Discrete-dipole approximation

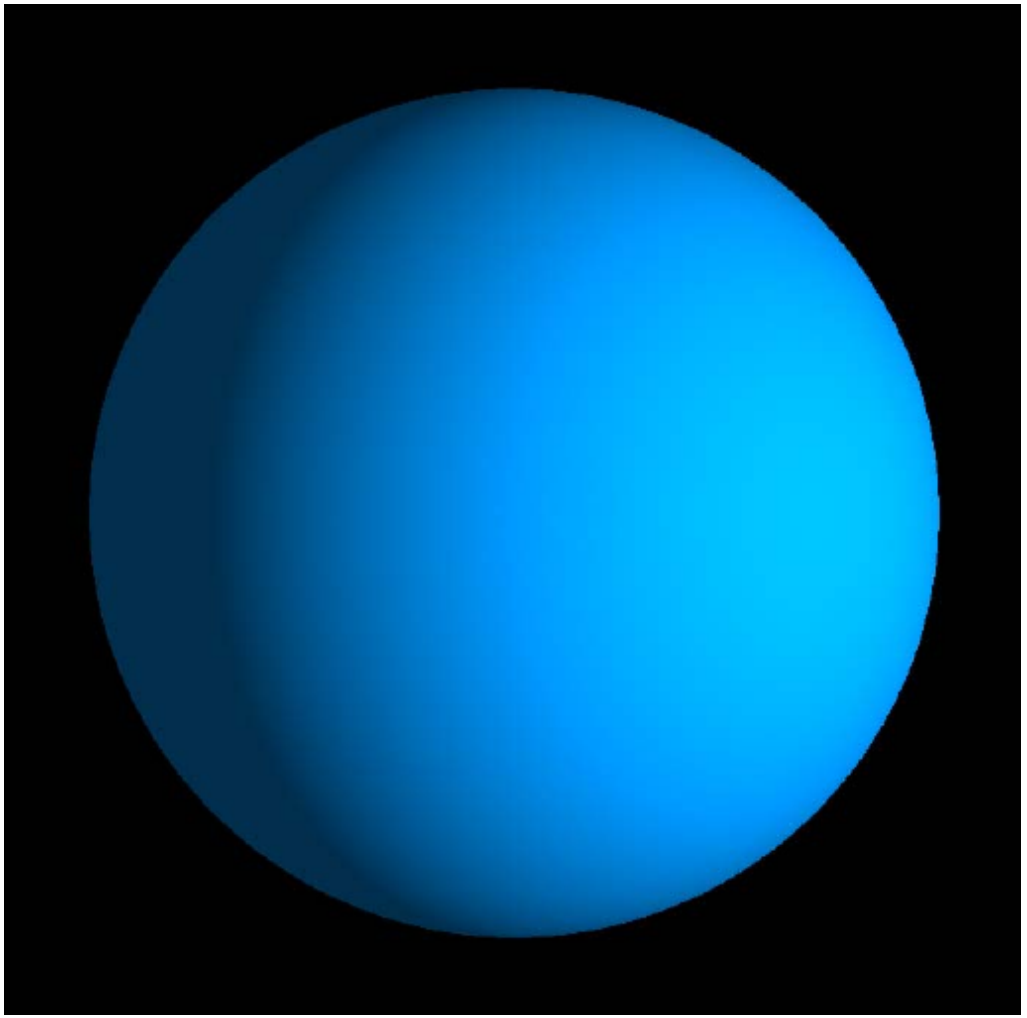
Jani Tyynelä

Electromagnetic Scattering I, 2014

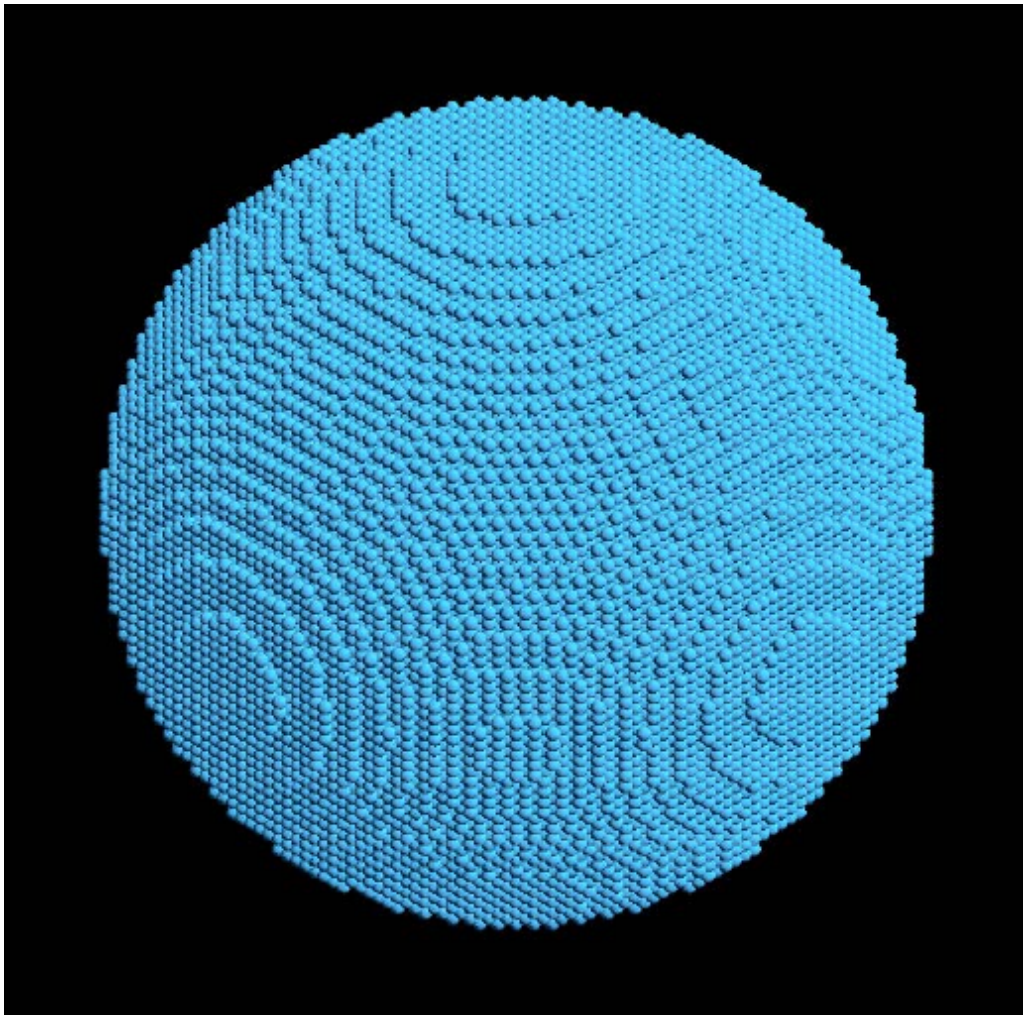
Theory and numerical methods

Solid matter is composed of polarizable atoms/molecules. In DDA, the particle is approximated as a cubic array of small volume elements that can electromagnetically interact with each other. There are no restrictions in shape or internal structure.

perfect sphere



DDA sphere



DDA formulation starts with the electric field of a dipole, which describes the EM behavior of small particles. Dipole responds to an external/incident field by starting to oscillate in the same frequency and immediately re-radiating in all directions.

The relationship between the incident and scattered field:

$$\mathbf{E}^{\mathrm{sca}} = \frac{\exp(\mathrm{i}kr)}{r}\alpha\left(k^2[\hat{\mathbf{e}}_r \times [\hat{\mathbf{e}}_r \times \mathbf{E}^{\mathrm{inc}}]] - \frac{1-\mathrm{i}kr}{r^2}\left(3(\hat{\mathbf{e}}_r \cdot \mathbf{E}^{\mathrm{inc}})\hat{\mathbf{e}}_r - r^2\mathbf{E}^{\mathrm{inc}}\right)\right)$$

spherical wave **far field** **near field**

k is wavenumber
r is distance
α is polarizability

The dipole moment: $\mathbf{p} = \alpha\mathbf{E}^{\mathrm{inc}}$

To compute the scattered field from a collection of dipoles as in DDA, the induced electric fields are first derived:

$$\mathbf{E}_i = \mathbf{E}_i^{\mathrm{inc}} + \sum_{j \neq i}^{N} \mathbf{G}_{ji}\mathbf{E}_j = \mathbf{E}_i^{\mathrm{inc}} + \sum_{j \neq i}^{N} \alpha_j \frac{\exp(\mathrm{i}kr_{ji})}{r_{ji}^3}\Big(k^2[\mathbf{r}_{ji} \times [\mathbf{r}_{ji} \times \mathbf{E}_j]] - \frac{1 - \mathrm{i}kr_{ji}}{r_{ji}^2}\big(3(\mathbf{r}_{ji} \cdot \mathbf{E}_j)\mathbf{r}_{ji} - r_{ji}^2\mathbf{E}_j\big)\Big)$$

Superposition of incident field and the scattered (near)field from the other dipoles.

The most time-consuming part in DDA is to numerically solve the induced fields from N equations → Iterative solvers are typically used, such as conjugate gradient or quasi-minimal residual methods with preconditioners.

Once the induced fields are solved, the scattered (far)field and other properties can be obtained:

$$\mathbf{E}^{\mathrm{sca}} \approx \frac{\exp(\mathrm{i}kr_{\mathrm{obs}})}{-ikr_{\mathrm{obs}}}\mathbf{F}_{i,\mathrm{obs}} = \frac{\exp(\mathrm{i}kr_{\mathrm{obs}})}{-\mathrm{i}kr_{\mathrm{obs}}}(-\mathrm{i}k^3)\sum_{i=1}^{N}\alpha_i \exp(-\mathrm{i}k(\hat{\mathbf{e}}_r \cdot \mathbf{r}_i))[\hat{\mathbf{e}}_r \times [\hat{\mathbf{e}}_r \times \mathbf{E}_i]]$$

F is called scattering amplitude

spherical wave **phase factor** **far field**

$$\sigma_{\mathrm{sca}} = \frac{1}{k^2}\oint d\Omega |\mathbf{F}|^2 \qquad \sigma_{\mathrm{abs}} = 4\pi k \sum_{i=1}^{N}\left(\Im(\alpha_i \mathbf{E}_i^{\mathrm{inc}} \cdot \mathbf{E}_i^{\mathrm{inc}*}) - \frac{2}{3}k^3 |\alpha_i \mathbf{E}_i^{\mathrm{inc}}|^2\right)$$

Far field components: $\hat{\mathbf{e}}_r \times [\hat{\mathbf{e}}_r \times \mathbf{E}_i] = (\hat{\mathbf{e}}_r \cdot \mathbf{E}_i)\hat{\mathbf{e}}_r - \mathbf{E}_i$

In practice, due to matrix-vector products (N^2), DDA is enhanced by the fast-Fourier transform (NlogN), which requires cubic, evenly-spaced grid of volumes.

$$\mathbf{E}_{i,j,l} = \mathbf{E}^{\mathrm{inc}}_{i,j,l} + \sum_{m=1}^{N_x}\sum_{n=1}^{N_y}\sum_{o=1}^{N_z}\mathbf{G}^{\mathrm{3D}}_{i,j,l;m,n,o}\mathbf{E}_{m,n,o}$$

In the limit of ideal dipoles in lattice ($kd \to 0$), Clausius-Mossotti formula is used for polarizability:

$$\alpha_{\mathrm{CM}} = \frac{3V}{4\pi}\frac{m^2-1}{m^2+2}, V = d^3$$

Corrections for finite, cubical dipoles: $\alpha = \dfrac{\alpha_{\mathrm{CM}}}{1 - M\alpha_{\mathrm{CM}}/V}$

Lattice dispersion relation: $M_{\mathrm{LDR}} = (b_1 + b_2 m^2 + b_3 m^2 S)(kd)^2 + \dfrac{2\mathrm{i}}{3}(kd)^3$

Filtered coupled dipoles: $M_{\mathrm{FCD}} = \dfrac{4}{3}(kd)^2 + \dfrac{2}{3}\left(\mathrm{i} + \dfrac{1}{\pi}\ln\dfrac{\pi - kd}{\pi + kd}\right)(kd)^3$

Limitations and requirements

Accuracy

Accuracy in the scattering properties:

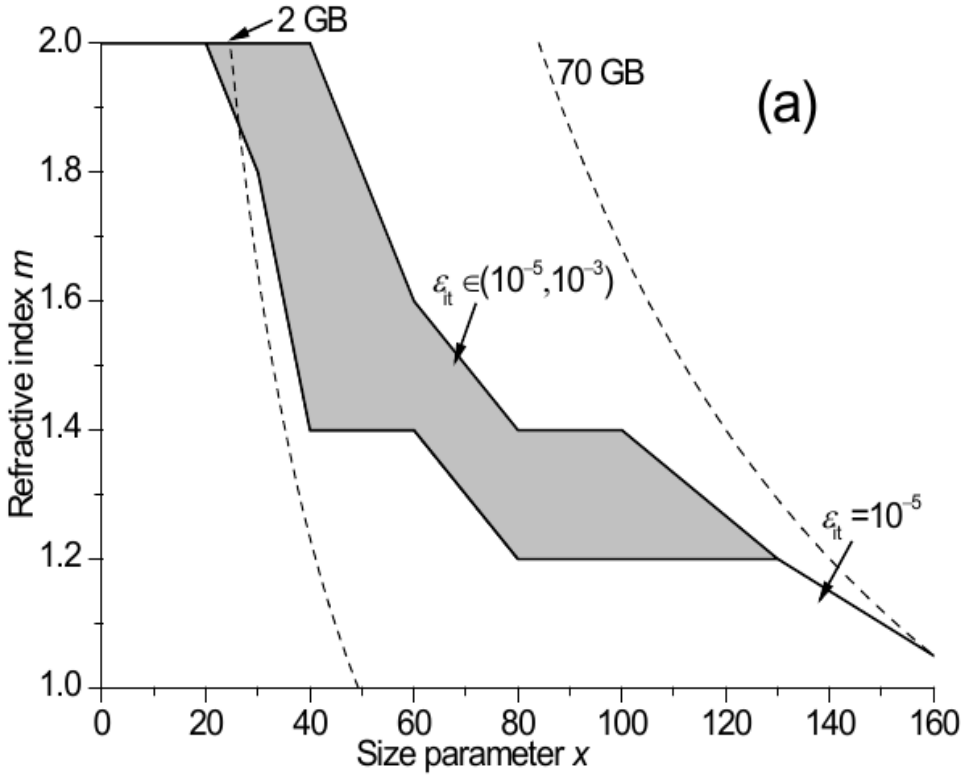
1. Shape → number of dipoles N must be large enough

2. Dipole assumption, phase: $|m|kd < 1$

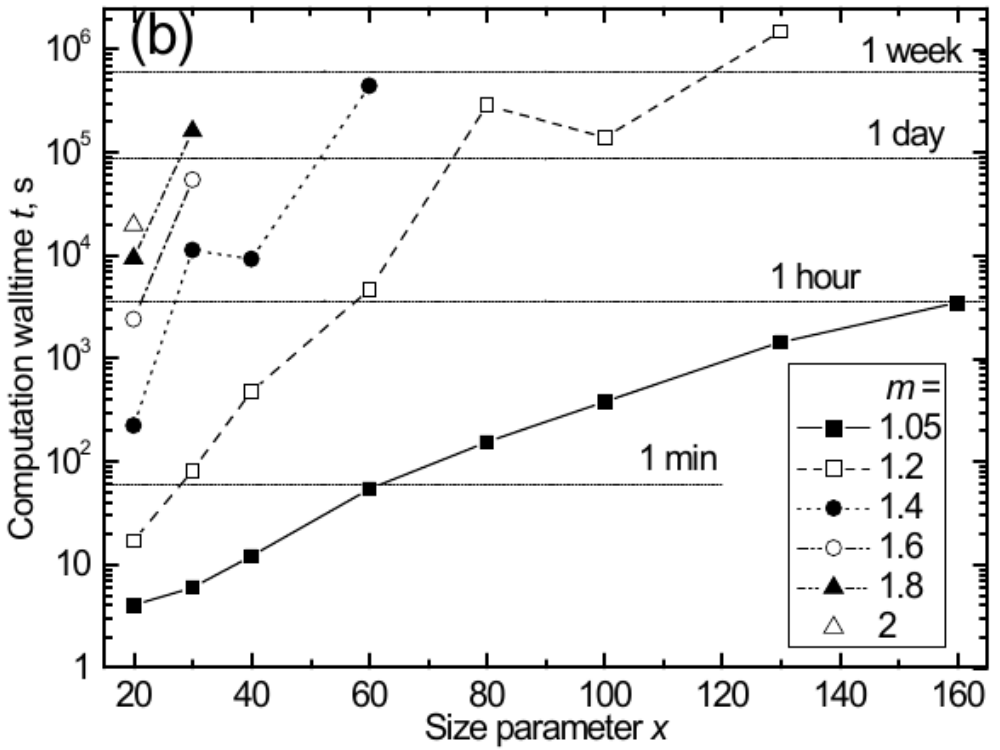
3. Dielectric (can be enhanced): $|m| < 2$

Computational resources

DDA is mainly restricted by computer resources:



memory



CPU time

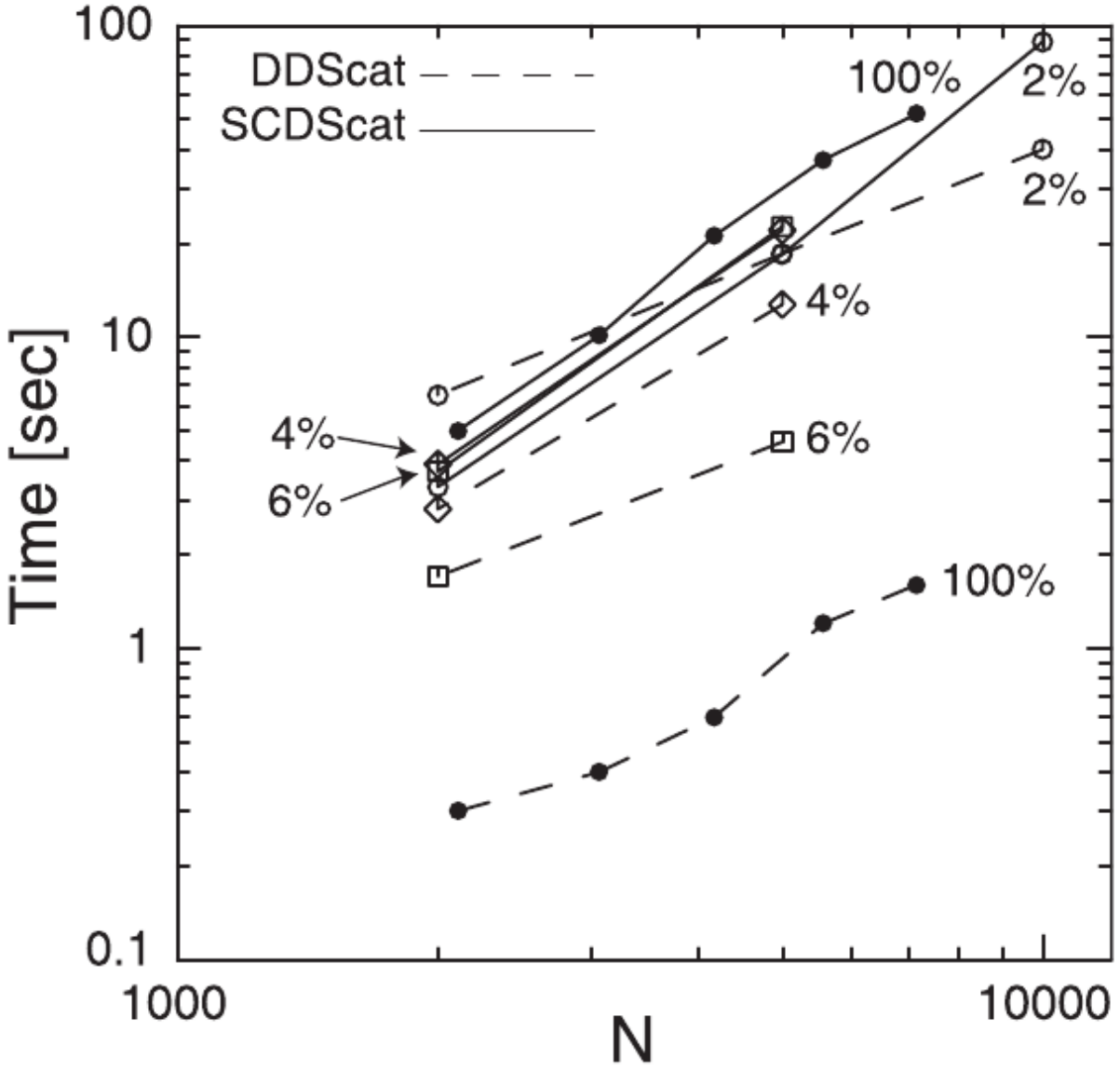
from A-DDA manual

Fast-Fourier transform

FFT reserves memory for the whole grid.

Fluffy, porous particle only has a small fraction of real dipoles.

→ 'Sparse' DDA without FFT can save memory by a factor of volume fraction ($N/N_x N_y N_z$).



from Petty and Huang (2010)

Conductivity

Conducting particles can be problematic for DDA due to the $|m|kd < 1$ requirement.

Polarizability definitions and grid size can be improved.

Skin depth:

$$kl_{\text{skin}} = \frac{1}{\sqrt{\Re(m)\Im(m)}}$$

Case		Relative error (%)		
Grid	FCD	σ_{back}	σ_{abs}	σ_{ext}
64^3	No	4.09	10.91	7.86
	Yes	2.20	2.68	3.24
128^3	No	1.56	3.73	2.97
	Yes	1.10	1.31	1.60
256^3	No	0.94	1.33	1.36
	Yes	0.55	0.65	0.79
384^3	No	0.67	0.77	0.89
	Yes	0.37	0.43	0.53

Spherical 8 mm water drop at 5.6 GHz:
x = 1, m = 8.34+i2.22

from Tyynelä et al. (2009)

Conductivity

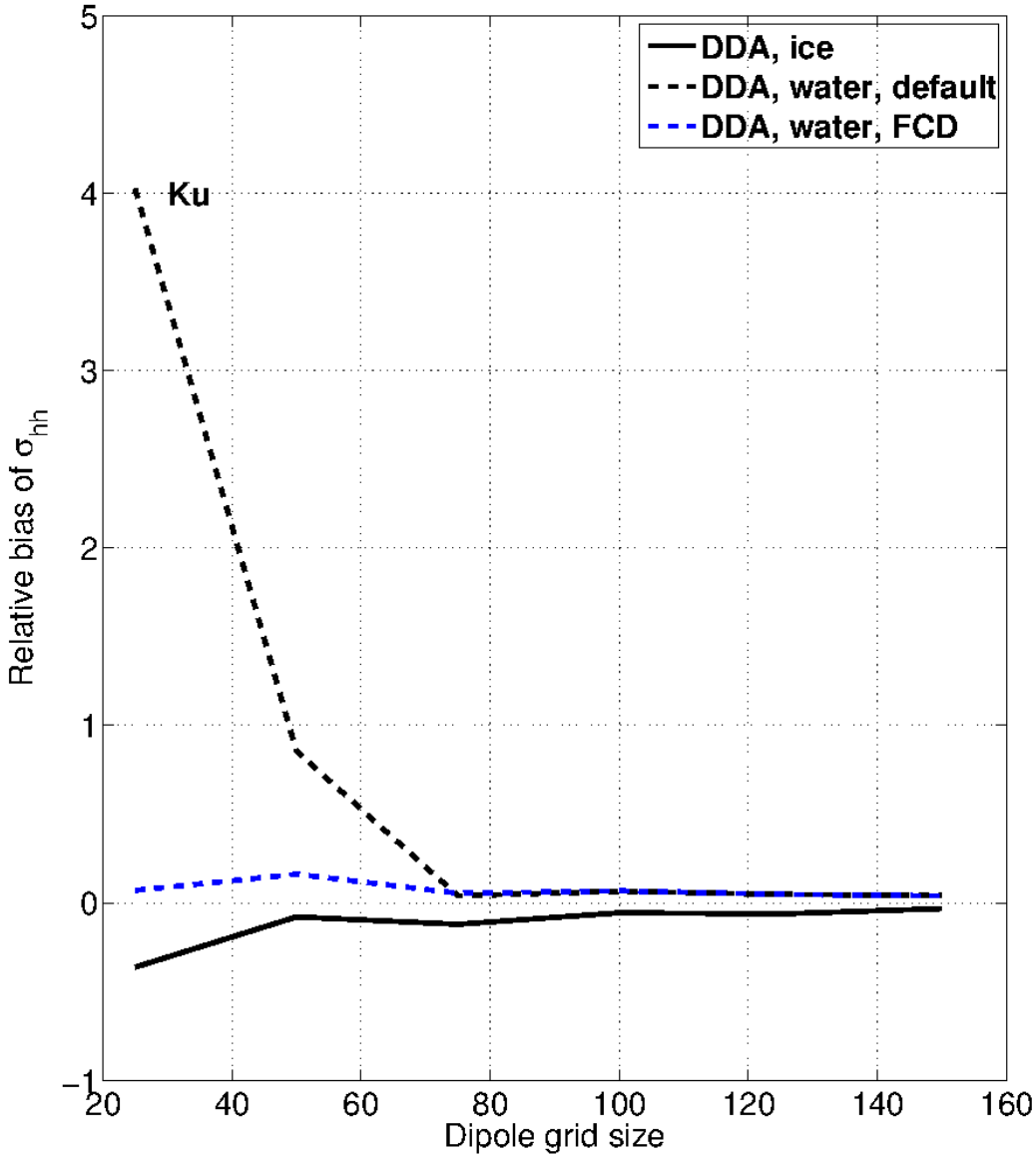
Spherical 5 mm water drop at 13.6 GHz:
x = 0.7, m = 6.27+i3.0

→ l_{skin} = 0.8 mm

Using default (LDR) polarizability:

→ N_{grid} > 75

→ d = 5 mm/N_{grid} < 0.07 mm = l_{skin}/12

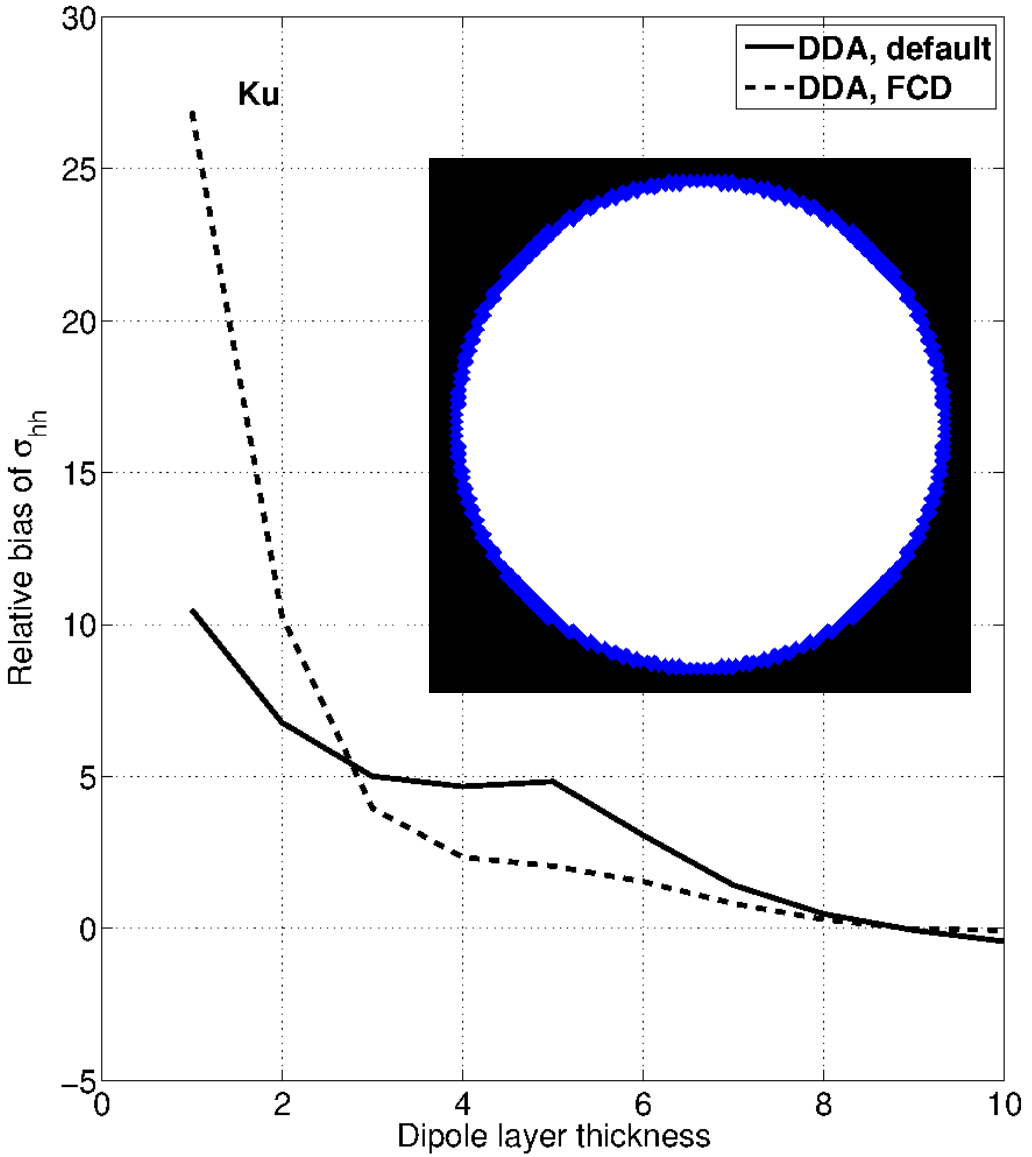


Conductivity

Conductive coating is also problematic due to the particle scaling.

Spherical 3 cm hail at 13.6 GHz:
x = 4.3
m_{ice} = 1.7844
m_{water} = 6.27+i3.0
4% water coating

→ l_{layer} > 9d



Convergence

Numerical convergence:

- iterative solvers do not always converge (qmr vs. cgnr)
- large size parameters and refractive indices produce convergence difficulties

Accuracy convergence:

- the accuracy of the scattering properties depend on the number of averaged shapes and orientations
- better to test than to assume

Introduction to A-DDA

- open source DDA code for Windows and Linux
- available at: https://code.google.com/p/a-dda/
- main developer: Maxime Yurkin
- comprehensive manual
- online support
- parallelizes well on a super-cluster
- relatively fast development schedule
- listen to user feedback
- many choices for iterative solvers
- many out-of-the-box shapes included
- better than DDSCAT: command line parameters, polarizability choices, separation of dipole positions and ref. indices, sparse mode, OpenCL mode, surface mode, amplitude scattering matrix output, internal fields output
- worse than DDSCAT: periodic boundaries

Example serial run:

```
jktyynel@dx0-flatm-01:~$ ./adda -lambda 1.110342E5 -m 1.7748 1.0920E-4 -dpl 5.5 -orient 0.0 90.0 0.0 -shape read ice_aggregate0001.shape
-iter cgnr -pol fcd -int fcd -scat_matr ampl -Csca -asym -store_int_field
```

Example parallel run:

```
#!/bin/csh
#SBATCH -J adda
#SBATCH -e adda.err%j
#SBATCH -o adda.out%j
#SBATCH --mem-per-cpu=1000
#SBATCH -t 96:00:00
#SBATCH -n 96
#SBATCH -p parallel

module swap PrgEnv-pgi PrgEnv-intel
module load fftw
cd $WRKDIR

srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.676471E+03  -shape read  shapes/ice_steldend_aggregate01_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 15.0  -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.676471E+03  -shape read  shapes/ice_steldend_aggregate01_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 87.0  -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.676471E+03  -shape read  shapes/ice_steldend_aggregate01_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 159.0 -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.676471E+03  -shape read  shapes/ice_steldend_aggregate01_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 231.0 -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.676471E+03  -shape read  shapes/ice_steldend_aggregate01_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 303.0 -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.181373E+04  -shape read  shapes/ice_steldend_aggregate02_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 15.0  -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.181373E+04  -shape read  shapes/ice_steldend_aggregate02_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 87.0  -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.181373E+04  -shape read  shapes/ice_steldend_aggregate02_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 159.0 -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.181373E+04  -shape read  shapes/ice_steldend_aggregate02_nc001_T258_f50_ng102.shape  -ori
ent 0.0 0.0 231.0 -store_int_field -pol fcd -int fcd -iter cgnr -Csca -asym
srun ./adda_mpi -lambda 9.993082E4 -m 1.7748 1.1211E-4 -size  0.181373E+04  -shape read  shapes/ice_steldend_aggregate02_nc001_T258_f50_ng102.shape  -ori
```

Shape file:

```
Nmat=2
1 54 62 1
1 57 60 1
1 58 60 1
1 59 60 1
1 59 62 1
1 59 67 1
1 59 68 2
1 60 60 2
1 61 58 1
1 61 68 2
```

Output:

```
theta s11 s12 s13 s14 s21 s22 s23 s24 s31 s32 s33 s34 s41 s42 s43 s44
0.00 1.0581946063E-14 -5.0414336477E-15 2.9106727749E-15 1.4631936587E-26 -5.0414335315E-15 1.0145666354E-14 -7.5565827866E-16 -3.4928
363455E-19 2.9106729763E-15 -7.5565905424E-16 9.2731067859E-15 -6.0497700983E-19 1.8262182558E-26 3.4928360185E-19 6.0497702871E-19 8.
8368270761E-15
1.00 1.0581102235E-14 -5.0422109666E-15 2.9103445690E-15 6.4032110414E-23 -5.0422772927E-15 1.0144888969E-14 -7.5598646955E-16 -3.4920
834421E-19 2.9102296551E-15 -7.5554396111E-16 9.2716944062E-15 -6.0484460079E-19 1.2327543692E-23 3.4920229915E-19 6.0484808762E-19 8.
8354811455E-15
2.00 1.0578571777E-14 -5.0445419762E-15 2.9093603511E-15 2.5596617653E-22 -5.0448075485E-15 1.0142557762E-14 -7.5697064236E-16 -3.4898
259500E-19 2.9088998265E-15 -7.5519871679E-16 9.2674576974E-15 -6.0444746866E-19 4.9240390843E-23 3.4895845182E-19 6.0446135509E-19 8.
8314437638E-15
3.00 1.0574357773E-14 -5.0484238363E-15 2.9077213203E-15 5.7546197098E-22 -5.0490212158E-15 1.0138675574E-14 -7.5860959798E-16 -3.4860
675208E-19 2.9066838958E-15 -7.5462342645E-16 9.2603979503E-15 -6.0378589825E-19 1.1071182971E-22 3.4855224202E-19 6.0381710327E-19 8.
8247161610E-15
4.00 1.0568465358E-14 -5.0538518173E-15 2.9054294737E-15 1.0219288902E-21 -5.0549131608E-15 1.0133247135E-14 -7.6090133952E-16 -3.4808
142323E-19 2.9035825381E-15 -7.5381826538E-16 9.2505173161E-15 -6.0286036394E-19 1.9666696470E-22 3.4798397320E-19 6.0291578545E-19 8.
8153003872E-15
5.00 1.0560901710E-14 -5.0608193058E-15 2.9024876036E-15 1.5945418270E-21 -5.0624762047E-15 1.0126279058E-14 -7.6384307472E-16 -3.4740
745759E-19 2.8995966982E-15 -7.5278347889E-16 9.2378188051E-15 -6.0167152920E-19 3.0700106974E-22 3.4725406972E-19 6.0175803558E-19 8.
8031993110E-15
```

Questions to think about

Do I need to use DDA? → TMM, Mie, Rayleigh

How do I generate shapes? → physical/stochastic properties

How many shapes do I need to generate? → scripting

How many dipoles do I need for the shapes? → memory, accuracy

Serial or parallel run? → memory, CPU-time, automation