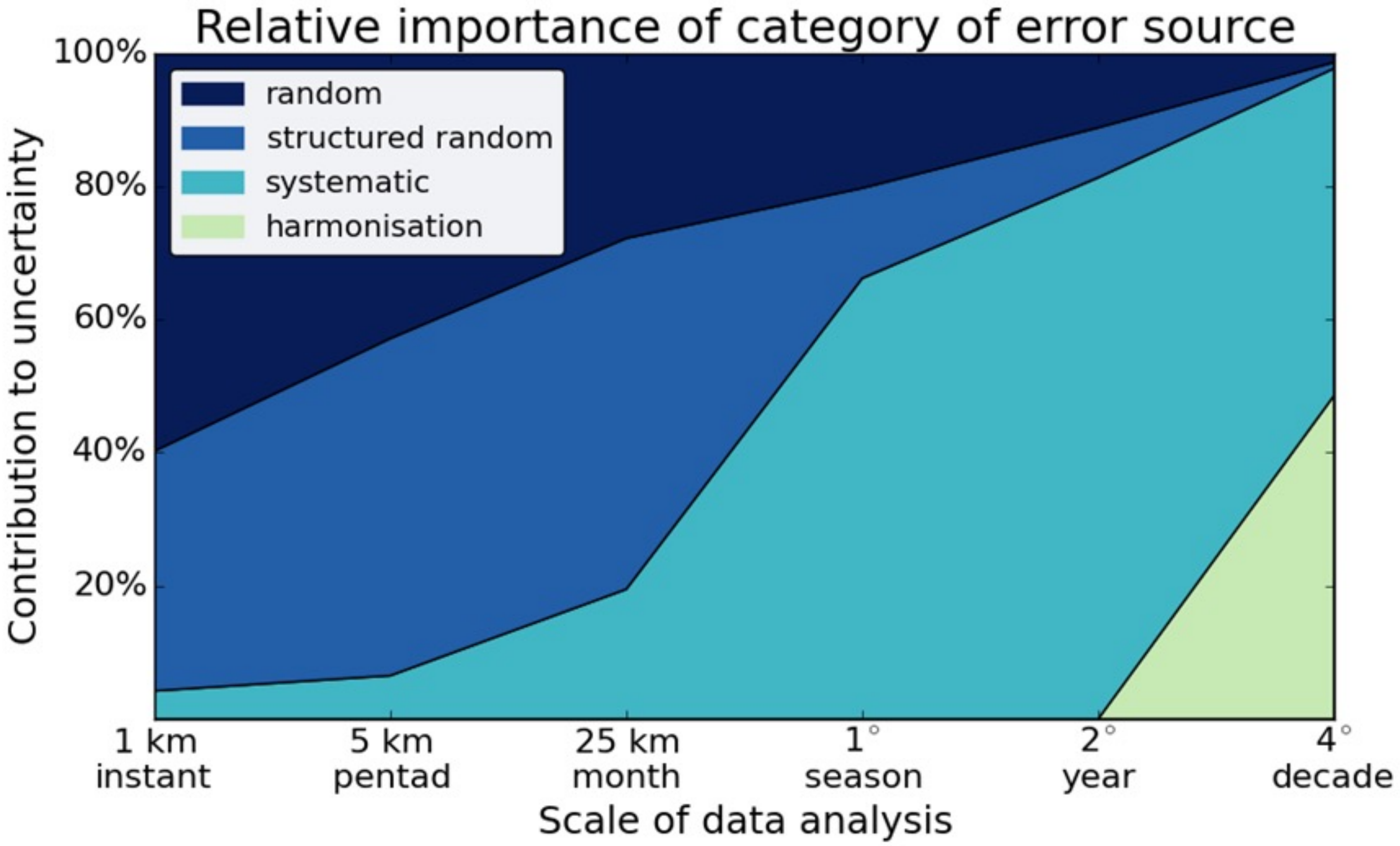
National Physical Laboratory

Community Tools for Metrology

QA4EO Cal/Val Workshop #3

Sam Hunt & Pieter De Vis

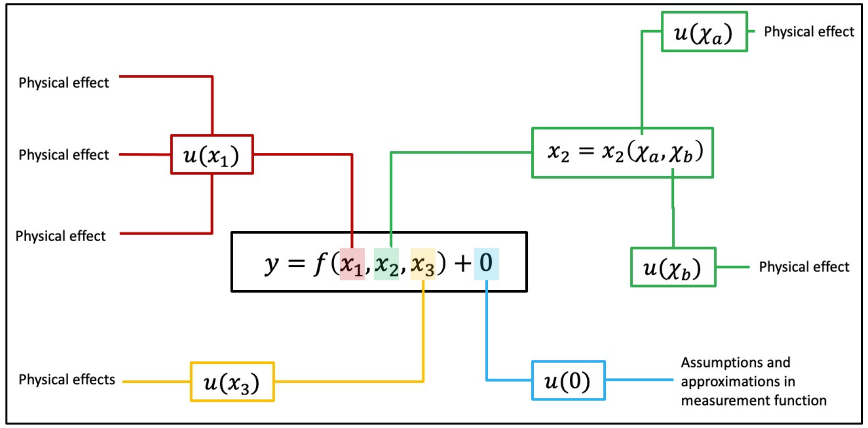
Why do we care about error-correlation?



Uncertainty Analysis

NPL National Physical Laboratory

- In full error-covariance matrices are impractical to evaluate and store for EO data
- “FIDUCEO-style” approach to uncertainty analysis offers a solution by parameterising error-covariance structure
- How to take this to the next step? How do I store and make use of this information in data?



Uncertainty Tree Diagram

		Comments
Name of effect		A unique name
Affected term in measurement function		Name and standard symbol
Instruments in the series affected		List names
Correlation type and form	Pixel-to-pixel [pixels]	From a set of defined correlation forms
	from scanline to scanline [scanlines]	
	between images [images]	
	Between orbits [orbit]	
	Over time [time]	
Correlation scale	Pixel-to-pixel [pixels]	As needed to define type
	from scanline to scanline [scanlines]	
	between images [images]	
	Between orbits [orbit]	
	Over time [time]	
Channels/bands	List of channels / bands affected	Channel names
	Error correlation coefficient matrix	A matrix
Uncertainty	PDF shape	Functional form
	units	Units
	magnitude	
Sensitivity coefficient		Value, equation or parameterisation of sensitivity of measurand to term

Effect Table

Encoded Observations

NPL National Physical Laboratory

Geospatial data is encoded with complex metadata, though users typically never have to interact with it.

Encoded Observations

NPL National Physical Laboratory

Geospatial data is encoded with complex metadata, though users typically never have to interact with it.

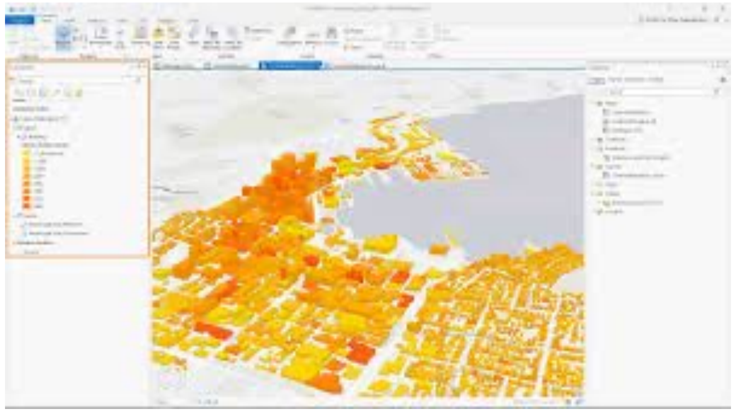
Example: Geocoding

1. Data is accompanied with standardised metadata
2. Tools provide means to
   A. Interface with this information
   B. Interpret and make use of this information

OGC® Making location count. WKT

ArcGIS

Cartopy



Encoded Observations

National Physical Laboratory

Geospatial data is encoded with complex metadata, though users typically never have to interact with it.

Why not take the same approach for error-covariance information for observations?

Encoded Observations

NPL National Physical Laboratory

Geospatial data is encoded with complex metadata, though users typically never have to interact with it.

Parallel: Error-covariance encoding

1. Data is accompanied with standardised metadata
2. Tools provide means to
   A. Interface with this information
   B. Interpret and make use of this information

Fiduceo

obsarray

punpy

Digital Effects Table

Interface for Handling Information

e.g. Uncertainty propagation

Standardised Error-Covariance Metadata: Digital Effects Tables

		Comments
Name of effect		A unique name
Affected term in measurement function		Name and standard symbol
Instruments in the series affected		List names
Correlation type and form	Pixel-to-pixel [pixels]	From a set of defined correlation forms
	from scanline to scanline [scanlines]	
	between images [images]	
	Between orbits [orbit]	
	Over time [time]	
Correlation scale	Pixel-to-pixel [pixels]	As needed to define type
	from scanline to scanline [scanlines]	
	between images [images]	
	Between orbits [orbit]	
	Over time [time]	
Channels/bands	List of channels / bands affected	Channel names
	Error correlation coefficient matrix	A matrix
Uncertainty	PDF shape	Functional form
	units	Units
	magnitude	
Sensitivity coefficient		Value, equation or parameterisation of sensitivity of measurand to term

FIDUCEO Effects Table

```
double u_str_temperature(x=2, y=2, time=3);
  :_FillValue = 9.969209968386869E36; // double
  :err_corr_1_dim = "x";
  :err_corr_1_form = "custom";
  :err_corr_1_units = ; // double
  :err_corr_1_params = "err_corr_str_temperature_x";
  :err_corr_2_dim = "y";
  :err_corr_2_form = "systematic";
  :err_corr_2_units = ; // double
  :err_corr_2_params = ; // double
  :err_corr_3_dim = "time";
  :err_corr_3_form = "systematic";
  :err_corr_3_units = ; // double
  :err_corr_3_params = ; // double
  :pdf_shape = "gaussian";
```

Print out of uncertainty variable attributes for netCDF file

Digital Effects Table

Interface to Error-Covariance Metadata: obsarray

The ***obsarray*** python module provides an extension to the widely used ***xarray*** package to interface with measurement error-covariance information encoded in datasets

```
print(ds.temperature)
```

```
<xarray.DataArray 'temperature' (x: 2, y: 2, time: 3)>
array([[[16.969685,  4.8038  , 27.2702  ],
        [23.128404, 18.645507, 20.901654]],

       [[19.035036,  8.91408 ,  4.718095],
        [ 6.079486, 18.107981, 11.597675]]])
Dimensions without coordinates: x, y, time
Attributes:
    unc_comps:  ['u_ran_temperature', 'u_str_temperature', 'u_sys_temperature']
```

Interface to Error-Covariance Metadata: obsarray

National Physical Laboratory

The ***obsarray*** python module provides an extension to the widely used ***xarray*** package to interface with measurement error-covariance information encoded in datasets

```
# Inspect uncertainty variables for a particular variable

print(ds.unc["temperature"])
```

```
<VariableUncertainty>
Variable Uncertainties: 'temperature'
Data variables:
    u_ran_temperature  (x, y, time) float64 0.8485 0.2402 ... 0.9054 0.5799
    u_str_temperature  (x, y, time) float64 0.5091 0.1441 ... 0.5432 0.3479
    u_sys_temperature  (x, y, time) float64 0.5091 0.1441 ... 0.5432 0.3479
```

Interface to Error-Covariance Metadata: obsarray

NPL National Physical Laboratory

The ***obsarray*** python module provides an extension to the widely used ***xarray*** package to interface with measurement error-covariance information encoded in datasets

```
# Get total variable uncertainty

ds.unc["temperature"].total
```

```
xarray.DataArray    (x: 2, y: 2, time: 3)

array([[[1.11277669, 0.31500626, 1.78822662],
        [1.51663088, 1.22266768, 1.37061312]],

       [[1.24821081, 0.58453532, 0.3093862 ],
        [0.39865853, 1.18741975, 0.76051039]]])
```

Tools for Error-Covariance Metadata: punpy

NPL National Physical Laboratory

punpy interfaces with **obsarray** to make uncertainty propagation as efficient and easy to use as possible. All flexibility of punpy is included as optional keywords. The propagate_ds() function returns an **obsarray** dataset with combined random, systematic and structured uncertainties on measurand.

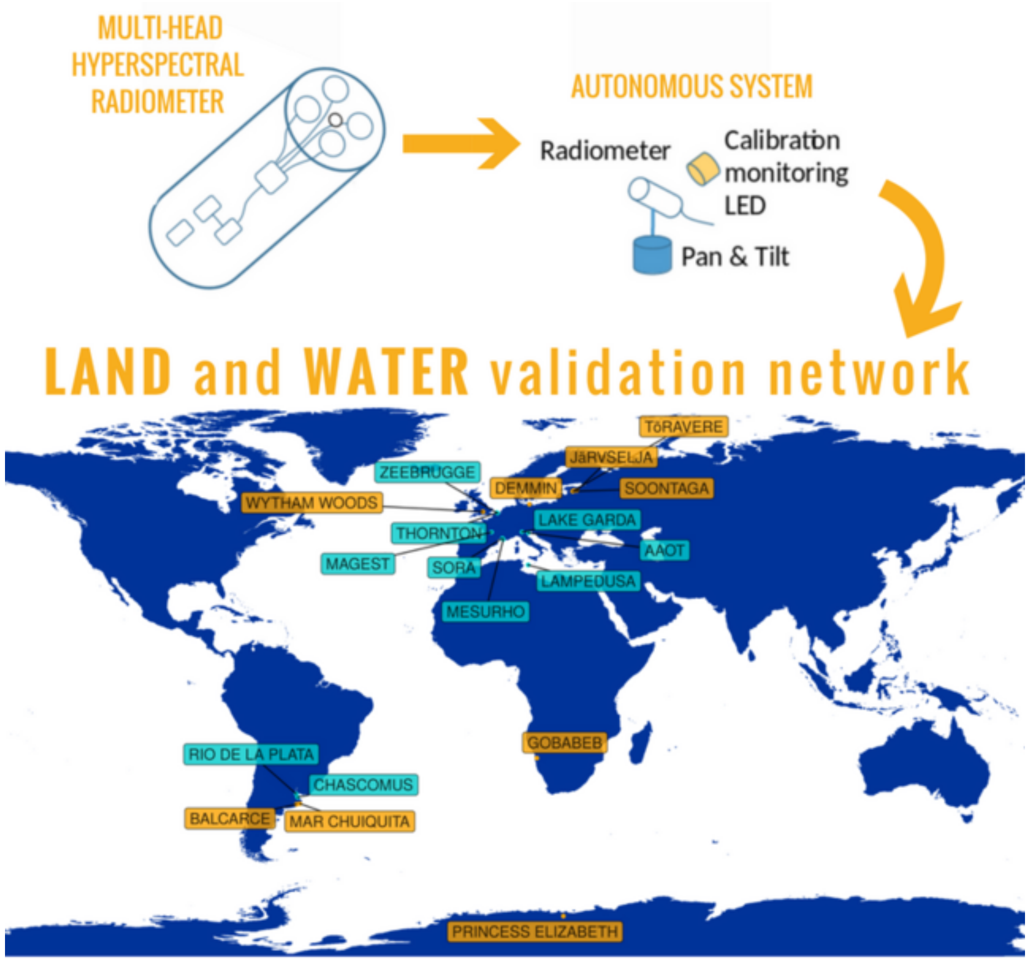
```
# Define your measurement function inside a subclass of MeasurementFunction
class GasLaw(MeasurementFunction):
    def function(self, pres, temp):
        return pres/(temp*8.134)

# create class object and pass all optional keywords for punpy
gl = GasLaw(["pressure","temperature"],steps=100000)

# propagate the uncertainties on the input quantities in ds to measurand uncertainties in ds_y
ds_y=gl.propagate_ds("V/n",ds)
```

Example: Hypernets Ground Processor

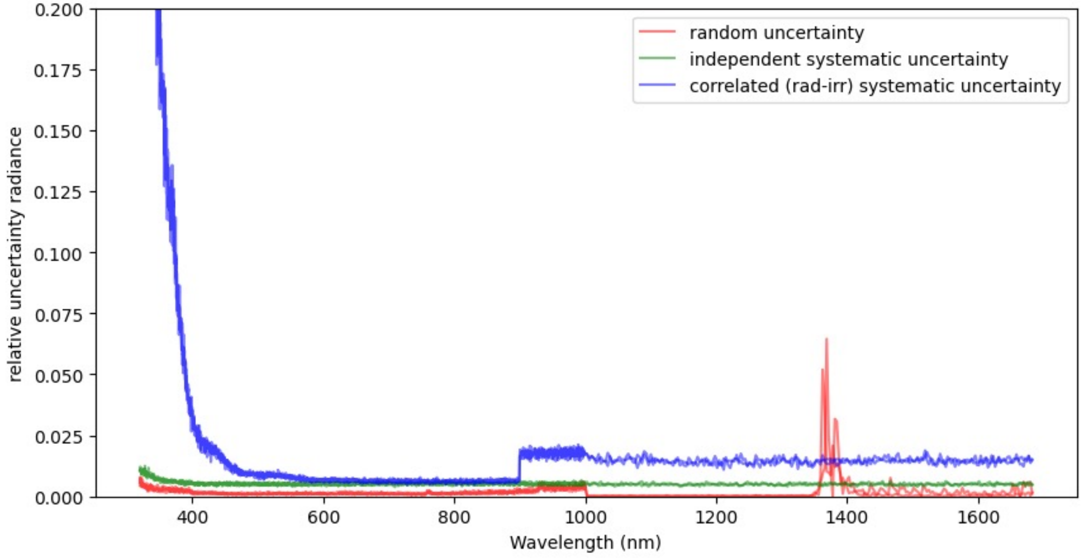
- Hypernets is an underdevelopment network of ground test sites with automated hyperspectral spectrometers for surface reflectance validation



Example: Hypernets Ground Processor

- Hypernets is an underdevelopment network of ground test sites with automated hyperspectral spectrometers for surface reflectance validation

- Uncertainty information is provided with every product, including error-correlation information.



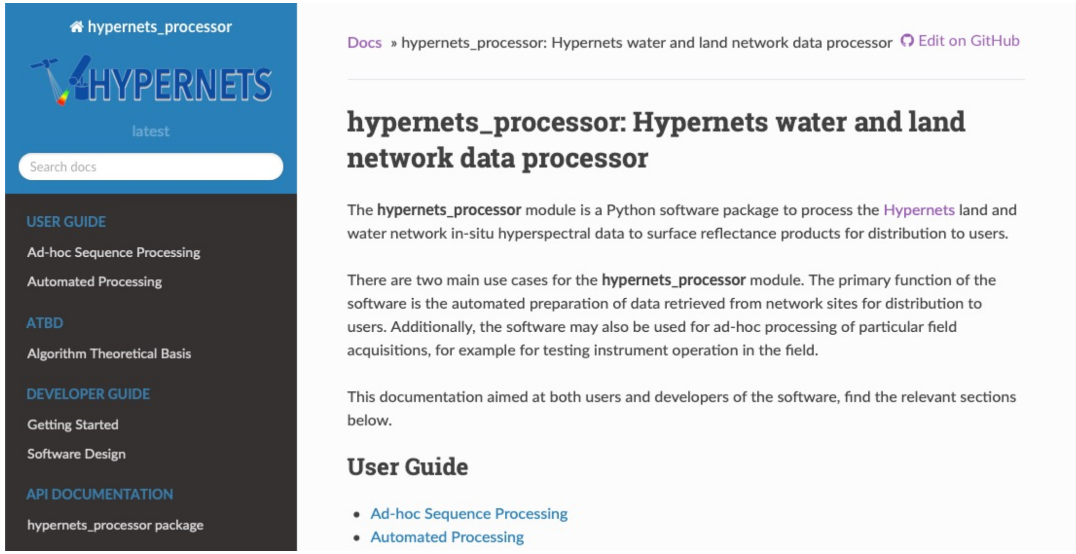
Example: Hypernets Ground Processor

National Physical Laboratory

- Hypernets is an underdevelopment network of ground test sites with automated hyperspectral spectrometers for surface reflectance validation

- Uncertainty information is provided with every product, including error-correlation information.

- Implementation in ground processor is powered by CoMet tools



CoMet: Community tools for Metrology

- An open-source software project to develop Python tools for the handling of error-covariance information in the analysis of measurement data
- Includes **obsarray** and **punpy** as initial offering, to be extended (optimisation next)
- Moving towards initial release on GitHub/PyPI platform

Next Steps

NPL National Physical Laboratory

- Developing tools further – including expanding scope to include more functionality, such as optimisation.

- Development of documentation, examples and dissemination approach

- We are looking for beta testers!