Cycle de vie des données : exemple et application

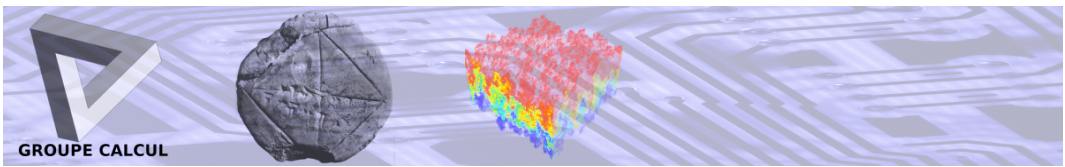
ANF Visualisation et données

http://calcul.math.cnrs.fr/spip.php?article272

Gabriel Moreau

Laboratoire LEGI - UMR5519 - CNRS / UGA / Grenoble-INP - France

28 novembre 2016



Licence

Cette présentation est sous : Licence Art Libre

http://artlibre.org/

Le LEGI

Une unité mixte de recherche : UMR5519

- Trois tutelles
- Crée en 1992 suite à la restructuration de l'IMG (Institut de Mécanique de Grenoble)

Le LEGI

Laboratoire des Écoulements Géophysiques et Industriels

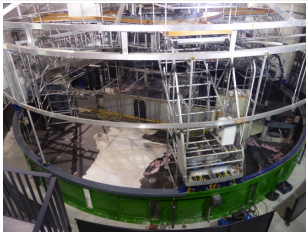
Laboratoire de recherche publique en mécanique des fluides

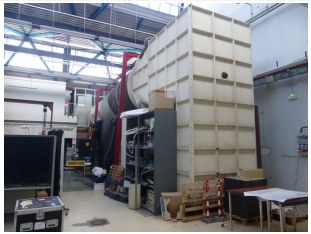
- Environ 140 personnes (moitié permanent, moitié temporaire)
- Trois bâtiments datant de la fin des années 60 dont deux hangars
- Deux bâtiments neufs (2013)
- Un fort historique expérimental (développement de la houille blanche)
- Des liens forts avec les centres de calcul nationaux (IDRIS, CINES...)

Le LEGI

Des expérimentations d'exceptions

- Un canal à houle de 36m (transport...)
- Une soufflerie (PTV...)
- Une plaque tournante Coriolis II de 13m (PIV...)
- Un tunnel hydrodynamique (cavitation...)
- ...





Le LEGI

Du numérique depuis plus de 30 ans

- Clusters de calcul en local (700 cœurs)
- Baies de stockage (environ 800To)
- Simulation dans les centres régionaux et nationaux (calcul jusqu'à 16000 cœurs par exemple)
- ...

Refondation 2011-2013

2011-2013 - Reconstruction

- Deux nouveaux bâtiments
- Profiter du neuf pour améliorer le vieux petit à petit
- **Mettre les données au cœur du projet**
- Centraliser le cœur du système d'information en un point, au centre
- Avoir un bon réseau d'interconnexion extensible et modulaire
- Choix : connections en **fibre optique monomode** de l'intégralité des locaux (encore en cours)

On n'a pas un bon cycle des données si l'infrastructure ne suit pas.

Refondation 2011-2013

Mettre les données au cœur du projet

- Acquisition des données de plus en plus volumineuses (par caméra rapide par exemple 30000fps)
- Transfert direct vers les baies centrales (pas de cascade de commutateur, du point à point au maximum)
- Traitement en parallèle des images sur le cluster du laboratoire

Un point central pour les données

- Une et une seule salle serveur au centre des bâtiments (dans la partie rénovée du G). **Aucun sous local technique**
- Un **réseau point à point direct** vers toutes les expérimentations et certains postes de travail pouvant aller à 10Gb/s ou plus à terme (40Gb/s, 100Gb/s)
- Avoir une arrivée 10Gb/s sur la plaque tournante Coriolis II (distante de plus de 75m de la salle serveur)

Refondation 2011-2013

Les anciens locaux (plus de 50 ans)

- Un bâtiment A de bureau
- Un immense hangar double GH ayant aussi des bureaux

Refondation 2011-2013

Des nouveaux locaux (2013)

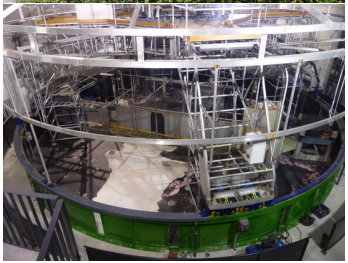
- Une très ancienne demande CPER
- Bâtiments rectorats délégués à Grenoble-INP
- Un nouveau bâtiment K de bureau (K'fet, Amphi)
- Une partie utilisée pour rénover un sous ensemble du G

Refondation 2011-2013

Une nouvelle plaque tournante Coriolis II

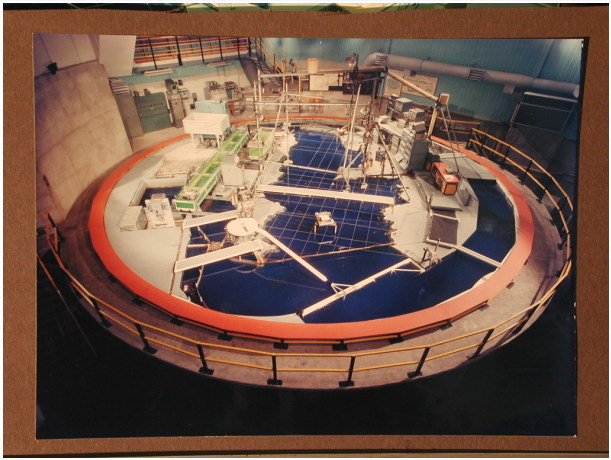
- Problème d'alignement entre Coriolis I et le nouveau tramway ligne B...
- Reconstruction sur le campus - bâtiment L
- Financement exceptionnel (retard du projet campus)

Si on veut des bonnes données, c'est mieux d'avoir une bonne manip !



CORIOLIS - Historique

1960 : l'usine marémotrice des îles Chausey / INSIS



Défaut : pas de nombre premier entre le nombre de galets porteurs et le nombre de morceaux de poutre du rail de roulement.

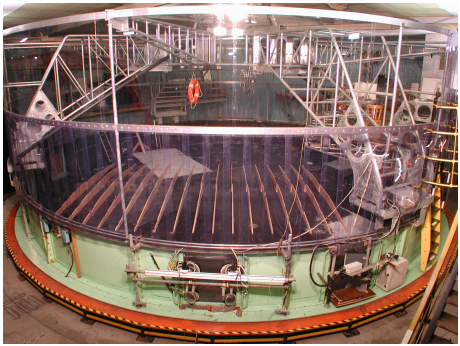
CORIOLIS - Historique

1985 : plateforme de recherche en dynamique des fluides géophysiques / INSU

Défaut : pieu central limitant la zone d'étude

CORIOLIS - Historique

2002 : instrumentation laser



Défaut : vitre circulaire déformant les rayons optiques (laser)

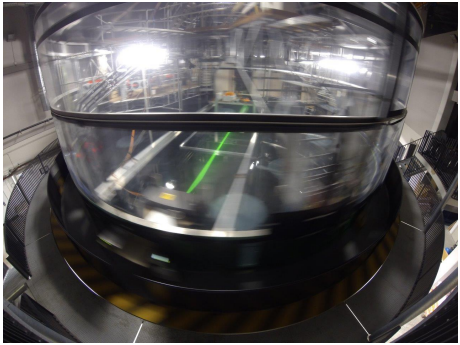
CORIOLIS - Historique

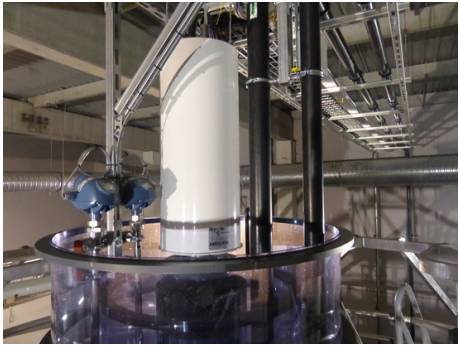
2011/06-2014/03 : destruction - reconstruction

Défaut : interruption de service en plein contrat européen

CORIOLIS II

2014 : ça tourne de nouveau

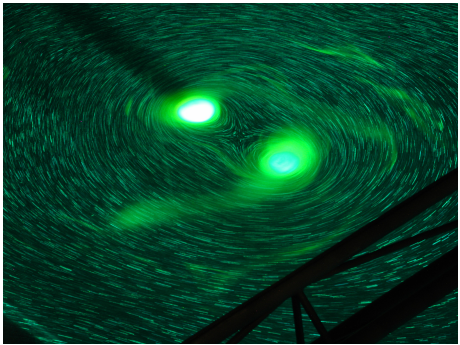


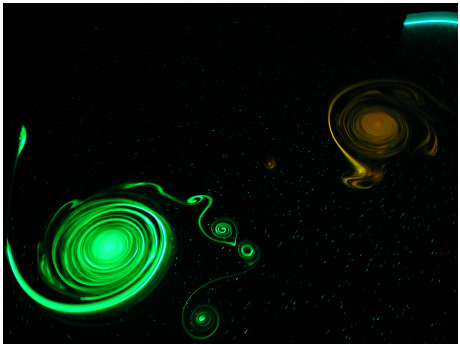


Défaut : plateforme parfaite !

CORIOLIS - Exemples

Du laser et des images (caméra rapide)





Un projet européen très ancien...

...centré sur des problématiques géophysiques, principalement atmosphère océan, mais pas que (expériences sur les galaxies réalisées sur Coriolis).

- 1997-2000 - Début de l'infrastructure européenne Hydralab comme *Concerted Action*
- 2000-2004 - Réseau Hydralab II
- 2006-2010 - Hydralab III - *Integrated Infrastructure Initiative* (I3). Accès possible à 22 infrastructures, 69 projets et plus de 500 chercheurs concernés.
- 2010-2014 - Hydralab IV - Suite
- 2015-2019 - Hydralab+, 11 partenaires et 18 plateformes à disposition.

Objectif actuel : mieux répondre aux questions d'adaptation liées au changement climatique

European High-Performance Infrastructures in Turbulence

centré sur l'innovation technologique et les grands enjeux sociétaux...

- 2013-2016 - European Community Framework Programme 7 (FP7)
- 25 partenaires, 2 industriels, 14 plateformes

Points intéressants

- Provide free access to the knowledgebase for turbulence data
- Enable the evaluation of data quality

Fractionnement des données

Problématique

- Trop de monde (150 personnes)
- Trop de données (des téra à la pelle !) et de plus en plus chaque année
- Évolution de la typologie des données avec les années (nouveaux capteurs, nouveaux algorithmes. . .)

Obligation de **fractionner** celles-ci et de les **distribuer**.

Fractionnement des données

Quelques pistes

- Découpage en projet
- Découpage en espace de travail (workdir personnel) / dossiers de capitalisation (données météo...) - choix équipe MEOM
- Découpage en équipe (à limiter)

Fractionnement - Fonctionnement par projet

- Projets classés par `équipe/YYYY/YYNOM` (exemple : `coriolis/2016/16RESEK`)
- Possibilité de dispatcher les projets sur des volumes (baies de stockages) différents
- Espace de données (dossier) avec un quota de groupe
- Un responsable de projet (propriétaire du dossier racine)
- Une liste de membre pour chaque projet

- Associée un projet sur la forge Trac (facultatif) - espace versionné, **wiki** privé/**public**, timeline, flux RSS, WebDAV public, WebDAV privé
- La forge est un plus pour décrire le projet de manière collaborative via une page web simple (wiki). On peut aussi (trop rarement à l'usage) y mettre l'historique des codes spécifiques utilisés.

Fractionnement - Exemple de la plus value wiki

Projet Coriolis 14CARR

`http://servforge.legi.grenoble-inp.fr/projects/pj-coriolis-14carr`

Suivis journalier des expériences 14CARR

6 - Table of experiments

List of parameter, Param1... , denoted by names defined in section 2.4.

Name	Date	ρ_0	ρ_1	$\Delta\rho_{(kg/m3)}$	$H_{s(cm)}$	$L_{s(cm)}$	$\Omega_{(s-1)}$	Q_0	$Q_{1(m3/h)}$	h_1	Remarks
#FJORD1	27/06	1000	1005	5	50	200	0	0	60	?	PIV, many problems
#FJORD2	01/07	1000.4	1015.1	14.7	43.3	200	0	0	10	?	PIV, problems - see below
#FJORD3	02/07	1000.4	1013.6	13.2	41.6	200	0	0	25	?	PIV, problems - see below
#FJORD4	03/07	1000.1	1007.6	7.5	43.0	200	0	0	25	?	PIV, problems - see below
#FJORD5	04/07	1000.1	1007.4	7.3	45.1	200	0	0	25	?	LIF, good images (a)&(b)
#FJORD6	09/07	1000.0	1003.3	3.3	43.4	200	0	0	25	?	LIF/PIV? (Fluorescene dye added)
#FJORD7	09/07	1001.0	1006.6	5.6	?	200	0	0	25	?	LIF/PIV? (" " ")
#FJORD8	10/07	1001.1	1005.3	4.2	40.5	200	0	0	25	?	LIF (rhodamine dye added)
#ESTUARY1	10/07	1001.1	1005.3	4.2	40.5	200	0	Varies	14.5, 3.0	?	Test experiment for exchange flows
#ESTUARY2	15/07	1000.0	1005.1	5.1	43.0	200	0	Varies	25	?	PIV
#ESTUARY3	16/07	1000.0	1009.6	9.6	45.0	200	0	Varies	25	?	PIV
#ESTUARY4	16/07	1000.0	1009.6	9.6	35.0	200	0	Varies	25	?	PIV
#ESTUARY5	16/07	1000.0	1009.6	9.6	35.0	200	0	Varies	9.5	?	PIV/LIF (rhodamine added)
#ESTUARY6	17/07	1000.0	1004.7	4.7	35.4	200	0	Varies	10.5	?	
#ESTUARY7	17/07	1000.0	1004.7	4.7	34.9	200	0	Varies	25	?	
#GULF1	18/07	1000.0	1004.7	4.7	35.0	200	0.0167	Varies	10.0	?	Rotating exchange flow run
#GULF2	18/07	1000.0	1004.7	4.7	35.0	200	0.0167	Varies	10.0	?	Rotating exchange flow run

7 - Diary:

FJORD1 - 27/06/2014

first test experiments. Initial condition poorly controled: bubbles in salt water flux + leaks in topography? leak in the salt water filling system.

probes to be checked ...

camera dalsa1 (sill top) did not record ?

camera Dalsa2 gives poor images (lines), not enough particles. Yet some PIV can be done .

Processing camera: first image at 16h51:43, gravity current along slope clearly visible with detachment at the level of the interface. Projection done on a grid along the slope.

Processing probes:

To check meaning of the time in the file (start of data acquisition?. Check time correspondance with the clock of the image computer.problem of ground for the electrical signal?

exp0.lvm:

time record 1600 s starting at 16h39:39 (end at about 17h06). camera signal (5 Hz) visible in t=153-212 s (300 images), t=258-337 s, (395 images), 452-1565s (5565 images). Profiles (motor) done at t=20-105 s, and for t=1146-1212 s, pos z=78 cm in between (not good).

1. C5: strange signal, may-be displacement wrong.
2. T5: (temperature), noisy around 1.9 Volt
3. C6: no signal (v=-5 with noise)
4. T6: very noisy, around V=7 volt
5. C2: like C5 but noisy
6. T2: signal -5 to -4 Volt
7. ADVP: noisy around 0
8. I1: slow increase in time beyond t=400, may correspond to the fillling of the basin.
9. I2: bad signal, V=+7.5 V.
10. I3: interesting oscillations around V=-7.5 V
11. I4: noisy, around V=-3.2 V

exp0_1.lvm: 17h12:38, duration 240s, single profile down-up, no camera signal (but noise 1.5 Volt peak to peak due to motor)

1. C5: seems a good profile (although some noise).

Format des données - Problématique

- Au LEGI, 154 Projets (dossiers) référencés très divers
- 179 Projets sur la forge (pas mal de projets d'un autre type - PHD Thesis, Working Group...)
- Grande variété des sources de données (code de calcul, caméra rapide...)
- Grande variété des structures de données (calcul, expérimentations...)
- Arrivée et départ régulière de personnes (notamment sur la plateforme Coriolis)
- Comment reprendre un projet plusieurs années après (exemple 10 ans) ?

Format des données - Extrêmes variétés

- Image caméra rapide
- Vidéo caméra rapide
- Mesures capteurs diverses (vélocimétre, fils chaud, thermocouple...)
- Plan de conception de la manip
- Photographie (et vidéo) de la manip
- Données d'entrée des codes
- Données de sortie des codes
- Script de soumission sur les clusters
- Version des logiciels utilisés
- ...

Format des données - Extrêmes variétés

Données simulations / expérimentations

- Fabrication très différentes (Méthodes pour les générer)
- Structures très différentes
- Visualisation utilise souvent des outils différents
- Parallélisation du traitement n'est pas en général de la même nature
- Pas de convergente forte à moyen terme à mon sens

Par la suite, le propos concernera plus particulièrement les données d'expérimentations

Format des données

- Pas de solution unique
- Choix de solutions unificatrices et pérennes dans le temps
- Être multiplateforme Windows - MacOSX - GNU/Linux
- Être multilangage. Il n'y a pas de langage absolu. Les langages à la mode aujourd'hui ne le seront pas forcément demain. Actuellement Fortran, C++, Python, Matlab sont indispensable au LEGI. On n'aime pas particulièrement le Java !
- Éviter les chemins absolus, toujours préférer le relatif ce qui facilite le déplacement des données
- Pouvoir être manipulé depuis le shell Bash est toujours un plus (`grep`, `ncdump`, `h5dump`...)

Format des données

- Pas de base de données relationnelles (pas assez d'autonomie pour l'utilisateur de création ainsi que de souplesse organisationnelle).
- Stockage sous forme d'une arborescence de fichier
- Format standard en priorité pour les images PNG (pas de perte, pas d'artefact), TIFF, NetCDF4...
- Format HDF5 pour les structures arborescentes (parallélisable en MPI via HDF5 Parallel)
- Format ad-hoc selon les applications (exemple VTK, IGES...)
- XML + schéma XML pour les méta données (JSON ou YAML sont plus sexy actuellement). Les schémas permettent de valider la conformité des fichiers.

Format des données - Évolution du support de stockage

- Disque dur extractible
- Disque dur ou clef USB
- Disque dur du poste de travail (transfert par réseau)
- Baie de stockage centralisée : type NAS, DAS ou SAN
- Stockage répartie local / accès POSIX (type glusterfs)
- Stockage dans les nuages / accès `https` (type MyCORE, Grille / iRODS)

Format des données - Protéger ses données scientifique

- Stockage centralisé en RAID6 ou équivalent (tolérance aux pannes du support)
- Utilisateur actuel peu au fait dans la gestion des droits UNIX. Les outils graphiques n'aident pas !
- Cloisonner en travaillant par projet (groupe d'utilisateurs)
- Mettre des quotas par projet (protection des projets les uns des autres)
- Essayer de **protéger les données contre** les virus et **contre soi-même** !
- Basculer le plus rapidement les dossiers contenant les données brutes en lecture seule
- Corrolaire : faire les traitements dans un dossier parallèle des données brutes mais pas dans un sous dossier !

Format des données - Protéger ses données scientifique

- Bascule automatique en lecture seule (Read-only) après une certaine durée d'inactivité (exemple 2 ans)
- Droit de modification par défaut pour le Groupe `chmod g+rwXs`
- Lancer ses programmes avec le masque `umask 0002` pour partager (en écriture) les résultats au groupe
- Pas de droit pour les autres utilisateur par défaut `chmod o-rwx` (projet non public)
- Une tâche planifiée (cron) corrige préventivement les droits toutes les nuits (propriétés du groupe, ajout du droit de lecture pour le groupe, suppression des droits d'exécution pour les images...)
- Avoir une **sauvegarde** à jour des données critiques (si financement). L'idéal est d'**avoir un système de nommage qui identifie clairement les données critiques** des autres données.

Cycle de création de la donnée expérimentale

Approche historique

- Mesure sur PC d'acquisition
- Pré-Traitement (facultatif) sur PC d'acquisition
- Transfert (copie) par disque externe
- Traitement sur PC de l'utilisateur

Cycle de création de la donnée expérimentale

Centralisation de la donnée

- Mesure sur PC d'acquisition
- Pré-Traitement (facultatif) sur PC d'acquisition
- Transfert (copie) sur des baies centrales
- Traitement sur PC de l'utilisateur

Cycle de création de la donnée expérimentale

Centralisation du traitement

- Mesure sur PC d'acquisition
- Pré-Traitement (facultatif) sur PC d'acquisition
- Transfert (copie) sur des baies centrales
- Traitement sur cluster du laboratoire (exemple OAR)

Cycle de création de la donnée expérimentale

Éviter les copies

- Mesure sur PC d'acquisition sauvée directement sur les baies stockages centralisées
- Pré-Traitement (facultatif mais à éviter) sur PC d'acquisition
- Traitement sur cluster du laboratoire
- Optimisation possible si traitement envisageable via des algorithmes de type map/reduce (map - traitement embarrassingly parallel)

C'est actuellement la solution préconisée en interne en partie grâce au réseau 10Gb/s en fibre optique reliant les expérimentations aux serveurs. Elle force par ailleurs le cadre de type projet (plus de données locales sur les postes terminaux).

Cycle de création de la donnée expérimentale

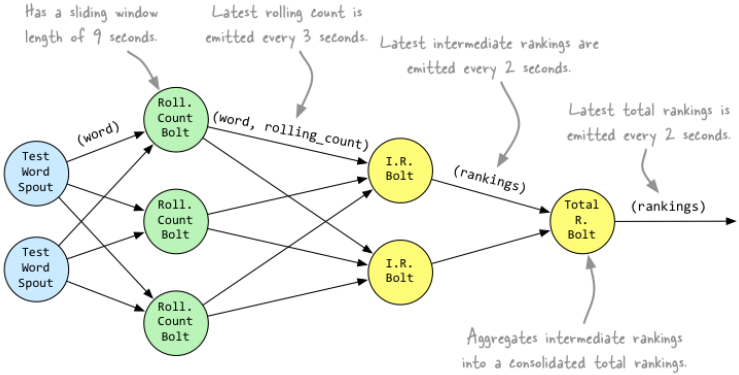
Raisonner en flux (prototype au LEGI)

- Gestionnaire de traitement de flux sur Cluster (exemple STORM `http://storm.apache.org/`)
- Mesure sur PC d'acquisition dirigé vers les **flux** (socket réseau, **pas de fichier**)
- Traitement pseudo temps réel des flux
- Stockage résultat final (souvent de taille très réduite vis à vis des mesures)

Basé sur le même principe que le traitement vidéo en temps réel. Découpage des flux en bloc et traitement parallèle des blocs.
Pas si facile actuellement de ne pas dépendre de l'administrateur système et réseau car pas facile de modifier le logiciel et de traiter plusieurs utilisateurs indépendamment sur la même plateforme numérique.

Cycle de création de la donnée expérimentale

Exemple de traitement via STORM



Traitement Visualisation des données expérimentales

Les codes de calcul ne sont pas ici référencés (OpenFoam, Yales2, Fluent, Fine...)

- Script Matlab ad-hoc
- Script Python ad-hoc (l'outil qui monte)
- Un zeste de Mathematica et de R pour certains
- Paraview (beaucoup - intégration avec OpenFoam)
- Visit (presque pas)
- Tecplot (utilisé ponctuellement)
- gnuplot (courbe) / ncview (image)
- Tableur
- ...

Traitement Visualisation - Cas de la PIV

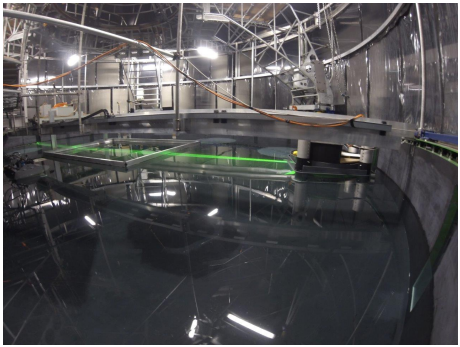
Les outils que nous poussons plus particulièrement en interne au LEGI

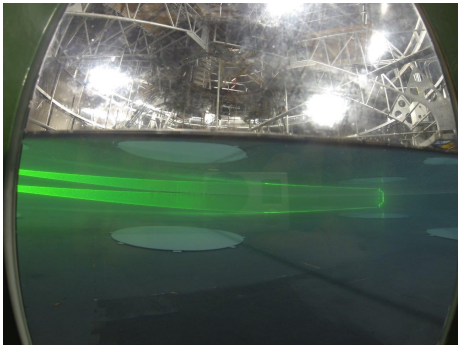
- UVmat - Boite à outils graphique Matlab orienté PIV (orienté NetCDF) `http://servforge.legi.grenoble-inp.fr/projects/soft-uvmat`
- FluidImage - Projet jeune - Boite à outils objet PIV en Python partiellement compatible UVmat (orienté HDF5) `https://bitbucket.org/fluiddyn/fluidimage`

Mais de nombreux chercheurs utilisent leurs propres outils sans vouloir les intégrer (La capitalisation du savoir faire autour d'un ou plusieurs codes est une action difficile à mener)

Traitement Visualisation - CORIOLIS - une fabrique à images

- 3 semaines de manip - 15To de données de PIV à traiter
- Nécessité de traiter rapidement pour régler les expériences suivantes





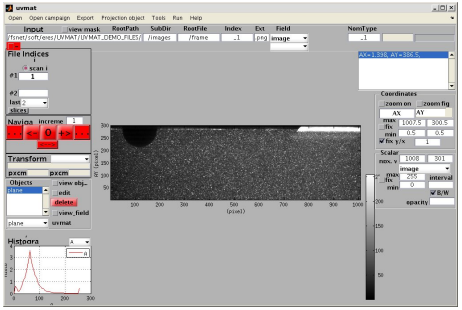
Traitement Visualisation - UVmat - Environnement Matlab

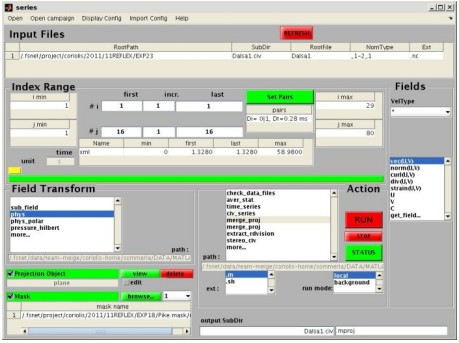
- Environnement graphique pour la calibration des mesures
- Environnement graphique de traitement de séries
- Principalement des séries d'images mais tout est possible
- Grosse gestion de fichier dans la numérotation des séries (simple, double indices...) afin d'avoir une gestion automatique des bornes (Les expérimentateurs sont très créatifs dans leur schéma de numérotation de leur fichier de mesure...)
- Un certain nombre d'algorithmes de traitement sur des séries sont de base dans la distribution. Ils peuvent être écrit en Fortran, Matlab, Python...
- Les réglages de ces algorithmes se font graphiquement puis sont enregistrés dans un fichier XML
- L'utilisateur peut rajouter des algorithmes dans sa boite à outil. Cela se fait couramment notamment sur les expériences avec de nouvelles méthodes de mesures.

Traitement Visualisation - UVmat

- Traitement synchrone dans la session Matlab ou asynchrone dans une session parallèle.
- Traitement peut être envoyés sur un cluster de calcul
- Les traitements (algorithmes) sont compilés en cas de batch Matlab (pas de jeton Matlab sur les noeuds de calcul)
- Les objets en sorties (très souvent des images) sont mappés sur un fichier NetCDF.
- UVmat a des fonctions pour passer du NetCDF au Matlab via quelques spécifications sur les formats NetCDF et les structures Matlab. Pas d'utilisation directe du HDF5 pour le moment.
- La simplicité des structures de données peut parfois être un gage de pérennité sur le long terme de celles-ci.

Traitement Visualisation - UVmat





Traitement Visualisation - UVmat

Objectifs d'UVmat

- Boite à outil graphique - ne pas refaire la ligne de commande Matlab (elle existe déjà)
- Visualiser dans UVmat les images (vidéo...) avant et après traitement
- Proposer des valeurs par défaut pour les algorithmes
- Permettre à l'utilisateur de facilement modifier ces paramètres et visualiser très rapidement le résultat (par exemple en lançant un calcul sur une petite partie de la série)
- Imposer une organisation physique des données sur les disques
- Enregistrer avec chaque dossier résultat le fichier XML décrivant complètement le traitement réalisé
- Pouvoir **reprendre** et **comprendre** n'importe quel traitement fait sur un jeu de données (il y a encore des traitements en cours sur des données de 2009)

Traitement Visualisation - UVmat - Extrait wiki d'un projet

Qualité

Le cadre UVmat permet (gratuitement - fonction bonus) de se placer dans un contexte de gestion de la qualité tant pour la partie mesure que pour les divers traitements des données expérimentales

(Le code source UVmat n'a pas toujours la qualité de ce qu'il fournit !)

Traitement Visualisation - UVmat - Organisation des données

5 - Organization of data files

All data related to the project are in: \\SERVAUTH4\share\project\coriolis\2016 or \\servauth4.legi.grenoble-inp.fr\share\project\coriolis\2016

- 0_DOC: miscellaneous documentation and reports
- 0_MATLAB_FCT: specific matlab functions
- 0_PHOTOS: photos of set-up
- 0_PIV
  - Each 'PIV' folder contains subfolders for each of the 3 PIV cameras: Dalsa (sometimes Falcon1 – it's the same thing); PCO2; PCO3 [these are named after the different brands of camera]. Other folders include PCO2.png and PCO3.png which contain processes images of the PCO cameras that are in a non-bespoke format. Other folders that can be within the Camera folder include: Dalsa.sback; Dalsa.sback_1; PCO2.png.civ; PCO2.png.civ_1; PCO2.png.civ_2; PCO2.png.sback: PCO2.png.sback_1; PCO3.png.sback_1. .sback files refer to those files where the background has been subtracted, then civ_1 contains images with the first PIV iteration as processed in UVMAT (Joel's code) and shows the raw data – with or without the rejected vectors; vectors are shown in four colours, blue = best, green = medium, red = poor, and pink = false. A box can be clicked to hide the false vectors. Civ_2 uses a spline interpretation to interpolate between vectors, so long as they are close enough to the surrounding vectors. Then interpolates all the vectors onto a regular grid. Times for the .png images are in the XML files, or netcdf files.

- 0_Processing: UVP processing scripts in Matlab
- 0_REF_FILES: files of general use (calibration data, grids ...)
- EXP1, EXP2, folder for each experiment with names given in the table below. The names refer to 'fix' for non-rotating fixed case, 'rot' for rotating case, 'str1' for the first straight position (also called position X1), and 'apex 2' , for the apex in bend 2 (also referred to as position X4).
  - Within each experiments, there is a folder with PIV imagery called 'Camera', one for ADV data – 'ADV', one for UVP data – 'UVP', and one for the data coming directly off of the Coriolis table control system 'LABVIEW'. Some experiments also contain an 'Images' folder or a 'Gopro folder' containing Gopro videos.
    - Each 'Camera' subfolder contains subfolders for each of the 3 PIV cameras: Dalsa (sometimes Falcon1 – it's the same thing); PCO2; PCO3 [these are named after the different brands of camera]. Other folders include PCO2.png and PCO3.png which contain processes images of the PCO cameras, that are in a non-bespoke format. Other folders that can be within the Camera folder include: Dalsa.sback; Dalsa.sback_1; PCO2.png.civ; PCO2.png.civ_1; PCO2.png.civ_2; PCO2.png.sback: PCO2.png.sback_1; PCO3.png.sback_1
    - Each 'ADV' subfolder, contains two sub-folders: 'nkt_files' containing raw Nortek files, and the 'mat_files' are the exported raw data in Matlab format.
    - Each 'UVP' subfolder contains two folders – one with the experiment name (which is the downstream velocity data) recorded downstream of the velocity inflection downstream of bend apex 2, and one with experiment name '_cross' which contains the cross-stream UVP data recorded at bend apex '2. These two folders contain text files for each of the probes. The convention is that Probe 1 is the basal probe, with each subsequent probe being successively higher. There are also .mfprof files which are the raw UVP data in native format. All probes are also integrated into single Matlab files. Lastly, there is a Logfile with the header file for the UVP detailing all of the parameters used in the run.
    - Each 'LABVIEW' subfolder contains: 1) a .lvm file which is a text file and contains a time-stamp, two voltages for the Conductivity probe on the traverse (C0 – Conductivity, and T0 – temperature [this latter one doesn't work]), a Trig_cam heading representing the Trigger for the PIV Cameras, Conductivity probe in the input box (C1 and T1), and C2 (this was conductivity for a second probe in the input box which was briefly used before breaking. There is always a record for this but it is just background noise. 2) _position.lvm file which is an XYZ file with a times for the movement of the traverse. 3)Some folders also contain probes.nc files. These are netcdf files and contain the vector data from the processed PIV images.

Traitement Visualisation - UVmat - Calibration

http://servforge.legi.grenoble-inp.fr/projects/soft-uvmat/wiki/UvmatHelp#GeometryCalib

4 - Methods of calibration and data processing

The MSCTI conductivity probe is calibrated after each set of experiments when the tank is drained (see Section 3.1 for full details). The ADVs have 4 heads and as such this enables some internal verification of the instrument. The ADV and UVP datasets are processed using a series of bespoke Matlab scripts. The PIV data will be processed using a bespoke script. Access to commercial PIV processing packages is also available.

The images for PIV are calibrated from images of grid put in 0_REF_FILES/Calib_absolu. The 3D calibration involves 'intrinsic parameters' of the optical system obtained from images of the same grid seen with tilt angle (put in /Calib-14-09-3D). Then rotations and translations of the calibration points are introduced to adjust the relationship between image coordinates and physical coordinates defined in section 2.2. See http://servforge.legi.grenoble-inp.fr/projects/soft-uvmat/wiki/UvmatHelp#GeometryCalib for details of the method. The calibration parameters are copied in a xml file beside each image folder with the same name (for instance PCO2.xml for PCO2/). The xml files also containing all the timing information.

All the images and processing results from the images are in the folder 0_PIV under the folder with the name of the experiment.

The images are first extracted from their initial format and written as .png images (compression with no loss of data) labelled by two indices i and j=1 to 20. The index i generally runs from 1 to 150 scanning 15 levels (then coming back to each level 10 times).

A first step in image processing after extraction is to subtract the fixed background and rescale the image intensity leading to a image folder with extension .sback. PIV results are stored as netcdf files (extension .nc) in a folder .sback.civ. These data are still in pixel displacement.

Final velocity data in phys coordinates are stored as 2D matrices under the netcdf format in folders with extension .sback.civ.mproj. They are defined on a physical grid with 1 cm mesh.

Traitement Visualisation - UVmat - Reprise des calculs

- Les calculs sur des séries peuvent planter (vont planter)
- Les erreurs sont même normales dans un contexte de bigdata
- Les outils doivent donc être robuste aux pannes
- Exemple : les outils relancés à l'identique doivent, par défaut, boucher les trous !

Accès distant

- De très nombreux Visiteurs / Collaborateurs extérieurs
- Accès distance via `ssh` (`mosh`) ou `x2go`
- OpenGL est parfois très pénible à distance !
- Autre solution possible `VNC` + `VirtualGL` (`Xrdp`, `xpra`...)
- Utilisation de Paraview en mode client serveur (pas de transfert du modèle sur le réseau)
- Traitement chez nous dans le mesure du possible
- Données conservées en interne au laboratoire
- Plus facile pour les aider à traiter les données en cas de besoin (ou de bogue logiciel)
- Attention à limiter la consommation de la bande passante (ressources réseaux)
- Être robuste aux coupures réseaux est un plus !

Diffuser ses données en OpenAccess (libre accès)

- Open or not open ?
- Faut-il faire de l'OpenAccess ?
- Faut-il partager nos données ?

Diffuser ses données - libre partage

- Le partage est au cœur de la connaissance. . .
- On a tous utilisé des données d'un autre un jour ou l'autre !
- Certains chercheurs ont réellement besoin de nos données
- La diffusion est un des moyens du contrôle de la véracité

Mur sismique d'un laboratoire de recherche à Mendoza / Argentine

Diffuser ses données en OpenAccess

- PyDAP (protocole DAP - OpenDAP - Data Access Protocol)
- Tête NAS exporte les données en Read-Only (sécurité) vers le serveur DAP (pas de modification possible)
- Question ouverte : diffusion de l'intégralité du projet ou d'un partie après ménage (risque de prendre plus de 2 ans à faire). Problème éventuel de confidentialité.
- Charte de diffusion en OpenAccess des données expérimentales en deux versions. Première version simple (V1), version après juriste de météofrance (V2) !
- Temps délais définit dans les contrats (5 ans) non respecté
- L'idéal est de mettre les données sur le serveur DAP le plus vite possible en accès restreint au groupe
- Retour expérience MEOM : les principaux utilisateurs du serveur DAP sont les membres du projet eux-mêmes !

Sauvegarde

- On atteint de tel volume que nous n'avons pas les moyens de tout sauver
- Parfois les données bruts sont jetables car les manips sont faciles à refaire
- Actuellement, les projets Coriolis sont sauvés sauf certains fichiers
- L'idéal serait de trier à la source ce qui est à sauver (on sauve rien sauf certains fichiers)

- `backuppc` - Très pratique pour l'utilisateur mais devient très lent sur des gros volumes - Remonte sur 3 mois sur les HOME
- `rdiff-backup` - Très rapide sur les gros volumes - Remonte sur 6 mois
- `cp` peut-être 3 fois plus rapide que `rsync` lors de la première copie

Sur des projets de plusieurs années comme Coriolis, il faudrait pouvoir remonter les sauvegardes sur plusieurs années (archivage ?)

Conclusion

- Chaque expérience est unique
- Le matériel évolue, des nouveaux capteurs. Rien n'est fixe
- Le volume de données explose
- Signer une **charte** de l'OpenData dès le début, mettre une **licence** sur les codes
- Mettre en OpenData (intranet) le plus rapidement possible
- Des points de convergence sur la structuration des données entre les expériences
- Enregistrer les paramètres des traitements effectués au plus près des données
- Une **approche méthodique** améliore la **qualité** inter-projet
- Attention a **ne pas sous investir** sur le matériel (réseau, stockage, calcul)
- Ne pas oublier les sauvegardes

Plus haut col routiers des Amériques

Premier village à 50km / Pose 2014

Merci de votre attention