

Hello! 🖐️



Maddie Stedman



Pieter De Vis



Rasma Ormane



Workshop Materials





Today's Agenda



1. Presentation 1

- ☐ *Toolkit Introduction (10 min)*



2. Exercise 1

- ☐ *“Uncertainty Propagation” (15 min)*



3. Presentation 2

- ☐ *“Error Correlation” (10 min)*

4. Exercise 2

- ☐ *“Error Correlation” (20 min)*



5. Wrap-up

- ☐ *Applications & Outlook (5 min)*



The CoMet Toolkit – Uncertainties Made Easy

Pieter De Vis, Maddie Stedman, Sam Hunt, Rasma Ormane
National Physical Laboratory

Wavelength 2026 Hands-on Tutorial - 09/04/2026



NPL



NPL Bushy House



NPL, Teddington

What is National Physical Laboratory?

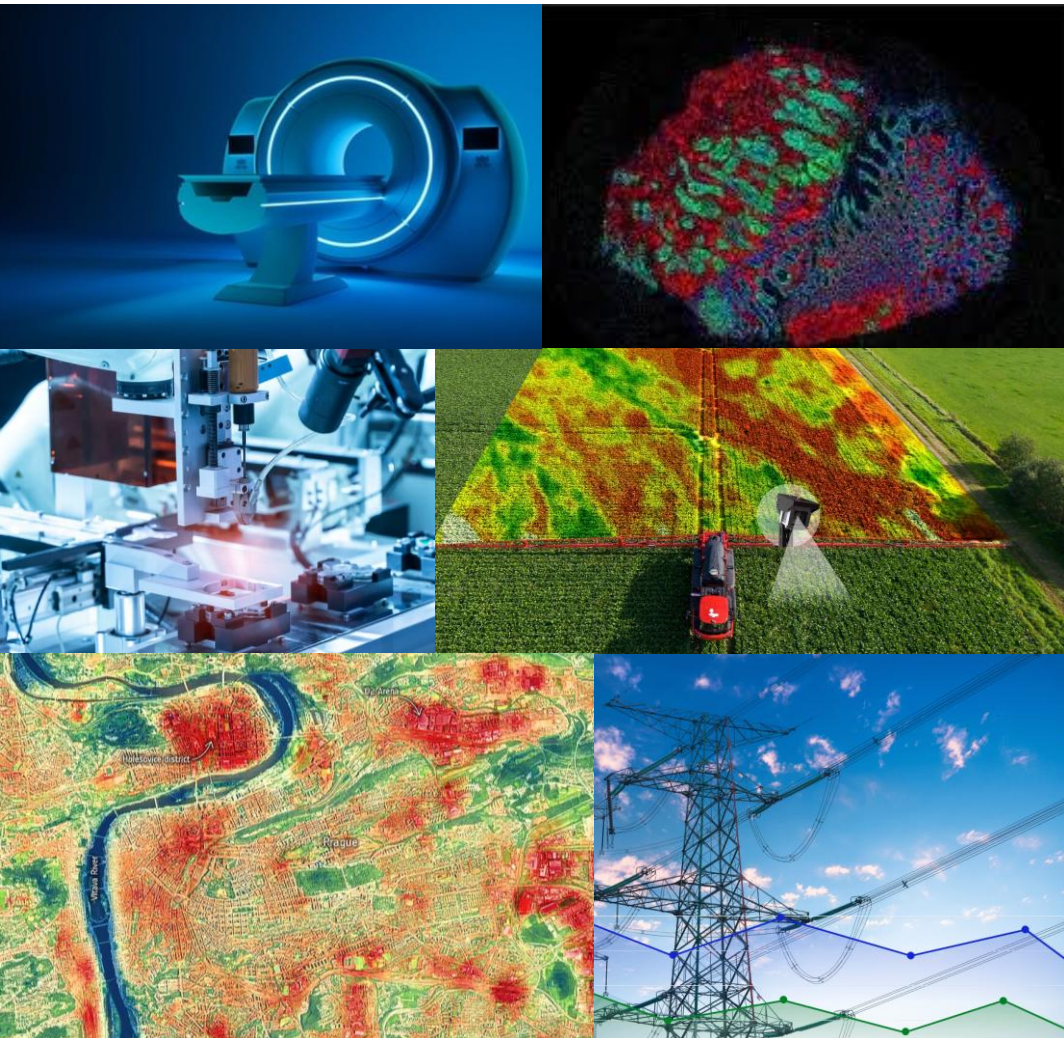
- UK's **National Metrology Institute (NMI)**
- “Realises, maintains and develops the UK's primary measurement standards”
- Established in 1900



ESA-developed Earth Observation Missions



Measurements in Society

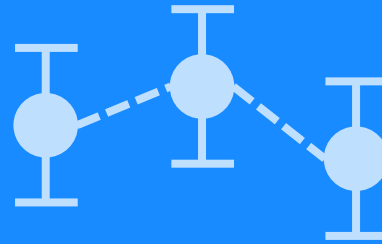


- ❑ Critical for e.g. **health**, **manufacturing**, and **environmental** monitoring.
- ❑ Growing in **size** and **complexity**.
- ❑ Reliable interpretation requires **uncertainty** and **error-covariance** information, often overlooked or non-standardised.
- ❑ **Error correlation** important to get uncertainties right when combining data

CoMet Toolkit



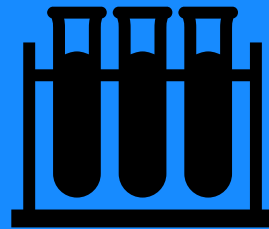
Python
Tools



Uncertainty
Handling



Open
Source



Tested



Applied

CoMet Toolkit



punpy

Propagation UNcertainties in Python

obsarray

Handling uncertainty and error-covariance in datasets

comet_maths

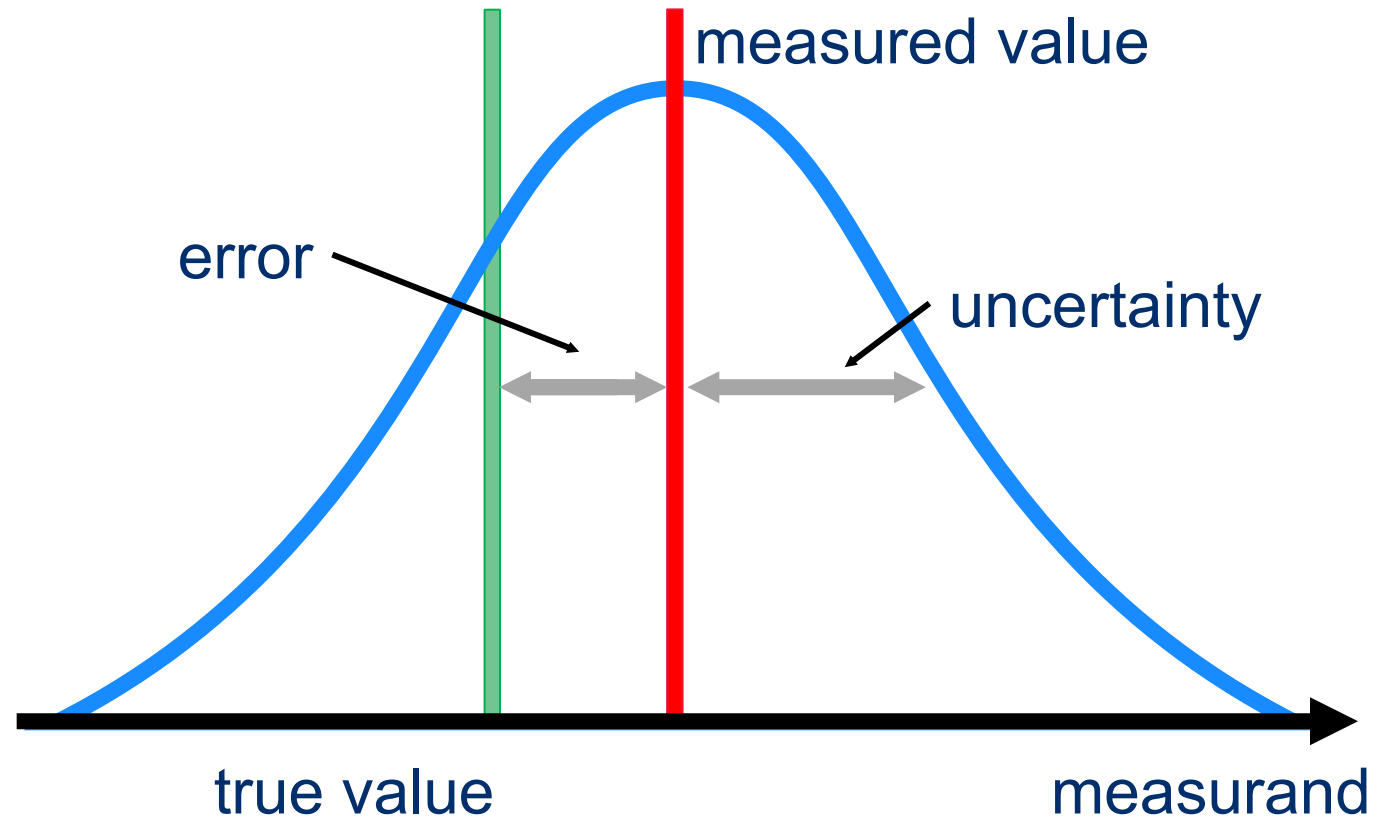
CoMet mathematical algorithms and interpolation tools

UNC Specification

Uncertainty metadata naming conventions



Uncertainties 101

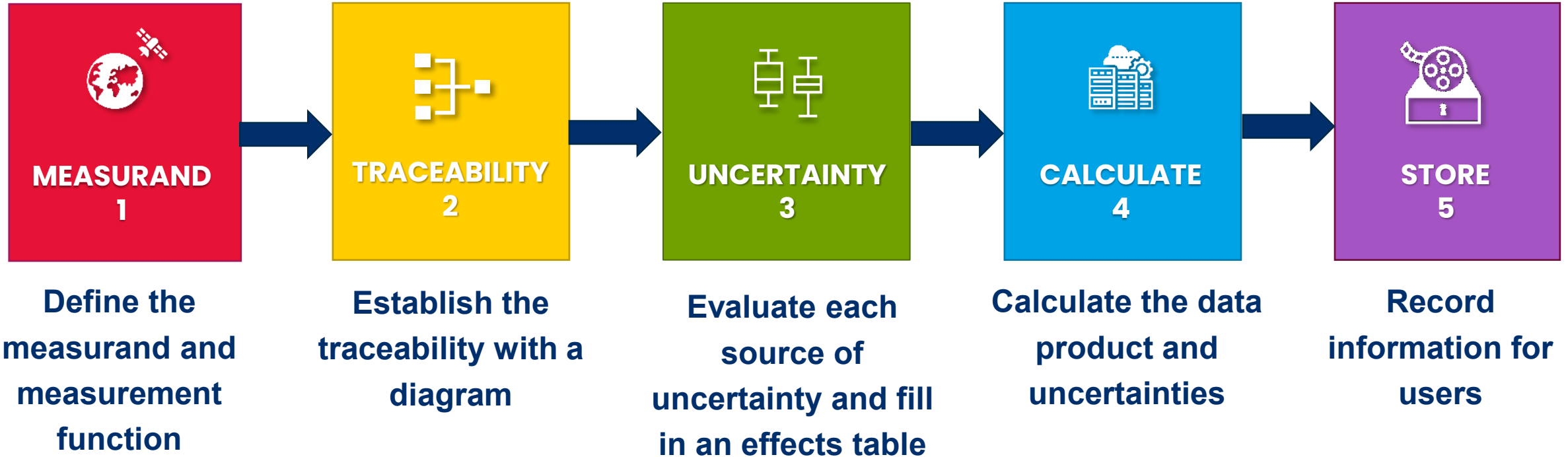




A Metrological Approach

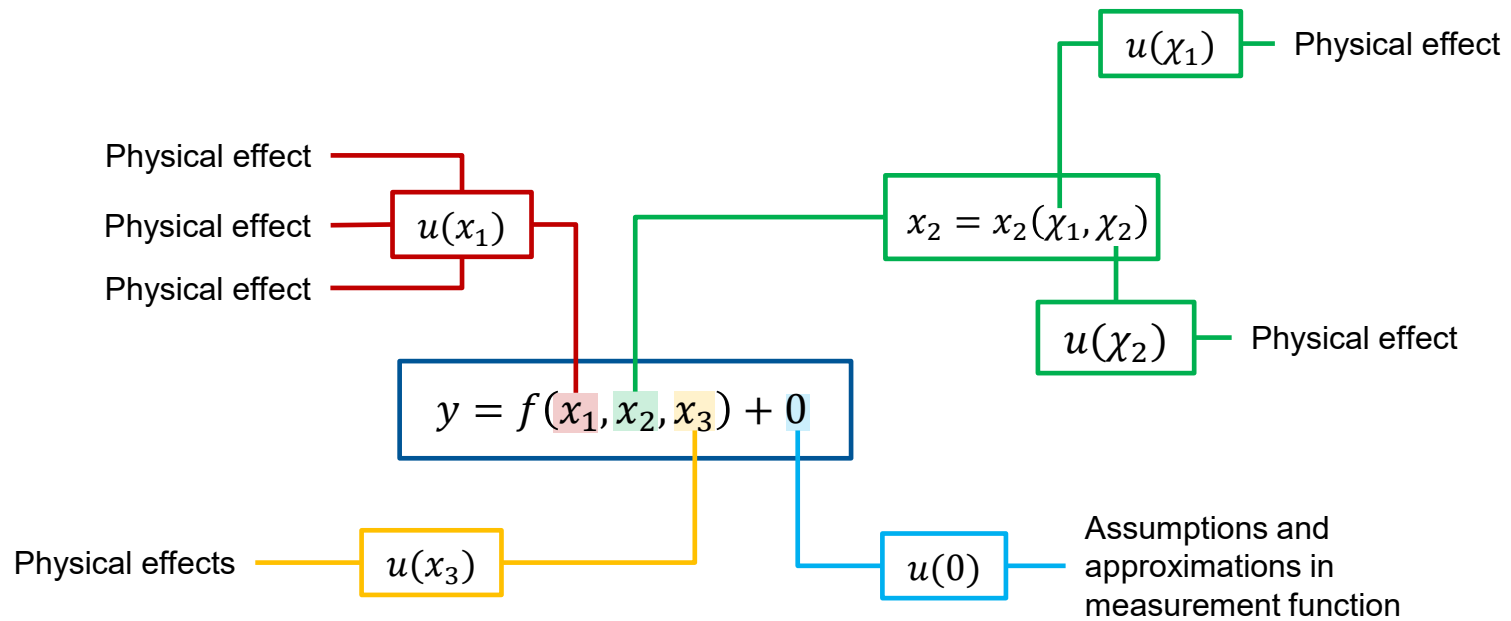
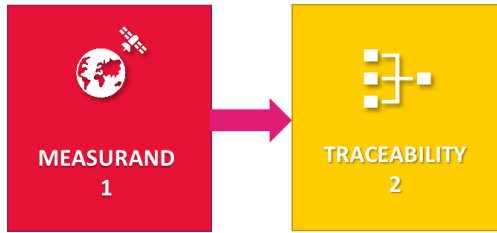


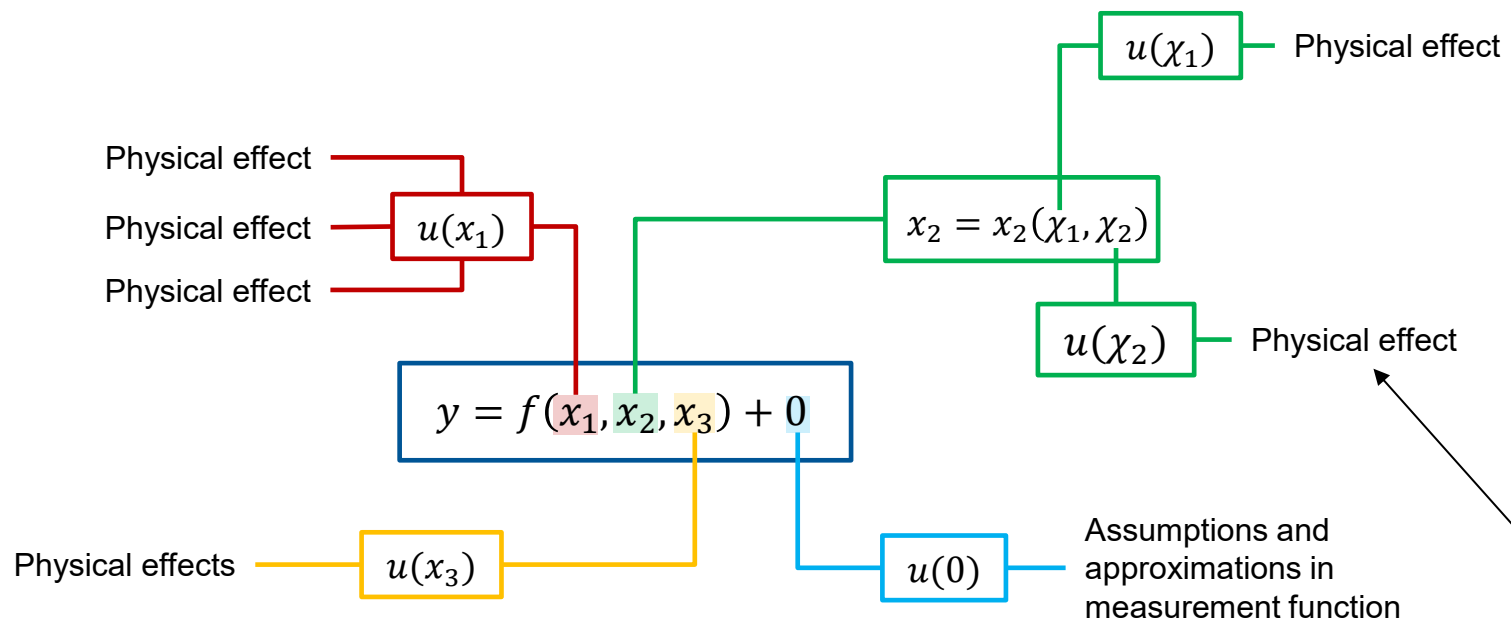
Uncertainties are evaluated and expressed following [QA4EO Five Steps](#), a framework which employs the principles of metrology.



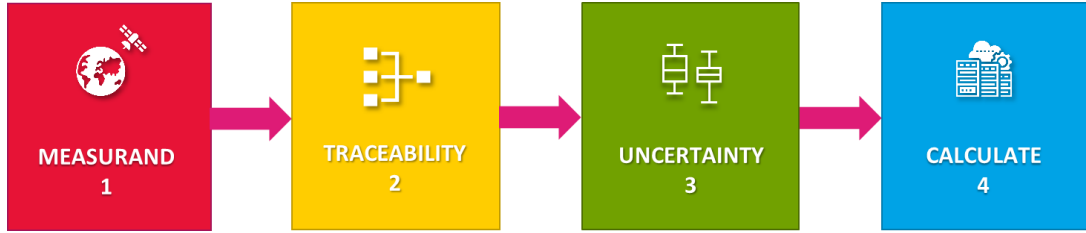


$$y = f(x_1, x_2, x_3) + 0$$

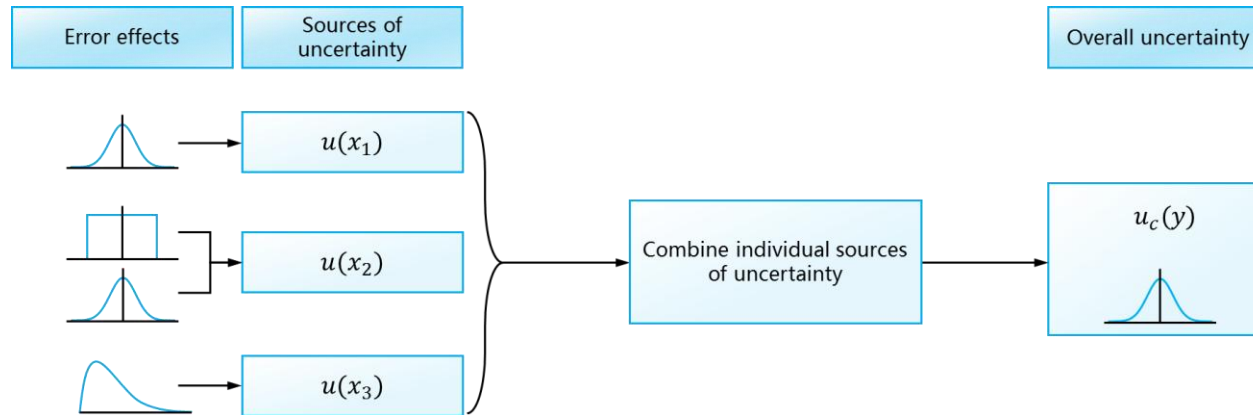


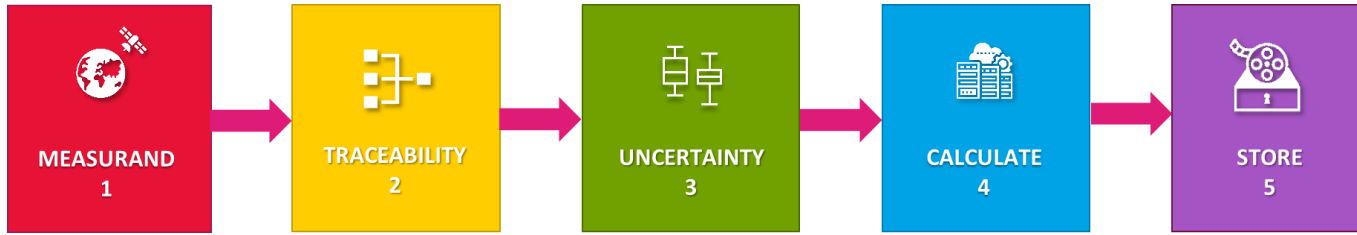


		Comments
Name of effect		A unique name
Affected term in measurement function		Name and standard symbol
Instruments in the series affected		List names
Correlation type and form	Pixel-to-pixel [pixels] from scanline to scanline [scanlines] between images [images] Between orbits [orbit] Over time [time]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



puppy





obsarray

		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 <i>measurand</i> to term



```

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";
  
```

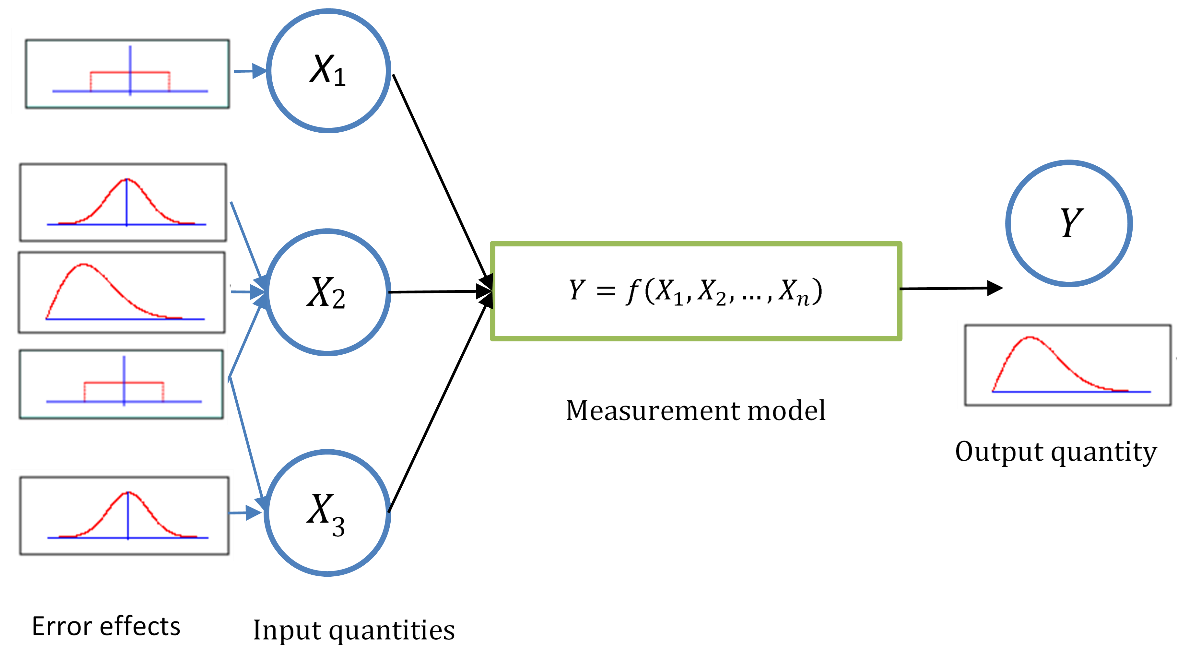
CoMet Packages:



puppy

Propagating Uncertainties with

- ❑ Python module for propagating **random, systematic** and **structured uncertainties** through any Python measurement function
- ❑ Flexible in terms of the **specified correlations** along given dimensions or between input quantities
- ❑ **Monte Carlo** and **Law of Propagation of uncertainties** methods available



Punpy as a Standalone Tool



□ Simple user **interface**:

1. Import punpy
2. Define measurement function
3. Create MC or LPU object
4. Propagate uncertainties

```
import punpy
prop=punpy.MCPropagation(10000)
unc_measurand=prop.propagate_random(measurement_func,
                                     [input_qt1,input_qt2],[unc_qt1,unc_qt2])
```

□ **Measurement function** are defined as python functions that take arrays as input quantities and return an array as measurand

□ Many optional **keywords** for flexible functionality

- *return_corr*
- *Corr_between*
- *Repeat_dims*
- *Parallel_cores*
- *Output_vars*
- ...

Punpy with digital effects tables

- ❑ **punpy** interfaces with **obsarray** to make uncertainty propagation as efficient and easy to use as possible
- ❑ **propagate_ds()** function returns an **obsarray** dataset with combined random, systematic and structured uncertainties on measurand

```
from punpy import MeasurementFunction

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

# create object of the measurement function class and specify the variable names
gl = IdealGasLaw(["pressure", "temperature", "n_moles"], "volume", yunit="m^3")

# propagate uncertainties on the input quantities in ds to measurand in ds_y
ds_y = gl.propagate_ds(ds)
```



Exercise 1



□ Please go to <https://www.comet-toolkit.org/user-guide/training/wavelength/>





The CoMet Toolkit Error Correlation

Maddie Stedman, Pieter De Vis, Sam Hunt, Rasma Ormane
National Physical Laboratory

Wavelength 2026 Hands-on Tutorial - 09/04/2026



Error Correlation

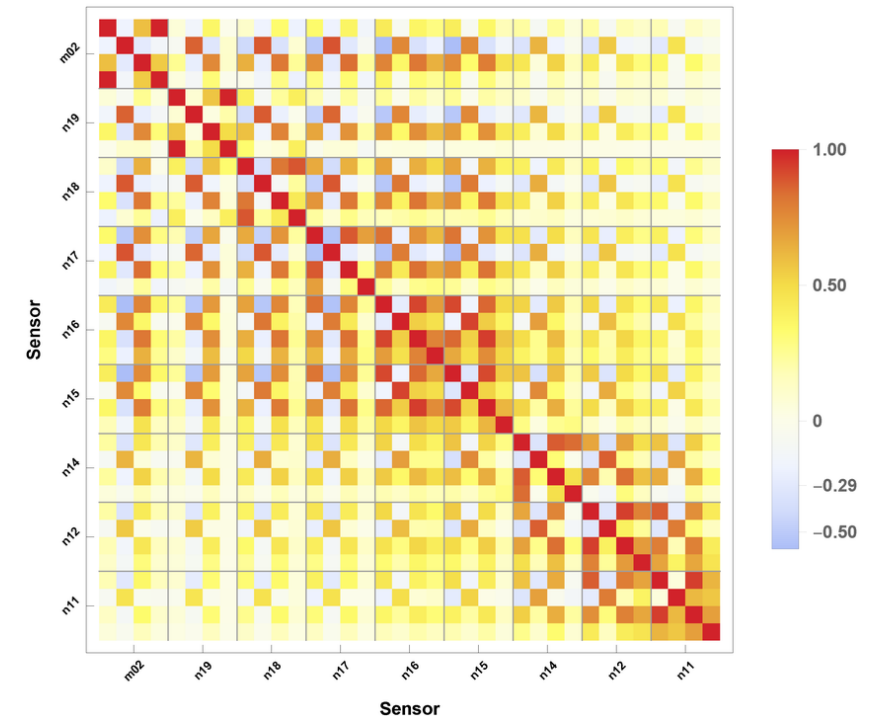


What is Error Correlation?

Errors in a dataset are **not always independent** — e.g., when errors in pixels, bands instruments, or time steps are systematically related.

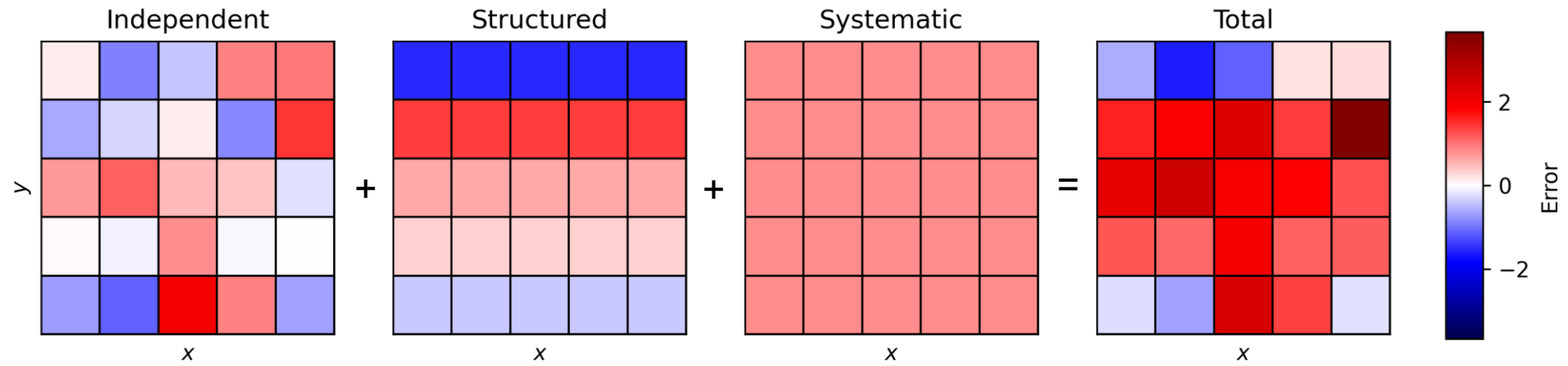
Why does it Matter?

- Bias uncertainties persist – averaging doesn't help!
- Band ratios might be off – affecting retrieval uncertainty
- Misleading confidence in trends

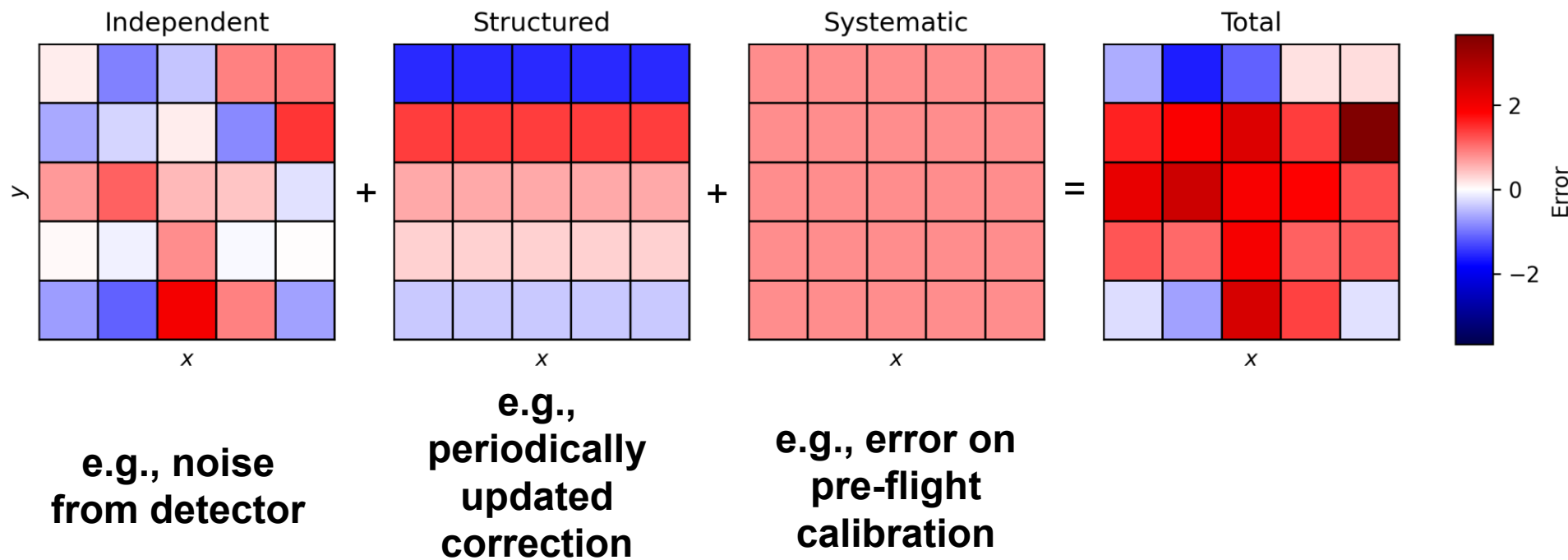


Error correlation matrix from Giering et al. 2019

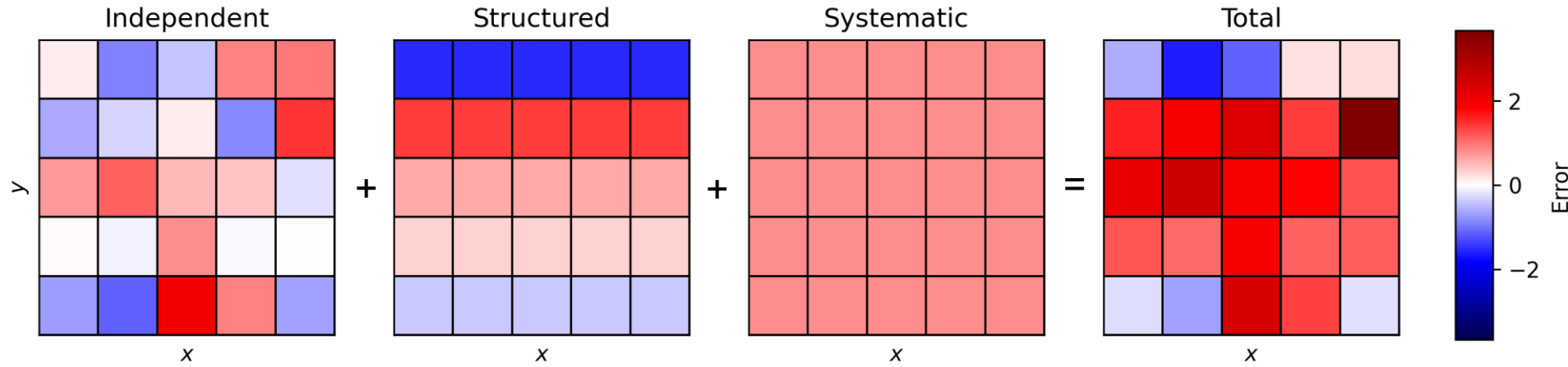
ND Datasets & Error-Correlation



ND Datasets & Error-Correlation



ND Datasets & Error-Correlation



Each of these effects has the same uncertainty ($u=1$) – but a very different behaviour in the overall dataset! This matters!

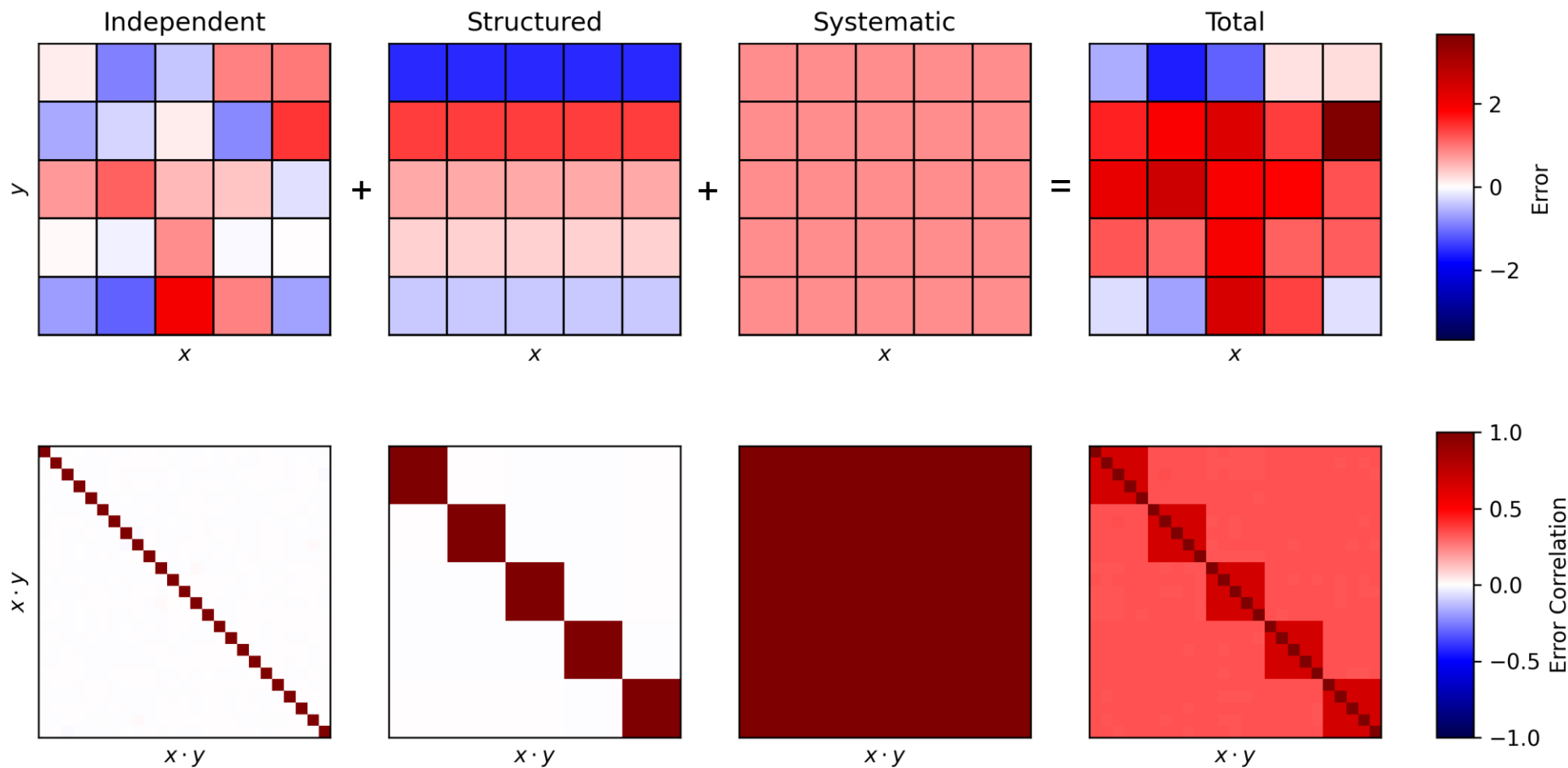
ND Datasets & Error-Correlation

Define this with an error-correlation matrix for the image, defining the cross-pixel error-correlation, $r_{i,j}$, for all pixels – label as:

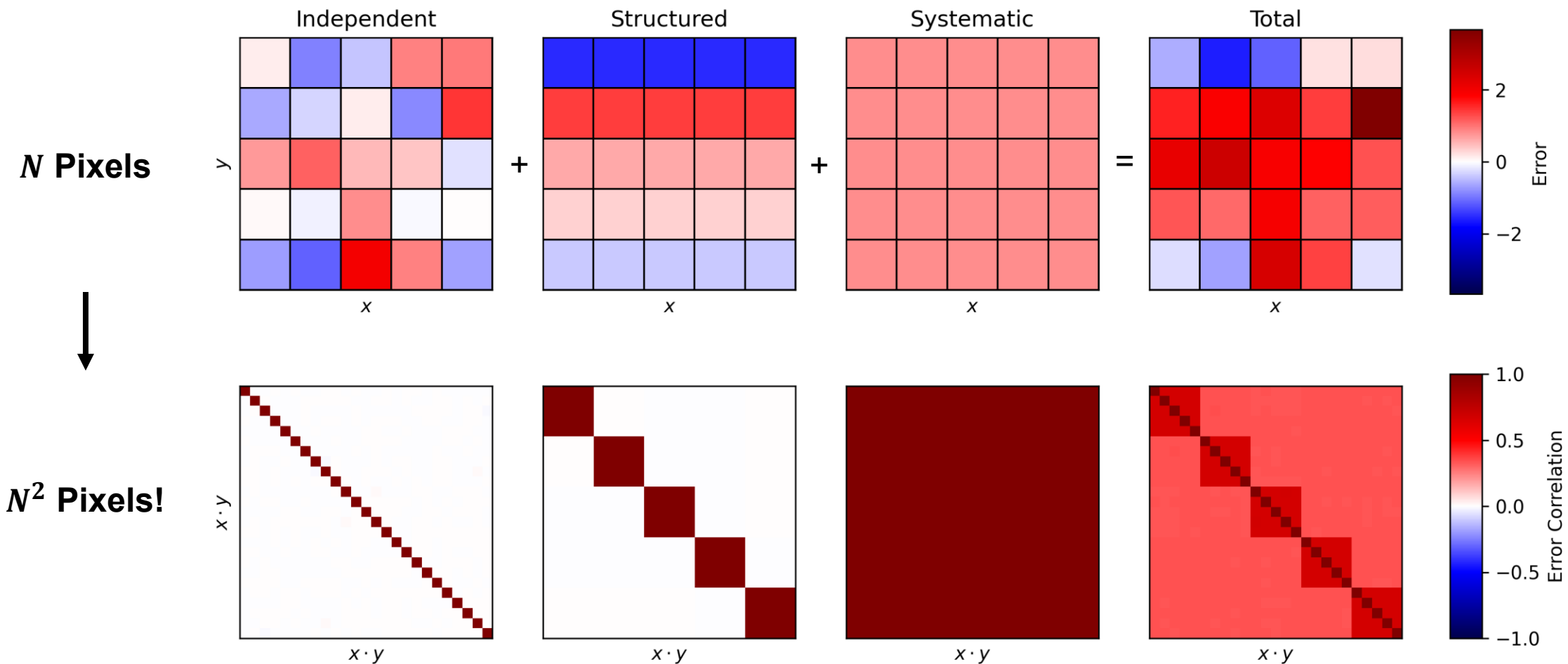
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

$$\begin{matrix} \text{---} & \text{---} \\ | & | \\ \left[\begin{array}{cccccc} r_{1,1} & r_{2,1} & r_{3,1} & r_{4,1} & r_{5,1} & r_{6,1} & \dots \\ r_{1,2} & r_{2,2} & r_{3,2} & & & & \\ r_{1,3} & r_{2,3} & r_{3,3} & & & & \\ r_{1,4} & & & \ddots & & & \\ r_{1,5} & & & & & & \\ r_{1,6} & & & & & & \\ r_{1,7} & & & & & & \\ r_{1,8} & & & & & & \\ \vdots & & & & & & \end{array} \right] \end{matrix}$$

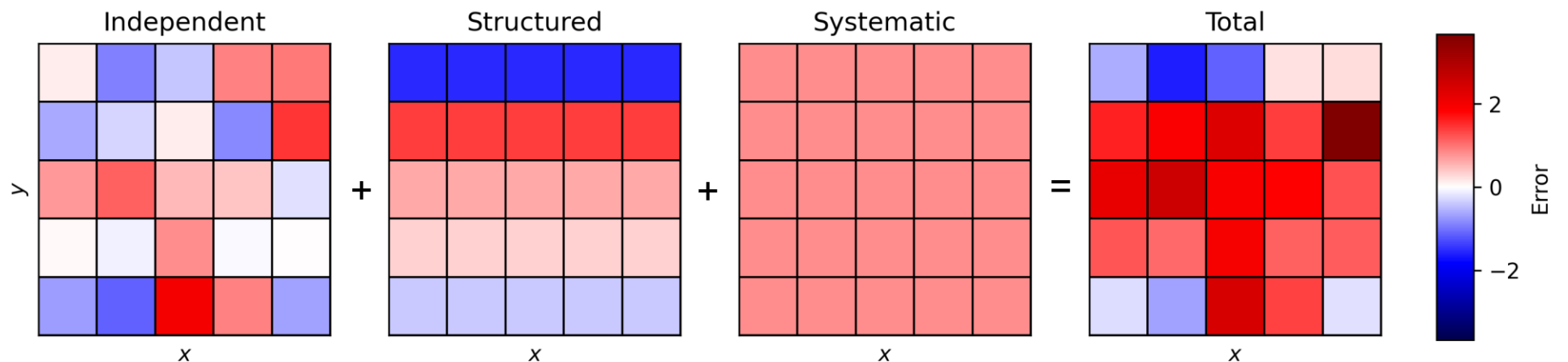
ND Datasets & Error-Correlation



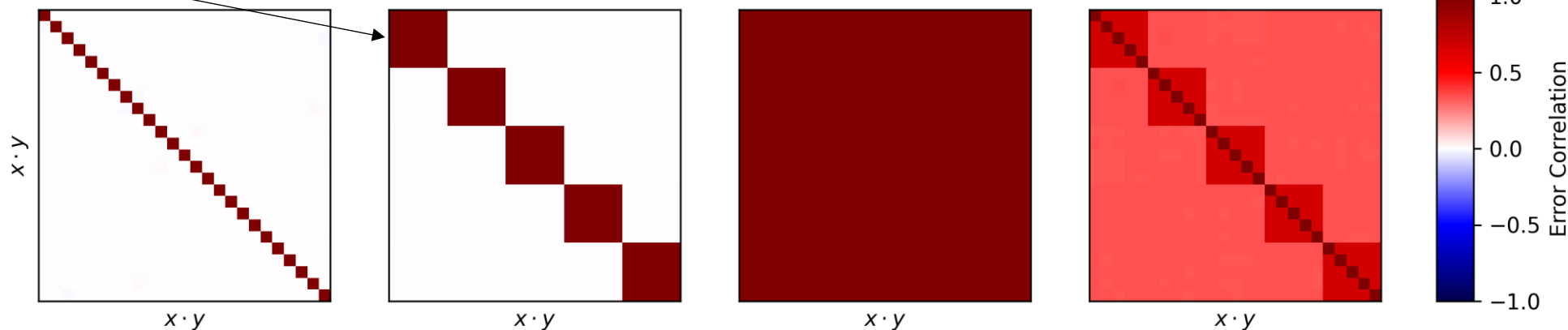
ND Datasets & Error-Correlation



ND Datasets & Error-Correlation



Sub-matrices for systematic error-correlation within row



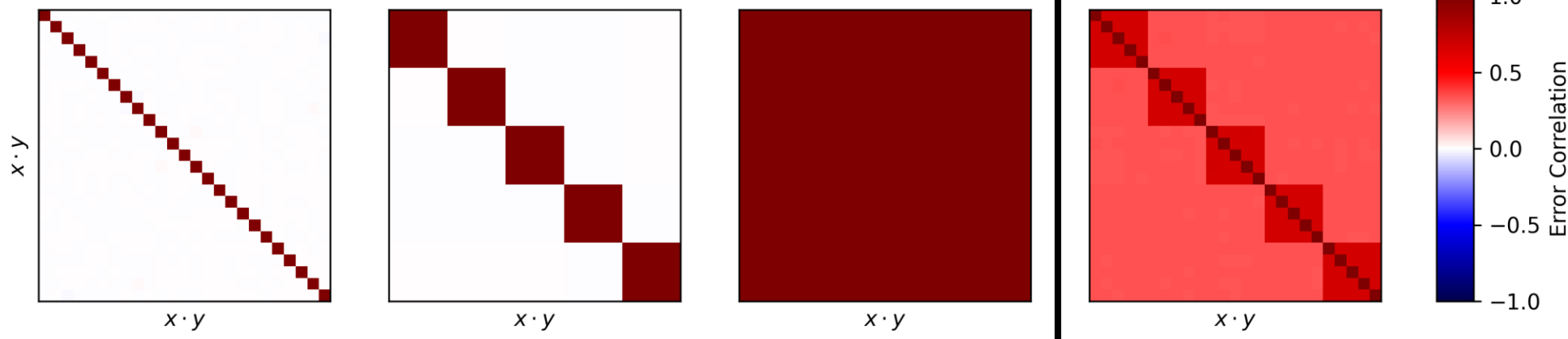
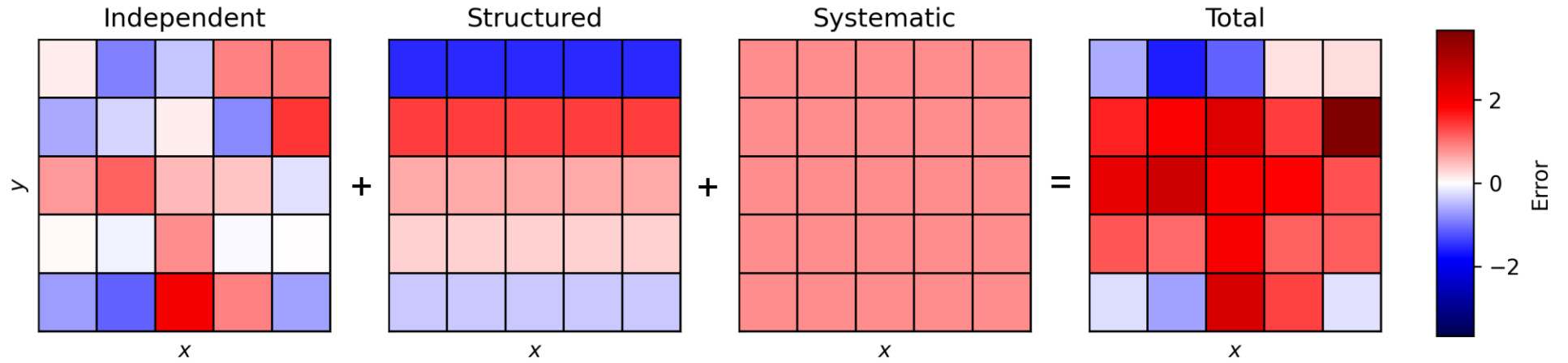
Identity

Block Diagonal

Full

Combination...

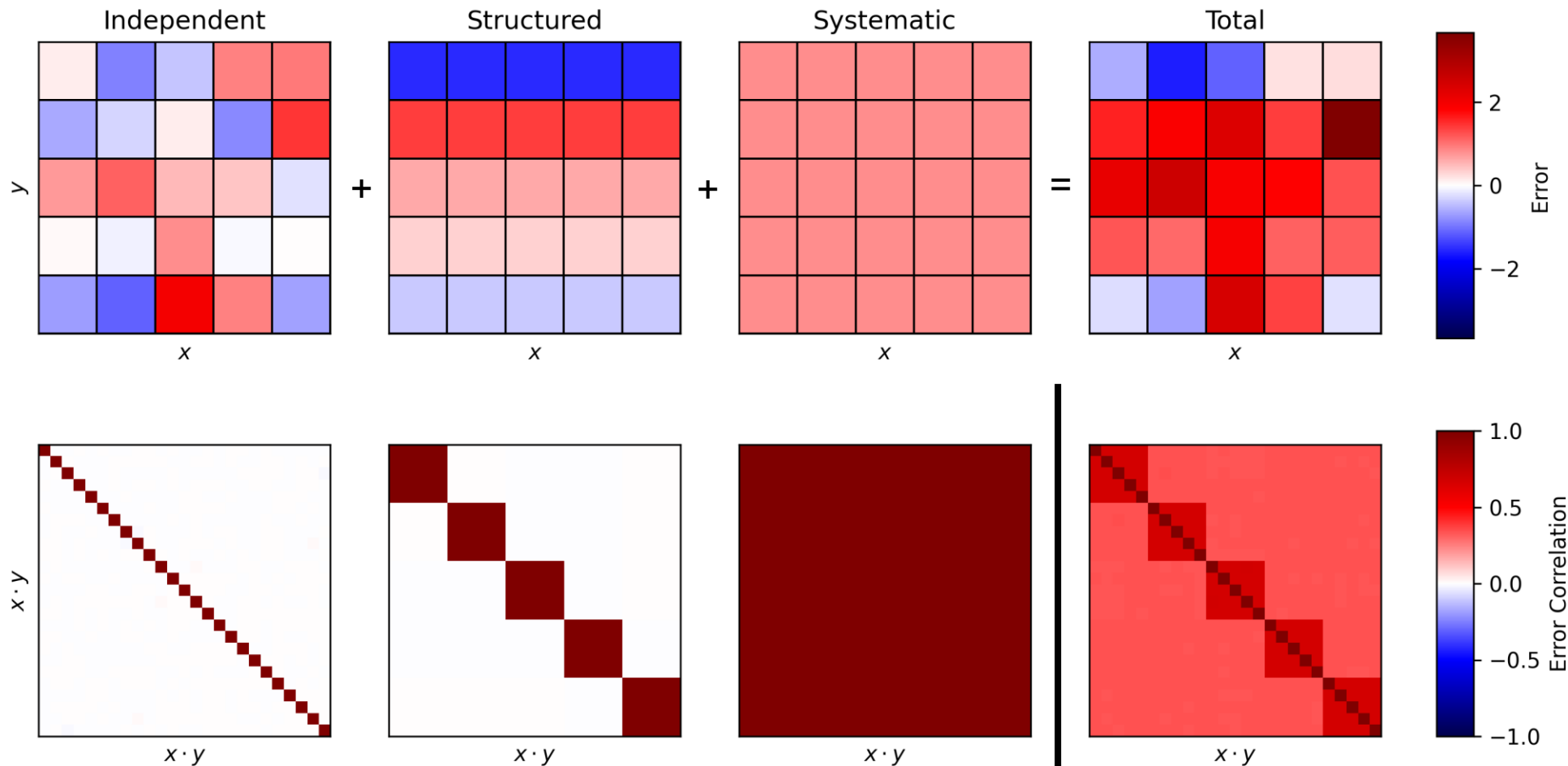
ND Datasets & Error-Correlation



- Simply parameterisable

- Parameterisable via other terms

ND Datasets & Error-Correlation



- Simply parameterisable
- One term might dominate at different scales

- Parameterisable via other terms
- Same complexity at all scales



Error Correlation



What is Error Covariance?

Combines error correlation and uncertainty

$$S = U R U^T$$

Random, systematic and structured uncertainty

- It is the errors that are correlated, not the uncertainty values
- Random means completely uncorrelated, i.e. error correlation is identity matrix
- Systematic means fully correlated, i.e. error correlation filled with 1's

 COMet Packages:



obsarray



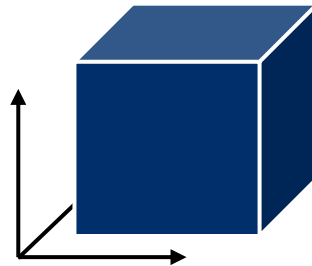
UNC Specification



Uncertainty Variable Metadata

Observation Variables

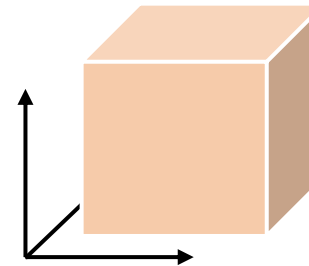
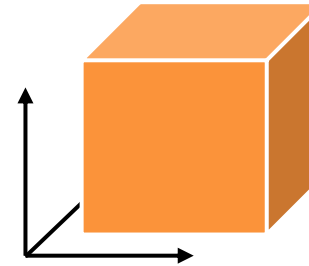
Uncertainty Variables



Metadata:

- Uncertainty Components

associated with



...

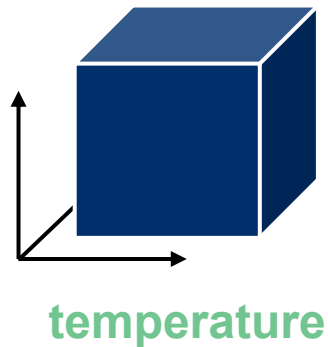
Metadata:

- PDF Shape (gaussian, ...)
- Units (abs. or rel.)
- Error-Correlation...

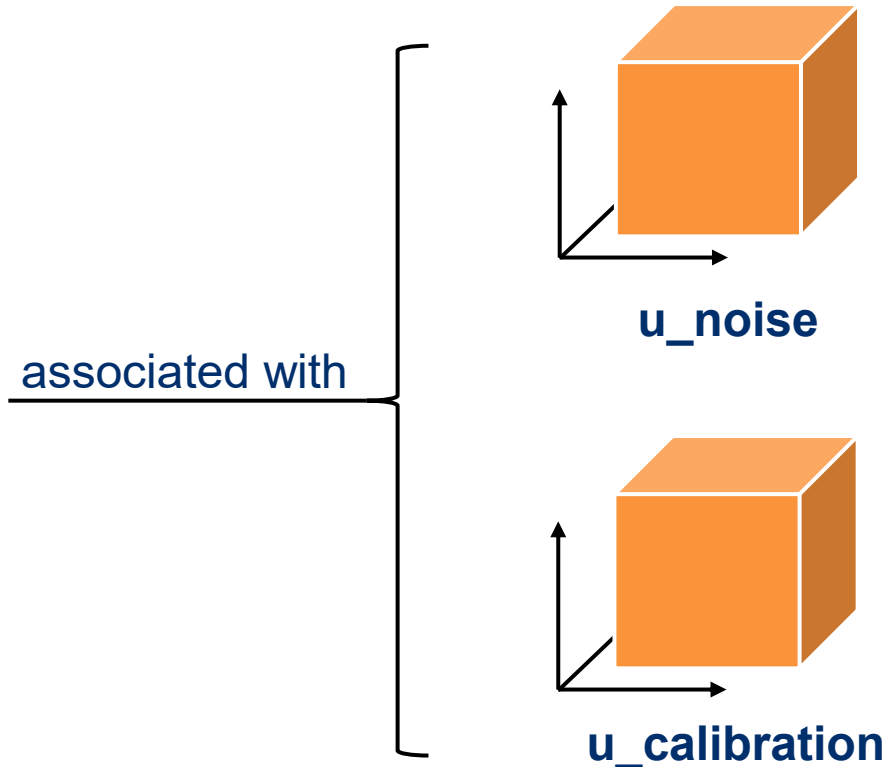
UNC Specification

Example – Temperature Dataset

Observation Variables



Uncertainty Variables



Metadata:

- PDF Shape – “gaussian”
- Units – %
- Error-Correlation:
 - All dims - Random

Metadata:

- PDF Shape – “rectangular”
- Units – “K”
- Error-Correlation:
 - **x, y** – systematic
 - **time** – defined by matrix



Example – Temperature Dataset

```
variables:
  float temperature(time, lat, lon);
    temperature:unc_comps=["u_calibration", "u_noise"];
    temperature:units="K"
  float u_calibration(time, lat, lon);
    u_calibration:units="K";
    u_calibration:pdf_shape="rectangular";
    u_calibration:err_corr_dim1_name=["lat", "lon"];
    u_calibration:err_corr_dim1_form="systematic";
    u_calibration:err_corr_dim2_name="time";
    u_calibration:err_corr_dim2_form="err_corr_matrix";
    u_calibration:err_corr_dim2_params=["err_corr_calibration_time"];
  float u_noise(time, lat, lon);
    u_calibration:err_corr_dim1_name=["time", "lat", "lon"];
    u_calibration:err_corr_dim1_form="random";
  float err_corr_calibration_time(time, time);
```



Measurement data handling in Python

- ❑ **obsarray** is an extension to xarray to support defining, storing and interfacing with measurement data – using the UNC specification.
- ❑ Also has functionality for defining flags following **CF Conventions**.
- ❑ It is designed to work well with netCDF files and for the **Earth Observation** community.

Plugs straight into punpy for propagation through measurement functions!



Exercise 2



□ Please go to <https://www.comet-toolkit.org/user-guide/training/wavelength/>





The CoMet Toolkit Wrap-up

Pieter De Vis, Maddie Stedman, Sam Hunt, Rasma Ormane
National Physical Laboratory

Wavelength 2026 Hands-on Tutorial - 09/04/2026

Comet Application Examples

 **punpy**

 **obsarray**

 **comet_math**

CoMet Toolkit in Action



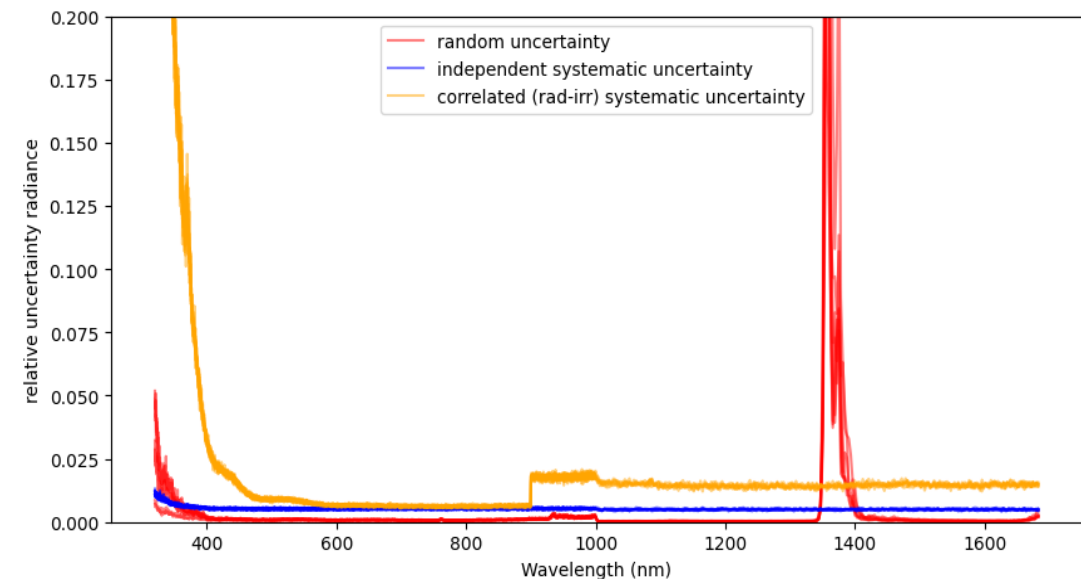
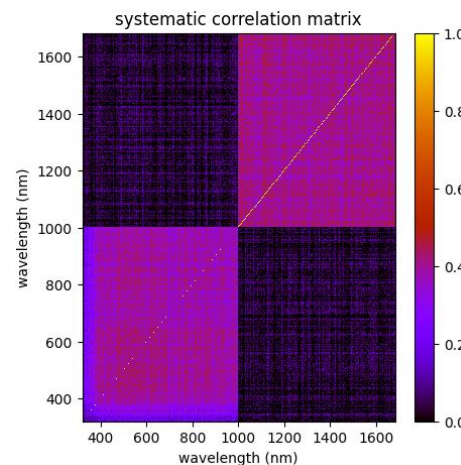
- ❑ Validated against **NIST** uncertainty engine:

https://colab.research.google.com/github/comet-toolkit/comet_training/blob/main/NIST_example.ipynb

- ❑ *CoMet* is used in various other projects, such as **QA4EO**, **HYPERNETS**, **CHIME L2**, **FLEX** validation, **TRUTHS** science studies, **LIME**, **FRM4SOC**, **RPV4PICS**

- ❑ **Example** from *hypernets_processor*:

Hypernets is an automated network of in-situ instruments measuring reflectance for L2 satellite validation





CoMet Release



☐ V1.0 of Comet toolkit has been released as **open source** toolkit:

- www.comet-toolkit.org
- github.com/comet-toolkit



☐ Accompanied by training material (**Jupyter** notebooks hosted on google colab):

- www.comet-toolkit.org/user-guide/examples

☐ Documentation & ATBD for individual tools:


- obsarray.readthedocs.io/en/latest/
- punpy.readthedocs.io/en/latest/
- comet-maths.readthedocs.io/en/latest/



Outlook



- ❑ **Current release** will be presented De Vis & Hunt (in prep)

- ❑ Looking to continue to expand the use cases the developed tools
 - Aiming to enable uncertainty propagation through **any python measurement function**
 -  Please get in touch if you are interested!

- ❑ This has been our first step into this way of working, **many more ideas in a roadmap** to building up a comprehensive set of tools
 - e.g. retrieval tool/optimisation, BRDF tool, Look-up tables for faster processing, etc.

Summary



- ❑ The **CoMet toolkit** is an open-source software project to develop Python tools for the handling of error-covariance information in the analysis of measurement data
- ❑ This toolkit is based on **robust metrology**, and makes dealing with complexities of uncertainties much easier
- ❑ Includes **obsarray**, **punpy** & **comet_maths** as initial offering, to be extended
- ❑ These tools are already being used operationally in various projects (e.g. Hypernets)