Manuel d'Utilisation
Fascicule U4.8- : Post-traitement et analyses dédiées
Document : U4.81.31

Outil de post-traitement interactif STANLEY

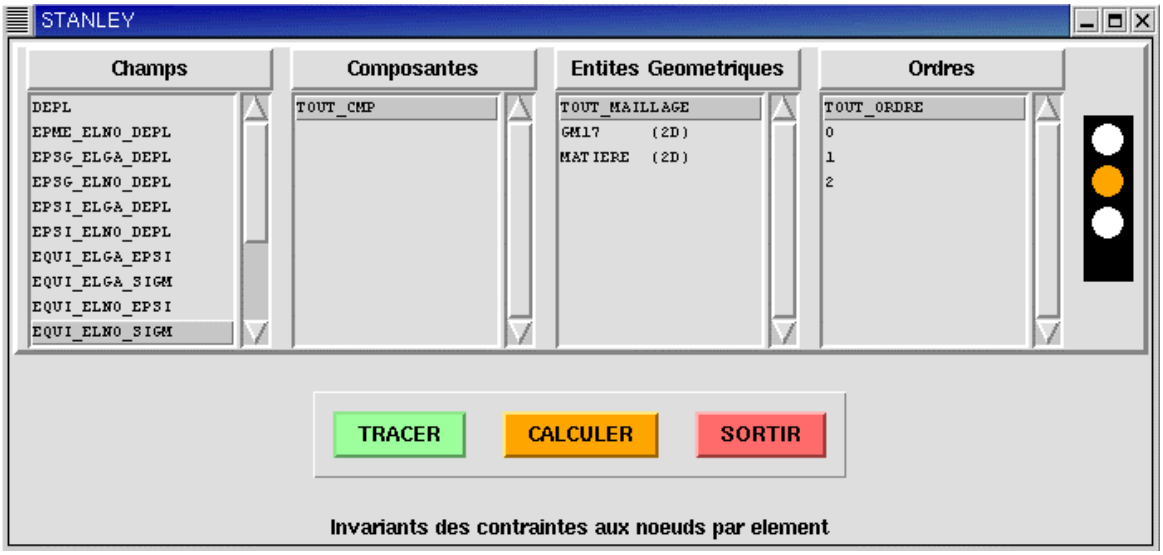
Résumé :

L'application STANLEY est un outil de post-traitement interactif pour les calculs *Aster*. Cette interface graphique permet d'accéder à la liste des grandeurs, de calculer celles qui ne le sont pas encore, de générer les sorties pour les outils de visualisation gmsh (isovaleurs) et xmgrace (courbes) et de lancer ceux-ci.

Table des matières

1 Introduction

L'application STANLEY est un outil de post-traitement interactif pour les calculs *Aster*. Cette interface graphique permet d'accéder à la liste des grandeurs, de calculer celles qui ne le sont pas encore, de générer les sorties pour les outils de visualisation gmsh (isovaleurs) et xmgrace (courbes) et de lancer ceux-ci.



2 Configurations supportées

Stanley est utilisable par les utilisateurs du *Code_Aster*, qui sont dans l'une des situations suivantes :

- Poste de travail sous Linux (ex : poste Calibre), calcul en local
- Poste de travail sous Linux (ex : poste Calibre), calcul sur serveur distant (y compris AlphaServeur)
- Poste de travail Windows + Exceed© "standard DIT" (y compris Portalis)
- Poste de travail Windows + Exceed© + extension Exceed 3D ou Terminal X Unix

Stanley est **disponible** à partir de la version **STA 7.1** de *Code_Aster* et **utilisable** avec la version 6 depuis la **STA 6.5** (l'application Stanley elle-même ne fait pas partie de *Code_Aster* version 6).

3 Paramétrages à effectuer

Paramétrage préalable : à faire uniquement pour utiliser Stanley avec *Code_Aster* version 6.5

L'application Stanley n'étant pas livrée dans les fichiers de *Code_Aster* v6, il faut créer sur la machine qui exécute effectivement *Code_Aster* un répertoire (par exemple `~/Stanley_v6`) et y déposer les fichiers Python de l'application, contenus sur `clayastr` dans le répertoire
`/aster/v6/STA6/Pack_Stanley/Contenu_stanley`

Configuration :

Pour utiliser Stanley (avec *Code_Aster* version **6.5** ou **7**), un paramétrage particulier doit être fait par chaque utilisateur, **sur la machine qui exécute *Code_Aster*** (qui peut être différente de la machine depuis laquelle est lancé ASTK). Voici la démarche à suivre pour utiliser Stanley :

- Dans le `$HOME` de l'utilisateur, ajouter dans le fichier `.cshrc` la ligne (remplacer ma-machine par le nom complet de votre machine (ne pas oublier .der.edf.fr pour EDF) ou bien l'adresse IP) :
  `SETENV DISPLAY ma-machine:0.0`

- Le mode interactif doit être fonctionnel. Vérifier notamment la présence de fichiers `.rhosts` correctement remplis sur toutes les machines impliquées (serveur de calcul et machine de l'utilisateur).

- Créer le répertoire `~/.stanley`.

- Récupérer le fichier de configuration `env.py` dans le répertoire `/aster/v`$_i$`/STA`$_i$`/Pack_Stanley` (avec i=6 ou 7 suivant la version) et le placer dans le répertoire `~/.stanley`.

- Editer ce fichier `env.py` en fonction de la configuration de votre poste de travail (voir ci-dessous).

Voici le détail des paramètres à renseigner dans `env.py` pour chacune des configurations possibles :

- Cas 1 : poste de travail sous Linux (ex: poste Calibre) et calcul *Aster* en local

  - Sur la machine locale, installer *Code_Aster*, gmsh et xmgrace, et éventuellement `ASTK` (sinon on peut utiliser `run_aster`).
  - Dans `env.py` : configurer le mode de fonctionnement `'LOCAL'` et les chemins des outils gmsh et xmgrace.
  - Lancer *Code_Aster* avec `ASTK` ou `run_aster` depuis votre machine locale.

- Cas 2 : poste de travail sous Linux (ex: poste Calibre) et calcul *Aster* sur serveur distant

  - Sur la machine locale, installer gmsh.
  - Stanley va utiliser xmgrace directement depuis le serveur de calcul. Pour l'AlphaServeur, le chemin d'execution de xmgrace est : `/aster/outils/xmgrace`.
  - Dans `env.py` : configurer le mode de fonctionnement `'DISTANT'`, les chemins des outils gmsh (sur votre machine linux) et xmgrace (sur le serveur de calcul) et les adresses des machines.
  - Lancer *Code_Aster* avec `ASTK` en interactif ou `run_aster` depuis le serveur de calcul (nœud 4 de l'AlphaServeur).

- Cas 3 : poste de travail Windows + Exceed "standard DIT" (y compris Portalis)

  - Note : pour que ce mode puisse fonctionner, il faut que le disque local Windows ait un répertoire partagé, qui servira à recevoir le fichier temporaire pour GMSH. Pour les postes Portalis, faire la demande suivante auprès du correspondant informatique : « création d'un répertoire partagé sur le disque local du PC Portalis ».
  - Installation de GMSH sous Windows (il n'y a pas besoin de la DIT pour cela) : télécharger le binaire Windows de GMSH, le dé-zipper dans un répertoire et lancer directement "gmsh.exe" (faire un lien vers le bureau pour plus de convivialité).
  - Stanley va utiliser xmgrace directement depuis le serveur de calcul. Pour l'AlphaServeur, le chemin d'execution de xmgrace est : `/aster/outils/xmgrace`.
  - Dans `env.py` : configurer le mode de fonctionnement `'WINDOWS'`, le chemin de l'outil xmgrace (sur le serveur de calcul). Pour GMSH, entrer le nom de partage Windows du répertoire Windows qui accueillera le fichier temporaire utilisé par gmsh-windows.
  - Lancer *Code_Aster* avec `ASTK` en interactif ou `run_aster` depuis le serveur de calcul (nœud 4 de l'AlphaServeur).
  - Utilisation : dans cette configuration, Stanley ne peut pas ouvrir tout seul GMSH sous Windows, il se contente donc de déposer le fichier `fort.37.pos` dans le répertoire partagé défini dans `env.py`. L'utilisateur doit lancer GMSH Windows, faire "`File / Merge`" et sélectionner lui même le fichier `fort.37.pos` pour l'ouvrir dans GMSH. A noter que Stanley reprend la main et que l'on peut relancer d'autres post-traitement qui iront remplacer le fichier `fort.37.pos` sans perturber GMSH Windows (on peut donc rajouter des vues sans fermer GMSH).

- Cas 4 : poste de travail Windows + Exceed + extension Exceed 3D ou Terminal X Unix

  - Note : sur la machine locale Windows, il faut avoir l'extension "Exceed 3D" pour Exceed (ce n'est pas en standard Portalis).
  - Note 2 : ce type de configuration devrait convenir aux personnes qui utilisent un Terminal X (non testé).
  - Stanley va utiliser la version de GMSH qui est installée sur le serveur Unix sur lequel Exceed est connecté. Se renseigner auprès de son administrateur pour connaître le chemin de l'exécutable.
  - Stanley va utiliser xmgrace directement depuis le serveur de calcul. Pour l'AlphaServeur, le chemin d'execution de xmgrace est : `/aster/outils/xmgrace`.
  - Dans `env.py` : configurer le mode de fonctionnement `'DISTANT'` et les chemins des outils gmsh (sur la machine qui l'exécute) et xmgrace (sur le serveur de calcul). Configurer les trois adresses des machines de calcul, d'exécution de gmsh et de votre machine Windows.
  - Lancer *Code_Aster* avec `ASTK` en interactif ou `run_aster` depuis le serveur de calcul (nœud 4 de l'AlphaServeur).

4 Mode d'emploi et précaution d'emploi

Il y a plusieurs façons de lancer Stanley : automatiquement par ASTK sur une base générée par un calcul soumis en batch, en poursuite d'un calcul déjà effectué, directement à la fin d'un calcul, ou bien sur une autre machine, via un fichier au format MED.

Des exemples d'utilisations sont contenus dans le répertoire
`/aster/v7/STA7/Pack_Stanley/Exemples`
Et pour la version 6 :
`/aster/v6/STA6/Pack_Stanley/Exemples`

Lancement de Stanley directement par l'interface de lancement ASTK (en version 7 uniquement) :

C'est le mode d'utilisation le plus simple. Il faut que le profil d'étude contienne une base générée précédemment. Ensuite, il faut choisir l'outil Stanley dans le Menu « Outils » et se laisser guider.

Lancement de Stanley après un calcul et dont une base a été créée :

On doit lancer, sur la machine où a été créée la base, une nouvelle étude dont le fichier de commande contiendra uniquement les lignes (en version 7) :

```
POURSUITE( PAR_LOT = 'NON' )
import Stanley.stanley
from Stanley import stanley
stanley.PRE_STANLEY()
FIN( )
```

`PRE_STANLEY` va lancer une interface graphique qui va explorer la base et donner à l'utilisateur le choix des concepts *Aster* qu'il souhaite utiliser dans Stanley. Stanley sera lancé automatiquement avec les concepts choisis.

Si l'utilisateur ne souhaite pas choisir ses concepts dans une liste, il a la possibilité de lancer directement Stanley en remplaçant la ligne `stanley.PRE_STANLEY()` par la ligne :

```
stanley.STANLEY(EVOLNOLI,MAILLAGE,MODELE,CHAMATER,CARAELEM)
```

dans laquelle il faudra remplacer `EVOLNOLI`, `MAILLAGE`, `MODELE`, `CHAMATER` et `CARAELEM` par les noms de concepts définis dans le fichier de commande. Si `CARAELEM` n'est pas défini, mettre obligatoirement `None`.

En version 6, `PRE_STANLEY` n'est pas disponible, et d'autre part, Stanley est contenu dans un répertoire utilisateur (dans cet exemple `~/Stanley_v6`) et les commandes permettant d'appeler Stanley sont alors :

```
POURSUITE( PAR_LOT = 'NON' )
import sys,os,os.path
home = os.environ['HOME']
rep_env = os.path.join(home,'Stanley_v6')
sys.path.insert(1,rep_env)
import stanley
stanley.STANLEY(EVOLNOLI,MAILLAGE,MODELE,CHAMATER,CARAELEM)
```

Lancement de Stanley directement à la fin d'un calcul (non conseillé) :

Note : pour les utilisateurs qui utilisent un serveur de calcul centralisé (comme l'AlphaServeur `clayastr`), nous attirons votre attention sur la gène que pourrait occasionner l'utilisation du mode interactif pour effectuer l'intégralité de votre calcul ! Nous vous incitons à effectuer votre calcul en batch sur le serveur centralisé puis à lancer Stanley depuis un deuxième fichier de commande en interactif, comme décrit dans le point précédent.

Pour pourvoir lancer Stanley, il faut être en `PAR_LOT='NON'` :

```
DEBUT(PAR_LOT = 'NON')
```

et insérer les trois lignes suivantes dans le fichier de commande *Aster*, avant l'instruction `FIN();` :

```
import Stanley.stanley
from Stanley import stanley
stanley.STANLEY(EVOLNOLI,MAILLAGE,MODELE,CHAMATER,CARAELEM)
```

Remplacer `EVOLNOLI`, `MAILLAGE`, `MODELE`, `CHAMATER` et `CARAELEM` par les noms de concepts définis dans le fichier de commande. Si `CARAELEM` n'est pas défini, mettre `None`.

En version 6, si le répertoire qui contient l'application Stanley est `~/Stanley_v6`,

```
import sys,os,os.path
home = os.environ['HOME']
rep_env = os.path.join(home,'Stanley_v6')
sys.path.insert(1,rep_env)
import stanley
stanley.STANLEY(EVOLNOLI,MAILLAGE,MODELE,CHAMATER,CARAELEM)
```

La conséquence du `PAR_LOT='NON'` est que le fichier n'est plus visualisable par EFICAS. La solution est donc de créer un deuxième fichier de commande pour lancer Stanley en POURSUITE, comme décrit dans le point précédent.

Lancement de Stanley sur une machine de post-traitement différente de la machine d'exécution :

Ce mode d'utilisation a l'avantage de permettre le dépouillement sur une machine locale (où est installée une version du *Code_Aster*), sans les limitations appliquées au mode interactif sur la machine centralisée clayastr.

La procédure générale est décrite ci-dessous :

- Lancement de l'étude complète en batch sur serveur centralisé, comprenant l'écriture au format MED du résultat produit par une commande globale (`STAT_NON_LINE`, `DYNA_NON_LINE`, `MECA_STATIQUE`, etc…). Par exemple :

```
DEFUFI(IMPRESSION=_F(NOM='MED',UNITE=81,),);
IMPR_RESU(RESU=_F(FORMAT='MED',FICHIER='MED',RESULTAT=RESU,),);
```

- Ce fichier MED contiendra les champs aux nœuds (par exemple DEPL) et les champs par éléments aux nœuds (par exemple `SIEF_ELNO_ELGA`).

- Recopie (manuelle) du fichier MED sur la machine de post-traitement.

- Sur cette machine, il faut régénérer les concepts nécessaires à STANLEY : maillage, modele, cham_mater et éventuellement cara_elem, en exécutant le fichier de commande de l'étude jusqu'à la commande globale (STAT_NON_LINE ou autre), qui sera remplacée par la lecture du résultat au format MED. Un exemple de l'appel à LIRE_RESU est donnée ci-dessous :

```
UUU=LIRE_RESU(TYPE_RESU='EVOL_NOLI', FORMAT='MED',
              MODELE=MO,
              FORMAT_MED=(_F( NOM_CHAM_MED=
                           'RESU____DEPL_____________________',
                           NOM_CHAM    ='DEPL' ),
                          _F( NOM_CHAM_MED=
                           'RESU____SIEF_ELNO_ELGA___________',
                           NOM_CHAM    ='SIEF_ELNO')) ,
              NOM_CHAM=('DEPL','SIEF_ELNO',),
              TOUT_ORDRE='OUI',);
```

(le nom des champs MED est donné dans le fichier messages lors de l'écriture au format MED)

- Dans ce fichier de commandes, l'appel à Stanley est défini par les commandes (version 7) :

```
import Stanley
from Stanley import stanley
stanley.PRE_STANLEY()
```

- En version 6,

```
import sys,os,os.path
home = os.environ['HOME']
rep_env = os.path.join(home,'Stanley_v6')
sys.path.insert(1,rep_env)
import stanley
stanley.STANLEY(EVOLNOLI, MAILLAGE,MODELE,CHAMATER,CARAELEM)
```

Attention, dans ce cas, le répertoire Stanley_v6 et les fichiers qui lui sont associés, ainsi que le fichier env.py doivent exister sur la machine de post-traitement.

Utilisation de l'interface graphique Stanley :

L'utilisation proprement dite de l'interface graphique ne pose pas de problème : le vocable est celui de CALC_ELEM et d'IMPR_RESU.

Concernant l'ergonomie :

- Sur le côté droit, le feu tricolore indique l'état des concepts : vert : concept calculé et visualisable, orange : concept que l'on peut calculer pour passer au feu vert, rouge : concept que l'on ne peut pas calculer dans le cadre du calcul.
- En cliquant sur Ordre on bascule entre les NUME_ORDRE et les INST.
- En cliquant sur Entités Géométriques on bascule entre le tracé d'isovaleurs (avec GMSH) ou de courbes (avec XMGRACE).

5 En cas de problème ou pour faire remonter des demandes

Problèmes courants :

- GMSH ne se lance pas.

  Si le `.mess` contient le message d'erreur "`Can't open display`", vérifier que la ligne `SETENV DISPLAY ma-machine.der.edf.fr:0.0` est bien renseignée dans le fichier `.cshrc` sur la machine qui exécute *Aster*. Vérifier que les fichiers `.rhost` sur les différentes machines sont bien renseignés.

- Je suis en configuration Windows et le fichier `fort.37.pos` n'est pas déposé dans mon répertoire temporaire Windows.

  Si le message "`putting file fort.37.pos as \fort.37.pos (xxx kb/s)`" apparaît, c'est que Stanley a bien posé le fichier. Vérifier dans `env.py` que la `machine_gmsh` est bien la bonne et que le répertoire partagé est bien celui dans lequel vous regardez...

  Si le message précédent n'apparaît pas, vérifier les droits d'écriture sur votre répertoire partagé en lançant manuellement une des deux commandes suivantes (depuis le serveur de calcul) :
  `smbclient '\\ma-machine.der.edf.fr\mon-rep-temp'` (répertoire en accès complet)
  `smbclient '\\ma-machine.der.edf.fr\mon-rep-temp' -U mon-log-win` (sinon)
  et vérifier que vous avez bien le prompt "`smb: \>`". Si ce n'est pas le cas, vérifiez les droits d'accès sur le répertoire partagé Windows.

Pour faire remonter des bugs ou des demandes d'évolution du produit, utiliser l'AREX *Aster* en ouvrant des fiches d'Anomalie ou d'Evolution Outil.

Page laissée intentionnellement blanche.