



Advances in OpenSSP

Kwo-Sen Kuo^{1,2}, Matthew Lammers^{1,3}, Ines Fenni^{4,5}, Craig Pelissier^{1,6}, Thomas Clune¹, Bruce van Aartsen^{1,6}, and Ian Adams¹

¹NASA Goddard Space Flight Center, Greenbelt, MD; ²ESSIC, Univ. of Maryland, College Park, MD; ³SGT, Inc., Greenbelt, MD; ⁴NASA Jet Propulsion Laboratory, Pasadena, CA; ⁵Univ. of California, Los Angeles, CA; ⁶SSAI, Lanham, MD

OpenSSP Portal **Browser Interface** **Web Service API**

Latest implement: Particle selection constrained by both PSD and $m-D$ relation. • Use of slide bars to ensure parameter values within reasonable range. • Interactive graphs supporting drill-down for details. • Normalized distribution with normalized water content (1 g m^{-3}) for easy scaling. • Query results in tab separated values (tsv) format for easy import into spreadsheet software.

Web Service Application Programming Interface (API)

Web service API capability supports facilitate access to the OpenSSP database to scriptable more automated analysis of the data.

```
import urllib
import numpy as np
import matplotlib as m
m.use('Agg') #because I am doing this on a server with no video drivers
import matplotlib.pyplot as plt
urlObject =
urllib.urlopen('https://storm.pps.eosdis.nasa.gov/storm/'
'%OpenSSPAPI?email=matthew.lammers@nasa.gov' +
'%frequency=089.062GHz&size=p-40/all&text=true')
"""Looking at the relationship between maximum dimension and
scattering/absorption efficiency."""
osspsArray = np.loadtxt(urlObject,comments='#',unpack=True,
skiprows=2,usecols=(4,8,9))
print osspsArray
plt.plot(osspsArray[0],osspsArray[1],ro,'label='Scattering Efficiency')
plt.plot(osspsArray[0],osspsArray[2],bo,'label='Extinction Efficiency')
plt.legend()
plt.xlabel('Maximum Dimension [um]')
plt.ylabel('Efficiency')
plt.savefig('ScattAbs.png')
```

Particle Simulations

Polycrystal Growth



The original *monocrystal Snowflake* model of Gravner and Griffiths (2009) is extended with multilattice capability to simulate polycrystal growth, such as the column rosette show above.

Figure #	Picture	Max # Iter.	Old Volume (.att)	New Volume (pov_*.dat)	Fraction	Factor
GG 01		43,000	6.2 GB	110 MB	.0177	56x
GG 04		277,500	420 GB	633 MB	.00151	664x
GG 06		287,000	365 GB	670 MB	.00183	544x
GG 07		150,000	121 GB	476 MB	.00393	254x
GG 08		78,500	4.6 GB	35 MB	.0076	131x
GG 11		1,200,000	47 GB	64 MB	.00136	734x
GG 13		123,000	121 GB	292 MB	.00241	414x
GG 14		166,500	432 GB	789 MB	.00183	548x
GG 15		200,000	250 GB	831 MB	.00332	301x
SUM			1767 GB	3,900 MB	.00221	453x

Storage Efficient Particle Structure Output

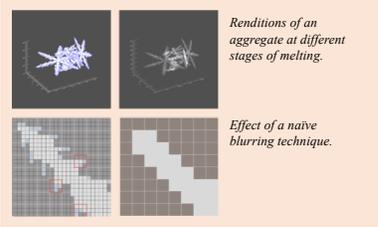
Previously, a snapshot of the particle structure is taken every 500 iterations during the growth simulation and saved as 3D byte arrays. As now the domain-decomposed parallelization code allows us to grow the crystals larger, i.e. more iterations, the storage requirement increases drastically. We had to find a more efficient way of saving the crystal structures for scattering calculations later. The table to the right demonstrate the improved efficiency of our new output format.

Scattering Calculations

Issues

- Number of orientations and the need for polarization quantities drive up the storage requirement.
- It is nearly difficult to achieve optimal computational efficiency while maintain particle structure consistency at the same time across a wide range of frequencies.

Although the popular discrete dipole approximation (DDA) methods, such as DDSCAT and ADDA, can handle general scatterer geometry, they cannot deal with these two issues effectively.



Issue 2

As the name indicates, DDA approaches are based on *discrete dipoles*. The computing time of DDA depends significantly on the number of dipoles.

The larger the particle relative to wavelength, the more dipoles are needed, i.e. inversely proportional to frequency. In other words, for the same particle, we could use dipoles of larger size at a lower frequency, which means we could "blurred" the particle structure. However, this blurring also changes the particle morphology, as demonstrated in the 4-panel image above. This shape deviation is especially pronounced when multiple compositions are involved, e.g. melting particles.

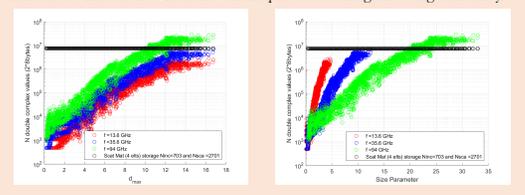
The way to avoid this modification to particle structure but also reap benefit from smaller number of dipoles is to use adaptive grid, i.e. use larger dipoles where we can but also smaller ones to maintain particle geometry. However, DDA approaches do not lend themselves to this approach.

Issue 1

The number of orientations needed for the orientation-averaged backscatter efficiency to converge grows with the complexity of the scatterer's geometry and size. However, a guideline as to how many orientations are necessary does not currently exist. It becomes a business of trial-and-error, which is particularly wasteful when the size parameter is large and computationally demanding.

Since polarization signal contains information of particle orientation, we need to save polarimetric information of the scattering solution as well, in addition to just the intensity information; that is, a multielement matrix of complex numbers instead of a real scalar. The storage requirement is thus an order or two greater!

The following figure contrasts the number of complex numbers that need to be stored for DDSCAT (black) solutions and CBFM (colored) solutions for a fixed number of orientations. CBFM has an unequivocal advantage in storage efficiency.

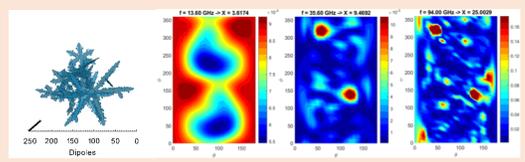


Characteristic Basis Function Method, CBFM

CBFM shows the best potential to effectively address both issues.

- It is versatile with scatterer geometry and composition like DDA approaches.
- It has demonstrated better efficiency, in both computation and storage, over DDA.
- It is amenable to multi-resolution adaptive grid.

Currently only the last bullet above requires further development.



CBFM solutions for backscatter efficiency at Ku, Ka, and W bands for an aggregate with a water equivalent diameter of 3 mm.