ABAQUS 静/动力隐式UEL 二次开发 ABAQUS Static/Dynamic Implicit UEL Secondary Development

有限元先生 Finite Element Master

更新于2024年8月29日 06:43 Updated on August 29, 2024, 06:43

概述：结合HHT时程积分法，推导了ABAQUS **静/动力隐式算法**中的UEL关键矩阵表达式，并将公式应用到自编CPS4/CPE4、C3D8 BBAR和C3D20用户自定义单元中，计算结果均与ABAQUS自带单元保持一致。其中，静力计算中，关键矩阵AMARTX和RHS等可直接按照刚度矩阵和方程右端不平衡力输出。动力隐式计算中，这两者的输出较为复杂，需要结合HHT时程积分法进行推导，将刚度矩阵、质量矩阵和阻尼矩阵依据LFLAGS数组的数值进行组合，RHS同样需要进行推导计算，并以合适的方法将解相关的状态量储存在SVARS中，供后面的增量步调用。

Overview: By combining the HHT time-history integration method, the key matrix expression in the ABAQUS static/dynamic implicit algorithm is derived, and the formula is applied to the self-written CPS4/CPE4, C3D8 BBAR, and C3D20 user-defined elements. The calculated results are consistent with the ABAQUS built-in elements. In static calculations, the key matrices AMARTX and RHS can be directly output as the stiffness matrix and the right-hand side unbalanced force. In dynamic implicit calculations, the output of these two is more complex and requires derivation in combination with the HHT time-history integration method. The stiffness matrix, mass matrix, and damping matrix should be combined according to the values of the LFLAGS array, and the RHS also needs to be derived and calculated. The state variables related to the solution should be stored in SVARS in an appropriate manner for use in subsequent incremental steps.

关于UEL的程序设计，只支持静力通用计算分析步的资料有很多，本帖子内容适不但适用于静力通用，而且适用于动力隐式、频率分析等分析步计算，采用模块化程序设计，所有的矩阵求解均被封装，调用方便，读者可以针对自己的需求对相应函数进行改编即可，尤其是涉及到动力隐式计算部分，**适用于任何运动方程的动力隐式求解，可直接移植使用。**

Regarding the program design of UEL, there are many materials that only support static general calculation steps. The content of this post is not only applicable to static general but also to dynamic implicit, frequency analysis, and other calculation steps. It adopts modular program design, and all matrix solutions are encapsulated, making it convenient to call. Readers can modify the corresponding functions according to their own needs. Especially for the part involving dynamic implicit calculations, it is applicable to the dynamic implicit solution of any motion equation and can be used directly.

（）UEL接口基本参数介绍 Introduction to Basic Parameters of UEL Interface

----

用户自定义单元(USER DEFINED ELEEMNT, UEL)适合进阶的工程师/学者使用，u、UEL的目的是实现一个单元的力学行为，即力-位移关系，这部分的功能需要的编程工作量和理论功底都比较高， 涉及到的知识大致包括：

User-defined element (USER DEFINED ELEEMNT, UEL) is suitable for advanced engineers/scholars. The purpose of UEL is to realize the mechanical behavior of an element, that is, the force-displacement relationship. This part requires a relatively high amount of programming work and theoretical foundation, and involves knowledge such as:

有限元理论：形函数插值、应力-应变关系，刚度矩阵组装，质量矩阵求解，阻尼矩阵求解，

Finite element theory: shape function interpolation, stress-strain relationship, assembly of stiffness matrix, solution of mass matrix, solution of damping matrix,

数值算法：高斯积分（全积分、缩减积分）、非线性方程组迭代求解流程（增量迭代，常刚度迭代等）、雅各比矩阵求解

Numerical algorithms: Gaussian integration (full integration, reduced integration), iterative solution process of nonlinear equations (incremental iteration, constant stiffness iteration, etc.), and solution of the Jacobian matrix

弹/塑性力学：应力-应变关系，位移-应变关系 Elastic/plastic mechanics: stress-strain relationship, displacement-strain relationship

熟练的FORTRAN编程操作是基础技能。 Proficient FORTRAN programming skills are fundamental.

UEL接口参数众多，包括自己本身的参数和与其他子程序联合的参数，下面介绍我比较重要的几个参数。

The UEL interface has many parameters, including its own parameters and parameters combined with other subroutines. Below, I will introduce several important parameters.

<>需要编程定义的参数 Parameters that need to be programmed-defined

RHS（right hand side）：这个命名是从方程组的角度来的，顾名思义，他就是方程的右端量，其本质是：外力-内力，外力部分程序的编写涉及到与其他子程序的联合使用，包括DLOAD和UTRACLOAD等等，这部分内容是给用户自定义单元施加复杂的广义力，UEL接口为其提供了相应的参数，如JDLTYP、NDLOAD等等，目前没有做过尝试。至于内力的求解，在静力线性计算中，数值上等于-KU（后面有这个公式的推导），即刚度与位移乘积负数。在动力隐式分析中，这个矩阵需要结合HHT时程积分法推导具体的表达式。

RHS (right-hand side): This naming comes from the perspective of the system of equations, as the name implies, it is the right-hand quantity of the equation, its essence is: external force - internal force. The programming of the external force part involves the joint use of other sub-programs, including DLOAD and UTRACLOAD, etc. This part of the content is for users to apply complex generalized forces to the custom unit, and the UEL interface provides the corresponding parameters, such as JDLTYP, NDLOAD, etc., which have not been tried yet. As for the solution of internal forces, in static linear calculations, it is numerically equal to -KU (the derivation of this formula is later), that is, the negative product of stiffness and displacement. In implicit dynamic analysis, this matrix needs to be combined with the HHT time history integration method to derive the specific expression.

AMATRX：这个参数往往被认为是刚度，但并不这样，他只有在静力计算的时候才是刚度。他的具体取值依据分析类型的不同而不同，具体的表达形式不唯一。在动力隐式分析中，这个矩阵需要结合HHT时程积分法推导具体

的表达式。

AMATRX: This parameter is often considered to be stiffness, but it is not. It is only stiffness when static calculations are performed. Its specific value depends on the analysis type, and the specific expression form is not unique. In implicit dynamic analysis, this matrix needs to be combined with the HHT time history integration method to derive the specific expression.

SVARS：这个参数官方文档说是取决于结果的状态量，他的具体意义和数据由我们自己确定，而我把它理解为一个小仓库，可以存放我们的数据。需要注意的是里面的数据可以在不同的增量步之间传输，就是说，里面的数据会传到下一个增量步，只要不更新他，他可以一直被使用。这个数据我们自己决定更新与否。

SVARS: The official documentation says that this parameter depends on the state quantity of the result, and its specific meaning and data are determined by ourselves. I understand it as a small warehouse where we can store our data. It should be noted that the data inside can be transmitted between different increment steps, that is, the data will be transmitted to the next increment step, and as long as it is not updated, it can be used continuously. Whether to update this data is decided by ourselves.

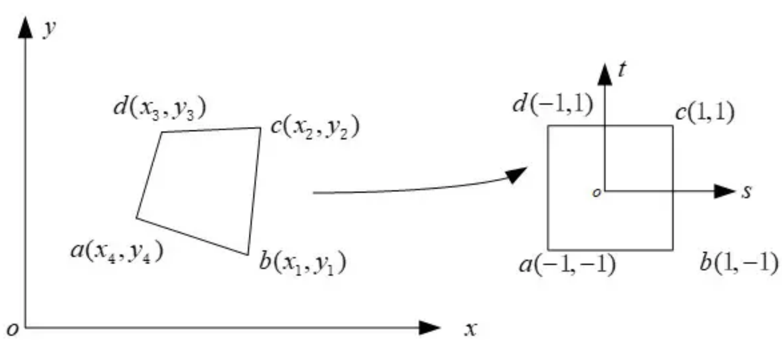
ENERGY：与用户自定义单元相关的一些动量等能量，也是依据不同的分析类型而不同。

ENERGY: Some momentum and energy related to user-defined elements, which also vary with different analysis types.

() CPS4/CPE4公式推导 () Derivation of CPS4/CPE4 formulas

等参单元中，母单元向笛卡尔坐标的转换为：（*注意：这里的箭头标反了*）

In isoparametric elements, the transformation of the parent element to Cartesian coordinates is: (Note: The arrow is reversed here)



要再现位移场，可假设单元内部位移为坐标的函数，即：

To reproduce the displacement field, it can be assumed that the internal displacement of the element is a function of the coordinates, i.e.:

$$u = a_1 + a_2 x + a_3 y + a_4 xy$$
$$v = a_5 + a_6 x + a_7 y + a_8 xy$$

将四个节点的位移和坐标代入上式，有： Substitute the displacements and coordinates of the four nodes into the above equation, and we have:

$$
\begin{aligned}
&u_1 = a_1 + a_2 x_1 + a_3 y_1 + a_4 x_1 y_1 \\
&u_2 = a_1 + a_2 x_1 + a_3 y_1 + a_4 x_1 y_1 \\
&\cdots \\
&v_1 = a_5 + a_6 x_1 + a_7 y_1 + a_8 x_1 y_1 \\
&v_1 = a_5 + a_6 x_2 + a_7 y_2 + a_8 x_2 y_2 \\
&\cdots
\end{aligned}
$$

如此，求解系数，并用节点位移表示单元内部位移： Thus, solve for the coefficients and express the internal displacement of the element in terms of nodal displacements:

$$
\begin{aligned}
u &= \frac{1}{4}\left[(1-x)(1-y)u_1 + (1+x)(1-y)u_2 + (1+x)(1+y)u_3 + (1-x)(1+y)u_4\right] \\
v &= \frac{1}{4}\left[(1-x)(1-y)v_1 + (1+x)(1-y)v_2 + (1+x)(1+y)v_3 + (1-x)(1+y)v_4\right]
\end{aligned}
$$

上式即为单元节点位移与坐标的插值关系。考虑到等参单元，即位移的插值函数与母单元-笛卡尔单元变换采用相同插值函数。则等参变化的插值函数为：

The above equation represents the interpolation relationship between the element nodal displacements and coordinates. Considering the isoparametric element, that is, the interpolation function for displacements and the Cartesian element transformation use the same interpolation function. Then, the interpolation function for isoparametric variation is:

$$
\begin{aligned}
x &= \frac{1}{4}\left[(1-s)(1-t)x_1 + (1+s)(1-t)x_1 + (1+s)(1+t)x_1 + (1-s)(1+t)x_1\right] \\
y &= \frac{1}{4}\left[(1-s)(1-t)y_1 + (1+s)(1-t)y_1 + (1+s)(1+t)y_1 + (1-s)(1+t)y_1\right]
\end{aligned}
$$

上式的矩阵表达为： The matrix expression of the above equation is:

$$
\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} N_1 & & N_2 & & N_3 & & N_4 & \\ & N_1 & & N_2 & & N_3 & & N_4 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix}
$$

其中： Among them:

$$
N_i = \frac{(1+s_i s)(1+t_i t)}{4} (i = 1 \sim 4)
$$

对形函数求导，得形函数对母单元坐标的导数： Derive the shape functions to obtain the derivative of the shape functions with respect to the parent element coordinates:

$$
\begin{aligned}
\frac{\partial N_i}{\partial s} &= \frac{(1+s_i)(1+t_i t)}{4} (i = 1 \sim 4) \\
\frac{\partial N_i}{\partial t} &= \frac{(1+s_i s)(1+t_i)}{4} (i = 1 \sim 4)
\end{aligned}
$$

由弹性力学几何方程，应变矩阵为： According to the geometric equations of elasticity, the strain matrix is:

$$B = LN = \begin{bmatrix} \frac{\partial}{\partial x} & \\ & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1 I & N_2 I & N_3 I & N_4 I \end{bmatrix}, I_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

合并为： Merge into:

$$B = \begin{bmatrix} \frac{\partial N_1}{\partial x} & & \frac{\partial N_2}{\partial x} & & \frac{\partial N_3}{\partial x} & & \frac{\partial N_4}{\partial x} & \\ & \frac{\partial N_1}{\partial y} & & \frac{\partial N_2}{\partial y} & & \frac{\partial N_3}{\partial y} & & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}_{3\times 8}$$

JOCABIN矩阵为： JOCABIN matrix is:

$$\begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial t} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = J \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix}$$

其中： Among them:

$$x = \sum_{i=1}^{4} N_i x_i, y = \sum_{i=1}^{4} N_i y_i$$

则JOCABIN的具体表达式为： The specific expression of JOCABIN is:

$$J = \begin{bmatrix} \frac{\partial N_1}{\partial s} & \frac{\partial N_2}{\partial s} & \frac{\partial N_3}{\partial s} & \frac{\partial N_4}{\partial s} \\ \frac{\partial N_1}{\partial t} & \frac{\partial N_2}{\partial t} & \frac{\partial N_3}{\partial t} & \frac{\partial N_4}{\partial t} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

刚度矩阵的求解： Solution of the stiffness matrix:

$$K = \iiint_v B^T DBdv = \sum_i \sum_j H_i H_i B^T DB \left| J \right|$$

质量矩阵分为两种，集中质量矩阵和协调质量矩阵，下面给出的是一致质量矩阵（又叫协调质量矩阵）矩阵，该种质量矩阵因其采用了与位移插值相同的插值函数而被称为一致质量矩阵。但是协调质量矩阵为满阵，不利于数值计算，所以有限元中一般采用集中质量矩阵，即只有对角元素不为零。集中质量矩阵通常由协调质量矩阵处理而来，

比如将协调质量矩阵每一行的质量集中到对角元素，然后将非对角元素数值置为零。

The mass matrix is divided into two types, the concentrated mass matrix and the consistent mass matrix. The following is the consistent mass matrix (also known as the consistent mass matrix) matrix. This type of mass matrix is called consistent mass matrix because it uses the same interpolation function as the displacement interpolation. However, the consistent mass matrix is a full matrix, which is not conducive to numerical computation. Therefore, in finite element analysis, the concentrated mass matrix is generally used, that is, only the diagonal elements are not zero. The concentrated mass matrix is usually derived from the consistent mass matrix, such as concentrating the mass of each row of the consistent mass matrix to the diagonal element, and then setting the non-diagonal elements to zero.

$$K = \iiint_v B^T DBdv = \sum_i \sum_j H_i H_i B^T DB \left| J \right|$$

阻尼矩阵，采用比例阻尼： Damping matrix, using proportional damping:

$$C = \alpha * K + \beta * M$$

UEL主程序采用常刚度法设计，刚度矩阵和质量矩阵只在第一个增量步计算一次，然后储存在变量SVARS中，后面一直到分析结束之间的所有增量步不再计算，均直接读取SVARS数组中的数据并输出会ABAQUS主程序。**该主程序将各个关键量的求解分为不同的子程序，这些子程序基本是所有用户自定义单元都会涉及的，因此该主程序的可移植性很强，在线弹性问题中，即可用常刚度迭代法求解的运动方程中，读者完全可以按照这个直接编写自己的程序。**

The UEL main program is designed using the constant stiffness method. The stiffness matrix and mass matrix are only calculated once in the first increment step, then stored in the variable SVARS. From then on until the end of the analysis, all increments are no longer calculated, and the data from the SVARS array is directly read and output to the ABAQUS main program. This main program divides the solution of various key quantities into different subroutines, which are basically involved in all user-defined elements. Therefore, the portability of this main program is very strong. In linear elastic problems, where the motion equations that can be solved using the constant stiffness iterative method, readers can completely write their own programs according to this.

```
      SUBROUTINE UEL(RHS,AMATRX,SVARS,ENERGY,NDOFEL,NRHS,NSVARS,
     1 PROPS,NPROPS,COORDS,MCRD,NNODE,U,DU,V,A,JTYPE,TIME,DTIME,
     2 KSTEP,KINC,JELEM,PARAMS,NDLOAD,JDLTYP,ADLMAG,PREDEF,NPREDF,
     3 LFLAGS,MLVARX,DDLMAG,MDLOAD,PNEWDT,JPROPS,NJPROP,PERIOD)
C
       INCLUDE 'ABA_PARAM.INC'

C
       DIMENSION RHS(MLVARX,*),AMATRX(NDOFEL,NDOFEL),PROPS(*),
     1 SVARS(*),ENERGY(8),COORDS(MCRD,NNODE),U(NDOFEL),
     2 DU(MLVARX,*),V(NDOFEL),A(NDOFEL),TIME(2),PARAMS(*),
     3 JDLTYP(MDLOAD,*),ADLMAG(MDLOAD,*),DDLMAG(MDLOAD,*),
     4 PREDEF(2,NPREDF,NNODE),LFLAGS(*),JPROPS(*)
       double precision DMATX(3,3)
       double precision KK(2*nnode,2*nnode),MM(2*nnode,2*nnode)
       KK=0.D0
```

```
      MM=0.D0
!     Print work state on screen and ANA_STAT.txt
      CALL PRINTSTATE(JPROPS,LFLAGS,DTIME,JELEM)
      IF(KSTEP.EQ.1.AND.KINC.EQ.1) THEN
!        material martix
         CALL DMATERIAL(DMatx,PROPS)
!        stiffness martix
         call KKmartix(KK,coords,DMatx,mcrd,nnode,jelem)
!        mass martix
         call MMmartix(MM,coords,props,mcrd,nnode,jelem,dmatx)
!        Store pre-calculated stiff/mass matrices
         CALL STOREMATRICES (SVARS,NSVARS,KK,MM,NDOFEL)
      else
         CALL READMATRICES(SVARS,NSVARS,KK,MM,NDOFEL)
      endif
!     Output required variables
      CALL OUTPUTVARIABLE(RHS,AMATRX,SVARS,PROPS,ENERGY,U,V,A,
     1 LFLAGS,DTIME,NDOFEL,NRHS,NSVARS,MLVARX,JELEM,PARAMS,KK,MM)
      RETURN
      END
```

不想写文字了，麻烦，直接把付费内容放在附件里面了：

Don't want to write text anymore,麻烦, just put the paid content in the attachment:

（1）CPE4/CPS4单元UEL，适用于静/动力隐式计算

(1) CPE4/CPS4 element UEL, applicable to static/dynamic implicit calculations

（2）C3D8 BBAR单元UEL，，适用于静/动力隐式计算，包含UMAT给应力应变可视化部分

(2) C3D8 BBAR element UEL, applicable to static/dynamic implicit calculations, including UMAT for stress-strain visualization

（3）C3D20 单元UEL，适用于静/动力隐式计算

(3) C3D20 element UEL, applicable to static/dynamic implicit calculations

（4）WORD文档，推导、讲解动力隐式计算中的AMARTX、RHS

(4) WORD document, derivation and explanation of AMARTX and RHS in dynamic implicit calculations

赠送内容： Gift Content:

（5）收集的一些UEL-UMAT程序 (5) Some collected UEL-UMAT programs

（6）一对一辅导，包括理论、公式和编程等……

(6) One-on-one tutoring, including theory, formulas, and programming, etc...

动力隐式计算中的KK、MM、CC、RHS、SVARS矩阵计算。C3D8_BBAR公式推导及UEL设计。C3D20 UEL设计。CPS4/CPE4 UEL设计。 Kinematic implicit calculation of KK, MM, CC, RHS, SVARS matrices. Derivation of C3D8_BBAR formula and UEL design. C3D20 UEL design. CPS4/CPE4 UEL design.

推荐阅读 Recommended Reading