

FEKO Lua Script

ISAR Image Processing

This plugin computes and displays an inverse synthetic aperture radar (ISAR) image from backscattered radar cross section (RCS) data.

Overview

The script computes and displays an inverse synthetic aperture radar (ISAR) image from backscattered radar cross section (RCS) data over frequency and angle. Windowing functions, resampling and extracting the local maxima positions to file are supported. The script requires a far field RCS request or far field RCS data imported from a file containing far field data (a .ffe file). Data imported from a .ffe file must be added manually to a 3D view.

The plugin comes in two primary parts:

1. *PF_ISAR.lua*: The main POSTFEKO script to calculate and display ISAR results.
2. *CF_ISAR_Resolution.lua*: A CADFEKO script that helps to set up the frequency and angular sampling for the required ISAR image resolution and range.

Preconditions

Backscattered RCS data must be in the UV plane of the local coordinate system or in the XY plane of the global coordinate system. Equivalently, the backscattered RCS should be requested in the $\theta' = 90$ cut. Sub-ranges of the total angular phi range could be used to change the viewing angle.

It is assumed that the image will be centred at the global origin. Thus the down-range (X) and cross-range (Y) should be sufficient to include all the structures or targets relative to the global origin.

The extent and resolution of the ISAR image depend on the frequency range, number of frequencies, angular range and number of incident angles.

Example 1

Introduction

In this example the view direction of the ISAR image is changed using the local workplane of the incident plane wave excitation. It is also possible to change the view direction and cross-range resolution of the ISAR image by using a subset of the angular data range (see Example 2 below).

The model for this example, *Three_Spheres_VPol_MoM.cfx*, is available for download.

Setting up the model in CADFEKO

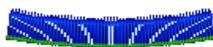
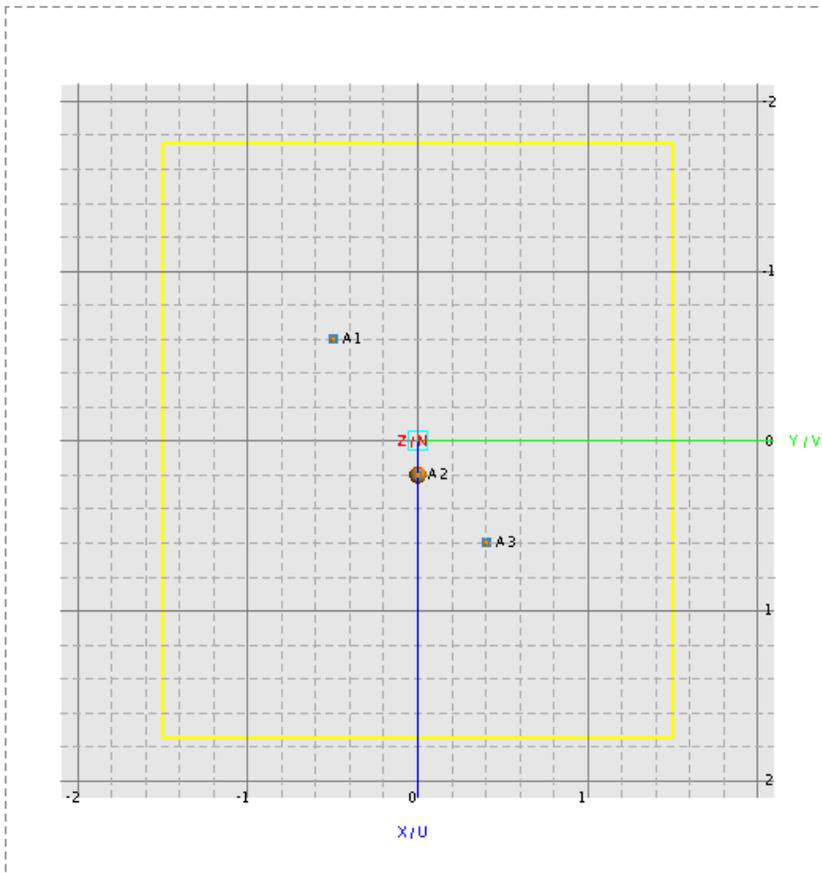
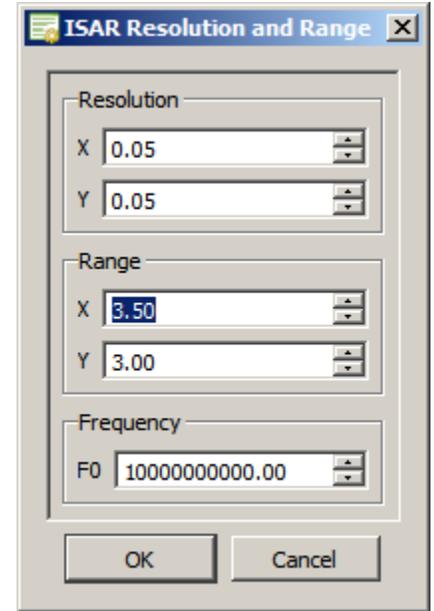
Three spheres with different radii are positioned at points A1, A2 and A3. The model is solved with a full method of moments (MoM) solution to obtain the backscattered RCS.

The CADFEKO *CF_ISAR_Resolution.lua* script is used to calculate the required frequency and angular ranges and associated sampling to create a 3.5 m x 3 m (down-range x cross-range) ISAR image with 0.05 m resolution in both down-range (X) and cross-range (Y). The script calculates the required start frequency ($f1$), end frequency ($f2$) and the number of frequency samples (nf) for a given centre frequency ($f0$). It also calculates the required start angle ($\phi1$), end angle ($\phi2$) and angular increment ($d\phi$). The script assumes that the polar to Cartesian interpolation will be applied to the RCS data.

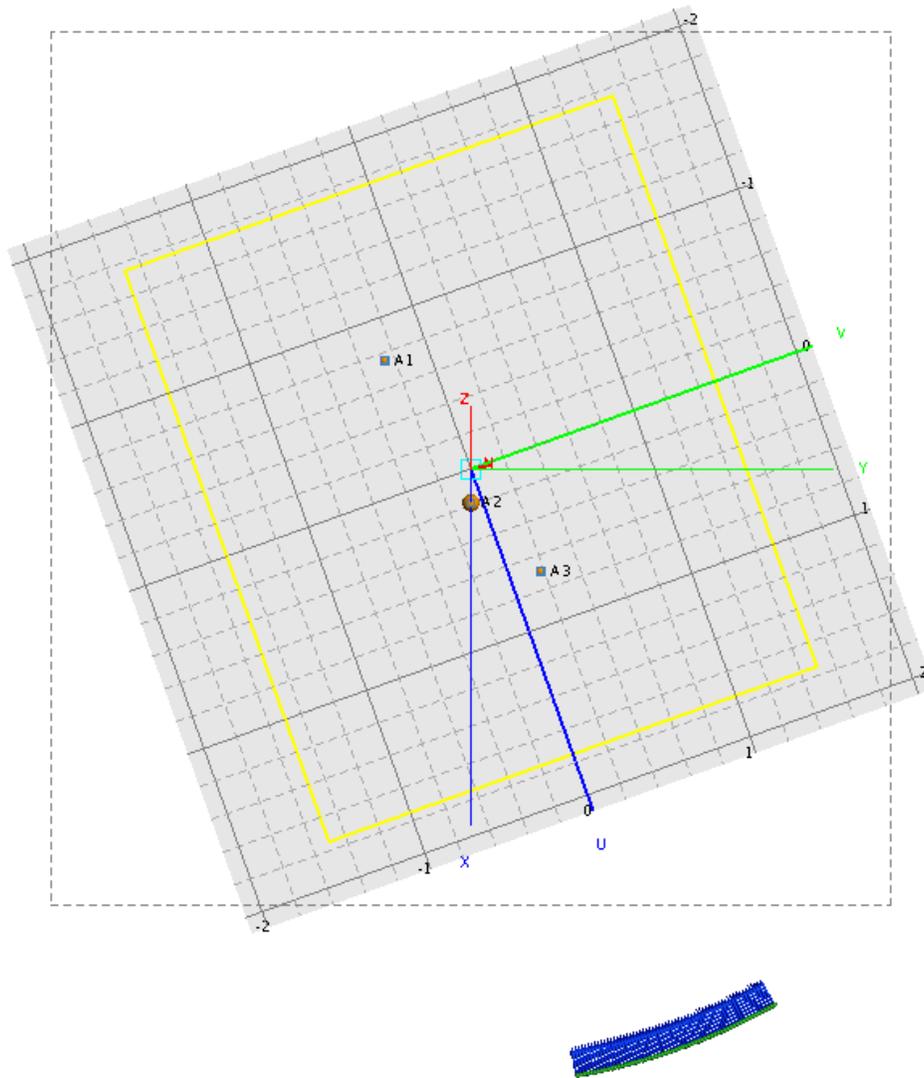
The XY_plane solution configuration uses the above-mentioned angular and frequency variables to create a plane wave excitation (XY_v0) in the global coordinate system with the ISAR view direction along the X axis.

The UV_plane solution configuration uses the same angular and frequency variables in its plane wave excitation (UV_v0), but the excitation is defined using a local workplane. The workplane is obtained from the XY workplane, rotating it by 20° around the local N axis and then by 10° around the local V axis.

Each configuration contains a far field request set to calculate fields in the plane wave incident direction.



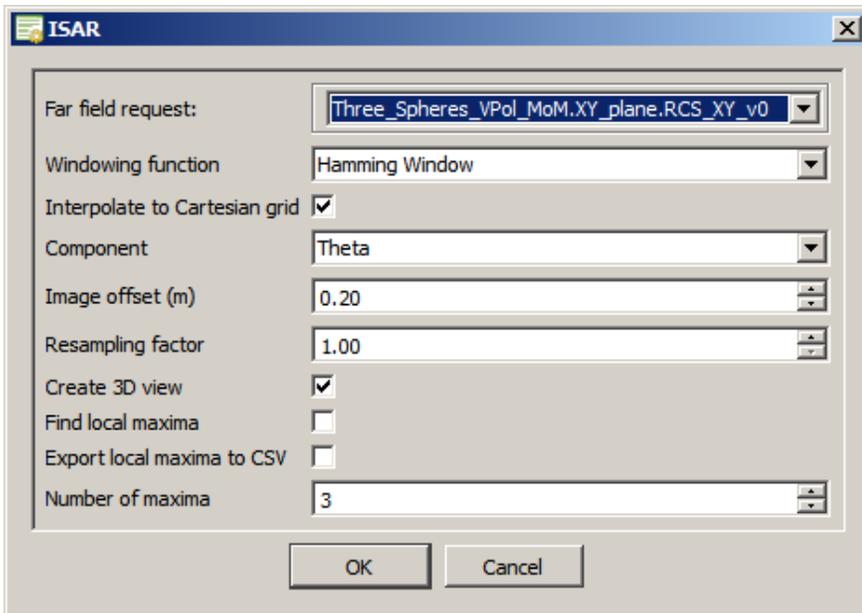
Top view of the XY_plane solution configuration showing the positions of the three spheres with the plane wave excitation. The yellow frame dimensions are 3.5 m x 3 m



Normal view of the UV_plane solution configuration showing the positions of the three spheres with the plane wave excitation. The yellow frame dimensions are 3.5 m x 3 m

Post-processing results in POSTFEKO

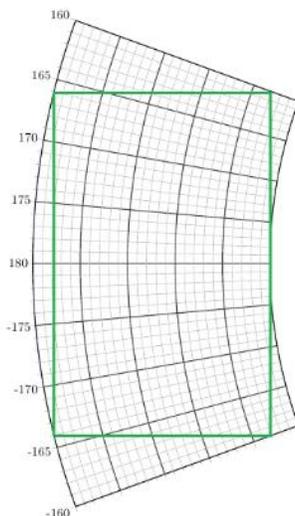
The *PF_ISAR.lua* script uses an input form to select the following input parameters:



The screenshot shows the ISAR dialog box with the following settings:

- Far field request: Three_Spheres_VPol_MoM.XY_plane.RCS_XY_v0
- Windowing function: Hamming Window
- Interpolate to Cartesian grid:
- Component: Theta
- Image offset (m): 0.20
- Resampling factor: 1.00
- Create 3D view:
- Find local maxima:
- Export local maxima to CSV:
- Number of maxima: 3

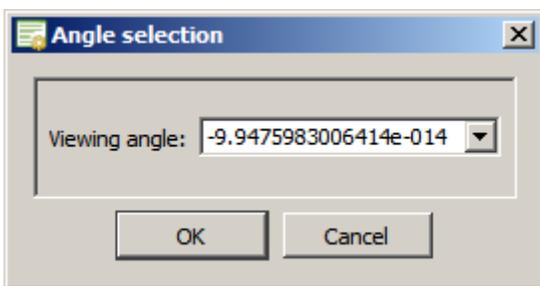
- **Far field request:** An RCS far field request.
- **Windowing function**
The following windows functions are supported:
 -  Hamming (default)
 -  Dirichlet
 -  Bartlett
 -  Blackman
- **Interpolate to Cartesian grid** (Default): The 2D FFT requires that the X and Y distances should be uniformly spaced. The basic ISAR implementation assumes that the angle range is small enough such that the polar sampling approximates a Cartesian grid. The small angle range assumption can be relaxed by transforming the polar grid to Cartesian. This reduces smearing and shift of the scattering points and increasing accuracy of the resulting ISAR image.



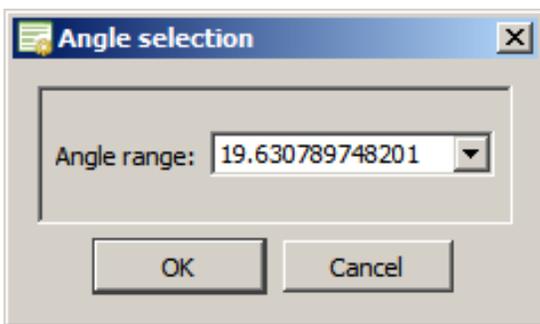
Polar to Cartesian transformation

- **Component:** Specify which E-field component to use to generate the ISAR result. Supported options are:
 - Theta (Default)
 - Phi
 - LHC (Left Hand Circular)
 - RHC (Right Hand Circular)
- **Image offset** (Default: 0.2 m): Offset distance of the ISAR image above the plane wave workplane.
- **Resample factor** (Default: 1): Factor used to up sample the image resolution to create a smoother image with more points. Note it does not change the underlying image resolution based on the frequency and angular information.
- **Create 3D view** (Default): Generates a normal view of the partially transparent ISAR image. By default the RCS data is displayed in a dB scale.
- **Find local maxima:** Calculates the first N local maxima in descending order in the ISAR image.
- **Export local maxima to CSV:** Export the N local maxima to a CSV file. The "Find local maxima" option must be enabled for this to work.
- **Number of local maxima** (Default: 3): The number of local maxima to find.

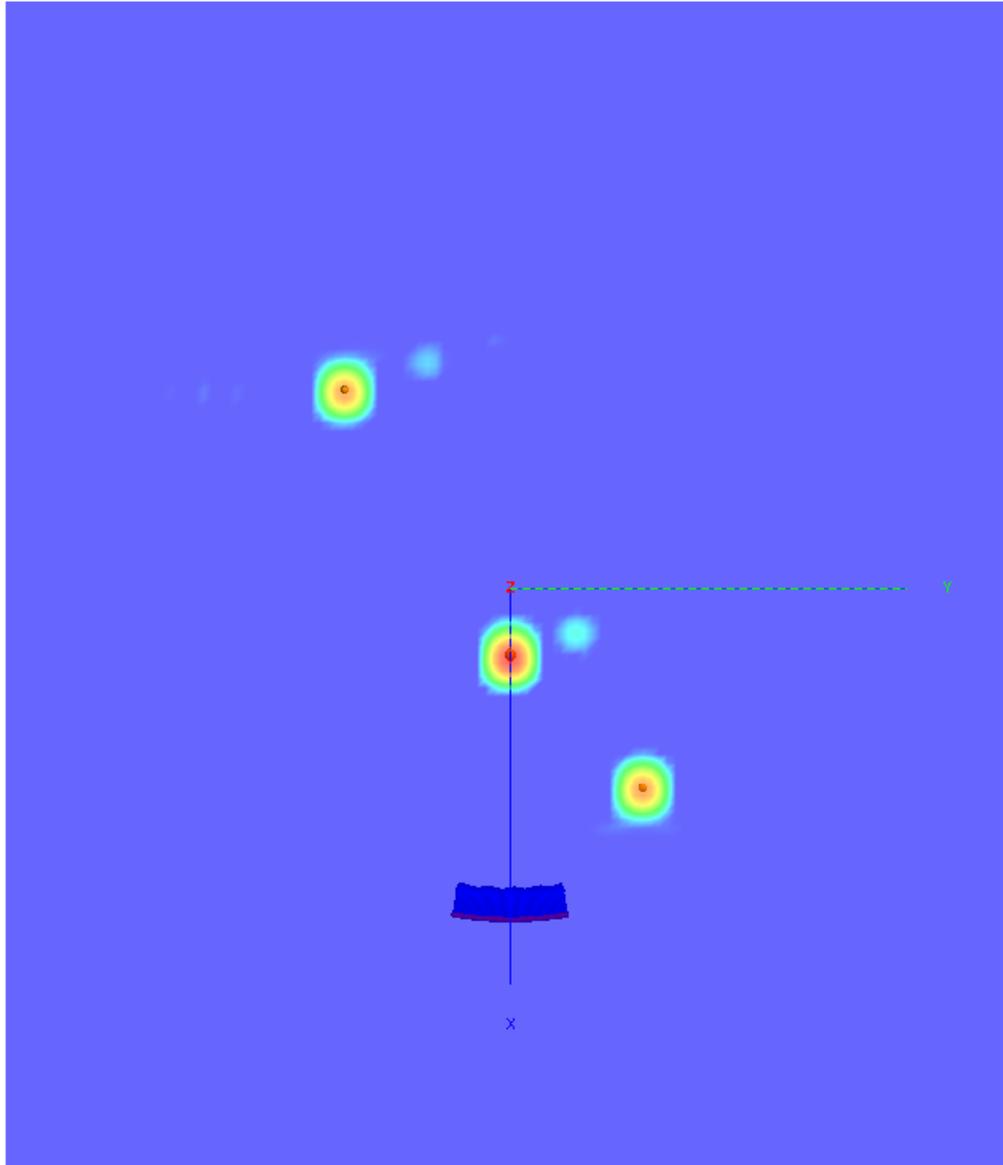
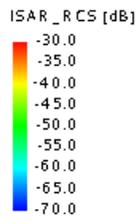
After accepting the input from the main form, the script will process the RCS data and display a dialog asking for the viewing angle. The default viewing angle is the average of the start angle and end angle. The user can select a different viewing angle, but should take note that this setting could reduce the maximum angular data range.



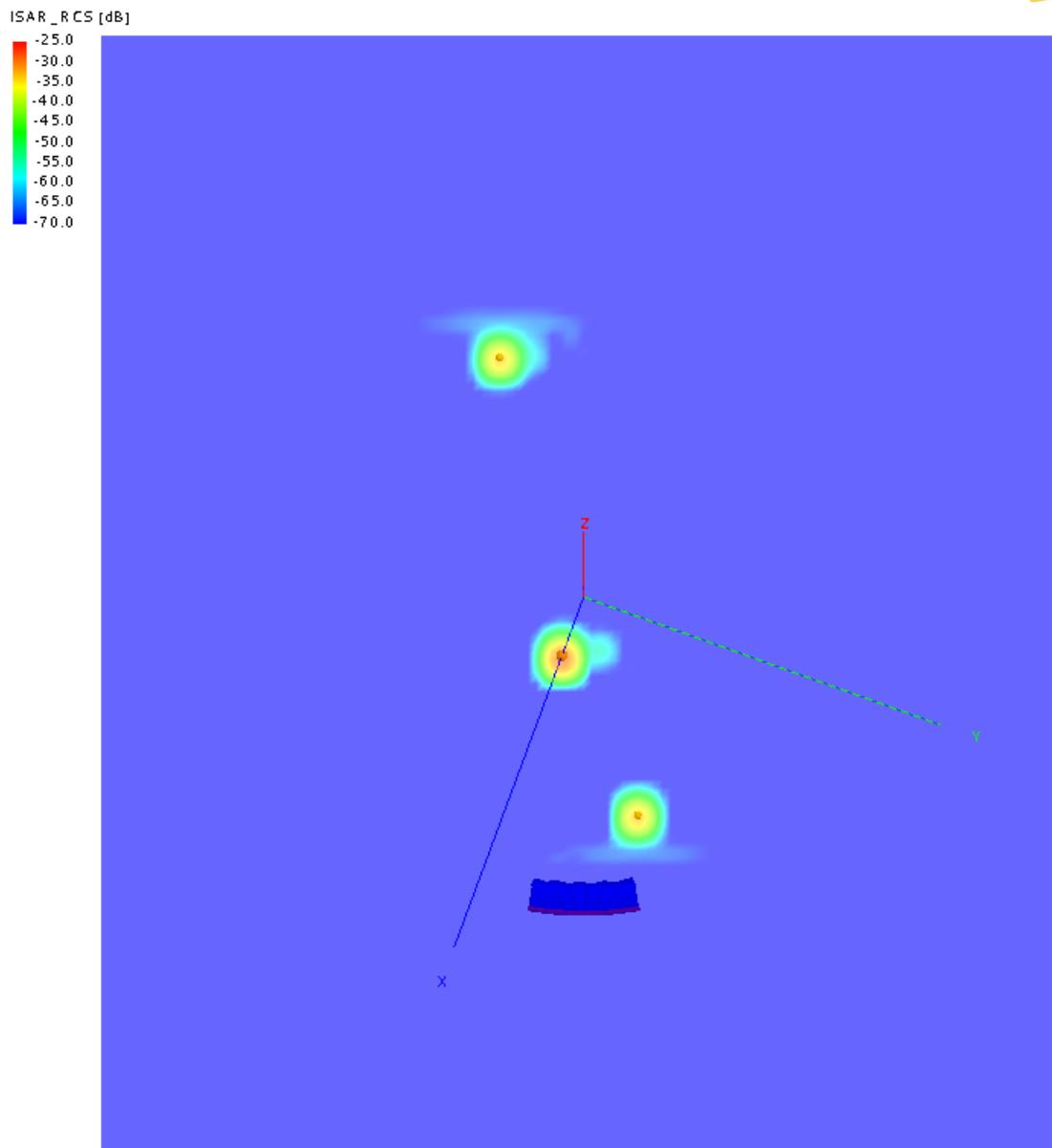
With the viewing angle specified, the script will calculate the maximum angular range that can be used. The user can then select an angular range from the list. Note that the angular range affects the cross-range resolution.



The ISAR images associated with the XY_plane and UV_plane solution configurations are shown below. A resampling factor of 2 was used with default values for the rest of the input parameters.

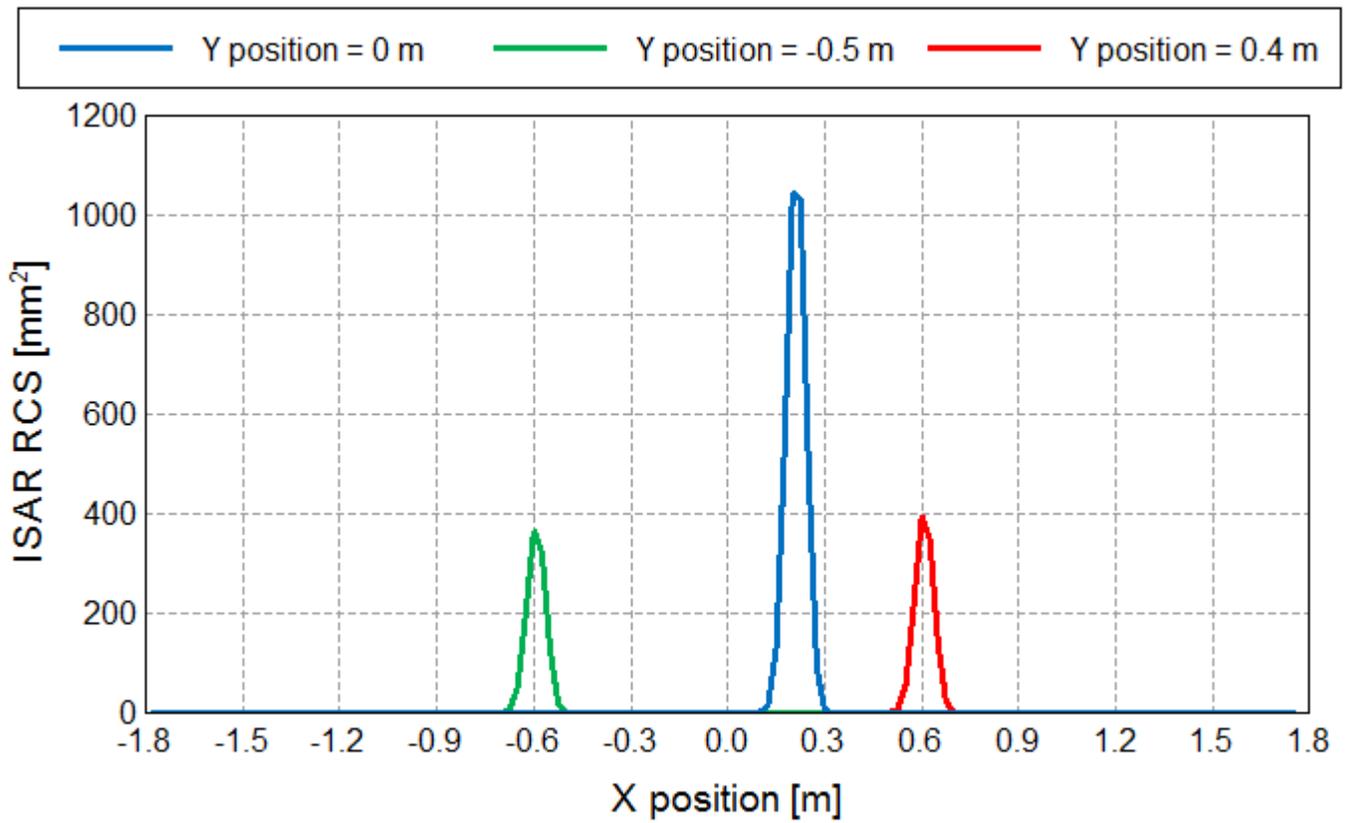


ISAR image (3.5 m x 3 m) obtained from the XY_plane solution configuration



ISAR image (3.5 m x 3 m) obtained from the UV_plane solution configuration

The ISAR dataset can also be used to generate 2D graphs, such as the image below showing the RCS versus X position corresponding to each sphere's Y coordinate.



RCS versus X position for the Y coordinates of points A1, A2 and A3

Example 2

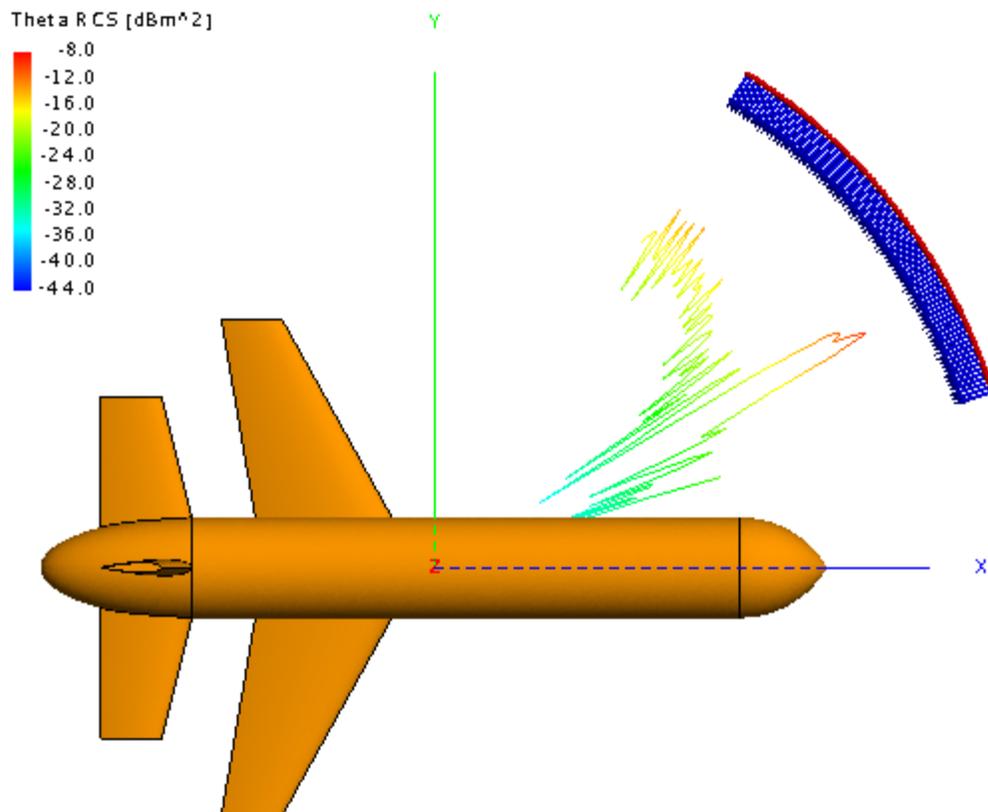
Introduction

In this example the view direction and cross-range resolution of the ISAR image is changed by using a subset of the angular data range. Some of the script's post-processing options are demonstrated.

The model for this example, *Missile_RCS_Vpol_PO.cfx*, is available for download.

CADFEKO model

The physical optics (PO) high frequency approximation is used to obtain the backscattered RCS. The RCS data is computed from $\phi_1 = 18^\circ$ to $\phi_2 = 57^\circ$ with $(1/3)^\circ$ increments and over a frequency range from 8 GHz to 12 GHz with 81 samples. Assuming a minimum angular range of 24° the view direction angle can be changed between 30° and 45° .

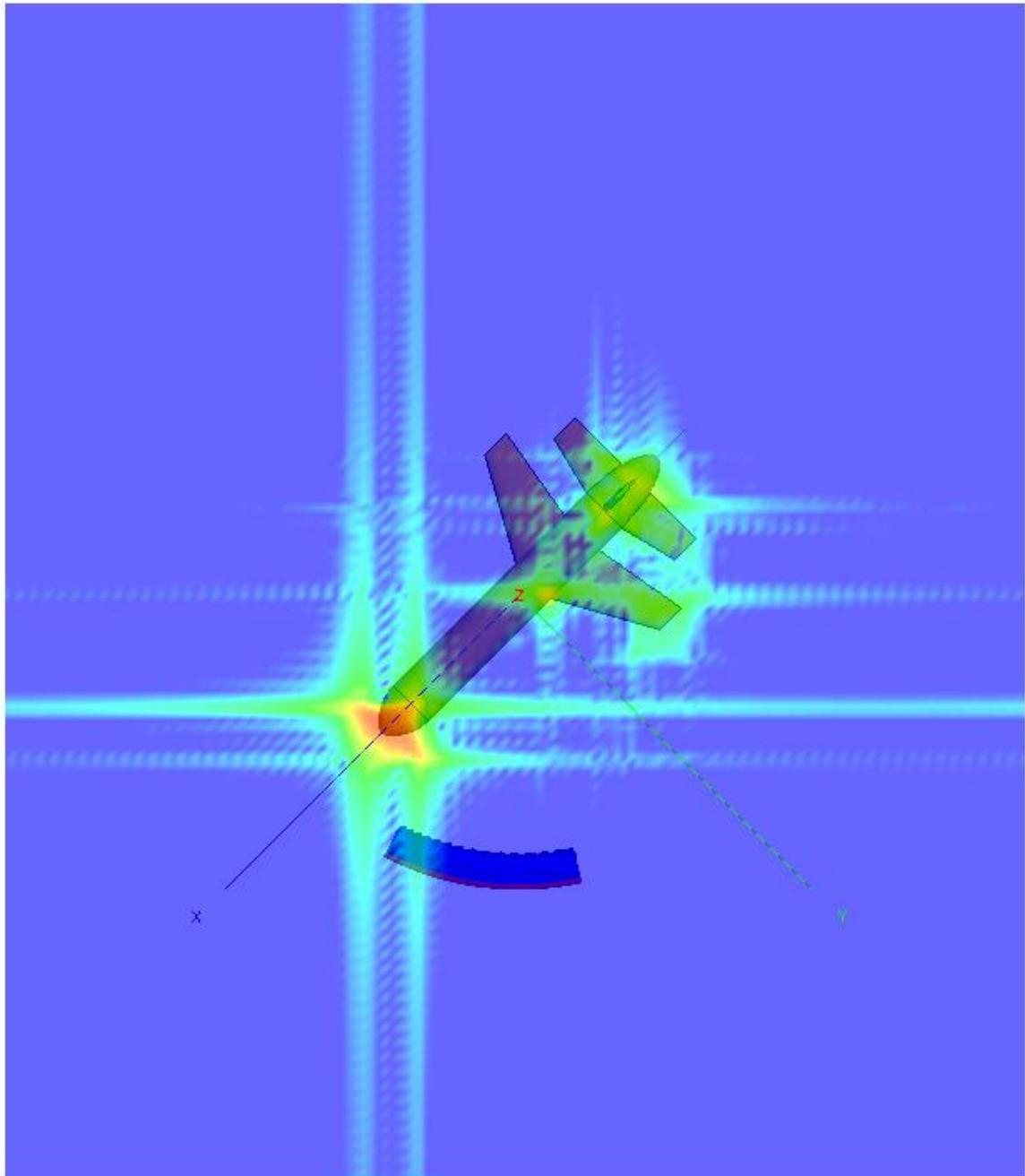
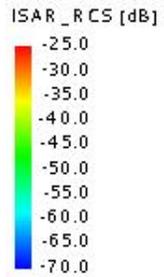


Top view of the *Missile_RCS_Vpol_PO* model

POSTFEKO results

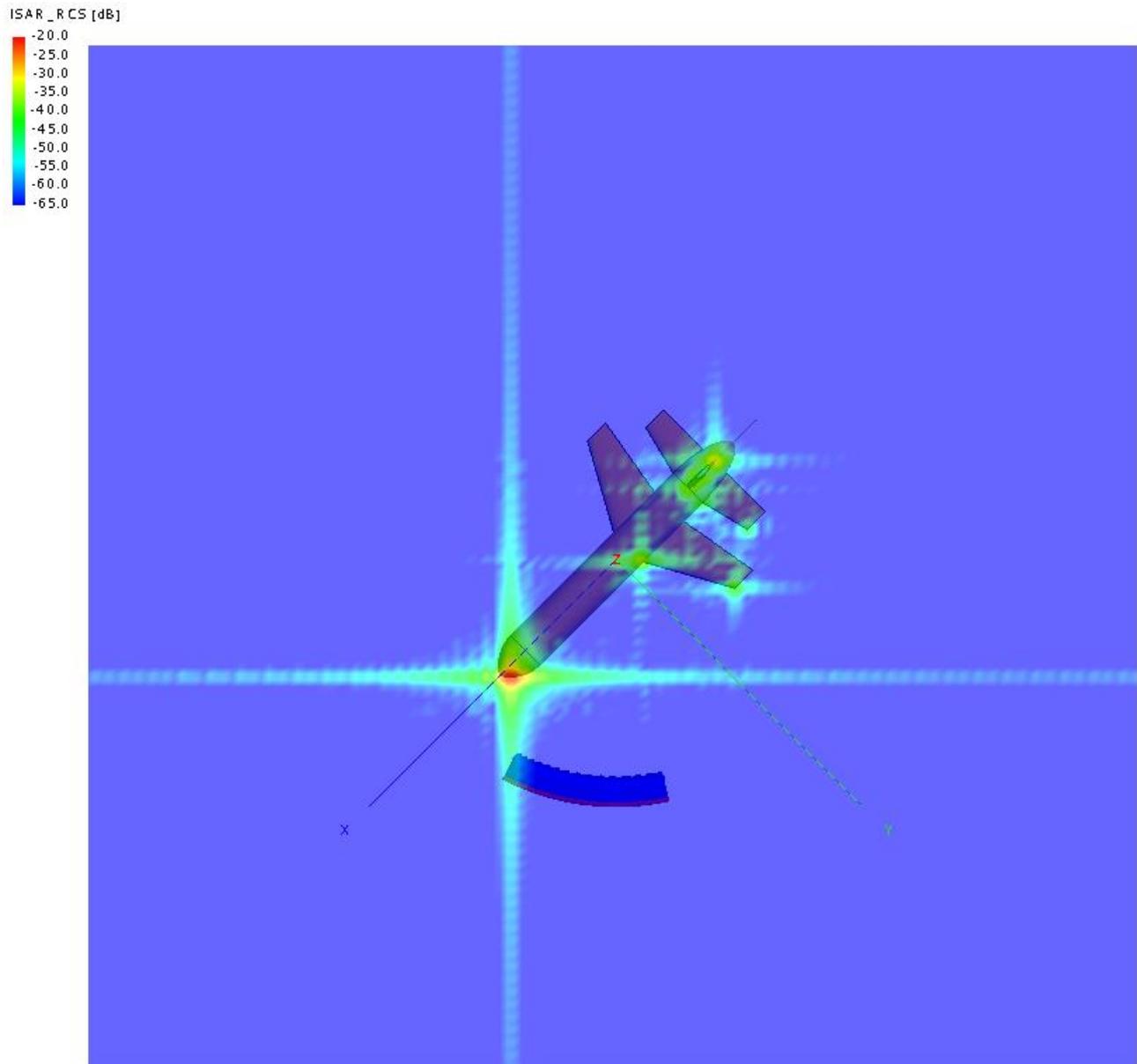
The images below show some results obtained by using different post-processing options available in the script.

The first ISAR image shows the result obtained without any post-processing of the RCS data for a view angle $\phi = 45^\circ$ and angular range of 24° . The down-range and cross-range resolutions are 0.0370 m and 0.0356 m, respectively.



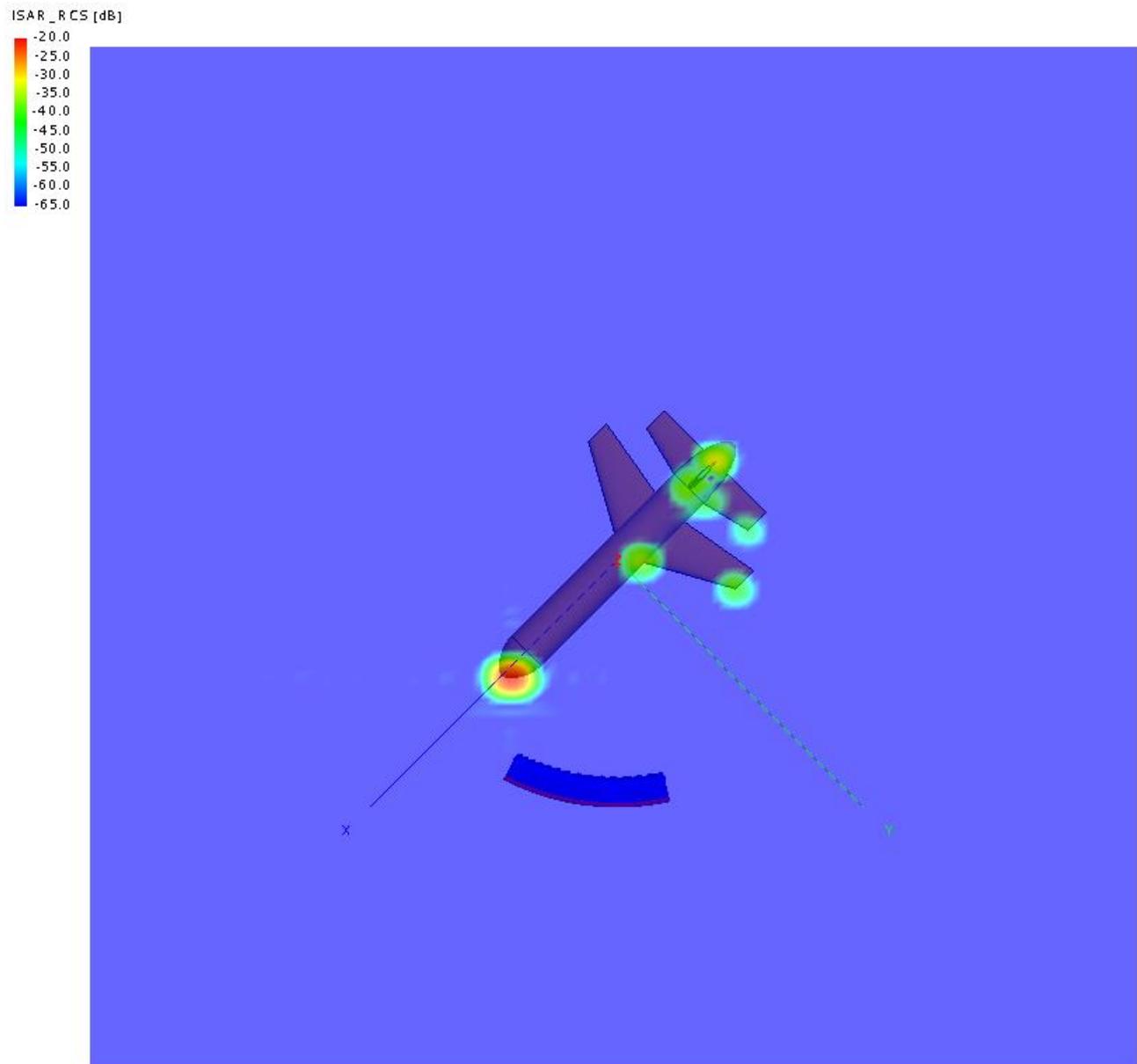
ISAR image with no post-processing of the RCS data

The angular smear away from the image origin can be reduced by transforming the polar data to Cartesian coordinates using bi-linear interpolation. ISAR image with polar to Cartesian transformation for a view angle $\phi_0 = 45^\circ$ and angular range of 24° . The down-range and cross-range resolutions are 0.0382 m and 0.0435 m, respectively.



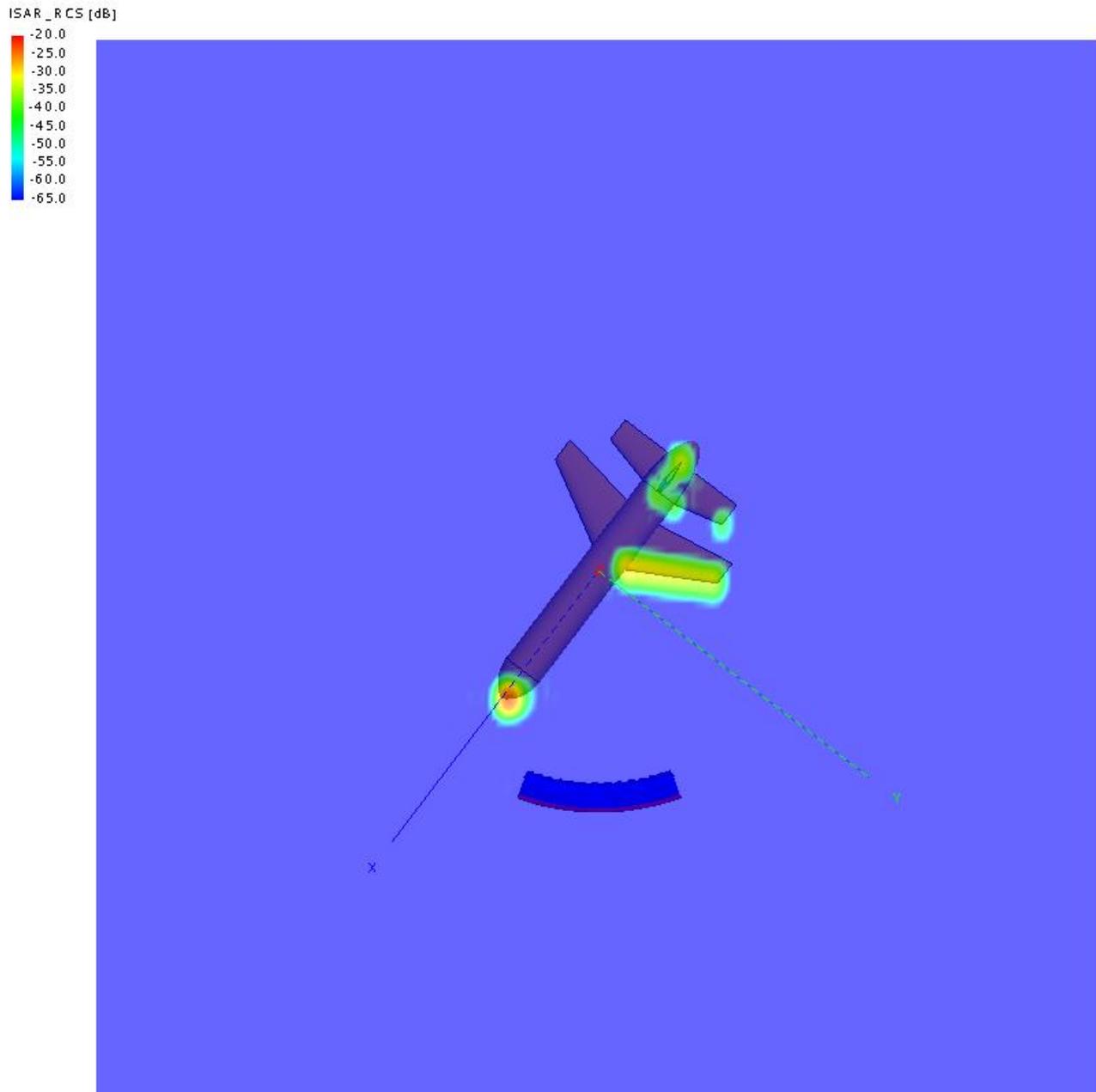
ISAR image with polar to Cartesian transformation

The image can be further processed by windowing the RCS data. The next ISAR image is the result of using polar to Cartesian transformation and a Hamming window for a view angle $\phi_0 = 45^\circ$ and angular range of 24° . The down-range and cross-range resolutions are 0.0382 m and 0.0435 m, respectively.



ISAR image with polar to Cartesian transformation and using a Hamming window

The script can be used to change the view angle and the angular range. The ISAR image below is obtained using polar to Cartesian transformation and a Hamming window for a view angle $\phi_{i0} = 37.33^\circ$ and angular range of 38.67° . The down-range and cross-range resolutions are 0.0408 m and 0.0265 m, respectively.



ISAR image with polar to Cartesian transformation and using a Hamming window with a different view angle

Download Archive Contents

The download archive contains the following files:

Main scripts

- *CF_ISAR_Resolution.lua* - CADFEKO script to help set up frequency and angular sampling for the required ISAR image resolution and range.
- *PF_ISAR.lua* - The main POSTFEKO script to calculate and display ISAR results.

Helper scripts (utilities)

- *Complex_helper.lua* - Functions for complex mathematics.
- *file_helper.lua* - Functions used for naming stored data.
- *FT_helper.lua* - Efficient two-dimensional Fourier transform.
- *Local_maximum_minimum.lua* - Find local maxima.
- *math_helper.lua* - Functions used for windowing.

Examples

- *Example1_Three_Spheres_VPol_MoM.zip*
- *Example2_Missile_RCS_Vpol_PO.zip*