Design des systèmes à concentration par le logiciel SOLSTICE

LABORATOIRE
PROCÉDÉS, MATÉRIAUX
et ENERGIE SOLAIRE
.UPR 8521 du CNRS.
conventionnée avec
l'université de Perpignan
PROCESSES,MATERIALS
and SOLAR ENERGY
LABORATORY

Cyril **CALIOT**[a] , Jean-Jacques **BEZIAN**[b], Stéphane **BLANCO**[c], Christophe **COUSTET**[d], Mouna **EL HAFI**[b], Vincent **EYMET**[d], Gilles **FLAMANT**[a], Vincent **FOREST**[d], Richard **FOURNIER**[c], Benjamin **GRANGE**[a], Benjamin **PIAUD**[d]

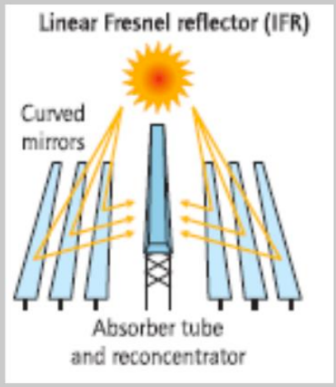
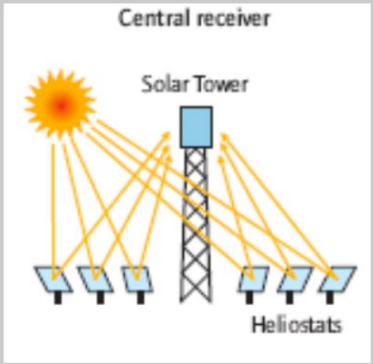
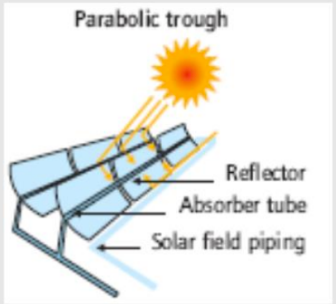
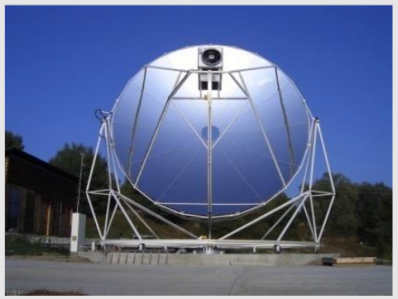
[a]Laboratoire PROcédés, Matériaux et Energie Solaire, PROMES-CNRS, Odeillo

[b] Centre RAPSODEE, CNRS, IMT Mines Albi, Université Fédérale Toulouse Midi Pyrénées, Albi

[c] Laboratoire Plasma et Conversion de l'Energie, LAPLACE, Université Paul Sabatier, Toulouse

[d] MESO-STAR SAS, www.meso-star.com, Longages

Contexte

Concentration / Réception	Linéaire (30-100 soleils) 300°C-500°C	Ponctuelle (500-10 000 soleils) 500°C-1500°C
Fixe		
Mobile		

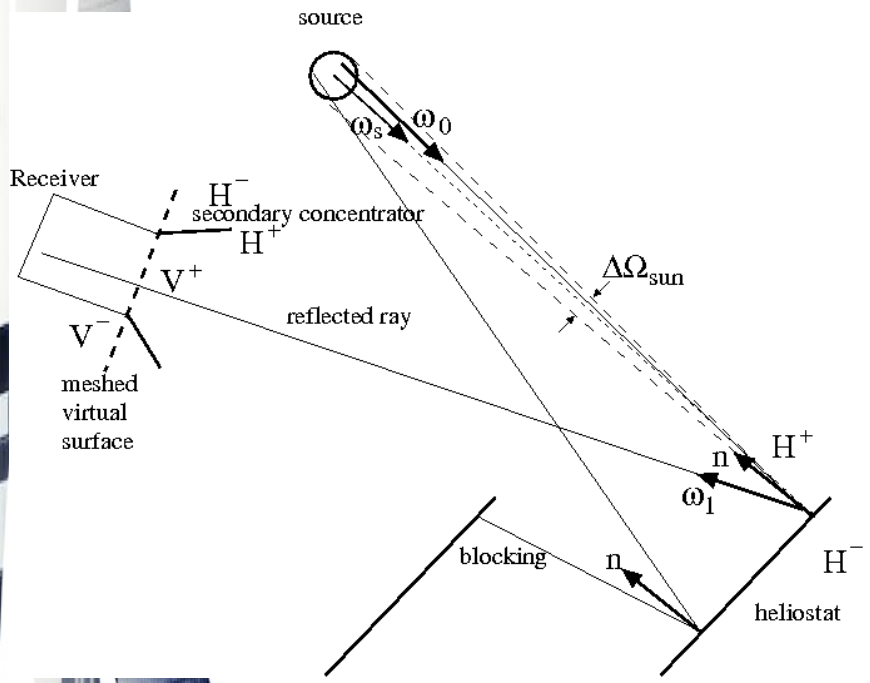
contact: cyril.caliot@promes.cnrs.fr

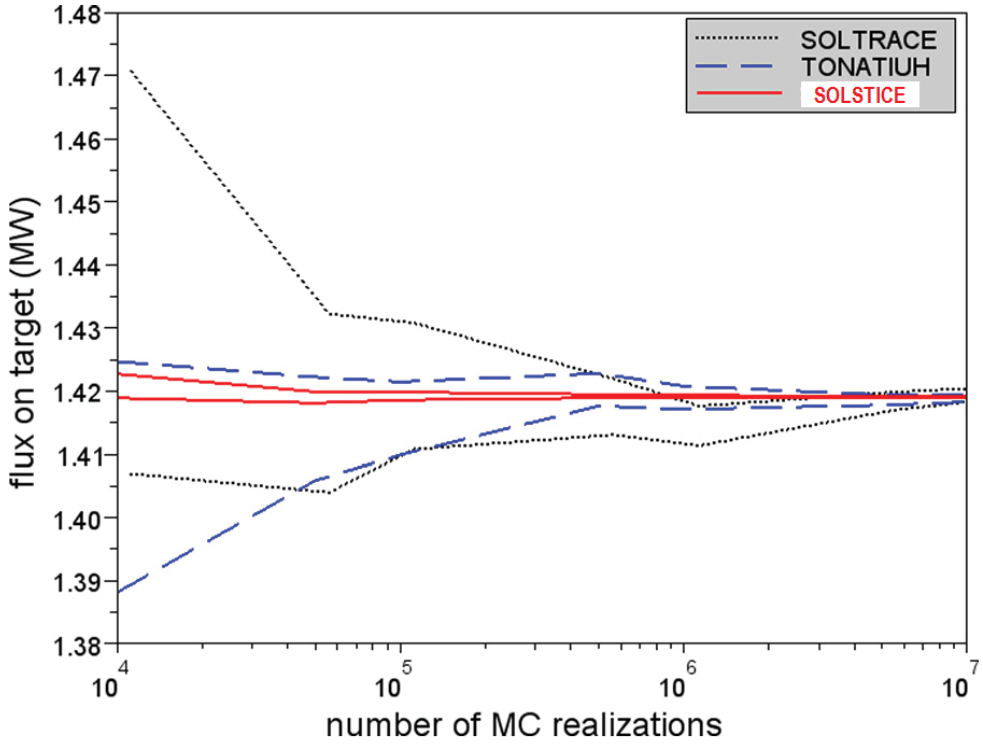
Objectives

- **Prediction of the concentrated solar fluxes distribution** in solar facilities
  - **To simulate** all types of **concentrating optics**
  - **Design and study** of reflectors and receivers
    - Compute the **optical efficiency** of reflectors
    - Compute the **flux maps** on surfaces
      - For optimization of receiver geometry (CAD geometry) by combining heat and CFD analysis
      - For thermomechanical stress studies
  - To optimize the **aiming strategy**
  - To **characterize** the heliostat canting errors

contact: cyril.caliot@promes.cnrs.fr

Comparison with SOA software





The Integral Formulation used to develop the SOLSTICE algorithm leads to higher convergence rate and faster runtime than State-Of-the-Art software

Flux on Themis target in function of the number of MC realizations. Two sets of curves are plotted: flux+standard deviation and flux-standard deviation

- C. Caliot, H. Benoit, E. Guillot, B. Piaud, C. Coustet, J.-L. Sans, A. Ferrière, G. Flamant. *Validation of a Monte Carlo integral formulation applied to solar facility simulations and use of sensitivities*. **ASME Journal of Solar Energy Engineering, 2015,** vol 137(2), p 021019.
- J. Delatorre, G. Baud, J.J. Bézian, S. Blanco, C. Caliot, J.F. Cornet, C. Coustet, J. Dauchet, M. El Hafi, V. Eymet, R. Fournier, J. Gautrais, O. Gourmel, D. Joseph, N. Meilhac, A. Pajot, M. Paulin, P. Perez, B. Piaud, M. Roger, J. Rolland, F. Veynandt, S. Weitz, *Monte-Carlo advances and concentrated solar applications*, **Solar Energy**, **2013,** vol. 103, p.653-681
- J. Delatorre. Calculs de sensibilités par la Méthode de Monte-Carlo pour la conception de procédés à énergie solaire concentrée. **Thèse 2011.** JJ Bezian, RAPSODEE, Mines d'Albi.

contact: cyril.caliot@promes.cnrs.fr

cnrs dépasser les frontières

|Méso|Star> Understanding Complex Systems

PRESENTATION

SOLar Simulation Tool In ConcEntrating optics SOLSTICE

- Collaborators of the **SOLSTICE** project:
  - **PROMES-CNRS:** commissioners **(LABEX SOLSTICE)***
  - **Meso-Star SAS:** developers

*This software was funded by the French "Investments for the future" program managed by the National Agency for Research under contract ANR-10-LABX-22-01-SOLSTICE

contact: cyril.caliot@promes.cnrs.fr

OVERVIEW

- **OPEN SOURCE** and **FREE** GPL GNU license
- **OS**: Linux, Windows

- Run in **batch mode** (command-line software)
- **Parallel** (shared-memory)
- **Visualization** with ParaView

INPUT	OUTPUT
YAML files	**VTK** files

contact: cyril.caliot@promes.cnrs.fr

INPUT

- Need to define:
  - The Sun **DNI** and atmospheric **extinction**
  - The **optical properties** of materials (spectra)
  - The **geometry** of surfaces (mirrors, target)
  - The solar **tracking mode** (single or dual-axis with offset)

- Need to create **Entities** like:
  - *Target* or *receiver*
  - *Heliostat* (includes materials, surfaces and the tracking)
  - *Parabolic-trough module*, etc.

contact: cyril.caliot@promes.cnrs.fr

OUTPUT

- Spatial **flux distribution** (maps) on the receivers
  - Incident flux; Absorbed flux
- Details of **losses**:
  - Cosine factor; Shadowing and Blocking; Spillage
  - Material absorption; Atmospheric extinction
- **Geometry** export
- Colored **ray** paths
- **Image** rendering of the scene

contact: cyril.caliot@promes.cnrs.fr

cyl.yaml

```yaml
# BEGINNING OF YAML
########################################
#
########################################
# Sun & Atmosphere
########################################
# Only the Direct Normal Irradiation (DNI) from the sun is set.
# As an example here, no solar spectrum is considered,
# but we also could define a solar intensity spectrum.
- sun: &sun
    dni: 1000

########################################
# Materials
########################################

# Define a material purely absorbant (e.g. for pylons)
- material: &occultant
    front: {matte: {reflectivity: 0}}
    back:  {matte: {reflectivity: 0}}

# Define a specular material for the reflectors
- material: &specular
    front: {mirror: {reflectivity: 0.9, roughness: 0}}
    back:  {matte:  {reflectivity: 0.0}}

# Define a diffusely reflecting material for the receiver
- material: &lambertian
    front: {matte: {reflectivity: 0.06}}
    back:  {matte: {reflectivity: 0.0}}

# Define a refractive interface between air and glass
- material: &refractive_out
    front:
      dielectric:                                                  # --> Front|Back
        medium_i: &air { refractive_index: 1, absorption: 0 }      # medium travelling in before hitting Front
        medium_t: &glass { refractive_index: 1.5, absorption: 0.02 }  # medium of refraction
    back:
      dielectric:           # Front|Back <---
        medium_i: *glass    # medium travelling in before hitting Back
        medium_t: *air      # medium of refraction

# Define a refractive interface between glass and vacuum
- material: &refractive_in
    front:
      dielectric:                                                  # --> Front|Back
        medium_i: *glass                                           # medium travelling in before hitting Front
        medium_t: &vacuum { refractive_index: 1, absorption: 0 }   # medium of refraction
    back:
      dielectric:           # Front|Back <---
        medium_i: *vacuum   # medium travelling in before hitting Back
        medium_t: *glass    # medium of refraction

```

Define the DNI

Define Materials

- Absorbing
- Reflecting
- Refracting

contact: cyl.saint@promes.cnrs.fr

cyl.yaml

```yaml
########################################
# Geometries
########################################

# Define the geometry of the parabolic-cylinder mirror surface
- geometry: &reflector
  - material: *specular
    transform: {translation: [0, 0, 0], rotation: [0, 0, 0]}
    parabolic-cylinder:
      focal: 1.73
      clip:
        - operation: AND
          vertices: [[-6, -2.88], [-6, 2.88], [ 6, 2.88], [ 6, -2.88]]
      slices: 32
# Define the tube surface  of the receiver
- geometry: &tube_steel_g
  - material: *lambertian
    transform: {translation: [0, 0, 0], rotation: [0, 90, 0]}
    cylinder: {height: 12, radius: 0.07, slices: 128, stacks: 32 }
# Define the tube surface of the internal glass interface
- geometry: &tube_glass_in_g
  - material: *refractive_in
    transform: {translation: [0, 0, 0], rotation: [0, 90, 0]}
    cylinder: {height: 12, radius: 0.119, slices: 256, stacks: 32 }
# Define the tube surface of the external glass interface
- geometry: &tube_glass_out_g
  - material: *refractive_out
    transform: {translation: [0, 0, 0], rotation: [0, 90, 0]}
    cylinder: {height: 12, radius: 0.125, slices: 256, stacks: 32 }
# Import the support structure from CAD in STL format
- geometry: &structure_g
  - material: *lambertian
    transform: {translation: [5, 0, 1.73], rotation: [0, -90, -90]}
    stl: {path: support_structure.stl }
```

Define the mirror

Define the tubes

Import STL

contact: cyril.caliot@promes.cnrs.fr

cyl.yaml

```
113 ########################################
114 # Template of the parabolic trough (includes tubes and mirror)
115 ########################################
116 # Create a template to link the tracking (pivot) with the surfaces of the parabolic-trough geometry
117 - template: &parab_trough_t
118     name: pt_pivot
119     transform: {translation: [0, 0, 0],rotation: [0, 0, 0]}
120     x_pivot:
121       target: { sun: *sun }
122       ref_point: [0, 0, 0]
123     children:
124 # Include the reflector
125     - name: pt_reflector
126       primary: 1
127       transform: { rotation: [0, 0, 0]}
128       geometry: *reflector
129 # Include the glass tube internal interface
130     - name: tube_glass_in
131       primary: 0
132       transform: { translation: [0, 0, 1.73], rotation: [0, 0, 0]}
133       geometry: *tube_glass_in_g
134 # Include the glass tube external interface
135     - name: tube_glass_out
136       primary: 0
137       transform: { translation: [0, 0, 1.73], rotation: [0, 0, 0]}
138       geometry: *tube_glass_out_g
139 # Include the receiver tube
140     - name: tube_steel
141       primary: 0
142       transform: { translation: [0, 0, 1.73], rotation: [0, 0, 0]}
143       geometry: *tube_steel_g
144 # Import STL geometry
145     - name: structure
146       primary: 0
147       transform: {translation: [0, 0., 0.0], rotation: [0, 0, 0]}
148       geometry: *structure_g
```

Create a template for the PT module

Include the pivot (tracking)

Attach the reflector and the tubes (glass & steel)

Attach the STL geometry

cyl.yaml

```
149 ########################################
150 # Entities defined to appear in the scene to simulate
151 ########################################
152 # Define the parabolic-trough module and locate it inside the scene
153 - entity:
154     name: parab_trough_e
155     transform: {translation: [0, 0, 0], rotation: [0, 0, 0]}
156     children: [*parab_trough_t]
157 # Define another one in another location
158 - entity: {name: pt_2_e, children: [*parab_trough_t], transform: {translation: [0, 10, 0]}}
159 ########################################
160 # END OF YAML
161 ########################################
```

Add entity to the scene

Instantiation of PT module

cyl-rcv.yaml

```
1 ########################################
2 # Receivers
3 ########################################
4 - {name: parab_trough_e.pt_pivot.tube_steel, side: FRONT_AND_BACK, per_primitive: 1}
5 - {name: pt_2_e.pt_pivot.tube_steel, side: FRONT_AND_BACK, per_primitive: 1}
```

Define the receivers

Launch batch command « solstice »

```
solstice -D270,40 -n 1000000 -R cyl-rcv.yaml -fo simul cyl.yaml
solstice -D270,40 -g format=obj:split=geometry -fo geom cyl.yaml
solstice -D270,40 -q -n 10 -R cyl-rcv.yaml -p default cyl.yaml > solpaths
```

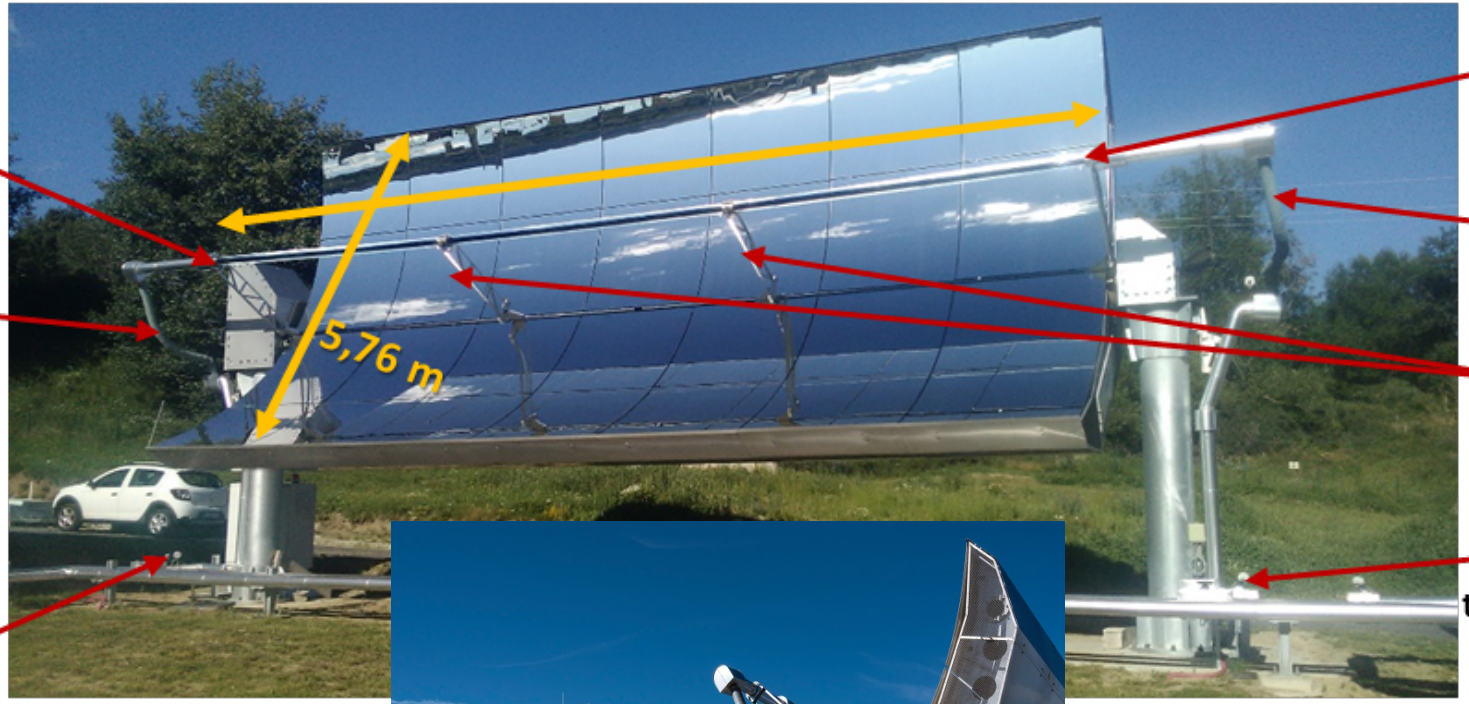
- To run the simulation (10^6 bundle of rays)
- To export the geometry for visualization
- To draw the ray paths

contact: cyril.caliot@promes.cnrs.fr

|Méso|Star>
Understanding Complex Systems

Example

A **parabolic-trough** collector installed at CNRS laboratory in Odeillo: **MicroSol-R** (3 modules, 50 kW_{th})

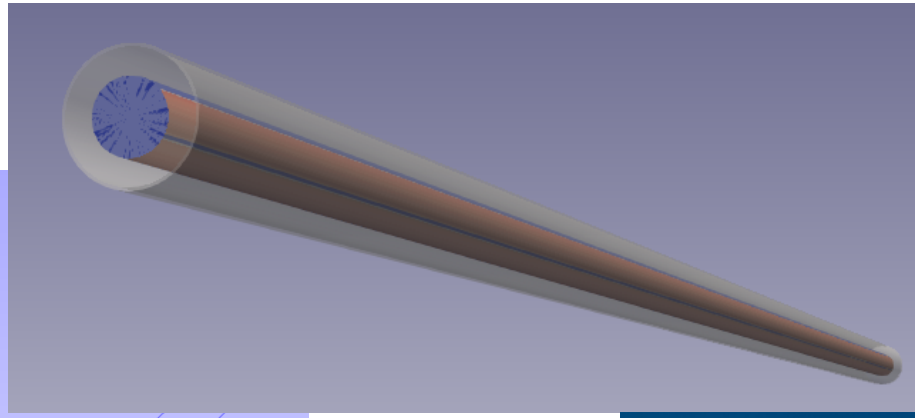


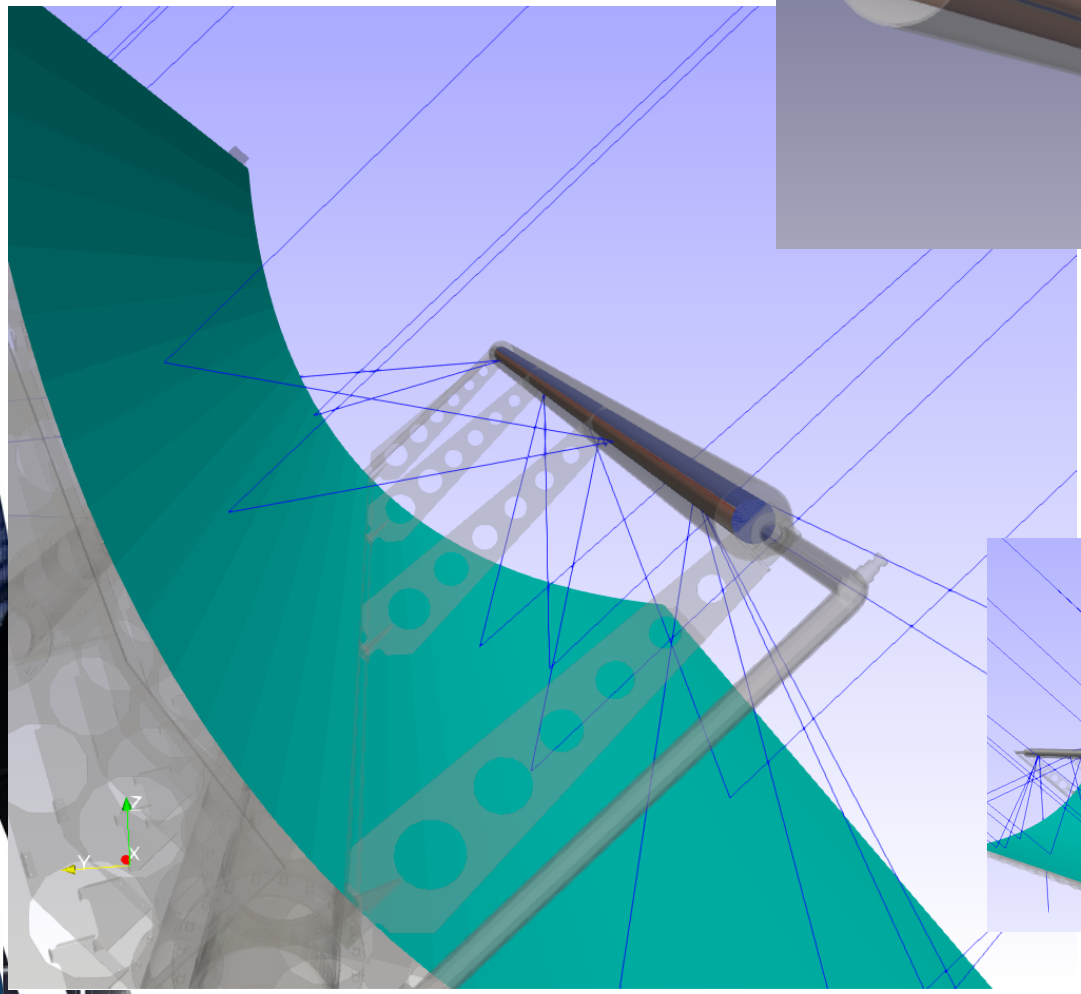
contact: cyril.caliot@promes.cnrs.fr

|Méso|Star>
Understanding Complex Systems

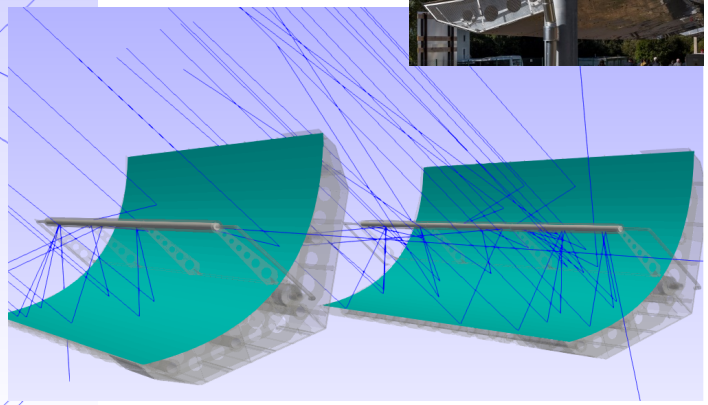
Visualization in ParaView

- PT module, tubular receiver and glass





MicroSol-R module

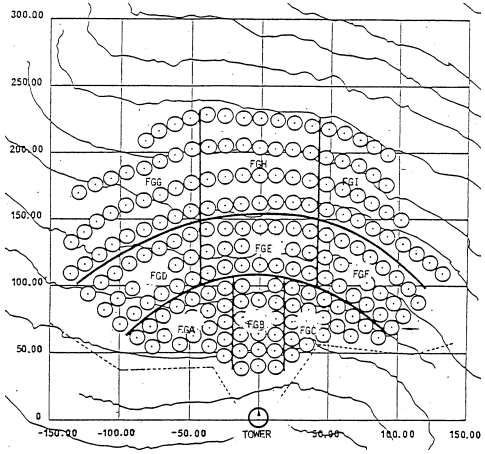


contact: cyril.caliot@promes.cnrs.fr

Research and development center THEMIS

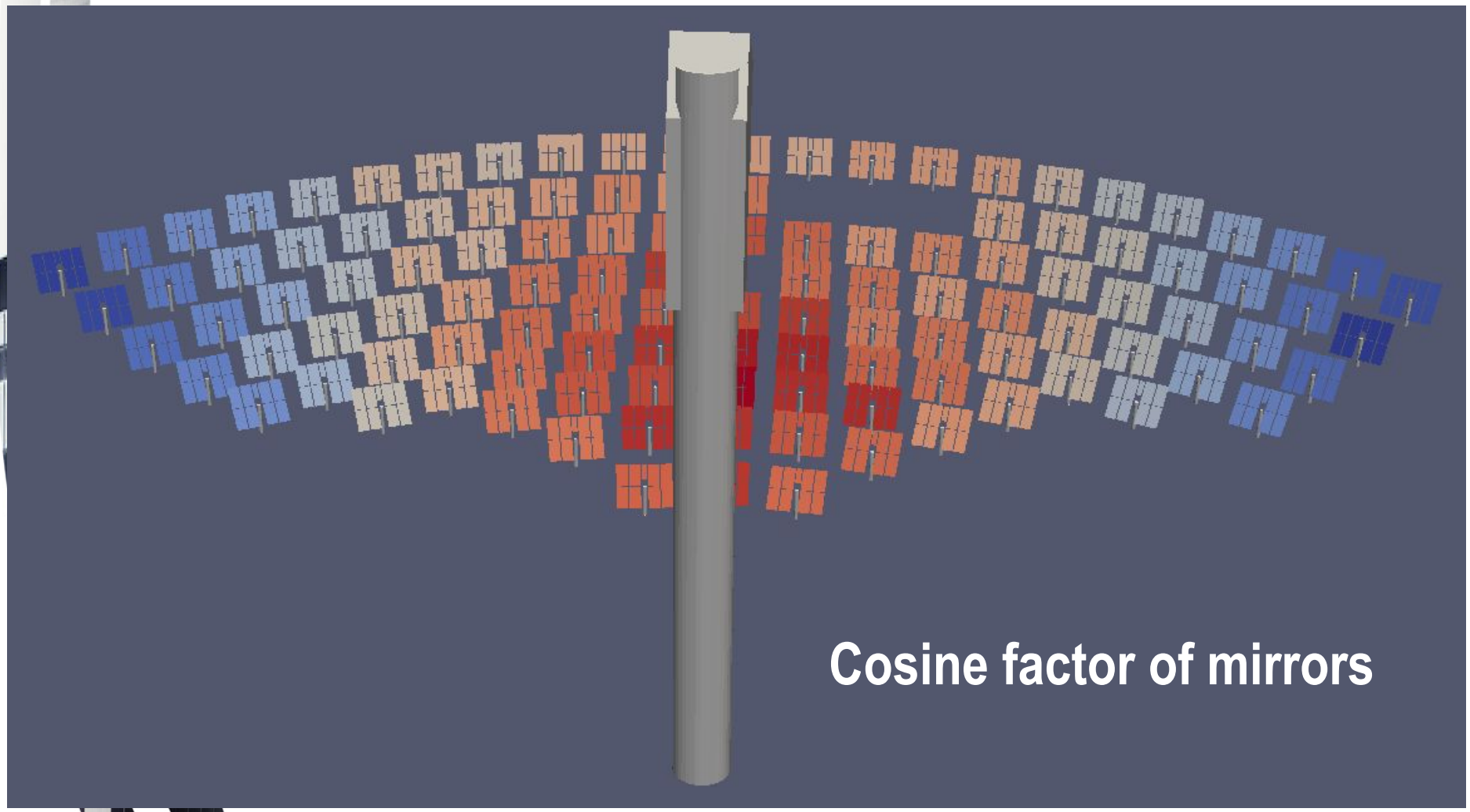
5 MW_{th}
1.2 MW_e

107 heliostats



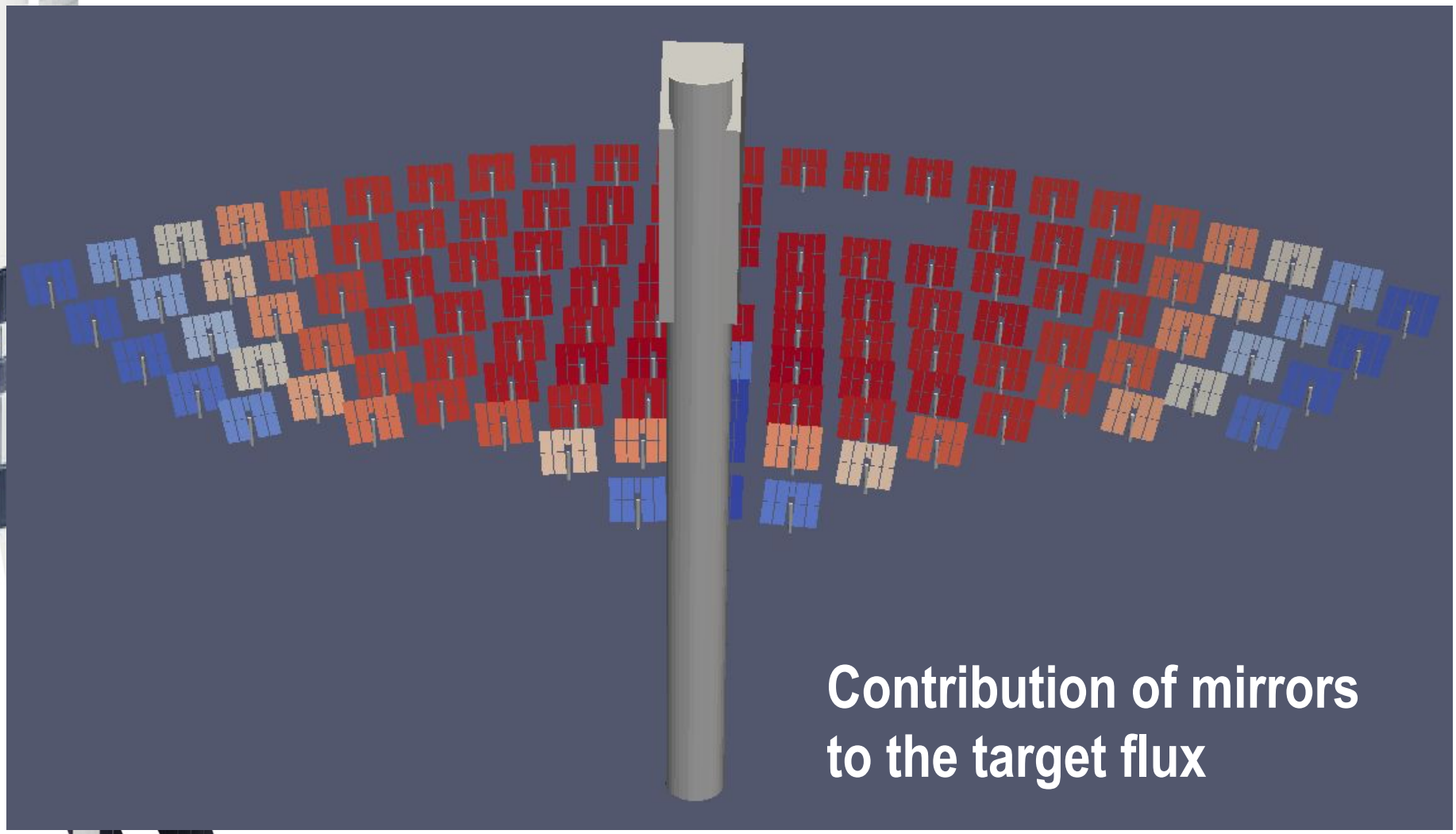
contact: cyril.caliot@promes.cnrs.fr

Advanced post-processing



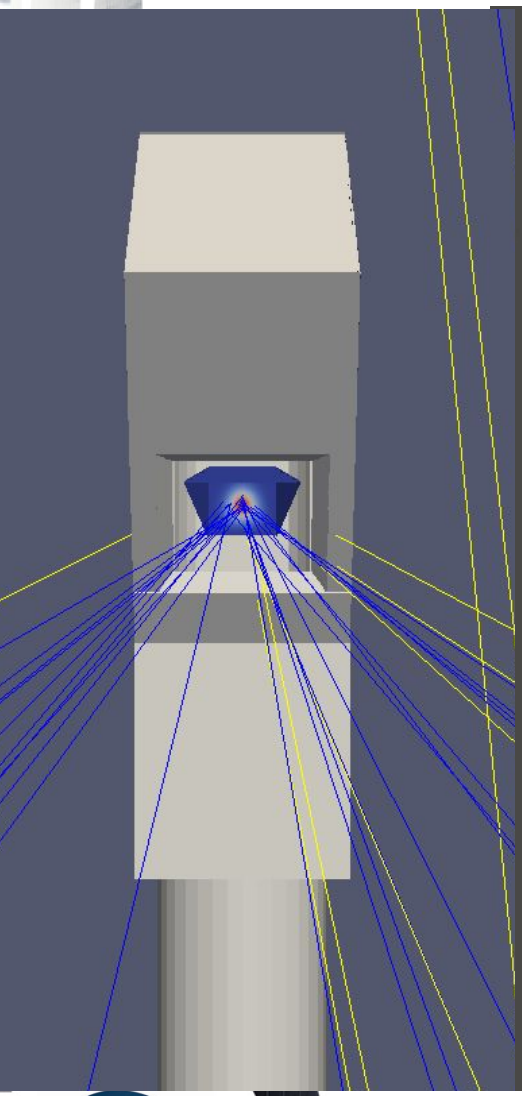
contact: cyril.caliot@promes.cnrs.fr

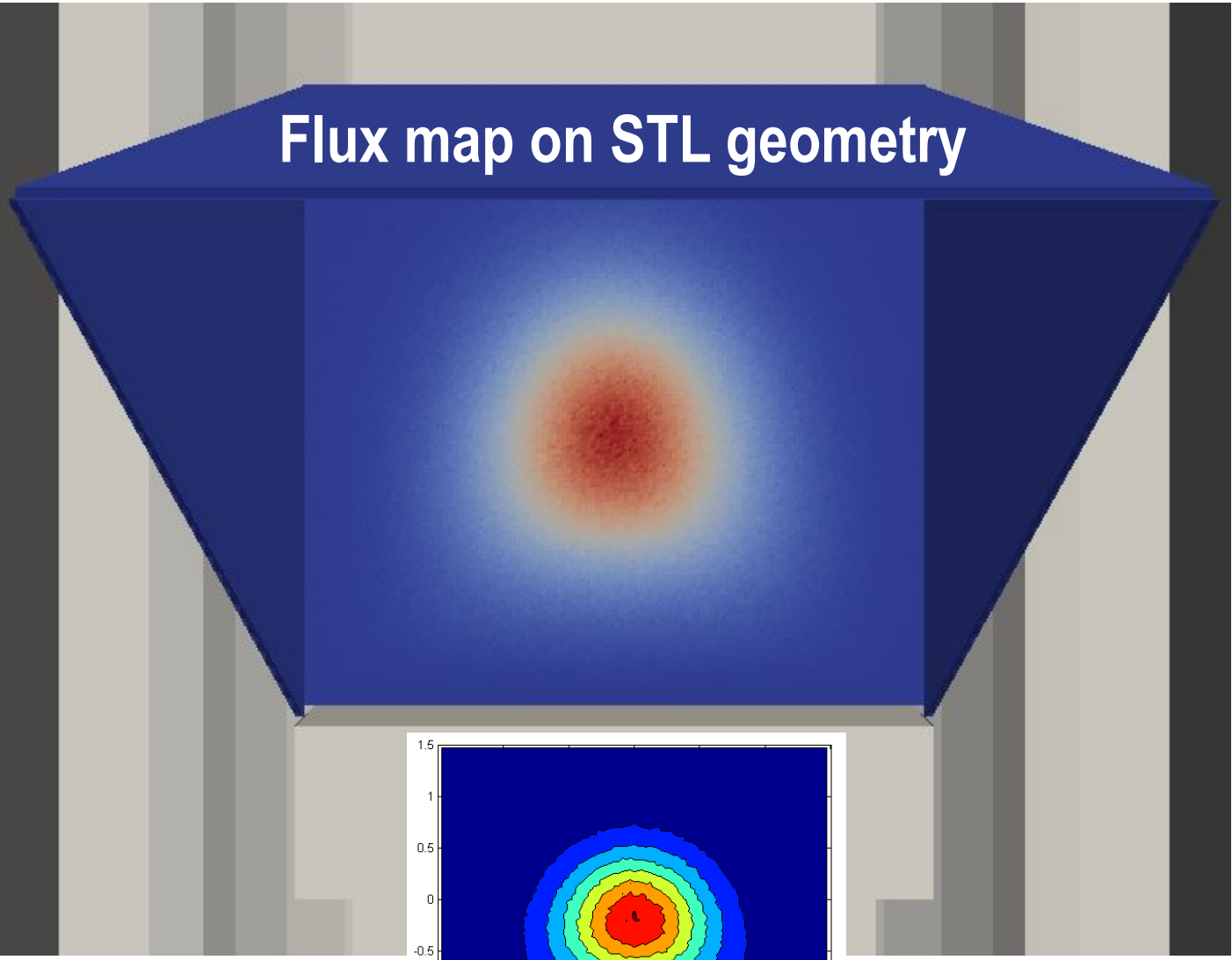
Advanced post-processing

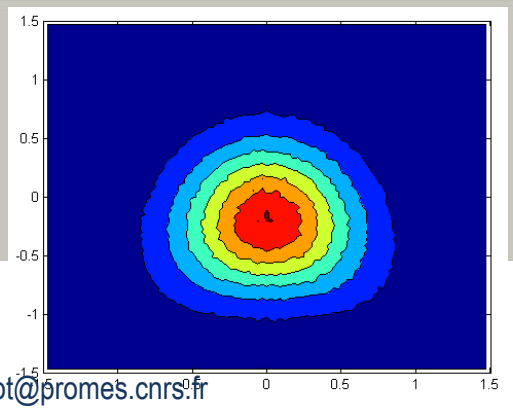


contact: cyril.caliot@promes.cnrs.fr

Advanced post-processing



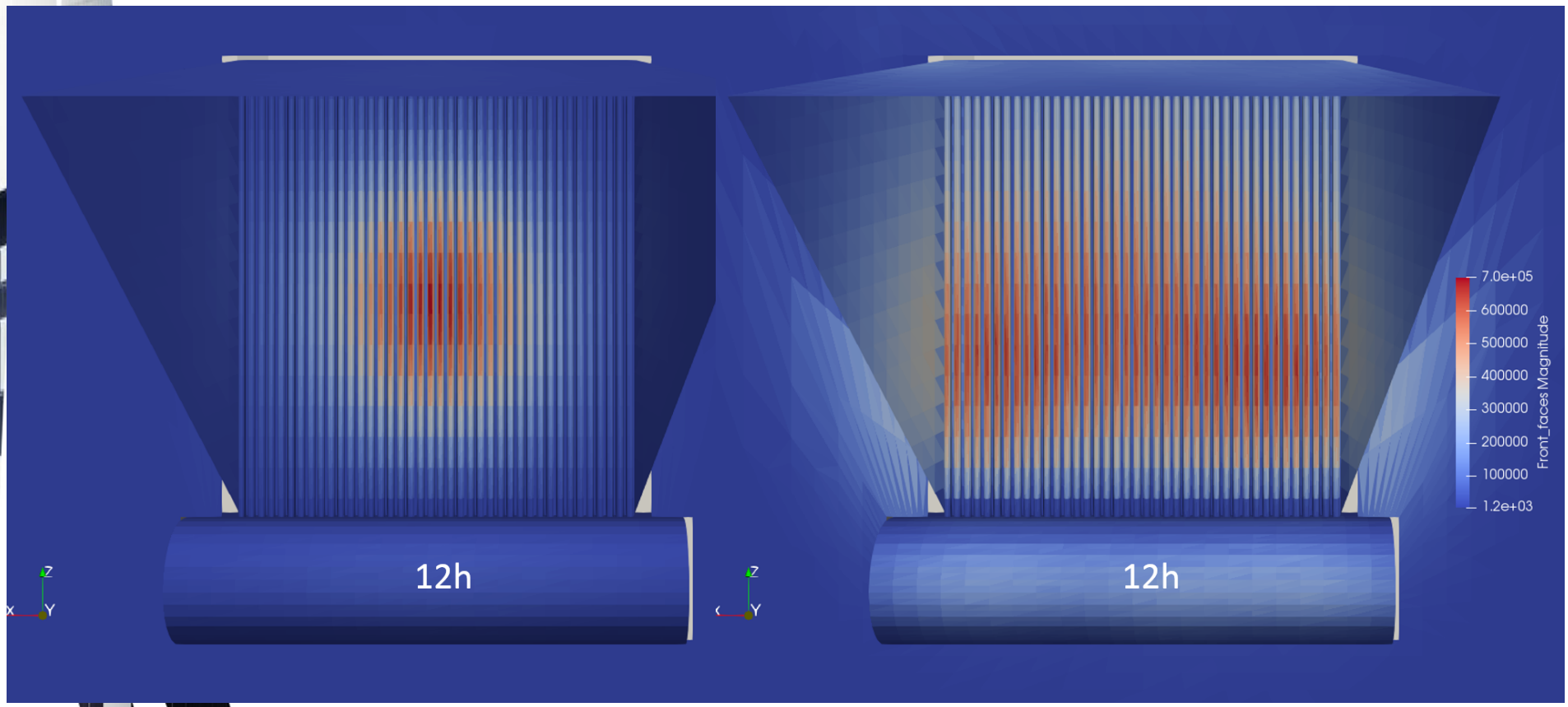




contact: cyril.caliot@promes.cnrs.fr

Validation of aiming point strategy

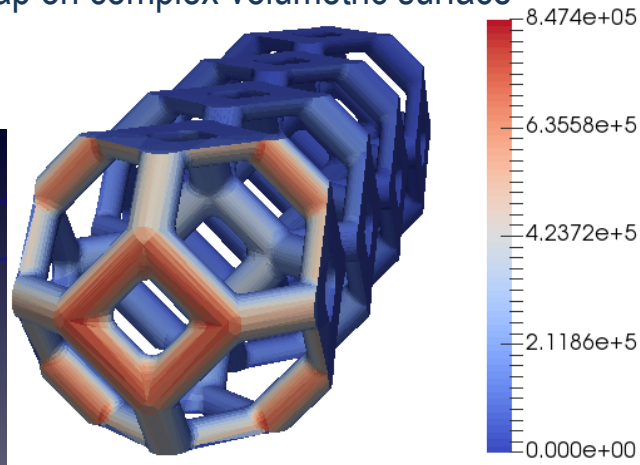
Tubular receiver of the Next-CSP EU project at Themis



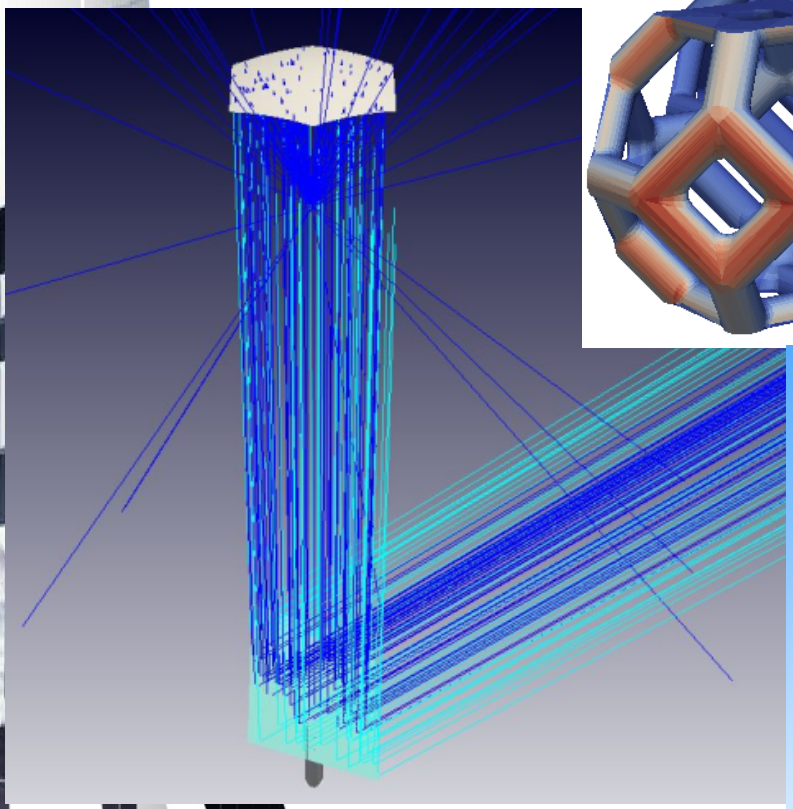
contact: cyril.caliot@promes.cnrs.fr

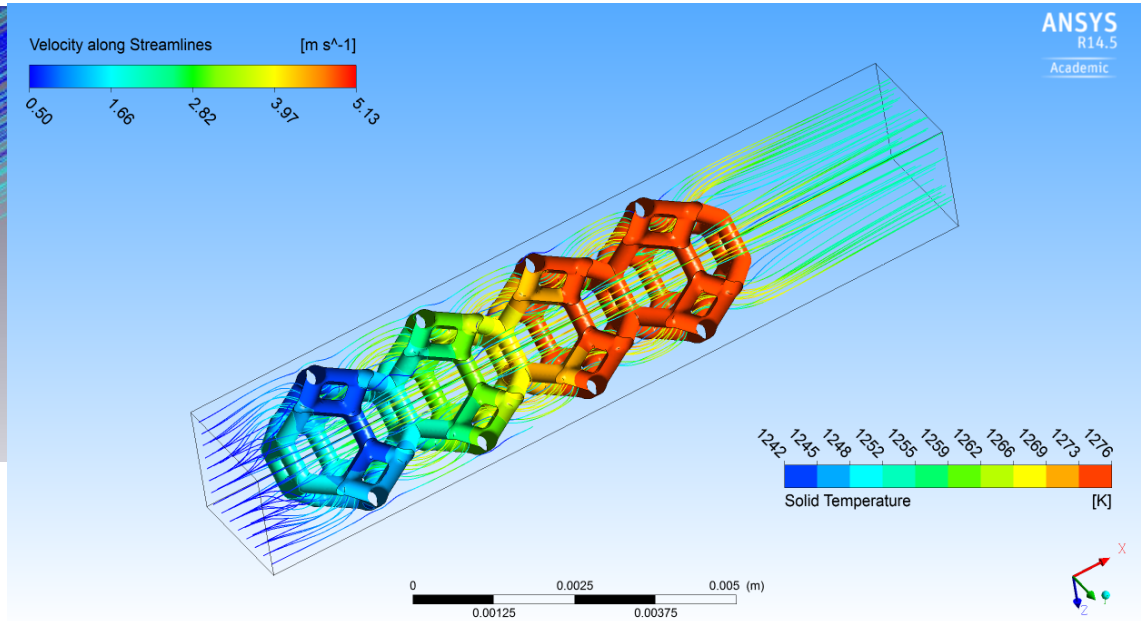
Combined modelling of high temperature volumetric receiver

Flux map on complex volumetric surface



V_{in} = 0.5 m/s ; d_c = 2 mm ; d_b = 0.3 mm
Fluid: Streamlines colored by the air velocity
Solid: Struts colored by its temperature



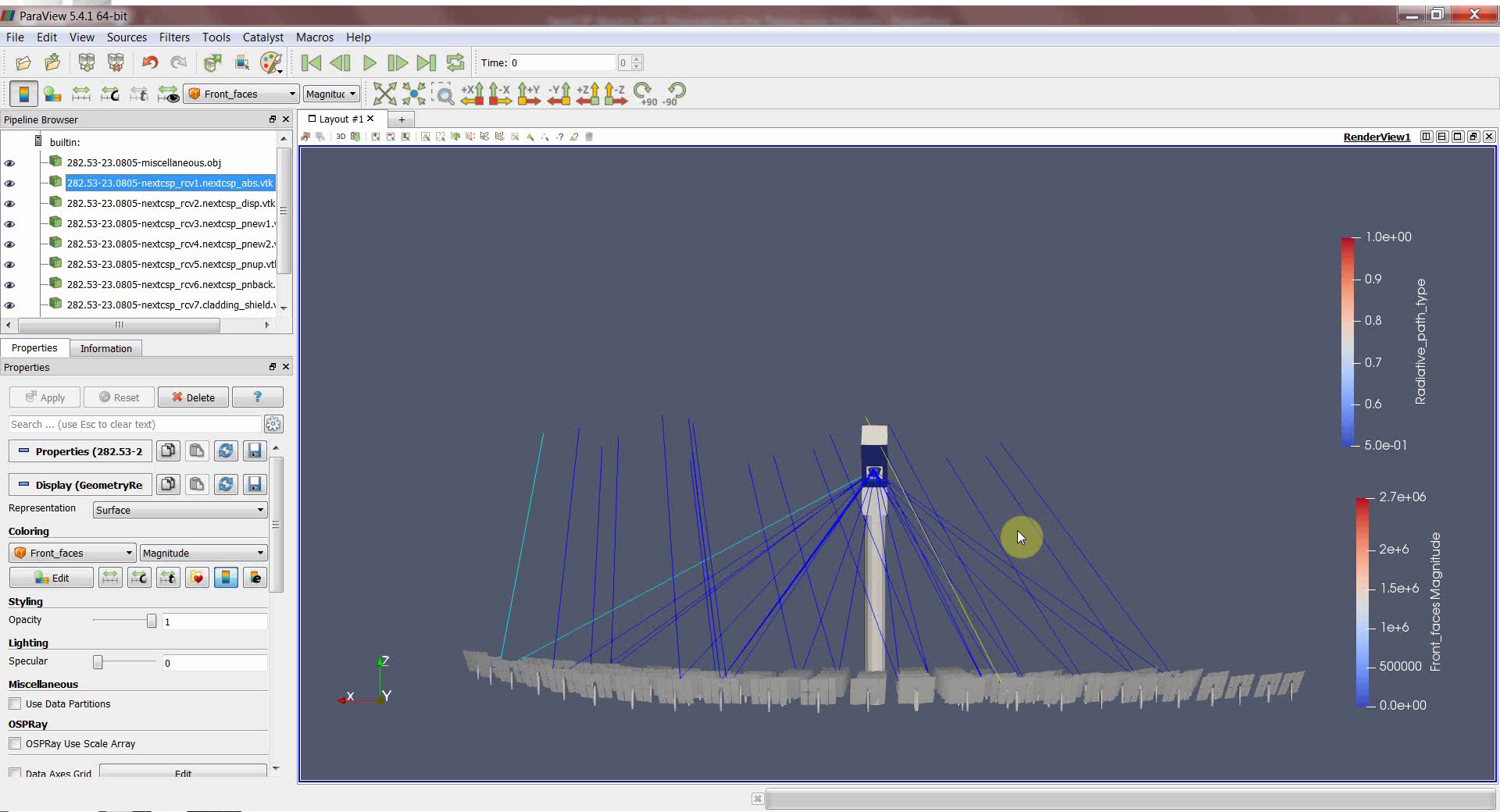


contact: cyril.caliot@promes.cnrs.fr

|Méso|Star>
Understanding Complex Systems

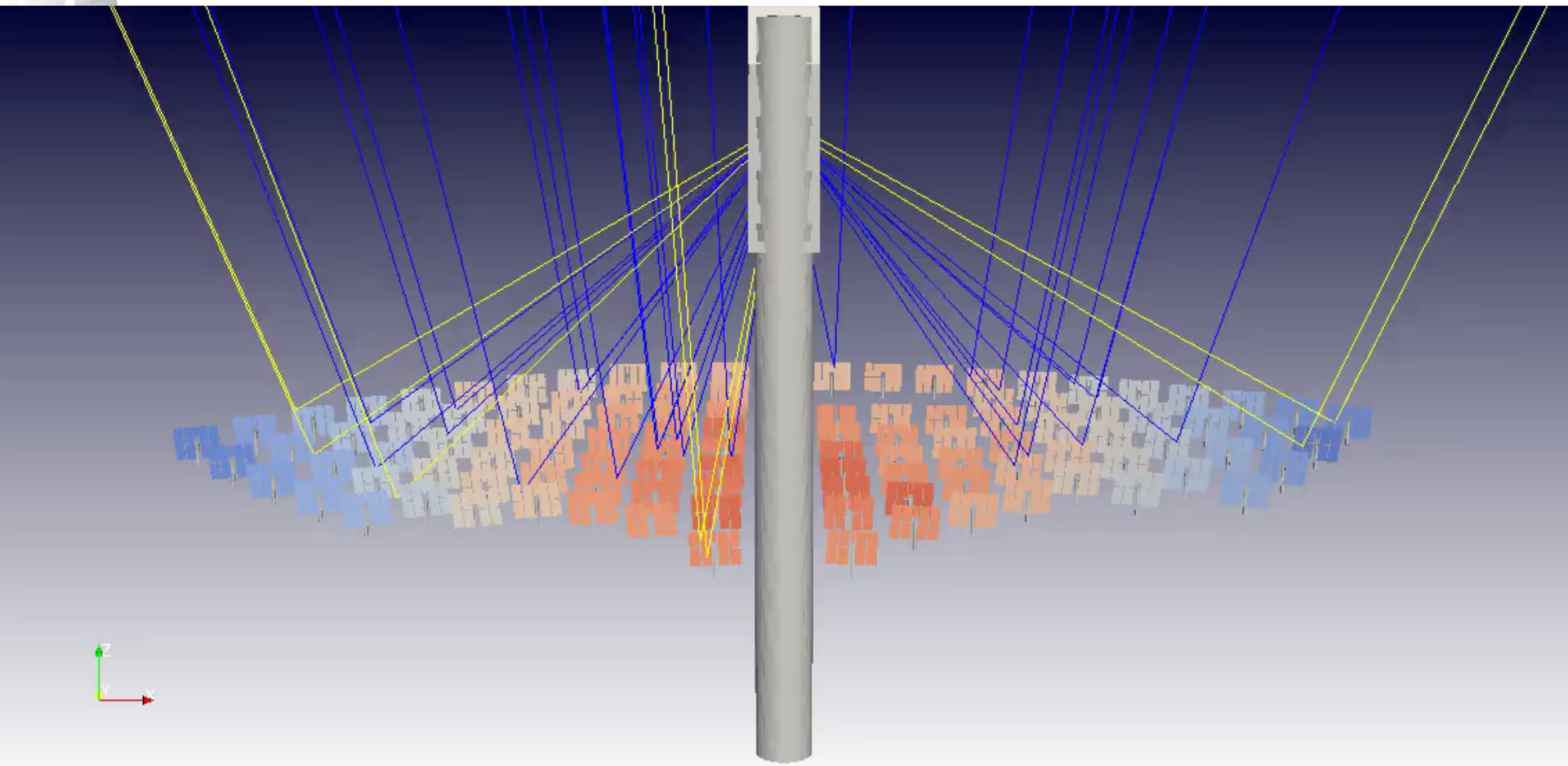
Thank you for your
attention

Post-processing demo of SOLSTICE Software



contact: cyril.caliot@promes.cnrs.fr

Themis (cosine factor of mirrors)



Rays

Blue: intersect the receiver

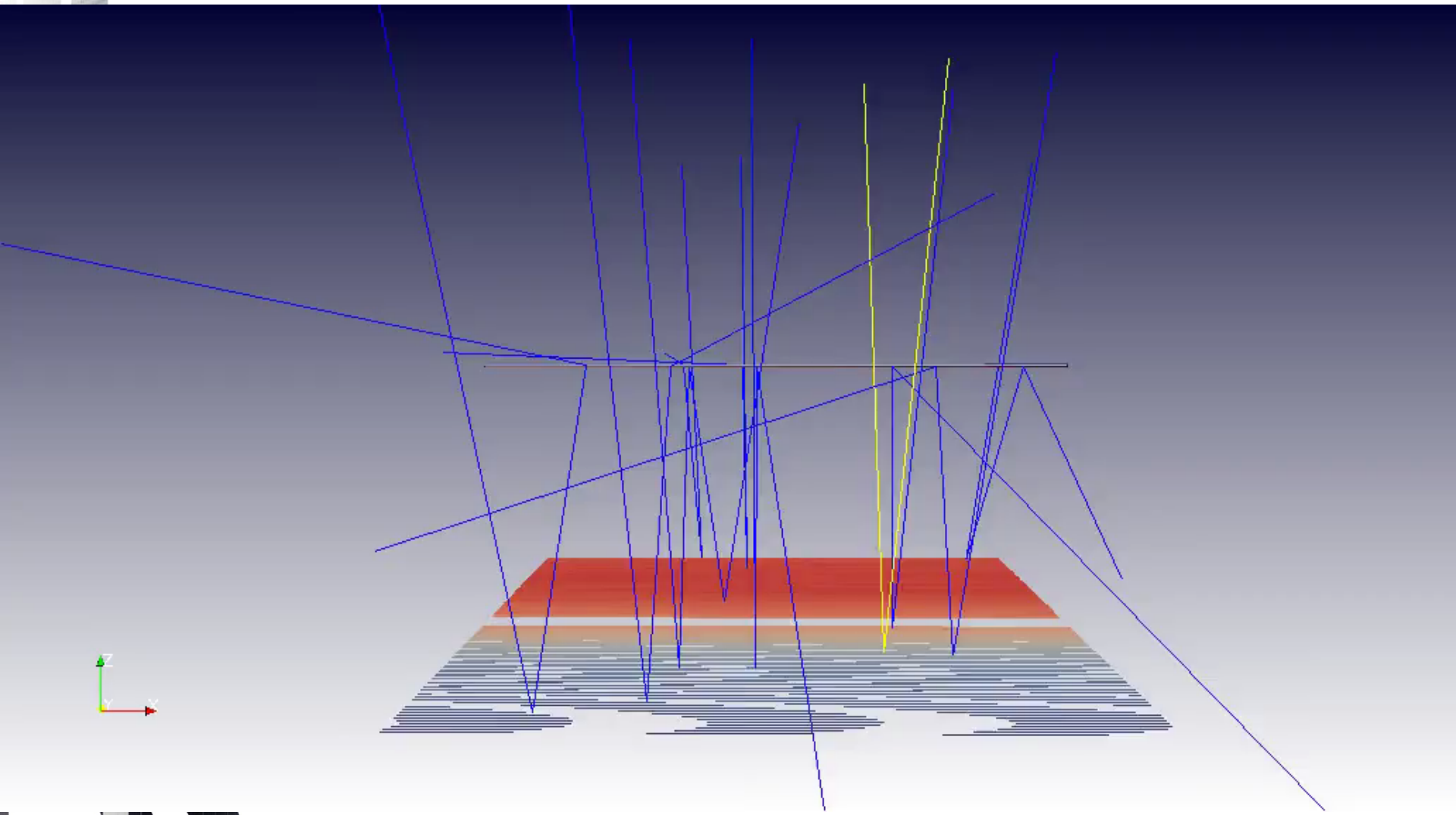
Yellow: miss the receiver

Red: shadowed by an obstacle

contact: cyril.caliot@promes.cnrs.fr

|Méso|Star>
Understanding Complex Systems

Linear Fresnel module (cosine factor of mirrors)



Rays

Blue: intersect the receiver

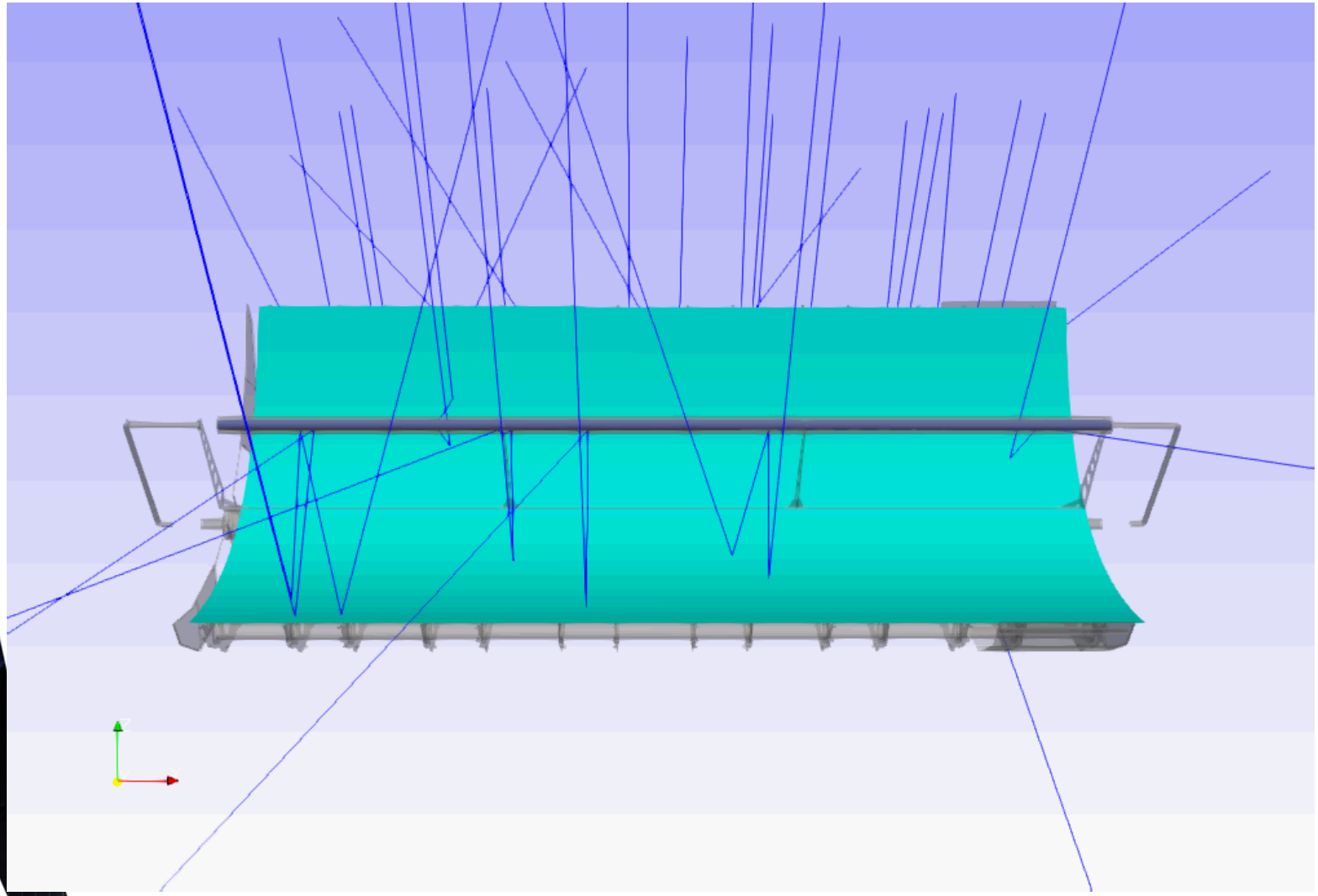
Yellow: miss the receiver

Red: shadowed by an obstacle

contact: cyril.caliot@promes.cnrs.fr

|Méso|Star>
Understanding Complex Systems

Parabolic-trough MicroSol-R (Odeillo, France)



contact: cyril.caliot@promes.cnrs.fr