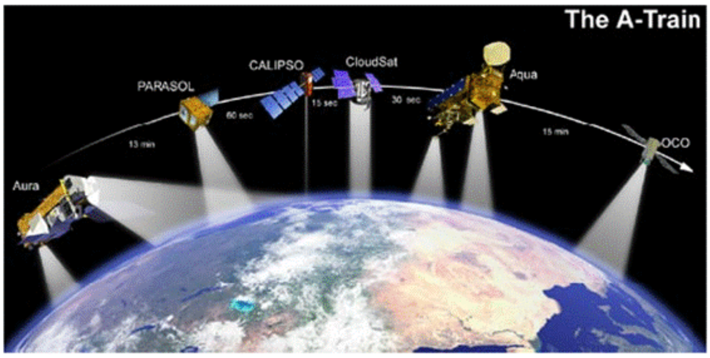
# Synergistic cloud retrievals from radar, lidar and radiometers

**Robin Hogan**
**Julien Delanoë, Nicola Pounder, Nicky Chalmers, Thorwald Stein, Anthony Illingworth**
**University of Reading**

## Spaceborne radar, lidar and radiometers





The A-Train
- CloudSat 94-GHz radar (launch 2006)
- Calipso 532/1064-nm depol. lidar
- MODIS multi-wavelength radiometer
- CERES broad-band radiometer
- 700-km orbit
- NASA

EarthCARE (launch 2013)
- 94-GHz Doppler radar
- 355-nm HSRL/depol. lidar
- Multispectral imager
- Broad-band radiometer
- 400-km orbit (more sensitive)
- ESA+JAXA

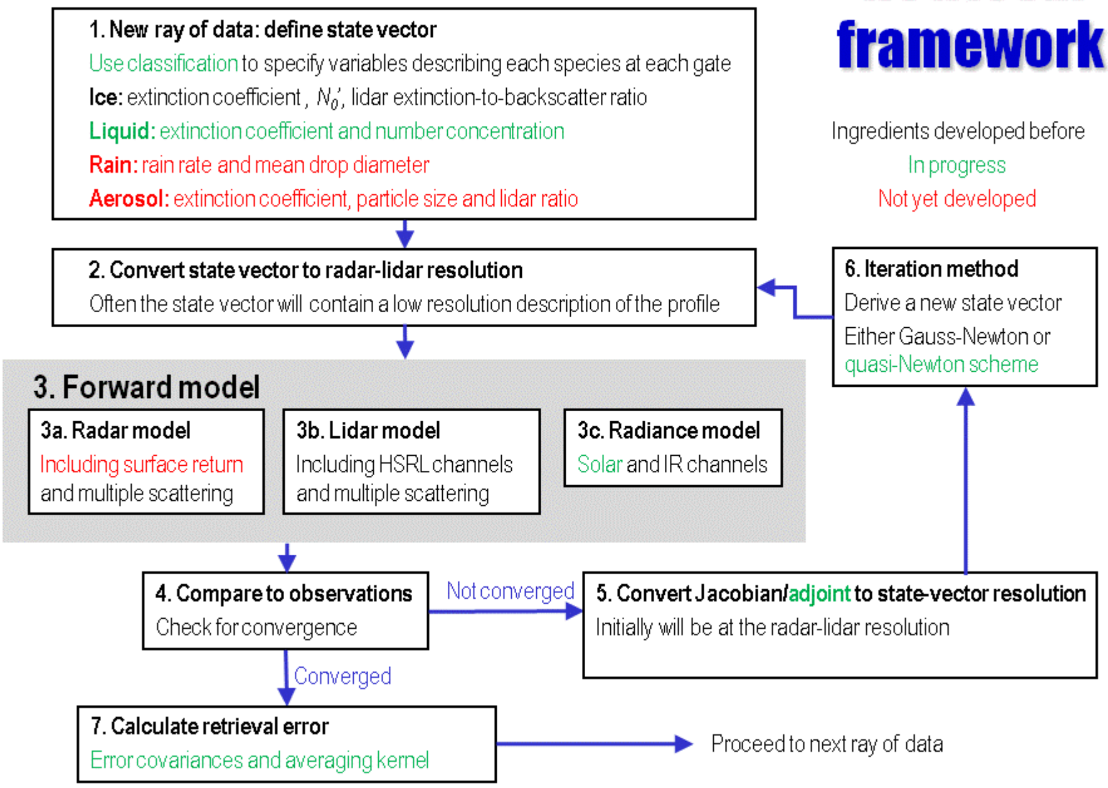
## Towards assimilation of cloud radar and lidar

- Before we assimilate radar and lidar into NWP models it is helpful to first develop variational cloud retrievals
  - Need to develop forward models and their adjoints: used by both
  - Refine microphysical and a-priori assumptions
  - Get an understanding of information content from observations
- Progress in our development of synergistic radar-lidar-radiometer retrievals of clouds:
  - Variational retrieval of ice clouds applied to ground-based radar-lidar and the SEVIRI radiometer (Delanoe and Hogan 2008)
  - Applied to >2 years of A-Train data (Delanoe and Hogan 2010)
  - Fast forward models for radar and lidar subject to multiple scattering (Hogan 2008, 2009; Hogan and Battaglia 2009)
  - With ESA & NERC funding, currently developing a "unified" algorithm for retrieving cloud, aerosol and precipitation properties from the EarthCARE radar, lidar and radiometers; will apply to other platforms

## Overview

- Retrieval framework
- Minimization techniques: Gauss-Newton vs. Gradient Descent
- Results from CloudSat-Calipso ice-cloud retrieval
- Components of unified retrieval: state variables and forward models
- Multiple scattering radar and lidar forward model
- Multiple field-of-view lidar retrieval
- First results from prototype unified retrieval

# Retrieval framework




# Minimizing the cost function

$$J = \frac{1}{2}\left[\mathbf{y} - H(\mathbf{x})\right]^T \mathbf{R}^{-1}\left[\mathbf{y} - H(\mathbf{x})\right] + \frac{1}{2}(\mathbf{x}-\mathbf{a})^T \mathbf{B}^{-1}(\mathbf{x}-\mathbf{a})$$

Gradient of cost function (a vector)

$$\nabla_{\mathbf{x}} J = -\mathbf{H}^T\mathbf{R}^{-1}\left[\mathbf{y} - H(\mathbf{x})\right] + \mathbf{B}^{-1}(\mathbf{x}-\mathbf{a})$$

and 2nd derivative (the Hessian matrix):

$$\nabla^2_{\mathbf{x}} J = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1}$$

**Gauss-Newton method**

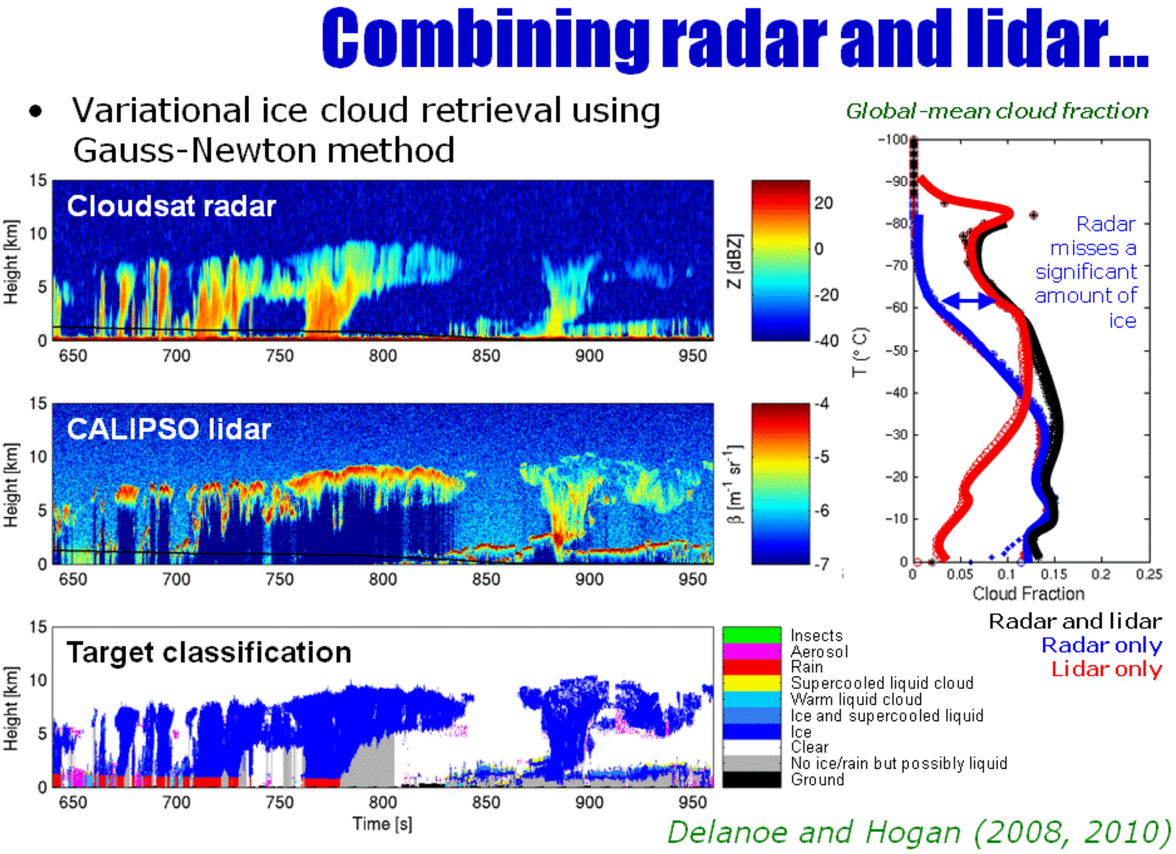
$$\mathbf{x}_{i+1} = \mathbf{x}_i - \left(\nabla^2_{\mathbf{x}} J\right)^{-1}\nabla_{\mathbf{x}} J$$

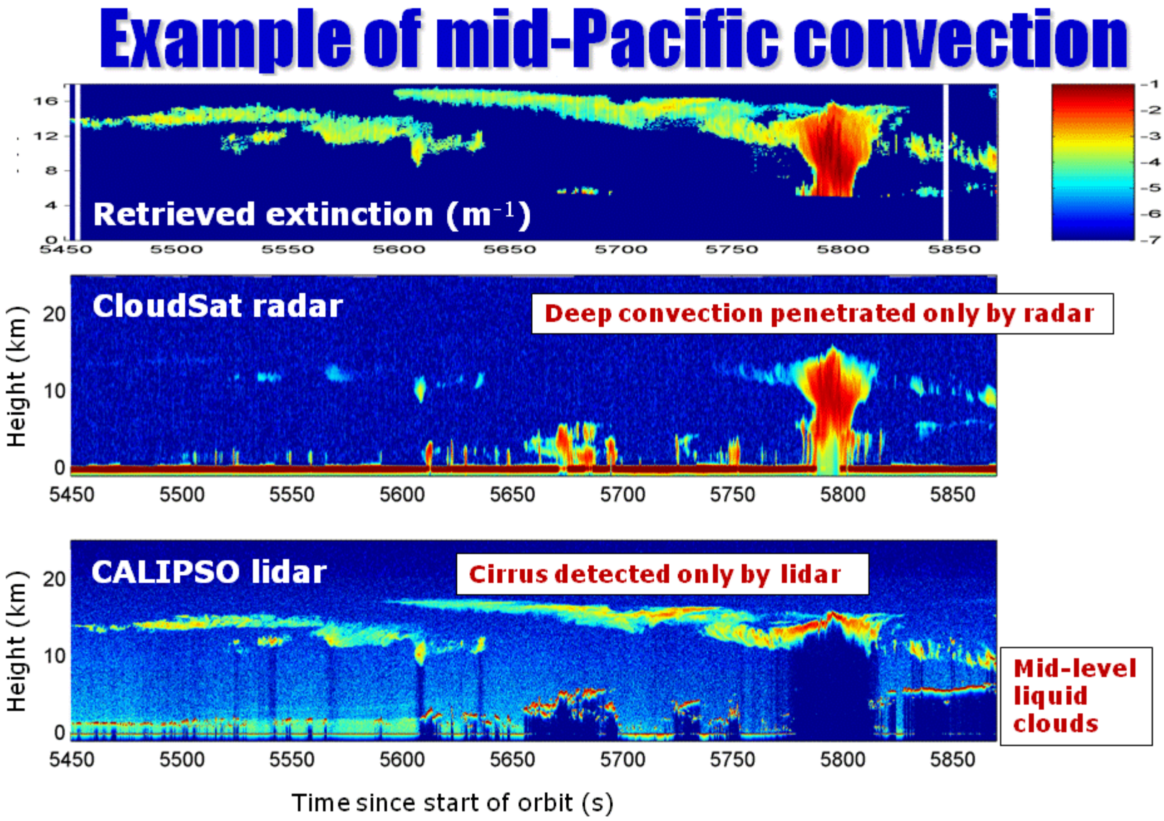
- Rapid convergence (instant for linear problems)
- Get solution error covariance "for free" at the end
- Levenberg-Marquardt is a small modification to ensure convergence
- Need the Jacobian matrix **H** of every forward model: can be expensive for larger problems as forward model may need to be rerun with each element of the state vector perturbed

**Gradient Descent methods**

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{A}\nabla_{\mathbf{x}} J$$

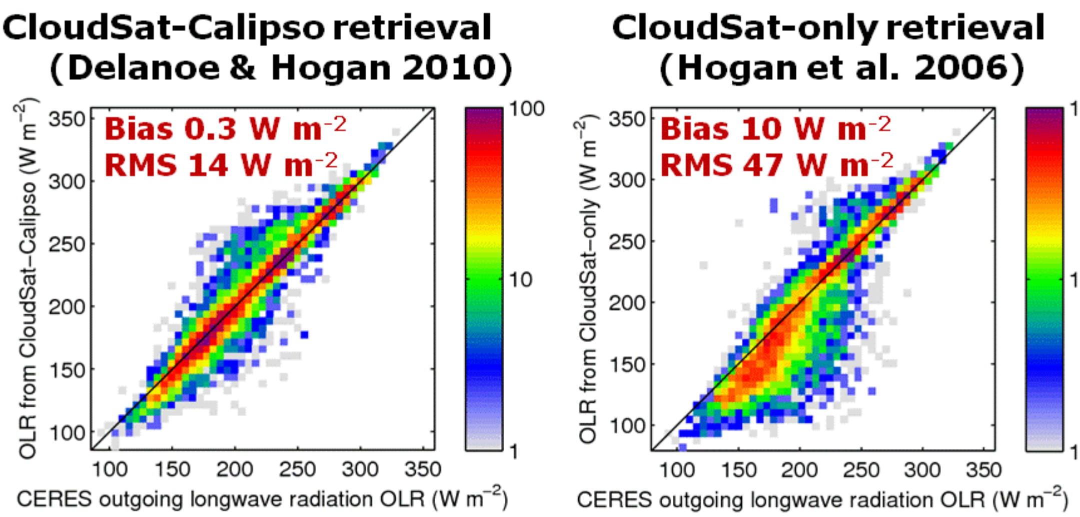
- Fast adjoint method to calculate $\nabla_{\mathbf{x}} J$ means don't need to calculate Jacobian
- Disadvantage: more iterations needed since we don't know curvature of $J(\mathbf{x})$
- Quasi-Newton method to get the search direction (e.g. L-BFGS used by ECMWF): builds up an approximate inverse Hessian *A* for improved convergence
- Scales well for large **x**
- Poorer estimate of the error at the end


Combining radar and lidar...
• Variational ice cloud retrieval using Gauss-Newton method
Cloudsat radar
Z [dBZ]
CALIPSO lidar
β [m⁻¹ sr⁻¹]
Target classification
Height [km]
Time [s]
Insects
Aerosol
Rain
Supercooled liquid cloud
Warm liquid cloud
Ice and supercooled liquid
Ice
Clear
No ice/rain but possibly liquid
Ground
Global-mean cloud fraction
Radar misses a significant amount of ice
T (°C)
Cloud Fraction
Radar and lidar
Radar only
Lidar only
Delanoe and Hogan (2008, 2010)


Example of mid-Pacific convection
Retrieved extinction (m⁻¹)
CloudSat radar
Deep convection penetrated only by radar
CALIPSO lidar
Cirrus detected only by lidar
Mid-level liquid clouds
Height (km)
Time since start of orbit (s)

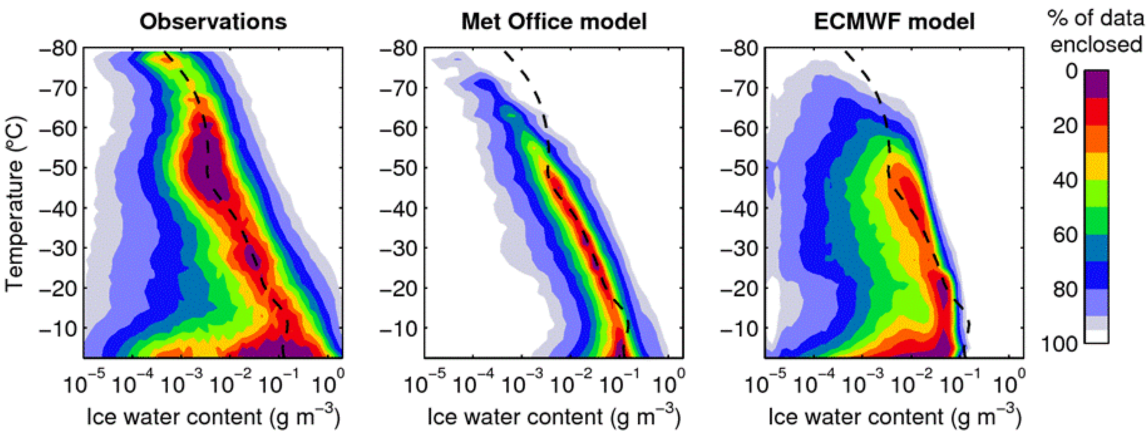
# Evaluation using CERES longwave flux

- Retrieved profiles containing only ice are used with Edwards-Slingo radiation code to predict outgoing longwave radiation, and compared to CERES




*Nicky Chalmers*

# Evaluation of models




- Comparison of the IWC distribution versus temperature for July 2006
- Met Office model has too little spread
- ECMWF model lacks high IWC values due to snow threshold
- New ECMWF model version remedies this problem

*Delanoe et al. (2010)*

# Unified algorithm: state variables

- Proposed list of retrieved variables held in the state vector **x**

| *State variable* | *Representation with height / constraint* | *A-priori* |
|---|---|---|
| **Ice clouds and snow** | | |
| Visible extinction coefficient | One variable per pixel with smoothness constraint | None |
| Number conc. parameter | Cubic spline basis functions with vertical correlation | Temperature dependent |
| Lidar extinction-to-backscatter ratio | Cubic spline basis functions | 20 sr |
| Riming factor | Likely a single value per profile | 1 |
| **Liquid clouds** | | |
| Liquid water content | One variable per pixel but with gradient constraint | None |
| Droplet number concentration | One value per liquid layer | Temperature dependent |
| **Rain** | | |
| Rain rate | Cubic spline basis functions with flatness constraint | None |
| Normalized number conc. $N_w$ | One value per profile | Dependent on whether from melting ice or coallescence |
| Melting-layer thickness scaling factor | One value per profile | 1 |
| **Aerosols** | | |
| Extinction coefficient | One variable per pixel with smoothness constraint | None |
| Lidar extinction-to-backscatter ratio | One value per aerosol layer identified | Climatological type depending on region |

Ice clouds follows Delanoe & Hogan (2008); *Snow & riming in convective clouds needs to be added*

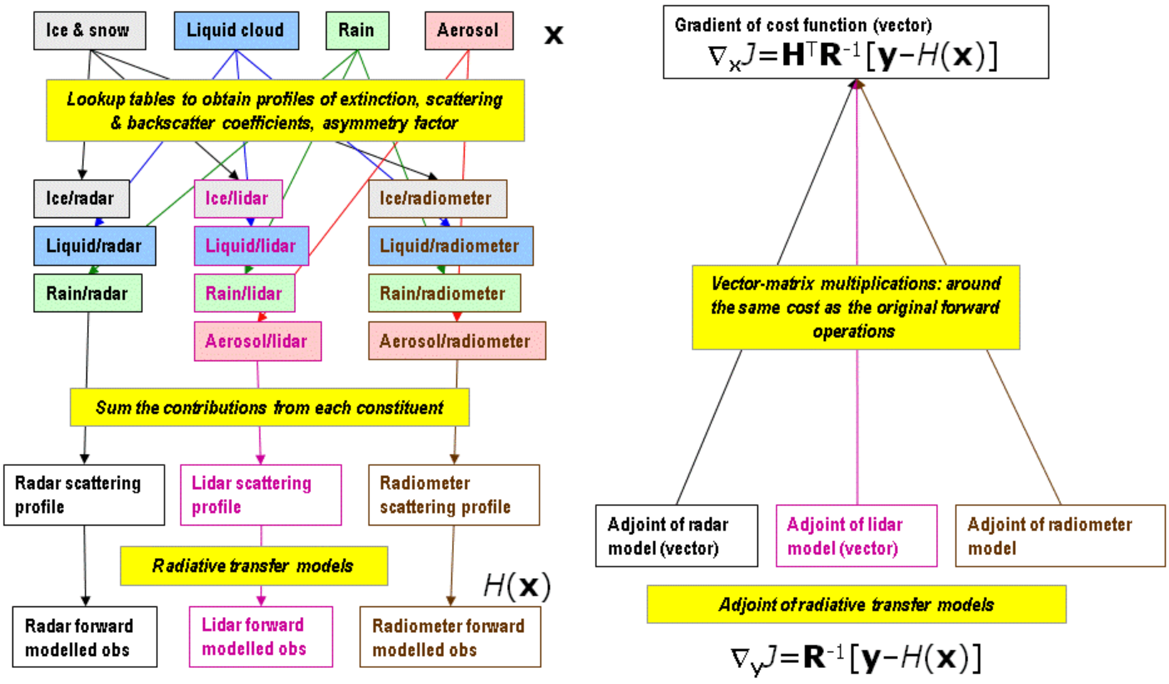
Liquid clouds currently being tackled

Basic rain to be added shortly; *Full representation later*

Basic aerosols to be added shortly; *Full representation via collaboration?*

# Forward model components

- From state vector **x** to forward modelled observations $H(\mathbf{x})$...



# Scattering models

- First part of a forward model is the *scattering and fall-speed model*
  - Same methods typically used for all radiometer and lidar channels
  - Radar and Doppler model uses another set of methods

| *Particle type* | *Radar (3.2 mm)* | *Radar Doppler* | *Thermal IR, Solar, UV* |
|---|---|---|---|
| Aerosol | *Aerosol not detected by radar* | *Aerosol not detected by radar* | Mie theory, Highwood refractive index |
| Liquid droplets | Mie theory | Beard (1976) | Mie theory |
| Rain drops | T-matrix: Brandes et al. (2002) shapes | Beard (1976) | Mie theory |
| Ice cloud particles | T-matrix (Hogan et al. 2010) | Westbrook & Heymsfield | Baran (2004) |
| Graupel and hail | Mie theory | **TBD** | Mie theory |
| Melting ice | Wu & Wang (1991) | **TBD** | Mie theory |

- Graupel and melting ice still uncertain

# Radiative transfer forward models

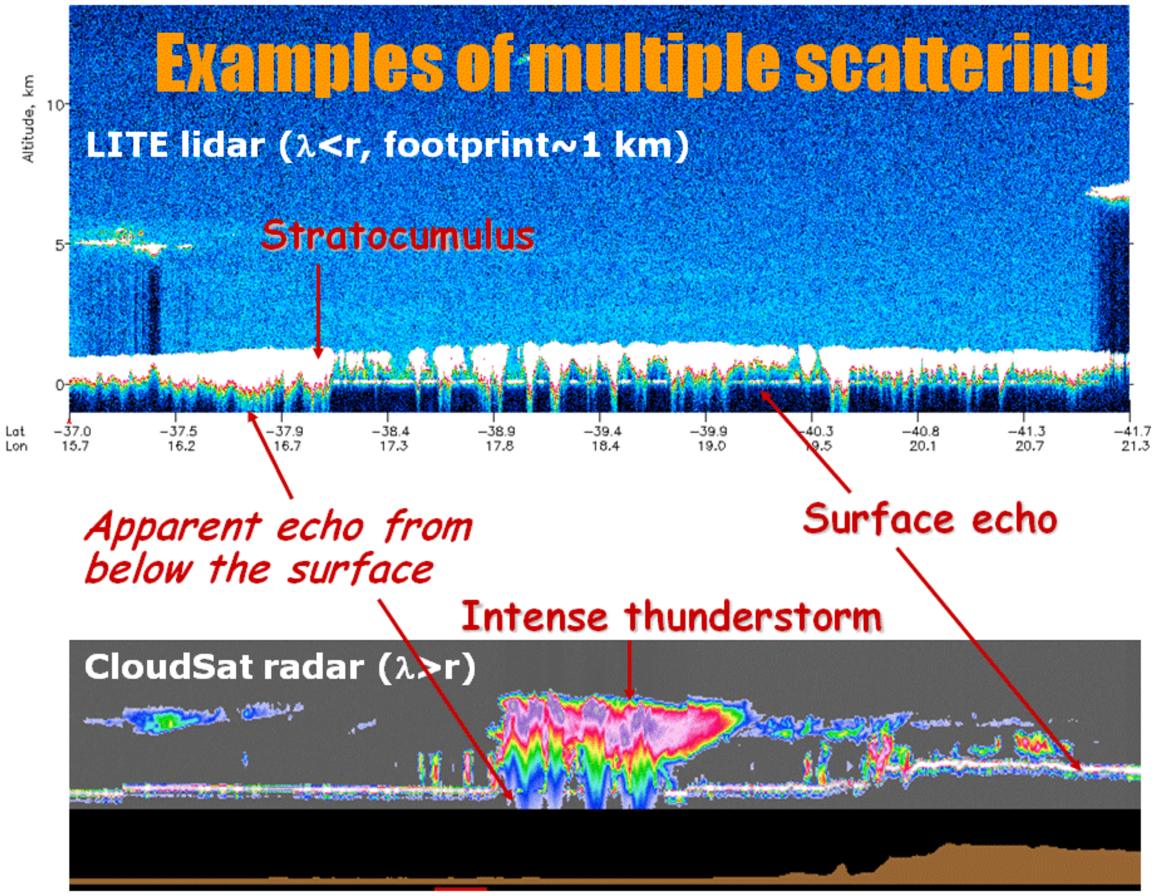
- Computational cost can scale with number of points describing vertical profile $N$; we can cope with an $N^2$ dependence but not $N^3$

| *Radar/lidar model* | *Applications* | *Speed* | *Jacobian* | *Adjoint* |
|---|---|---|---|---|
| Single scattering: $\beta'=\beta\exp(-2\tau)$ | Radar & lidar, no multiple scattering | $N$ | $N^2$ | $N$ |
| Platt's approximation $\beta'=\beta\exp(-2\eta\tau)$ | Lidar, ice only, crude multiple scattering | $N$ | $N^2$ | $N$ |
| Photon Variance-Covariance (PVC) method (Hogan 2006, 2008) | Lidar, ice only, small-angle multiple scattering | $N$ or $N^2$ | $N^2$ | $N$ |
| Time-Dependent Two-Stream (TDTS) method (Hogan and Battaglia 2008) | Lidar & radar, wide-angle multiple scattering | $N^2$ | $N^3$ | $N^2$ |
| Depolarization capability for TDTS | Lidar & radar depol with multiple scattering | $N^2$ | | $N^2$ |

- Lidar uses PVC+TDTS ($N^2$), radar uses single-scattering+TDTS ($N^2$)
- Jacobian of TDTS is too expensive: $N^3$
- We have recently coded adjoint of multiple scattering models
- Future work: depolarization forward model with multiple scattering

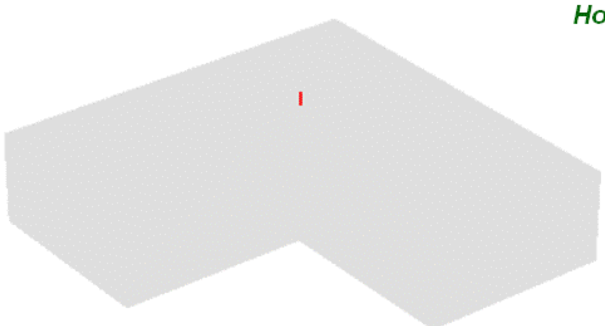
| *Radiometer model* | *Applications* | *Speed* | *Jacobian* | *Adjoint* |
|---|---|---|---|---|
| RTTOV (used at ECMWF & Met Office) | Infrared and microwave radiances | $N$ | | $N$ |
| Two-stream source function technique (e.g. Delanoe & Hogan 2008) | Infrared radiances | $N$ | $N^2$ | |
| LIDORT | Solar radiances | $N$ | $N^2$ | $N$ |

- Infrared will probably use RTTOV, solar radiances will use LIDORT
- Both currently being tested by Julien Delanoe

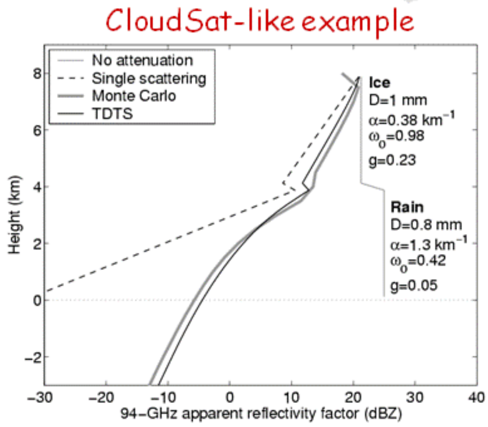


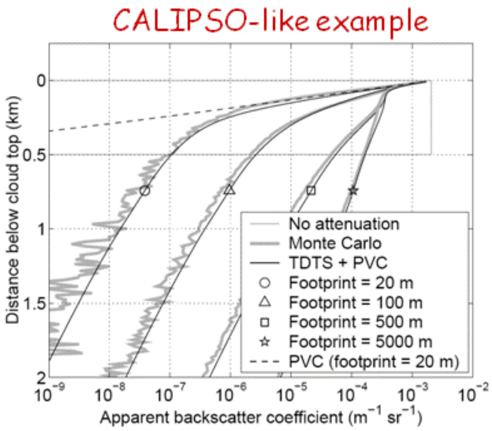

# Fast multiple scattering forward model

*Hogan and Battaglia (J. Atmos. Sci. 2008)*



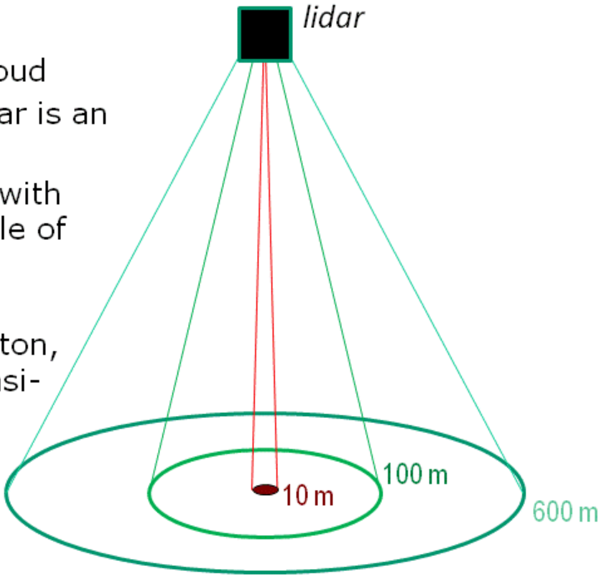
- New method uses the *time-dependent two-stream approximation*
- Agrees with Monte Carlo but ~$10^7$ times faster (~3 ms)
- Added to CloudSat simulator



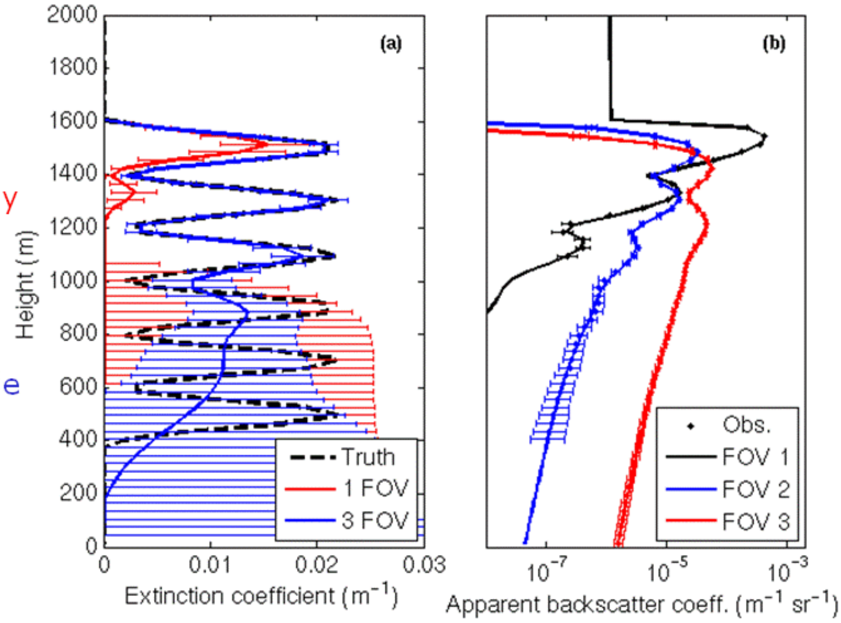
# Multiple field-of-view lidar retrieval

- To test multiple scattering model in a retrieval, and its adjoint, consider a multiple field-of-view lidar observing a liquid cloud
- Wide fields of view provide information deeper into the cloud
- The NASA airborne "THOR" lidar is an example with 8 fields of view
- Simple retrieval implemented with state vector consisting of profile of extinction coefficient
- Different solution methods implemented, e.g. Gauss-Newton, Levenberg-Marquardt and Quasi-Newton (L-BFGS)



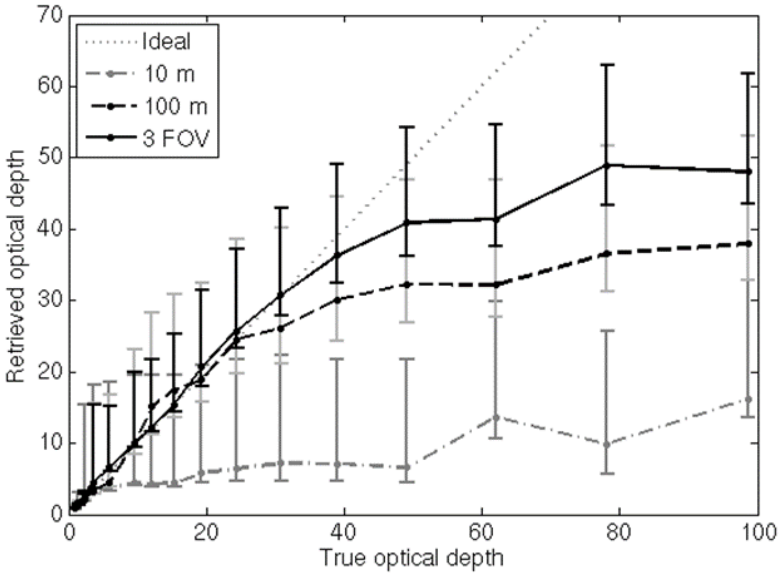

# Results for a sine profile

- Simulated test with 200-m sinusoidal structure in extinction
- With one FOV, only retrieve first 2 optical depths
- With three FOVs, retrieve structure of extinction profile down to 6 optical depths
- Beyond that the information is smeared out




*Nicola Pounder*

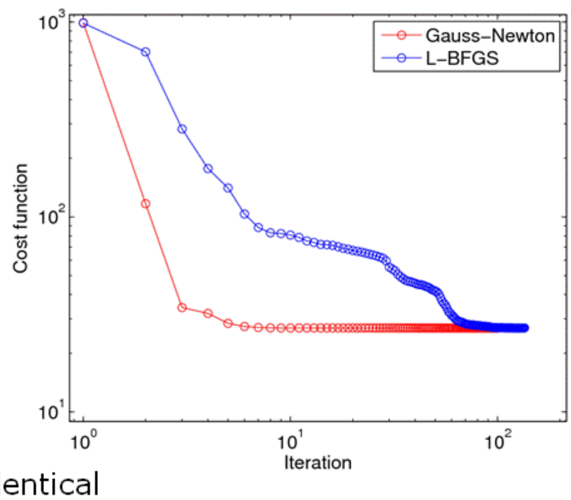
# Optical depth from multiple FOV lidar



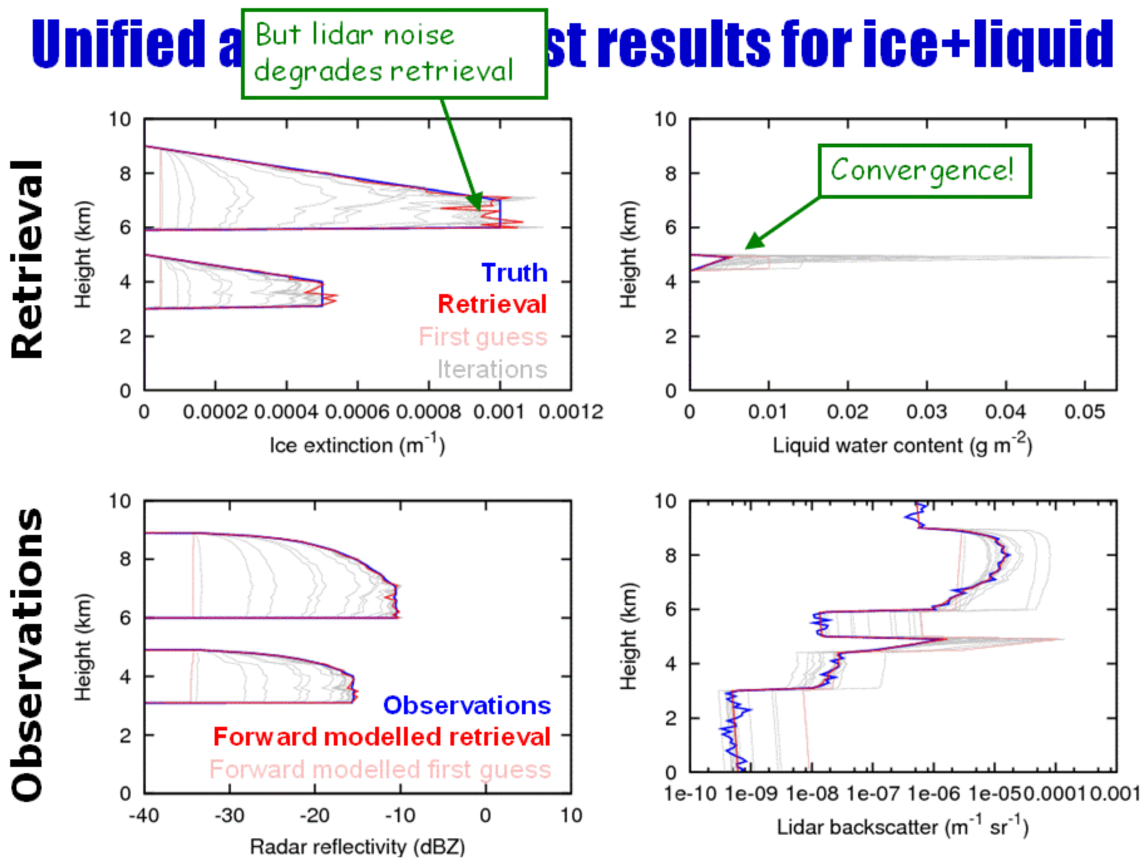

- Despite vertical smearing of information, the total optical depth can be retrieved to ~30 optical depths
- Limit is closer to 3 for one narrow field-of-view lidar

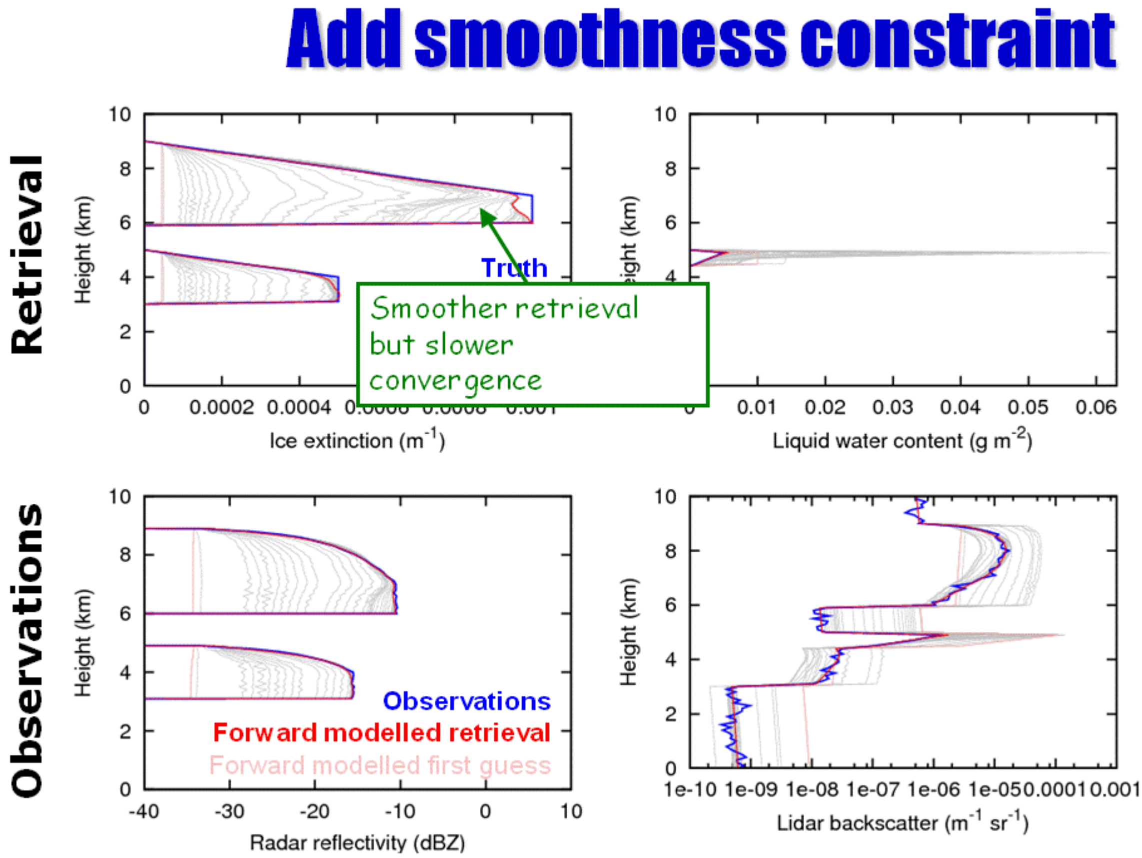
*Nicola Pounder*

# Comparison of convergence rates




- Solution is identical
- Gauss-Newton method converges in < 10 iterations
- L-BFGS Gradient Descent method converges in < 100 iterations
- Conjugate Gradient method converges a little slower than L-BFGS
- Each L-BFGS iteration >> 10x faster than each Gauss-Newton one!
- Gauss-Newton method requires the Jacobian matrix, which must be calculated by rerunning multiple scattering model multiple times


Unified a
st results for ice+liquid
But lidar noise degrades retrieval
Convergence!
Retrieval
Observations
Truth
Retrieval
First guess
Iterations
Height (km)
Ice extinction (m-1)
Liquid water content (g m-2)
Observations
Forward modelled retrieval
Forward modelled first guess
Radar reflectivity (dBZ)
Lidar backscatter (m-1 sr-1)


Add smoothness constraint
Retrieval
Observations
Truth
Smoother retrieval but slower convergence
Height (km)
Ice extinction (m-1)
Liquid water content (g m-2)
Observations
Forward modelled retrieval
Forward modelled first guess
Radar reflectivity (dBZ)
Lidar backscatter (m-1 sr-1)

# Unified algorithm: progress

- Done:
  - Functioning algorithm framework exists
  - C++: object orientation allows code to be completely flexible: observations can be added and removed without needing to keep track of indices to matrices, so same code can be applied to different observing systems
  - Code to generate particle scattering libraries in NetCDF files
  - Adjoint of radar and lidar forward models with multiple scattering and HSRL/Raman support
  - Interface to L-BFGS algorithm in GNU Scientific Library
- In progress / future work:
  - Debug adjoint code (so far we are using numerical adjoint - slow)
  - Implement full ice, liquid, aerosol and rain constituents
  - Estimate and report error in solution and averaging kernel
  - Interface to radiance models
  - Test on a range of ground-based, airborne and spaceborne instruments, particularly the A-Train and EarthCARE satellites
  - Assimilation?