ReFRACtor

Reusable FRamework for Atmospheric Composition

James McDuffie

ROSES 16-AIST-0019

Jet Propulsion Laboratory
California Institute of Technology

INTRODUCTION

ReFRACtor

- Reusable FRamework for Atmospheric Composition
- Fork and descendant of operational OCO-2 retrieval software
- Descendant of earlier code bases and over 10 years of work

OBJECTIVE

Develop an extensible multi-instrument atmospheric composition radiative transfer (RT) and retrieval framework to reduce the cost and risk of Level-2 development for future atmospheric Earth science missions

TEAM MEMBERS

Team Member	Responsibilities
James McDuffie	PI / Software engineering
Kevin Bowman	Co-I / Science
Jonathan Hobbs	Co-I / Uncertainty quantification
Vijay Natraj	Co-I / Radiative transfer
Edwin Sarkissian	Retrieval implementation
Matthew Thill	Signal processing
Sebastian Val	Software engineering

GOALS

- Serve as a common, reusable and extensible framework for processing atmospheric composition Level-1 data through to Level-2
- Support the data fusion of observations from multiple instruments in joint retrievals into multi-species data products

REUSABILITY

- New instrument missions can start with a framework that has tested and verified algorithms
- New missions can focus on the novel algorithmic pieces unique to their instrument
- Eliminates duplication, and lowers costs while providing critical capabilities that support missions and longer term national needs

REUSABILITY

- Reusable and extensible allowing different instrument teams to use the same code base
- Multi-instrument not only in the sense that different instruments can use the same software, but also in the sense that data products can be enhanced from multiple instruments through joint retrievals

DUAL ROLES

- Use in operational processing of remote sensing measurements
- Use of framework components piecemeal for research

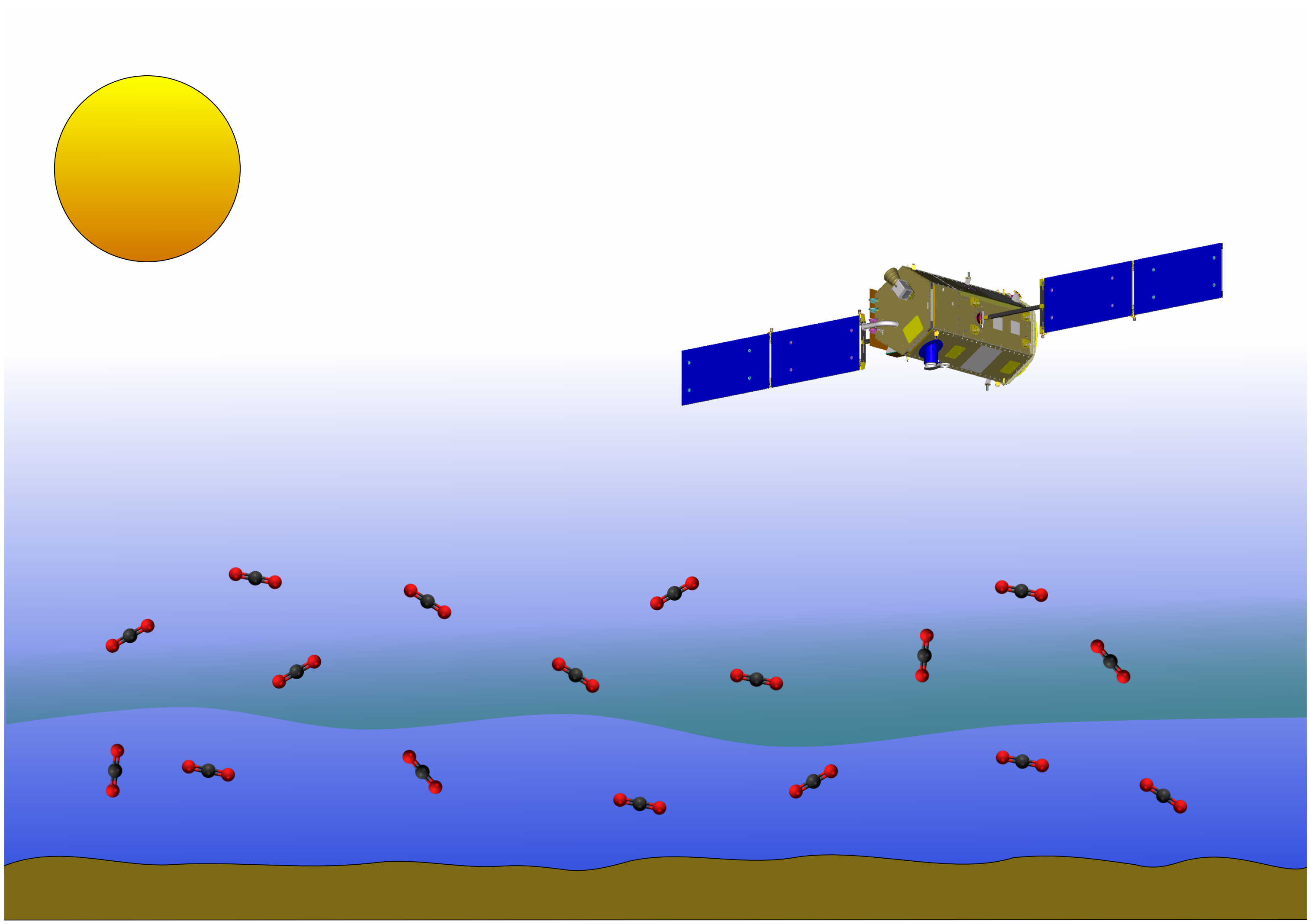
DATA FUSION

- Support data fusion of radiance measurements from multiple instruments through joint retrievals
- Data fusion of multiple instruments provides products with greater vertical sensitivity
- Joint retrievals additionally reduce measurement uncertainty.
- Increasing the density of observations with complementary time and spatial scales can improve our understanding of dynamic Earth systems such as the carbon cycle

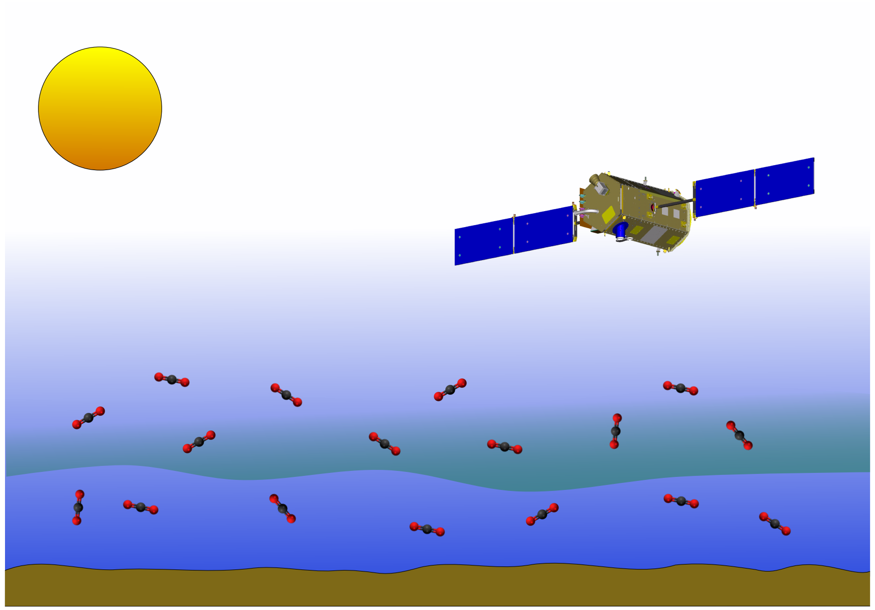
FAR FUTURE

This framework could enable an atmospheric science analytic center to support a common code base for joint retrievals and data fusion among multiple instrument teams

SCIENCE

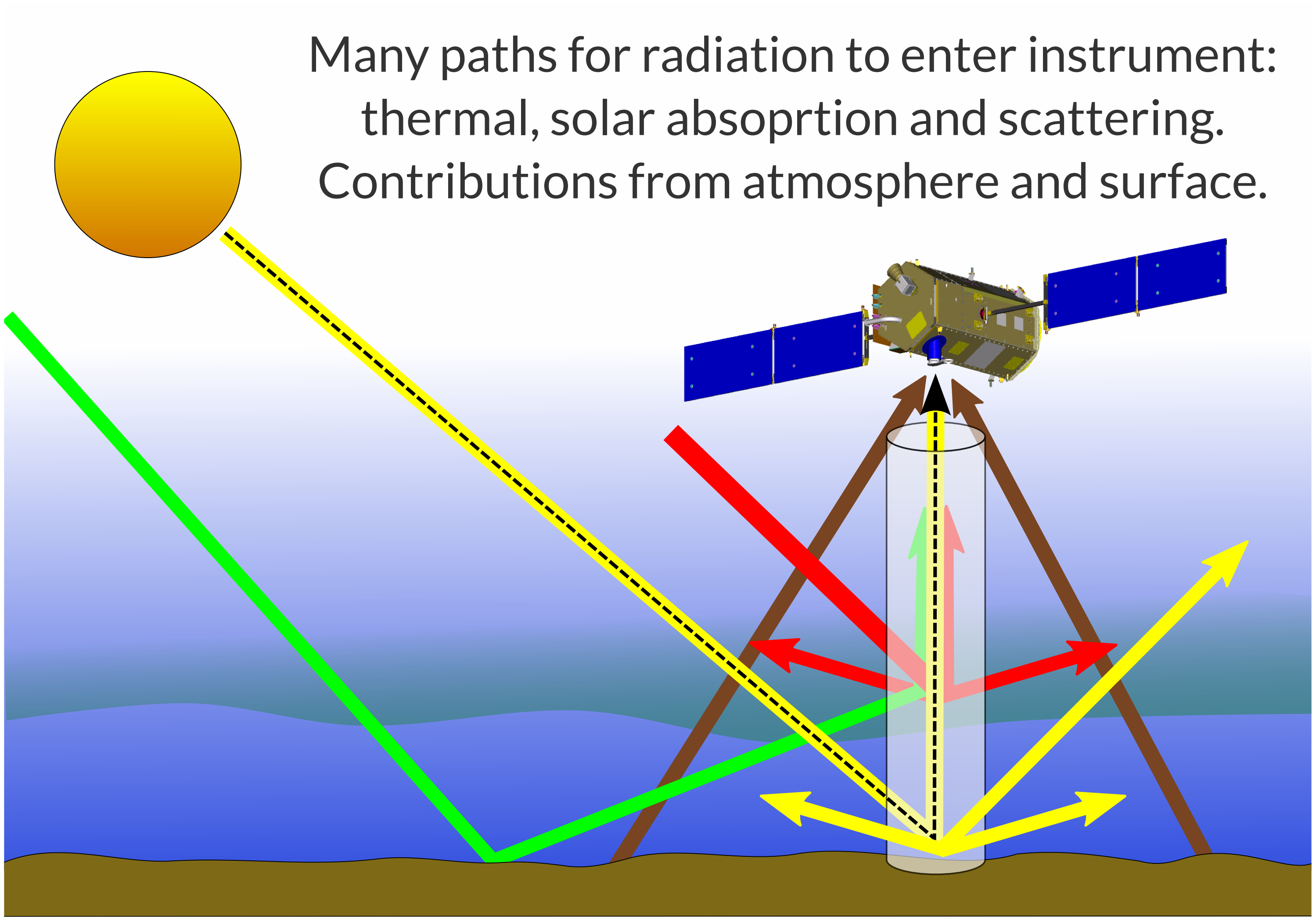


SCIENCE OBJECTIVE



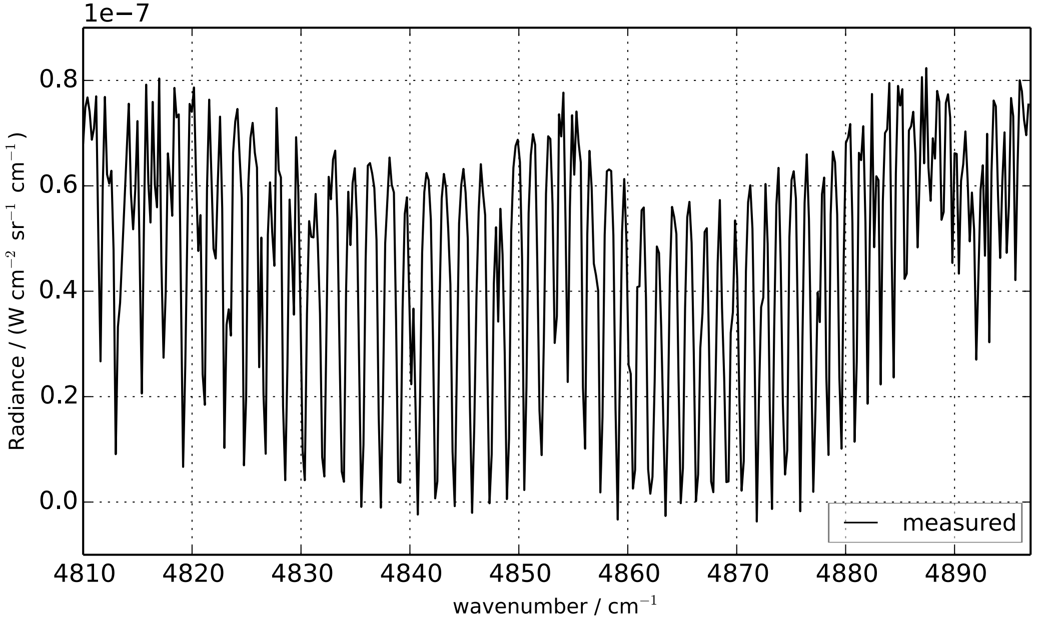
Determine atmospheric composition constituents for a column of air observed by a satellite, aircraft or uplooking instrument.

LIGHT PATHS



Many paths for radiation to enter instrument: thermal, solar absoprtion and scattering. Contributions from atmosphere and surface.

ATMOSPHERIC REMOTE SENSING



- Determine the concentration of gases in a vertical column of the atmosphere by the measurement of solar radiation and emission of thermal energy
- Takes advantage of the differences in the interaction of light and molecules at different wavelengths and pressures

RADIATIVE TRANSFER

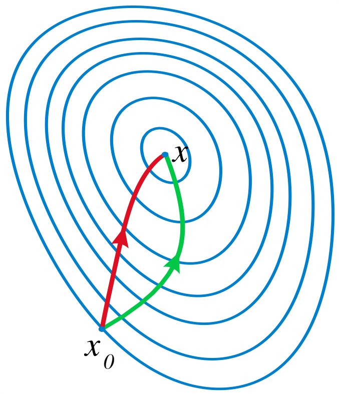
$$\mu \frac{\mathrm{d}\boldsymbol{I}(\tau, \theta, \phi)}{\mathrm{d}\tau} = \boldsymbol{I}(\tau, \theta, \phi) - \boldsymbol{J}(\tau, \theta, \phi)$$

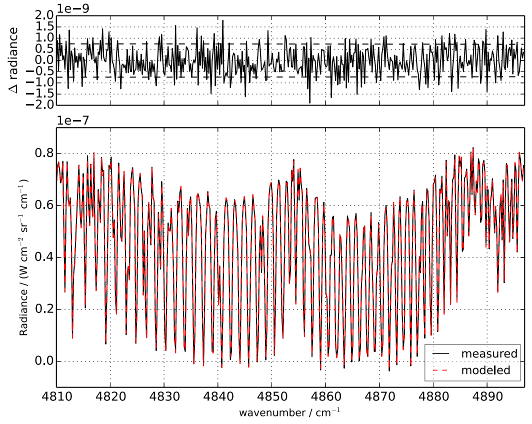
$$\boldsymbol{J}(\tau, \theta, \phi) = \frac{\omega(\tau)}{4\pi} \int_{-1}^{1} \int_{0}^{2\pi} \boldsymbol{\Pi}(\tau, \theta, \theta^{'}, \phi - \phi^{'})\boldsymbol{I}(\tau, \theta^{'}, \phi^{'})\mathrm{d}\phi^{'}\mathrm{d}\theta^{'} + \boldsymbol{Q}(\tau, \mu, \phi$$

$$\boldsymbol{Q}(\tau, \mu, \theta) = \frac{\omega(\tau)}{4\pi}\boldsymbol{\Pi}(\tau, \mu - \mu_0, \phi - \phi_0)\boldsymbol{I}_0 e^{\frac{-\tau}{\mu_0}}$$

Using radiative transfer mathematics we can model the propagation of radiation through the atmosphere to derive a synthetic spectra.

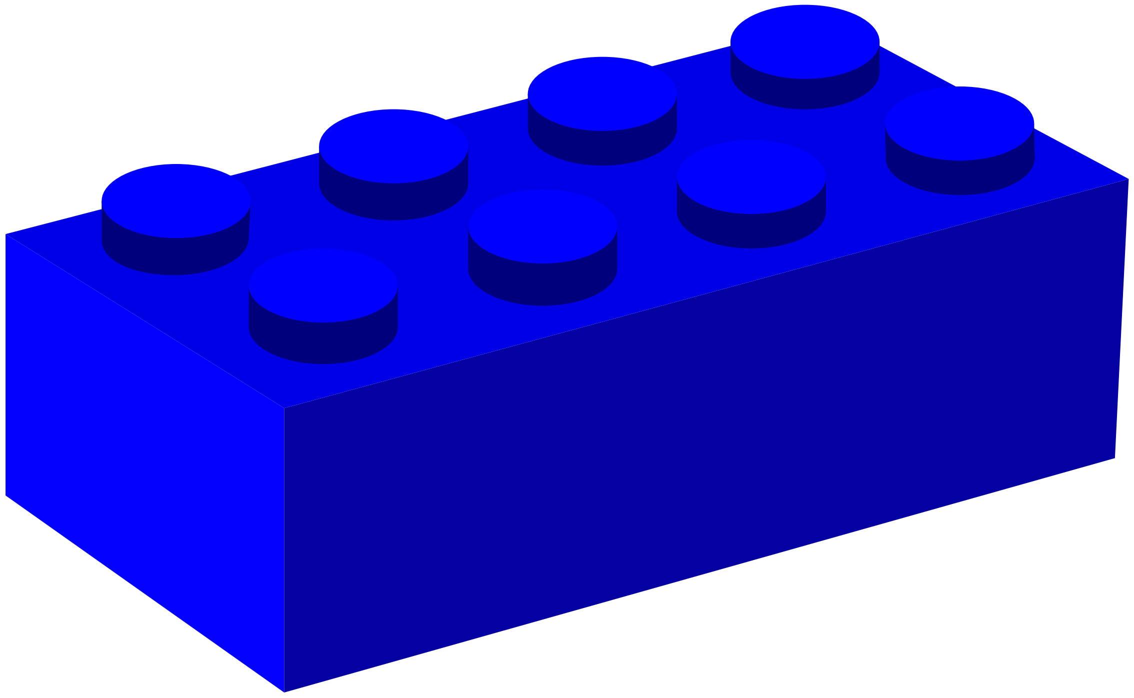
SOLVE FOR X



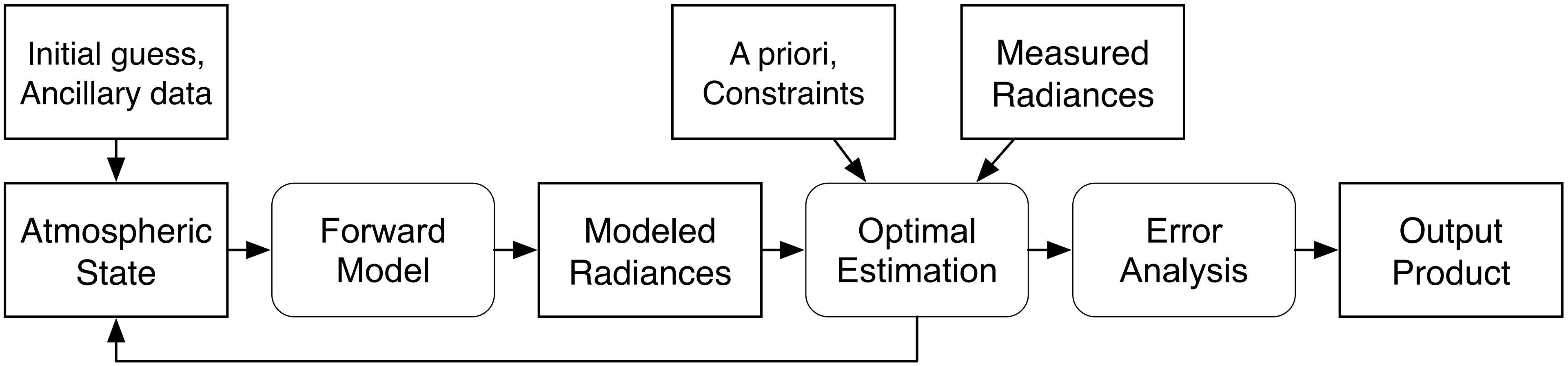


Using optimal estimation, update a state vector primed by our best initial guess of the atmospheric components to get the smallest possible difference between modeled and measured spectra.

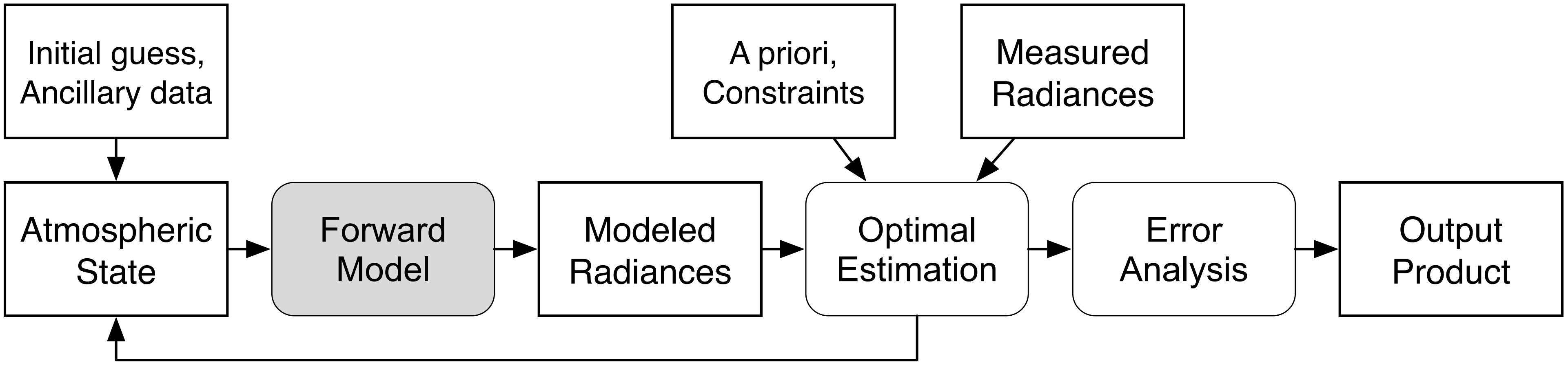
COMPONENTS



DATA FLOW

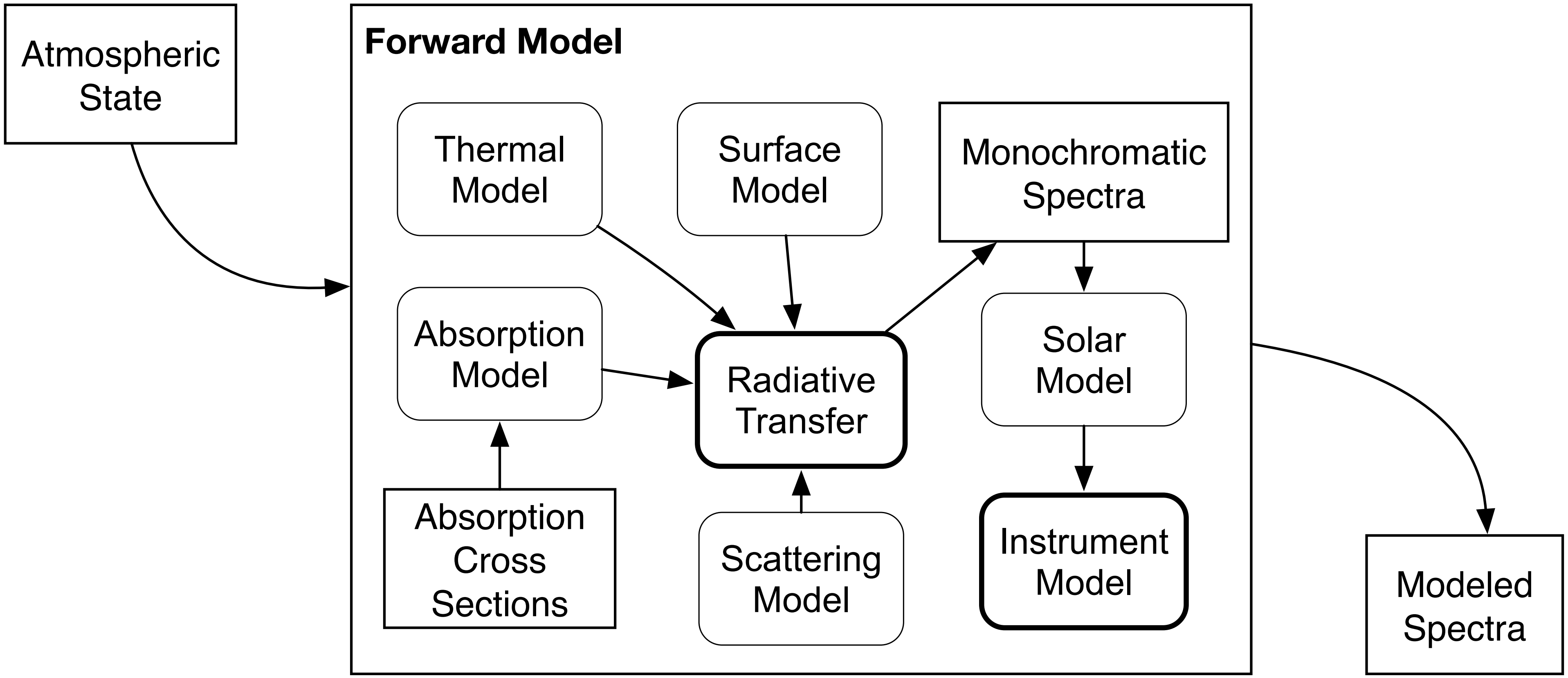

Initial guess,
Ancillary data
A priori,
Constraints
Measured
Radiances
Atmospheric
State
Forward
Model
Modeled
Radiances
Optimal
Estimation
Error
Analysis
Output
Product

FORWARD MODEL



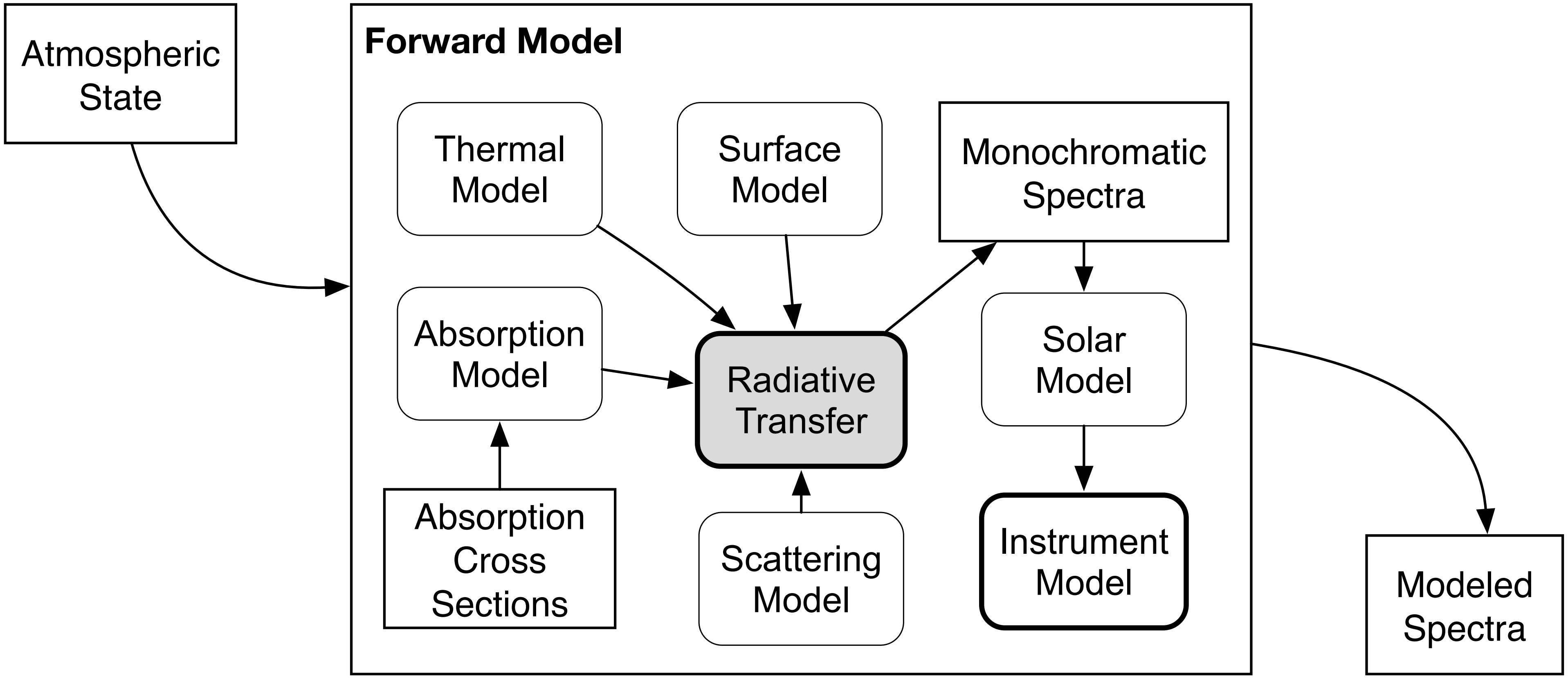
Models the spectra an instrument would have measured given a particular combination of atmospheric contents and viewing geometry using a physics based software.

FORWARD MODEL COMPONENTS



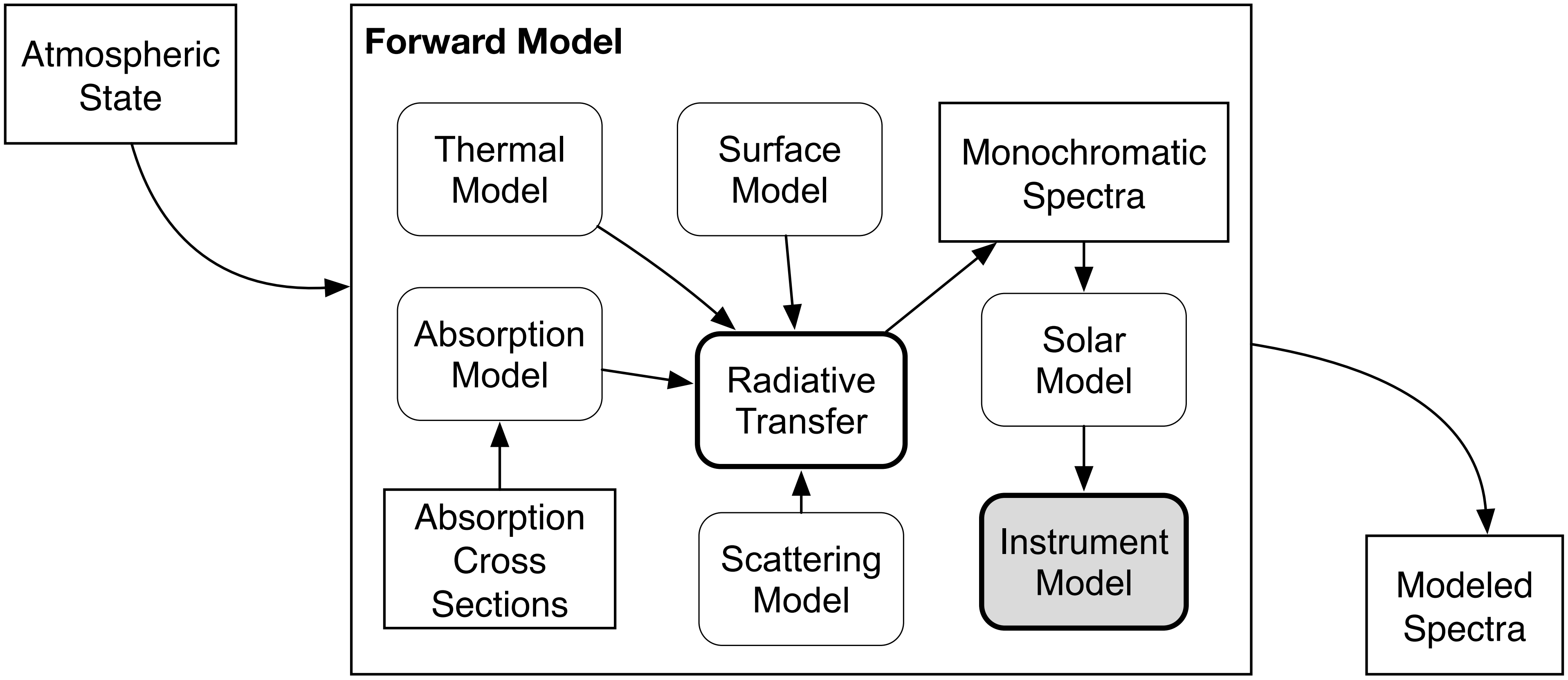
The largest important components in the forward model are a radiative transfer model and instrument model

RADIATIVE TRANSFER



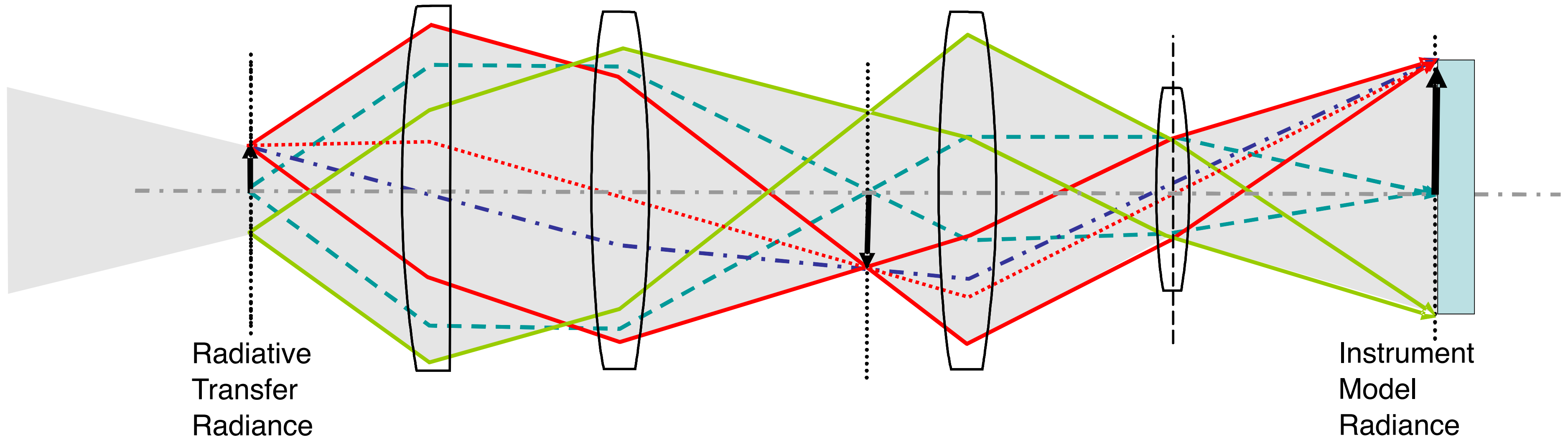
The radiative transfer component creates a normalized monochromatic radiance that accounts for thermal emissions, molecular absorption and particle scattering of surface and solar radiation.

INSTRUMENT MODEL



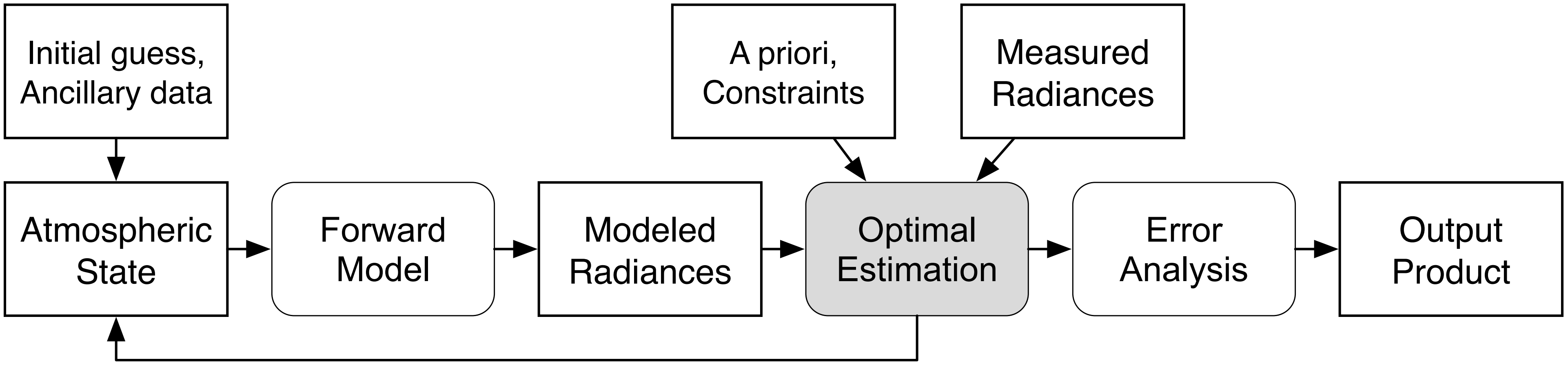
Applies the effects of the modifications to radiation as a result of the instrument performing the measurement.

RADIANCE IN PERSPECTIVE



- Radiative transfer radiances are a model of the radiation before it enters the instrument
- Instrument model radiances represent the radiation after going through the instrument optical path to the detector

OPTIMAL ESTIMATION

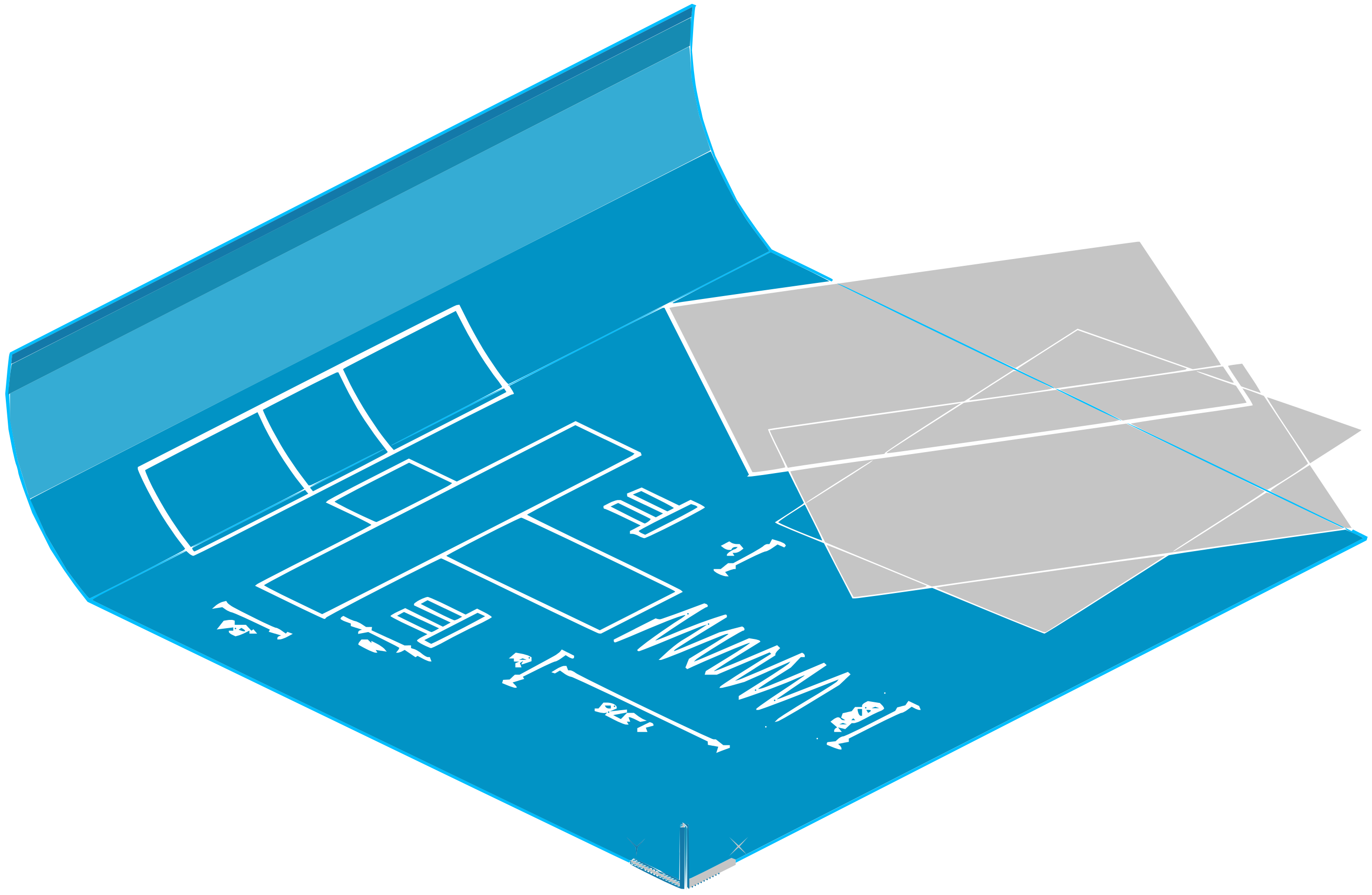


- Repeatedly evaluates the forward model given some constraints and a priori information
- Compares modeled and measured radiances to make changes to the atmospheric state for subsequent iterations.
- Exits when the cost function meets a defined convergence criteria

RETRIEVAL APPROACH

- Use optimal estimation (maximum a posteriori) approach described by Rodgers in *Inverse methods for atmospheric sounding [2000]*
- Various interchangeable Non-Linear Least Squares Solvers (NLLS) available in framework

SOFTWARE ARCHITECTURE



DESIGN PRINCIPLES

- Use abstract interfaces to decouple behavior from implementation
    - Open/closed principle
    - Dependency inversion
- Use Adapter pattern to reuse existing code

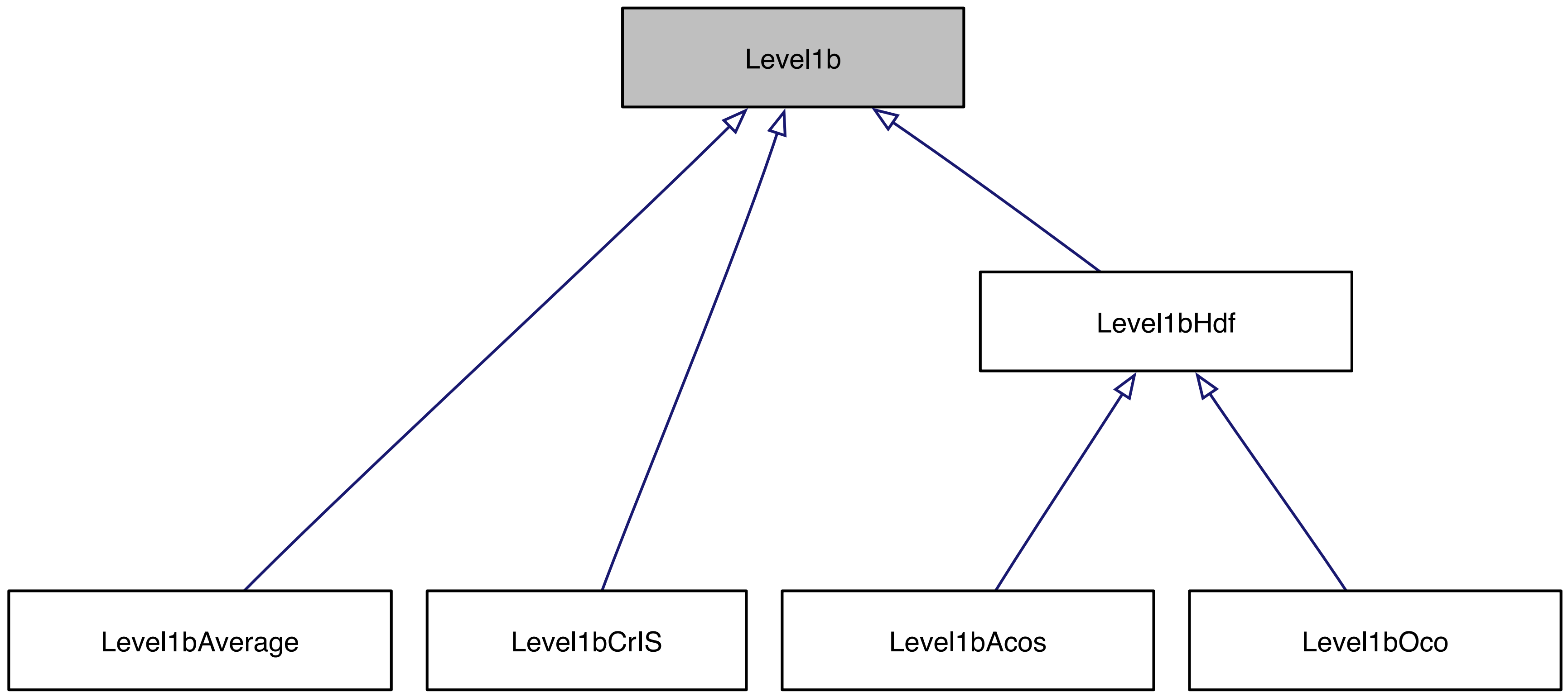
OPEN/CLOSED PRINCIPLE

- “Open for extension”
  Behavior can be extended as requirements change
- “Closed for modification”
  Working, tested software should not change to meet requirements

OPEN/CLOSED PRINCIPLE

- Core ideas at the heart of good object oriented design
- Use abstract base classes to define an interface
- Interfaces define possible behaviors of implementations

EXAMPLE: LEVEL 1B



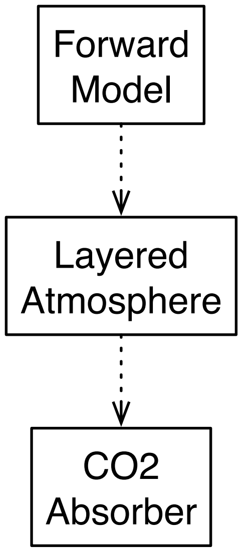
New implementations only require adding new classes that conform to a predefined abstract interface.

DEPENDENCY INVERSION

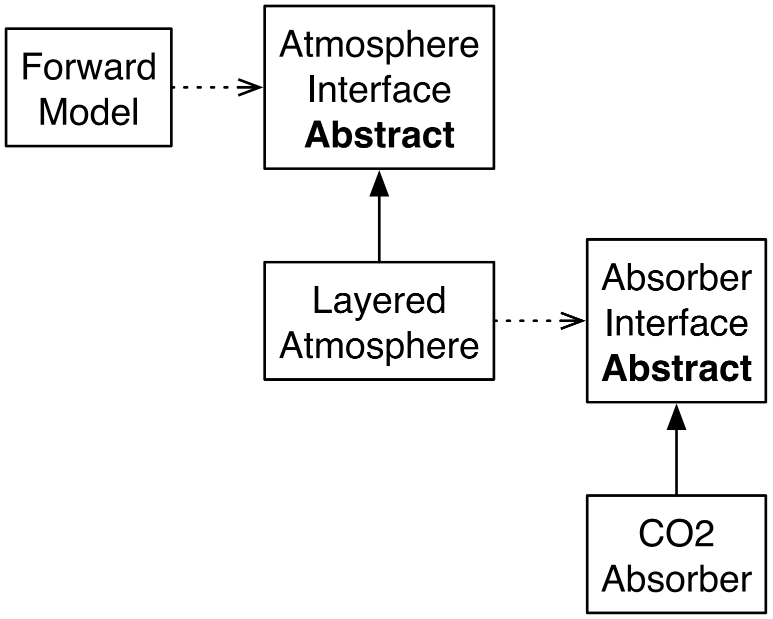
- High level modules should not depend on low level modules. Both should depend on abstractions.
- Abstractions should not depend on details. Instead, details should depend on abstractions.

DEPENDENCY INVERSION EXAMPLE

Change this:



Into this:


Forward Model
Atmosphere Interface **Abstract**
Layered Atmosphere
Absorber Interface **Abstract**
CO2 Absorber

Place interfaces between components to eliminate dependence on a specific implementation.

LANGUAGE CHOICE

- We chose C++ for the top level
  - Can express our object oriented design with it
  - Easy to interface with other languages
  - Well established with mature third party libraries
- Kept major portions, such as RT in Fortran
  - Use Fortran 2003's iso_c_bindings
  - Built interfaces and drivers to conform to our design
  - No loss in speed due to memory copying

THIRD PARTY SOFTWARE

- Blitz++
- Eigen
- Boost C++ Libraries
- GNU Scientific Library
- HDF5
- Python

PYTHON INTERFACE

python™

REASON

- Framework is complex collection of modules
- Python flexibility
  - Try out new algorithms and approaches
  - Speeds up prototyping of algorithmic additions
  - Allows using components piecemeal

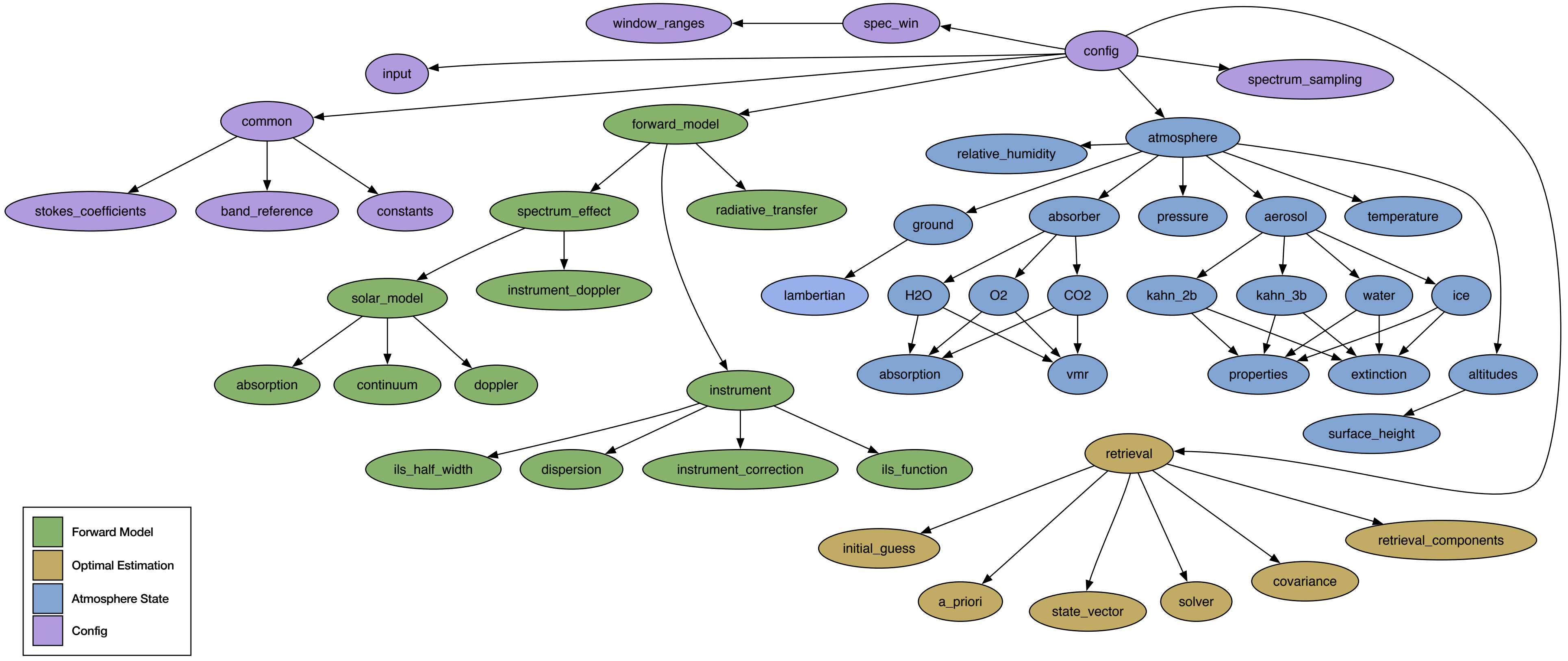
IMPLEMENTATION

- Wraps C++ classes using SWIG
- Python classes can be made to stand in for C++ classes
- Doxygen documentation is available through standard Python help system

CONFIG FACTORY LAYER

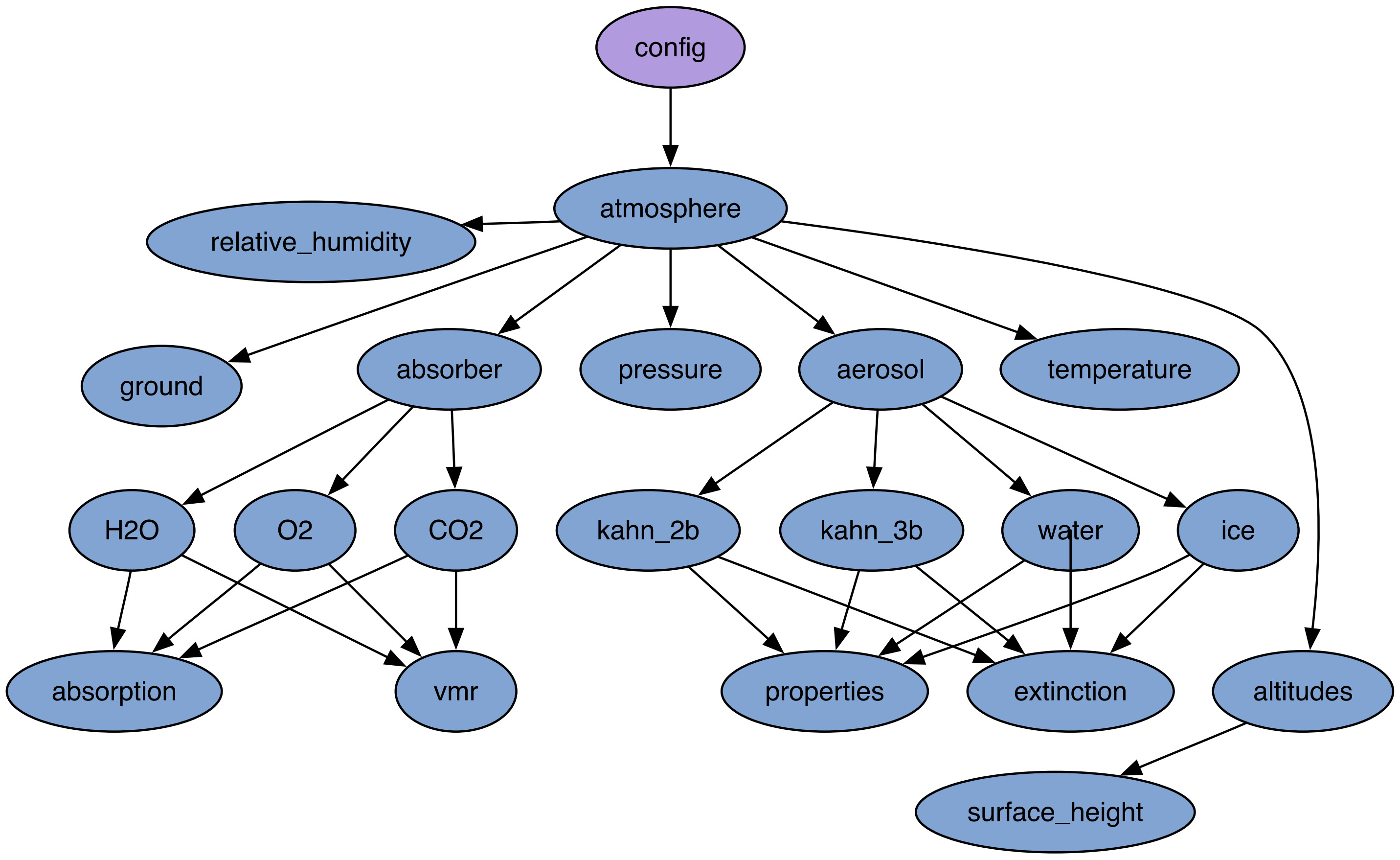
- Provides a "business" logic layer connecting wrapped C++ classes with inputs/outputs
- Structured as a logical hierarchy that matches the framework interface design
- Simplifies connecting components together and ensuring their input types through "Creator" classes
- Provides a keyword/value interface to connecting data and configuration into the framework
- Does not tie the hands of users by enforcing a strict expectation of inputs, ie very flexible

CONFIGURATION HIERARCHY



Each oval evaluates to data or a class

ATMOSPHERE HIERARCHY



Highlights the different pieces of information needed to construct an example atmosphere

EXAMPLE CONFIG PIECE

```
'absorber': {
    'creator': creator.absorber.AbsorberAbsco,
    'gases': ['CH4', 'CO', 'CO2', 'O3', 'H2O', 'HNO3', 'NO', 'NO2', 'N2O', 'NH3' ],
    # Default gas definition when a custom block named for the gas is not supplied
    'default_gas_definition': {
        'creator': creator.absorber.AbsorberGasDefinition,
        'vmr': {
            'creator': creator.absorber.AbsorberVmrLevel,
            'value': {
                'creator': creator.absorber.GasVmrApriori,
            },
        },
        'absorption': {
            'creator': creator.absorber.AbscoHdf,
            'table_scale': 1.0,
            'filename': "{absco_base_path}/{gas!l}_cris.h5"
```

USING INTERFACE

```
from refractor.factory import process_config
import cris_config
config_inst = process_config(cris_config.config_def)
```

EVALUATED CONFIG

```
>>> pprint(config_inst, indent=2)

{ 'atmosphere': <refractor_swig.atmosphere_oco.atmosphereoco; proxy="" of="" <swig="" obje
  'common': { 'absco_base_path': '/mnt/data1/absco/cris',
              'band_reference': <refractor_swig.array_with_unit.arraywithunit_double_1; pr
              'constants': <refractor_swig.default_constant.defaultconstant; proxy="" of='
              'desc_band_name': ['LW', 'MW', 'SW'],
              'hdf_band_name': ['LW', 'MW', 'SW'],
              'num_channels': 3,
              'stokes_coefficients': array([[1., 0., 0., 0.],
       [1., 0., 0., 0.],
       [1., 0., 0., 0.]])},
  'forward_model': <refractor_swig.standard_forward_model.standardforwardmodel; proxy="" o
  'instrument': <refractor_swig.ils_instrument.ilsinstrument; proxy="" of="" <swig="" obje
  'radiative_transfer': <refractor_swig.lidort_rt.lidortrt; proxy="" of="" <swig="" object
  'scenario': { 'altitude': <refractor_swig.array_with_unit.arraywithunit_double_1; proxy=
```

USING PIECES

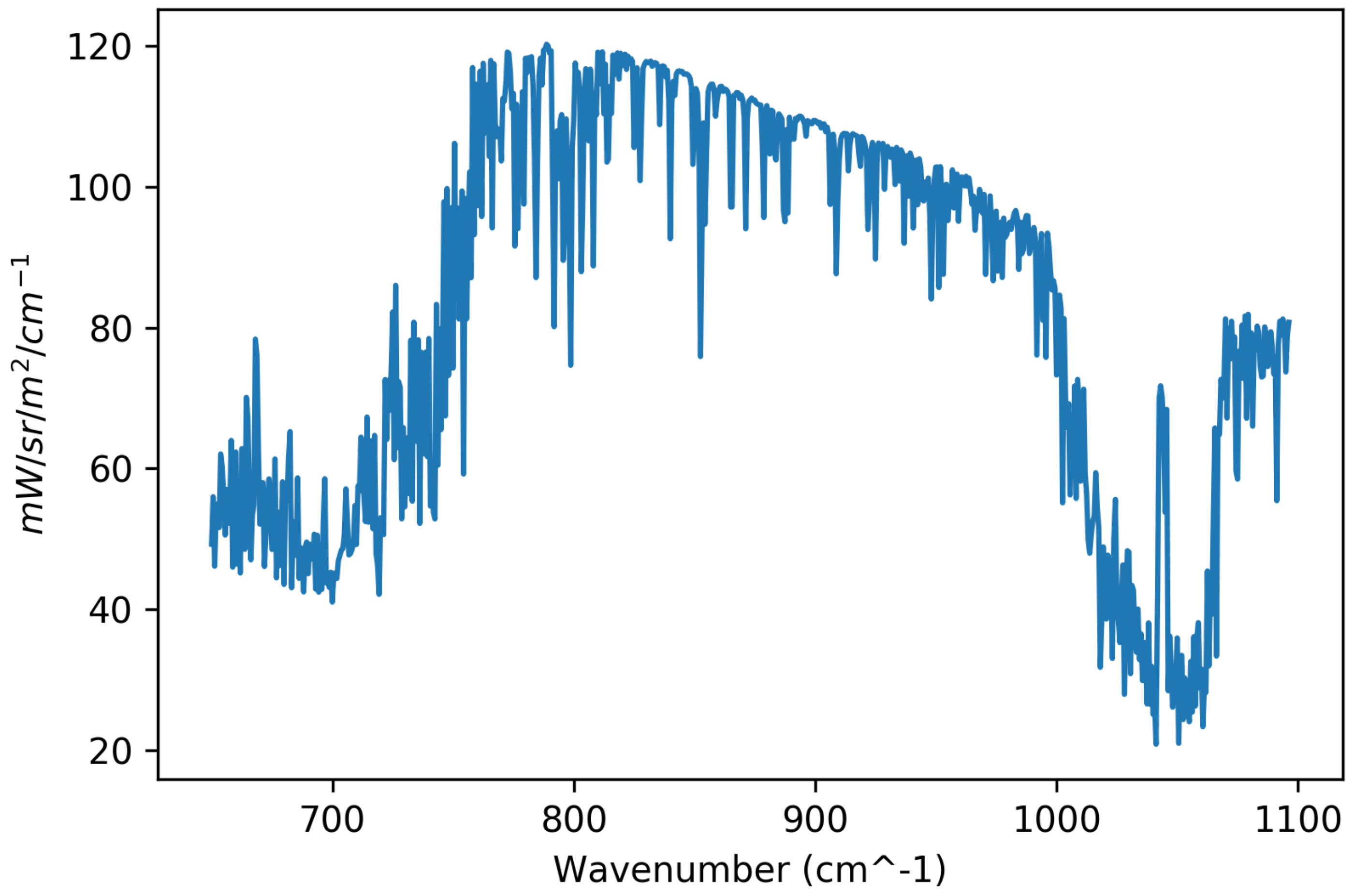
```
from refractor import framework as rf
from matplotlib.pyplot import *

fm = config_inst.forward_model
meas_grid = fm.spectral_grid.low_resolution_grid(channel_idx).data

channel_index = 0
rad = fm.radiance(channel_idx)

plot(meas_grid, rad.spectral_range.data / 1e-3)
ylabel("$ mW / sr / m^2 / cm^{-1} $")
xlabel("Wavenumber (cm^-1)")
```

RESULT


120
100
80
60
40
20
$mW/sr/m^2/cm^{-1}$
700
800
900
1000
1100
Wavenumber (cm^-1)

AUTO DERIVATIVES

$$\begin{tikzcd}
\boxed{\texttt{f(x) \{...\};}} \arrow[rrr, "\text{Automatic differentiation}"] & & & \boxed{\texttt{df(x) \{...\};}} \\
\boxed{y = f(x)} \arrow[u, "\text{human programmer}"'] \arrow[rrr, dashed, "\text{symbolic differentiation (human/computer)}"'] & & & \boxed{y' = f'(x)} \arrow[u, dashed]
\end{tikzcd}$$

TECHNIQUE

- Use a technique called Rall numbers, as described in Scientific and Engineering C++ by Barton and Hackman
- Named for the original book by L.B. Rall Automatic Differentiation, Springer-Verlag, Berlin, New York, 1981
- Works well with code that calculates Jacobians in other ways

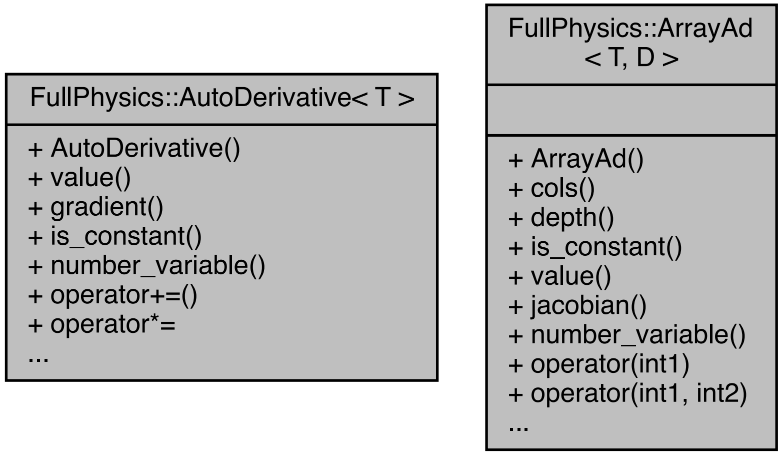
TECHNIQUE

- Only adds a bit more arithmetic operations than original equations, although is not as fast as hand-coded derivative calculations
- For locations in the code that profiling show as bottlenecks, we can replace the auto derivate calculation with hand-created code that gives the same results
- Easy to create derivatives for equations in code that does not already have it, just by changing types

IMPLEMENTATION

- Instead of using a standard C double we use a new C++ class AutoDerivative that keeps track of both a value and the gradient of the value with respect to the state vector elements.
- All normal mathematical functions are overloaded to use the chain rule to propagate the gradient through the calculation
- Instead of using normal Blitz++ array class use a ArrayAD class to keep jacobians with matrix values

CLASS OVERVIEW



INTEGRATION WITH OTHER TECHNIQUES

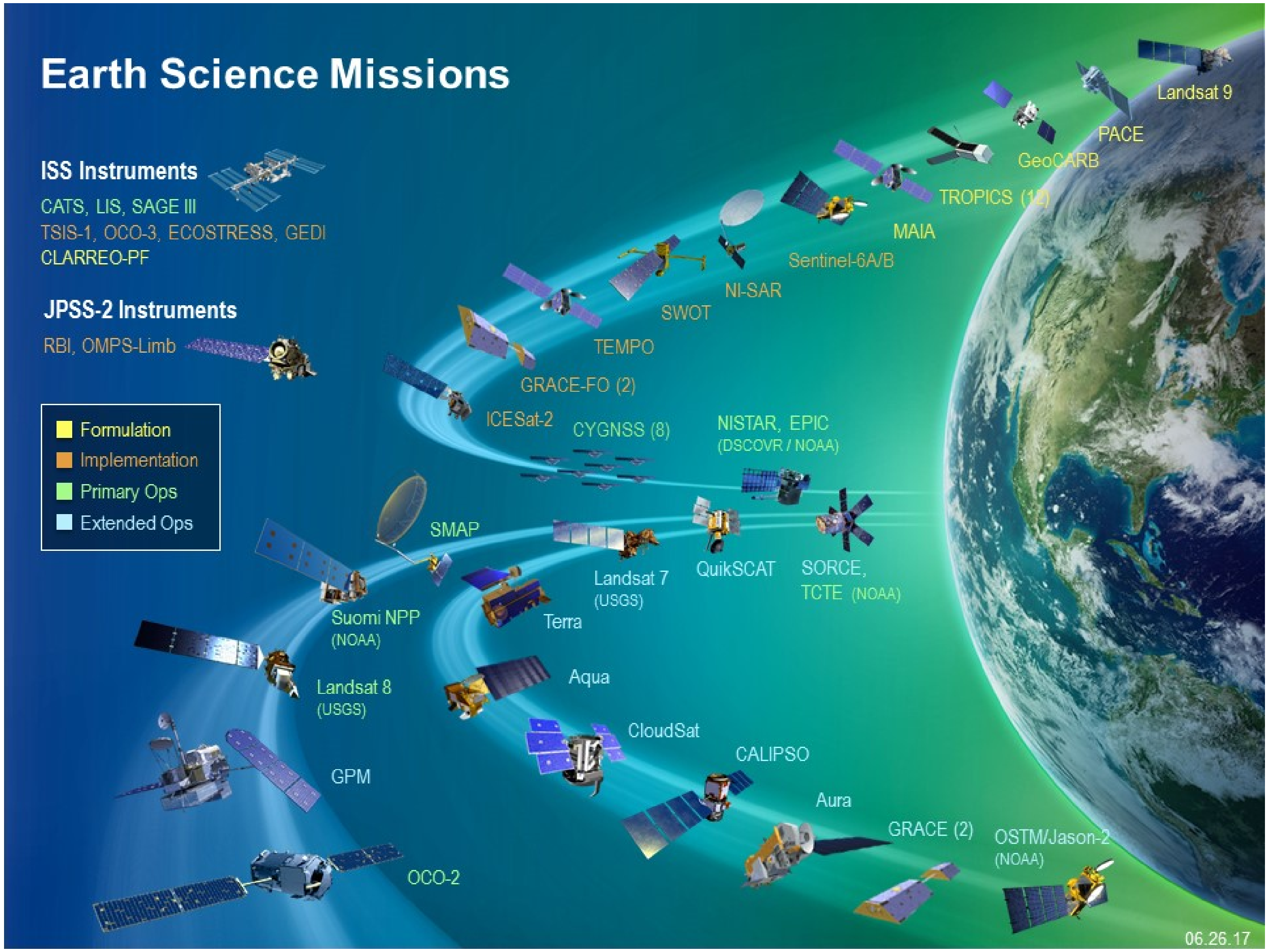
- Our radiative transver code, LIDORT is already calculating Jacobians
- We calculate the Jacobian of the parameter up to the point where LIDORT is called
- Use LIDORT itself to propagate through the RT calculation
- Take the results of LIDORT, wrap them up as AutoDerivative objects, and continue using this to propagate through the remaining forward model calculation.

EXAMPLE

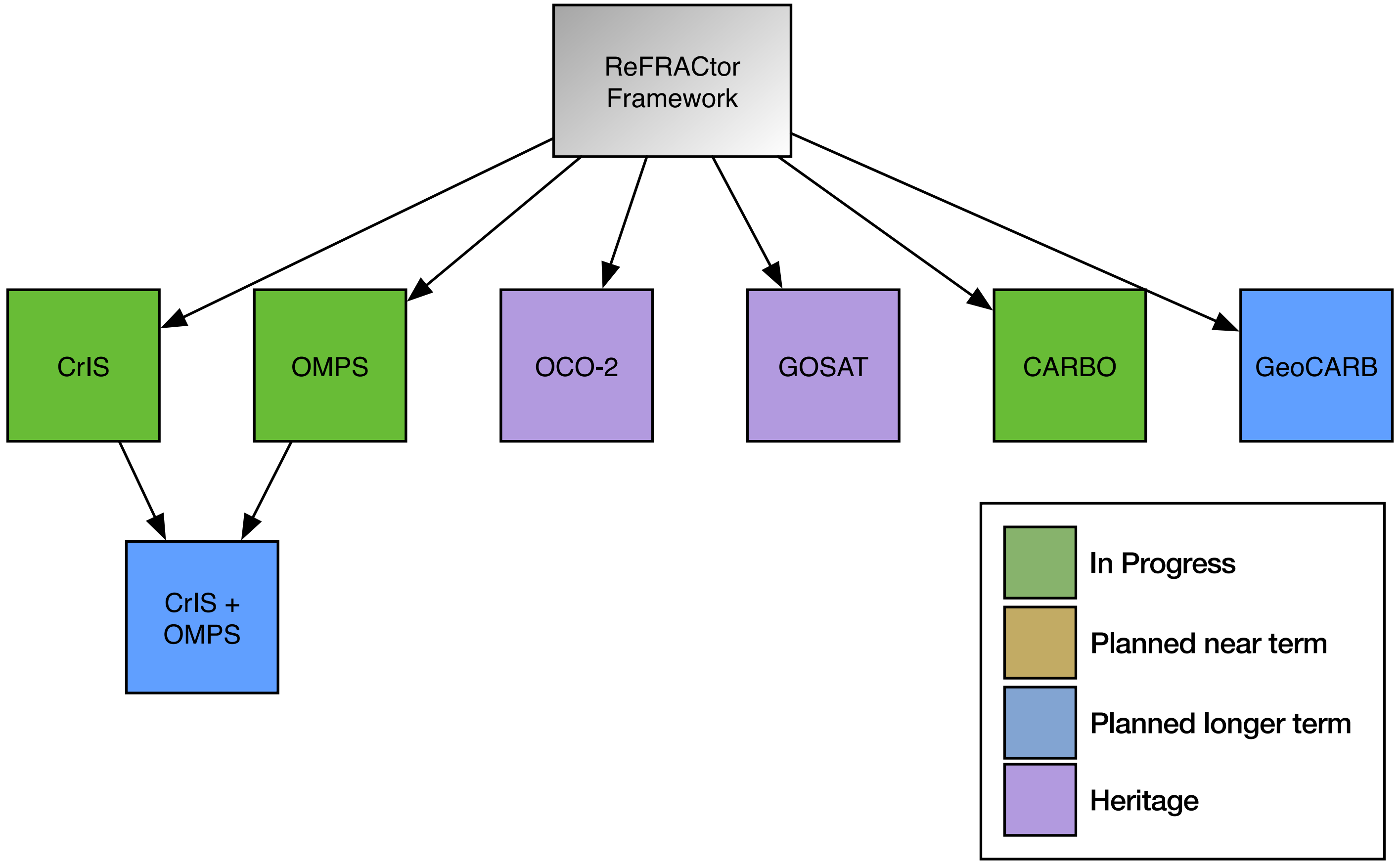
```
double x = 1;
double y = 2;
double z = x * x + x * y + 2 * y * y;
// Result is 11
```

```
AutoDerivative<double> x(1, 0, 2);
AutoDerivative<double> y(2, 1, 2);
AutoDerivative<double> z = x * x + x * y + 2 * y * y;
// Result is 11 gradient is [4, 9]
```

MULTIPLE INSTRUMENT SUPPORT



INSTRUMENTS



IMPLEMENTATION

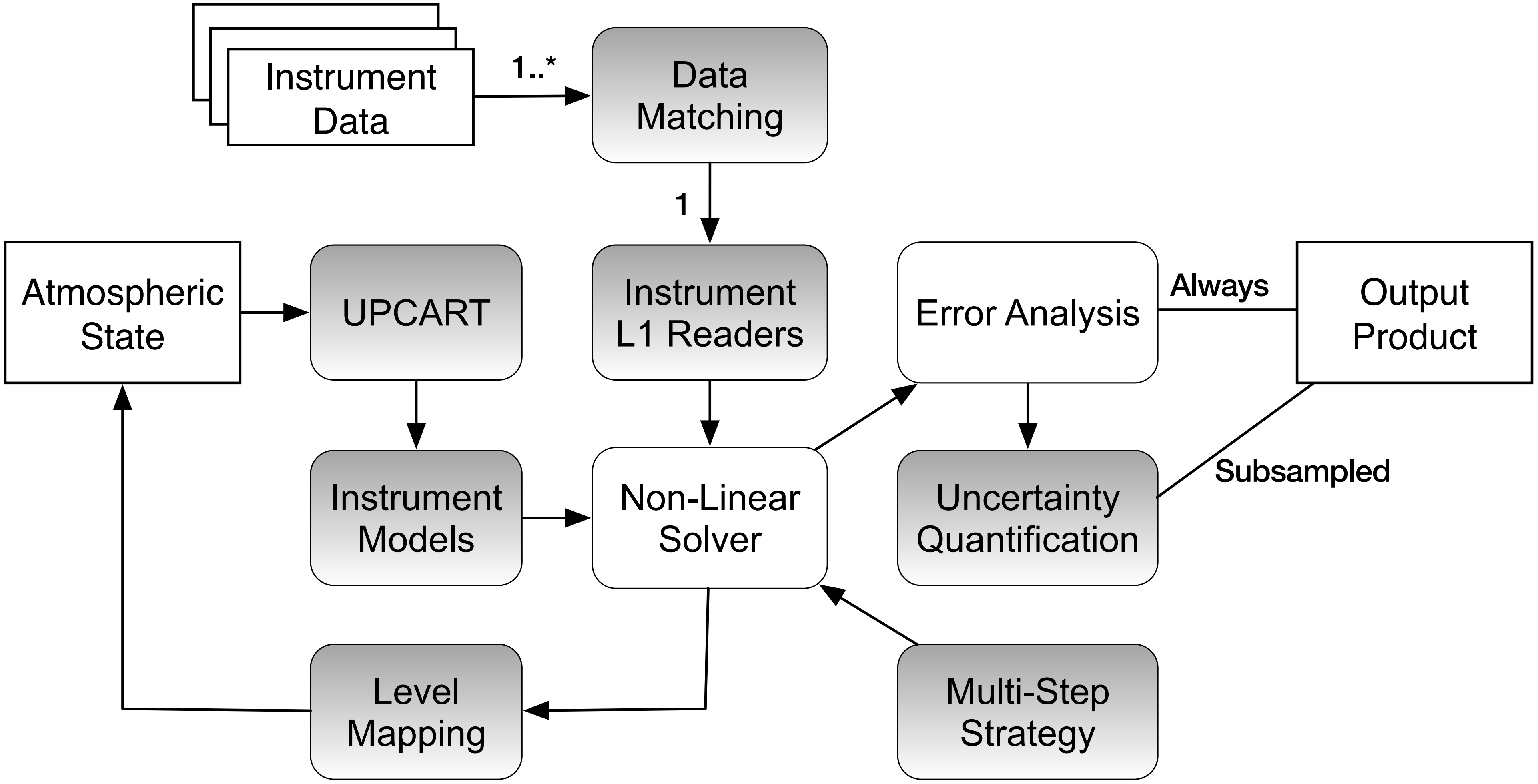
- Adding support for multiple instruments through framework’s abstract interfaces and separate source code repositories
- Code is added instead of modifying framework
- Each instrument specific repository just has the code specific to that instrument plus a configuration

NEEDED FOR EACH INSTRUMENT

- Measured data reader, ie Level 1B reader
- Instrument model
  - Instrument Line Shape
  - Instrument radiance effects
- Domain specific radiative transfer algorithms
- Absorption Coefficient tables

CURRENT WORK

PLANNED WORK



FRAMEWORK DEVELOPMENT

- Took heritage repository from OCO-2, cleaned up and moved out non reusable components
- Changed the build/compilation system to use CMake, a more modern and extensible compilation system
- Created Python configuration layer for building “business logic” of instrument implementations

INSTRUMENT IMPLEMENTATIONS

- Cross-track Infrared Sounder (CrIS)
  - Built Level 1B data reader
  - Built configuration
  - Working on adopting instrument team’s instrument line shape (ILS) data
- Carbon Balance Observatory (CARBO)
  - Created prototype Python only configuration for use in instrument simulations
- Orbiting Carbon Observatory (OCO-2)
  - Inherited, but needs to be put in place in a separate repository

RADIATIVE TRANSFER

- Upgraded interfaced versions of existing radiative transfer software to latest versions
  - LIDORT
  - Two stream model
- Integrated support for performing thermal infrared radiative transfer calculations
  - OCO-2 code has only been using in the near infrared

RETRIEVAL METHODS

- Upgraded existing Non-Linear Least Squared (NLLS) solvers by integrating newer versions with several more years of work on them beyond OCO-2 versions
- Added additional NLLS solvers that make different assumptions about the problem state
- Working on integrating white-box implementation of Moire's algorithm

UNCERTAINTY QUANTIFICATION

- Created a generalized Python code for simulation state vectors from mixture of statistical distributions
- Creating a report on canonical examples for doing Uncertainty Quantification, focused on implications for algorithm design
  - How is bias/variance impacted when selected state vector elements are fixed and not retrieved?
  - What is the impact of fast/simplified RT in the retrieval when true forward model is more complex?
- Investigating simplified RT usage of state vector mixtures

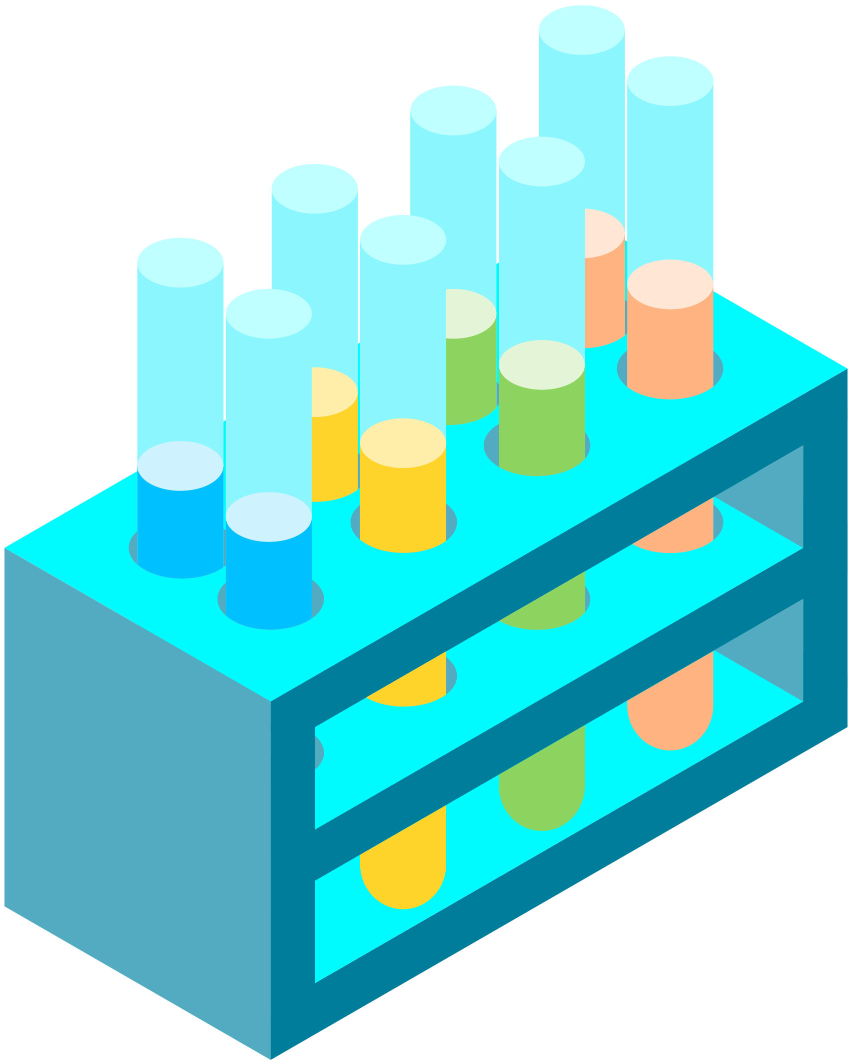
NEAR TERM ACTIVITIES

- Software for computing absorption coefficient (ABSCO) tables from infrared to ultraviolet spectral regions
- Integration of a radiative transfer speed up method using principal component analysis
- OMPS instrument implementation
- Improve retrieval strategy techniques

END

BACKUP MATERIAL

TESTING



UNIT TESTING

- Using the Boost Unit Test Framework
- Tests individual pieces of the system, for example individual classes
- Unit tests are good at:
  - Correct handling of error conditions
    Ex: Out of bounds, bad values
  - Detecting memory leaks in component
  - Repeatability of calculation
    Do we get the same answer today as last week?
  - Comparison between calculations
    Ex: Jacobian calculation vs. finite difference

    Jacobian

- Unit tests are bad at anything that involves analysis
  We are happy to get the same wrong answer every time

UNIT TEST EXAMPLE

```
#include "auto_derivative.h"
#include "unit_test_support.h"

using namespace FullPhysics;
using namespace blitz;

BOOST_FIXTURE_TEST_SUITE(auto_derivative, GlobalFixture)

BOOST_AUTO_TEST_CASE(basic_test)
{
  double xv = 2; double yv = 3;
  AutoDerivative<double> x(xv, 0, 2);
  AutoDerivative<double> y(yv, 1, 2);
  AutoDerivative<double> z;

  z = 2 * x + y;
```

UNIT TEST EXAMPLE

```
$ make fast_check run_test=auto_derivative/basic_test
                      ...

0%   10   20   30   40   50   60   70   80   90   100%
|----|----|----|----|----|----|----|----|----|----|
Running 1 test case...
***************************************************

*** No errors detected
```

SPEED

60
5
10
15
20
25
30
35
40
45
50
55
30
5
10
15
20
25
LA PUNTUALITÀ È LA
CORTESIA DEI RE

PLACEHOLDER

SUPPORT TOOLS

PLACEHOLDER

INSTRUMENT DETAILS

CROSS-TRACK INFRARED SOUNDER (CRIS)

- Fourier transform spectrometer
- LWIR (9.14 - 15.38 μm)
- MWIR (5.71 - 8.26 μm)
- SWIR (3.92 - 4.64 μm)
- Launched October 28, 2011 on S-NPP. Also 2017 on JPSS-1

OZONE MAPPING PROFILER SUITE (OMPS)

- Six-slit prism spectrometer.
- Single, two-dimensional (740×340) charged-coupled device (CCD) focal-plane array
- Spectral range from 290 to 1000 nm
- Launched October 28, 2011 on S-NPP. Also 2017 on JPSS-1

HISTORY

HISTORY

- Changes for GOSAT processing begins May 2009
- Mike Smyth joins team November 2009
- Start using Git, Feb 2010
- Last Fortran only version 2.06.02 March 2010
- Refactoring started March 2010
- First refactored version 2.07.00 July 2010
- Lua configuration introduced June 2011
- OCO-2 algorithmic work begins Dec 2011
- OCO-2 Launched on July 2, 2014
- ReFRACtor Project begins September 14, 2017

REFERENCES

Barton, John J., and Lee R. Nackman. Scientific and Engineering C. Addison-Wesley, Reading, MA, 1994.

Bösch, H., L. Brown, R. Castano, M. Christi, B. Connor, D. Crisp, A. Eldering et al. "Orbiting Carbon Observatory (OCO)-2 Level 2 Full Physics Retrieval Algorithm Theoretical Basis Document.", Version 2.0 Rev 2, March 13, 2015: http://disc.sci.gsfc.nasa.gov/OCO-2/documentation/oco-2-v6/OCO2_L2_ATBD.V6.pdf

Bowman, K. W., Rodgers, C. D., Kulawik, S. S., Worden, J., Sarkissian, E., Osterman, G., Steck, T., Lou, M., Eldering, A., Shephard, M., Worden, H., Lampel, M., Clough, S., Brown, P., Rinsland, C., Gunson, M., and Beer, R.: Tropospheric Emission Spectrometer: retrieval method and error analysis, IEEE Trans. Geosci. Remote Sens., 44, 1297–1307, 2006.

Connor B. J., H. Boesch, G. Toon, B. Sen, C. Miller and D. Crisp (2008), Orbiting Carbon Observatory: Inverse method and prospective error analysis, J. Geophys. Res., 113, D05305, doi: 10.1029/2006JD008336.

Connor, B., Bösch, H., McDuffie, J., Taylor, T., Fu, D., Frankenberg, C., ... & Hobbs, J. (2016). Quantification of uncertainties in OCO-2 measurements of XCO 2: simulations and linear error analysis. Atmospheric Measurement Techniques, 9(10), 5227.

Cressie, N., R. Wang, M. Smyth, and C. E. Miller (2016), Statistical bias and variance for the regularized inverse problem: Application to space-based atmospheric CO2 retrievals. J. Geophys. Res., 121, 5526-5537, doi:10.1002/2015JD024353.

Deeter, M. N., et al., Operational carbon monoxide retrieval algorithm and selected results for the MOPITT instrument, J. Geophys. Res., 108(D14), 4399, doi:10.1029/2002JD003186, 2003.

Cuesta, J., M. Eremenko, X. Liu, G. Dufour, Z. Cai, M. H ̈opfner, T. von Clarmann, P. Sellitto, G. Foret, B. Gaubert, M. Beekmann, J. Orphal, K. Chance, R. Spurr, and J. M. Flaud (2013), Satellite observation of lowermost tropospheric ozone by multispectral synergism of IASI thermal infrared and GOME-2 ultraviolet measurements over Europe, Atmos. Chem. Phys., 13(19), 9675–9693, doi:10.5194/acp-13-9675-2013.

Chivers, I., & Sleightholme, J. (2016, December). Compiler support for the Fortran 2003 and 2008 Standards Revision 20. In ACM SIGPLAN Fortran Forum (Vol. 35, No. 3, pp. 29-50). ACM.

Fu, D., K. W. Bowman, H. M. Worden, V. Natraj, J. R. Worden, S. Yu, P. Veefkind, I. Aben, J. Landgraf, L. Strow, and Y. Han (2016), High-resolution tropospheric carbon monoxide profiles retrieved from CrIS and TROPOMI, Atmos. Meas. Tech., 9(6), 2567–2579, doi:10.5194/amt-9-2567-2016.

Fu, D., J. R. Worden, X. Liu, S. S. Kulawik, K. W. Bowman, and V. Natraj (2013), Characterization of ozone profiles derived from Aura TES and OMI radiances, Atmos. Chem. Phys., 13(6), 3445–3462, doi:10.5194/acp-13-3445-2013.

Hobbs, J., A. Braverman, N. Cressie, R. Granat, and M. Gunson (2017), Simulation-based uncertainty quantification for estimating atmospheric CO2 from satellite data. (under review).

Hua, Hook. "High-Resiliency and Auto-Scaling of Large-Scale Cloud Computing for OCO-2 L2 Full Physics Processing." 2015 AGU Fall Meeting. Agu, 2015.

Kopparla, P., V. Natraj, R. J. D. Spurr, R.-L. Shia, Y. L. Yung, and D. Crisp (2016), A Fast and Accurate PCA Based Radiative Transfer Model: Extension to the Broadband Shortwave Region, J. Quant. Spectrosc. Radiat. Transfer, 173, 65–71, doi:10.1016/j.jqsrt.2016.01.014.

Kuai, L., J. Worden, S. Kulawik, K. Bowman, M. Lee, S. C. Biraud, J. B. Abshire, S. C. Wofsy, V. Natraj, C. Frankenberg, D. Wunch, B. Connor, C. Miller, C. Roehl, R. L. Shia, and Y. Yung (2013), Profiling tropospheric CO2 using Aura TES and TCCON instruments, Atmos. Meas. Tech., 6(1), 63–79, doi:10.5194/amt-6-63-2013.

Landgraf, J., and O. P. Hasekamp (2007), The synergistic use of thermal infrared emission and ultraviolet reflectivity measurements from space, J. Geophys. Res., 112, D08310, doi:10.1029/2006JD008097.

Liu, X., P. K. Bhartia, K. Chance, R. J. D. Spurr, and T. P. Kurosu (2010), Ozone profile retrievals from the ozone monitoring instrument, At- mos. Chem. Phys., 10(5), 2521–2537, doi:10.5194/acp-10-2521-2010.

Livesey, N. J., & Wu, D. L. (2004). EOS MLS retrieval processes algorithm theoretical basis. Jet Propul. Lab. Doc. D, 16159.

Martin, R. (1994). OO design quality metrics. An analysis of dependencies, 12, 151-170.

Meyer, Bertrand. Object-oriented software construction. Vol. 2. New York: Prentice hall, 1988.

National Research Council. 2007. Earth Science and Applications from Space: National Imperatives for the Next Decade and Beyond. Washington, DC: The National Academies Press. doi:https://doi.org/10.17226/11820.

National Aeronautical and Space Administration, “NASA Strategic Plan 2014,” NP-2014-01-965-HQ; NASA Headquarters, Washington, DC, 2014.

Natraj, V., H. Boesch, R. J. D. Spurr, and Y. L. Yung (2008), Retrieval of XCO2 from simulated Orbiting Carbon Observatory measurements using the fast linearized R-2OS radiative transfer model, J. Geophys. Res., 113, D11212, doi:10.1029/2007JD009017.

Natraj, V., Jiang, X., Shia, R. L., Huang, X., Margolis, J. S., & Yung, Y. L. (2005). Application of principal component analysis to high spectral resolution radiative transfer: A case study of the O2A band. Journal of Quantitative Spectroscopy and Radiative Transfer, 95(4), 539-556.

Natraj, V., and R. J. D. Spurr (2007), A fast linearized pseudo-spherical two orders of scattering model to account for polarization in vertically inhomogeneous scattering-absorbing media, J. Quant. Spectrosc. Radiat. Transfer, 107(2), 263–293, doi:10.1016/j.jqsrt.2007.02.011.

Natraj, V., Shia, R. L., & Yung, Y. L. (2010). On the use of principal component analysis to speed up radiative transfer calculations. Journal of Quantitative Spectroscopy and Radiative Transfer, 111(5), 810-816.

Natra j, V., X. Liu, S. Kulawik, K. Chance, R. Chatfield, D. P. Edwards, A. Eldering, G. Francis, T. Kurosu, K. Pickering, R. Spurr, and H. Worden (2011), Multi-spectral sensitivity studies for the retrieval of tropospheric and lowermost tropospheric ozone from simulated clear-sky GEO-CAPE measurements, Atmospheric Environment, 45(39), 7151–7165, doi: http://dx.doi.org/10.1016/j.atmosenv.2011.09.014.

Reid, J. (2007, April). The new features of fortran 2003. In ACM SIGPLAN Fortran Forum (Vol. 26, No. 1, pp. 10-33). ACM.

Rodgers, Clive D. Inverse methods for atmospheric sounding: theory and practice. Vol. 2. World scientific, 2000.

Rozanov, A., Rozanov, A., Buchwitz, M., Kokhanovsky, A. and Burrows, J.P., 2016. User's guide for the software package SCIATRAN.

Schimel, D., Sellers, P., Moore III, B., Chatterjee, A., Baker, D., Berry, J., ... & Duren, R. (2016). Observing the carbon-climate system. arXiv preprint arXiv:1604.02106.

Schoeberl, M. R., Douglass, A. R., Hilsenrath, E., Bhartia, P. K., Barnett, J., Gille, J., ... & DeCola, P. (2004). Earth observing system missions benefit atmospheric research. Eos, 85(18), 177-178.

Spurr, R., Natraj, V., Lerot, C., Van Roozendael, M., & Loyola, D. (2013). Linearization of the Principal Component Analysis method for radiative transfer acceleration: Application to retrieval algorithms and sensitivity studies. Journal of Quantitative Spectroscopy and Radiative Transfer, 125, 1-17.

Warner, J. X., Z. Wei, L. L. Strow, C. D. Barnet, L. C. Sparling, G. Diskin, and G. Sachse (2010), Improved agreement of AIRS tropospheric carbon monoxide products with other EOS sensors using optimal estimation retrievals, Atmos. Chem. Phys., 10(19), 9521–9533, doi:10.5194/acp-10-9521-2010.

Worden, J., X. Liu, K. Bowman, K. Chance, R. Beer, A. Eldering, M. Gunson, and H. Worden (2007), Improved tropospheric ozone profile retrievals using OMI and TES radiances, Geophys. Res. Lett., 34, L01809, doi:10.1029/2006GL027806.

Worden, H. M., M. N. Deeter, D. P. Edwards, J. C. Gille, J. R. Drummond, and P. N´ed´elec (2010), Observations of near-surface carbon monoxide from space using MOPITT multispectral retrievals, J. Geophys. Res., 115(D18).

Worden, J. R., A. J. Turner, A. Bloom, S. S. Kulawik, J. Liu, M. Lee, R. Weidner, K. Bowman, C. Frankenberg, R. Parker, and V. H. Payne (2015), Quantifying lower tropospheric methane concentrations using GOSAT near-IR and TES thermal IR measurements, Atmos. Meas. Tech., 8(8), 3433–3445, doi:10.5194/amt-8-3433-2015.