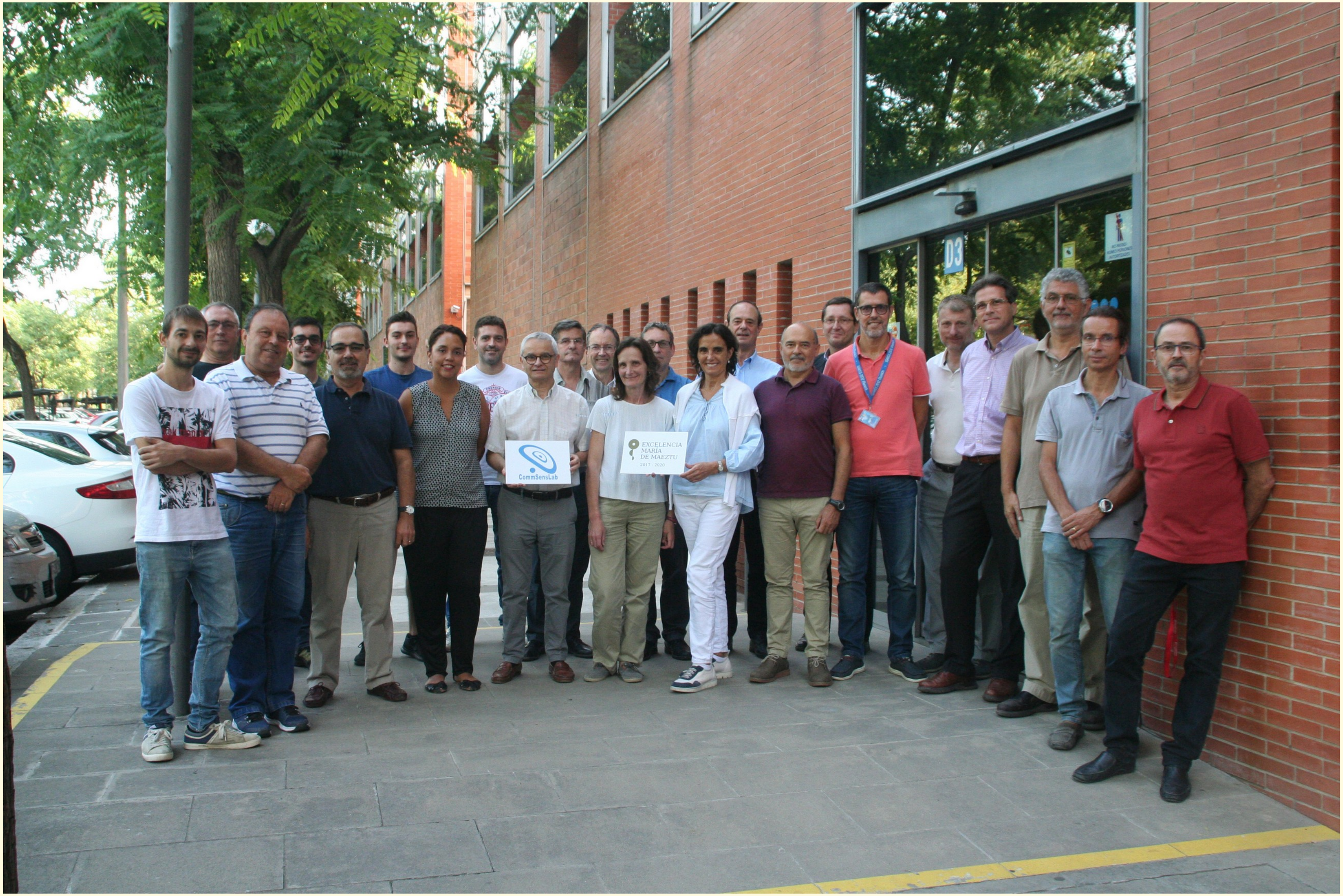
Group activities
+
Randomized low-rank matrix compression

Juan M. Rius*, Hector Lopez-Menchon, Robert Molina,
A. Heldring and E. Ubeda

CommSensLab
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain

The CommSensLab group



The CommSensLab group

https://www.tsc.upc.edu/en/research/research-groups/commsenslab

https://vimeo.com/554429284/ee7ee023f0

- AntennaLab
  - **Computational Electromagnetics**
- Free Space Optical Communications
- Microwave Systems
- NanoSat Lab
- Remote Sensing Lab
  - Active Microwave Remote Sensing Group (SAR)
  - Passive Microwave Remote Sensing Group (Radiometry)
  - Optical Remote Sensing Group

AntennaLab Research Group

STAFF

- 5 Full Professors
- 2 Associate Professors
- 8 PhD Students
- 7 Master Students
- 3 Lab technicians
- 45 PhD dissertations
- More than 150 journal and 200 conference papers published

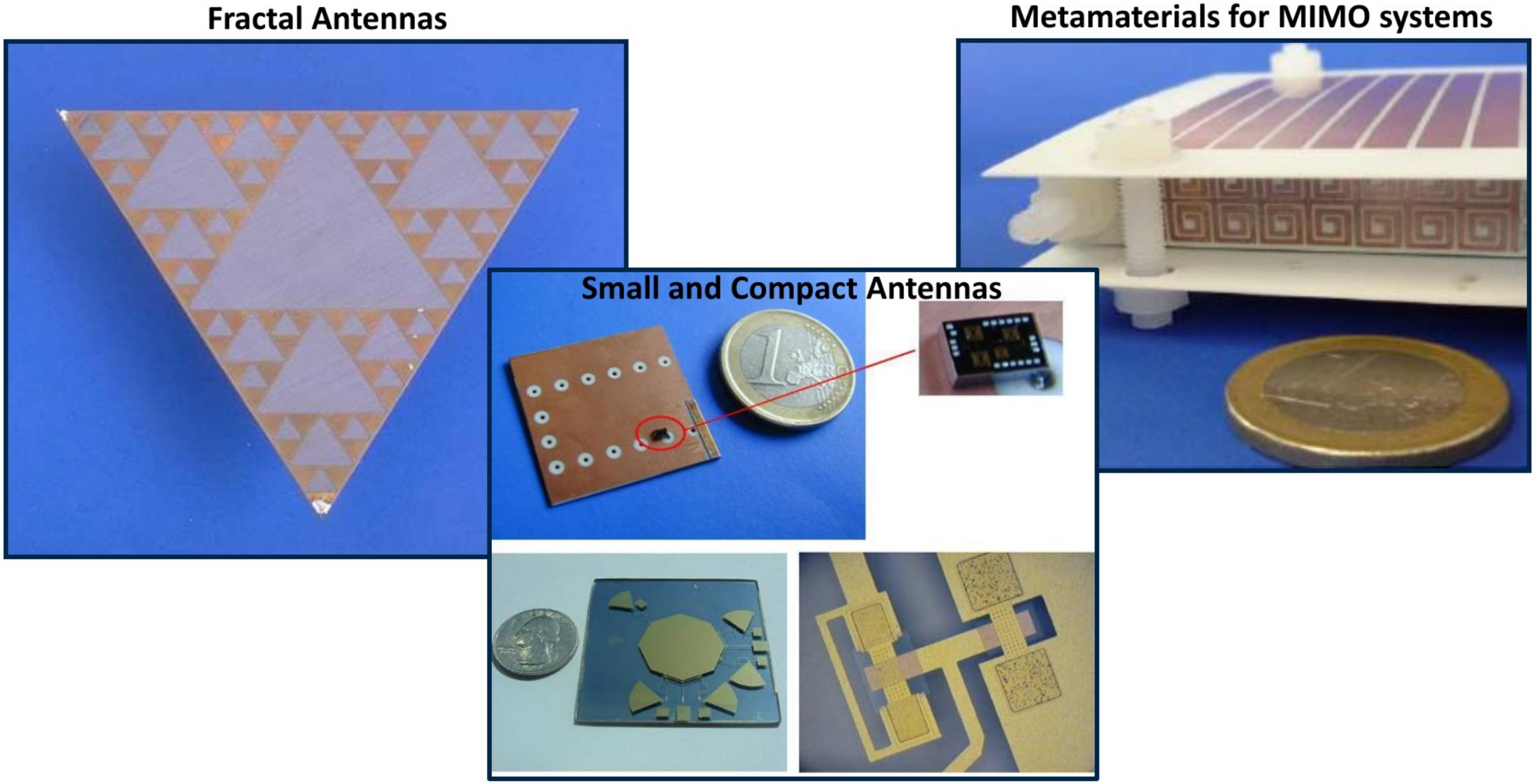
http://www.tsc.upc.edu/antennalab Contact: Jordi Romeu – romeu@tsc.upc.edu

AntennaLab

- Antenna Design
- Antenna Measurements and Diagnostics
- **Computational Electromagnetics**

Antenna Design

The AntennaLab research group has extensive experience in the design, testing and integration of antennas for communication and remote sensing systems.



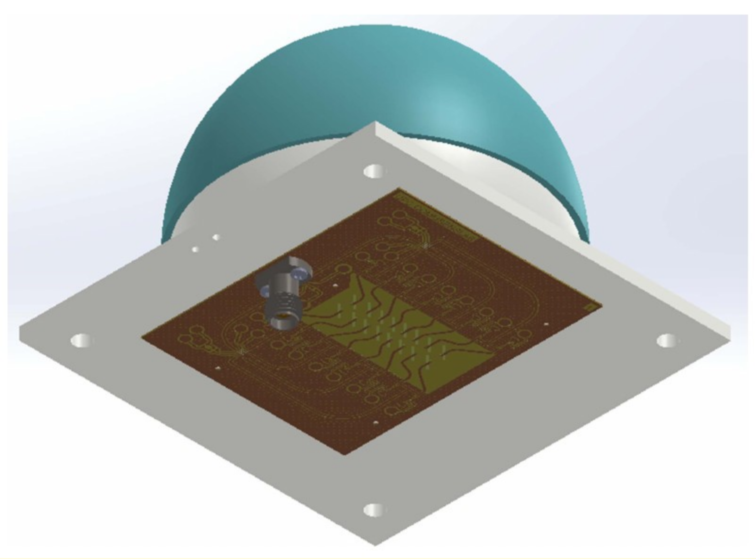
Fractal Antennas

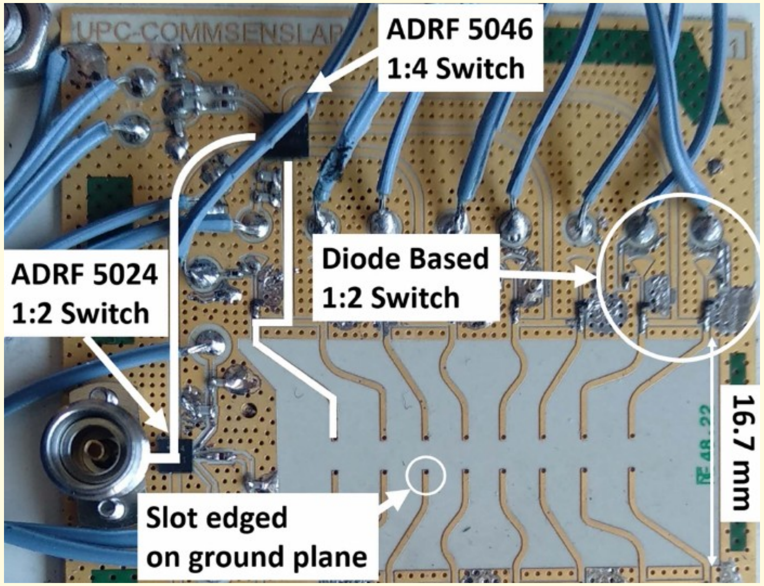
Metamaterials for MIMO systems

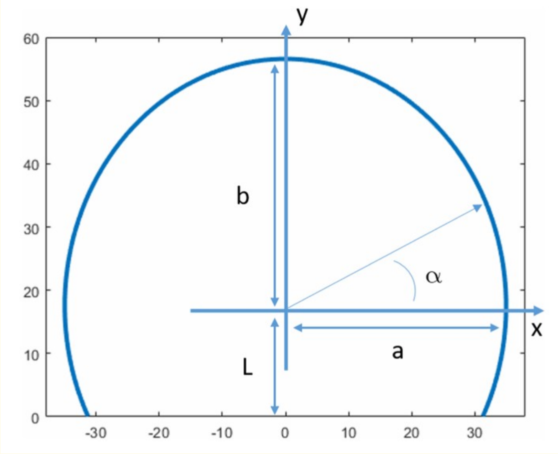
Small and Compact Antennas

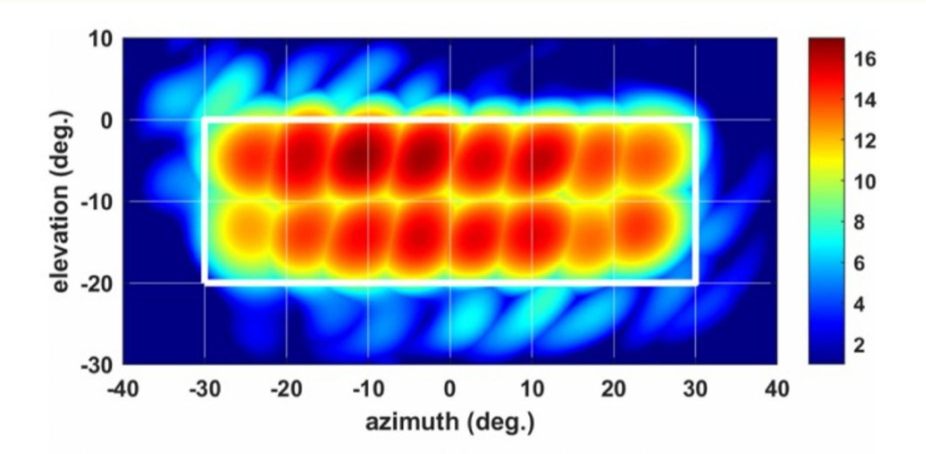
Antenna Design

- Lens-Based Switched-Beam Antenna for a 5G Smart Repeater

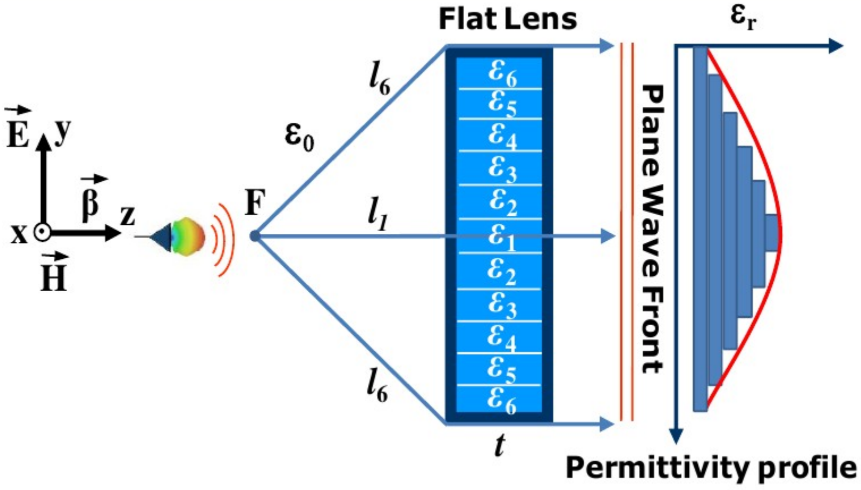


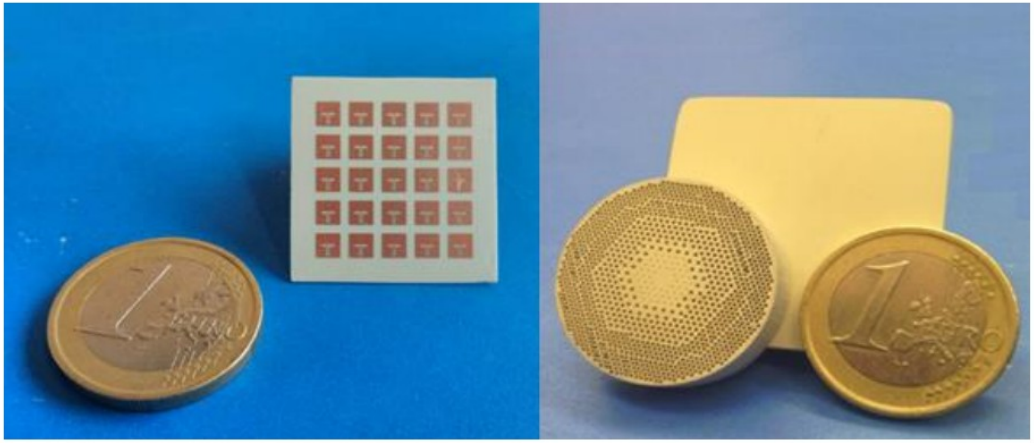




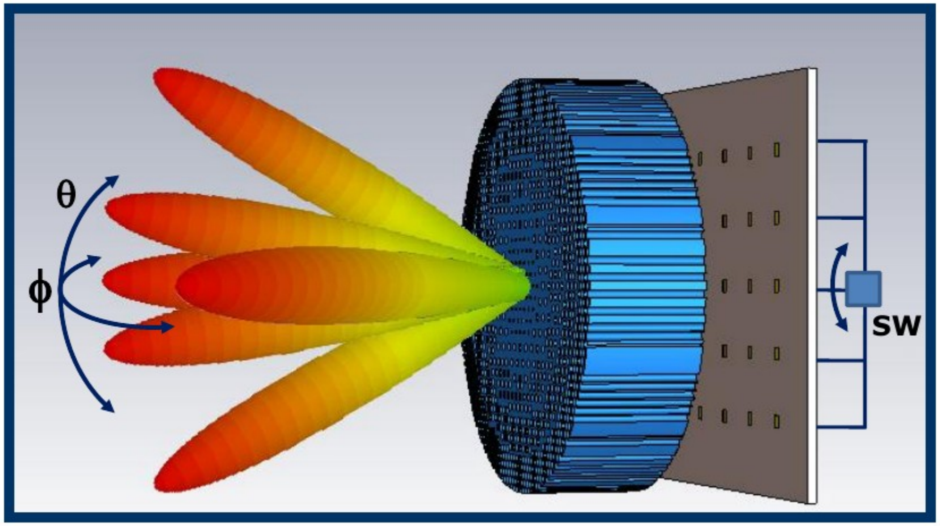


Dielectric Flat Lens Antennas for mmW Applications





Switched-Beam Antenna Array



CAPABILITIES

- HIGH GAIN RADIATION BEAMS (20 dB)
- FULL 2-D BEAM-STEERING CAPABILITIES

APPLICATIONS

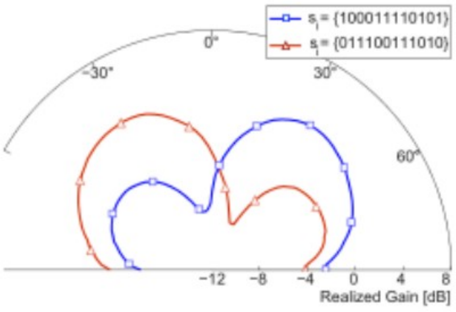
- WPAN communication systems at 60 GHz band.
- Automotive radar systems at 77 GHz
- Passive Imaging Systems at 94 GHz

Reconfigurable Antennas: Pixel Antenna

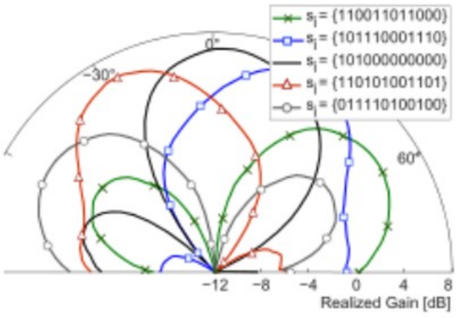
The antenna is reconfigured activating the switches (PIN diodes) interconnecting antenna pixels.

Reconfiguration capabilities are enhanced by using multi-size pixel geometries. Only 12 switches are required.

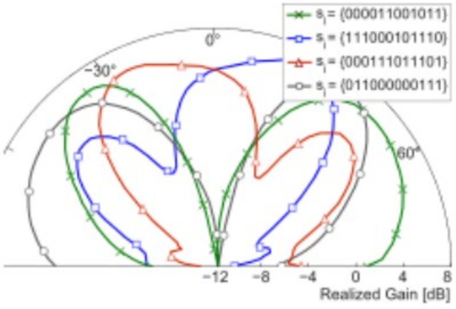
The antenna can adjust configurations to cover from 1.2 GHz till 6.0 GHz.



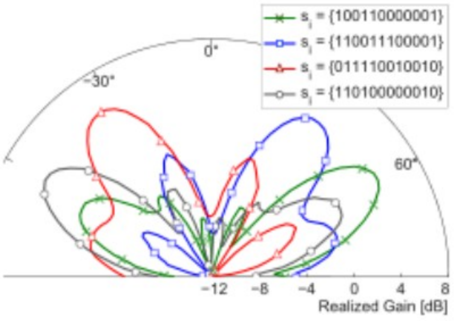
(a) f = 1.5 GHz



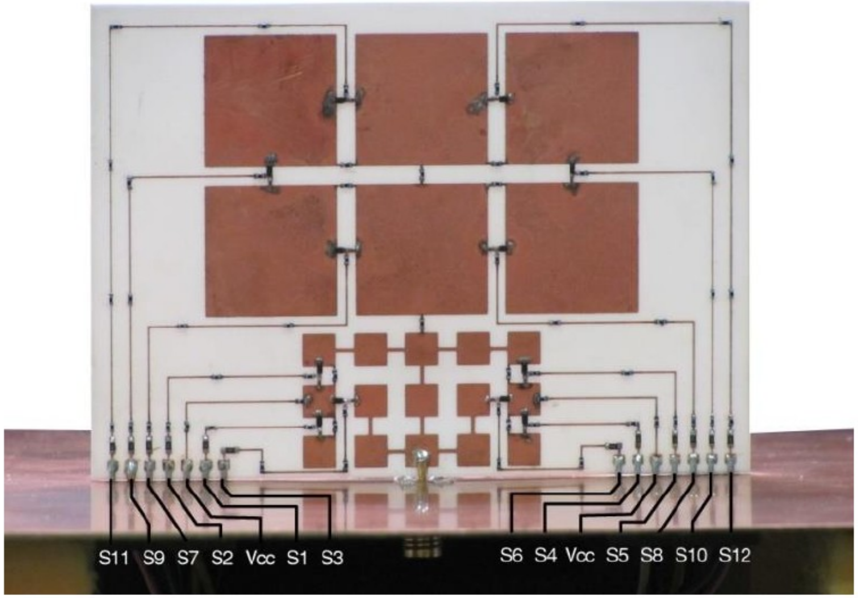
(b) f = 2.5 GHz

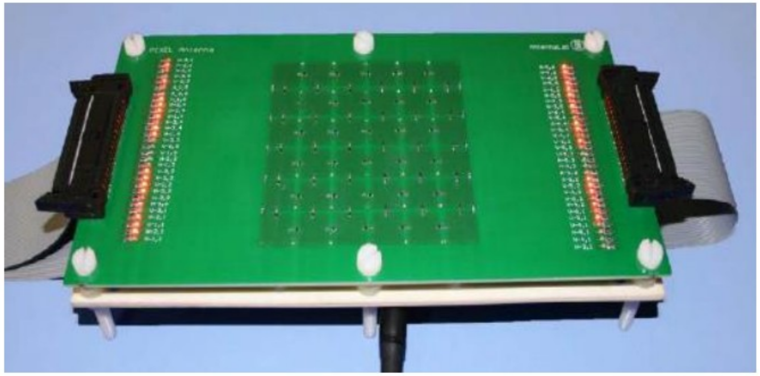


(c) f = 3.5 GHz



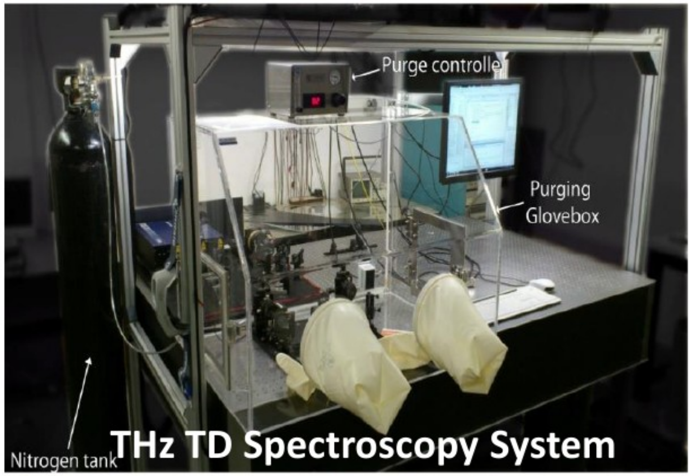
(d) f = 4.5 GHz



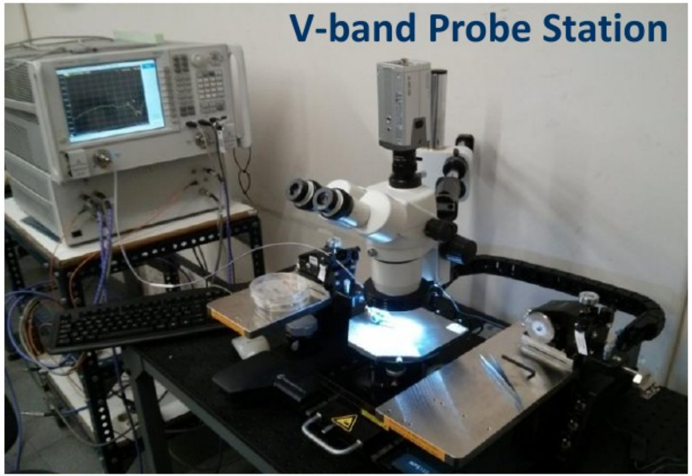


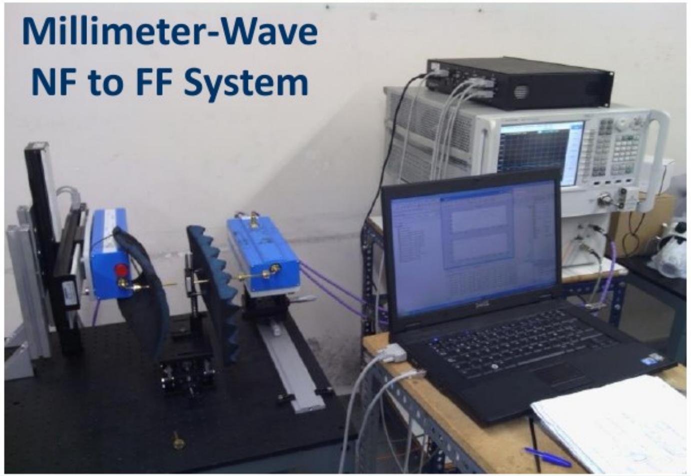
Antenna Measurements

Research facilities include full instrumentation to perform circuital and radiation measurement into the whole microwave band. The availability of its own measurement resources has supported an intensive research activity in the domain of near to far field transformation and antenna diagnostics.

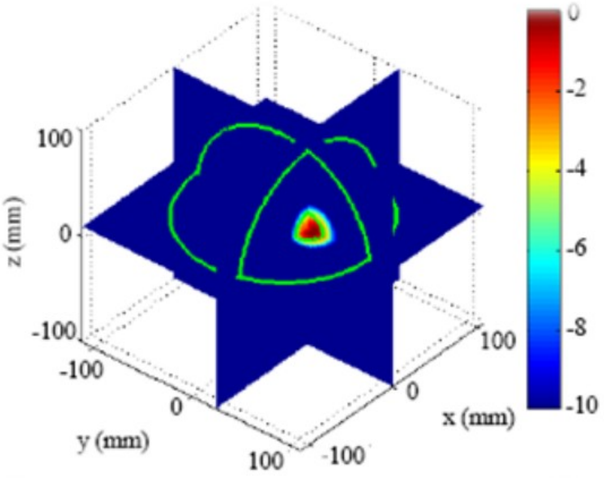

THz TD Spectroscopy System

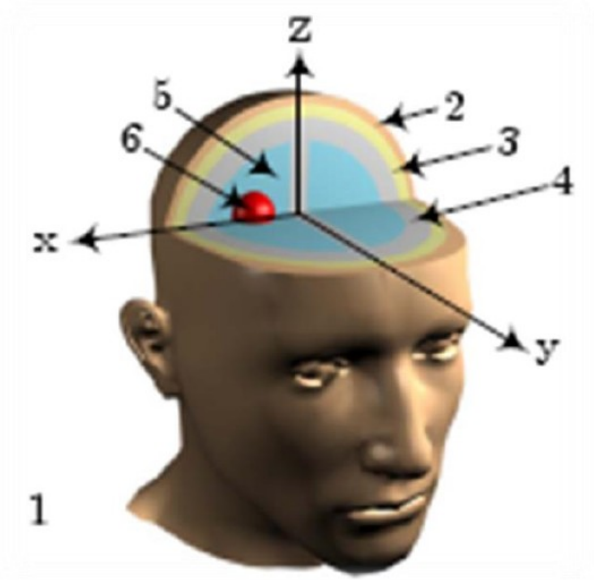

Anechoic Chamber


V-band Probe Station

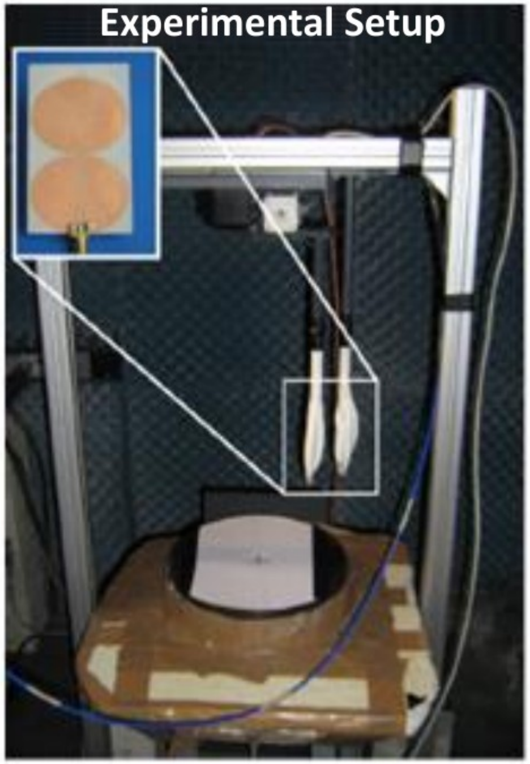

Millimeter-Wave NF to FF System

Microwave Imaging for Medical Applications

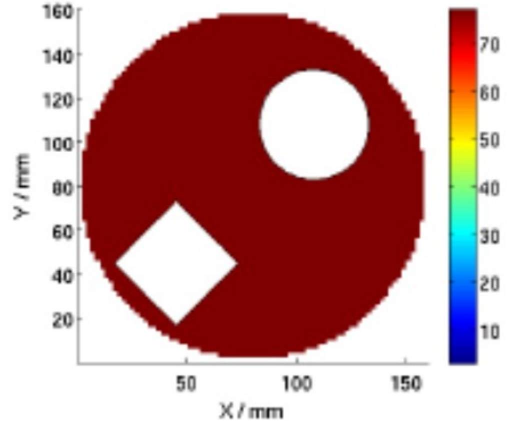




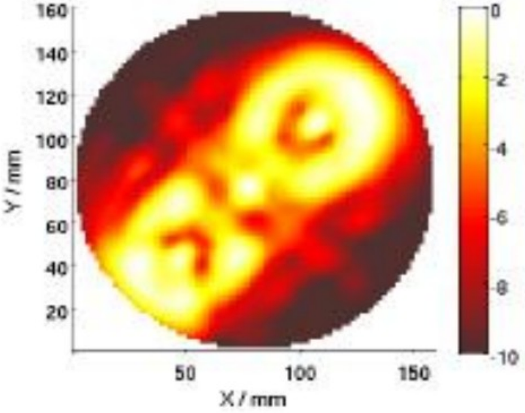
Breast and hemorrhagic brain stroke cancer detection


Experimental Setup

- **Original Geometry**

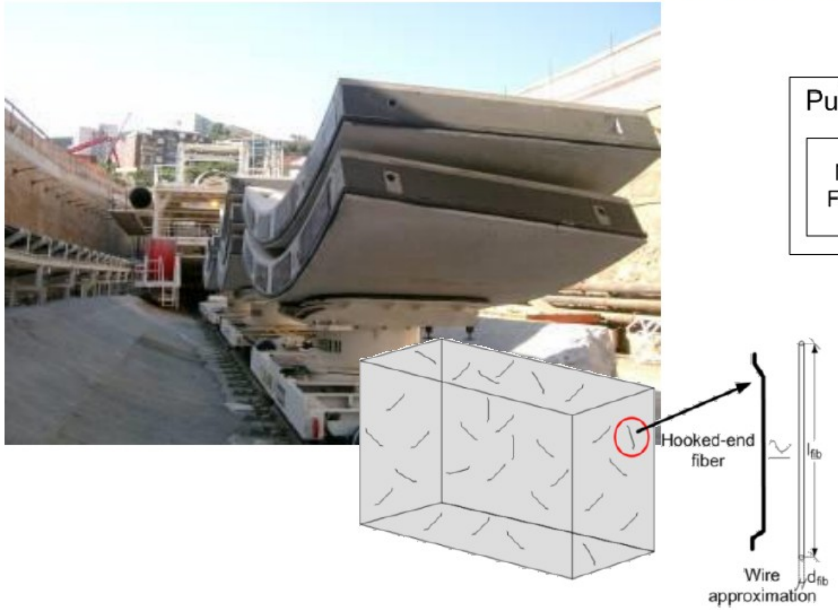


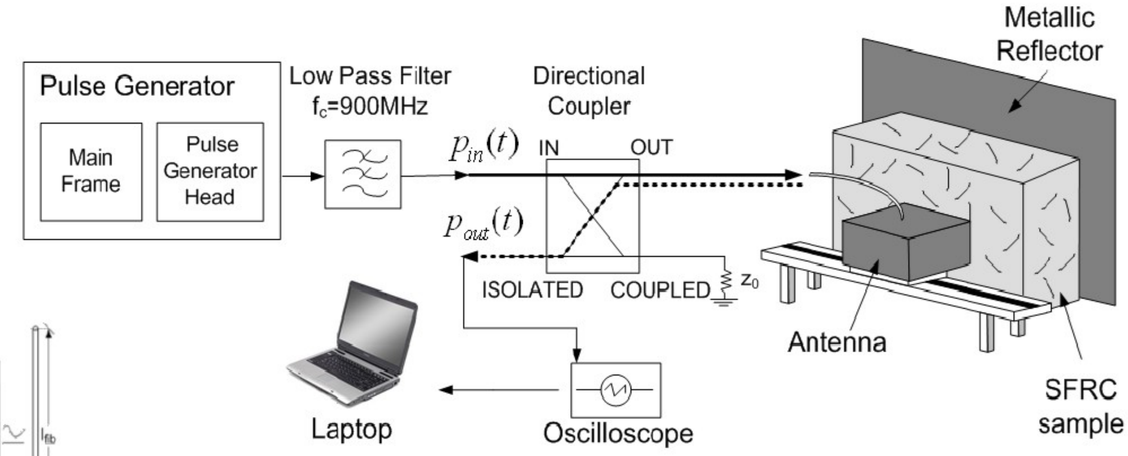
- **Reconstruction**

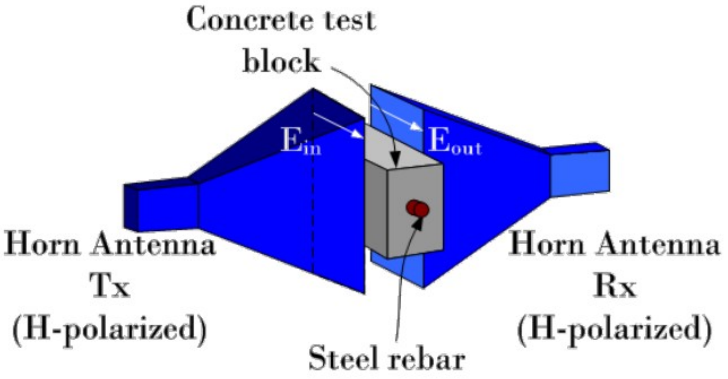


Microwave Nondestructive Evaluation of Civil Structures

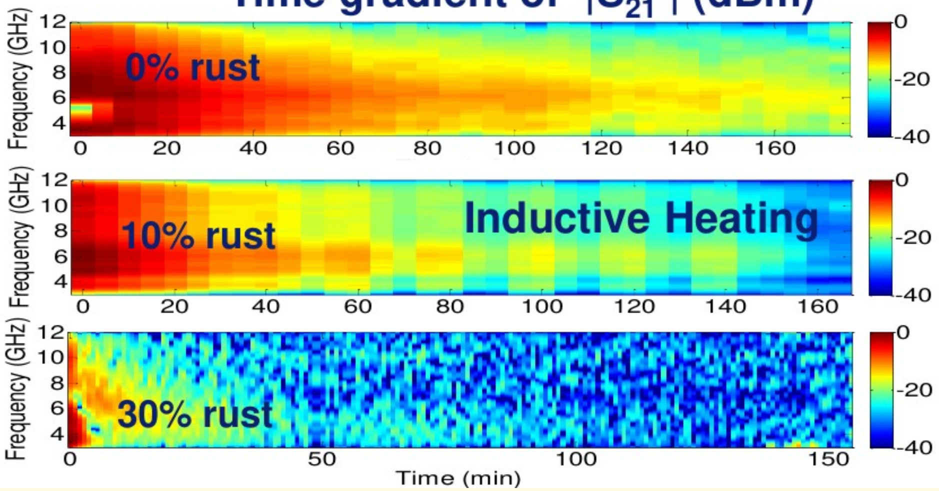
Noninvasive and Nondestructive Evaluation with microwaves of the corrosion in reinforced rebars.





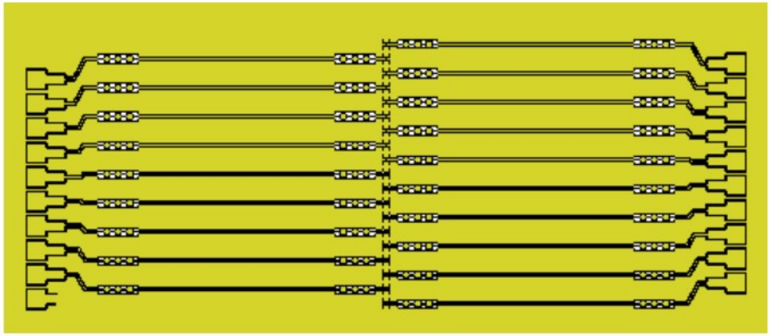


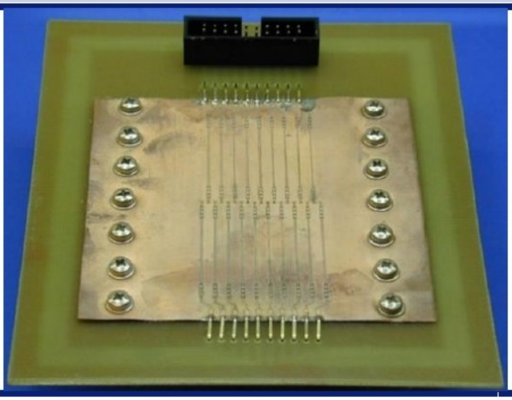
Time gradient of $|S_{21}|$ (dBm)

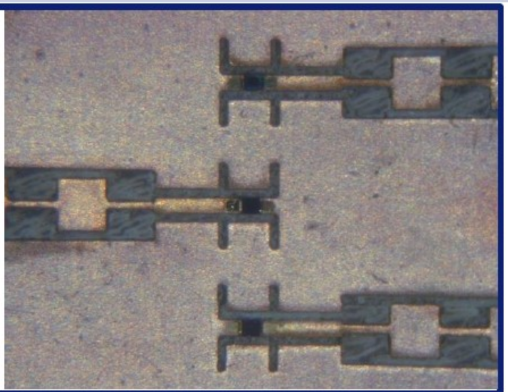


MST Retina System for THz Imaging

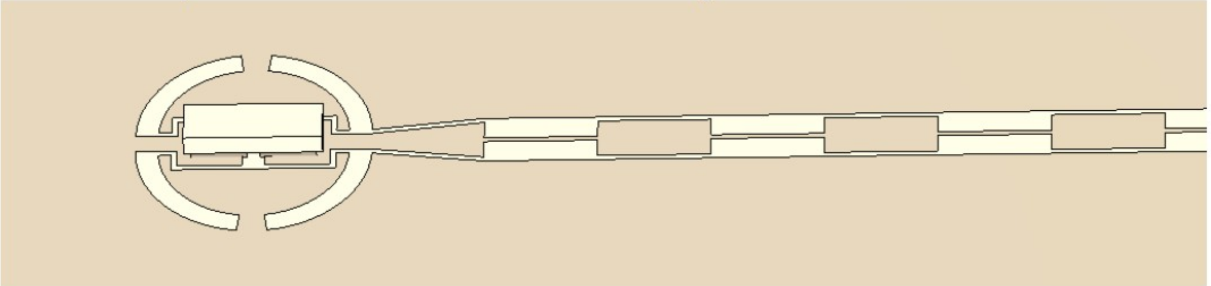
- PIN Diode Retina at 100 GHz

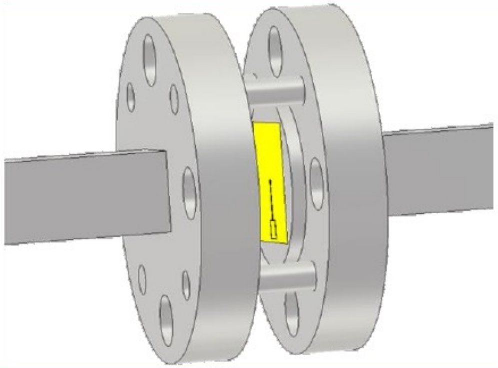




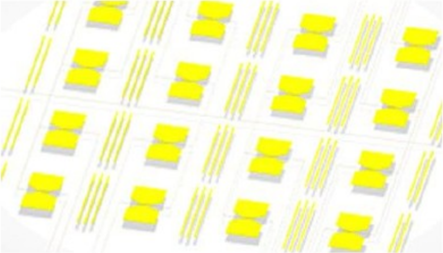


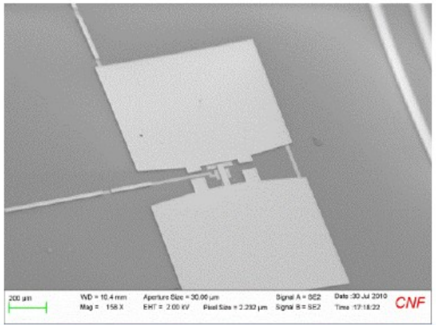
- Schottky Diode Retina at 300 GHz (One element characterization)

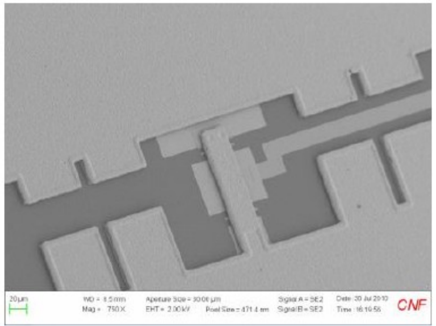




- MEMs Retina at 25 - 75 GHz





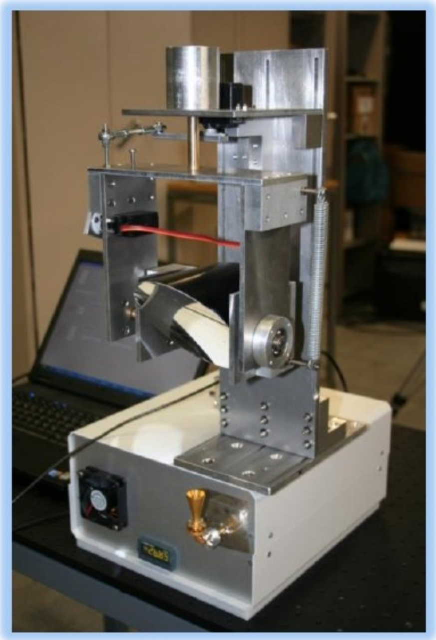


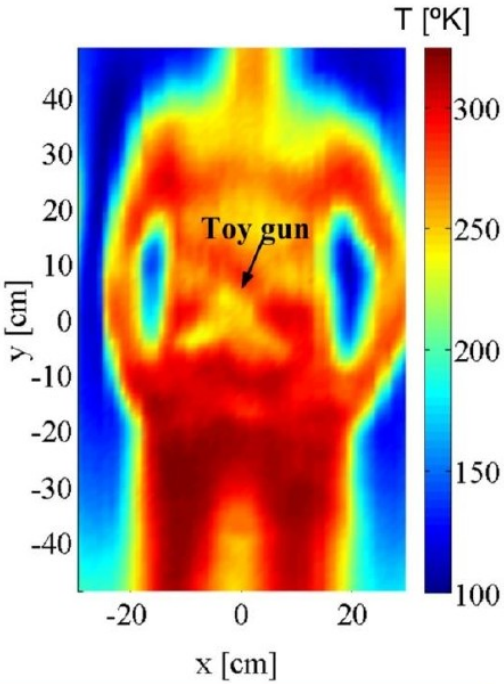
W-band Passive Imaging and THz Spectroscopy

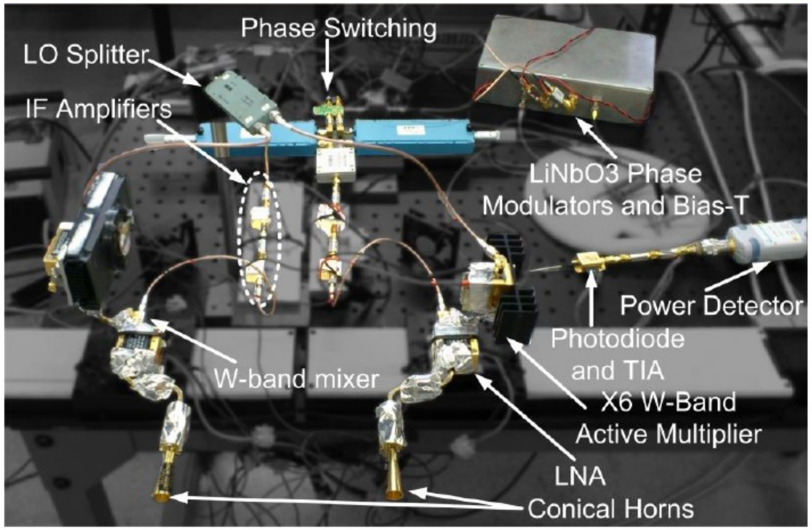
94 GHz TPR

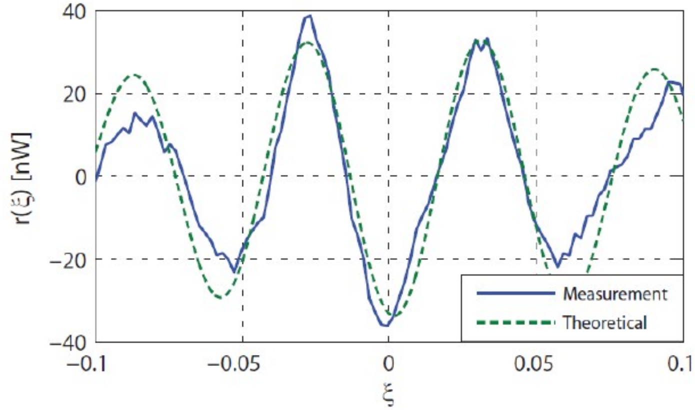
mmW-Outdoor

2-channel SA radiometer with optical correlator









Spatial resolution: 35mm

Radiometric resolution: 0.3 K (70ms integration time)

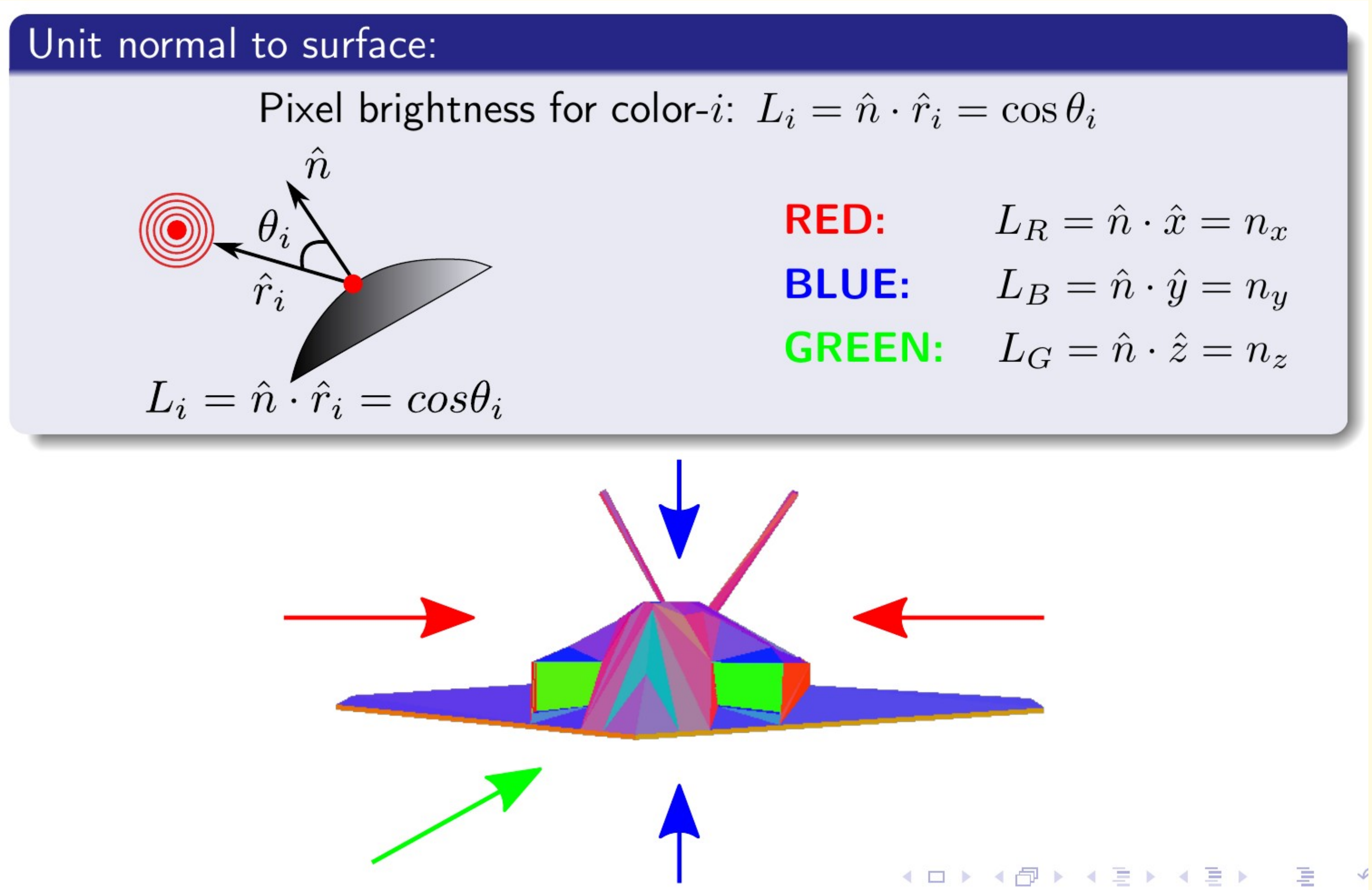
Integration time per pixel: Adjustable (1ms-500ms)

Scanning time: 7 minutes (100x50 pixels image)

- **Computational Electromagnetics**
  - High frequency methods
  - Integral equations and MoM discretization
  - Efficient direct solvers
  - **Low-rank matrix compression**

High frequency methods

- **GRECO** code for fast RCS computation using **Graphical Processing**



Unit normal to surface:

Pixel brightness for color-i: $L_i = \hat{n} \cdot \hat{r}_i = \cos\theta_i$

$L_i = \hat{n} \cdot \hat{r}_i = cos\theta_i$

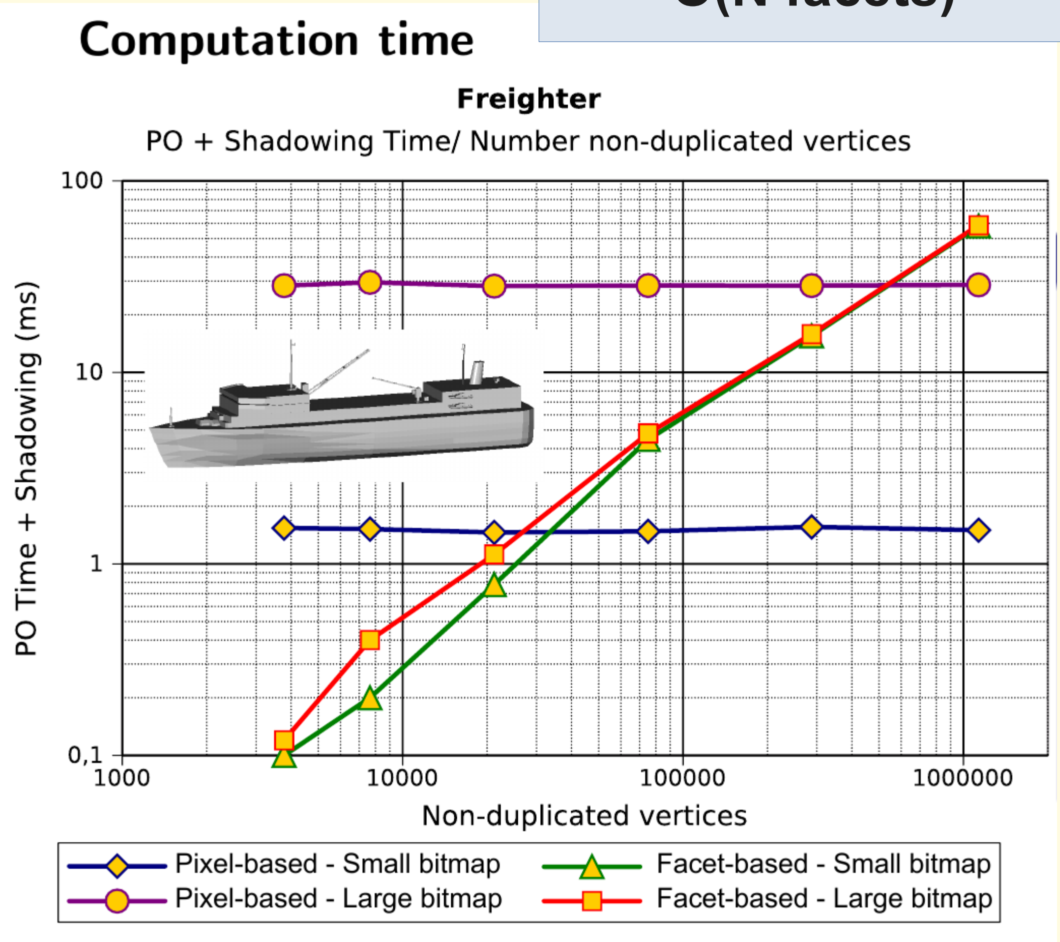
RED: $L_R = \hat{n} \cdot \hat{x} = n_x$

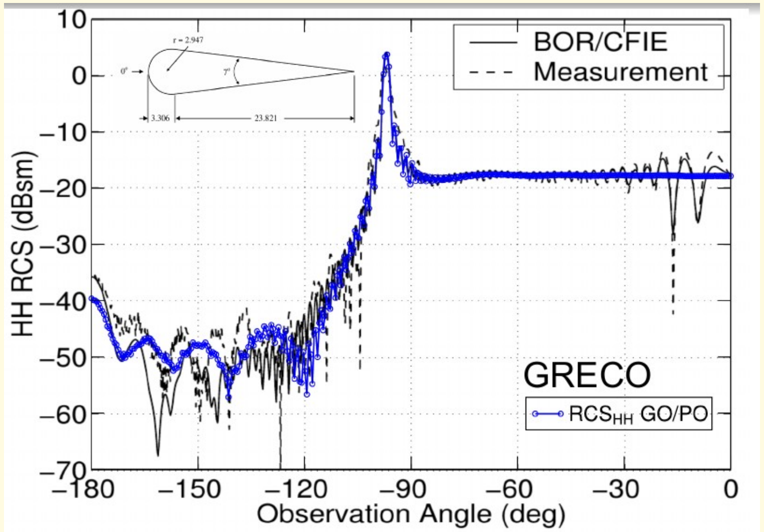
BLUE: $L_B = \hat{n} \cdot \hat{y} = n_y$

GREEN: $L_G = \hat{n} \cdot \hat{z} = n_z$

High frequency methods

Computation time,
including shadowing,
O(N facets)





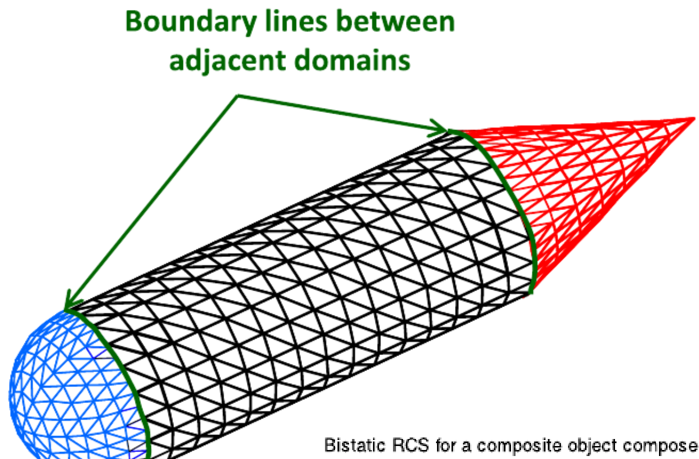
<u>From 1990 to present</u>

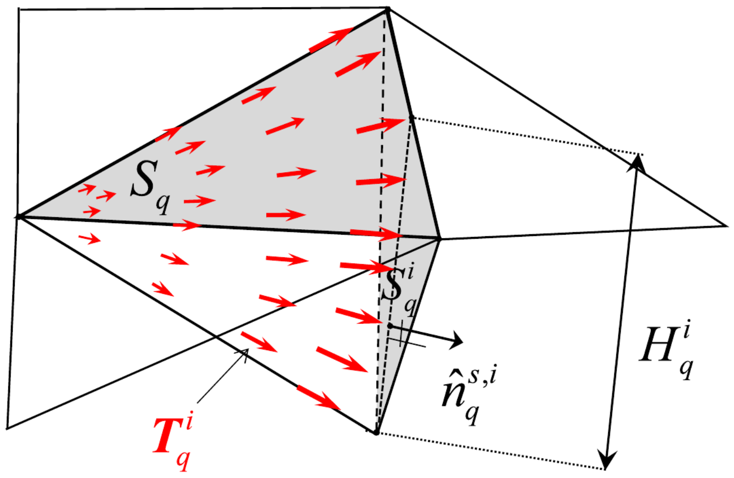
Licensed to:

Aircraft companies
Defense Research Institutions

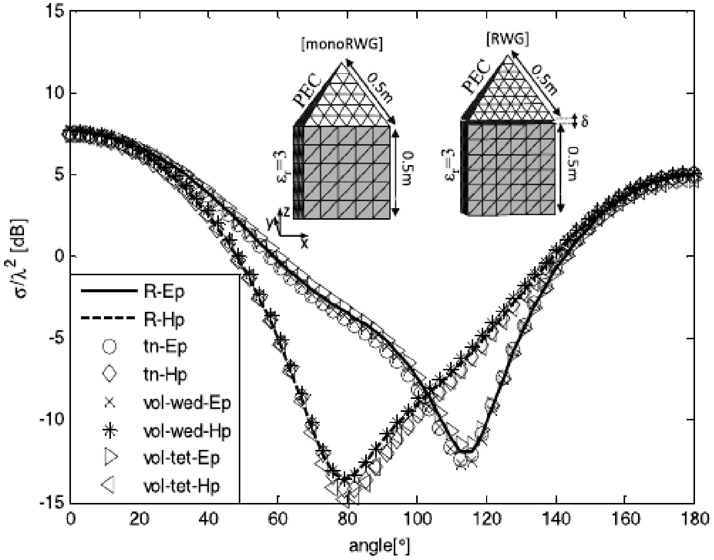
Integral equations and MoM discretization

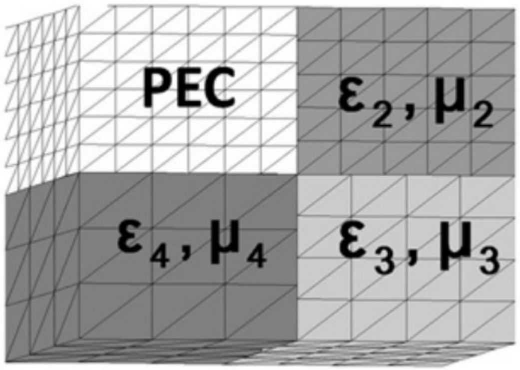
- Volumetric testing for non-conformal meshes





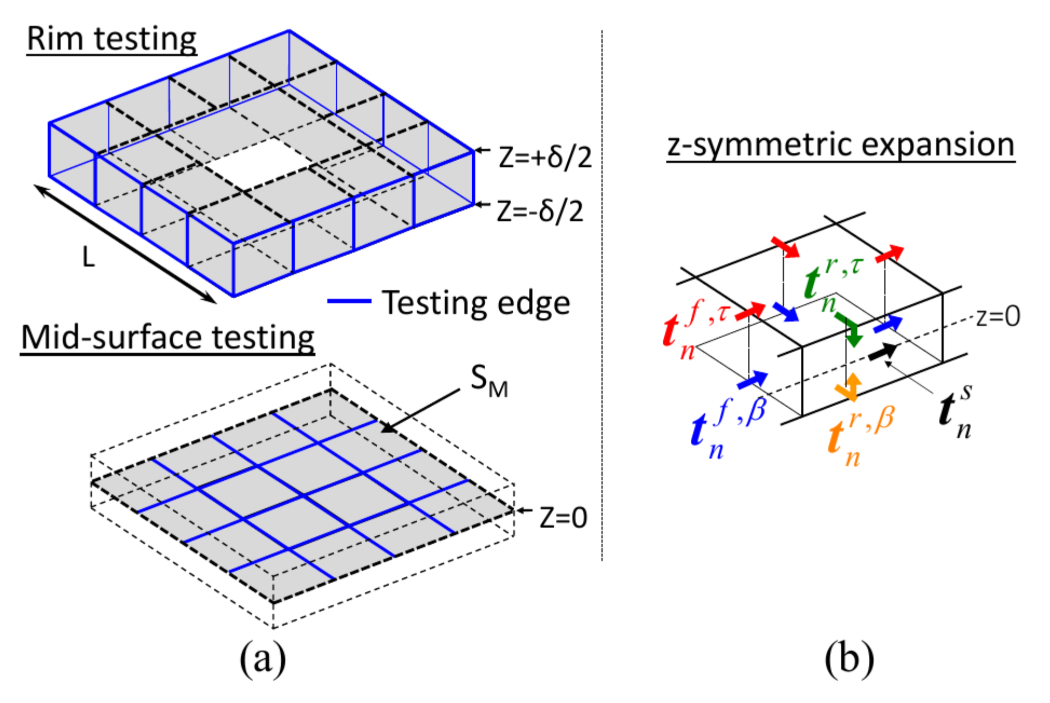
Bistatic RCS for a composite object composed of a dielectric cube and PEC

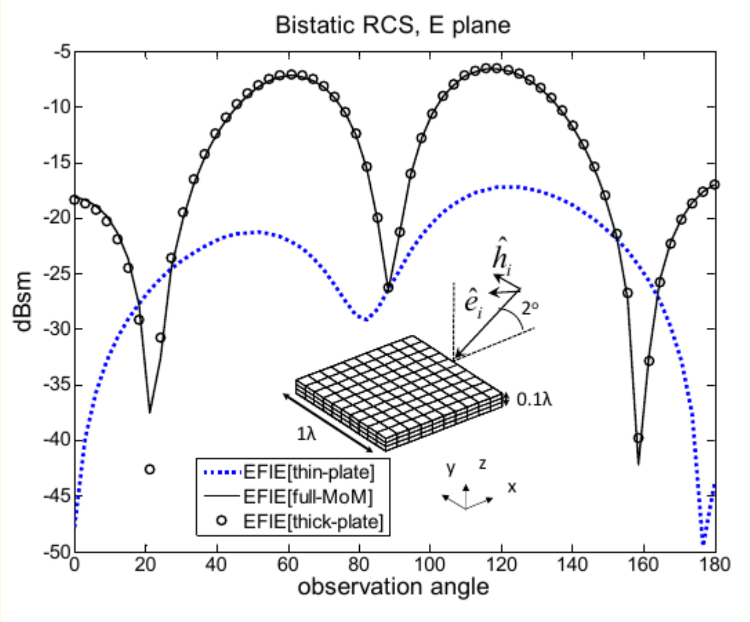




Integral equations and MoM discretization

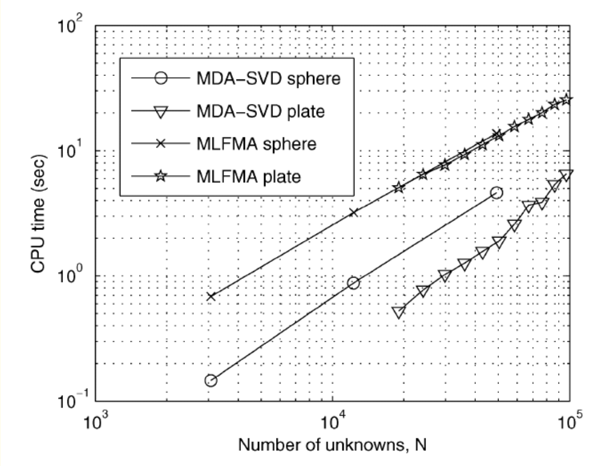
- Efficient and accurate modeling of thick plates

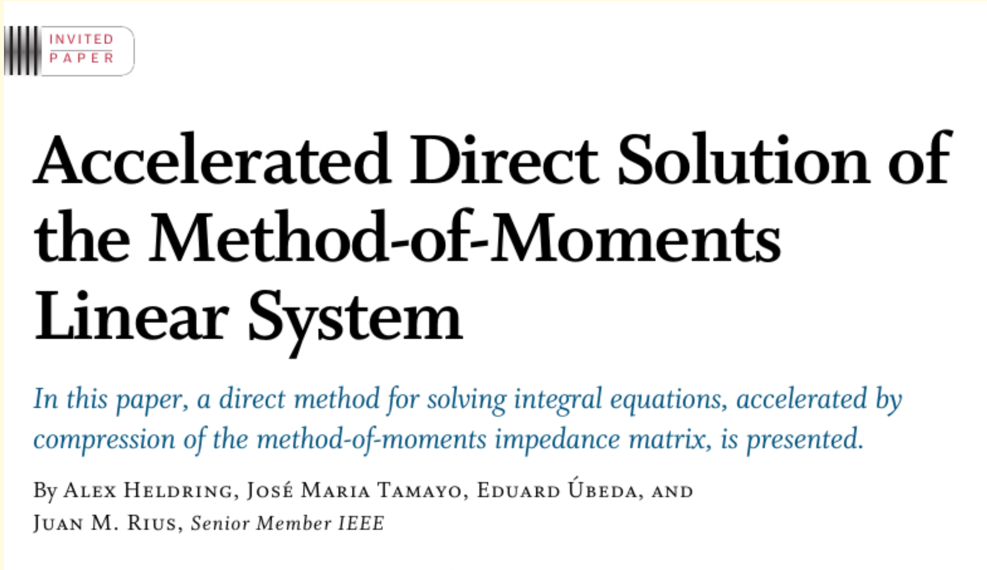




Efficient fast solvers

- IE-MEI: the most efficient IE solver ever in 2D: **N~10^9 in 32min** (1996)
- MDA in 3D with (**the 1^{st}**) SVD **post-compression** (2008)
- MLCBD: **1^{st}** successful and efficient fast **Direct Solver** for MoM linear system (2011)
- Sparsified ACA (SPACA), 2013

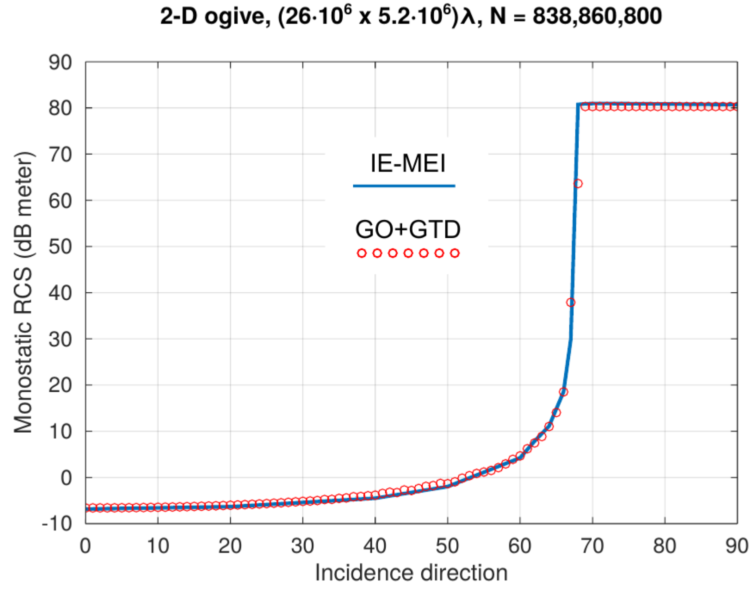



INVITED PAPER

Accelerated Direct Solution of the Method-of-Moments Linear System

In this paper, a direct method for solving integral equations, accelerated by compression of the method-of-moments impedance matrix, is presented.

By ALEX HELDRING, JOSÉ MARIA TAMAYO, EDUARD ÚBEDA, AND JUAN M. RIUS, *Senior Member IEEE*



Low-rank matrix compression

- **Integral equations discretized by method of moments**

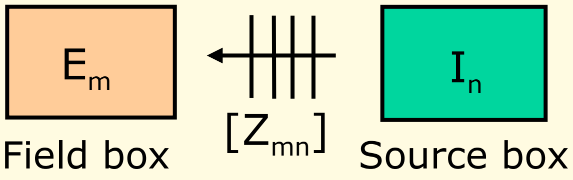
EFIE + **MoM** $\Rightarrow$ $-[E^i]=[Z]\cdot[I]$ $\Rightarrow$ **GMRES** $\Rightarrow$ $[I]$

[I] : Induced current in RWG basis functions (N unknowns)
[Z]: Impedance matrix, NxN
$[E^i]$: Incident field tested by RWG basis functions (Galerkin)

- Iterative solution: i.e. GMRES
  - In iteration k, $[Z][I^{(k-1)}] \rightarrow [I^{(k)}]$ with computational cost N^2

Compressing [Z] to a chain of small matrices saves storage memory and computation time

Low-rank submatrix compression



Z_{mn} is low rank, $R << M,N$ and can be compressed

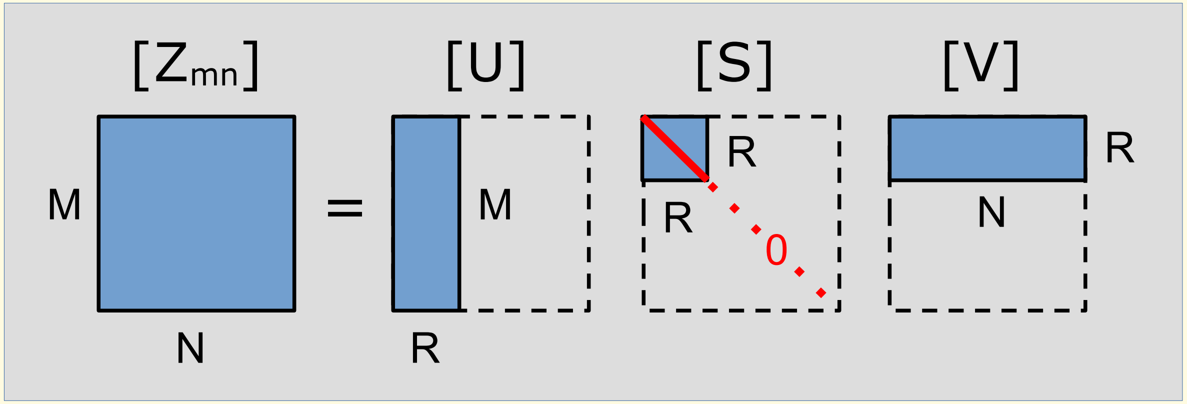
Storage $MR + R^2 + NR << MN$

Singular value decomposition (**SVD**):

$[Z_{mn}] = [U][S][V]$

$[S]$ = diagonal
R = rank of $[Z_{mn}]$

U,V = orthogonal matrices



- Size of $[U][S][V]$ much smaller than size of $[Z_{mn}]$: compression
- SVD is very expensive (N^3), and needs all elements of $[Z_{mn}]$ (N^2)
- Fast compression algorithms are **much faster** and only need to **compute a few elements** of $[Z_{mn}]$

Fast matrix compression algorithms

- Basic algorithms: simple, but no so efficient in time and storage (N^p):
  - **Adaptive Cross Approx**. (ACA): the most widely used
  - **CUR** (or Matrix Decomposition Algorithm MDA): our favourite (since 1997)
- Very efficient (N log N) but very complex multilevel algorithms:
  - **MLFMA**: only for specific G
  - Low-rank "algebraic": Applicable to any compressible integral equation
    - **Butterfly** & friends: Use ACA or CUR as basic compression routine

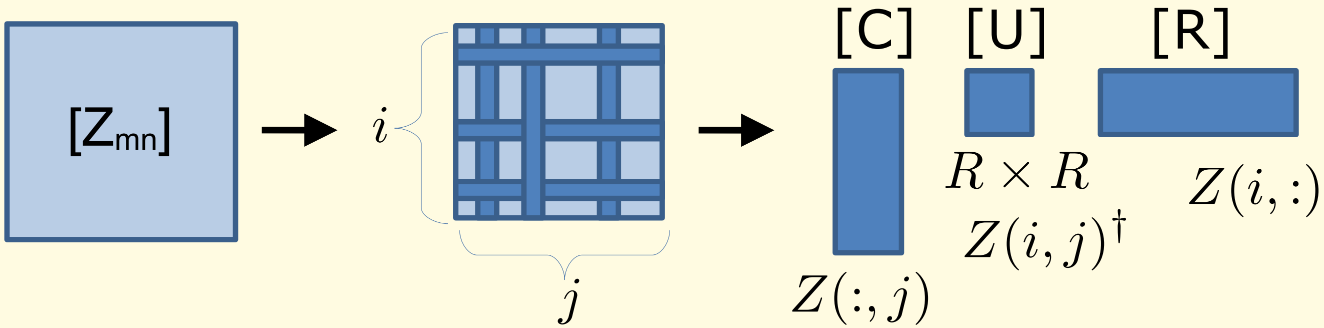
Our aim:

Improve CUR to achieve the maximum possible performance

(basic compression routine of Butterfly)

CUR algorithm

- $[Z_{mn}]$ can be also compressed as a [C][U][R] product:



[C] = cols of [Z] = Z(:,j)
[R] = rows of [Z] = Z(i,:)
[U] = pinv(Z(i,j))

Pivots are the indices (i,j) of selected rows and columns of Z

- **CUR vs ACA**: Great advantage for **parallel computation** of rows/columns
  - … but we are not sure about the compression error
- The rank R must be at least equal to the number of Degrees of Freedom (DoF)

$$R \approx 3 + 30\frac{R_s R_f}{d\lambda}$$

R_s = min. sphere source box
R_f = min. sphere field box
d = between spheres centers

ACA & CUR
Compressed matrices
rank **R > SVD-rank**

Random pivots:
No algorithm to find a uniform distribution of pivots along object surface, (that must be repeated for every block of Z)

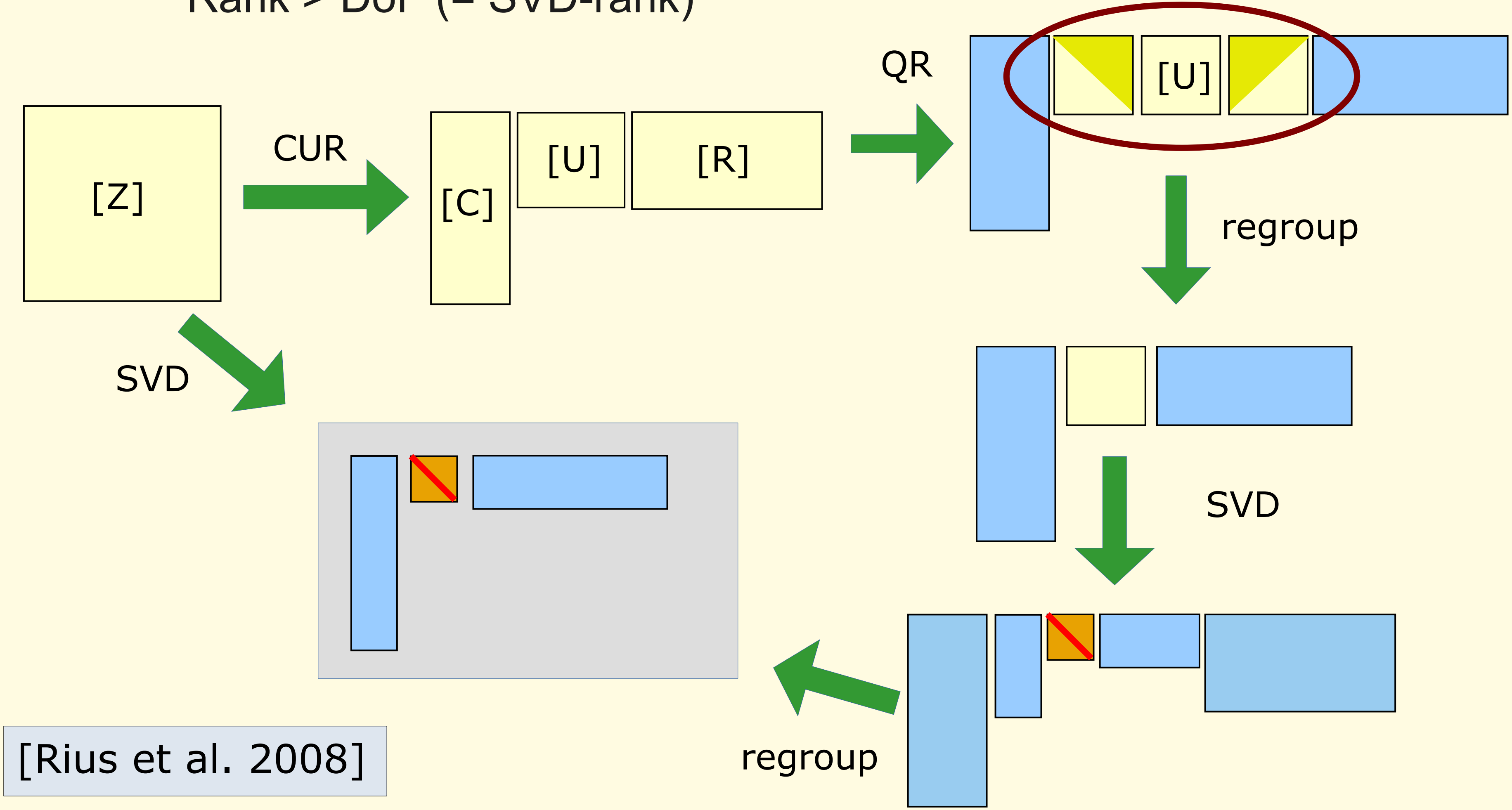
2008

QR-SVD post-compression

🟧 Diagonal matrix

🟦 Orthogonal matrix

- ACA or CUR compressed size not optimum
  - Rank > DoF (= SVD-rank)

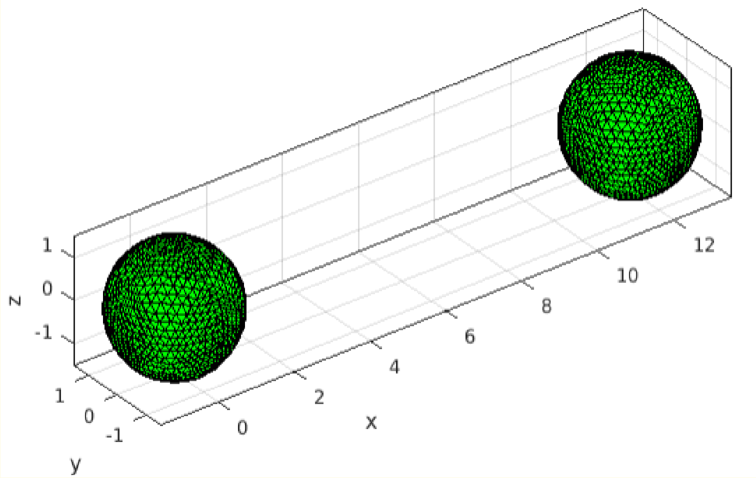


[Rius et al. 2008]

But, …

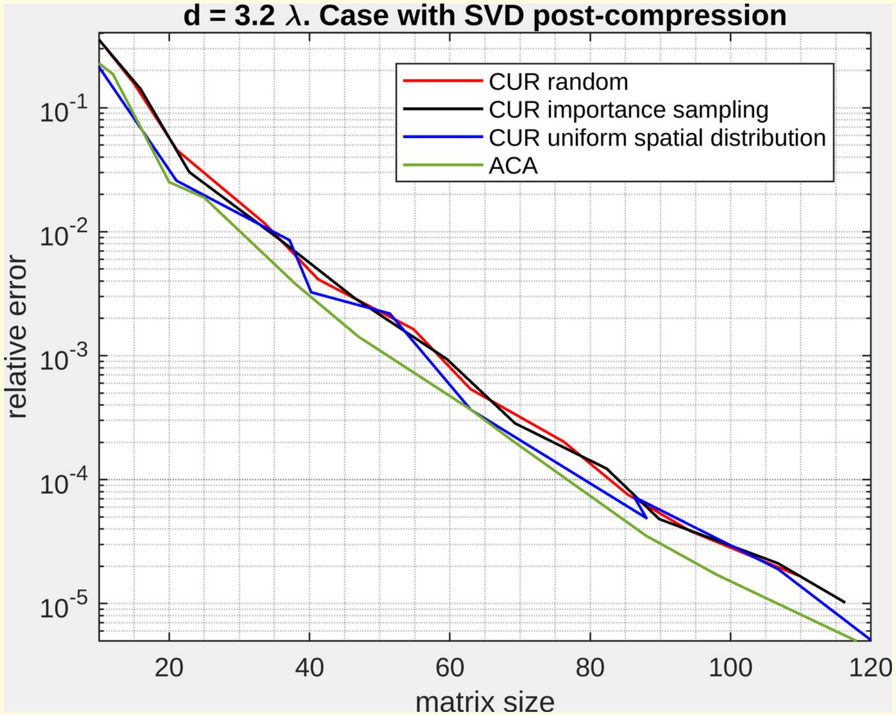
Do random (i,j) work?

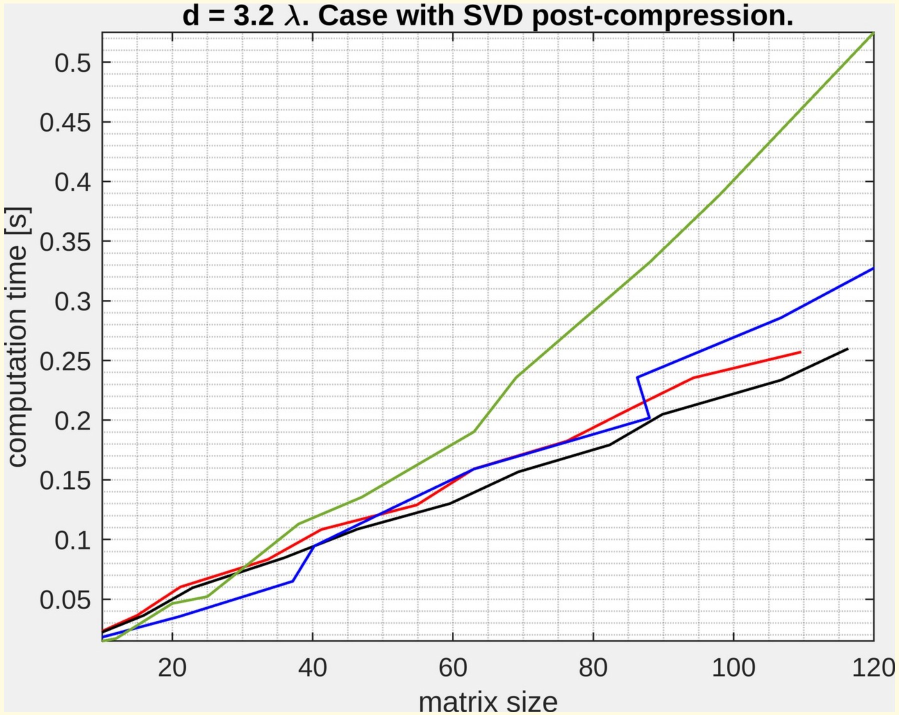
- Analysis of the interaction between two objects
  - Compression of the submatrix representing this interaction



Randomized CUR

- Interaction between 2 spheres R=0.8m d=3.2m

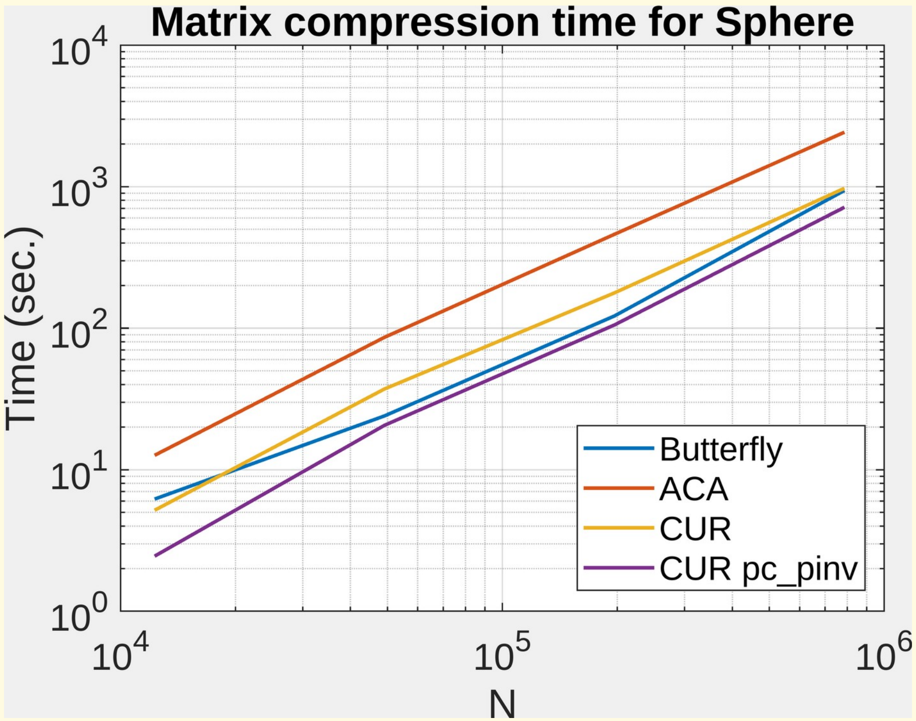


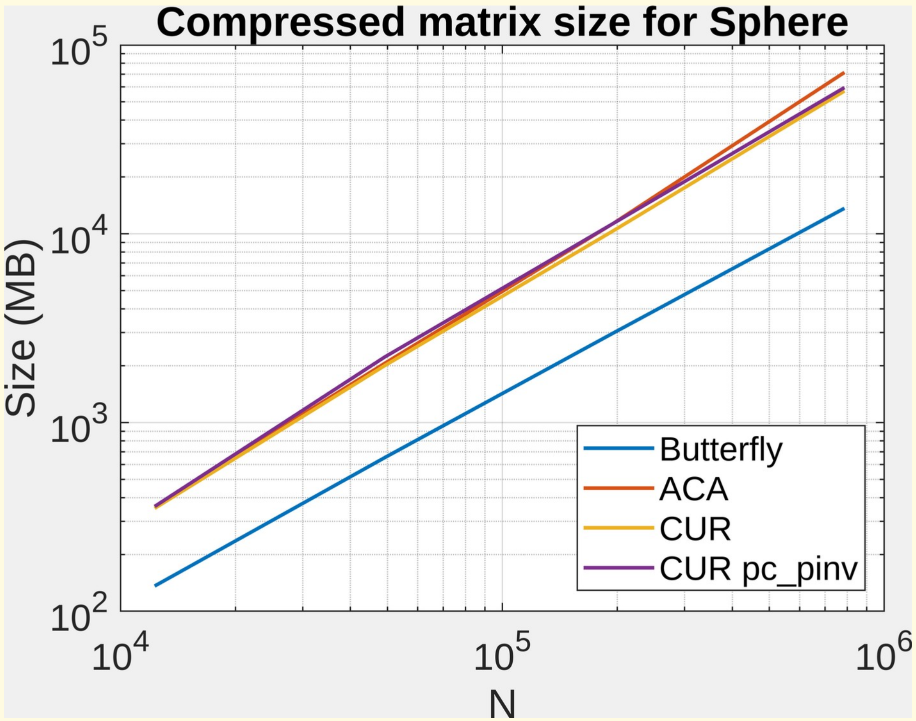


Closest boxes in a multilevel subdivision of object

- Error decreases with matrix size
- Computation time increases, as expected

Results: easy case (sphere)



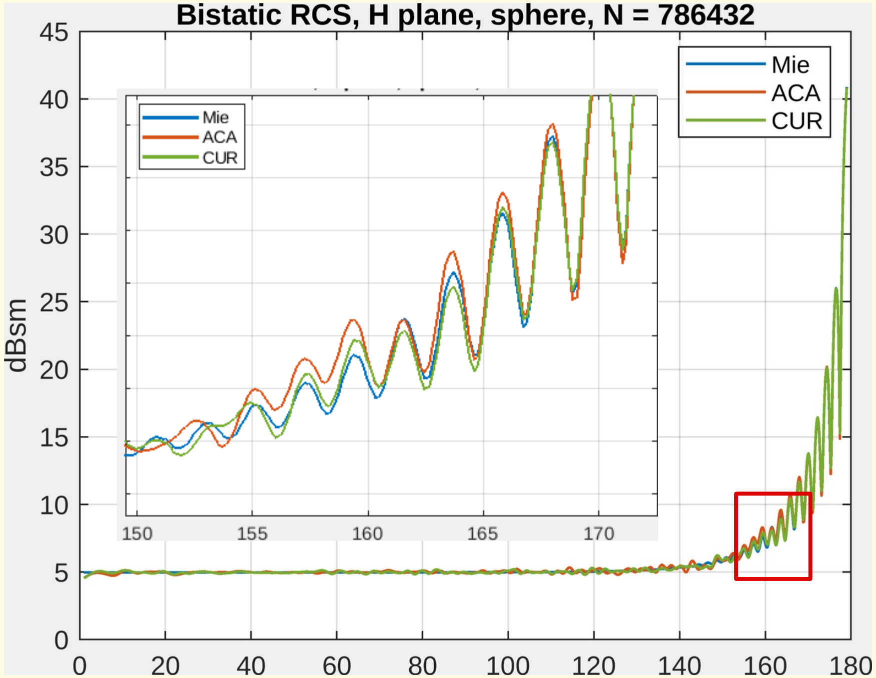


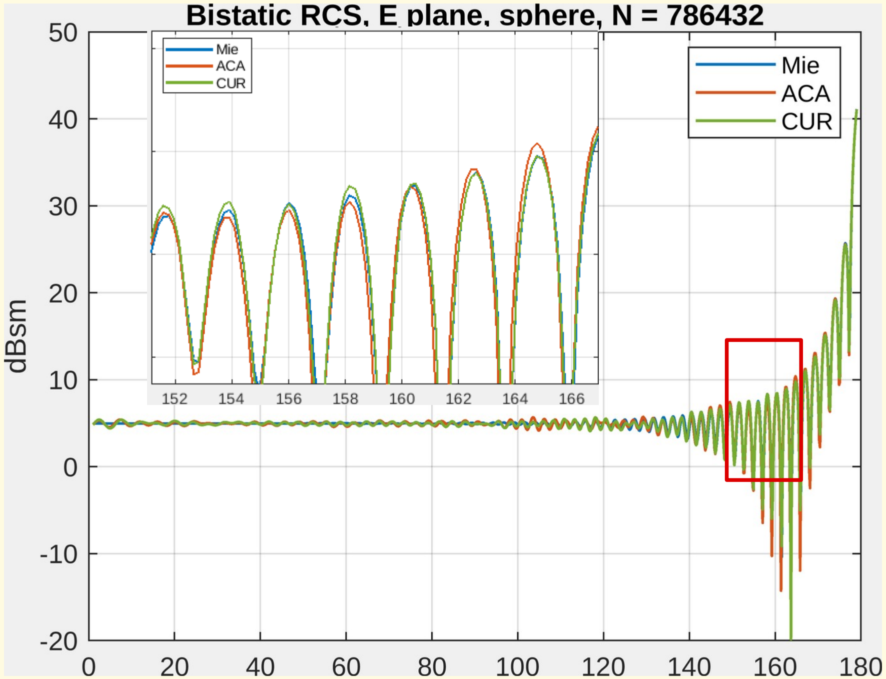
CFIE, discretization size λ/10 in all cases

CPU: Intel core i5-11500 @ 2.70GHz, 6 cores
(cheap office desktop + 128GB RAM)

- ‘Cutting-edge’ Butterfly implementation by Alex Heldring
- CUR pc_inv will be implemented next in the Butterfly

Results: easy case (sphere)





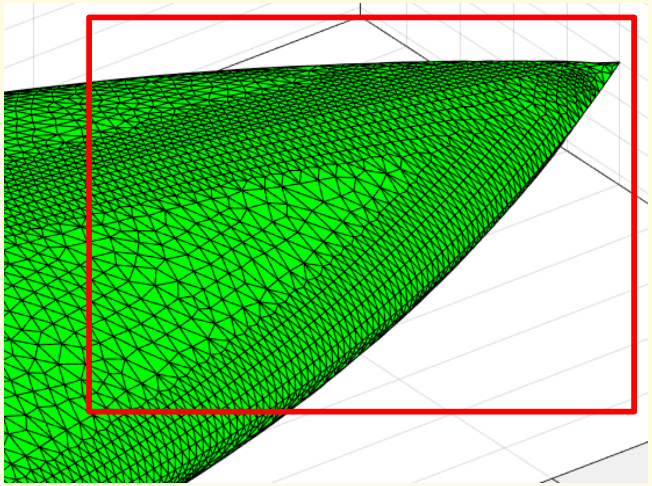
Sphere R=12.3 λ,

N=786432, CFIE

	Compression time	Compressed size	RCS error
Butterfly	939 sec.	13643 MB	0.09 dB
ACA	2421 sec.	71580 MB	0.13 dB
CUR	972 sec.	57096 MB	0.17 dB
CUR pc	714 sec.	59518 MB	0.17 dB

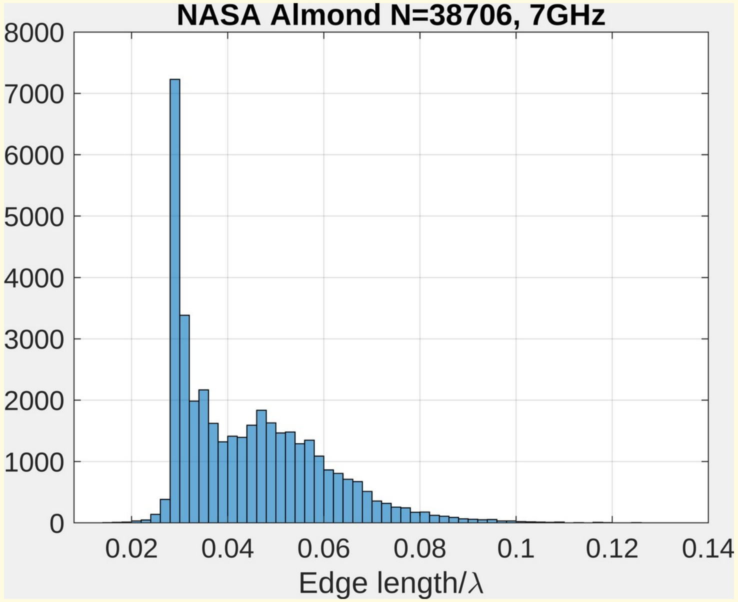
Results: difficult case (the NASA almond)

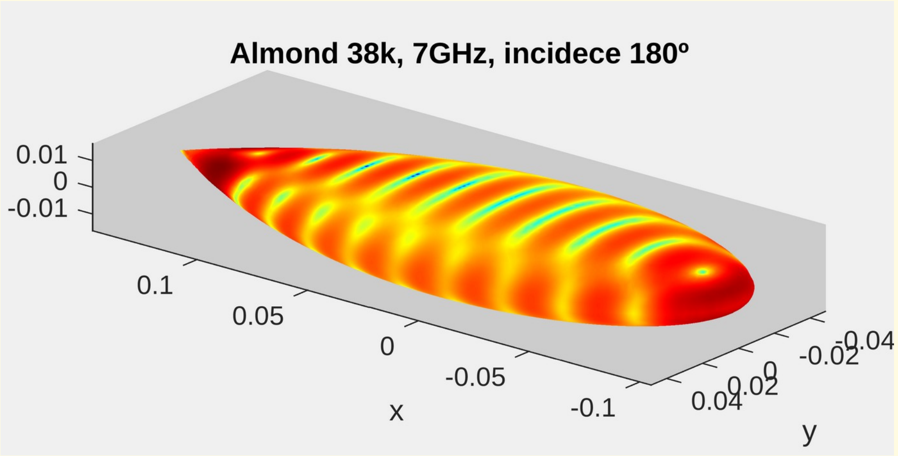
- Difficult case: NASA Almond
  - Smooth surfaces and a tip
  - Irregular meshing: random samples concentrate in the denser areas



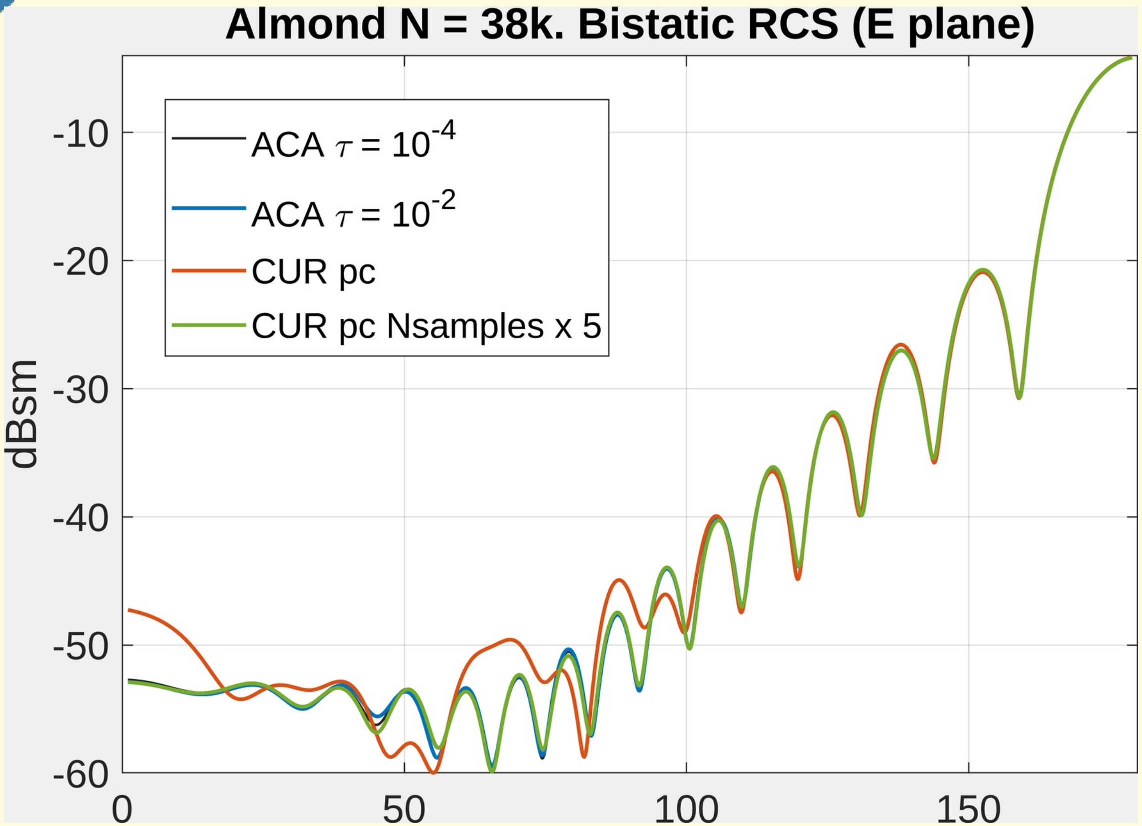
  - Needs larger N samples

$$R > 3 + 30\frac{R_s R_f}{d\lambda}$$


NASA Almond N=38706, 7GHz


Almond 38k, 7GHz, incidece 180º

Almond 38k unknowns



The simple formula to estimate Nsamples

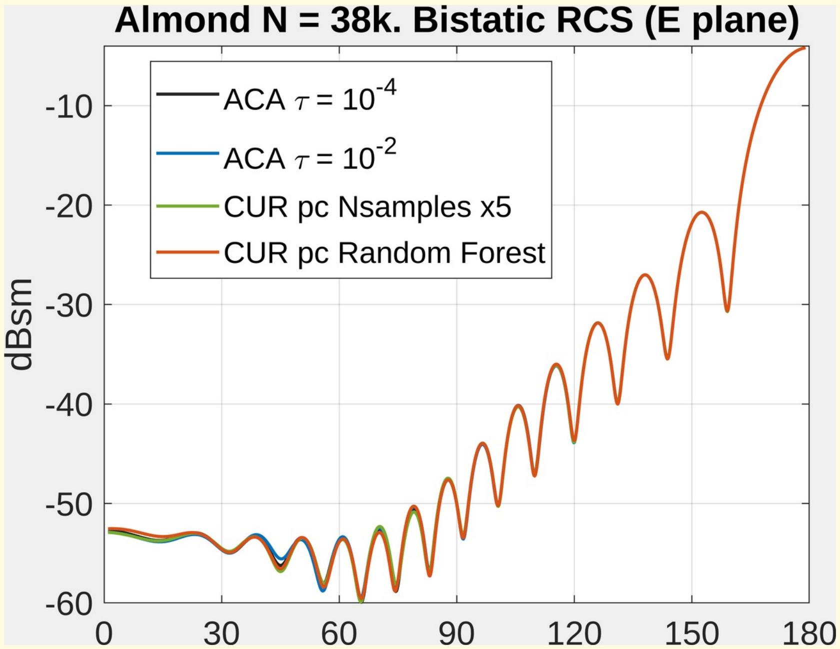
$$R = 3 + 30\frac{R_s R_f}{d\lambda}$$

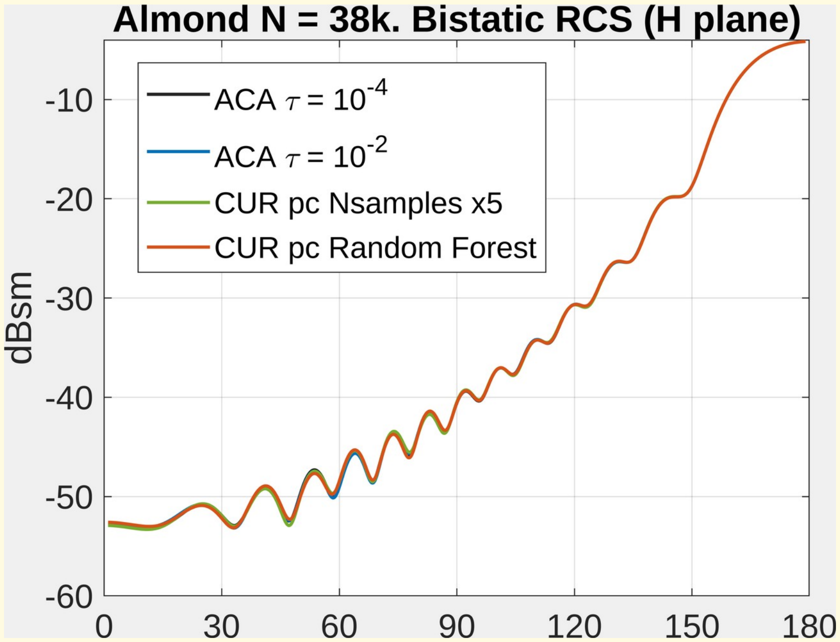
is very aggressive for fast computation, and areas with a coarse mesh are under-sampled: we need to oversample x5 the whole object

Algorithm	error E-plane	error H-plane	compression time (sec.)
ACA (τ = 10^{-2})	0.13 dB	0.071 dB	38.4
CUR PC	1.7 dB	1.1 dB	13.6
CUR PC Nsamples x5	0.14 dB	0.073 dB	19.5

CUR PC x5 is still faster than ACA, but we are never sure about the error

Almond 38k unknowns


Almond N = 38k. Bistatic RCS (E plane)


Almond N = 38k. Bistatic RCS (H plane)

Preliminar results with Random Regression Forest (to be improved)

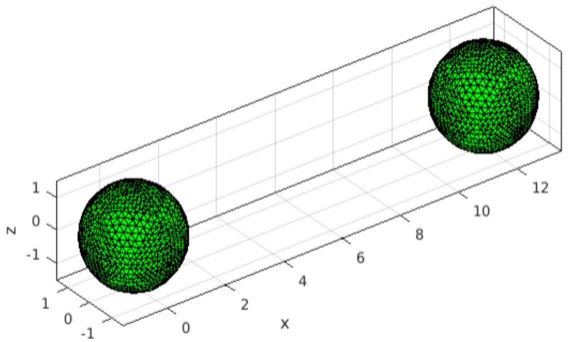
Algorithm	error E-plane	error H-plane	compression time (sec.)
ACA ($\tau = 10^{-2}$)	0.13 dB	0.071 dB	38.4
CUR pc	1.7 dB	1.1 dB	13.6
CUR pc QR 5 x Ns	0.14 dB	0.073 dB	14.2
CUR Random Forest	**0.15 dB**	**0.08 dB**	**32.6**

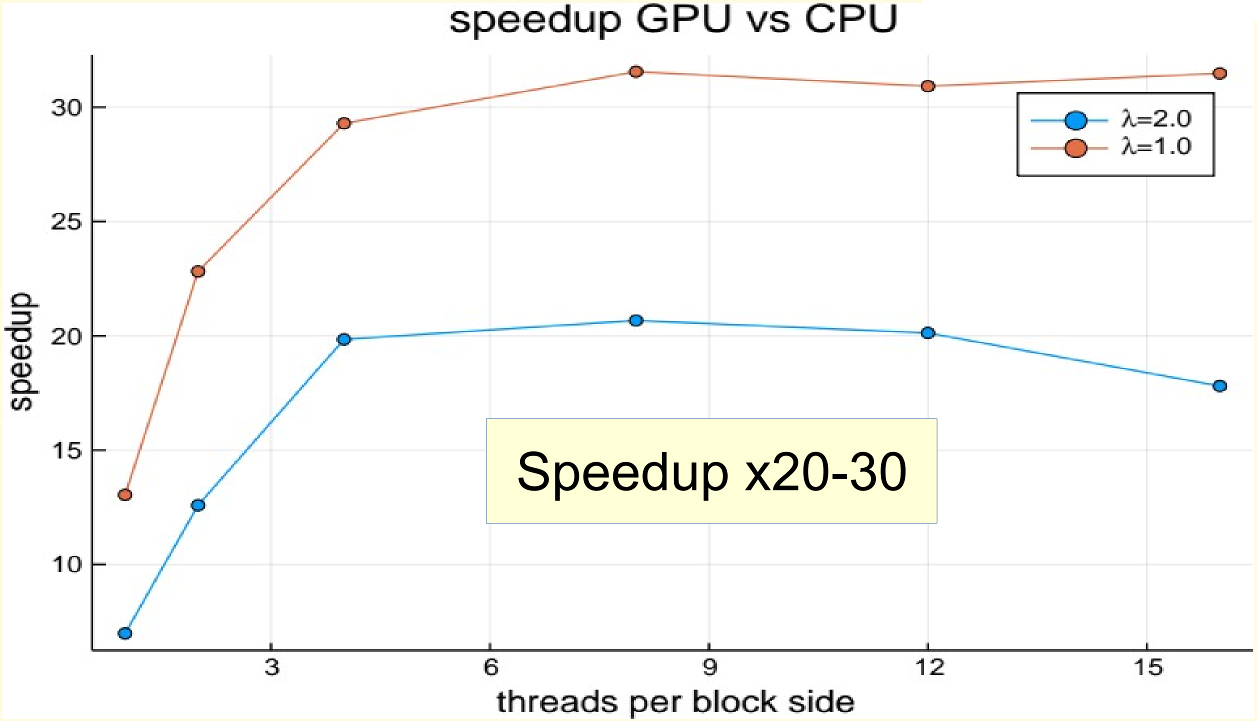
GPU implementation

- CUR very efficient in a parallel environment

Interaction between two PEC spheres:

- R=1m, d=12m
- lambda = 2m: N=3072
- lambda = 1m: N=12288





Speedup x20-30

Conclusions

- Randomized CUR:
  - Compression x2 faster than ACA, same compressed size
  - Post-compression integrated in the pseudo-inverse:
    x3 faster than ACA, comparable to Butterfly up to 1M unkowns
  - RRQR to select important samples:
    Allows oversampling x5 with small increase in compression time
  - Compressed size similar to ACA, and much larger than Butterfly
- Parallelization of CUR:
  - No inter-core communications:
    Easy and efficient paralellization
  - x30 speed-up in GPU implementation