for elliptic problems on varying domains [J]. *Mathematische Nachrichten*, 2007, **280**(1): 28-49.
- [2] 郭日修. 弹性力学与张量分析[M]. 武汉: 湖北科学技术出版社, 2001. (GUO Ri-xiu. *Tensor Analysis and Elasticity Mechanics* [M]. Wuhan: Hubei Science Press, 2001. (in Chinese))
- [3] SANFORD R J. *Principles of Fracture Mechanics* [M]. Prentice Hall, 2003.
- [4] ZIENKIEWICZ O C. *The Finite Element Methods* [M]. Butterworth Heinemann, 2006.

## A recursive crack propagation method and its matlab implementation

ZHU Xu-cheng\*, HOU Zhi-qiang

(Department of Aircraft Engineering, NAEI, Yantai 264001, China)

**Abstract:** The crack automatic growth problem is solved by transforming a domain variation integration into the normal one using a mapping function. The iteration equations for the continuum crack propagation is formulated according to the minimum potential principle. The Matlab implementation is demonstrated for crack growth simulation by the finite element method, and there is no need of updating and remeshing the configuration. This approach has some value in engineering application.

**Key words:** fracture mechanics; domain variation; FEM; crack propagation; Matlab