

# TP NISP du 07-02-2011

Michael Baudin (INRIA)  
Jean-Marc Martinez (CEA)

January 2011

## Abstract

Ce document est une feuille de route pour le TP du 07-02-2011.

## Contents

<a href="#">1 Contact</a>	1
<a href="#">2 Documents</a>	1
<a href="#">3 Fichiers</a>	2
<a href="#">4 Exercices</a>	2

## 1 Contact

michael.baudin@scilab.org

## 2 Documents

Pour s'initier a Scilab: "Introduction to Scilab", Michael Baudin, Consortium Scilab DIGITEO, 2010:

<http://forge.scilab.org/index.php/p/docintrotoscilab/downloads/>

"In this document, we make an overview of Scilab features so that we can get familiar with this environment. The goal is to present the core of skills necessary to start with Scilab. In the first part, we present how to get and install this software on our computer. We also present how to get some help with the provided in-line documentation and also thanks to web resources and forums. In the remaining sections, we present the Scilab language, especially its structured programming features. We present an important feature of Scilab, that is the management of real matrices and overview the linear algebra library. The definition of functions and the elementary management of input and output variables is presented. We present Scilab's graphical features and show how to create a 2D plot, how to configure the title and the legend and how to export that plot into a vectorial or bitmap format."

Pour programmer avec Scilab: "Programming in Scilab", Michael Baudin, 2010:

<http://forge.scilab.org/index.php/p/docprogscilab/downloads/>

”In this document, we present programming in Scilab. In the first part, we present the management of the memory of Scilab. In the second part, we present various data types and analyze programming methods associated with these data structures. In the third part, we present features to design flexible and robust functions. In the last part, we present methods which allow to achieve good performances. We emphasize the use of vectorized functions, which allow to get the best performances, based on calls to highly optimized numerical libraries. Many examples are provided, which allow to see the effectiveness of the methods that we present.”

Le module Scilab NISP est open-source:

<http://forge.scilab.org/index.php/p/nisp/>

La page wiki presente le module:

[http://wiki.scilab.org/NISP\\_Module](http://wiki.scilab.org/NISP_Module)

Le module est disponible dans ATOMS:

<http://atoms.scilab.org/toolboxes/NISP>

Pour comprendre le module Scilab NISP: ”NISP Toolbox Manual”, Baudin, Martinez, 2011

<http://forge.scilab.org/index.php/p/docintrotoscilab/downloads/>

### 3 Fichiers

J’ai a votre disposition un repertoire contenant les fichiers suivants.

- docs: les trois pdf evoques plus haut
- roadmap: ce document en  $\text{\LaTeX}$
- scripts: les scripts utilises dans ce TP
- bonus: des scripts supplementaires, utiles, mais pas strictement necessaires
- bonus/demos: des demonstrations du module NISP (.sce)
- bonus/scripts: des fonctions supplementaires (.sci)

### 4 Exercices

La session s’appuie sur le ”NISP Toolbox Manual”. Nous allons realiser les exercices suivants.

- Objectif: se familiariser avec Scilab et la classe ”randvar”. Section 4.3.1, ”A sample session”. scripts/demo\_randvar1.sce Exercices: voir la page d’aide `help randvar`, appeler `nisp_size`, `nisp_tokens`, `nisp_getlog`, changer les parametres de la loi, changer pour une loi uniforme, une loi exponentielle, une loi log-normale.

- Objectif: se familiariser avec la classe `setrandvar`. Section 5.2.1, "A Monte-Carlo design with 2 variables". `scripts/demo_setrandvar1.sce` Exercices: appeler `setrandvar_getlog`, `setrandvar_size`  
`setrandvar_tokens`, `setrandvar_save`, utiliser un plan LHS et afficher le plan avec `plot`, un plan "QmcSobol" et afficher le plan, idem avec Petras
- Objectif: appliquer l'analyse de sensibilité a un modele affine par les indices SRC. Section 7.4, "Using scatter plots". `scripts/affine_SRC_scatter.sce` Exercices: changer les coefficients du modele (exemple:  $2X_1 - 4X_2 + X_3 - X_4$ ), changer les parametres des lois (changer la moyenne, l'ecart-type), voir l'evolution des indices quand le nombre d'experiences augmente
- Objectif: appliquer l'analyse de sensibilité a un modele simple par une methode naive. `scripts/product_naive.sce` Exercices: compter le nombre d'appel a la fonction, voir l'evolution des indices quand le nombre d'experiences augmente
- Objectif: appliquer l'analyse de sensibilité par la methode de Sobol. Section 7.11.6, "The Ishigami function by the Sobol method". `scripts/ishigami_hommasaltelli.sce` Exercices: compter le nombre d'appel a la fonction, comparer avec la methode precedente, voir l'evolution des indices quand le nombre d'experiences augmente
- Objectif: appliquer l'analyse de sensibilité par chaos polynomial. Section 6.2.2, "The Ishigami test case". `scripts/polychaos_ishigami.sce` Exercices: essayer avec un plan Monte-Carlo, augmenter le nombre d'experiences et voir si les indices convergent, essayer avec un plan QMCSobol, changer le modele et calculer les indices (verifier que le calcul a la main n'est pas simple)

Pour ceux qui en veulent plus.

- Objectif: se familiariser avec la classe `setrandvar`. Section 5.2.3, "A LHS design". Le script est a ecrire... Exercices: faire un plan avec 10 points et faire un plot (voir que les points sont au centre de chaque cellule)
- Objectif: appliquer l'analyse de sensibilité par chaos polynomial. Section 7.6, "The effect of the interactions" `bonus/demos/polychaos_product2.sce` Exercices: voir que le resultat calcule est tres proche du resultat exact, voir que la somme des indices n'est pas 1, essayer `bonus/demos/polychaos_product.sce`, calculer les indices de sensibilité exacts, comparer les indices exacts avec les indices calcules, voir que le resultat calcule est plus proche que precedemment