EOLDAS User Manual

Prof. Philip Lewis
UCL

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

User Training Manual

- We want other researchers to use eoldas
- Had discussions e.g. Edinburgh
- Also useful teaching tool

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

Eodas code

- Mainly in python
  - Portable, flexible
- RT model (semidiscrete) in FORTRAN
  - Called from python module
- Documentation:
  - Self documenting python modules
  - User manual
    - Web interface, pdf etc.

eo-ldas

support to science element

EOLDAS Documentation

Release 0

P Lewis, J Gomez-Dans

September 02, 2011

National Centre for Earth Observation

NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

EOLDAS

This is a tutorial guide for the EOLDAS software, a prototype Data Assimilation system for estimating land surface properties from radiometric measurements (Earth Observation data).

The EOLDAS project is funded by ESA as part of the Support to Science Element (STSE) of the Earth observation Envelope Programm

The aims of the projects are:

- To develop and document a generic data assimilation scheme to assist the retrieval of geophysical parameters from medium re optical EO data
- To develop a prototype software package implementing selected aspects of the scheme
- To validate the prototype using multi-sensor EO data and field measurements

The project is led by the UK National Centre for Earth Observation (NCEO) with a team that includes:

The University of Reading

University College London

Swansea University

The Joint Research Center

FSU Jena

FastOpt GMbH

Scisys Ltd

The primary authors of this software and documentation are:

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

Structure

- Introduction
- Simple Observation Operator
  - Dynamic model and Identity Op
- RT Observation Operator
  - Obs Operator and Identity Op
- DA in eoldas
  - Obs Operator and dynamic model

Introduction

- What problems solve with eoldas

$$J_{obs} = (1/2)(\mathbf{y_{obs}} - H(\mathbf{x}))^T (\mathbf{C_{obs}}^{-1} + \mathbf{C_H}^{-1})(\mathbf{y_{obs}} - H(\mathbf{x})),$$

$$J_{prior} = (1/2)(\mathbf{y_{obs}} - I(\mathbf{x}))^T \mathbf{C_{obs}}^{-1} (\mathbf{y_{obs}} - I(\mathbf{x})),$$

$$J_{smooth} = (1/2)\gamma^2 (\Delta\mathbf{x})^T \mathbf{C_{smooth}}^{-1} (\Delta\mathbf{x}).$$

$$J_{smooth}(\mathbf{x}) + J_{obs}(\mathbf{x}) + \ldots = \min(\mathbf{x})$$

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

A simple data assimilation example using an identity observation operator

```
# An EOLDAS confioguration file for
# a simple Identity operator and a regulariser

[parameter]
location = ['time']
limits = [[1,365,1]]
names = gamma_time,NDVI
solve = 0,1
datatypes = x

[parameter.result]
filename = output/Identity/NDVI_Identity.params
format = 'PARAMETERS'
help_filename='Set the output state filename'

[parameter.x]
datatype = x
names = $parameter.names
default = 200,0
help_default="Set the default values of the states"
apply_grid = True
sd = [1.]*len($parameter.names)
bounds =  [[0.000001,1000000],[-1,1]]
```

Setup problem in config file

Parameters section

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

```
[operator]
modelt.name=DModel_Operator
modelt.datatypes = x
obs.name=Operator
obs.datatypes = x,y
```

```
[operator.modelt.x]
names = $parameter.names
sd = [1.0]*len($operator.modelt.x.names)
datatype = x

[operator.modelt.rt_model]
model_order=1
help_model_order='The differential model order'
wraparound=periodic,365
```

```
[operator.obs.x]
names = $parameter.names[1:]
sd = [1.0]*len($operator.obs.x.names)
help_sd='Set the observation sd'
datatype = x

[operator.obs.y]
control = [mask]
names = $parameter.names[1:]
sd = [0.15]*len($operator.obs.x.names)
help_sd="set the sd for the observations"
datatype = y
state = data/Identity/random_ndvi1.dat
help_state = "Set the y state vector"

[operator.obs.y.result]
filename = output/Identity/NDVI_fwd.params
```

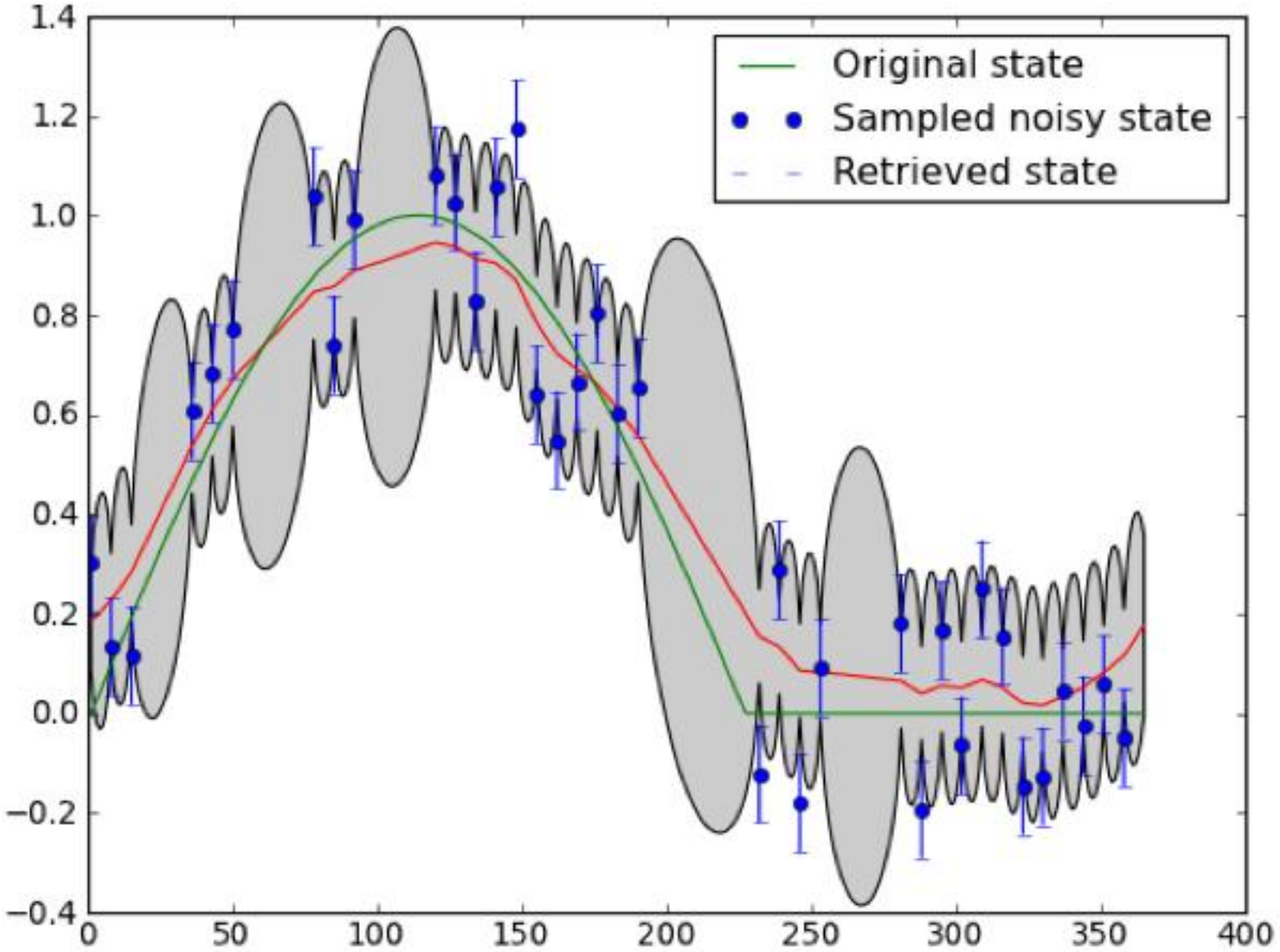
```
[general]
doplot=True
help_do_plot='do plotting'
```

Interaction: call from python

- Set up problem / data from within python
- Here:
  - Synthetic noisy NDVI time series
  - With correlated gaps
    - Simulate cloud impacts

Interaction: call from python

```
gamma = ideal_gamma*0.5

# set op file names
xfile = 'output/Identity/NDVI_Identity.params'
yfile = 'output/Identity/NDVI_fwd.params'

# initialise optuions for DA overriding any in config files
cmd = 'eoldas ' +  \
            ' --conf=eoldas_config.conf --conf=confs/Identity.conf ' + \
            ' --logfile=mylogs/Identity.log ' + \
            ' --calc_posterior_unc ' + \
            ' --parameter.solve=0,1 ' + \
            ' --parameter.result.filename=%s '%xfile +\
            ' --parameter.x.default=%f,0.0 '%(gamma) + \
            ' --operator.obs.y.result.filename=%s'%yfile +\
            ' --operator.obs.y.state=%s'%file+\
            ' --operator.modelt.rt_model.model_order=%d '%model_order

# initialise eoldas
self = eoldas.eoldas(cmd)
# solve DA
```



National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

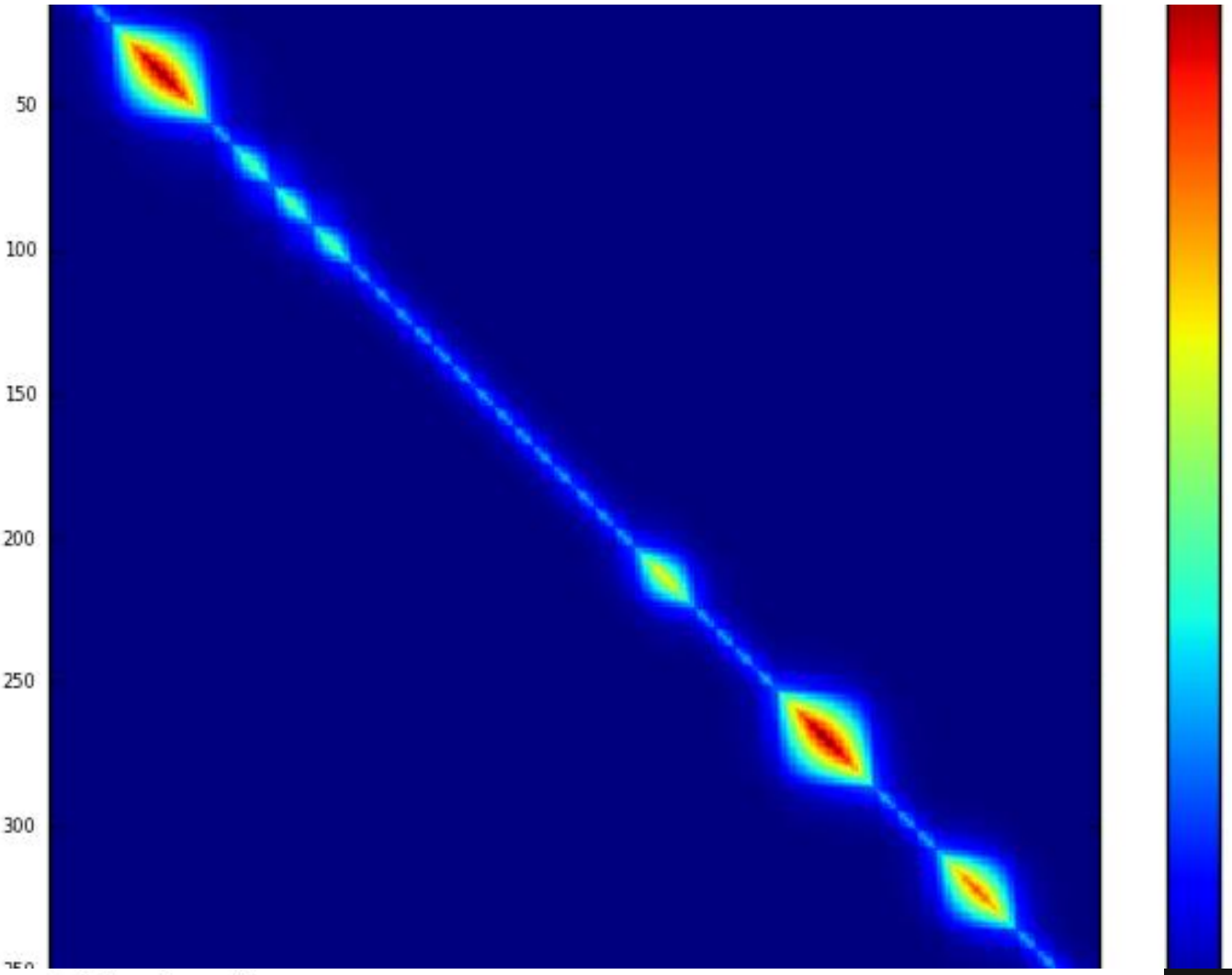
UCL

Show user how to ‘delve’ into code

```
# initialise eoldas
self = eoldas.eoldas(cmd)
# solve DA
self.solve(write=True)

# now pull some data out of the eoldas

# the 'root' of the DA data structure is in self.solver.root

root = self.solver.root

# The state vector data are stored in root.x
# with ancillary information in root.x_meta
# so the state vector is e.g. in root.x.state
# and the names are in root.x_meta.state

state_names = root.x_meta.state
state = root.x.state

# The sd representation of the posterior is in root.x.sd
# This is all set up in eoldas_Solver.py
# All storage is of type ParamStorage, an extended
# dictionary structure. You can explore it interactively
# with e.g. root.dict().keys() or self.keys() since
# self here is a straight dictionary
sd = root.x.sd

# The full inverse Hessian is in self.solver.Ihessian
Ihessian = self.solver.Ihessian

# A mask to reduce this to only the state variables
# being targeted (solve == 1) is through a call to:
NDVI_Ihessian = self.solver.unloader(None,Ihessian,M=True)
```

e.g. eoldas doesn’t output graphics of posterior

posterior



National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

Running EOLDAS from the command line

- bin/eoldas –conf=confs/Identity.conf

```
BRDF 67 7 645 858.5 469 555 1240 1640 2130  0.003 0.004 0.004 0.015 0.013 0.01 0.006
181 1 62.830002 -83.040001 37.910000 23.219999 0.032900 0.068600 0.018000 0.028400 0.098100 0.097600 0.081900
182 1 33.980000 99.540001 44.970001 39.290001 0.050500 0.091800 0.030000 0.045300 0.123200 0.137100 0.124800
184 1 51.660000 99.300003 46.889999 42.349998 0.056200 0.106000 0.029500 0.053800 0.151600 0.154600 0.141400
185 1 32.880001 -81.900002 40.540001 31.510000 0.039200 0.069600 0.022900 0.034400 0.089800 0.102500 0.092800
186 1 63.049999 100.470001 48.910000 45.169998 0.058200 0.138600 0.038900 0.055200 0.184600 0.183800 0.143100
187 1 6.430000 -79.459999 42.090000 35.259998 0.043700 0.074600 0.026400 0.038700 0.103100 0.118400 0.103600
189 1 22.389999 97.720001 43.790001 38.750000 0.044500 0.088400 0.027100 0.040200 0.129500 0.135800 0.127600
190 1 57.400002 -82.720001 38.029999 26.480000 0.050700 0.087700 0.024600 0.045200 0.119900 0.118000 0.098100
191 1 44.040001 99.419998 45.619999 41.990002 0.062800 0.107800 0.035600 0.057100 0.141600 0.154800 0.137600
```

- Take user through example of smoothing reflectance dataset
- Then explain that lots of the ‘noise’ can be explained by BRDF … observation operator

Radiative transfer modelling for optical remote sensing

1. xlai : LAI, the single sided leaf area per unit ground area
2. xhc : The canopy height
3. rpl : The leaf radius/dimension (same units as canopy height)
4. xkab : C_{ab}, the concentration of chlorophyll a+b $[\mu g cm^{-2}]$,
5. scen : C_{sen}, the proportion of senescent material (fractional, between 0 and 1).
6. xkw : C_w, equivalent leaf water $[cm]$,
7. xkm : C_{dm}, dry matter $[\mu g cm^{-2}]$,
8. xleafn : N, the number of leaf layers,
9. xs1 : Soil PC1 (soil brightness)
10. xs2 : Soil wetness
11. xs3 : not used
12. xs4 : not used
13. lad : leaf angle distribution (coded 0-5). See Semidiscrete code for details.

Radiative transfer modelling for optical remote sensing

However, the section [parameter.x.assoc_transform] contains:

```
invtransform=$parameter.names
transform=$parameter.names
```

```
[parameter.x.assoc_transform]
xlai=np.exp(-xlai/2.)
xkab=np.exp(-xkab/100.)
xkw=np.exp(-xkw*50.)
xkm=np.exp(-100.*xkm)
```

```
[parameter.x.assoc_invtransform]
xlai=-2.*np.log(xlai)
xkab=-100.*np.log(xkab)
xkw=-(1./50.)*np.log(xkw)
xkm=-(1./100.)*np.log(xkm)
```

```
[parameter.assoc_solve]
gamma_time = 0
lad = 0
xs3 = 0
xs4 = 0
xhc = 0
rpl = 0
```

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

Radiative transfer modelling for optical remote sensing

Set up conditions for Obs op

```
[operator.obs.rt_model]
model=rtmodel_ad_trans2
use_median=True
help_use_median = "Flag to state whethe
bounds = [400,2500,1]
help_bounds = "The spectral bounds (min
ignore_derivative=False
help_ignore_derivative = "Set to True t
```

Run from cmd line

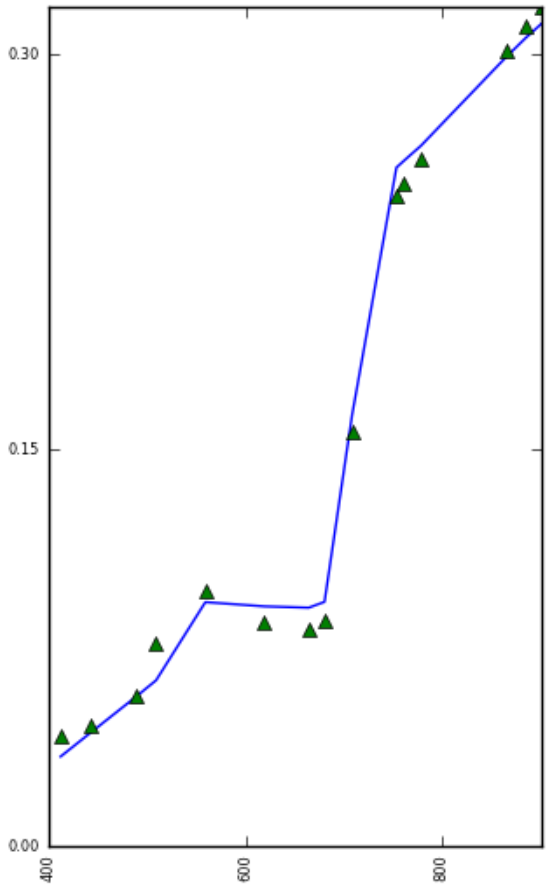
../eoldaslib/eoldas.py –conf=confs/meris_single.conf –calc_posterior_unc –parameter.limits="[[232,232,1]]"

which tells eoldas to solve for 8 state varibles: xlai, xkab, scen, xkw, xkm, xleafn, xs1,xs2 from any meris data found on day 232.

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

Solve for 8 RT states from single MERIS



Write out results data files and appropriate graphs

From MERIS, find solution (optimisation converges)

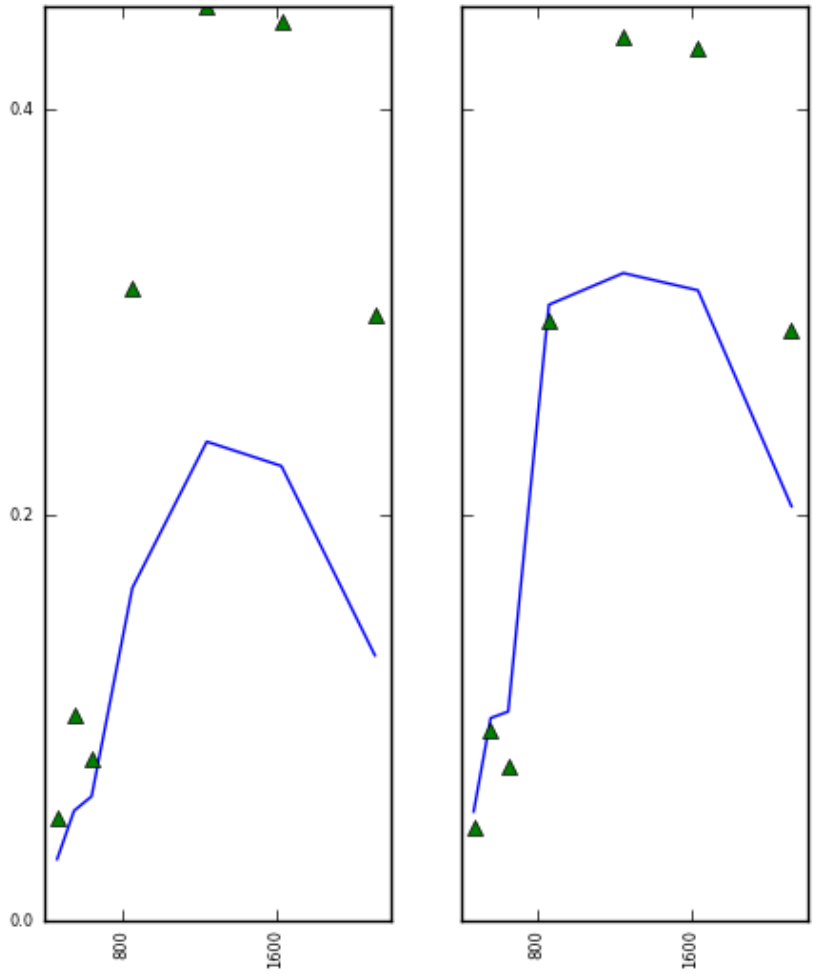
But very high uncertainty and correlation

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

Use MERIS solution to predict MODIS

../eoldaslib/eoldas.py –conf=confs/meris_single.conf –parameter.limits='[[232,232,1]]' –passer –conf=confs/modis_single.conf



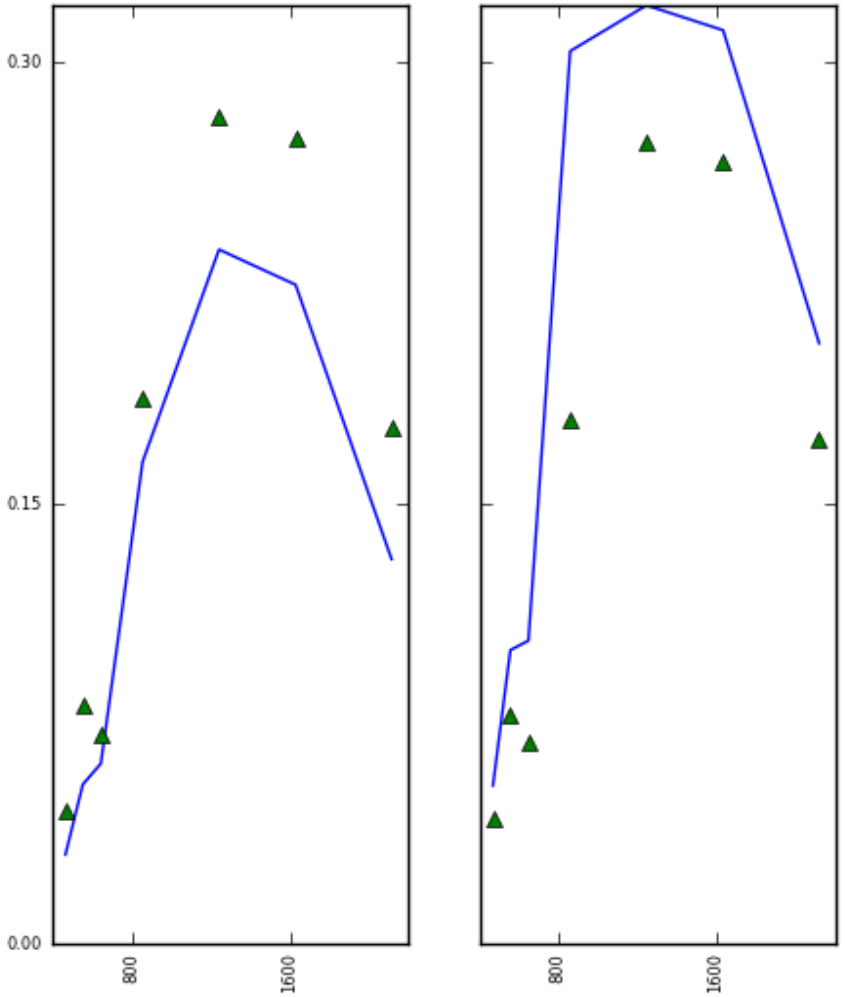
Poor estimation …

High uncertainty in state

UCL

Try to solve from MODIS (2 samples)

- no convergence



Show how to solve from both MERIS and MODIS

- Set up Operator for MODIS data
- Set up Operator for MERIS data
- Run from simple cmd line

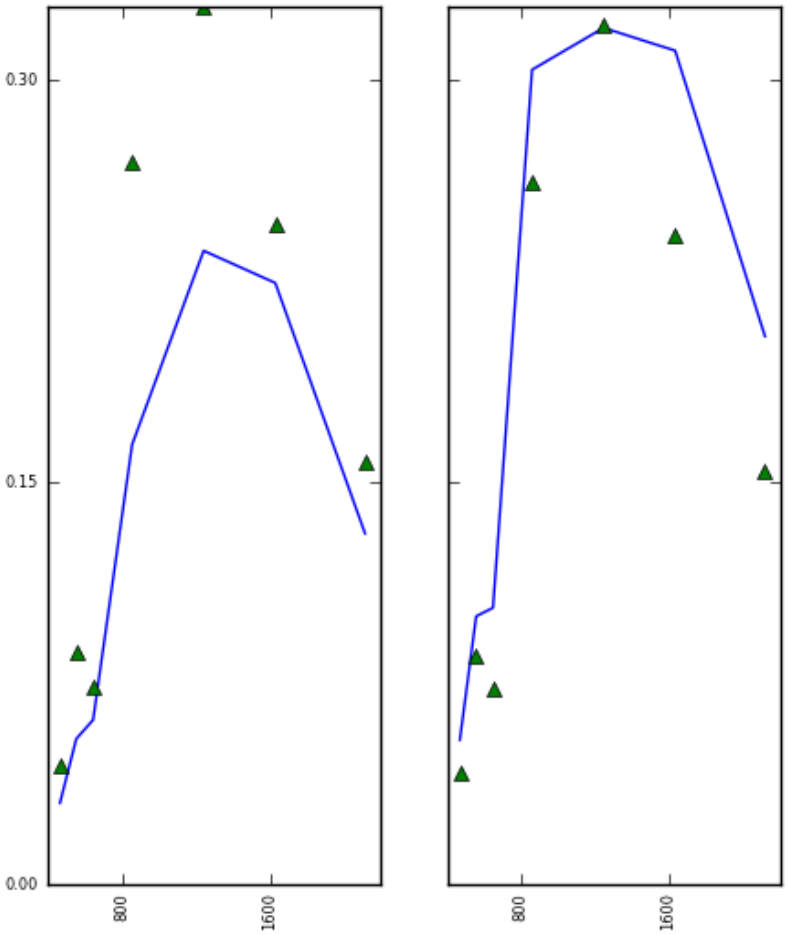
- But no convergence … ill-conditioned
- (as expected)

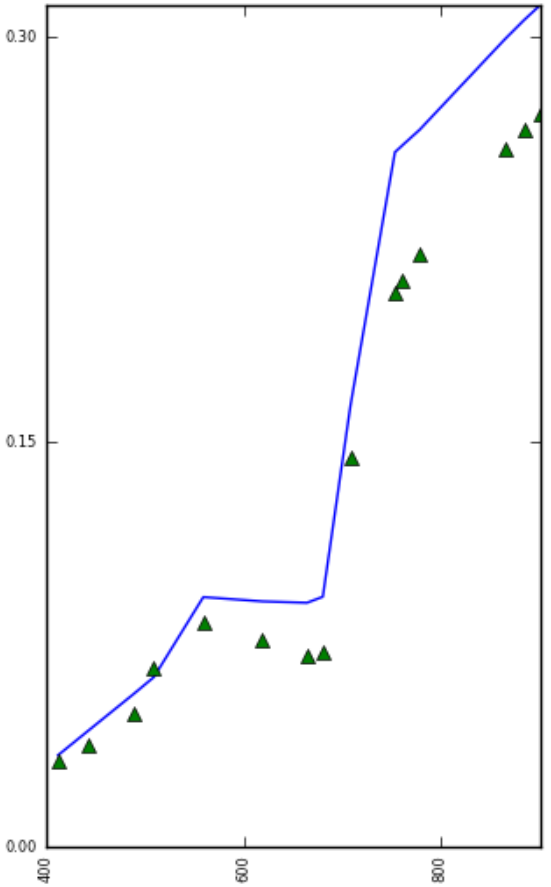
Other forms of constraint

- Prior (Identity operator)
- Core operator class
- Set high σ

```
[operator]
obs_modis.name=Observation_Operator
obs_modis.datatypes = x,y
prior.name=Operator
prior.datatypes = x,y

[operator.prior.result]
filename = output/modis/MODIS_MODIS_WW_1_A_1_prior_c.dat

[operator.prior.x]
names = $parameter.names[1:]
sd = [1.0]*len($operator.prior.x.names)
datatype = x

[operator.prior.y]
names = $parameter.names[1:]
sd = 0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5,0.5
state = 0.5,5,0.01,0.5,0.5,0.5,0.5,1.5,1.0,0.0,0,0,5
datatype = x
```

Now problem solveable …





National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

recap

- Explained rationale for eoldas
- Shown by example how to use when observation and measurement in same domain
  - Smooth, using prior and differential constraints
- Shown how to apply observation operator
  - RT model
  - observation and measurement in different domains
- Shown how RT ‘inversion’ ill-posed
  - prior aids conditioning

Data Assimilation and EOLDAS

Cost functions with set of operators:

$$J = (1/2)(\mathbf{y_{obs}} - H(\mathbf{x}))^T (\mathbf{C_{obs}}^{-1} + \mathbf{C_H}^{-1})(\mathbf{y_{obs}} - H(\mathbf{x})),$$

Main sets:

- **Identity (Prior) Operator**
- **Model Operator**
  - Relates states from different time/space
- **Observation operator**
  - Maps from x to y

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

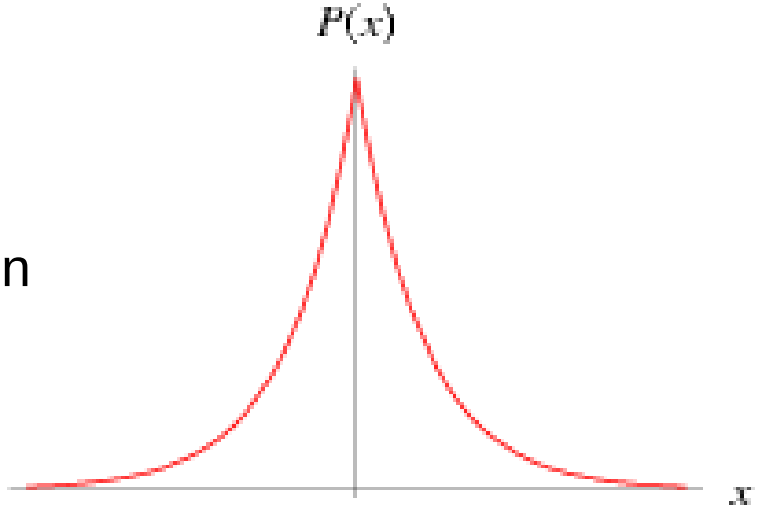
Operators in eoldas

- Operators H(x) embedded in cost function class
  - Main role to provide J, J', (J'')
  - Can make methods specific to H(x)
    - E.g. time/space independence

Operators in eoldas

- eoldas_Operator
- eoldas_DModel_Operator

$$H(\mathbf{x}) = \Gamma \Delta^N (\mathbf{x}) = 0$$



Can set lag for differential (default 1)

equivalent to Laplace distribution

rather peaky

Role for broader-topped filters

apply correlation function

Can apply D in time or space

Use multiple operators if want multi dimensions

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL

Operators in eoldas

- eoldas_Observation_Operator
  - ‘generic’ interface to multi-angle/wavelength RT models
  - Spectral concept efficient
    - Only loads wavelengths needed
    - Can make runfaster by using waveband median
      - Then refine solution with full bandpass
  - E.g. interfacing to FORTRAN library
    - this is sometimes difficult … flavours of FORTRAN … flavours of compiler …
    - It works, but quite difficult to give generic advice to users on compiling
      - Use gfortran VX.X+ etc.

Operators in eoldas

- eoldas_Kernel_Operator
  - ‘non spectral’ models
- Observation_Operator_AtCorr
- Observation_Operator_AtCorrCpld
  - Derived from eoldas_Observation_Operator

    Very slow at present

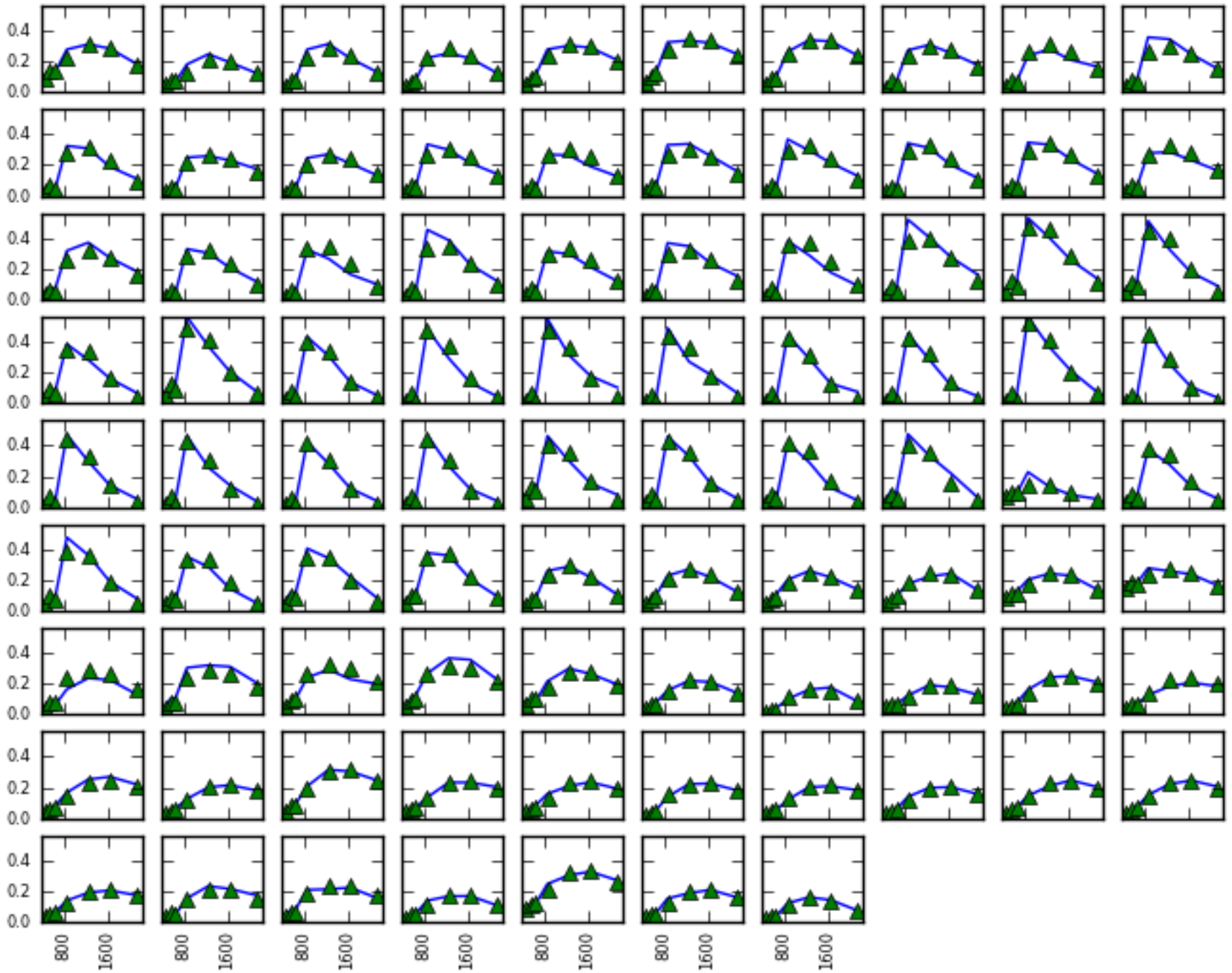
Limitations

- Some speed limitations
  - Rt models …
- C_H currently not treated
  - Though has placeholder in code
- Not all operators can use full C at present
  - So safest to use SD only

Using Model and Observation Operators

Temporal model and semidiscrete Observation Operator

Example of varying gamma … movie


0.4
0.2
0.0
800
1600

Conclusions

- Developed flexible eoldas
  - Interface through command line
  - Or from python code
- Variational
  - Minimsation of cost function
  - Though should be able to use sequentially
- Only demonstrated temporal here
  - But can also do spatial …
  - And multiple space/time

Conclusions

- Main RT Obs op semi discrete
  - But interface for others
    - Spectral/ not spectral
  - Includes linear kernel models
    - Should put in linear solver?

Conclusions

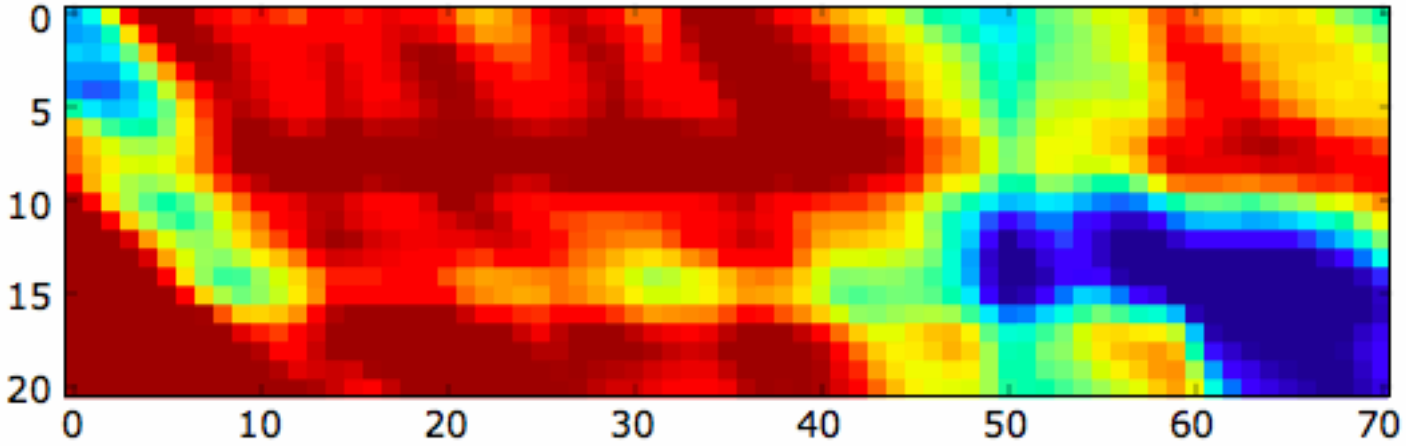
- Model
  - Only have differential
    - But potential for changing ‘shape’ built in
    - And this is of itself very powerful
    - Need edge preserving methods
      - All essentially adapt gamma
  - Want users to implement their own operator
  - Other than education/learning/trying ideas, this is probably the main practical use of eoldas

Conclusions

- Ready to start exploiting it …

Disturbance DA

Spatio-temporal regularisation: can treat multi-resolution data



Col 88

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

UCL