Dassault Aviation experiences on the interoperability of Matlab and Scilab: case study on the 2D radar image formation

Bruno Patin & Gilles Zalamansky

17th November 2004

Abstract

In order to make Scilab widely adopted inside our company, the first step is allowing Scilab to interoperate with Matlab. It means to allow the user to decide on an opportunity basis to use Scilab or Matlab. To achieve such a goal, we need an easy way to: a/ translate Scilab scripts from/to Matlab; b/ load and save Matlab data files from Scilab.

Scilab scripts are under development to take into account these needs. The two scripts, **savematfile** and **loadmatfile**, allow the user to work inside Scilab with Matlab formatted files ([1]) and the script **translatepaths** allows him to translate Matlab script into Scilab. In order to assess these scripts, we worked with scripts developed to compute 2D radar images from measurements ([2]). -This paper will detail in its first part how we see the integration of Scilab scripts with Matlab ones. Then, in the second part, it will describe what is done during the computation and, in the third part, it will present the results of this work. It will conclude with what has to be done in order to allow a complete interoperability.

We express our deepest thanks to Mr Couvert and Mr Steer from INRIA who corrected and improved their tools following our requests.

1 Integration methodology

Matlab happens to be a de facto standard in the industry. The point of view of replacing it completely with Scilab is not accessible. Therefore using Scilab means to interoperate correctly with Matlab in order to allow its use throughout our company when Matlab is not necessary or when we have more specific needs. For example, when launching a computation on several computers is useful. The following figure describes the way to do it as we see it in our company.

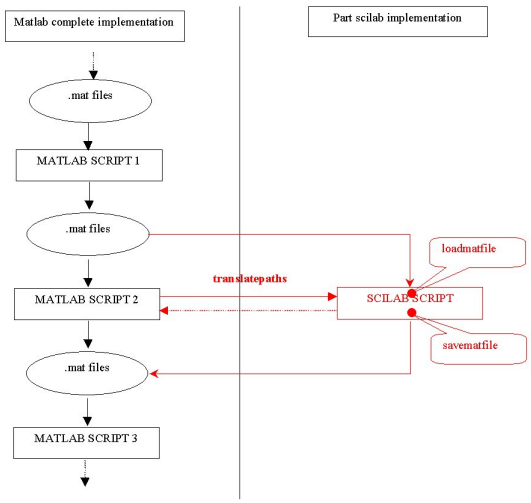


Figure 1: How to insert Scilab scripts

As illustrated, the user, inside an existing Matlab processing flow, wants to run some scripts in Scilab. He extracts the Matlab script and then applies the **translatepaths**, inside the Scilab tool, on the directory where he puts this script. Once he has done this, he can load the mat files thanks to the **loadmatfile** Scilab script and apply the translated Scilab script to them. Then, the result is translated back to the

Matlab format thanks to the **savematfile** script.

This methodology can be used for several reasons:

- you want to give to someone else the development of the second script that he cannot develop on Matlab;
- you need several instances of the same computation at the same time that will block several instances of Matlab when it is not necessary;
- you work in cooperation with laboratories and so forth.

2 2D radar imaging computation

The case study that we have chosen to present here is extracted from the radar field. We consider the image synthesis from 2D radar measurements. Any object, when illuminated by an electromagnetic wave, gives another secondary wave that can be compared with the incident one (which is called the Radar Cross Section). This quantity can be measured for a set of wave frequencies (it gives us a spectrum, H). The picture that follows shows a 2D acquisition device working on the Petit Duc plane.

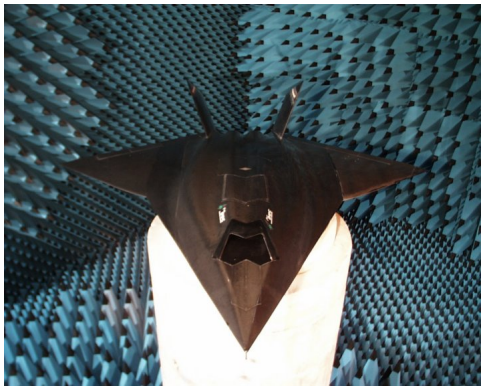


Figure 2: Petit DUC plane in its 2D measurement environment

After the adequate processing, we can create an image, α, that shows the hot spots of the object: the surface elements that contribute to what is reflected. Here is the equation that relates image,α, to spectrum, H.

$$\alpha\left(\overrightarrow{r}\right) = \int\int\int H\left(\overrightarrow{\sigma}\right) e^{i2\pi\overrightarrow{\sigma}\cdot\overrightarrow{r}} d^3\overrightarrow{\sigma} \qquad (1)$$

Due to measurements reasons, instead of directly applying a cartesian Fourier transform, we use the Fourier-Bessel transform. The following equations describe this transform.

$$H\left(\sigma,\phi\right) = \sum_{n=-\infty}^{\infty} H_n\left(\sigma\right) e^{in\phi} \qquad (2)$$

$$\alpha\left(r,\phi\right) = \sum_{n=-\infty}^{\infty} \alpha_n\left(r\right) e^{in\phi} \qquad (3)$$

$$\alpha_n\left(r\right) = i^n 2\pi \int_0^{\infty} \sigma H_n\left(\sigma\right) J_n\left(2\pi\sigma r\right) d\sigma \qquad (4)$$

$$H_n\left(\sigma\right) = i^{-n} 2\pi \int_0^{\infty} r\alpha_n\left(r\right) J_n\left(2\pi r\sigma\right) dr \qquad (5)$$

J_n in (4) and (5) is first kind cylindrical and spherical Bessel functions.

These equations are the continuous forms of the transformation. In order to apply them on discrete data, like measurements, new expressions have to be derived from them that in fact introduce the extended shannon criteria for the Fourier-Bessel transform. It is these discrete expressions that are put into the Matlab scripts.

The case study presented in this paper is a dedicated calibration object which properties are well known as far as hot spots are concerned.

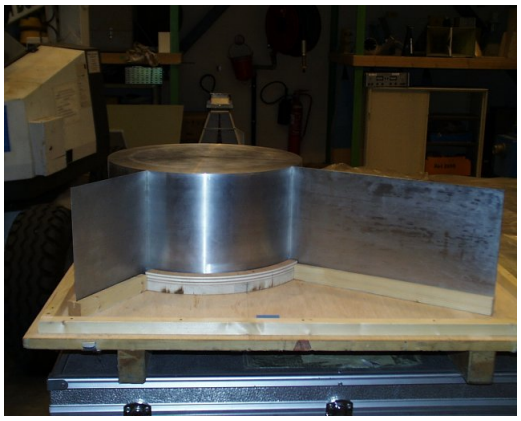


Figure 3: The calibration object

Below are the data that we used:

- Spectrum: measurements are made from 2 to 18 Ghz (frequency sweep) and from 0 to 360° (rotation around the z axis). We show for each point the logarithm (to have the necessary contrast) of the module of the complex number measured.

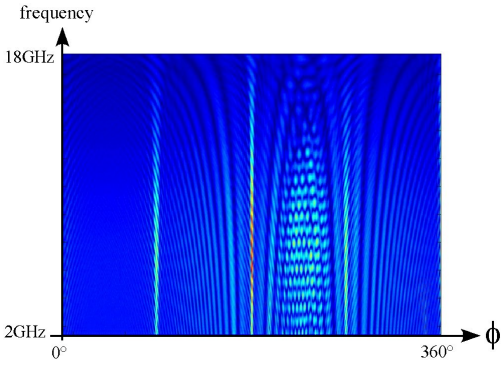


Figure 4: Calibration object spectrum

- Image: the axis are x and y and the greatest extent (3x3 mxm) is ruled by the corresponding increments in the hologram.

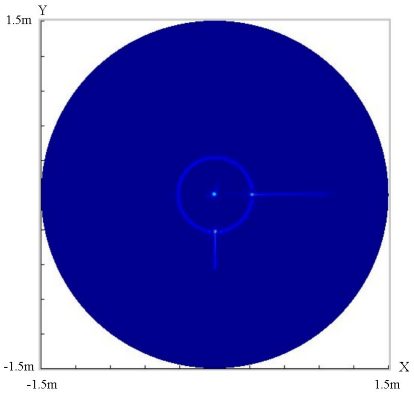


Figure 5: Calibration object radar image

The goal is to have as exactly as possible the same image synthesized when using the Scilab scripts.

3 Assessment

3.1 Loading and saving Matlab files

For each of the two following structures, we are able to create them in Matlab, load them in Scilab, read them and create some new ones in Scilab, then write them and use Matlab to open them directly. We show the equivalent reading of Matlab and Scilab on the first structure and the writing capability on the second.

LATTICE structure

This structure is dedicated to the manipulation of implicitly structured data. One of its fields contains binary data, The others allow the user to decode these binary data. Following can be seen how each tool sees the structure after loading it from the Matlab file.

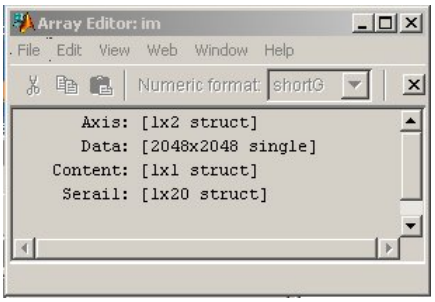


Figure 6: The LATTICE seen from MATLAB

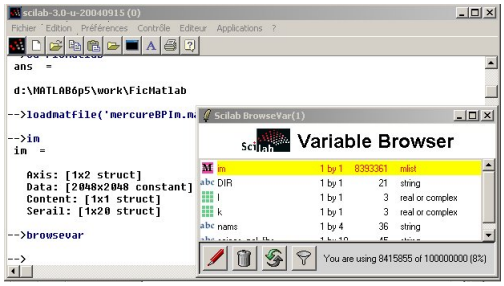


Figure 7: The LATTICE seen from SCILAB

MESH structure

This structure is dedicated to the manipulation of mesh. Figures 8 and 9 show the modification of a name in the mesh in Scilab that can be seen in Matlab after reading the mat file saved thanks to the **savematfile** Scilab command.

In order to verify the binary contents, we applied in both tools the command:

plot(SMESH_outModified.Resultats.Resultat.Datas.Abscisse.Valeurs)

The result can be seen on figures 10 and 11.

3.2 Script translation

Three scripts are used to compute the image. We'll only consider for the translation the first two ones. We present the code of the Matlab script (see figure 12) and its corresponding translation (see figure 13).

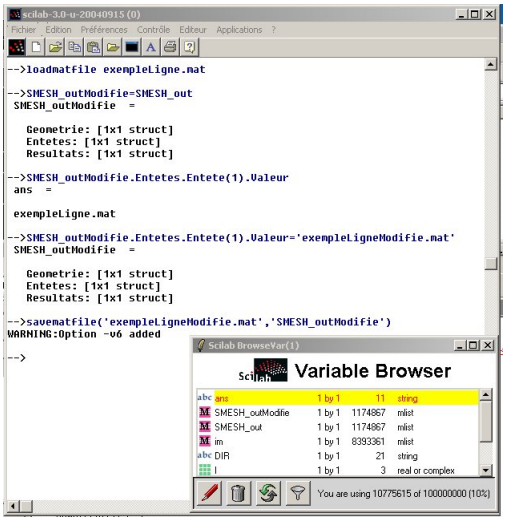


Figure 8: The MESH seen from SCILAB

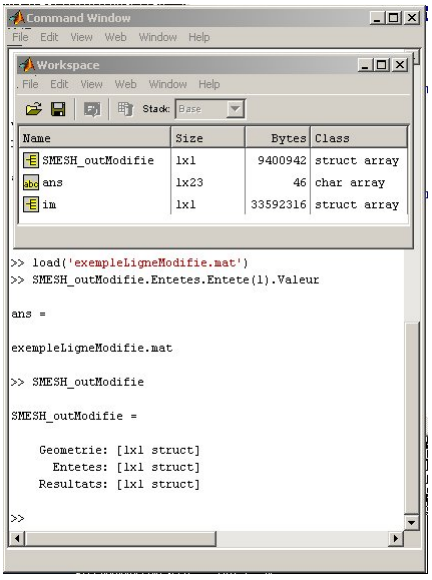


Figure 9: MESH seen from MATLAB

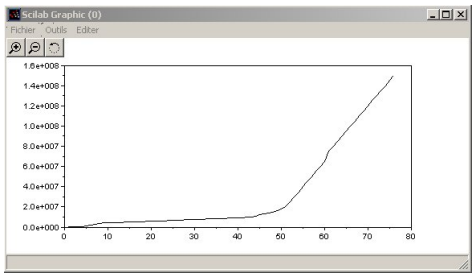


Figure 10: Scilab plot of the mesh abscisse

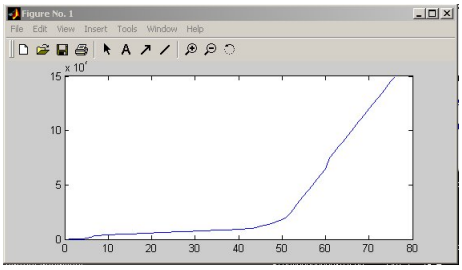


Figure 11: Scilab plot of the mesh abscisse

We remind that to run the translation we execute under Scilab the **translatepaths** script.

- **init_thd_riemann**: this script precomputes the weighing coefficients of the Fourier-Bessel transform;

  The goal to be reached during the translation was to let the Matlab code as it is. The difficulties encountered (not describing the various bugs that previously existed) were on the Scilab wrappers that created functions which mirrored the Matlab ones found in the code. Every time a **mtlb_** prefix exists, it means a wrapper. In our case, the problem was mainly the **besselj** function that was really different of its Matlab counterpart.

- **tfd2d**: this script computes the image using the bessel functions.

  The main problem we encountered was the name of one of the variables because its name was a

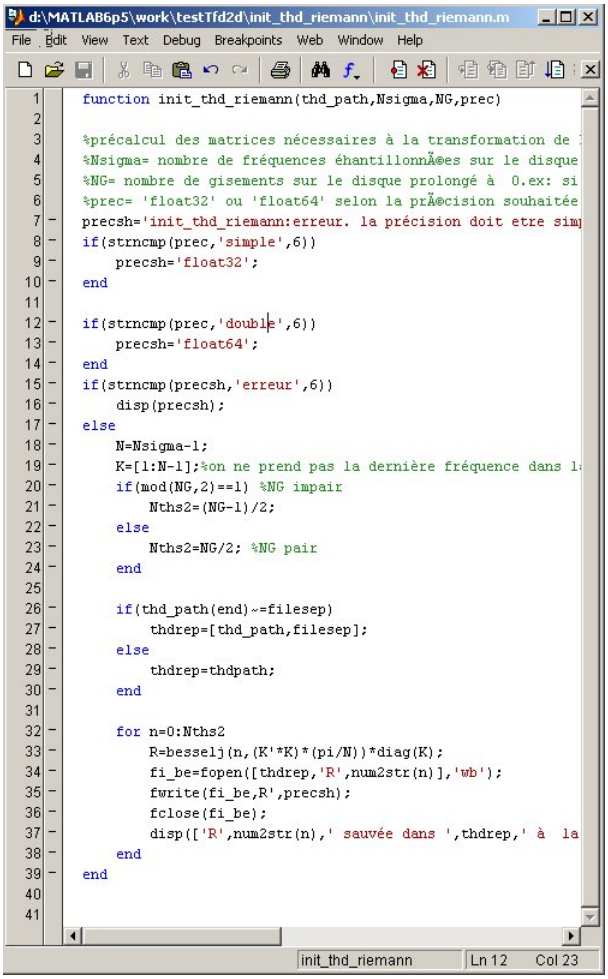


Figure 12: init_thd_riemann Matlab code

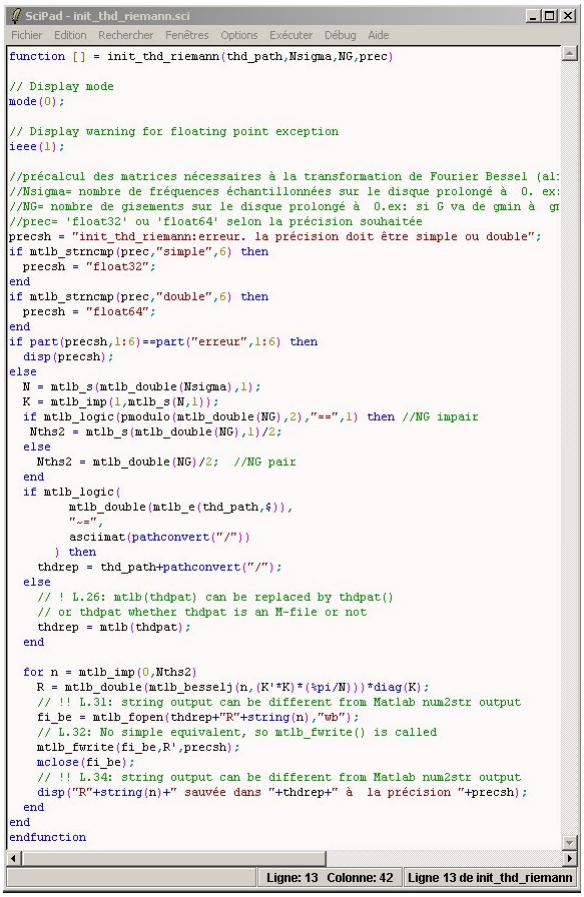


Figure 13: init_thd_riemann Scilab code

reserved name for Scilab. This bug has not been corrected during this work but will be taken into account in a new version.

- **showcart**: this script visualizes the image in the cartesian format (see figure 5). The polar computation method gives us an image indexed by distance and angle instead of x and y. As there were a lot of graphic functions in this script, the translation gives us a Scilab script for which a lot of wrappers lacks. Figure 4 shows the output of the function.

We used the two processing flows, one only with Matlab scripts, the other with two Scilab scripts and one Matlab script, to verify that the results were exactly the same. In fact, we used a unix diff function on the binary datas produced that gives no differences at all. It means that the underlying implementation of the used math functions (fft, bessel, etc.) are issued of the same mathematical libraries (fftw, gsl, etc.).

4 Conclusion

4.1 pros

- The capability of reading and writing Matlab files from Scilab is a great step toward the interoperability as it allows us to directly integrate Scilab scripts that will use Matlab files as inputs and Matlab files as outputs.
- The translation capability from Matlab to Scilab has been verified on a real test and it works.

4.2 cons

- The Matlab file writing capability has not been verified entirely. For example, we do not know if we can write objects.
- In order to realize translations, we had to correct wrappers. This work has to be conducted for the translation to be effective.
- There is no translation tool from Scilab to Matlab at this step. It will be necessary to close the loop.

- The wrappers for the graphic parts of Matlab are in their infancy.

4.3 Future

- For Scilab to interoperate correctly with Matlab, the tests have to go on. We need to cover the most used Matlab functions. A way to do it could be workshops where each interested partner could be present with his scripts in order to achieve translation with INRIA help. The different modifications that would be made by INRIA during these workshops could be included in a release delivered some days after.
- Matlab scripts could be developed in order to control the Scilab tool through Matlab.
- Matlab works a lot with toolboxes which means that these toolboxes have to be translated when needed in order for Scilab to function. This raises technical and juridic matters.

References

[1] Mat-File Format mathworks document Version 7

[2] JINA 2004 - Dassault Aviation poster on radar imagery.

Scilab conference 2-3 december 2004

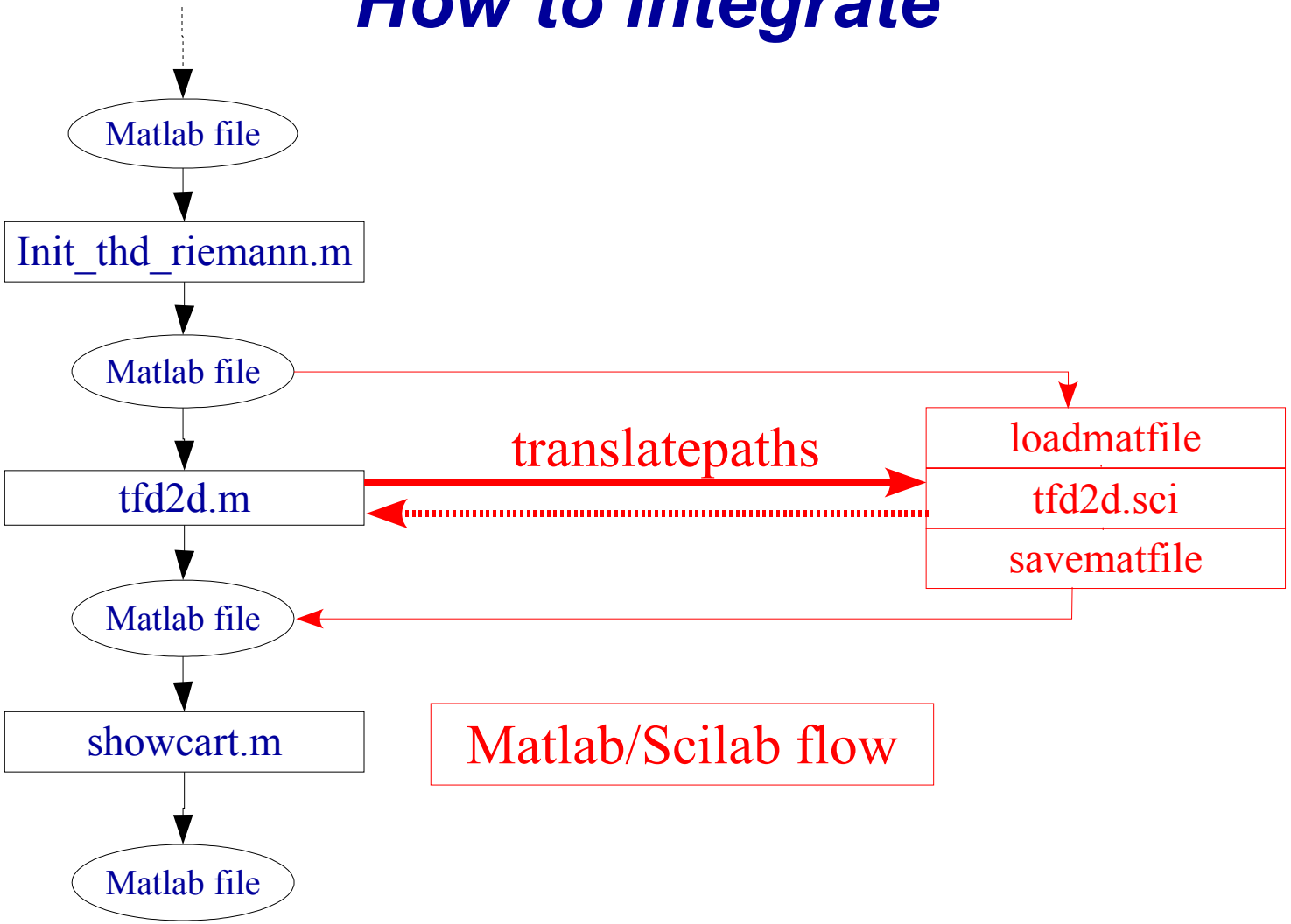
SCILAB/MATLAB interoperability

DASSAULT AVIATION
Division DÉFENSE

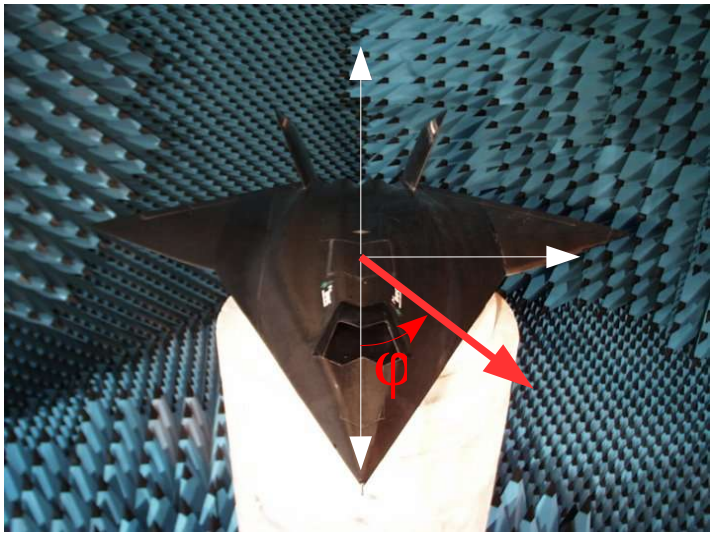
Contents

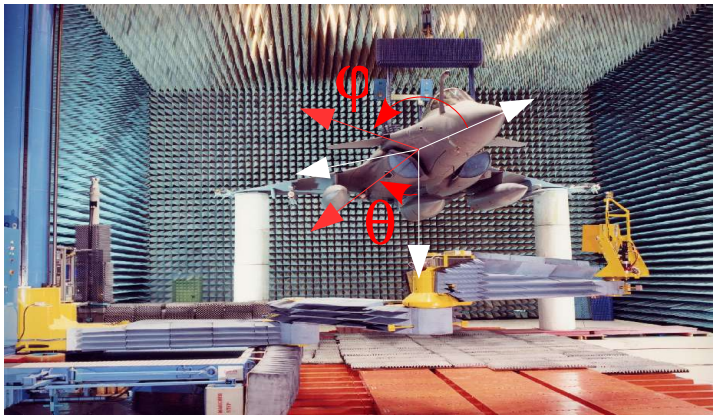
- How to integrate
- 2D Radar Image computing
- Assessment
- Conclusion

How to integrate



2D Radar Image computing (1/3)





Frequency Signal to be analysed

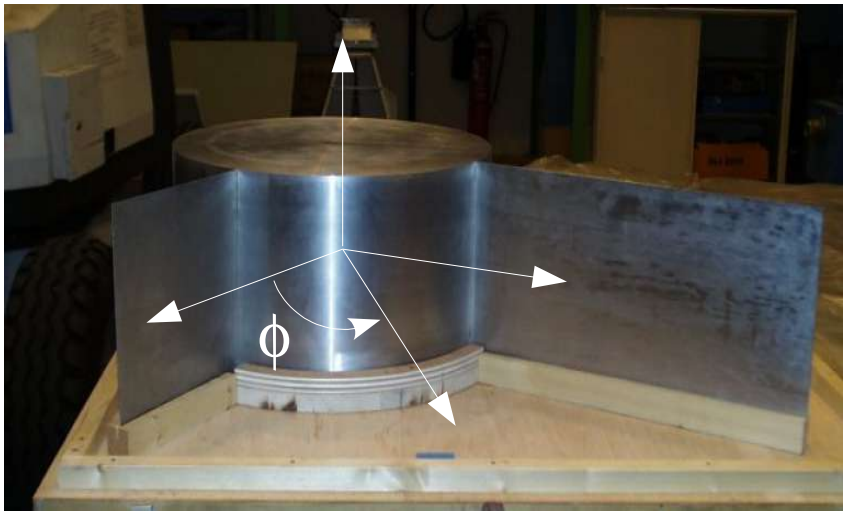
$$H_{\pi\nu}(\vec{k}) = \oiint \alpha_{\pi\nu}(\vec{OF})\, e^{-i\vec{OF}\cdot\vec{k}}\, OF^2\, \delta\Omega$$

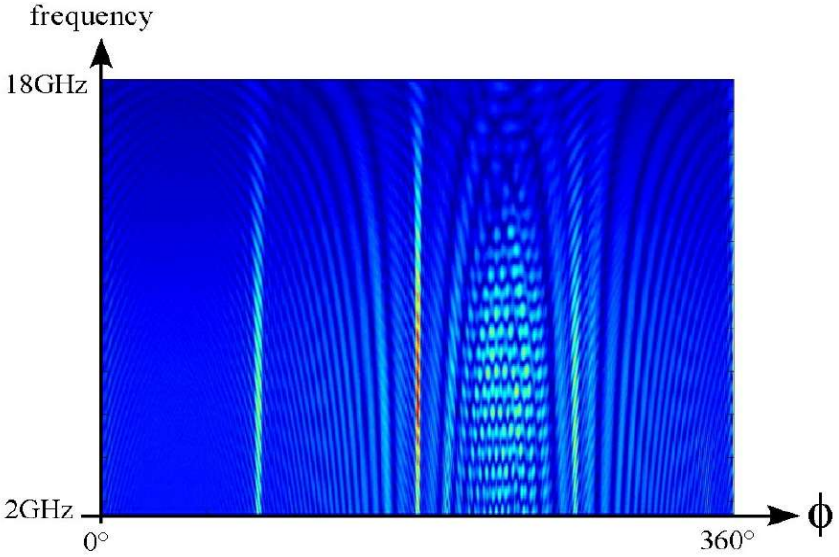
$$H_{\pi\nu}(\vec{k}) = \iiint \alpha_{\pi\nu}(\vec{r})\, e^{-i\vec{r}\cdot\vec{k}}\, d\vec{r}^{\,3}$$

Image to be computed

$$\alpha_{\pi\nu}(\vec{r}) = \iiint H_{\pi\nu}(\vec{\sigma})\, e^{i\,2\pi\vec{r}\cdot\vec{\sigma}}\, d\vec{\sigma}$$

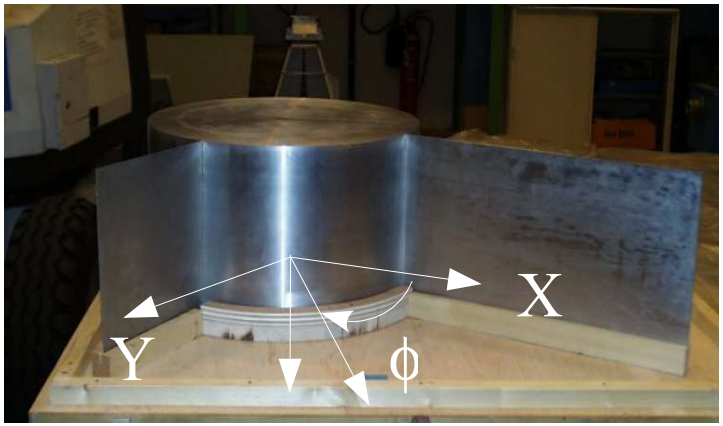
2D Radar Image computing (2/3)

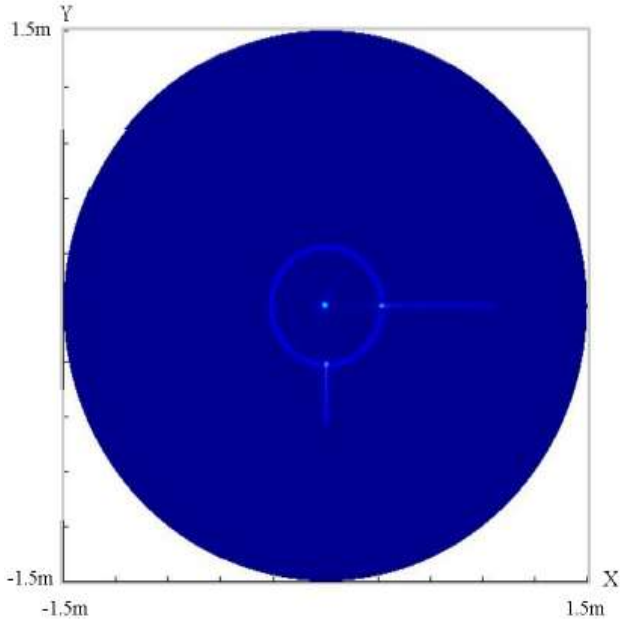




$$\alpha_{\pi\nu}(\vec{r}) = \iint H_{\pi\nu}(\vec{\sigma})\, e^{i\,2\pi\,\vec{r}\cdot\vec{\sigma}}\, d\,\vec{\sigma}^{\,2}$$

2D Radar Image computing (3/3)





Angular fourier coefficients

$$\alpha(r,\phi)=\sum_{m=-\infty}^{\infty}\alpha_m(r)e^{im\phi}$$

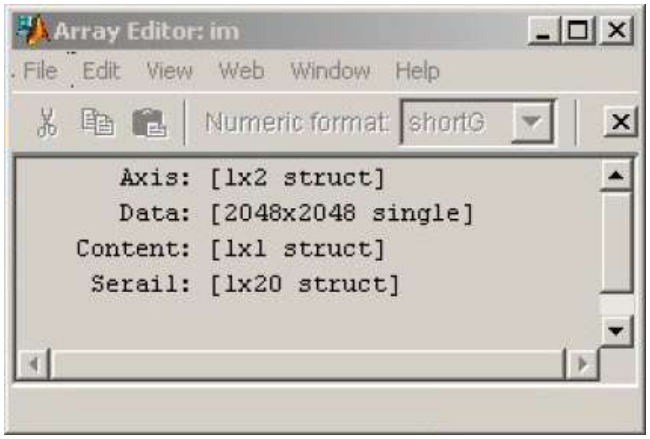
$$H(\sigma,\phi)=\sum_{m=-\infty}^{\infty}H_m(\sigma)e^{im\phi}$$

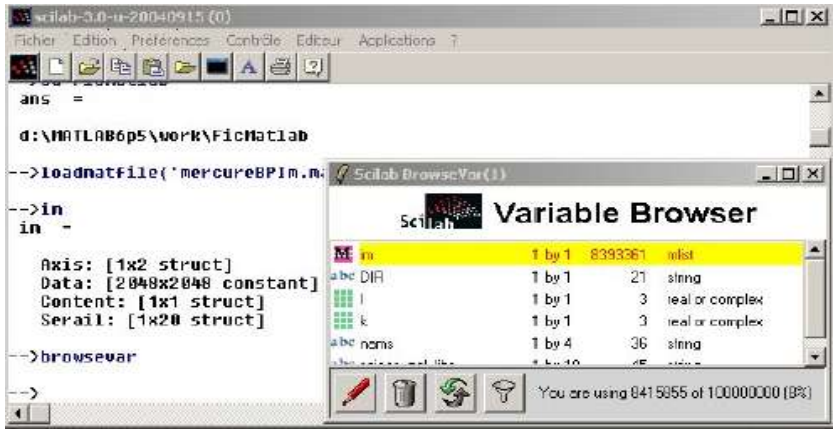
Fourier-Bessel transform

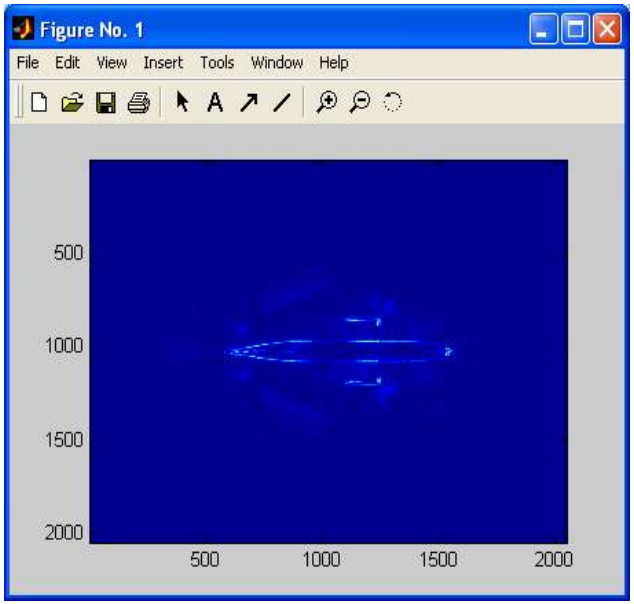
$$\alpha_m(r)=2\pi i^m\int_{\sigma=0}^{\sigma_{max}}\sigma H_m(\sigma)J_m(2\pi\sigma r)d\sigma$$

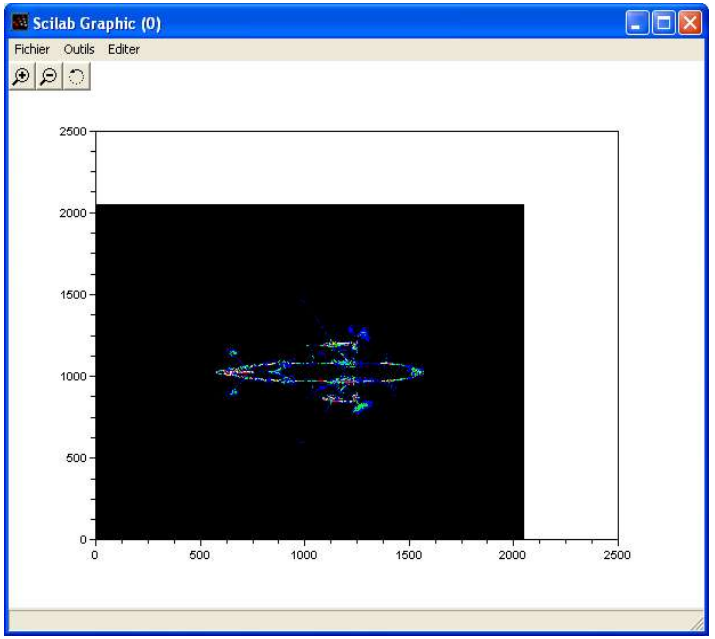
$$H_m(\sigma)=2\pi i^{-m}\int_{r=0}^{\infty}r\alpha_m(r)J_m(2\pi\sigma r)dr$$

Assessment: loadmatfile

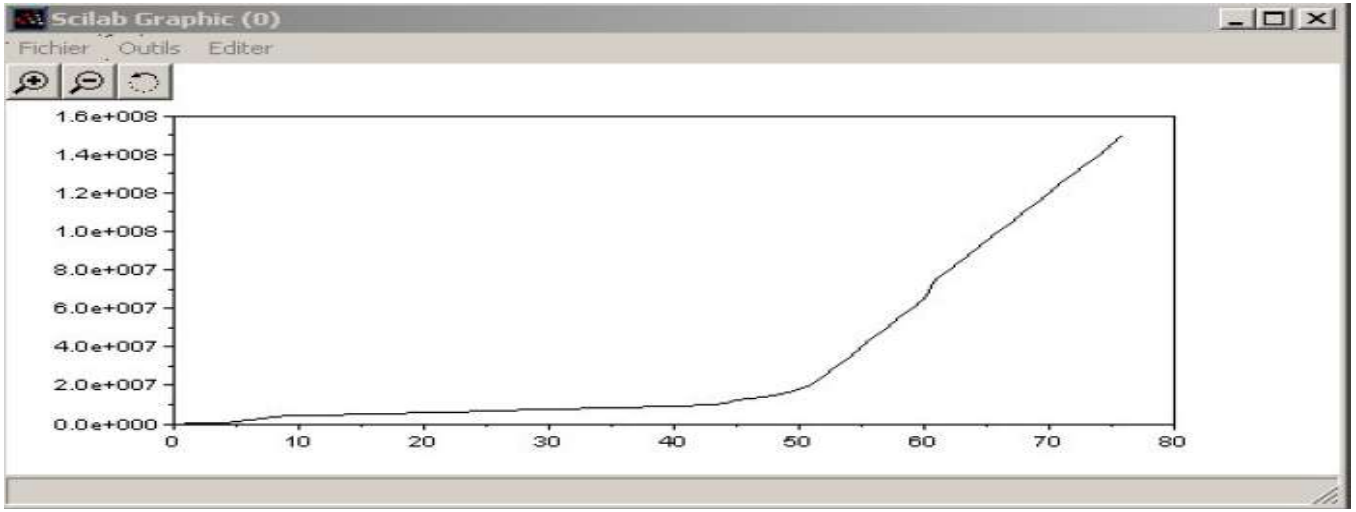


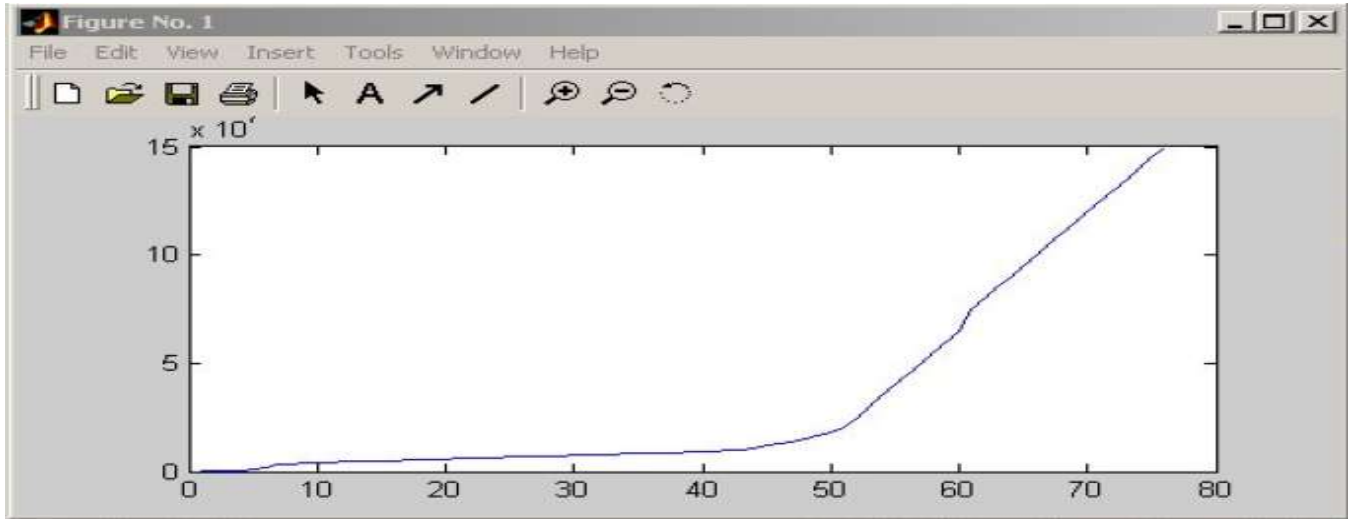




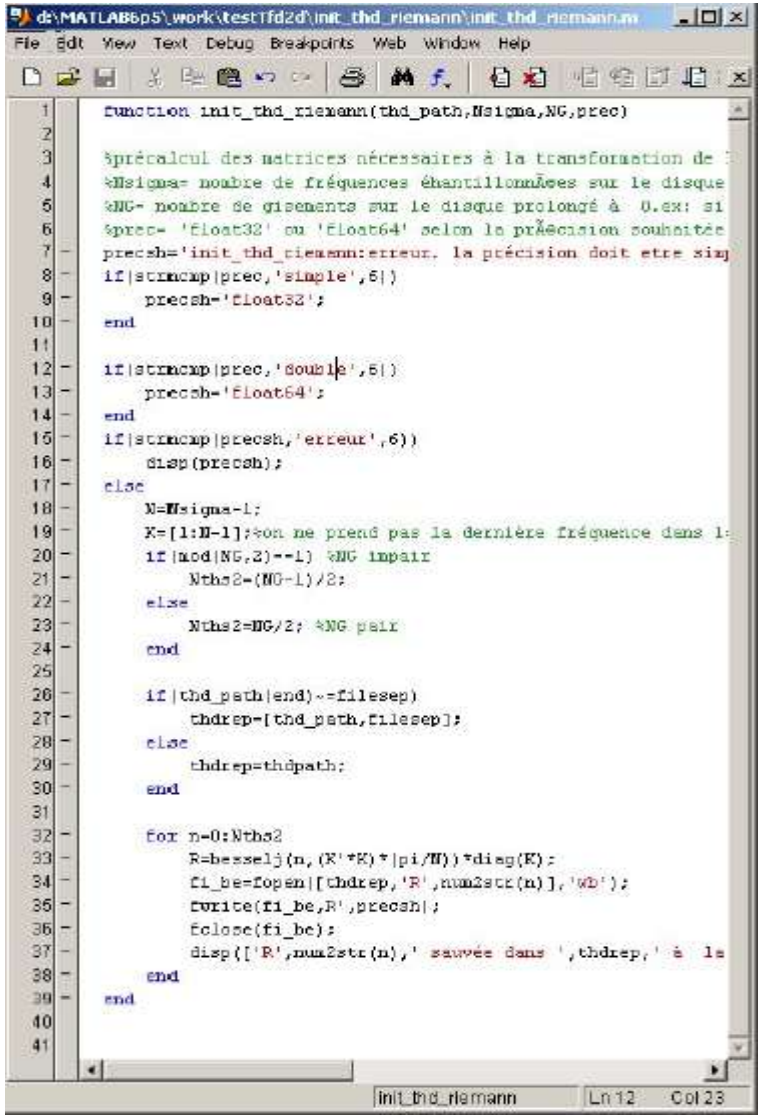


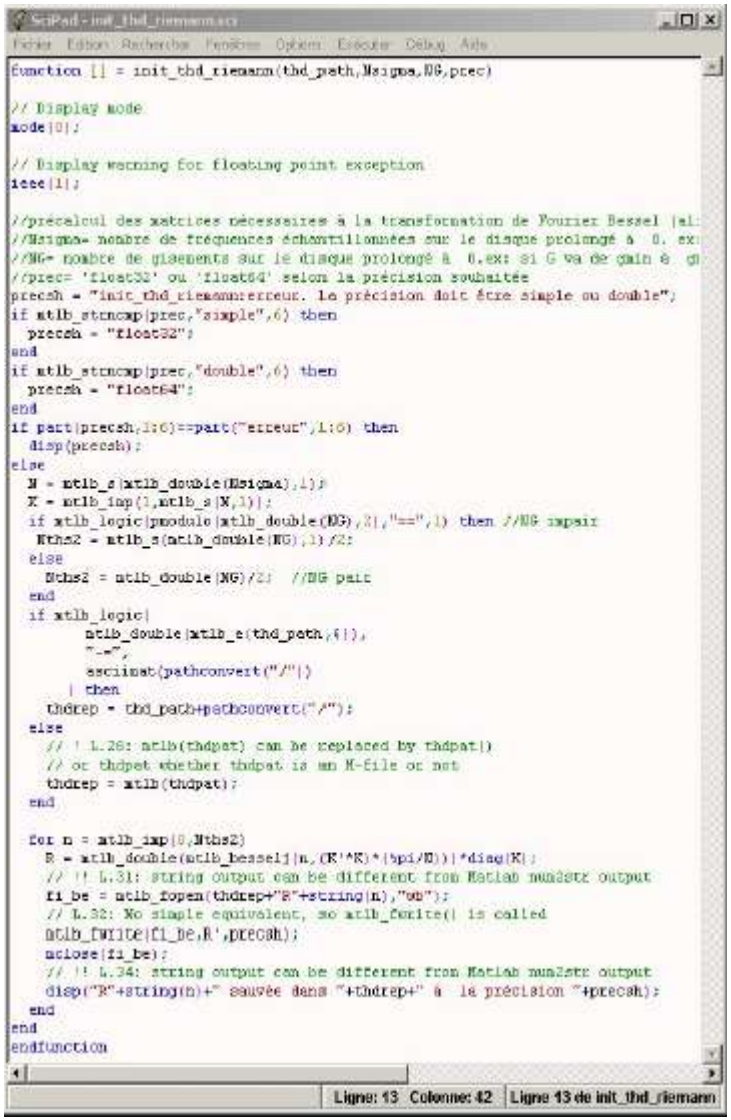
Assessment: savematfile





Assessment: translatepaths





Conclusion (1 / 2)

- Developments present in the 23.11.2004 version only
- Load and Save of matlab files operational
- Translation of matlab script in an advanced stage
- For the time being, no support of the reverse translation

Conclusion (2 / 2)

- ***savematfile/loadmatfile***
  - To be verified on objects
  - To be verified (or remade) on matlab 7.0 files
- ***Translatepaths***
  - To be completed by new emulation functions
  - To be studied for the scilab/matlab translation

DEDICATED WORKSHOPS ?