Introduction to scattering calculations: Python T-matrix package

`pytmatrix`

Davide Ori – University of Bologna

Problem: simulate radar quantities

We have an ensamble of particles and we want to calculate the expected radar measurements

Radar Reflectivity

$$Z_{h,v} = \frac{4\lambda^4}{\pi^4 |K|^2} \int_{D_{min}}^{D_{max}} |f_{hh,vv}(\pi, x)|^2 N(x)\, dx \quad [mm^6/m^3] \rightarrow Z_{H,V} = 10 \log_{10}(Z_{h,v}) \quad [dBZ]$$

We integrate over the particle distribution the square modulus of the single particle ***complex scattering amplitude*** at backscattering direction and horizontal or vertical polarization.

Scattering amplitude

It represents the amount of energy that is scattered in a particular direction with a particular polarization state by each particle which is defined by the characteristic quantity *x*

$$f_{xy}(\theta, x)$$

Expressed as a function of the characteristic variable of the particle distribution (which is almost always some particle "size")

Dependent on the scattering direction (θ=0 is *forward scattering* and θ=π is *backscattering*)

Taking into account of the polarization state of the incoming and scattered electomagnetic wave

- hh → copolar horizontal
- vv → copolar vertical
- hv → crosspolar

Problem: simulate radar quantities

<u>Summary of radar variables you may want to calculate</u>

Radar Reflectivity

$$Z_{h,v} = \frac{4\lambda^4}{\pi^4 |K|^2} \int_{D_{min}}^{D_{max}} |f_{hh,vv}(\pi, x)|^2 N(x)\, dx \quad [mm^6/m^3] \rightarrow Z_{H,V} = 10 \log_{10}(Z_{h,v}) \quad [dBZ]$$

Differential Reflectivity

$$Z_{dr} = 10 \log_{10}\left(\frac{Z_h}{Z_v}\right) = Z_H - Z_V \quad [dBZ]$$

Correlation Coefficient

$$\rho_{hv} = \frac{\int_{D_{min}}^{D_{max}} f_{hh}(\pi, x) f_{vv}(\pi, x) N(x)\, dx}{\sqrt{\int_{D_{min}}^{D_{max}} |f_{hh}(\pi, x)|^2 N(x)\, dx \int_{D_{min}}^{D_{max}} |f_{vv}(\pi, x)|^2 N(x)\, dx}}$$

Specific Differential Phase shift

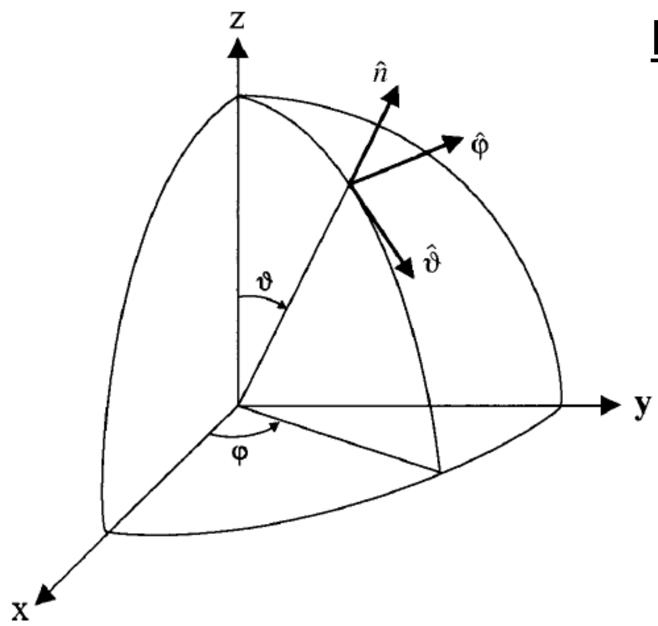
$$K_{dp} = \frac{180\lambda}{\pi} \int_{D_{min}}^{D_{max}} \Re\left[f_{hh}(0,x) - f_{vv}(0,x)\right] N(x)\, dx \quad [°/km]$$

Specific Attenuation

$$A_{H,V} = 8.686 \cdot \lambda \int_{D_{min}}^{D_{max}} \Im\left[f_{hh,vv}(0,x)\right] N(x)\, dx \quad [dB/km]$$

General Scattering Theory: Reference Frames

Mishchenko, Travis, Lacis (2002) - Scattering, Absorption and Emission of light by small particles. Cambridge University Press. Free online book at http://www.giss.nasa.gov/staff/mmishchenko/publications/book_2.pdf



Plane Wave in the laboratory reference frame (xyz)

Both incident and scattered wave propagation vector are expressed by the couple of angles θ (zenith) and φ (azimuth). Electromagnetic field in any position **r** is therefore

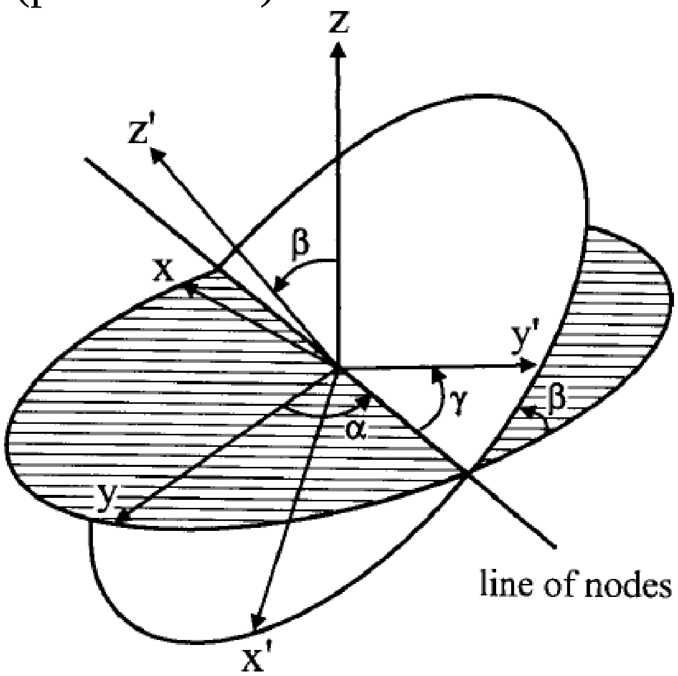
$$\boldsymbol{E}(\boldsymbol{r}) = (E_{\theta 0} \cdot \hat{\theta} + E_{\varphi 0} \cdot \hat{\varphi}) \exp(ik\,\hat{n}\,\boldsymbol{r})$$

here time armonic part has been suppressed.
note that for radar studies θ and φ components are respectively vertical and horizontal component (polarization)

Transform of lab (xyz) to particle (x'y'z') frame

is performed using the so called zyz-convention with the three Euler's angles (α,β,γ). First rotate z axis by α turning y to the line of nodes, then rotate this line by β turning z to z' and finally rotate z' by γ.

Note: Since we are going to deal with particles which are rotationally simmetric around z', last angle γ is useless for our purposes.
You can just think about β as the canting angle of the particles



Scattering Theory: amplitude matrix

In the laboratory reference frame the electrical field of the scattered wave is related to the amplitude of the incident wave via the complex 2x2 **amplitude matrix S** which depends on the orientation of the scattering particle and the propagation direction of the incident and scattered wave

$$\begin{pmatrix} E_\theta^{sca} \\ E_\varphi^{sca} \end{pmatrix} = \frac{\exp(ik\,\boldsymbol{r})}{r}\,\boldsymbol{S}(\hat{n}^{sca}, \hat{n}^{inc}, \alpha, \beta, \gamma) \begin{pmatrix} E_{\theta 0}^{inc} \\ E_{\varphi 0}^{inc} \end{pmatrix} \qquad \boldsymbol{S}(\hat{n}^{sca}, \hat{n}^{inc}) = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}$$

Note: In the particle reference frame the same relation holds, but the amplitude matrix does not depend on the particle orientation. The advantage of laboratory coordinate system is that it can be chosen such that it most adequately represents the physical mechanism of particle orientation (like falling of drops)

Returning to our problem of finding the complex scattering amplitude ...

- if we consider a specific particle x=D,
- with a specific orientation in space (α,β,γ)
- and we define the laboratory reference frame such that the radar is in the origin of the frame with z axis coincident with the vertical passing through the radar

$$f_{hh}(\pi, D) = 2\sqrt{\pi}\, S_{22}(\hat{n}^{sca} = -\hat{n}^{inc}) \qquad f_{vv}(\pi, D) = 2\sqrt{\pi}\, S_{11}(\hat{n}^{sca} = -\hat{n}^{inc})$$

$$f_{hh}(0, D) = 2\sqrt{\pi}\, S_{22}(\hat{n}^{sca} = \hat{n}^{inc}) \qquad f_{vv}(\pi, D) = 2\sqrt{\pi}\, S_{11}(\hat{n}^{sca} = \hat{n}^{inc})$$

Scattering Theory: Phase Matrix

But electromagnetic vector field are difficult to measure. It is generally more convenient to switch to the Stoke's vector notation which describes scattering in terms of **intensity** (*monochromatic energy flux*)

$$\boldsymbol{W} = \begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix} = \begin{pmatrix} E_\theta E_\theta^* + E_\varphi E_\varphi^* \\ E_\theta E_\theta^* - E_\varphi E_\varphi^* \\ -E_\theta E_\varphi^* - E_\varphi E_\theta^* \\ i\left(E_\varphi E_\theta^* - E_\theta E_\varphi^*\right) \end{pmatrix}$$

The Stoke's vector of the incident electromagnetic wave is therefore transformed in the Stoke's vector of the scattered electromagnetic wave using the real 4x4 **phase matrix Z**

$$\boldsymbol{W}^{sca} = \frac{1}{R^2} \boldsymbol{Z}\left(\hat{\boldsymbol{n}}^{sca}, \hat{\boldsymbol{n}}^{inc}, \alpha, \beta, \gamma\right) \boldsymbol{W}^{inc}$$

In this notation it is straightforward to define radar scattering cross sections
Example:

$$\sigma_{hh} = 4\pi \frac{I_h^{sca}}{I_h^{inc}} = 4\pi \frac{|E_\varphi^{sca}|^2}{|E_\varphi^{inc}|^2} = 2\pi\left(Z_{11} - Z_{12} - Z_{21} + Z_{22}\right)$$

T-matrix

Central point of the T-matrix method is not the solution, but the mathematical formalism:

"Expand incident and scattered field in vector *spherical vector wave functions*, and relate their expansion coefficients via the transition matrix **T**"

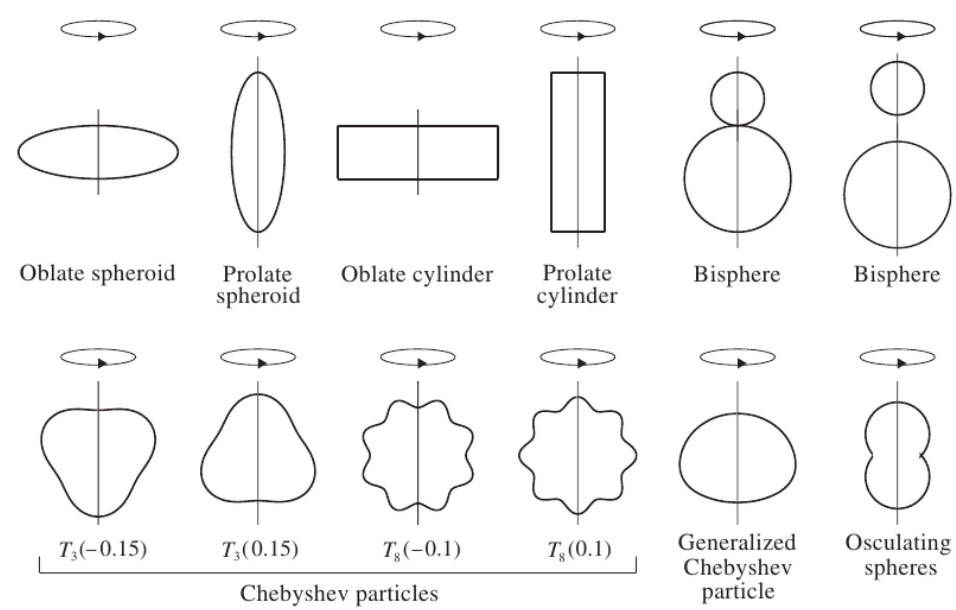
$$\boldsymbol{E}^{inc}(\boldsymbol{r})=\sum_{n=1}^{\infty}\sum_{m=-n}^{n}\left[a_{mn}\boldsymbol{M}_{mn}(k_1\boldsymbol{r})+b_{mn}\boldsymbol{N}_{mn}(k_1\boldsymbol{r})\right]$$
$$\boldsymbol{E}^{sca}(\boldsymbol{r})=\sum_{n=1}^{\infty}\sum_{m=-n}^{n}\left[p_{mn}\boldsymbol{M}_{mn}(k_1\boldsymbol{r})+q_{mn}\boldsymbol{N}_{mn}(k_1\boldsymbol{r})\right]$$

$$\boxed{\begin{pmatrix}\boldsymbol{p}\\\boldsymbol{q}\end{pmatrix}=\boldsymbol{T}\begin{pmatrix}\boldsymbol{a}\\\boldsymbol{b}\end{pmatrix}=\begin{pmatrix}\boldsymbol{T}_{11} & \boldsymbol{T}_{12}\\\boldsymbol{T}_{21} & \boldsymbol{T}_{22}\end{pmatrix}\begin{pmatrix}\boldsymbol{a}\\\boldsymbol{b}\end{pmatrix}}$$

Fundamental property of the T-matrix is that it is dependent only on the scatterer shape and composition and does not depend on particle orientation or scattering direction. Once the T-matrix has been calculated it is relatively easy to evaluate S or Z matrix for particular scattering geometries ($\alpha,\beta,\gamma,n^{inc},n^{sca}$)

This means that the computational cost of evaluating scattering properties for orientation averaged particle is similar to that of single particle orientation.

T-matrix can be applied to arbitrarily shaped particles, but analytical and efficient methods exists only for special cases.
First attempt was the Extended Boundary Condition Method (EBCM) which is particularly efficient for **rotationally symmetric** shapes.



pytmatrix

Leinonen, J., High-level interface to T-matrix scattering calculations: architecture, capabilities and limitations, Opt. Express, vol. 22, issue 2, 1655-1660 (2014), doi: 10.1364/OE.22.001655.

Public, open source, python2 high level interface to the fortran77 T-matrix code (still public and open source) for homogeneous rotationally symmetric particles by M. Mishchenko
http://www.giss.nasa.gov/staff/mmishchenko/t_matrix.html

Installation:

```
(sudo) pip install pytmatrix
```

Installation on the provided VM is straightforward. For installation on other machines requirments are `numpy`, `scipy` and a fortran77 compiler such as `gfortran`

Modules

tmatrix – interface class to fortran77 T-matrix code
tmatrix_aux – auxiliary module defines useful constants and functions (radar wavelengths, shortcut to forward and backward polarized scattering geometry, axis ratio as a function of diameter)
refractive – dielectric properties of water and ice and effective medium approximations for mixtures
orientation – scattering properties of particles orientation averaged over specified pdf
psd – common pdf used to describe hydrometeors psd
tmatrix_psd – T-matrix for psd (deprecated)
scatter – compute single scattering properties (scattering cross sections, ldr, single scattering albedo ...)
radar – from scattering properties to radar variables (Z, Zdr, Ai, Kdp,ρ_{hv})

to load pytmatrix modules:

```python
from pytmatrix import module_name
```

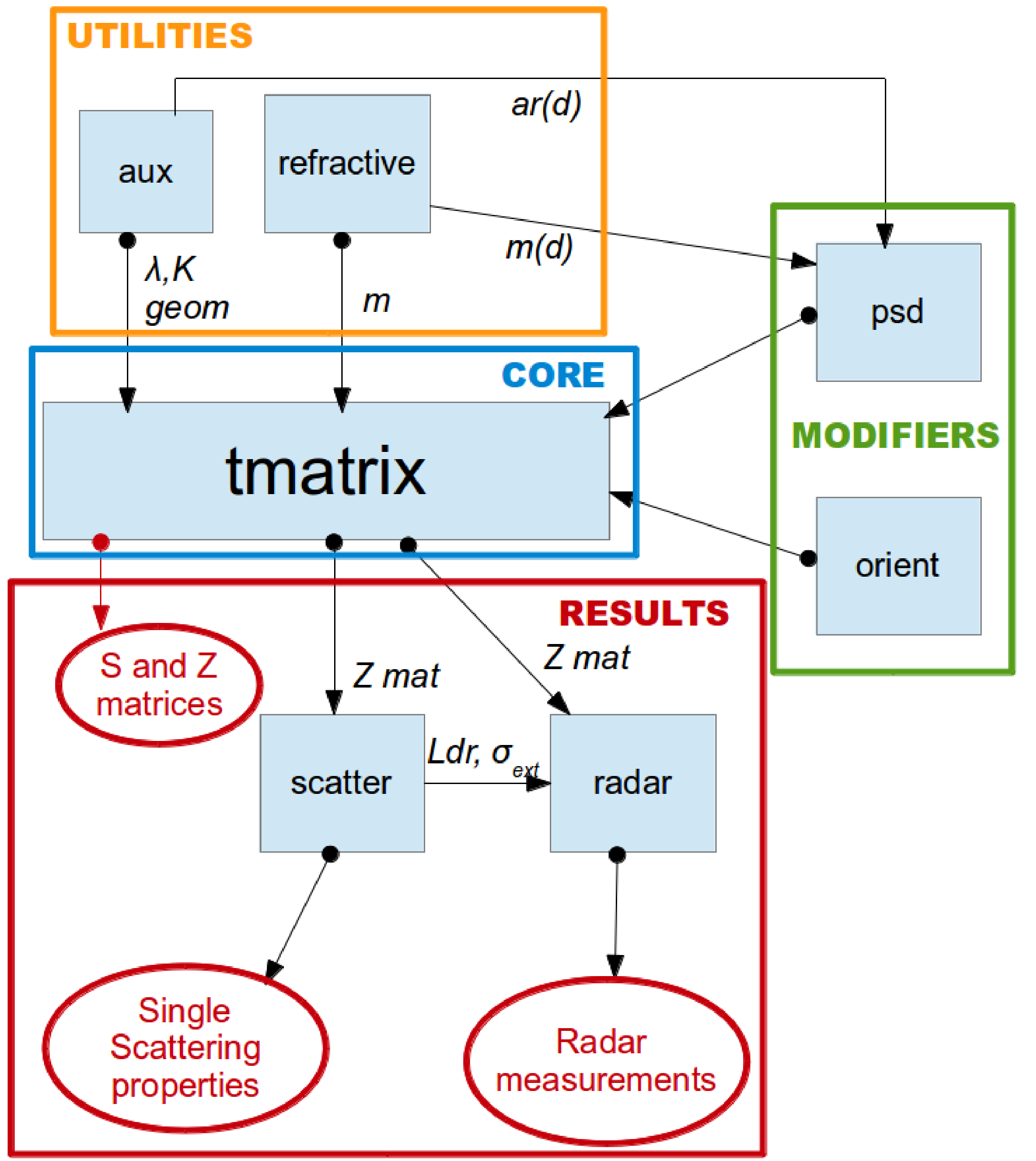
pytmatrix – module scheme

`tmatrix` is the core of the computation an provides S and Z matrix. Basic computation are for single particle in fixed orientation, but it is possible to use modifier modules (**`orientation`** and **`psd`**) to compute S and Z matrix for populations of particles and averages over distributions of orientations.

Finally **`scatter`** and **`radar`** modules uses the Z matrix to compute final products.

`tmatrix` recalculates the T-matrix if and only if the shape or the composition of the scatterer changes

It is also possible to store scattering lookup table to reuse scattering properties of a particular "family" of particles with different PSDs



pytmatrix – tmatrix module

Central module is **`tmatrix`**, which defines the scattering configuration and call the fortran77 routines. Its interface is the **`Scatterer`** class which manages many options. You have to know which options you need for your particular configuration

Attributes

radius: Equivalent radius. (default = 1)
radius_type: Scatterer.RADIUS_EQUAL_VOLUME (default), Scatterer.RADIUS_MAXIMUM Scatterer.RADIUS_EQUAL_AREA
wavelength: The wavelength of incident light (same units as axi, default =1).
m: The complex refractive index (default 2.0 +j0.0)
axis_ratio: The horizontal-to-rotational axis ratio (default 1.0)
shape: Scatterer.SHAPE_SPHEROID (default) Scatterer.SHAPE_CYLINDER
alpha, **beta**: The Euler angles of the particle orientation (degrees, default 0.0).
thet0, **thet**: The zenith angles of incident and scattered radiation (degrees, default 90.0)
phi0, **phi**: The azimuth angles of incident and scattered radiation (degrees, default 0 and 180)

Kw_sqr: The squared reference water dielectric factor for computing radar reflectivity (default 0.93)

orient: Function to use to compute the scattering properties. Should be one of the `orientation` module methods (orient_single (default), orient_averaged_adaptive, orient_averaged_fixed)
or_pdf: Particle orientation PDF for orientation averaging (default gaussian)
n_alpha: Number of integration points in the alpha Euler angle (default 5)
n_beta: Number of integration points in the beta Euler angle (default 10)

psd_integrator: Set this to a PSDIntegrator instance to enable size distribution integration (default None)
psd: Set to a callable object giving the PSD value for a given diameter

Once you have initiated a **`Scatterer`** object you can call its methods **`get_S()`** or **`get_Z()`**