CESAR-LCPC
version 4.x

CONCEPTION LOGICIELLE

Format des fichiers de résultats .rsv4

2e édition **Mai 2007**

Référence : VS4x-DCL_Fichiers-rsv4_02

Auteur(s) : Pierre Humbert (LCPC-DPr/MN)
Gérard Fezans (Itech)

Nomenclature d'activité : 11P052

LCPC **Etablissement Public national à caractère Scientifique et Technologique**
Paris 58, boulevard Lefebvre - 75732 Paris cedex 15
Nantes Route de Bouaye - BP 4129 - 44341 Bouguenais cedex
Marne-la-Vallée LMSGC - Cité Descartes, Parc Club de la Haute Maison
2, allée Kepler - 77420 Champs-sur-Marne
Satory LIVIC - Bâtiment 140 - 13 route de la Minière - Satory - 78000 Versailles
Internet www.lcpc.fr

Distribution et maintenance CLEO :
ITECH Informatique et technologie (SARL)
12 - 16, rue de Vincennes - 93100 MONTREUIL (France)
www.itech-soft.com

Notes relatives à la présente édition :

2e édition : *Mai 2007*
Intégration dans les documents de conception logicielle de CESAR-LCPC version 4.x.

Notes relatives aux éditions précédentes :

1re édition : *Octobre 2002*
Édition issue du dossier de conception des travaux CLEO-ext2 (1999-2002), sous le titre initial "Format of Results files".

CESAR-LCPC version 4.x

CONCEPTION LOGICIELLE

Format des fichiers de résultats .rsv4

Table des matières

1. INTRODUCTION

1.1. Public visé et pré-requis

Le présent fascicule est principalement destiné à l'équipe maintenance-qualité et aux développeurs de CESAR-LCPC du Laboratoire Central des Ponts et Chaussées et de ses partenaires, ainsi qu'aux organismes ayant acquis une licence de développement de CESAR-LCPC. Ce fascicule peut être mis également à la disposition des organismes ayant acquis une licence d'utilisation, afin de leur permettre d'assurer l'interopérabilité entre leurs propres logiciels et le progiciel CESAR-LCPC, par l'intermédiaire de la base de données de ce dernier.

Ce fascicule s'adresse donc à des utilisateurs non débutants. Si tel n'était pas le cas, et pour se prémunir contre d'éventuelles difficultés de compréhension, le lecteur est encouragé à se reporter à la documentation d'utilisation, dont en particulier :

- [1] Manuel d'utilisation, fascicule « Prise en main »
- [2] Manuel de référence, fascicule « Glossaire et Lexiques »
- [3] Manuel de référence, fascicule « Solveur CESAR »

1.2. Quelques rappels

A chacune de ses exécutions, le solveur CESAR produit un fichier de résultats au format **.rsv4**, Ce fichier est une composante de la base de données CESAR-LCPC relative à une étude (cf. références [1] et [2]).

Le format des résultats .rsv4 a été établi lors du projet CLEO, en particulier au cours des travaux CLEO-ext2 ayant abouti à la version Standard 4.0 de CESAR-LCPC (2003). Ce format rsv4 a été développé afin d'accélérer les interprétations graphiques, et pour pouvoir mettre en œuvre avec un maximum d'efficacité les nouvelles fonctionnalités de post-traitement des logiciels CLEO2D et CLEO3D.

Le nom du fichier de résultats au format .rsv4 (syntaxe) est ***étude_calcul*.rsv4u**, *étude* et calcul étant des noms donnés par l'utilisateur au lancement du solveur CESAR.

L'ancien format de résultats .resu peut être encore utilisé dans certaines conditions : se reporter au cahier de conception logicielle « Format des fichiers de résultats .resu ».

1.3. Objectif du présent cahier

Chaque module de calcul du solveur CESAR crée un fichier de résultats .rsv4 avec un format adapté au problème traité. L'objectif de ce cahier est de décrire ces différents formats.

2. DESCRIPTION GÉNÉRALE

2.1. Nature du fichier .rsv4

Le fichier de résultats .rsv4 est un fichier à accès direct.

2.2. Contenu schématique

De manière très schématique, le contenu du fichier .rsv4 peut être illustré par la figure ci-dessous :

<File Header>
<Entité N° 1 >
<Entité N° 2 >
<..... >

L'enregistrement d'en tête du fichier de résultats décrit la version du fichier et contient éventuellement un certain nombre de renseignements généraux.

Le fichier de résultats comprend ensuite une ou plusieurs « entités ». Chaque entité possède une structure et une taille identique. La notion d'entité est ici utilisée pour caractériser par exemple les résultats obtenus pour un pas de temps en dynamique, un incrément dans un calcul non linéaire, etc.

Chaque entité est ensuite composée comme suit :

< Entity Header >
< Table N° 1 >
< Table N° 2 >
< Table N° 3 >
< .... >

L'enregistrement de tête de chaque entité contient un certain nombre d'informations générales sur l'entité considérée (Nom, Numéro (du pas de temps, de l'incrément, …), Valeur (du temps, …)).

Suivent ensuite un certain nombre de « tables » contenant les résultats effectifs correspondant à l'entité considérée. Chaque table de résultats est composée comme suit :

< Table Header >
< Data Set > < Data Set > < Data Set > < Data Set > < Data Set > < Data Set >
< Data Set > < Data Set > < Data Set > < Data Set > < Data Set > < Data Set >

L'enregistrement de tête de chaque table contient un certain nombre d'informations parmi lesquelles nous pouvons citer :

- Type des entités sur lesquelles portent les résultats (Nœuds d'un groupe, éléments d'un groupe)
- Type du « Data Set » considéré
- Type de stockage (constant ou par pointeur)
- …

Cet enregistrement est ensuite suivi par une liste de « Data sets » de structure et de longueur identiques.

A titre d'exemple, le format d'un « Data Set » de type « Déplacements 3d avec rotations » est le suivant :

< U, V , W, Teta X, Teta Y, Teta Z >

Ces « data sets » sont décrits dans un fichier externe contenant en particulier les informations suivantes :

Name	Unit	Type
U		R*8
V		R*8
W		R*8
Tx		R*8
Ty		R*8
Tz		R*8

2.3. Notes

Note 1 :

Dans la proposition ci-dessus, les « Data sets » contenus dans une table ont une structure identique. Cette proposition présente un certain nombre d'avantages parmi lesquels :

- Facilités de gestion (unités …)
- Pas de problèmes aux « interfaces »
- Possibilité d'associer les visualisations possibles à chaque type de Data Set
- …

Cette proposition suppose essentiellement un stockage des résultats « par groupes ».

Note 2 :

Pour des questions de compréhension avec l'équipe de maintenance des logiciels CLEO2D et CLEO3D, la suite du présent cahier de conception logicielle est rédigée en langue anglaise.

3. STRUCTURE OF THE RESULTS FILE .rsv4

3.1. Record length

The record length is taken to be 250 words (1ko).

3.2. The first record of every header

To use optimally the minimum length of a record, we can use for the unique record of every header the following structure:

Variable	Type	Length	Nature
K	I4	100	Vector of integers
V	R4	100	Vector of real*4
C	C50	4	Vector of strings

The record of every header contains at least the following elements:

Type	Length	Nature
K(1)	1	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	1	Number of records to jump to find the following header
C(1)	1	Type of the header. This type permits to define the contents of the present header.

3.3. The header of the category “File”

If the header is of category “File”, the complementary information contained in this first record of the header may be defined as follows:

Type	Length	Nature
K(4)	1	Problem dimension NDIM (used for check)
K(5)	1	Total number of nodes NNT (used for check)
K(6)	1	Total number of elements NELT (used for check)
K(7)	1	Total number of element groups NGRPE (used for check)
K(8)	1	Total number of *dof* NDLT (used for check)
C(2)	1	Solver used (Ex : MCNL)
C(3)	1	Solver version
C(4)	1	Names of study and model (max 20 char. each)

For steady state calculations performed by transient solvers (for ex. module DTLI with NPAS1=1 and IPERM=1) , the name of solver used is followed in C(2) by “SteadyState”.

Ex. : DTLI SteadyState

Concerned modules :

DTLI, DTNL, NAPP, NSAT, CSLI, CSNL, MPLI, MPNL, TEXO
(if one calculation step, NPAS1=1) ;
LINE (diffusion problem);
SURF.

3.4. The headers of the category "Entity"

The headers of the category "Entity" may have for example" the following types:

Type "Characteristics"

If the type of the header is "Characteristics", the complementary information contained in the header may be defined as follows (Version 0):

Type	Length	Nature
K(4)	1	Length of table KPROP (LKPROP=NGRPE*NCG)
K(5)	1	Length of table VPROP (LVPROP=NGRPE*(NCM+NPREL)

The header is followed by tables KPROP and VPROP :

Type	Length	Nature
KPROP	LKPROP	Table KPROP as defined in solver CESAR

Type	Length	Nature
VPROP	LVPROP	Table VPROP as defined in solver CESAR

Type "Otherproperties"

If the type of the header is "Otherproperties", the complementary information contained in the header may be defined as follows (Version 0):

Type	Length	Nature
K(4)	1	Length of table KPROCO (LKPROC=NGRPE*2)

The header is followed by tables KPROCO :

Type	Length	Nature
KPROCO	LKPROC	Table KPROCO (IGACT and IGCOL for each group)

Type "LoadingCase"

If the type of the header is "LoadingCase", the complementary information contained in the header may be defined as follows (Version 0):

Type	Length	Nature
K(4)	1	Loading case number
C(2)	1	Name of the loading case

Type "Increment"

Type	Length	Nature
K(4)	1	Increment number
K(5)	1	Convergence indicator ICONV (0 or 1)
K(6)	1	Number of iterations NIT for the current increment
V(1)	1	Convergence tolerance in terms of displacements
V(2)	1	Convergence tolerance in terms of residual forces
V(3)	1	Convergence tolerance in terms of energy

Type “TimeStep”

Type	Length	Nature
K(4)	1	Time step number
V(1)	1	Value of time

Type “EigenVector”

Type	Length	Nature
K(4)	1	Order of the eigenvector
V(1)	1	Eigenvalue

Type “BucklingLoad”

Type	Length	Nature
K(4)	1	Order of the buckling mode
V(1)	1	Value of the buckling load

Type “FrequencyStep”

Type	Length	Nature
K(4)	1	Frequency step number
V(1)	1	Value of the frequency

Type “Fourier”

Type	Length	Nature
K(4)	1	Order of the component

Type “TimeFunctions”

Type	Length	Nature
K(4)	1	NSTEP : Number of considered time steps

The records that follow this header are used to define the time values for which the values are stored in the following tables.

Type	Length	Nature
VTime	NSTEP	Value of time for considered time steps

3.5. The headers of the category "Table"

The headers of the category "table" may have for "example" the following types:

Type "NodesByGroup"

If the type of the header is "NodesByGroup", the complementary information contained in the first record of the header may be defined as follows (Version 0):

Type	Length	Nature
K(4)	1	Group number considered
K(5)	1	Number of nodes of the considered group (used for check)
C(2)	1	Type of "DataSet" stored for the nodes of the considered group

Type "ElementsByGroup"

If the type of the header information is "ElementsByGroup", the complementary information contained in the first record of the header may be defined as follows (Version 0):

Type	Length	Nature
K(4)	1	Group number considered
K(5)	1	Number of elements NEG of the considered group (used for check)
K(6)	1	Type of storage = 0 if every element has a different number of "data sets" stored. This functionality may be used for example for beam elements where the length of VCONE may be very different between 2 elements (Version 4.1) = N : Constant number of DataSets stored for every element of the group.
K(7)	1	"Place" of calculation. = 1 Nodes = 2 Integration points
C(2)	1	Type of "DataSet" stored for the elements of the considered group

If the variable K(6) = 0, the records which follows this header are used to define the pointer KP which precise the beginning of results for every element of the group (Ex : KPCOEL).

Type	Length	Nature
KPointer	NEG+1	Pointer for every element of the group

Type "NodesByList"

If the type of the header information is "NodesByList", the complementary information contained in the first record of the header may be defined as follows:

Type	Length	Nature
K(5)	1	Number of nodes NNL of the considered list
C(2)	1	Type of “DataSet” stored for the nodes of the considered list

The records that follow this header are used to define the nodes considered in the present list.

Type	Length	Nature
KList	NNL	List of considered nodes

This type of header may be used for example to store the “Nodal Reactions”.

Type “ElementsByList”

If the type of the header information is “ElementsByList”, the complementary information contained in the first record of the header may be defined as follows:

Type	Length	Nature
K(5)	1	Number of elements NEL of the considered list
C(2)	1	Type of “DataSet” stored for the elements of the considered list

The records that follow this header are used to define the elements considered in the present list.

Type	Length	Nature
KList	NEL	List of considered elements

Type “AllNodes”

If the type of the header information is “AllNodes”, the complementary information contained in the first record of the header may be defined as follows:

Type	Length	Nature
K(5)	1	Number of nodes NNT (for check)
C(2)	1	Type of “DataSet” stored for all the nodes

Type “AllElements”

If the type of the header information is “AllElements”, the complementary information contained in the first record of the header may be defined as follows:

Type	Length	Nature
K(5)	1	Number of elements NELT (for check)
C(2)	1	Type of “DataSet” stored for all the elements

3.6. Types of « Elementary set of results »

If we except the information contained in the header, a table is essentialy a succession of “Elementary Data sets”.

A table with have the following form defines every elementary set of results.

Code of the set		Description of the set	
Name1	Unit1	Type1	Description of the first value of the set
Name2	Unit2	Type2	Description of the second value of the set
…			

Note : These tables have to be precised.

Mise à jour des unités en cours...
Déplacements : Dp et non L ? (voir TableOfSymbols)
Pressions : en hydraulique L ?

Nodal results

		2D Displacements	
U	Dp	R*4	X component of the displacement
V	Dp	R*4	Y component of the displacement

		2D Displacements with rotation parameter	
U	L	R*4	X component of the displacement
V	L	R*4	Y component of the displacement
Teta	A	R*4	Rotation

		3D Displacements	
U	L	R*4	X component of the displacement
V	L	R*4	Y component of the displacement
W	L	R*4	Z component of the displacement

		3D Displacements with rotation parameters	
U	L	R*4	X component of the displacement
V	L	R*4	Y component of the displacement
W	L	R*4	Z component of the displacement
Tx	A	R*4	Rotation about Ox
Ty	A	R*4	Rotation about Oy
Tz	A	R*4	Rotation about Oz

		Cylindrical displacements	
Ur	L	R*4	r component of the displacement
Uz	L	R*4	z component of the displacement
Ut	L	R*4	teta component of the displacement

		Temperature	
T	Te	R*4	

		Hydraulic Head	
H	L	R*4	

		Pressure	
P	P ?	R*4	

		NormPlasticStrain	
NDP		R*4	

		Hydration degree	
DH		R*4	

		Water content	
WCT		R*4	

		2D Reactions	
Rx	F	R*4	
Ry	F	R*4	

		2D Reactions with moment	
Rx	F	R*4	
Ry	F	R*4	
RM	F*L	R*4	

		3D Reactions	
Rx	F	R*4	
Ry	F	R*4	
Rz	F	R*4	

		3D Reactions with moments	
Rx	F	R*4	
Ry	F	R*4	
Rz	F	R*4	
RMx	F*L	R*4	
RMy	F*L	R*4	
RMz	F*L	R*4	

Element results

		2D Stress tensor	
Sxx	P	R*4	
Syy	P	R*4	
Sxy	P	R*4	
Szz	P	R*4	

		3D Stress tensor	
Sxx	P	R*4	
Syy	P	R*4	
Szz	P	R*4	
Sxy	P	R*4	
Syz	P	R*4	
Sxz	P	R*4	

		Cylindrical stress tensor	
Srr	P	R*4	
Szz	P	R*4	
Stt	P	R*4	
Srz	P	R*4	
Srt	P	R*4	
Stz	P	R*4	

		Shell Stress tensor	
Sxx Sup	P	R*4	
Syy Sup	P	R*4	
Sxy Sup	P	R*4	
Sxx Inf	P	R*4	
Syy Inf	P	R*4	
Sxy Inf	P	R*4	
Sxz	P	R*4	
Syz	P	R*4	

		2D Beam Force results	
X1		R*4	
X2		R*4	
N1		R*4	
V1		R*4	
M1		R*4	
N2		R*4	
V2		R*4	
M2		R*4	
Nm		R*4	
Vm		R*4	
Mm		R*4	

		3D Beam Force results	
X1		R*4	
X2		R*4	
N1		R*4	
Vy1		R*4	
Vz1		R*4	
Mx1		R*4	
My1		R*4	
Mz1		R*4	
N2		R*4	
Vy2		R*4	
Vz2		R*4	
Mx2		R*4	
My2		R*4	
Mz2		R*4	
Nm		R*4	
Vym		R*4	
Vzm		R*4	
Mxm		R*4	
Mym		R*4	
Mzm		R*4	

		2D Truss Force results	
X1		R*4	
X2		R*4	
N1		R*4	
N2		R*4	
Nm		R*4	

		3D Truss Force results	
X1		R*4	
X2		R*4	
N1		R*4	
N2		R*4	
Nm		R*4	

		2D Strain tensor	
Exx		R*4	
Eyy		R*4	
Exy		R*4	
Ezz		R*4	

		3D Strain tensor	
Exx		R*4	
Eyy		R*4	
Ezz		R*4	
Exy		R*4	
Eyz		R*4	
Exz		R*4	

		2D Gradient of hydraulic head	
Hx		R*4	
Hy		R*4	

		3D Gradient of hydraulic head	
Hx		R*4	
Hy		R*4	
Hz		R*4	

		2D Gradient of pressure	
Px		R*4	
Py		R*4	

		3D Gradient of pressure	
Px		R*4	
Py		R*4	
Pz		R*4	

		2D Gradient of temperature	
Tx		R*4	
Ty		R*4	

		3D Gradient of temperature	
Tx		R*4	
Ty		R*4	
Tz		R*4	

		2D Velocity on hydraulic head	
VHx		R*4	
VHy		R*4	

		3D Velocity on hydraulic head	
VHx		R*4	
VHy		R*4	
VHz		R*4	

		2D Velocity on pressure	
VPx		R*4	
VPy		R*4	

		3D Velocity on pressure	
VPx		R*4	
VPy		R*4	
VPz		R*4	

		2D Velocity on temperature	
VTx		R*4	
VTy		R*4	

		3D Velocity on temperature	
VTx		R*4	
VTy		R*4	
VTz		R*4	

		State of contact elements	
Ic		I*4	

4. EXAMPLES

4.1. File MCNL

The result file generated by the solver MCNL for a model that contains 2 groups may have the following structure.

File Header

Variable	Value	Nature
K(1)	1	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	1	Number of records to jump to find the following header
C(1)	FILE	Type of the header. This type permits to define the contents of the present header.
C(2)	MCNL	Solver type

EntityHeader (Increment 1)

Variable	Value	Nature
K(1)	2	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	1	Number of records to jump to find the following header
K(4)	1	Increment Number
V(1)	0.000145	Convergence
C(1)	Increment	Type of the header.

TableHeader (Displacements of nodes for group 1)

Variable	Value	Nature
K(1)	3	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	NR=???	Number of records to jump to find the following header
K(4)	1	Group number considered
K(5)	NNG=???	Number of nodes of the considered group (used for check)
C(1)	NodesByGroup	Type of the header.
C(2)	2DDisplacements	Type of “DataSet”

Data Sets 2DDisplacements

After this header, we store a table of dimension 2*NNG which contains the displacements of the nodes contained in the first group.

<U1,V1><U2,V2>,….

TableHeader (Displacements of nodes for group 2)

Variable	Value	Nature
K(1)	3	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	NR=???	Number of records to jump to find the following header
K(4)	2	Group number considered
K(5)	NNG=???	Number of nodes of the considered group (used for check)
C(1)	NodesByGroup	Type of the header.
C(2)	2DDisplacements	Type of “DataSet”

Data Sets 2DDisplacements

After this header, we store a table of dimension 2*NNG which contains the displacements of the nodes contained in the second group.

<U1,V1><U2,V2>,….

TableHeader (Stresses for group 1)

Variable	Value	Nature
K(1)	3	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	NR=???	Number of records to jump to find the following header
K(4)	2	Group number considered
K(5)	NEG=???	Number of elements of the considered group (used for check)
K(6)	8	Type of storage = 0 if every element has a different number of “data sets” stored. This functionality may be used for example for beam elements where the length of VCONE may be very different between 2 elements (Version 4.1) = N : Constant number of DataSets stored for every element of the group.
K(7)	1	“Place” of calculation. = 1 Nodes = 2 Integration points
C(1)	ElementsByGroup	Type of the header.
C(2)	2DStressTensor	Type of “DataSet”

Data Sets 2DStressTensor

After this header, we store for example a table of dimension 8(Value of K(6)) * 4 (Length of the data set) * NEG which contains the stresses of the elements contained in the first group.

<Sx1,Sy1,Sxy1,Szz1>< Sx2,Sy2,Sxy2,Szz2>,….

TableHeader (Stresses for group 2)

…

Data Sets 2DStressTensor

….

TableHeader (Plastic strains)

Variable	Value	Nature
K(1)	3	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	NR=???	Number of records to jump to find the following header
K(4)		
K(5)	NNT	Number of nodes of the considered group (used for check)
C(1)	AllNodes	Type of the header.
C(2)	NormPlasticStrain	Type of “DataSet”

Data Sets NormPlasticStrain

After this header, we store a table of dimension NNT which contains the norm of the plastic strain tensor for every node.

<V1>< V2>,….

EntityHeader (for increment 2)

TableHeader (Displacements of nodes for group 1)

Data Sets 2DDisplacements

TableHeader (Displacements of nodes for group 2)

Data Sets 2DDisplacements

TableHeader (Stresses for group 1)

Data Sets 2DStressTensor

TableHeader (Stresses for group 2)

Data Sets 2DStressTensor

TableHeader (Plastic strains)

Data Sets NormPlasticStrain

The above entities are repeated for every increment.

4.2. File DYNI

The result file generated by the solver DYNI for a model that contains 2 groups may have the following structure.

File Header

Variable	Value	Nature
K(1)	1	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	1	Number of records to jump to find the following header
C(1)	FILE	Type of the header. This type permits to define the contents of the present header.
C(2)	DYNI	Solver type

EntityHeader (Time step 1)

Variable	Value	Nature
K(1)	2	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	1	Number of records to jump to find the following header
K(4)	1	Time step number
V(1)	0.002	Value of time
C(1)	TimeStep	Type of the header.

TableHeader (Displacements of nodes for group 1)

Data Sets 2DDisplacements

TableHeader (Displacements of nodes for group 2)

Data Sets 2DDisplacements

TableHeader (Stresses for group 1)

Data Sets 2DStressTensor

TableHeader (Stresses for group 2)

Data Sets 2DStressTensor

EntityHeader (Time step 2)

TableHeader (Displacements of nodes for group 1)

Data Sets 2DDisplacements

TableHeader (Displacements of nodes for group 2)

Data Sets 2DDisplacements

TableHeader (Stresses for group 1)

Data Sets 2DStressTensor

TableHeader (Stresses for group 2)

Data Sets 2DStressTensor

The above entities are repeated for every stored time step.

After these sequences of time steps, we can store other entities such as displacements of a list of nodes for every time step. To do this, we can proceed as follows:

EntityHeader (TimeFunctions)

Note : The presence of this header indicates also the sequence of time steps is stopped.

Variable	Value	Nature
K(1)	2	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	???	Number of records to jump to find the following header
K(4)	1	NSTEP : Number of considered time steps
C(1)	TimeFunctions	Type of the header.

After this header, we store a table of dimension NSTEP that contains the values of time for every considered time step

TableHeader (Displacements of nodes for a list)

Variable	Value	Nature
K(1)	3	Category of the header (1=File, 2=Entity, 3=Table)
K(2)	1	Version of the header
K(3)	NR=???	Number of records to jump to find the following header
K(4)		
K(5)	NNL=???	Number of nodes of the considered list
C(1)	NodesByList	Type of the header.
C(2)	2DDisplacements	Type of “DataSet”

The records that follow this header are used to define the nodes considered in the present list.

Type	Length	Nature
KList	NNL	List of considered nodes

Data Sets 2DDisplacements

After this header, we store a table of dimension 2 (Length of the data set) * NNL (Nb of nodes in the list)*NSTEP which contains the displacements of the nodes in the list for every time step.

<U1(t1),V1(t1)><U2(t1),V2(t1)>,….
<U1(t2),V1(t2)><U2(t2),V2(t2)>,….