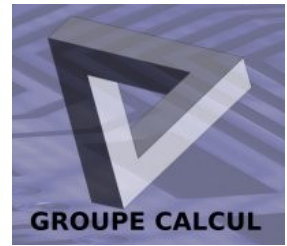




Emmanuel Courcelle
Pierre Elyakime
Hervé Neau
Annaïg Pedrono



Atelier Visualisation in Situ : TP VisIt InSitu

Ce TP s'appuie sur l'exemple pjacobi fourni par J. Favre. La documentation libsim et les exemples de codes sources sont disponibles sur la page <https://wci.llnl.gov/simulation/computer-codes/visit/manuals> dans la partie Getting Data into Visit.

Objectif : connecter la bibliothèque libsim2 de VisIt avec un code de calcul

Prérequis :

- installation de VisIt 2.1* et de VisIt 2.1* « With mesa support » pour la partie III du TP
- make
- mpi
- un compilateur C ou Fortran ou Python selon le langage choisi

Attention, Pour faire tourner VisIt InSitu, il faut que le chemin vers l'exécutable « visit » soit dans le path !

Le TP est prévu pour du bash. Modifier les chemins vers l'installation de VisIt dans les fichiers visit_GUI.sh et visit_batch.sh

Partie I : Utilisation de VisIt in Situ

a – Modifier le Makefile pour :

- Définir le chemin vers le répertoire d'installation de VisIt (VISIT_DIR)
- Définir le compilateur

Charger une distribution MPI (cf Readme)

Recompiler avec :

```
make clean; make  
./pjacobi
```

Le programme pjacobi est fonctionnel, mais il est pour l'instant complètement indépendant de VisIt. Modifier le Makefile pour compiler avec les bibliothèques VisIt.

```
source ../../visit_GUI.sh  
make clean; make
```

```
./pjacobi
```

Dans un nouveau terminal, lancer VisIt (après avoir sourcé **visit_GUI.sh**).

Open : ouvrir le dernier fichier *.sim2 du répertoire \$HOME/.visit/simulations

Add/Pseudocolor/Temperature

Draw

Add/Mesh

Draw

b – Lancer l'exécutable avec un fichier de trace pour vérifier que tout se passe bien :

```
mpirun -np 4 ./pjacobi -trace trace.txt (ou sur Calmip sbatch slurm.sh)
```

Dans un nouveau terminal, lancer VisIt (après avoir sourcé **visit_GUI.sh**).

Open : ouvrir le dernier fichier *.sim2 du répertoire \$HOME/.visit/simulations

Add/Pseudocolor/Temperature

Draw

Add/Mesh

Draw

Menu **File/Simulations**

Cliquer sur le bouton **run** puis **halt**

Tester le bouton **step**

Pour déconnecter la simulation de VisIt sans attendre la fin, fermer VisIt.

c – Modifier l'exécutable afin que le calcul se lance dès le démarrage du programme.

Vérifiez que lorsque vous connecterez VisIt par la suite le programme devra s'interrompre et que lorsque vous quittez VisIt, la simulation devra reprendre.

En même temps, introduisez une attente de 1 seconde dans le programme de calcul d'un pas de simulation : sinon ce programme « jouet » s'exécute trop rapidement et on n'a le temps de rien voir.

Démarrez le programme de calcul, attendez quelques secondes puis démarrez VisIt comme précédemment. Vérifiez que vous pouvez ouvrir et fermer plusieurs fois la visualisation sans perturber la simulation.

Partie II : Modification du code source pour ajouter de nouvelles fonctionnalités

a – Modifier le fichier **PJacobi_Visit.f** pour faire afficher la variable vectorielle **gradTemp** dans VisIt puis recompiler.

```
mpirun -np 4 ./pjacobi -trace trace.txt
```

Lancer VisIt,

Open : ouvrir le dernier fichier *.sim2 du répertoire \$HOME/.visit/simulations

Add/Mesh

Draw

Add/vector/gradtemp

Draw

Add/Contour/temperat

Draw

Menu File/Simulations

Cliquer sur le bouton **run** puis **halt**

b – Modifier le fichier **PJacobi_Visit.f** pour ajouter un bouton **save_bin** dans l'interface de contrôle de VisIt afin de faire écrire le fichier binaire **Jacobi_<numiter>.bin**.

Partie III : Utilisation de libsim en mode batch (sans lancer VisIt)

Pour cette partie, il faut ouvrir un nouveau terminal et sourcer l'environnement « batch » :

```
source ../visit_batch.sh
```

Pour sauvegarder des extractions de données et/ou des images au cours du calcul, il est possible d'utiliser VisIt en mode batch. A partir de l'exemple fourni par VisIt présent dans le dossier libsim_examples_batch, adapter le programme pjacobi.f90 pour :

- supprimer la connexion interactive à VisIt
- faire écrire grâce aux fonctions de libsim une image et un fichier VTK tous les 10 pas de temps au début de la simulation (jusqu'à l'itération 100).

Attention, pour ouvrir les fichiers VTK créés par le code avec VisIt, il faut utiliser la version de VisIt avec GUI et donc ouvrir un nouveau terminal et positionner les variables avant de lancer VisIt avec :

```
source ../visit_GUI.sh
```

NB :

Pour le Fortran, regarder la documentation sur :

<https://github.com/visit->

[vis/VisIt/blob/master/tools/DataManualExamples/Simulations/fupdateplots.f](https://github.com/visit-visit/VisIt/blob/master/tools/DataManualExamples/Simulations/fupdateplots.f)

Pour Python, regardez la documentation, par exemple, sur :

- ajout vecteur gradient: <https://github.com/visit->

[vis/VisIt/blob/master/tools/DataManualExamples/Simulations/var.py](https://github.com/visit-visit/VisIt/blob/master/tools/DataManualExamples/Simulations/var.py)

- pour le batch, partez d'aide Fortran et utilisez la définition des fonctions python de la librairie libsim : <https://pypi.org/project/visitsimv2/f/visitsimv2/simV2.py>

Annexe à l'intention des personnes qui travaillent sur CalMip

Visit 2.10.2 est installé sur Eos. Pas la peine d'utiliser les scripts d'initialisation de visit, à la place utilisez les modules prédéfinis :

module load visit/2.10.2

ou

module load visit/vbatch-2.10.2

Pour utiliser l'interface graphique de Visit, utilisez les nœuds de visualisation graphique d'Eos. Pour un meilleur confort d'utilisation, nous vous recommandons d'utiliser une grande fenêtre d'affichage :

runVisuSession.sh -g 1600x1000

Bien sûr vous devrez avoir installé sur votre poste de travail le logiciel virtual VNC.

Lorsque la fenêtre graphique est lancée, vous pouvez appeler Visit à partir du menu principal :

Bouton de droite

Vizualisation applications / VisIt

Vous avez également à votre disposition un gestionnaire de fichiers (pratique pour visualiser les fichiers jpg générés par le programme de calcul lors de la partie III).

Toutes les explications sur l'affichage graphique à distance se trouvent ici :

<https://www.calmip.univ-toulouse.fr/spip.php?article463>

Pour lancer les calculs il est **indispensable** d'utiliser **les nœuds de calcul**. Ceux-ci n'ont pas besoin de graphique grâce à la bibliothèque Mesa (module vbatch-2.10.2). Un script de lancement utilisant slurm est fourni, donc pour lancer les calculs il suffit de taper :

sbatch slurm.sh

Vous pouvez lire en temps réel ce qui est envoyé sur la sortie standard avec la commande :

tail -f slurm.<job-id>.out

Toutes les explications sur slurm et la configuration des queues sur eos se trouvent ici :

<https://www.calmip.univ-toulouse.fr/spip.php?rubrique96>