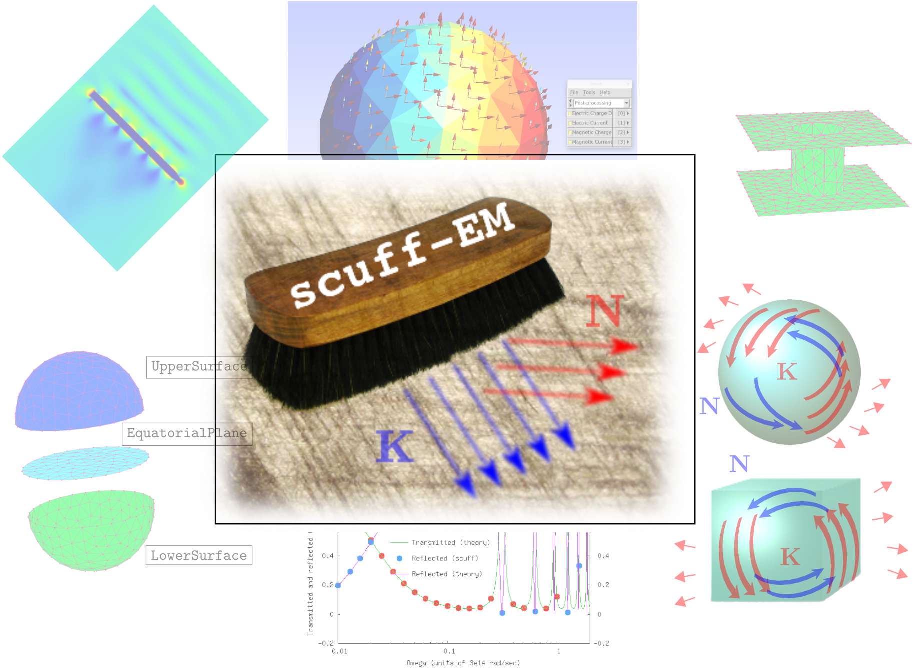

scuff-EM
N
K
UpperSurface
EquatorialPlane
LowerSurface
File Tools Help
Post-processing
Electric Charge D [0]
Electric Current [1]
Magnetic Charge [2]
Magnetic Current [3]
Transmitted (theory)
Reflected (scuff)
Reflected (theory)
Omega (units of 3e14 rad/sec)

The one-slide compactification of this talk

Surface-
CUrrent /
Field
Formulation
| of
Electro-
Magnetism

SCUFF-EM is a free, open-source software implementation of the boundary-element method of electromagnetic scattering.

SCUFF-EM supports a wide range of geometries, including compact scatterers, infinitely extended scatterers, and multi-material junctions.

The SCUFF-EM suite includes 8 standalone application codes for specialized problems in EM scattering, fluctuation physics, and RF engineering.

The SCUFF-EM suite also includes a core library with C++ and PYTHON APIs for designing homemade applications.

A couple of announcements before we begin...

A couple of announcements before we begin...

Steven
Could
Use
Four
Freshmen
|
Each
Monday

A couple of announcements before we begin...

Steven
Could
Use
Four
Freshmen
|
Each
Monday

Somebody,
Could
U
Feed
Frank
|
Every
Month?

A couple of announcements before we begin...

Steven
Could
Use
Four
Freshmen
|
Each
Monday

Somebody,
Could
U
Feed
Frank
|
Every
Month?

S
C insert
U your
F best
F acronym
|
E here...
M

Outline

1. A quick review of the Boundary-Element Method

2. A brief history of the evolution of SCUFF-EM

3. **What SCUFF-EM can do**

    1. Inputs: The geometries, materials, incident fields that SCUFF-EM can handle
    2. Outputs: The various calculations that SCUFF-EM can do
    3. Mechanics: How to run SCUFF-EM

4. Under the hood: How SCUFF-EM works

Outline

1. A quick review of the Boundary-Element Method

2. A brief history of the evolution of SCUFF-EM

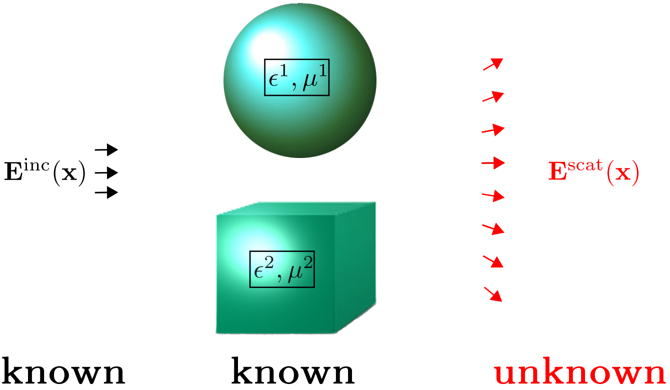
3. **What SCUFF-EM can do**
    1. Inputs: The geometries, materials, incident fields that SCUFF-EM can handle
    2. Outputs: The various calculations that SCUFF-EM can do
    3. Mechanics: How to run SCUFF-EM

4. Under the hood: How SCUFF-EM works

Quick Review of the Boundary-Element Method

Start by considering a general electromagnetic scattering problem.

We have some known **incident field** (such as a plane wave), scattering from some known **geometry** (including objects of known shapes and materials) and we want to know the scattered fields. (Note: all quantities $\sim e^{-i\omega t}$.)

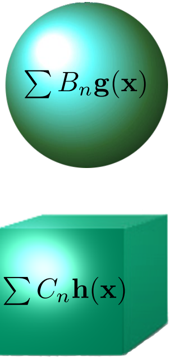


Quick Review of the Boundary-Element Method

One approach to scattering problems: special-function expansions

Write the fields inside and outside the scatterer as expansions in sets of known Maxwell solutions (in some convenient coordinate system) and match coefficients.

- **Spherical geometries**: $\mathbf{f}(\mathbf{x}) \sim j_l(r) Y_{lm}(\theta, \phi)$ ("Mie scattering")
- **Planar geometries**: $\mathbf{f}(\mathbf{x}) \sim e^{i\mathbf{k}\cdot\mathbf{x}}$ ("Fresnel scattering")



$$\mathbf{E}(\mathbf{x}) = \sum A_n \mathbf{f}(\mathbf{x})$$

Advantages:

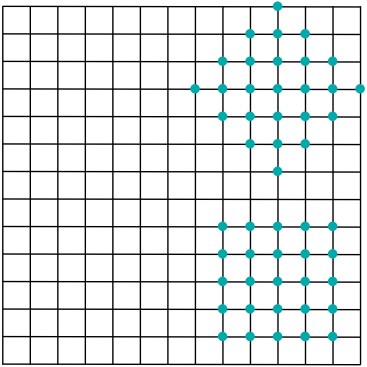
- Exploits known Maxwell solutions $\Longrightarrow$ **efficient**

Disadvantages:

- Only works for very special geometries $\Longrightarrow$ **not general.**

Another approach to scattering problems: the finite-difference method

- Discretize the geometry onto a grid (each grid point can have different ϵ, μ)
- Write Maxwell's equations using finite-difference approximations to derivatives
- Solve sparse linear system for the $\mathbf{E}$-field values at grid points



$$\Big[\nabla\times\nabla\times - k^2\Big]\mathbf{E} = -i\omega\mathbf{J} \Longrightarrow \begin{pmatrix} \mathbf{M} \end{pmatrix}\begin{pmatrix} \mathbf{E}_1 \\ \vdots \\ \mathbf{E}_n \end{pmatrix} = i\omega \begin{pmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_n \end{pmatrix}$$

Advantages:

- Allows different ϵ, μ at each grid point $\longrightarrow$ **very general**

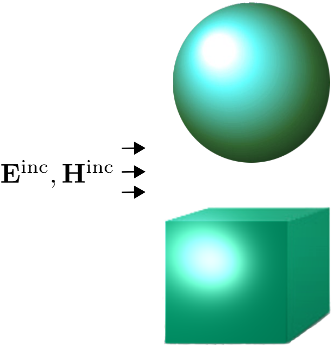
Disadvantages:

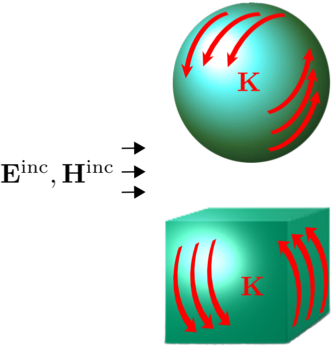
- Does not make use of known Maxwell solutions

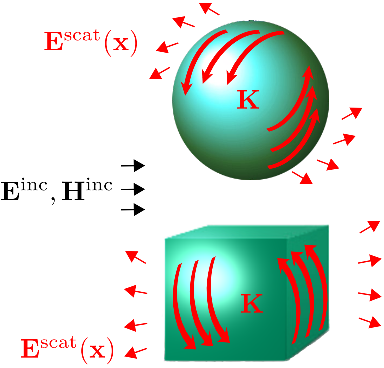
  $\longrightarrow$ **not the most efficient method**

Quick Review of the Boundary-Element Method

The Surface-Integral-Equation (SIE) Approach




K
$\mathbf{E}^{\rm inc}, \mathbf{H}^{\rm inc}$
K


$\mathbf{E}^{\rm scat}(\mathbf{x})$
K
$\mathbf{E}^{\rm inc}, \mathbf{H}^{\rm inc}$
K
$\mathbf{E}^{\rm scat}(\mathbf{x})$

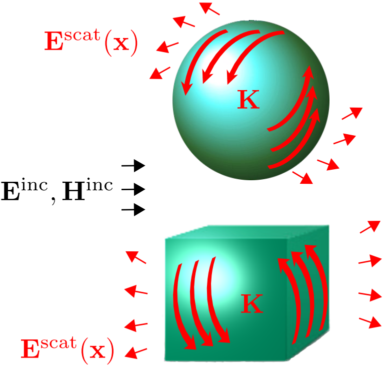
Quick Review of the Boundary-Element Method

The Surface-Integral-Equation (SIE) Approach

- First compute the surface current distribution $\mathbf{K}(\mathbf{x})$ induced by the incident field
- Then compute the scattered fields using $\mathbf{K}(\mathbf{x})$ and known Maxwell solutions:

$$\mathbf{E}^{\text{scat}}(\mathbf{x}) = \oint_{\mathcal{S}} \mathbf{G}(\mathbf{x}-\mathbf{x}')\mathbf{K}(\mathbf{x}')d\mathbf{x}'$$

where $\mathbf{G}$ solves $\left[\nabla\times\nabla\times - k^2\right]\mathbf{G}(\mathbf{r}) = -i\omega\mathbf{1}\delta(\mathbf{r})$



Advantages:

- Exploits known Maxwell solutions $\Longrightarrow$ **efficient**
- Allows scatterers of arbitrary shapes and arbitrary (homogeneous) materials $\Longrightarrow$ **general**

Disadvantages:

- Restricted to linear, isotropic, homogeneous materials, i.e. piecewise-constant ϵ, μ

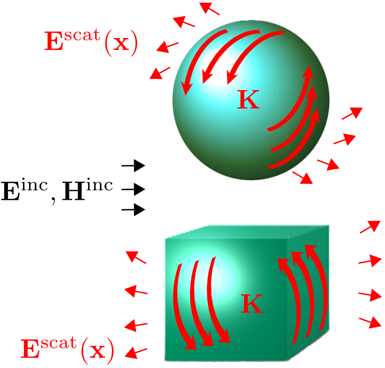
Quick Review of the Boundary-Element Method

The Surface-Integral-Equation (SIE) Approach

- First compute the surface current distribution $\mathbf{K}(\mathbf{x})$ induced by the incident field
- Then compute the scattered fields using $\mathbf{K}(\mathbf{x})$ and known Maxwell solutions:

$$\boxed{\mathbf{E}^{\text{scat}}(\mathbf{x}) = \oint_{\mathcal{S}} \mathbf{G}(\mathbf{x}-\mathbf{x}')\mathbf{K}(\mathbf{x}')d\mathbf{x}'}$$

where $\mathbf{G}$ solves $\left[\nabla\times\nabla\times\ -k^2\right]\mathbf{G}(\mathbf{r}) = -i\omega\mathbf{1}\delta(\mathbf{r})$



Advantages:

- Exploits known Maxwell solutions $\Longrightarrow$ **efficient**
- Allows scatterers of arbitrary shapes and arbitrary (homogeneous) materials $\Longrightarrow$ **general**

Disadvantages:

- Restricted to linear, isotropic, homogeneous materials, i.e. piecewise-constant ϵ, μ

- PEC: only electric surface currents $\mathbf{K}$

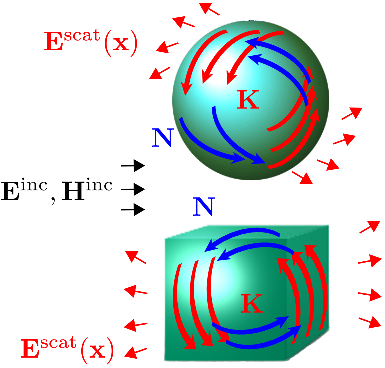
Quick Review of the Boundary-Element Method

The Surface-Integral-Equation (SIE) Approach

- First compute the surface current distribution $\mathbf{K}(\mathbf{x})$ induced by the incident field
- Then compute the scattered fields using $\mathbf{K}(\mathbf{x})$ and known Maxwell solutions:

$$\mathbf{E}^{\text{scat}}(\mathbf{x}) = \oint_{\mathcal{S}} \mathbf{G}(\mathbf{x}-\mathbf{x}')\mathbf{K}(\mathbf{x}')d\mathbf{x}'$$

where $\mathbf{G}$ solves $\left[\nabla\times\nabla\times\ -k^2\right]\mathbf{G}(\mathbf{r}) = -i\omega\mathbf{1}\delta(\mathbf{r})$



Advantages:

- Exploits known Maxwell solutions $\Longrightarrow$ **efficient**
- Allows scatterers of arbitrary shapes and arbitrary (homogeneous) materials $\Longrightarrow$ **general**

Disadvantages:

- Restricted to linear, isotropic, homogeneous materials, i.e. piecewise-constant ϵ, μ

- PEC: only electric surface currents $\mathbf{K}$
- In general: electric **and magnetic** currents $\mathbf{K}, \mathbf{N}$

Boundary conditions relate $\mathbf{E}^{\text{scat}}$ (and thus $\mathbf{K}$) to $\mathbf{E}^{\text{inc}}$:

- For all points $\mathbf{x}$ on object surfaces, we have an integral equation for the surface currents:

$$\boxed{\mathbf{E}_{\parallel}^{\text{scat}}(\mathbf{x}) \quad = \quad \left[\int_{\mathcal{S}} \mathbf{G}(\mathbf{x},\mathbf{x}')\cdot\mathbf{K}(\mathbf{x}')d\mathbf{x}'\right]_{\parallel} \quad = \quad -\mathbf{E}_{\parallel}^{\text{inc}}(\mathbf{x})} \qquad \text{(PEC)}$$

Boundary conditions relate $\mathbf{E}^{\text{scat}}$ (and thus $\mathbf{K}$) to $\mathbf{E}^{\text{inc}}$:

- For all points $\mathbf{x}$ on object surfaces, we have an integral equation for the surface currents:

$$\left(\begin{array}{c}\mathbf{E}\\ \mathbf{H}\end{array}\right)^{\text{scat}}_{\parallel} = \left[\int_{\mathcal{S}} \left(\begin{array}{cc}\mathbf{G}^{\text{EE}} & \mathbf{G}^{\text{EM}}\\ \mathbf{G}^{\text{ME}} & \mathbf{G}^{\text{MM}}\end{array}\right)\cdot\left(\begin{array}{c}\mathbf{K}\\ \mathbf{N}\end{array}\right) d\mathbf{x}'\right]_{\parallel} = -\left(\begin{array}{c}\mathbf{E}\\ \mathbf{H}\end{array}\right)^{\text{inc}}_{\parallel} \qquad \text{(General)}$$

Boundary conditions relate $\mathbf{E}^{\text{scat}}$ (and thus $\mathbf{K}$) to $\mathbf{E}^{\text{inc}}$:

- For all points $\mathbf{x}$ on object surfaces, we have an integral equation for the surface currents:

$$\left(\begin{array}{c}\mathbf{E}\\ \mathbf{H}\end{array}\right)^{\text{scat}}_{\parallel} = \left[\int_{\mathcal{S}} \left(\begin{array}{cc}\mathbf{G}^{\text{EE}} & \mathbf{G}^{\text{EM}}\\ \mathbf{G}^{\text{ME}} & \mathbf{G}^{\text{MM}}\end{array}\right)\cdot\left(\begin{array}{c}\mathbf{K}\\ \mathbf{N}\end{array}\right) d\mathbf{x}'\right]_{\parallel} = -\left(\begin{array}{c}\mathbf{E}\\ \mathbf{H}\end{array}\right)^{\text{inc}}_{\parallel} \qquad \text{(General)}$$

Expand $\mathbf{K},\mathbf{N}$ in discrete set $\{\mathbf{b}_\alpha\}$:

$$\left(\begin{array}{c}\mathbf{K}(\mathbf{x})\\ \mathbf{N}(\mathbf{x})\end{array}\right) = \sum_\alpha \left(\begin{array}{c}k_\alpha\\ n_\alpha\end{array}\right)\mathbf{b}_\alpha(\mathbf{x})$$

$\{\mathbf{b}_\alpha(\mathbf{x})\}$ are tangential vector-valued basis functions localized on the object surfaces ("boundary elements")

Boundary conditions relate $\mathbf{E}^{\text{scat}}$ (and thus $\mathbf{K}$) to $\mathbf{E}^{\text{inc}}$:

- For all points $\mathbf{x}$ on object surfaces, we have an integral equation for the surface currents:

$$\left(\begin{array}{c}\mathbf{E}\\ \mathbf{H}\end{array}\right)^{\text{scat}}_{\parallel} = \left[\int_{\mathcal{S}} \left(\begin{array}{cc}\mathbf{G}^{\text{EE}} & \mathbf{G}^{\text{EM}}\\ \mathbf{G}^{\text{ME}} & \mathbf{G}^{\text{MM}}\end{array}\right)\cdot\left(\begin{array}{c}\mathbf{K}\\ \mathbf{N}\end{array}\right) d\mathbf{x}'\right]_{\parallel} = -\left(\begin{array}{c}\mathbf{E}\\ \mathbf{H}\end{array}\right)^{\text{inc}}_{\parallel} \qquad \text{(General)}$$

Expand $\mathbf{K},\mathbf{N}$ in discrete set $\{\mathbf{b}_\alpha\}$:

$$\left(\begin{array}{c}\mathbf{K}(\mathbf{x})\\ \mathbf{N}(\mathbf{x})\end{array}\right) = \sum_\alpha \left(\begin{array}{c}k_\alpha\\ n_\alpha\end{array}\right)\mathbf{b}_\alpha(\mathbf{x})$$

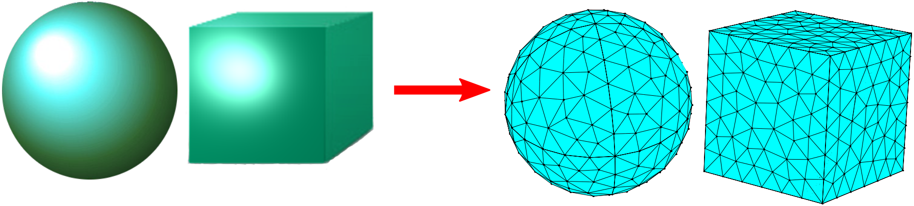
$\Downarrow$

$\{\mathbf{b}_\alpha(\mathbf{x})\}$ are tangential vector-valued basis functions localized on the object surfaces ("boundary elements")

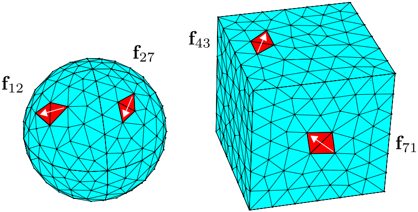
$\Longrightarrow$ Integral equation becomes a linear system, $\boxed{\mathbf{Mk} = \mathbf{v}}$

$$\mathbf{k} = \left(\begin{array}{c}k_\alpha\\ n_\alpha\end{array}\right), \qquad \mathbf{v} = -\left(\begin{array}{c}\langle \mathbf{b}_\alpha | \, \mathbf{E}^{\text{inc}}\rangle\\ \langle \mathbf{b}_\alpha | \, \mathbf{H}^{\text{inc}}\rangle\end{array}\right), \qquad M_{\alpha\beta} = \left\langle \mathbf{b}_\alpha \middle| \mathbf{G} \middle| \mathbf{b}_\beta \right\rangle$$

Begin by discretizing ("meshing") object surfaces into triangles:



Associate one basis function with each internal edge:



- These are "RWG basis functions" (named for their inventors: Rao, Wilton, Glisson)
- # of basis functions $N \propto$ # of triangles
- As we refine the discretization (shrink the triangles), the discretization errors decrease, but the cost of solving the linear system grows like N^3

Quick Review of the Boundary-Element Method

What does a BEM solver actually need to do?

The steps involved in solving any BEM scattering problem:

1. Mesh object surfaces into triangles.

   Not done by SCUFF-EM; high-quality free meshing packages exist (e.g. GMSH).

2. Assemble the BEM matrix $\mathbf{M}$ and RHS vector $\mathbf{v}$.

   SCUFF-EM **does this.**

3. Solve the linear system $\mathbf{M}\mathbf{k} = \mathbf{v}$ for the surface currents $\mathbf{k}$.

   SCUFF-EM uses LAPACK for this.

4. Post-process to compute scattered fields $\{\mathbf{E}, \mathbf{H}\}^{\text{scat}}$ from $\mathbf{k}$.

   SCUFF-EM **does this.**

Quick Review of the Boundary-Element Method

What does a BEM solver actually need to do?

The steps involved in solving any BEM scattering problem:

1. Mesh object surfaces into triangles.

   Not done by SCUFF-EM; high-quality free meshing packages exist (e.g. GMSH).

2. Assemble the BEM matrix $\mathbf{M}$ and RHS vector $\mathbf{v}$.

   SCUFF-EM **does this.**

3. Solve the linear system $\mathbf{M}\mathbf{k} = \mathbf{v}$ for the surface currents $\mathbf{k}$.

   SCUFF-EM uses LAPACK for this.

4. Post-process to compute scattered fields $\{\mathbf{E}, \mathbf{H}\}^{\text{scat}}$ from $\mathbf{k}$.

   SCUFF-EM **does this.**

Innovations unique to SCUFF-EM:

- Bypass step 4: Compute scattered/absorbed power, force, and torque **directly from $\mathbf{k}$**

Quick Review of the Boundary-Element Method

What does a BEM solver actually need to do?

The steps involved in solving any BEM scattering problem:

1. **Mesh object surfaces into triangles.**

   Not done by SCUFF-EM; high-quality free meshing packages exist (e.g. GMSH).

2. **Assemble the BEM matrix $\mathbf{M}$ and RHS vector $\mathbf{v}$.**

   SCUFF-EM **does this.**

3. **Solve the linear system $\mathbf{M}\mathbf{k} = \mathbf{v}$ for the surface currents $\mathbf{k}$.**

   SCUFF-EM uses LAPACK for this.

4. **Post-process to compute scattered fields $\{\mathbf{E}, \mathbf{H}\}^{\text{scat}}$ from $\mathbf{k}$.**

   SCUFF-EM **does this.**

Innovations unique to SCUFF-EM:

- Bypass step 4: Compute scattered/absorbed power, force, and torque **directly from $\mathbf{k}$**
- Bypass steps 3 and 4: Compute Casimir forces and heat transfer **directly from $\mathbf{M}$**

Outline

1. A quick review of the Boundary-Element Method

2. A brief history of the evolution of SCUFF-EM

3. **What SCUFF-EM can do**

    1. Inputs: The geometries, materials, incident fields that SCUFF-EM can handle
    2. Outputs: The various calculations that SCUFF-EM can do
    3. Mechanics: How to run SCUFF-EM

4. Under the hood: How SCUFF-EM works

The Prehistory of SCUFF-EM

From computational Casimir physics to general E&M

SCUFF-EM was originally born as a numerical Casimir solver.

First-generation numerical Casimir solvers: FDTD	PHYSICAL REVIEW A **76**, 032106 (2007) **Virtual photons in imaginary time: Computing exact Casimir forces via standard numerical electromagnetism techniques** Alejandro Rodriguez,[1] Mihai Ibanescu,[1] Davide Iannuzzi,[2] J. D. Joannopoulos,[1] and Steven G. Johnson[1] [1]*Center for Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA* [2]*Faculty of Sciences, Department of Physics and Department of Astronomy, Vrije Universiteit Amsterdam, The Netherlands* (Received 25 May 2007; published 6 September 2007)
Second-generation numerical Casimir solvers: BEM	PRL **103**, 040401 (2009) PHYSICAL REVIEW LETTERS week ending 24 JULY 2009 **Efficient Computation of Casimir Interactions between Arbitrary 3D Objects** M. T. Homer Reid,[1,2,*,†] Alejandro W. Rodriguez,[1] Jacob White,[2,3] and Steven G. Johnson[2,4] [1]*Department of Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA* [2]*Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA* [3]*Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA* [4]*Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA* (Received 4 April 2009; revised manuscript received 10 June 2009; published 20 July 2009)

The Casimir application mandated several features from the start:

- Support for complex-valued frequencies
- Efficient calculations at many frequencies (near-DC→optical)
- Ability to displace objects without starting over from scratch

The Evolution of SCUFF-EM

A chronological progression of new features and broader generality

- As of February 2011: Imaginary frequencies, lossless materials

- As of February 2011: Imaginary frequencies, lossless materials

"If only you added real frequencies and lossy materials, this would be a useful code."

The Evolution of SCUFF-EM

A chronological progression of new features and broader generality

- As of February 2011: Imaginary frequencies, lossless materials

"If only you added real frequencies and lossy materials, this would be a useful code."

- By Fall 2011: real frequencies, lossy materials

The Evolution of SCUFF-EM

A chronological progression of new features and broader generality

- As of February 2011: Imaginary frequencies, lossless materials

"If only you added real frequencies and lossy materials, this would be a useful code."

- By Fall 2011: real frequencies, lossy materials

"If only you added periodic boundary conditions, this would be a useful code."

The Evolution of SCUFF-EM

A chronological progression of new features and broader generality

- As of February 2011: Imaginary frequencies, lossless materials

"If only you added real frequencies and lossy materials, this would be a useful code."

- By Fall 2011: real frequencies, lossy materials

"If only you added periodic boundary conditions, this would be a useful code."

- By Spring 2012: periodic boundary conditions

"If only you added multi-material junctions, this would be a useful code."

The Evolution of SCUFF-EM

A chronological progression of new features and broader generality

- As of February 2011: Imaginary frequencies, lossless materials

"If only you added real frequencies and lossy materials, this would be a useful code."

- By Fall 2011: real frequencies, lossy materials

"If only you added periodic boundary conditions, this would be a useful code."

- By Spring 2012: periodic boundary conditions

"If only you added multi-material junctions, this would be a useful code."

- By Summer 2012: multi-material junctions

The Evolution of SCUFF-EM

A chronological progression of new features and broader generality

- As of February 2011: Imaginary frequencies, lossless materials

"If only you added real frequencies and lossy materials, this would be a useful code."

- By Fall 2011: real frequencies, lossy materials

"If only you added periodic boundary conditions, this would be a useful code."

- By Spring 2012: periodic boundary conditions

"If only you added multi-material junctions, this would be a useful code."

- By Summer 2012: multi-material junctions

"If only you added ..."

The Evolution of SCUFF-EM

A chronological progression of new features and broader generality

- As of February 2011: Imaginary frequencies, lossless materials

"If only you added real frequencies and lossy materials, this would be a useful code."

- By Fall 2011: real frequencies, lossy materials

"If only you added periodic boundary conditions, this would be a useful code."

- By Spring 2012: periodic boundary con

"... dded multi-material junctions,

... 2012: multi-mat... al juncti

... f only you added ..."

No new features for you!

The Current Status of SCUFF-EM

SCUFF-EM Version 1.0 Public Release (Fall 2012)

- Arbitrary complex frequencies
- Perfect/imperfect metals, lossless/lossy dielectrics, linear magnetic materials
- Periodic boundary conditions
- Multi-material junctions
- Fast computation and caching of BEM matrix elements
- Fast computation of power, force, torque
- 8 standalone application codes
- C++ / python interface

My goal for the short-term future: **Expand the user base**

SCUFF-EM Version 2.0 (Hypothetical)

- Fast solver: reduce complexity scaling from $O(N^3)$ to $O(N \log N)$

Outline

1. A quick review of the Boundary-Element Method
2. A brief history of the evolution of SCUFF-EM
3. **What SCUFF-EM can do**
    1. Inputs: The geometries, materials, incident fields that SCUFF-EM can handle
    2. Outputs: The various calculations that SCUFF-EM can do
    3. Mechanics: How to run SCUFF-EM
4. Under the hood: How SCUFF-EM works

Mechanics of SCUFF-EM

How to run a SCUFF-EM calculation

1. Generate surface meshes for all object surfaces in your geometry.

2. Write a SCUFF-EM geometry file describing objects and materials.

3A. Run one of the 8 standalone command-line applications bundled with the SCUFF-EM suite.

OR

3B. Write your own C++ or PYTHON code using the SCUFF-EM core library API.

Geometry descriptions in SCUFF-EM

- Geometries in SCUFF-EM are described by simple text files.

  - These files are conventionally given the file extension `.scuffgeo`.

- Various types of geometries are possible

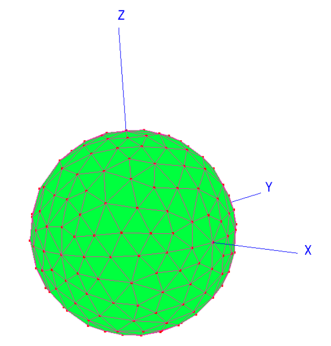
  - The simplest case: One or more compact objects (possibly nested)
  - More complicated cases: multi-material junctions
  - Extended geometries: periodic boundary conditions

The geometries that SCUFF-EM can handle

Simple geometries: One or more compact homogeneous objects (possibly nested)

A single gold sphere:

The geometry:

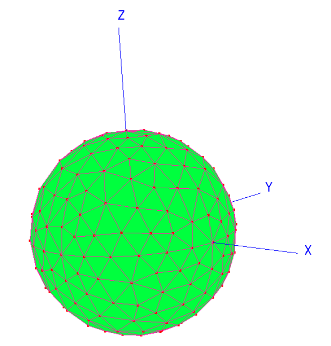


The .scuffgeo file:

```
OBJECT TheSphere
   MESHFILE Sphere.msh
   MATERIAL Gold
ENDOBJECT
```

A single gold sphere:

The geometry:



Mesh file generated by external meshing software (such as GMSH)

The .scuffgeo file:

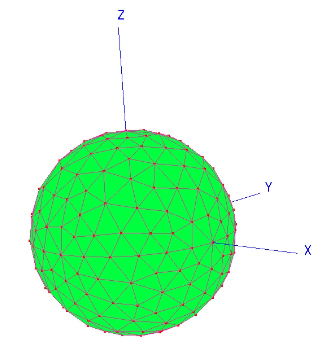
```
OBJECT TheSphere
   MESHFILE Sphere.msh
   MATERIAL Gold
ENDOBJECT
```

The geometries that SCUFF-EM can handle

Simple geometries: One or more compact homogeneous objects (possibly nested)

A single gold sphere:

The geometry:



Mesh file generated by external meshing software (such as GMSH)

The .scuffgeo file:

```
OBJECT TheSphere
   MESHFILE Sphere.msh
   MATERIAL Gold
ENDOBJECT
```

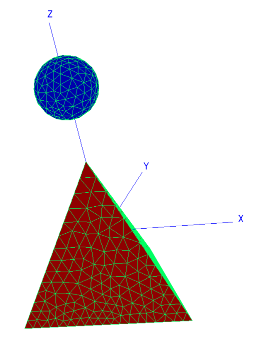
SCUFF-EM Material Designation

The geometries that SCUFF-EM can handle

Simple geometries: One or more compact homogeneous objects (possibly nested)

A gold sphere and an SiO2 tetrahedron:

The geometry:



The .scuffgeo file:

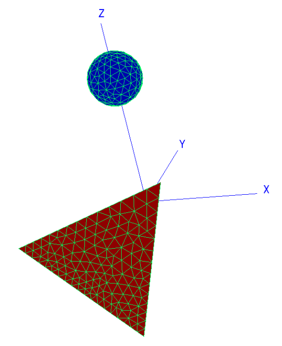
```
OBJECT TheSphere
   MESHFILE Sphere.msh
   MATERIAL Gold
ENDOBJECT

OBJECT ThePyramid
   MESHFILE Pyramid.msh
   MATERIAL SiO2
ENDOBJECT
```

The geometries that SCUFF-EM can handle

Simple geometries: One or more compact homogeneous objects (possibly nested)

A gold sphere and a displaced and rotated SiO2 tetrahedron:

The geometry:



The .scuffgeo file:

```
OBJECT TheSphere
   MESHFILE Sphere.msh
   MATERIAL Gold
ENDOBJECT

OBJECT ThePyramid
   MESHFILE Pyramid.msh
   MATERIAL SiO2
   DISPLACED 0 0 -1
   ROTATED 45 ABOUT 0 1 0
ENDOBJECT
```

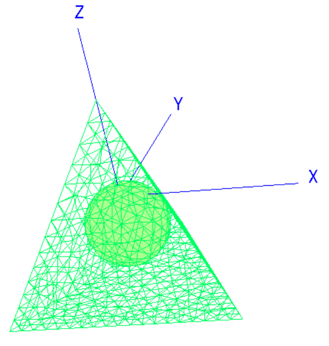
$\Longrightarrow$ Handle displacements and rotations without re-meshing.

The geometries that SCUFF-EM can handle

Simple geometries: One or more compact homogeneous objects (possibly nested)

A gold sphere inside an SiO2 tetrahedron:

The geometry:



The .scuffgeo file:

```
OBJECT TheSphere
   MESHFILE Sphere.msh
   MATERIAL Gold
   DISPLACED 0 0 -3
ENDOBJECT

OBJECT ThePyramid
   MESHFILE Pyramid.msh
   MATERIAL SiO2
ENDOBJECT
```

$\Longrightarrow$ Object inclusions are *autodetected.* (Thanks to SGJ for this feature.)

Material Designations in SCUFF-EM

Many ways to specify frequency-dependent permittivity ϵ and permeability μ

Special Materials

```
MATERIAL VACUUM

MATERIAL PEC
```

Frequency-independent ϵ and μ

- Useful for single-frequency calculations

```
MATERIAL CONST_EPS_12.8

MATERIAL CONST_EPS_3.4+5.6I_MU_12.9
```

Functional Forms

- Arbitrary user-specified expressions

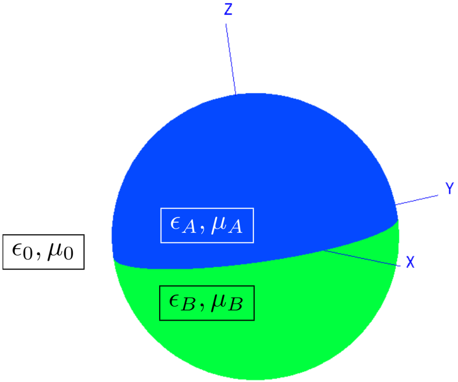
```
MATERIAL GOLD
  wp = 1.37e16;
  gamma = 5.32e13;
  Eps(w) = 1-wp^2 / (w*(w+i*gamma));
ENDMATERIAL
```

Tabulated Data

- SCUFF-EM will automatically interpolate

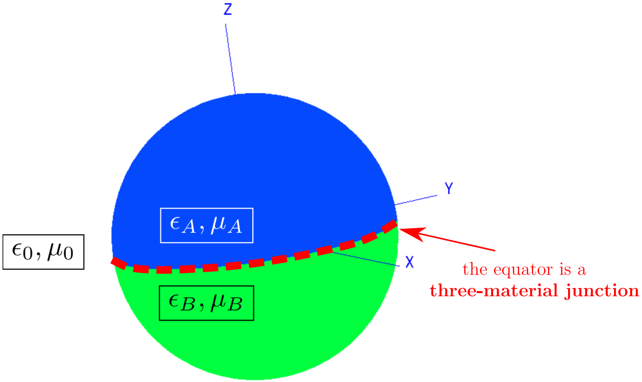
```
MATERIAL FILE_SiliconDataFile.dat

 1.0e11   12.83 0.1
 ...
 1.0e14   -9.11 3.9
```

Some geometries cannot be described as a collection of compact objects:



$\Longrightarrow$ These geometries are described in terms of regions and surfaces.

Some geometries cannot be described as a collection of compact objects:

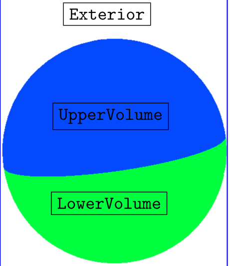


$\Longrightarrow$ These geometries are described in terms of regions and surfaces.

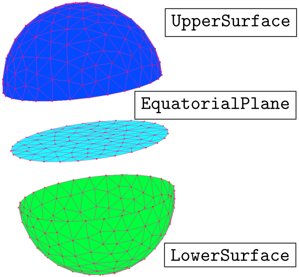
The geometries that SCUFF-EM can handle

Geometries with multi-material junctions are described using regions and surfaces

Regions



Surfaces



`.scuffgeo` File

```
REGION Exterior    MATERIAL Vacuum
REGION UpperVolume MATERIAL Gold
REGION LowerVolume MATERIAL Silicon

SURFACE UpperSurface
     MESHFILE UpperSurface.msh
     REGIONS Exterior UpperVolume
ENDSURFACE

SURFACE LowerSurface
     MESHFILE LowerSurface.msh
     REGIONS Exterior LowerVolume
ENDSURFACE

SURFACE EquatorialPlane
     MESHFILE EquatorialPlane.msh
     REGIONS UpperVolume LowerVolume
ENDSURFACE
```

The geometries that SCUFF-EM can handle

Extended geometries: periodic boundary conditions

To describe something like this...

[Garcia-Vidal et al, RMP **82** 729 (2010)]

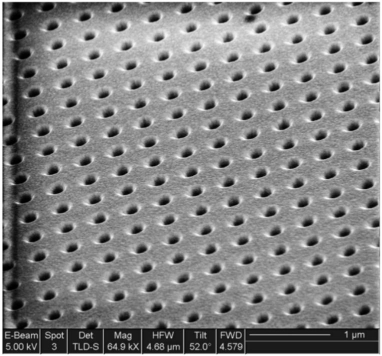
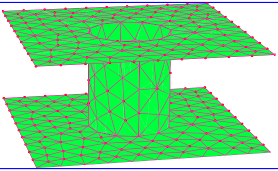
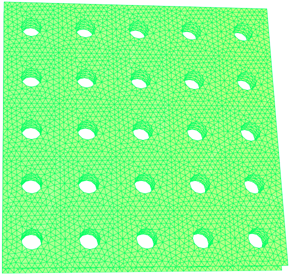


FIG. 19. SEM image of a 2D hole array of circular holes (diameter of 150 nm) milled in a 260-nm-thick Au film that is deposited on a glass substrate. The hole arrays count 30×30 holes and the period of the square array is 460 nm. Courtesy of Eric Laux.

The geometries that SCUFF-EM can handle

Extended geometries: periodic boundary conditions

...we define a lattice and a unit cell mesh.

Unit cell mesh	**First 25 lattice cells**
	
`.scuffgeo` **file**	

```
LATTICE
        VECTOR 0.75 0
        VECTOR 0    0.75
ENDLATTICE

OBJECT UnitCell
        MESHFILE UnitCellMesh.msh
        MATERIAL Silver
ENDOBJECT
```

Command-line applications in the SCUFF-EM suite

Electromagnetic Scattering

- SCUFF-SCATTER: general-purpose scattering
- SCUFF-TRANSMISSION: plane-wave transmission through extended structures
- SCUFF-TMATRIX: spherical-basis T-Matrix of compact objects

RF / Microwave Device Engineering

- SCUFF-RF: Circuit parameters and radiated fields of passive RF devices

Fluctuation Physics

- SCUFF-CAS3D: Casimir energy, force, torque in 3D geometries
- SCUFF-CAS2D: Casimir energy, force, torque in 2D geometries
- SCUFF-CASPOL: Casimir-Polder potentials for polarizable particles near surfaces
- SCUFF-NEQ: Nonequilibrium fluctuations: Radiative heat transfer & non-EQ Casimir forces

Command-line applications in the SCUFF-EM suite

Electromagnetic Scattering

- SCUFF-SCATTER: general-purpose scattering
- SCUFF-TRANSMISSION: plane-wave transmission through extended structures
- SCUFF-TMATRIX: spherical-basis T-Matrix of compact objects

RF / Microwave Device Engineering

- SCUFF-RF: Circuit parameters and radiated fields of passive RF devices

Fluctuation Physics

- SCUFF-CAS3D: Casimir energy, force, torque in 3D geometries
- SCUFF-CAS2D: Casimir energy, force, torque in 2D geometries
- SCUFF-CASPOL: Casimir-Polder potentials for polarizable particles near surfaces
- SCUFF-NEQ: Nonequilibrium fluctuations: Radiative heat transfer & non-EQ Casimir forces

SCUFF-SCATTER: A general-purpose scattering application

Inputs to SCUFF-SCATTER:

1. Your scattering geometry (`.scuffgeo` file)
2. Incident field specification: plane wave, point source, Gaussian beam, or any combination
3. Frequency or Frequency range
4. Optional: List of field evaluation points

SCUFF-SCATTER: A general-purpose scattering application

Inputs to SCUFF-SCATTER:

1. Your scattering geometry (`.scuffgeo` file)
2. Incident field specification: plane wave, point source, Gaussian beam, or any combination
3. Frequency or Frequency range
4. Optional: List of field evaluation points

Outputs available from SCUFF-SCATTER:

- **E** and **H** Field Components (scattered and total) at user-specified evaluation points
- Power absorbed by and scattered from each scattering object
- Force/Torque imparted to the scattering objects by the incident field (radiation pressure)
- Induced dipole moments (Cartesian basis)
- Induced multipole moments (Spherical basis)
- Visualization files for surface currents and scattered fields

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

geometry file

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

angular frequency

(units: $\frac{c}{1\mu\text{m}} = 3{\cdot}10^{14}$ rad/s)

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

incident field (plane wave)

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

list of field evaluation points

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

PFTFile: request power, force, torque

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

request induced dipole moments

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

request surface current visualization

Mie scattering in SCUFF-SCATTER

Options may be specified on the command line *or* via text file piped to `stdin`

Put command-line arguments into a text file (call it `scuff-scatter.args`):

```
geometry Sphere_681.scuffgeo

Omega 1.1

pwDirection 0 0 1
pwPolarization 1 0 0

EPFile MyEvalPoints

PFTFile Sphere_681.PFT

MomentFile Sphere_681.moments

PlotSurfaceCurrents
```

Run SCUFF-SCATTER from the command line:

```
% scuff-scatter < scuff-scatter.args
```

Incident fields in SCUFF-EM

Several built-in types available; also easy to define your own in API programs

Plane waves: specify direction and polarization

- A circularly-polarized wave traveling in the $+\hat{\mathbf{z}}$ direction:

```
--pwDirection 0 0 1 --pwPolarization 1 i 0
```

Point dipoles: specify location, strength, and type (electric or magnetic)

- A point electric dipole at $\mathbf{x} = (2, 3, 4)$ μm with dipole moment $\mathbf{p} = (4, 5i, 6)$ V/(μm)^2

```
--psLocation 2 3 4 --psStrength 4 5i 6
```

Gaussian laser beams: specify direction, polarization, beam center, and beam waist

- An upward-propagating beam with beam waist 1μm:

```
--gbDirection 0 0 1 --gbPolarization 1 0 0 --gbCenter 0 0 0 --gbWaist 2
```

Thanks to **Johannes Feist** for contributing the Gaussian beam code

Or: any combination of the above; or, define your own in API codes.

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

```
0.1 Sphere 1.1e-11 5.6e-11 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.2 Sphere 2.2e-11 1.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.3 Sphere 3.6e-11 3.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.4 Sphere 6.5e-11 1.0e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.5 Sphere 1.1e-10 2.9e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.6 Sphere 1.8e-10 8.4e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.7 Sphere 3.0e-10 2.1e-08 2.2e-06 -1.5e-05 -9.6e-06 0.0e+00 0.0e+00 0.0e+00
```

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The --PFTFile MyFile.PFT option writes power, force, and torque data to the given file.

```
0.1 Sphere 1.1e-11 5.6e-11 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.2 Sphere 2.2e-11 1.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.3 Sphere 3.6e-11 3.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.4 Sphere 6.5e-11 1.0e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.5 Sphere 1.1e-10 2.9e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.6 Sphere 1.8e-10 8.4e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.7 Sphere 3.0e-10 2.1e-08 2.2e-06 -1.5e-05 -9.6e-06 0.0e+00 0.0e+00 0.0e+00
```

angular frequency

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

0.1	Sphere	1.1e-11	5.6e-11	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00	
0.2	Sphere	2.2e-11	1.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00	
0.3	Sphere	3.6e-11	3.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00	
0.4	Sphere	6.5e-11	1.0e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00	
0.5	Sphere	1.1e-10	2.9e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00	
0.6	Sphere	1.8e-10	8.4e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00	
0.7	Sphere	3.0e-10	2.1e-08	2.2e-06	-1.5e-05	-9.6e-06	0.0e+00	0.0e+00	0.0e+00	

object label in `.scuffgeo` file

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

```
0.1 Sphere 1.1e-11 5.6e-11 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.2 Sphere 2.2e-11 1.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.3 Sphere 3.6e-11 3.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.4 Sphere 6.5e-11 1.0e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.5 Sphere 1.1e-10 2.9e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.6 Sphere 1.8e-10 8.4e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.7 Sphere 3.0e-10 2.1e-08 2.2e-06 -1.5e-05 -9.6e-06 0.0e+00 0.0e+00 0.0e+00
```

absorbed power (W)

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

```
0.1 Sphere 1.1e-11 5.6e-11 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.2 Sphere 2.2e-11 1.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.3 Sphere 3.6e-11 3.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.4 Sphere 6.5e-11 1.0e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.5 Sphere 1.1e-10 2.9e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.6 Sphere 1.8e-10 8.4e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.7 Sphere 3.0e-10 2.1e-08 2.2e-06 -1.5e-05 -9.6e-06 0.0e+00 0.0e+00 0.0e+00
```

scattered power (W)

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

0.1	Sphere	1.1e-11	5.6e-11	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.2	Sphere	2.2e-11	1.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.3	Sphere	3.6e-11	3.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.4	Sphere	6.5e-11	1.0e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.5	Sphere	1.1e-10	2.9e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.6	Sphere	1.8e-10	8.4e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.7	Sphere	3.0e-10	2.1e-0[illegible]	2.2e-06	-1.5e-05	-9.6e-06	0.0e+00	0.0e+00	0.0e+00

x, y, z components of force (radiation pressure) in nN

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

```
0.1 Sphere 1.1e-11 5.6e-11 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.2 Sphere 2.2e-11 1.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.3 Sphere 3.6e-11 3.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.4 Sphere 6.5e-11 1.0e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.5 Sphere 1.1e-10 2.9e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.6 Sphere 1.8e-10 8.4e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.7 Sphere 3.0e-10 2.1e-08 2.2e-06 -1.5e-05 -9.6e-06 0.0e+00 0.0e+00 0.0e+00
```

x, y, z components of torque in nN $\cdot\mu$m

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

```
0.1 Sphere 1.1e-11 5.6e-11 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.2 Sphere 2.2e-11 1.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.3 Sphere 3.6e-11 3.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.4 Sphere 6.5e-11 1.0e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.5 Sphere 1.1e-10 2.9e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.6 Sphere 1.8e-10 8.4e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.7 Sphere 3.0e-10 2.1e-08 2.2e-06 -1.5e-05 -9.6e-06 0.0e+00 0.0e+00 0.0e+00
```

The `--MomentFile MyMoments.dat` option writes induced dipole moments to the given file.

```
0.1 TheSphere 1.2e+01 1.3e-04 -1.1e-02 9.2e-06 -4.3e-03 1.0e-05
0.2 TheSphere 1.2e+01 2.0e-04 -1.1e-02 1.2e-05 -4.3e-03 1.3e-05
0.3 TheSphere 1.2e+01 2.8e-04 -1.1e-02 1.4e-05 -4.3e-03 1.6e-05
```

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

```
0.1 Sphere 1.1e-11 5.6e-11 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.2 Sphere 2.2e-11 1.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.3 Sphere 3.6e-11 3.6e-10 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.4 Sphere 6.5e-11 1.0e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.5 Sphere 1.1e-10 2.9e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.6 Sphere 1.8e-10 8.4e-09 2.2e-06 -1.5e-05 -9.5e-06 0.0e+00 0.0e+00 0.0e+00
0.7 Sphere 3.0e-10 2.1e-08 2.2e-06 -1.5e-05 -9.6e-06 0.0e+00 0.0e+00 0.0e+00
```

The `--MomentFile MyMoments.dat` option writes induced dipole moments to the given file.

```
0.1 TheSphere 1.2e+01 1.3e-04 -1.1e-02 9.2e-06 -4.3e-03 1.0e-05
0.2 TheSphere 1.2e+01 2.0e-04 -1.1e-02 1.2e-05 -4.3e-03 1.3e-05
0.3 TheSphere 1.2e+01 2.8e-04 -1.1e-02 1.4e-05 -4.3e-03 1.6e-05
```

angular frequency

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

0.1	Sphere	1.1e-11	5.6e-11	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.2	Sphere	2.2e-11	1.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.3	Sphere	3.6e-11	3.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.4	Sphere	6.5e-11	1.0e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.5	Sphere	1.1e-10	2.9e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.6	Sphere	1.8e-10	8.4e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.7	Sphere	3.0e-10	2.1e-08	2.2e-06	-1.5e-05	-9.6e-06	0.0e+00	0.0e+00	0.0e+00

The `--MomentFile MyMoments.dat` option writes induced dipole moments to the given file.

0.1	TheSphere	1.2e+01	1.3e-04	-1.1e-02	9.2e-06	-4.3e-03	1.0e-05
0.2	TheSphere	1.2e+01	2.0e-04	-1.1e-02	1.2e-05	-4.3e-03	1.3e-05
0.3	TheSphere	1.2e+01	2.8e-04	-1.1e-02	1.4e-05	-4.3e-03	1.6e-05

object label in `.scuffgeo` file

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

0.1	Sphere	1.1e-11	5.6e-11	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.2	Sphere	2.2e-11	1.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.3	Sphere	3.6e-11	3.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.4	Sphere	6.5e-11	1.0e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.5	Sphere	1.1e-10	2.9e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.6	Sphere	1.8e-10	8.4e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.7	Sphere	3.0e-10	2.1e-08	2.2e-06	-1.5e-05	-9.6e-06	0.0e+00	0.0e+00	0.0e+00

The `--MomentFile MyMoments.dat` option writes induced dipole moments to the given file.

0.1	TheSphere	1.2e+01	1.3e-04	-1.1e-02	9.2e-06	-4.3e-03	1.0e-05
0.2	TheSphere	1.2e+01	2.0e-04	-1.1e-02	1.2e-05	-4.3e-03	1.3e-05
0.3	TheSphere	1.2e+01	2.8e-04	-1.1e-02	1.4e-05	-4.3e-03	1.6e-05

x, y, z components of induced electric dipole moment

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The `--PFTFile MyFile.PFT` option writes power, force, and torque data to the given file.

0.1	Sphere	1.1e-11	5.6e-11	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.2	Sphere	2.2e-11	1.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.3	Sphere	3.6e-11	3.6e-10	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.4	Sphere	6.5e-11	1.0e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.5	Sphere	1.1e-10	2.9e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.6	Sphere	1.8e-10	8.4e-09	2.2e-06	-1.5e-05	-9.5e-06	0.0e+00	0.0e+00	0.0e+00
0.7	Sphere	3.0e-10	2.1e-08	2.2e-06	-1.5e-05	-9.6e-06	0.0e+00	0.0e+00	0.0e+00

The `--MomentFile MyMoments.dat` option writes induced dipole moments to the given file.

0.1	TheSphere	1.2e+01	1.3e-04	-1.1e-02	9.2e-06	-4.3e-03	1.0e-05
0.2	TheSphere	1.2e+01	2.0e-04	-1.1e-02	1.2e-05	-4.3e-03	1.3e-05
0.3	TheSphere	1.2e+01	2.8e-04	-1.1e-02	1.4e-05	-4.3e-03	1.6e-05

x, y, z components of induced magnetic dipole moment

The --PlotSurfaceCurrents option produces a GMSH visualization file named Sphere_681.pp:

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The --PlotSurfaceCurrents option produces a GMSH visualization file named Sphere_681.pp:

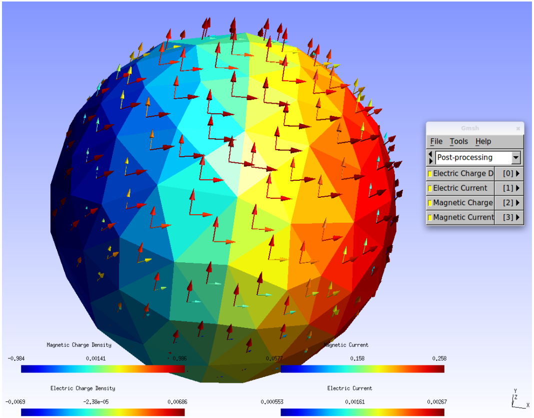
```
% gmsh Sphere_681.pp
```

Mie scattering in SCUFF-SCATTER

Interpreting the output files

The --PlotSurfaceCurrents option produces a GMSH visualization file named Sphere_681.pp:

```
% gmsh Sphere_681.pp
```



- Arrows indicate induced electric and magnetic surface currents
- Panel colors indicate induced electric and magnetic charge densities

The --EPFile option in SCUFF-SCATTER

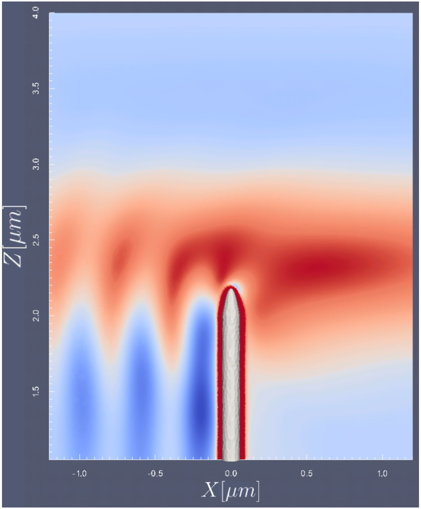
Evaluating fields at arbitrary user-specified points

The command

```
% scuff-scatter ... --EPFile MyPoints ...
```

produces two files:

1. MyPoints.scattered (scattered fields)
2. MyPoints.total (total fields)



(Image: Johannes Feist)

Command-line applications in the SCUFF-EM suite

Electromagnetic Scattering

- SCUFF-SCATTER: general-purpose scattering
- SCUFF-TRANSMISSION: plane-wave transmission through extended structures
- SCUFF-TMATRIX: spherical-basis T-Matrix of compact objects

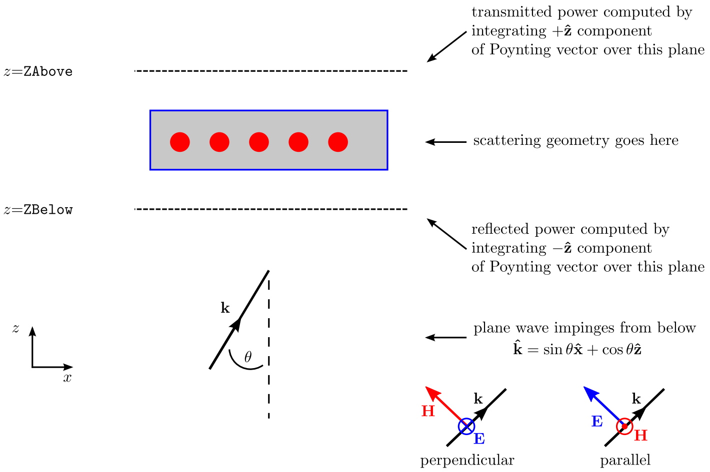
RF / Microwave Device Engineering

- SCUFF-RF: Circuit parameters and radiated fields of passive RF devices

Fluctuation Physics

- SCUFF-CAS3D: Casimir energy, force, torque in 3D geometries
- SCUFF-CAS2D: Casimir energy, force, torque in 2D geometries
- SCUFF-CASPOL: Casimir-Polder potentials for polarizable particles near surfaces
- SCUFF-NEQ: Nonequilibrium fluctuations: Radiative heat transfer & non-EQ Casimir forces

SCUFF-TRANSMISSION illuminates your structure from below with plane waves, then integrates the Poynting vector over the unit-cell area to compute transmitted and reflected flux.



Inputs to SCUFF-TRANSMISSION:

1. Your scattering geometry (`.scuffgeo` file)
2. Frequency or Frequency range (ω)
3. Incident angle or Incident angle range (θ)

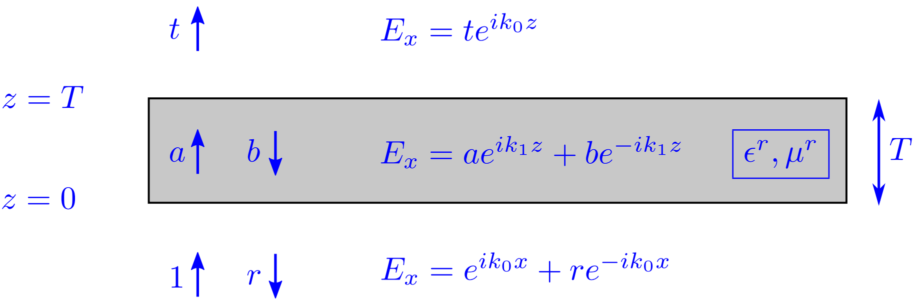
Inputs to SCUFF-TRANSMISSION:

1. Your scattering geometry (`.scuffgeo` file)
2. Frequency or Frequency range (ω)
3. Incident angle or Incident angle range (θ)

Output produced by SCUFF-TRANSMISSION:

- Transmission and reflection coefficients (magnitudes only) vs. ω and θ.

Plane wave impinging from below on a dielectric film of thickness T:



Exact transmission and reflection coefficients (normal incidence, $\mu^r = 1, n = \sqrt{\epsilon^r}$):

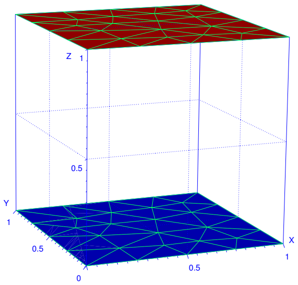
$$t(\omega) = \frac{2in}{(1+n^2)\sin(nk_0T) + 2in\cos(nk_0T)}, \qquad r(\omega) = \frac{(1-n^2)\sin(nk_0T)}{(1+n^2)\sin(nk_0T) + 2in\cos(nk_0T)}$$

SCUFF-TRANSMISSION

Running the thin dielectric film in SCUFF-TRANSMISSION

Create **(1)** a SCUFF-EM geometry for the thin film, and **(2)** a list of frequencies:

Unit cell mesh



ThinFilm_58.scuffgeo

```
LATTICE
        VECTOR 1 0
        VECTOR 0 1
ENDLATTICE

REGION Exterior    MATERIAL Vacuum
REGION ThinFilm    MATERIAL CONST_EPS_100
REGION UpperSpace  MATERIAL Vacuum

SURFACE LowerFilmSurface
 MESHFILE Square_58.msh
 REGIONS Exterior ThinFilm
ENDSURFACE

SURFACE UpperFilmSurface
 MESHFILE Square_58.msh
 DISPLACED 0 0 1
 REGIONS ThinFilm UpperSpace
ENDSURFACE
```

Omega.dat

```
0.1
0.2
0.3
...
1.0
```

Now solve the problem using SCUFF-TRANSMISSION:

```
% scuff-transmission --geometry ThinFilm_58.scuffgeo --OmegaFile Omega.dat
```

The SCUFF-TRANSMISSION run generates a file named ThinFilm_58.transmission.

0.1	0.0	4.7e-02	9.5e-01	4.7e-02	9.5e-01
0.2	0.0	1.0e-01	8.9e-01	1.0e-01	8.9e-01
0.3	0.0	9.9e-01	8.2e-03	9.9e-01	8.2e-03
0.4	0.0	6.8e-02	9.3e-01	6.8e-02	9.3e-01
0.5	0.0	4.2e-02	9.5e-01	4.2e-02	9.5e-01

The SCUFF-TRANSMISSION run generates a file named ThinFilm_58.transmission.

0.1	0.0	4.7e-02	9.5e-01	4.7e-02	9.5e-01
0.2	0.0	1.0e-01	8.9e-01	1.0e-01	8.9e-01
0.3	0.0	9.9e-01	8.2e-03	9.9e-01	8.2e-03
0.4	0.0	6.8e-02	9.3e-01	6.8e-02	9.3e-01
0.5	0.0	4.2e-02	9.5e-01	4.2e-02	9.5e-01

angular frequency

The SCUFF-TRANSMISSION run generates a file named ThinFilm_58.transmission.

0.1	0.0	4.7e-02	9.5e-01	4.7e-02	9.5e-01
0.2	0.0	1.0e-01	8.9e-01	1.0e-01	8.9e-01
0.3	0.0	9.9e-01	8.2e-03	9.9e-01	8.2e-03
0.4	0.0	6.8e-02	9.3e-01	6.8e-02	9.3e-01
0.5	0.0	4.2e-02	9.5e-01	4.2e-02	9.5e-01

incident angle

The SCUFF-TRANSMISSION run generates a file named `ThinFilm_58.transmission`.

0.1	0.0	4.7e-02	9.5e-01	4.7e-02	9.5e-01
0.2	0.0	1.0e-01	8.9e-01	1.0e-01	8.9e-01
0.3	0.0	9.9e-01	8.2e-03	9.9e-01	8.2e-03
0.4	0.0	6.8e-02	9.3e-01	6.8e-02	9.3e-01
0.5	0.0	4.2e-02	9.5e-01	4.2e-02	9.5e-01

$|t_{\rm perp}|^2$ $|r_{\rm perp}|^2$

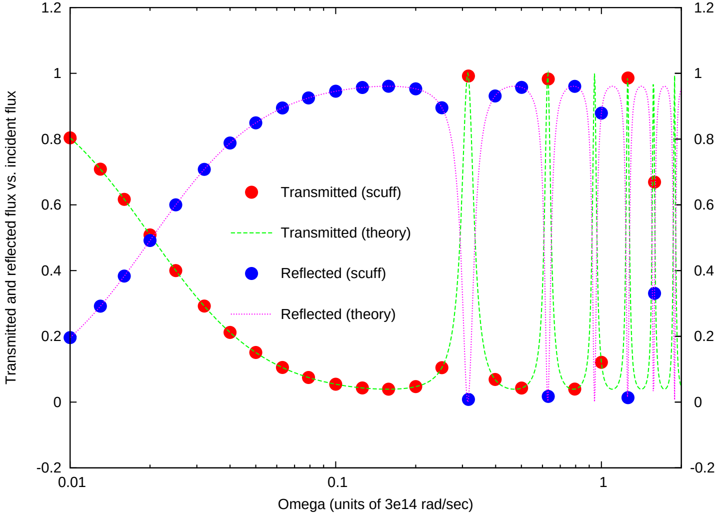
The SCUFF-TRANSMISSION run generates a file named ThinFilm_58.transmission.

0.1	0.0	4.7e-02	9.5e-01	4.7e-02	9.5e-01
0.2	0.0	1.0e-01	8.9e-01	1.0e-01	8.9e-01
0.3	0.0	9.9e-01	8.2e-03	9.9e-01	8.2e-03
0.4	0.0	6.8e-02	9.3e-01	6.8e-02	9.3e-01
0.5	0.0	4.2e-02	9.5e-01	4.2e-02	9.5e-01

$|t_{\rm par}|^2$ $\qquad$ $|r_{\rm par}|^2$

SCUFF-TRANSMISSION

SCUFF-EM vs. exact calculation for thin dielectric film



SCUFF-TRANSMISSION

Extraordinary optical transmission through a perforated metallic film

VOLUME 86, NUMBER 6 PHYSICAL REVIEW LETTERS 5 FEBRUARY 2001

Theory of Extraordinary Optical Transmission through Subwavelength Hole Arrays

L. Martín-Moreno,[1] F. J. García-Vidal,[2] H. J. Lezec,[3] K. M. Pellerin,[4] T. Thio,[4] J. B. Pendry,[5] and T. W. Ebbesen[3]

[1]*Departamento de Física de la Materia Condensada, ICMA-CSIC, Universidad de Zaragoza, E-50015 Zaragoza, Spain*

[2]*Departamento de Física Teórica de la Materia Condensada, Universidad Autónoma de Madrid, E-28049 Madrid, Spain*

[3]*ISIS, Université Louis Pasteur, 67000 Strasbourg, France*

[4]*NEC Research Institute, Princeton, New Jersey 08540*

[5]*The Blackett Laboratory, Imperial College, London SW7 2BZ, United Kingdom*

(Received 14 August 2000)

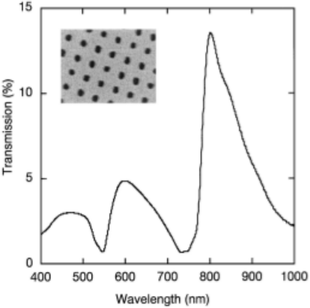
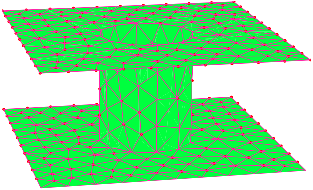


FIG. 1. Experimental zero-order power transmittance, T_{00}, at normal incidence for a square array of holes (lattice constant $L = 750$ nm, average hole diameter of 280 nm) in a freestanding Ag film (thickness $h = 320$ nm). Inset: electron micrograph of the perforated metal film.

Create **(1)** a SCUFF-EM geometry for the thin film, and **(2)** a list of frequencies:

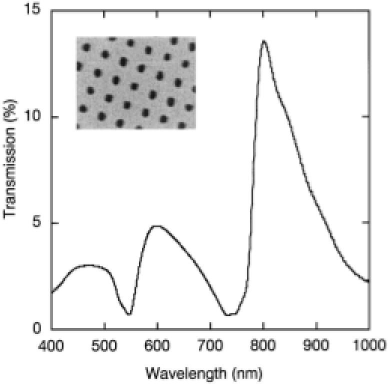
Unit cell mesh	PTF_794.scuffgeo	Omega.dat
	LATTICE VECTOR 0.75 0 VECTOR 0 0.75 ENDLATTICE OBJECT UnitCell MESHFILE PTFUnitCell_794.msh MATERIAL Gold ENDOBJECT	0.1 0.2 0.3 ... 1.0

Now solve the problem using SCUFF-TRANSMISSION:

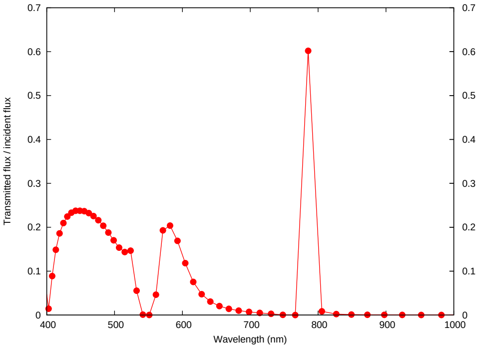
```
% scuff-transmission --geometry PTF_794.scuffgeo --OmegaFile Omega.dat
```

SCUFF-TRANSMISSION

Perforated metallic film: published data vs. SCUFF-TRANSMISSION



Martin-Moreno et al. (data)



SCUFF-TRANSMISSION

SCUFF-TRANSMISSION

Perforated metallic film: published theory vs. SCUFF-TRANSMISSION

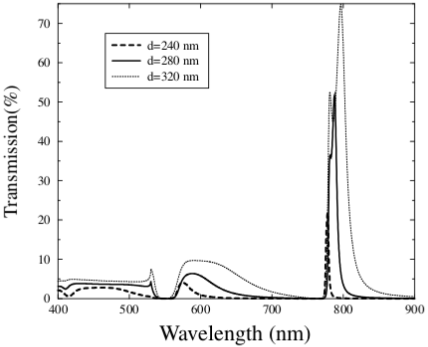
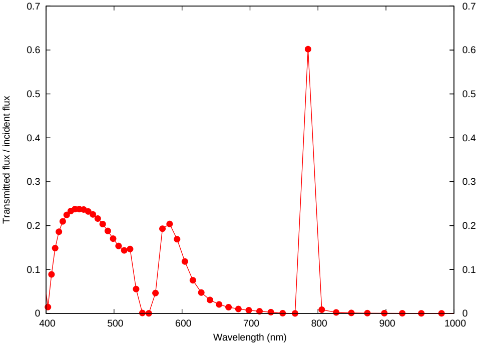


FIG. 2. Calculated T_{00} at normal incidence for an array of holes in a Ag film, defined by $L = 750$ nm, $h = 320$ nm, and three different hole side lengths d.

Martin-Moreno et al. (theory)



SCUFF-TRANSMISSION

Command-line applications in the SCUFF-EM suite

Electromagnetic Scattering

- SCUFF-SCATTER: general-purpose scattering
- SCUFF-TRANSMISSION: plane-wave transmission through extended structures
- SCUFF-TMATRIX: spherical-basis T-Matrix of compact objects

RF / Microwave Device Engineering

- SCUFF-RF: Circuit parameters and radiated fields of passive RF devices

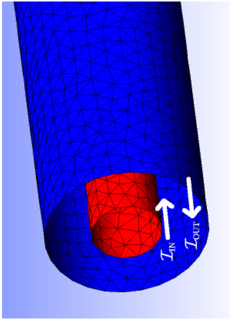
Fluctuation Physics

- SCUFF-CAS3D: Casimir energy, force, torque in 3D geometries
- SCUFF-CAS2D: Casimir energy, force, torque in 2D geometries
- SCUFF-CASPOL: Casimir-Polder potentials for polarizable particles near surfaces
- SCUFF-NEQ: Nonequilibrium fluctuations: Radiative heat transfer & non-EQ Casimir forces

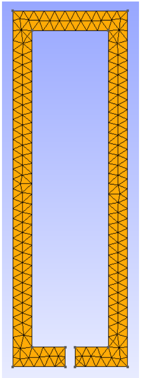
SCUFF-RF: Modeling of passive RF devices

SCUFF-RF is designed to model devices like this:

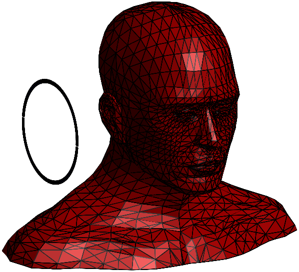
Coaxial cable



Planar antenna

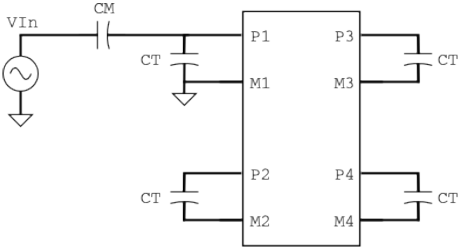


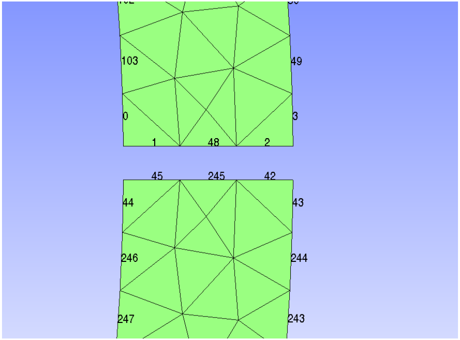
MRI Coil



Port definitions in SCUFF-RF

To interface a passive structure with an electric circuit, define *ports*.





A current forced into a port defines a new type of incident field for the BEM scattering problem.

SCUFF-RF Inputs and outputs

Network parameter mode:

Inputs to SCUFF-RF:

1. Your scattering geometry (`.scuffgeo` file)
2. Port definitions
3. Frequency or Frequency range

Output returned by SCUFF-RF:

- Network parameters (S or Z parameters) for your multiport structure

Radiated field mode:

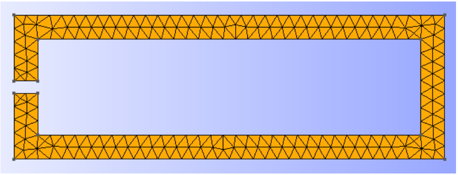
Inputs to SCUFF-RF:

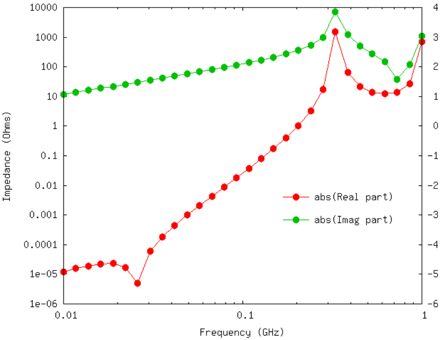
1. Your scattering geometry (`.scuffgeo` file)
2. Port definitions
3. Port currents
4. List of field evaluation points

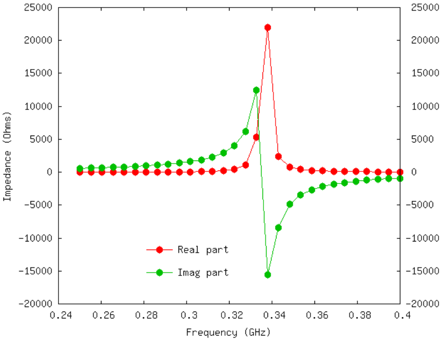
Output returned by SCUFF-RF:

- Fields at evaluation points radiated by your structure as driven by the specified port currents.

SCUFF-RF: Input impedance of a planar antenna







1. Generate surface meshes for all object surfaces in your geometry.

2. Write a SCUFF-EM geometry file describing objects and materials.

3A. Run one of the 8 standalone command-line applications bundled with the SCUFF-EM suite.

OR

3B. Write your own C++ or PYTHON code using the SCUFF-EM core library API.

C++ API to the SCUFF-EM Core Library

Offers maximal flexibility and customization.

```
#include <libscuff.h>

int main(...)
 {
   // read in the .scuffgeo file
   RWGGeometry *G = new RWGGeometry("MyGeometry.scuffgeo");

   // assemble the BEM matrix
   cdouble Omega = 1.2 + 3.4i;      // angular frequency
   HMatrix *M = G->AssembleBEMMatrix(Omega, M);

   // assemble the RHS vector for a plane-wave incident field
   double PWDir[3] =  { 0.0, 0.0, 1.0 };   // plane wave direction
   cdouble PWPol[3] = { 1.0, 0.0, 0.0 };   // plane wave polarization
   PlaneWave PW(PWDir, PWPol);
   HVector *KN = G->AssembleRHSVector(Omega, &PW);

   // solve the BEM system
   M->LUFactorize();
   M->LUSolve(KN);

   // compute scattered fields at the origin
   double X[3] = {0.0, 0.0, 0.0};
   G->GetFields(&PW, KN, Omega, X, EH);
   printf("E_x at origin = (%e,%e)\n",real(EH[0]),imag(EH[0]));
 };
```

Also available: PYTHON interface (thanks to Steven and Johannes)

Outline

1. A quick review of the Boundary-Element Method

2. A brief history of the evolution of SCUFF-EM

3. **What SCUFF-EM can do**
    1. Inputs: The geometries, materials, incident fields that SCUFF-EM can handle
    2. Outputs: The various calculations that SCUFF-EM can do
    3. Mechanics: How to run SCUFF-EM

4. Under the hood: How SCUFF-EM works

Why is it so hard to assemble the BEM matrix?

Consider a scattering geometry with surfaces discretized into $N \sim 10,000$ triangles.

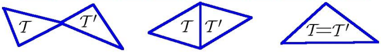
1. We have N^2=100 million matrix elements.
2. Each matrix element involves a 4 dimensional integral (surface integrals over two triangles) that must be evaluated numerically.
3. A sizeable fraction of these are singular integrals.

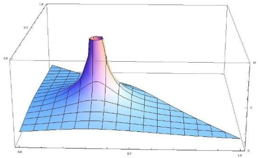
$$M_{mn} = \Big\langle \mathbf{f}_m \Big| \mathbf{G} \Big| \mathbf{f}_n \Big\rangle$$

$$\mathbf{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} & \cdots & M_{1N} \\ M_{21} & M_{22} & M_{23} & \cdots & M_{2N} \\ M_{31} & M_{32} & M_{33} & \cdots & M_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{N1} & M_{N2} & M_{N3} & \cdots & M_{NN} \end{pmatrix}$$

10,000 (rows) × 10,000 (columns)

$$\int_{\mathcal{T}} d\mathbf{x} \int_{\mathcal{T}'} d\mathbf{x}'\, h(\mathbf{x},\mathbf{x}') g\big(|\mathbf{x}-\mathbf{x}'|\big)$$





Evaluate singular (or nearly-singular) PPIs using a two-step process.

Consider

$$\mathcal{I} = \int_{\mathcal{T}} d\mathbf{x} \int_{\mathcal{T}'} d\mathbf{x}' \, h(\mathbf{x},\mathbf{x}') \frac{e^{ikR}}{4\pi R} \qquad \Big(R \equiv |\mathbf{r}-\mathbf{r}'|\Big) \qquad \text{singular when } \mathbf{x}=\mathbf{x}'$$

$$= \underbrace{\int_{\mathcal{T}} d\mathbf{x} \int_{\mathcal{T}'} d\mathbf{x}' \, h(\mathbf{x},\mathbf{x}') \frac{\Big\{ e^{ikR} - 1 - ikR - \frac{1}{2}(ikR)^2 - \frac{1}{6}(ikr)^3 \Big\}}{4\pi R}}_{\text{nonsingular, so easy to evaluate}}$$

$$+ \sum_{p=0}^{3} C_p (ik)^p \underbrace{\int_{\mathcal{T}} d\mathbf{x} \int_{\mathcal{T}'} d\mathbf{x}' \, \frac{h(\mathbf{x},\mathbf{x}')}{R^p}}_{\text{singular but } k\text{-independent!}}$$

$\Longrightarrow$ 1. Evaluate singular PPIs *once* per structure

2. Store in `.scuffcache` files

3. Reuse at all frequencies and for subsequent computations.

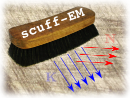
Caching in SCUFF-EM

All application codes in the SCUFF-EM suite have a --cache option.

```
scuff-scatter ... --cache MyGeometry.cache
```

```
scuff-rf ... --cache MyGeometry.cache
```

Cached data depend only on the mesh, *not* on material or frequency.



SCUFF-EM website: http://homerreid.com/scuff-em

m t homer reid

physics problems | research | music | about me | miscellany

research

Codes

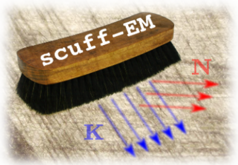
- SCUFF-EM
- Quantum Chemistry
- Numerical Libraries

Talks

- Thesis Defense
- Zeta Function
- Intro Quantum Chem
- IFC Poster 12/05
- CNTFET Modeling

Memos

- PhD Thesis
- BEM Tutorial
- 1D Waveguide Tapering
- Debye-Huckel



SCUFF-EM: Free, open-source software for boundary-element analysis of problems in computational physics and engineering

SCUFF-EM

- Installation
- Core Library
  - LIBSCUFF
  - Main Flow Routines
  - Ancillary Routines
  - Incident Fields
  - Matrices and Vectors
  - C++ examples
  - PYTHON examples

SCUFF-EM is a free, open-source software package for analysis of electromagnetic scattering problems using the boundary-element method (BEM). (The BEM is also known as the "method of moments.")

The SCUFF-EM suite consists of two components: a *core library* that implements the essential algorithms of the boundary-element method, and a set of *application programs* built atop the core library for solving specific problems in various fields of physics and engineering

SCUFF-EM website:

http://homerreid.com/scuff-em

Thanks!