

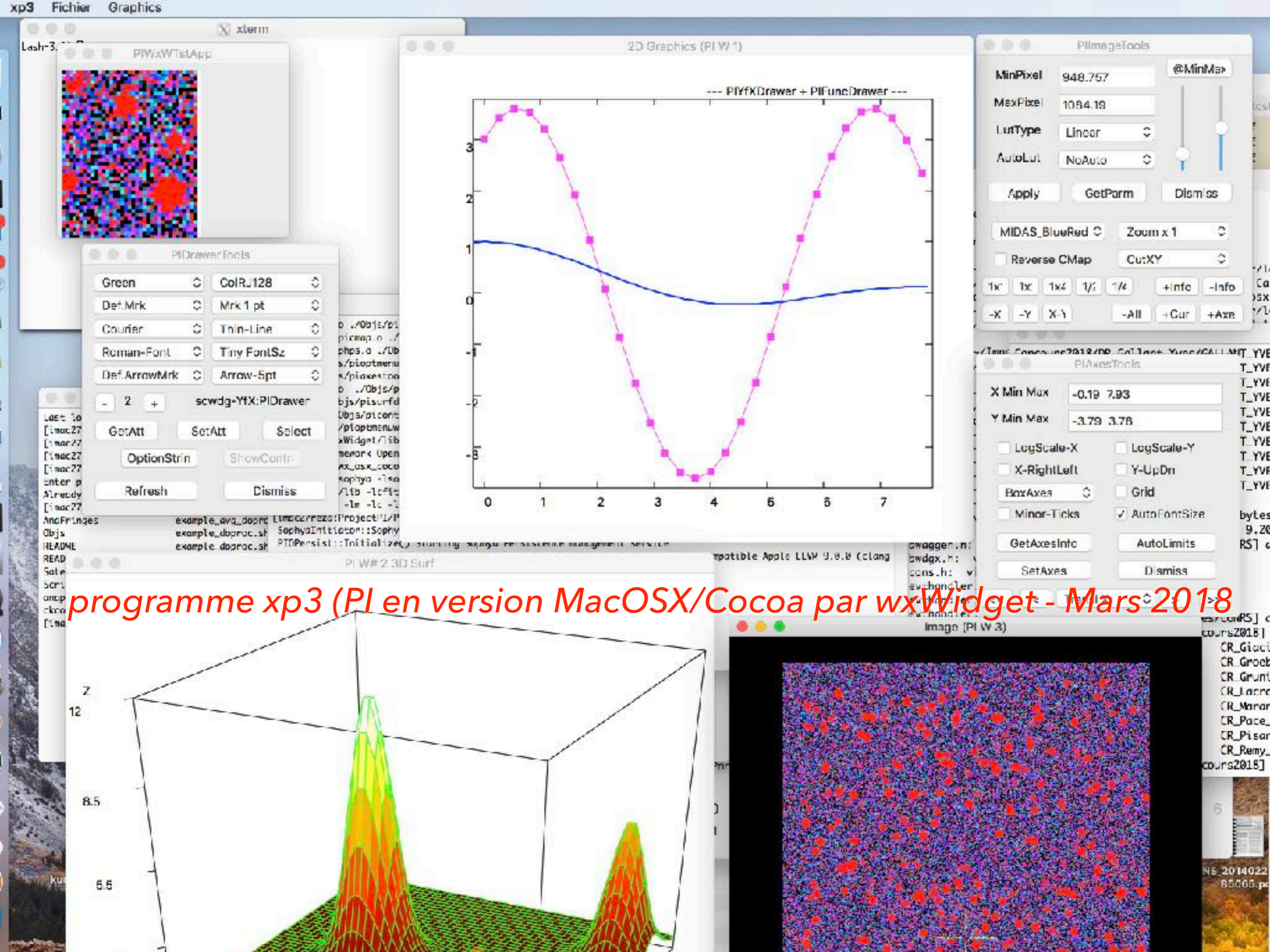
PI , PIAPP WITH WXWIDGET

R. ANSARI, AVRIL 2020

- Librairie PI (Peida Interactif) : Librairie de classes (C++) pour composants GUI et graphiques/UI - début de développement en 1996
- Implémentation X11/Motif et Mac OS8/Code Warrior initialement - X11/Motif uniquement depuis 2000
- PI n'a plus grand intérêt en tant que librairie/framework GUI (depuis +15ans), avec les librairies comme Qt, wxWidget
- PI est super basique comparé à Qt et wxWidget - peut donc avoir un intérêt comme base pour une implémentation spécifique
- Les mécanismes de PIDrawer , PIScDrawWdg et même PI3DWdg restent très utiles, avec les outils de gestion d'axe, attributs graphiques ... et aussi PImage pour visualisation tableaux et images astro - Pas d'équivalent ds wxWidget

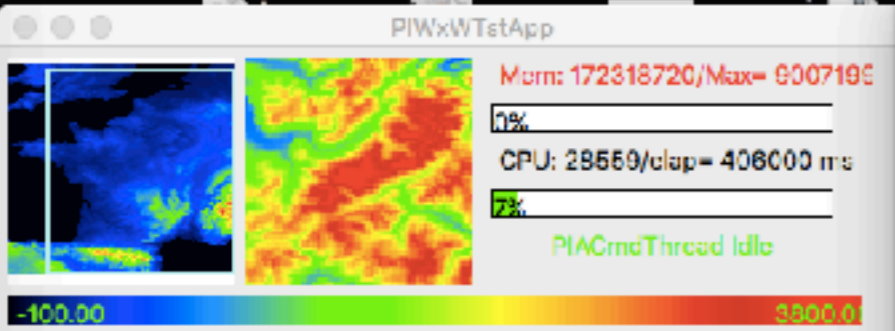
- PI sert de librairie support pour piapp, programme d'analyse interactif , outil compagnon de SOPHYA
- Services graphiques regroupés principalement sur quelques classes , PIStdImgApp en particulier
- Avec quelques améliorations, piapp qui combine la manipulation interactive de représentations graphiques d'objets avec pas mal de possibilités de scripting pourrait rester (ou devenir) un outil complémentaire des pythons et autres root ?
- Implémentation de PI sur la librairie de classes c++ wxWidget commencé en 2016
- L'application piapp a été profondément modifié en 2004 pour passer en multi-thread - on va voir que cela a été une source importante de difficultés pour la version PIWxW

- Création du répertoire PIWxW ds ProjectPI/PI en décembre 2016
- Implémentation des classes de base PIWdgWxW , et des widgets standard GUI : PILabelWxW, PIButtonWxW, PIMenuBarWxW courant 2017
- Mars 2018 : première version presque opérationnelle de la librairie PI en version wxWidget, avec presque tous les widgets GUI standard, PIBaseWdgWxW implémentation partielle des tracés graphiques dans la classe PIGraphicWxW
- Septembre 2018 : implémentation PIWxW complétée avec les classe PIScrollBarWxW, PIListWxW, PIFileChooserWxW, tracés graphiques et gestion des fontes complétés ...
- Pas d'autres modifications probablement jusqu'au tag de la version SOPHYA , PI , piapp V=2.5 en Janvier 2019



programme xp3 (PI en version MacOSX/Cocoa par wxWidget - Mars 2018)

- Reprise du développement de PIWxW pour arriver à compiler piapp fin octobre 2019, Novembre, décembre 2019, puis vacances de Noël 2019
- Travail sur PIStdImgApp (qui m'a posé pas mal de problème à cause de l'architecture multi-thread de piapp), debug de l'objet PIConsole (qui hérite désormais de PIContainer, et comprend un PIConsTextWdg et un PIScrollBar)
- Fin Dec 2019 : nouvelle logique mise en place pour la gestion des interactions graphiques dans l'environnement multi-threads de piapp - Prise en charge par une classe auxiliaire **EventLoopCmdHandler** - Mise en place aussi d'un mécanisme de requêtes pour la fenêtre gestionnaire d'objets
- Janvier 2020 : debug divers , suite aussi aux tests effectués par Guy - en particulier problème de lenteur, dû à l'utilisation de SOPHYA::Periodic (intervalle 1s) au lieu de PIPeriodic (intervalle ~5ms)



PIImageTools

MinPixel: -100 @MinMax
MaxPixel: 3800
LutType: Log
AutoLut: NoAuto

Apply GetParm Dismiss

ColBR128 Red. / 2
 Reverse CMap CutXY

1x 1x 1x 1/2 1/4 +Info -Info
-X -Y X-Y -All +Cur +Axe

AboutPIApp

piapp
SOPHYA

© 1996-2019
<http://www.sophya.org>

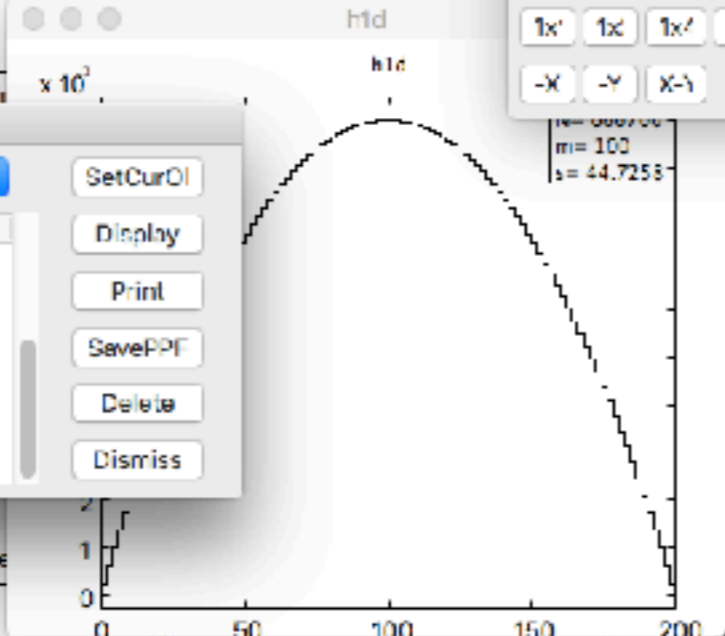
Software Version: SOPHYA- 2.535 PI-1.85 piapp- 4.92

OK

..... starting piapp

SOPHYA Version 2.5 Revision 35 (V_Nov2019) -- Dec 8 2019 01:22:46

Cmd>

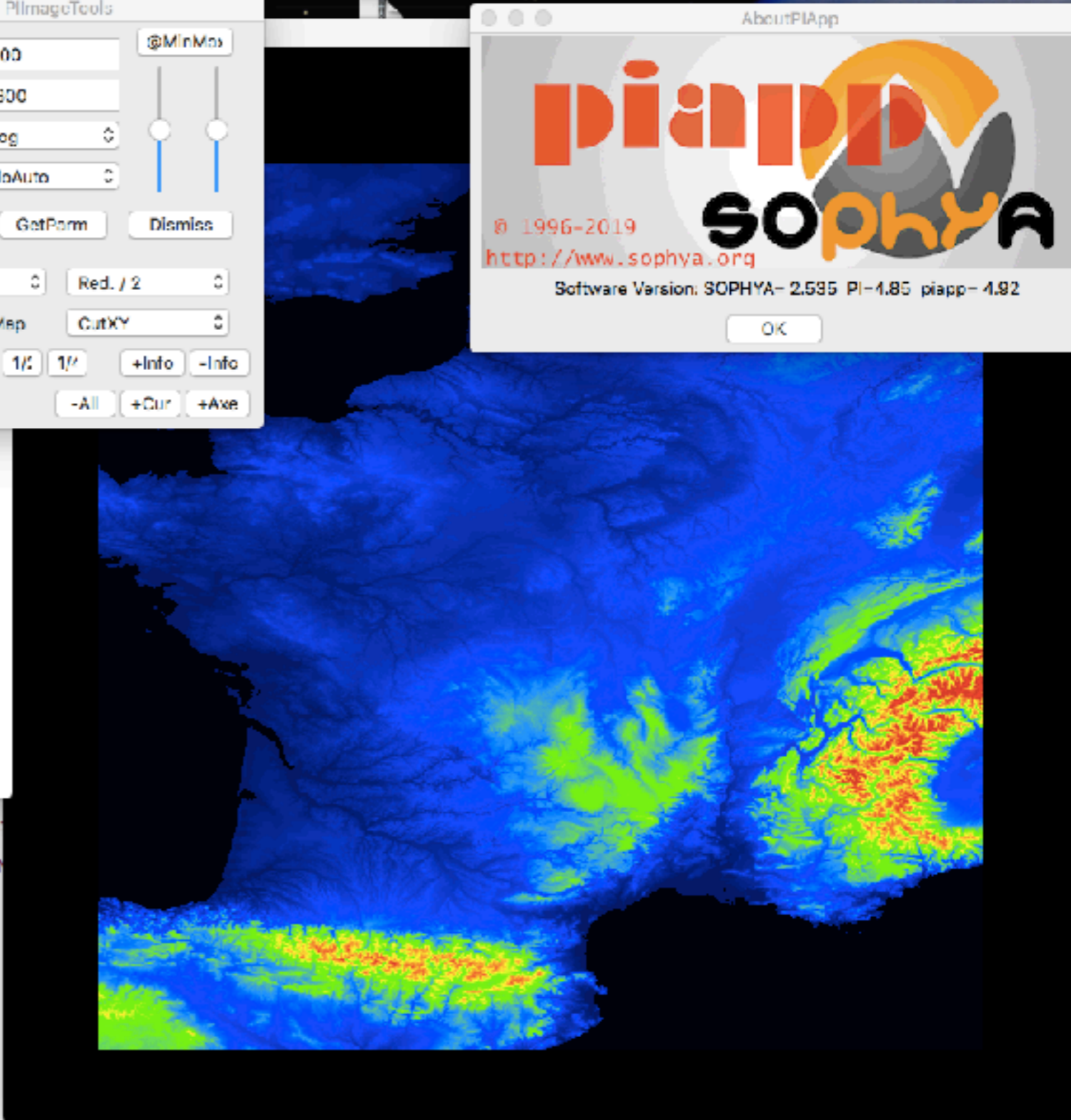
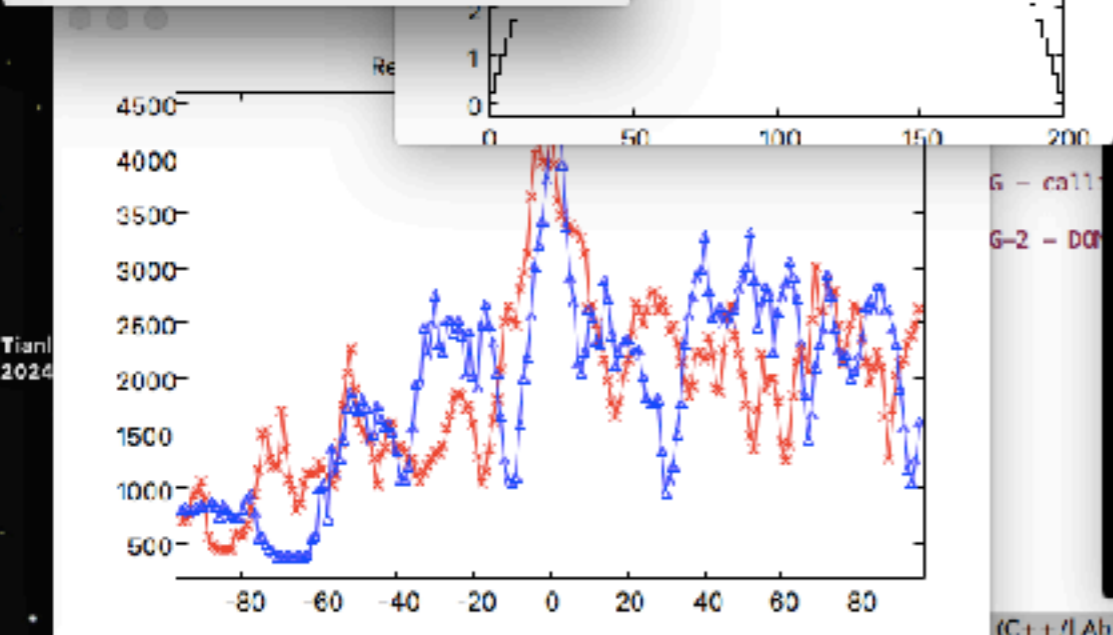


abjmgr

/home

SetCurOI
Display
Print
SavePPF
Delete
Dismiss

- nt31 (T=...
- nt32 (T=...
- vx1 (T= T..
- vx2 (T= T..
- vy1 (T= T..
- francoctop..



LA SUITE (COURT TERME)

- Installer et tester PI et piapp version wxWidget sur Mac OS (Cocoa) et X11 (GTK)
- identifier les problèmes - corriger ou en faire part à l'équipe de développement wxWidget
- Partir sur la base du makefile ad-hoc ds PI/PIWxW
- Restructurer le code PI en **PIBase**, **PIX11Motif**, **PIWxW** et **PI**
- **PIBase** regrouperait classes support de base, ex *PIGraphicAtt*, les classes définissant l'interface xxxGen, ex *PIWdgGen*)
- **PIX11Motif** (ou PIX11) l'implémentation des classes xxxGen avec X11/Motif
- **PIWxW** l'implémentation des classes xxxGen avec wxWidget
- **PI** regrouperait alors uniquement les classes qui sont génériques, comme **PIScDrawWdg**
- les répertoires **Plext**, **PIGcont**, **ProgPI** resteraient inchangés
- Il faudra adapter les Makefiles et script ...

LA SUITE (MOYEN TERME)

- Voir comment intégrer le vrai 3D (PIOpenGL ou ??)
- Compléter les fonctionnalités non encore implémentés ds PIWxW (Copier/coller, drag & Drop (Mac OS X uniquement) ...)
- Construire une application complète, installable sur MacOS (Guy) et Linux
- Explorer l'utilisation d'objets wxWidget sans les implémenter en version X11 : Affichage HTML (wxHtmlWindow) , affichage de Tables (wxGrid)
- Améliorer / optimiser l'affichage d'images (code de l'interface des tableaux P2DArrayAdapter , PImage) , décodage coordonnées astro (fait en partie - intégration WCSTools) ...
- Améliorer / compléter l'affichage des cartes sphériques (projections ...)
- Intégration de l'interpreteur LUA
 - (voir SophyaProgs/Eval/DivTst/luatst.cc)