Welcome! 👋

Mentimeter Survey

Code: 2412 1208

https://www.menti.com/altf8h43rc5j

Today's Exercises

https://www.comet-toolkit.org/user-guide/training/aria/

IDEAS-QA4EO

NPL National Physical Laboratory

Hello! 👋

Sam Hunt

Pieter De Vis

Maddie Stedman

Enis Gerxhalija

Workshop Materials

https://www.comet-toolkit.org/user-guide/training/aria/

IDEAS-QA4EO

NPL National Physical Laboratory

📋 Today's Agenda

1. **Presentation 1:** *Uncertainties & CoMet Toolkit Introduction (20 min)*
2. **Exercise 1:** *"Punpy Uncertainty Propagation" (15 min)*
3. **Presentation 2:** *"Error Correlation" (20 min)*
4. **Exercise 2:** "Punpy with *Error Correlation" (20 min)*

Break (30 min)

5. **Exercise 3:** *"Spectrometer example" (30 min)*
6. **Presentation 3:** *"UNC & obsarray" (20 min)*
7. **Exercise 4:** *"Multidimensional datasets" (30 min)*
8. **Wrap-up:** *"Applications & Outlook" (10 min)*

IDEAS-QA4EO

NPL National Physical Laboratory

The CoMet Toolkit – Uncertainties Made Easy

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

ARIA workshop - 15/04/2026

NPL

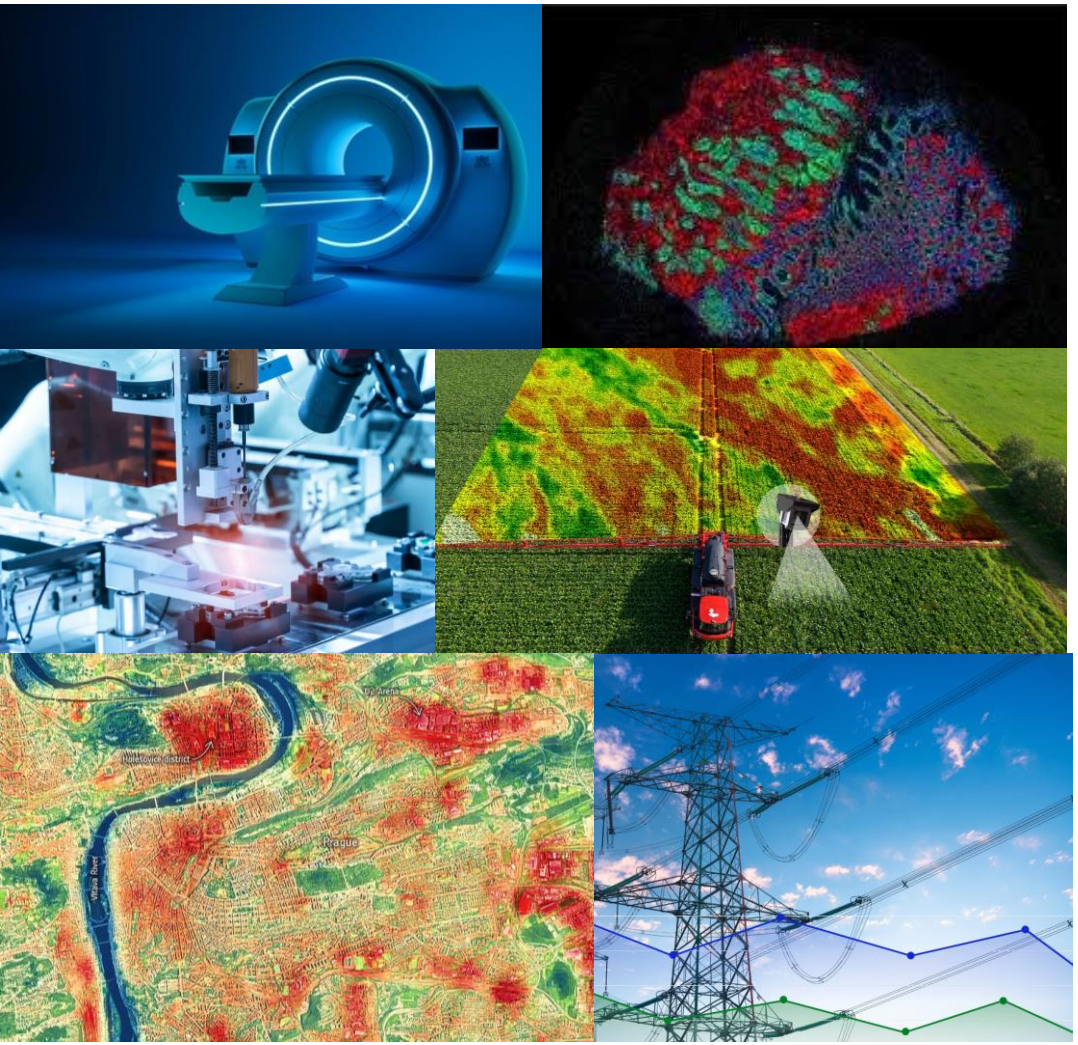
What is National Physical Laboratory?

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

IDEAS-QA4EO

NPL National Physical Laboratory

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

HYPERNETS

truths

IDEAS-QA4EO

NPL National Physical Laboratory

CoMet Toolkit

punpy

Propagation UNcertainties in Python

obsarray

Handling uncertainty and error-covariance in datasets

comet_maths

CoMet mathematical algorithms and interpolation tools

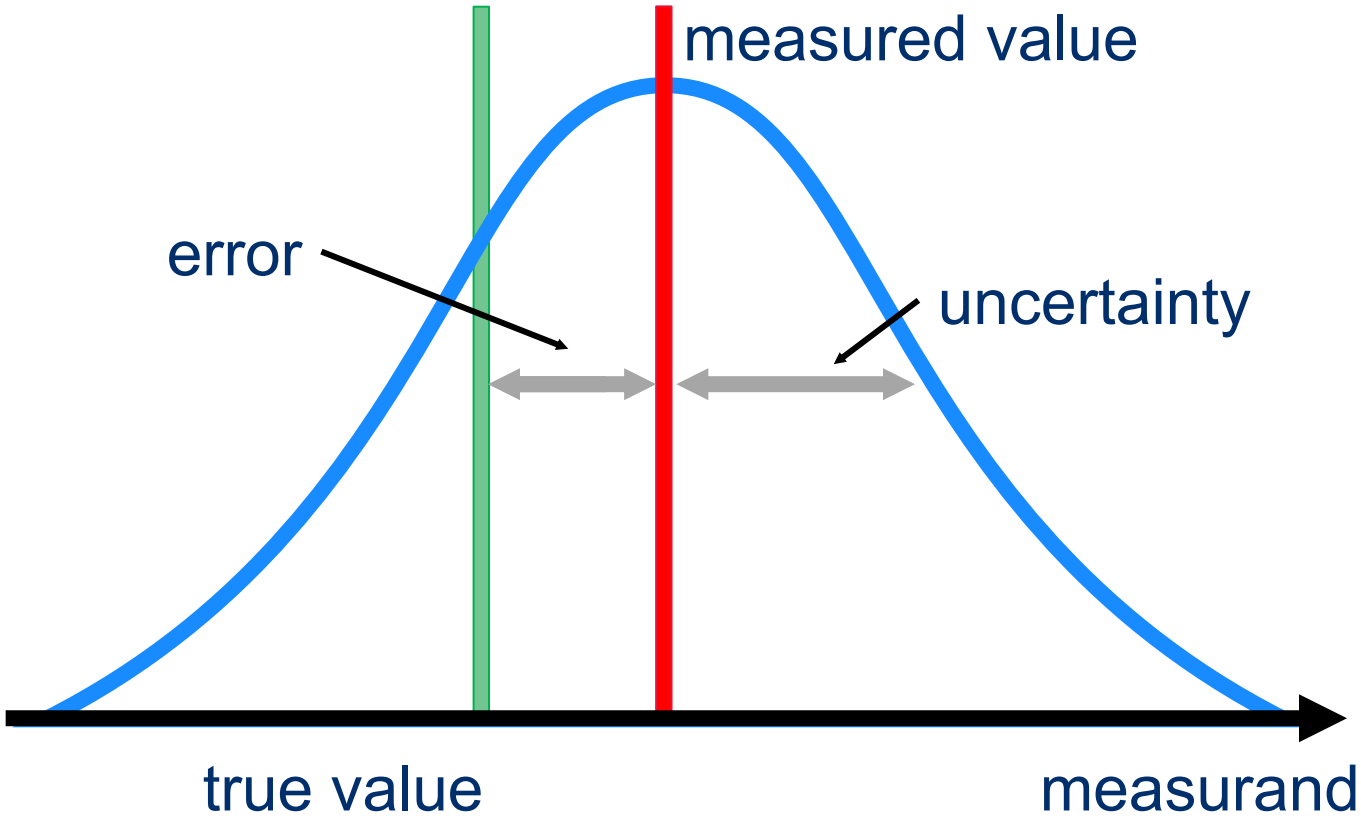
UNC Specification

Uncertainty metadata naming conventions

IDEAS-QA4EO

NPL National Physical Laboratory

Uncertainties 101



A Metrological Approach

Uncertainties are evaluated and expressed following ***QA4EO*** *Five Steps*, a framework which employs the principles of metrology.

Define the measurand and measurement function

Establish the traceability with a diagram

Evaluate each source of uncertainty and fill in an effects table

Calculate the data product and uncertainties

Record information for users

QA4EO

IDEAS-QA4EO

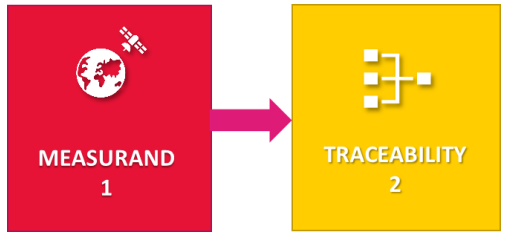
Fiduceo

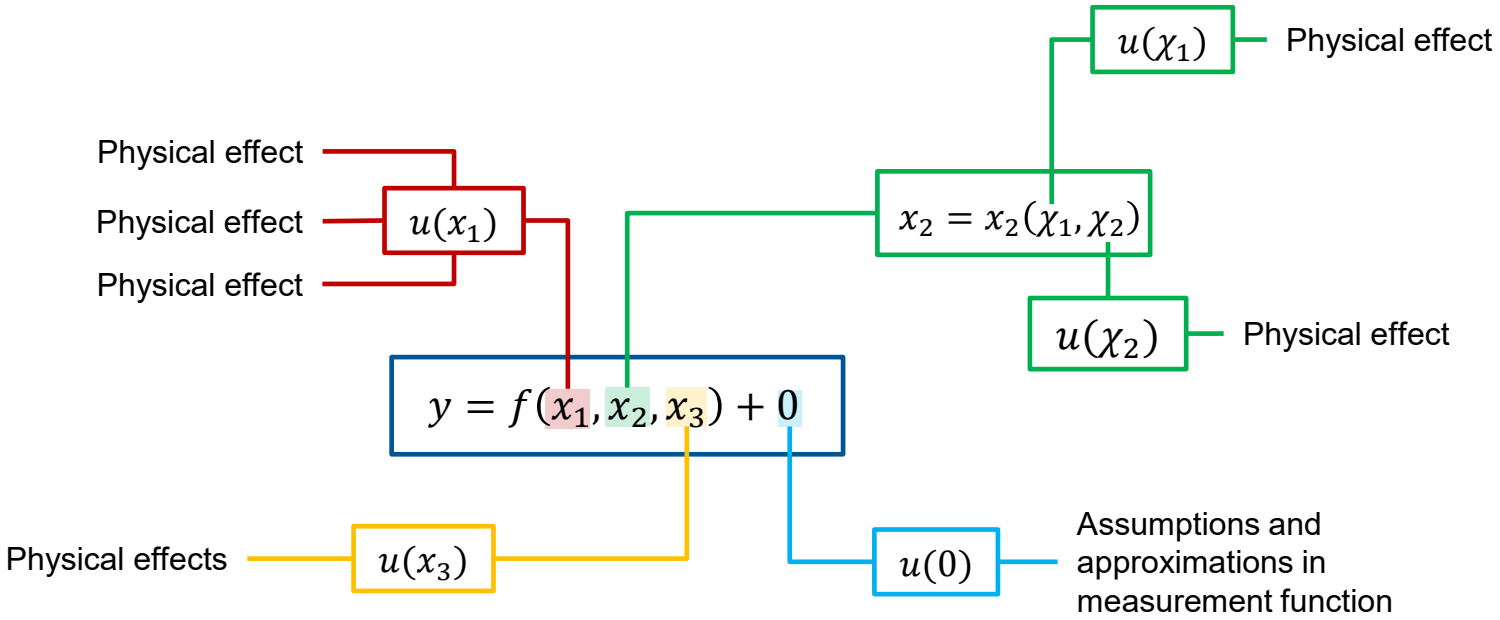
MEASURAND
1

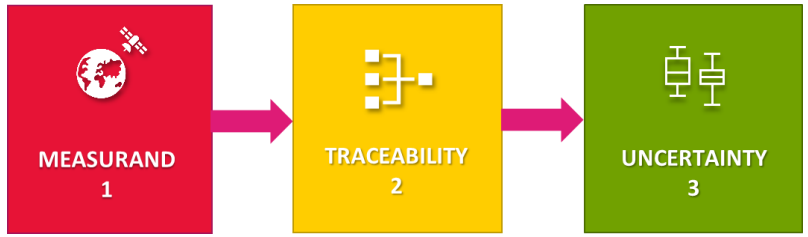
$$y = f(x_1, x_2, x_3) + 0$$

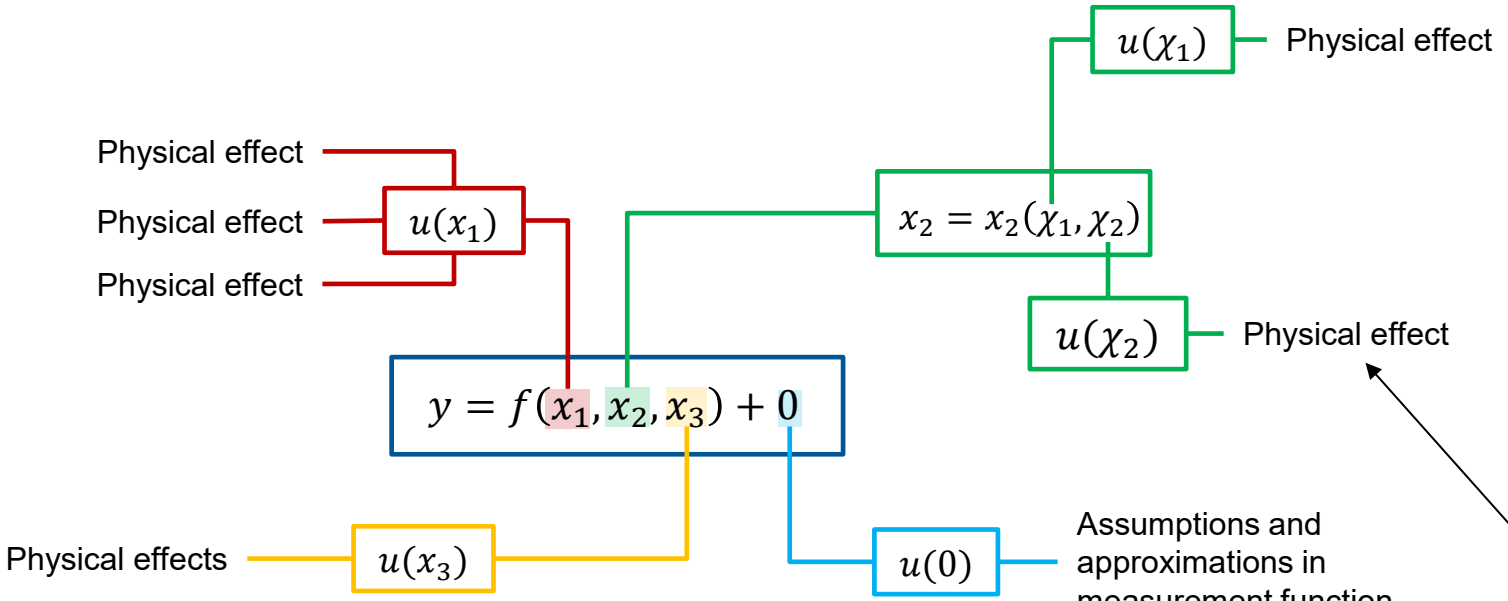
IDEAS-QA4EO

NPL
National Physical Laboratory

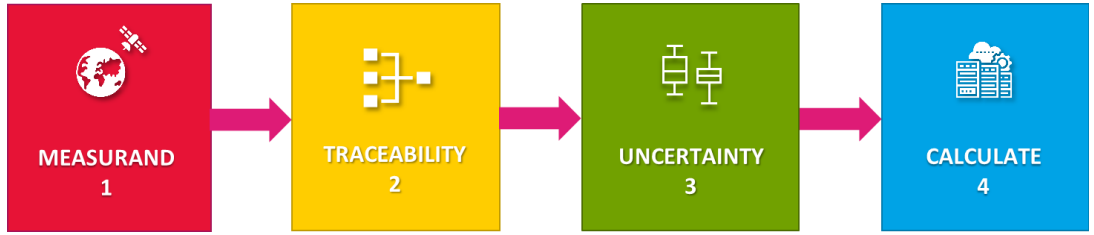

MEASURAND
1
TRACEABILITY
2


$u(\chi_1)$
Physical effect
Physical effect
Physical effect
Physical effect
$u(x_1)$
$x_2 = x_2(\chi_1, \chi_2)$
$u(\chi_2)$
Physical effect
$y = f(x_1, x_2, x_3) + 0$
Physical effects
$u(x_3)$
$u(0)$
Assumptions and approximations in measurement function

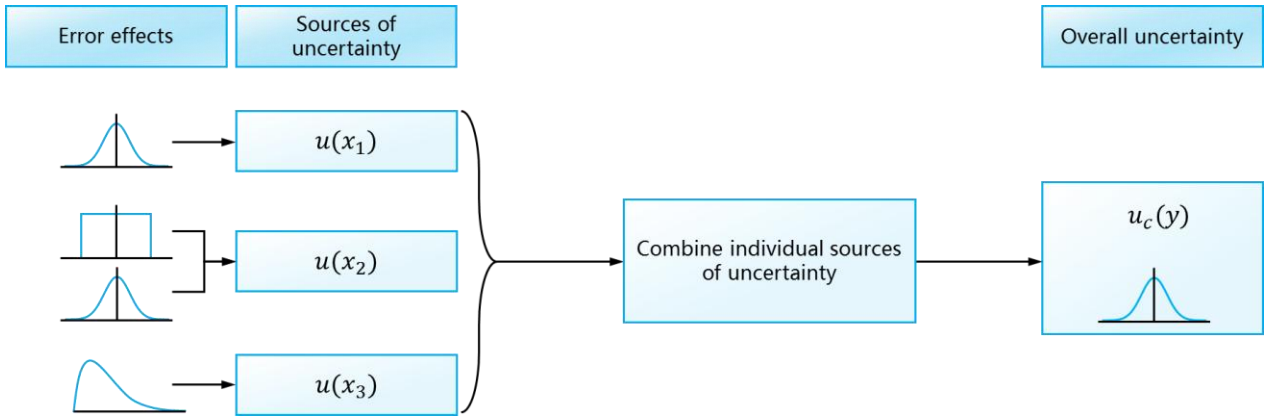




		Comments
Name of effect		A unique name
Affected term in measurement function		Name and standard symbol
Instruments in the series affected		List names
Correlation type and form	Pixel-to-pixel [pixels]	From a set of defined correlation forms
	from scanline to scanline [scanlines]	
	between images [images]	
	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



punpy



- Monte Carlo (MC)
  - ❑ Draw random input samples
  - ❑ Correlate samples
  - ❑ Run samples through measurement function
  - ❑ Calculate statistics

- Law of Propagation of Uncertainty (LPU)
  - ❑ $u^2(y) = \sum_{i=1}^{N} c_i^2 u^2(x_i) + 2 \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_i c_j u(x_i, x_j).$
  - ❑ $S_y = J^T S_x J$

IDEAS-QA4EO

NPL National Physical Laboratory

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 measurand 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";
```

IDEAS-QA4EO

NPL National Physical Laboratory

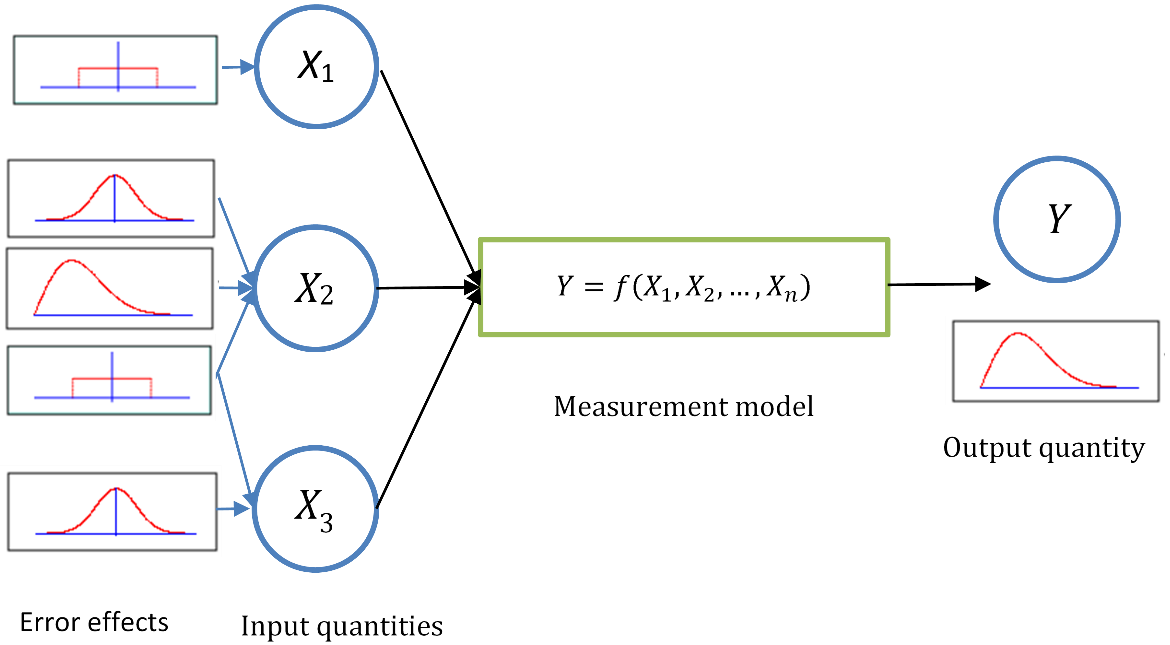
CoMet Packages:

IDEAS-QA4EO

NPL National Physical Laboratory

Propagating Uncertainties with punpy

- 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



IDEAS-QA4EO

NPL National Physical Laboratory

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*
  - *…*

Exercise 1

punpy

- Please go to https://www.comet-toolkit.org/user-guide/training/aria/

IDEAS-QA4EO

NPL National Physical Laboratory

The CoMet Toolkit – Error Correlation

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

ARIA workshop - 15/04/2026

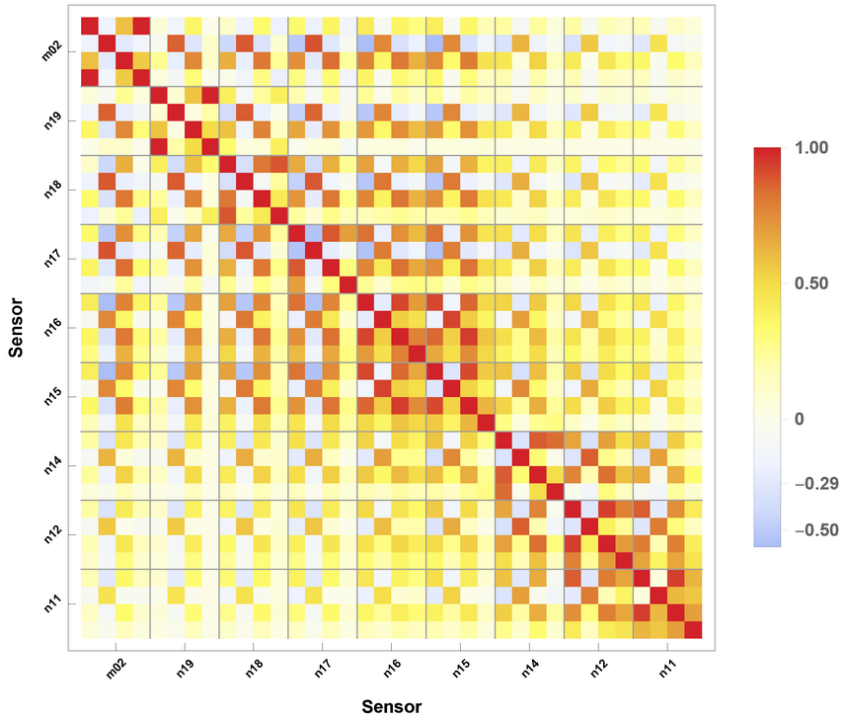
Error Correlation

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.

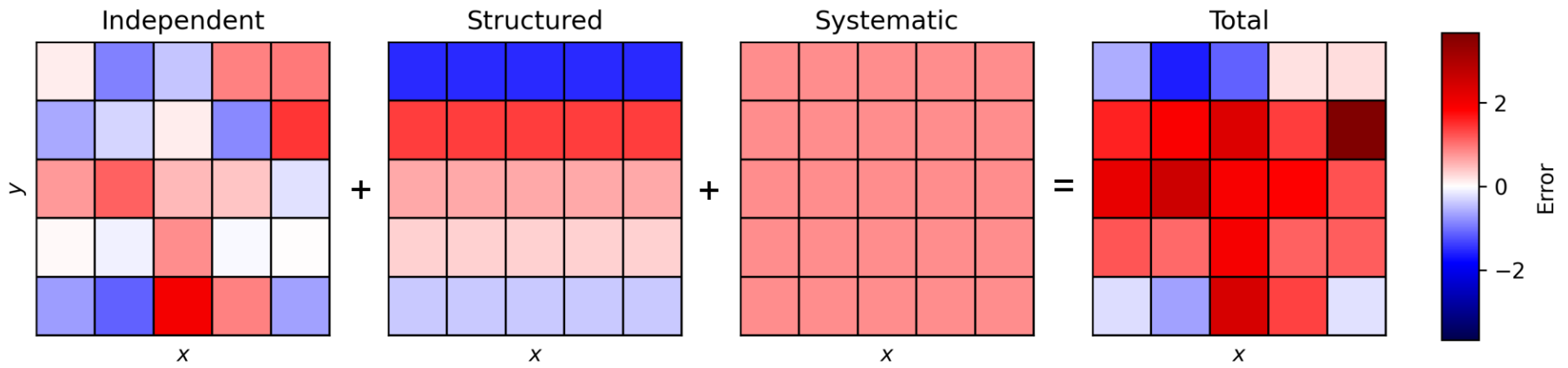
Why does it Matter?

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

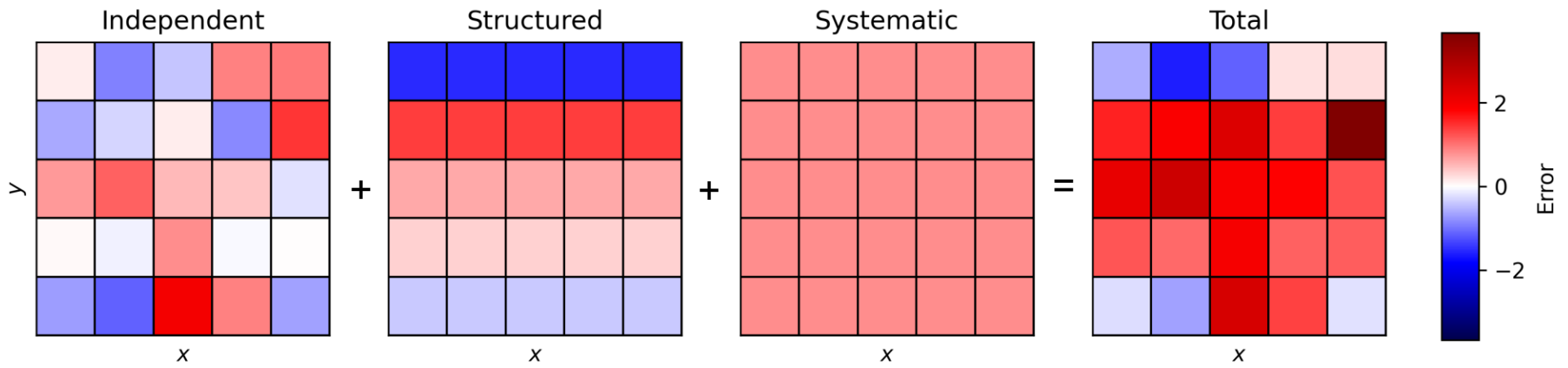


Error correlation matrix from Giering et al. 2019

ND Datasets & Error-Correlation



ND Datasets & Error-Correlation

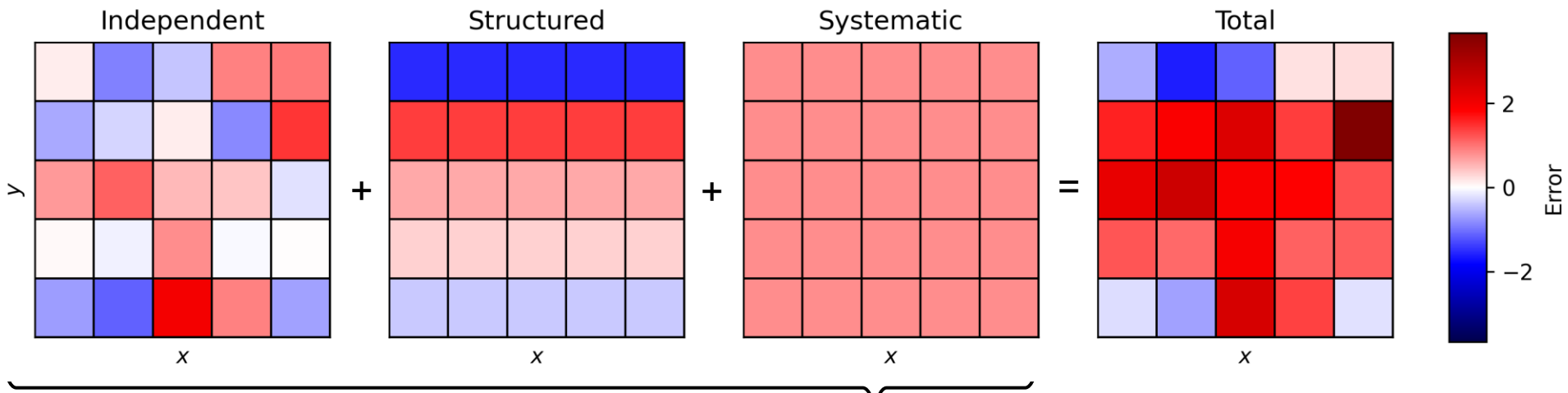


e.g., noise from detector

e.g., periodically updated correction

e.g., error on pre-flight calibration

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!

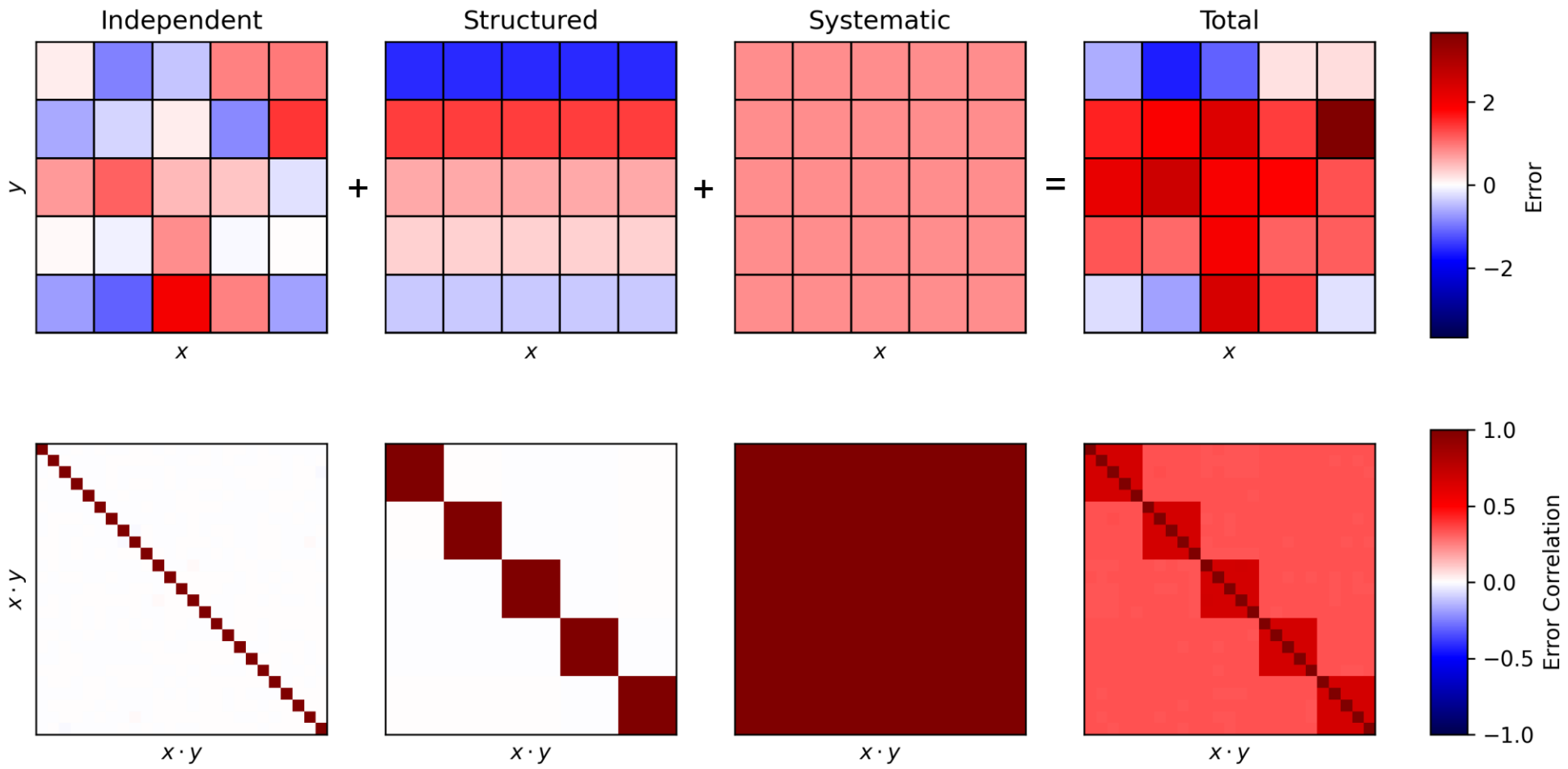
ND Datasets & Error-Correlation

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

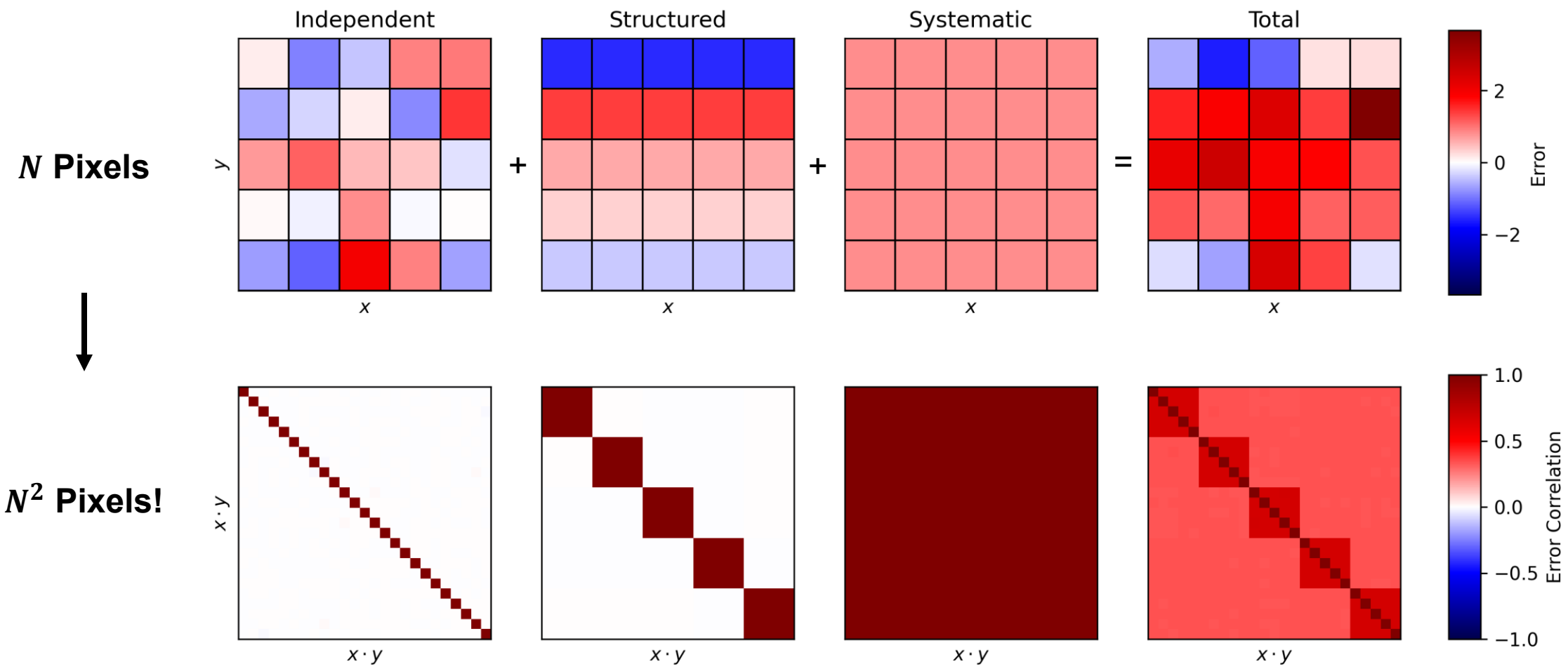
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

$$\begin{bmatrix} r_{1,1} & r_{2,1} & r_{3,1} & r_{4,1} & r_{5,1} & r_{6,1} & \cdots \\ r_{1,2} & r_{2,2} & r_{3,2} & & & & \\ r_{1,3} & r_{2,3} & r_{3,3} & & & & \\ r_{1,4} & & & \ddots & & & \\ r_{1,5} & & & & & & \\ r_{1,6} & & & & & & \\ r_{1,7} & & & & & & \\ r_{1,8} & & & & & & \\ \vdots & & & & & & \end{bmatrix}$$

ND Datasets & Error-Correlation

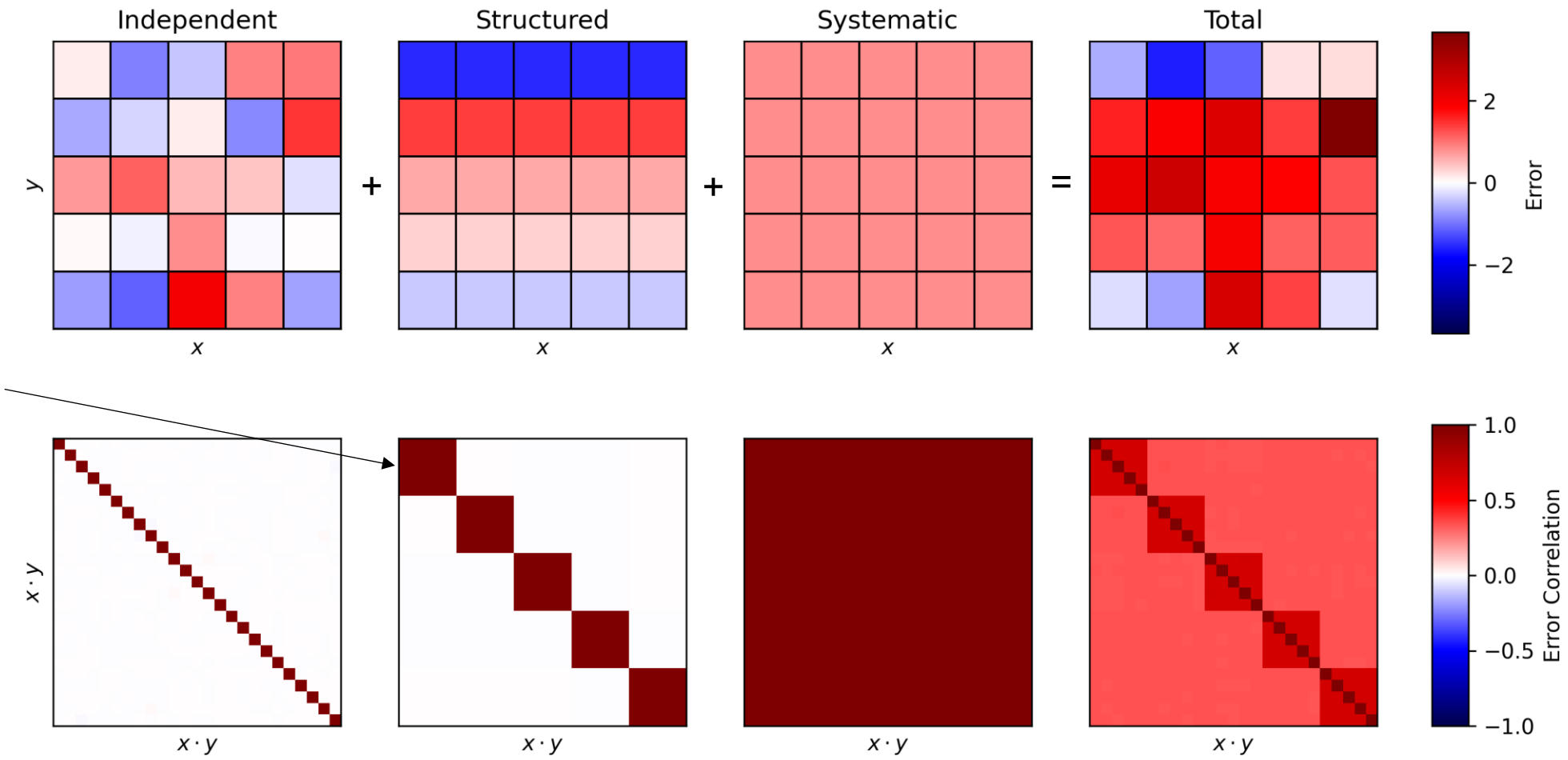


ND Datasets & Error-Correlation



ND Datasets & Error-Correlation

Sub-matrices for systematic error-correlation within row



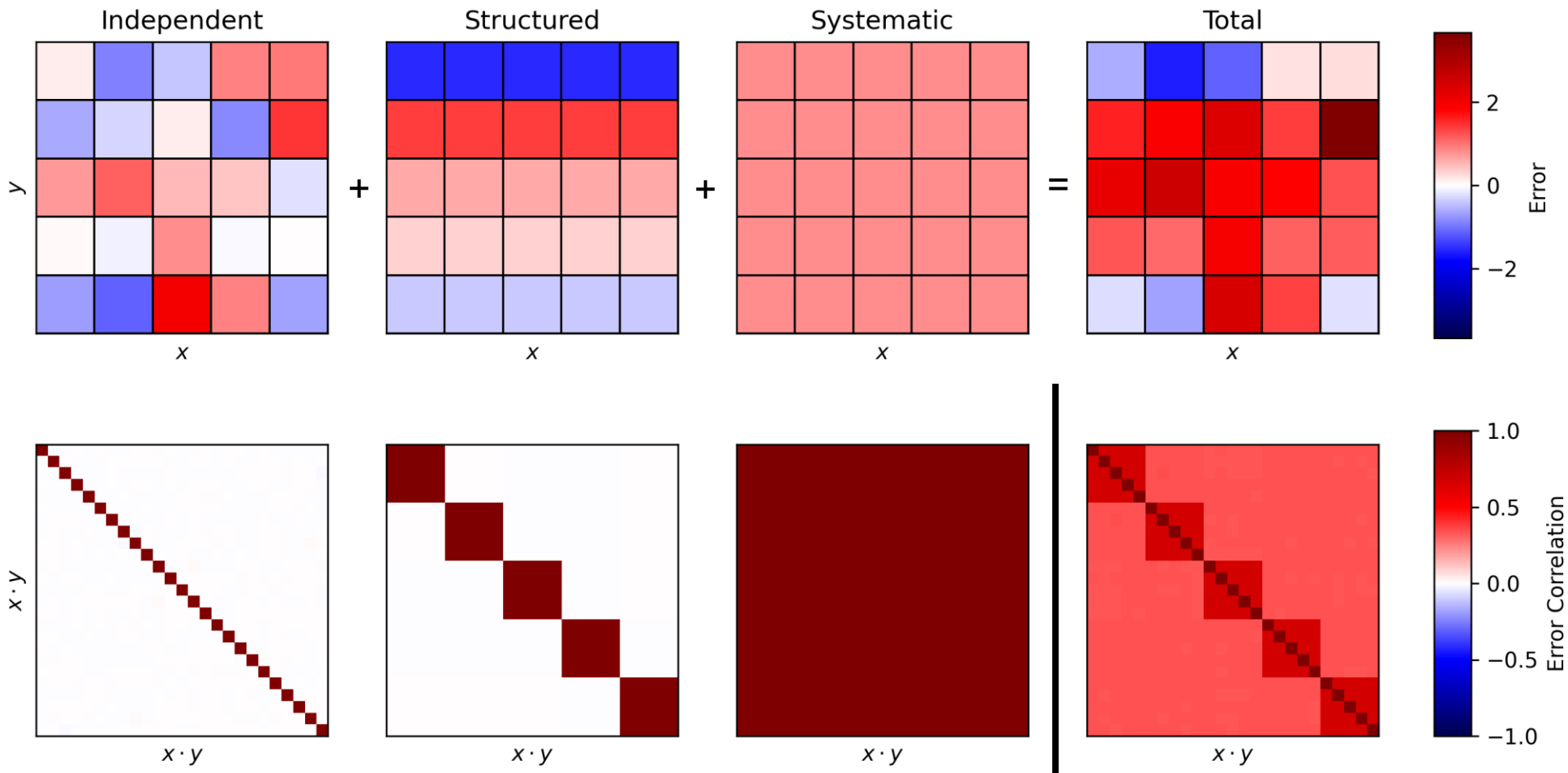
Identity

Block Diagonal

Full

Combination…

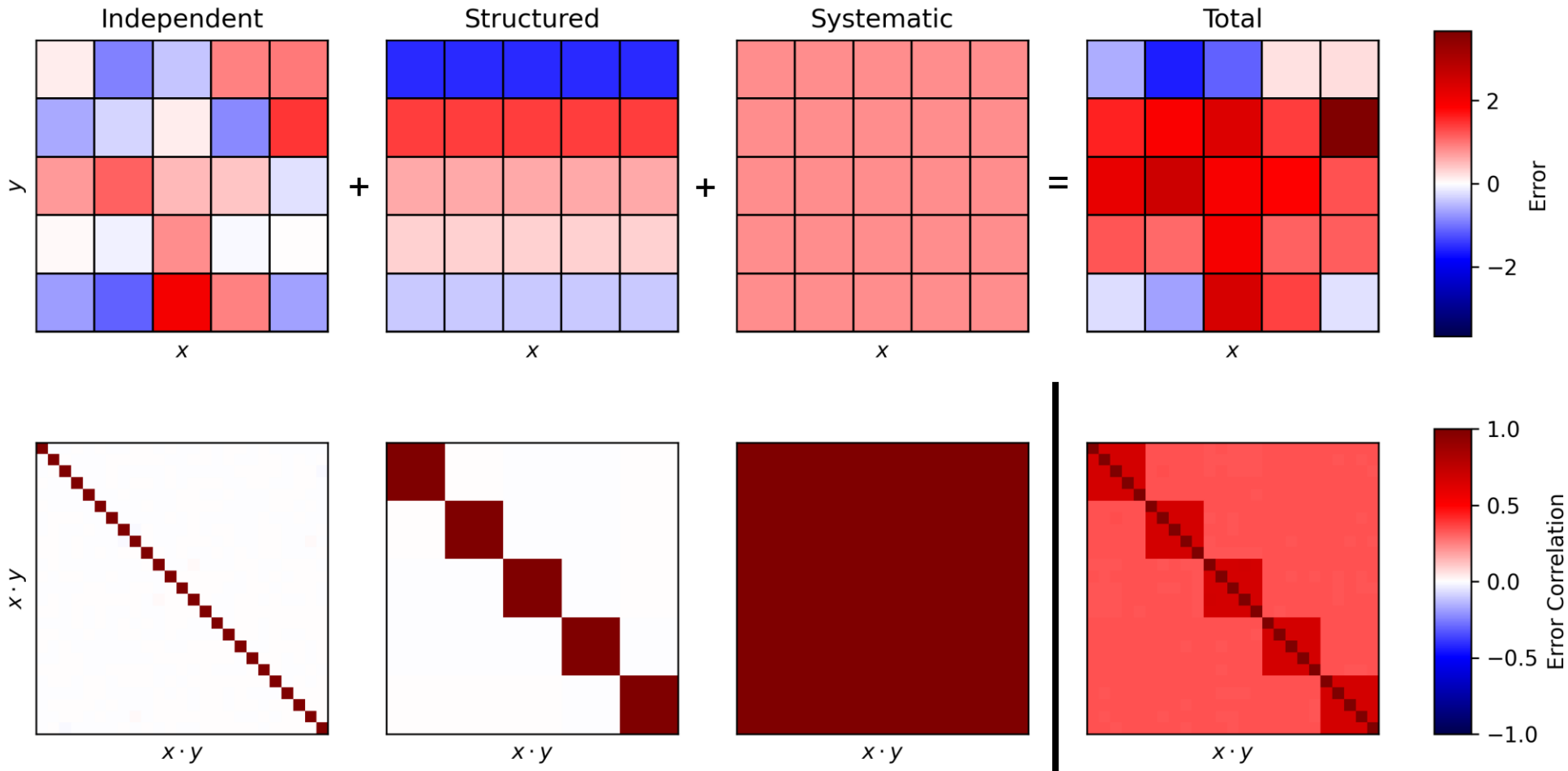
ND Datasets & Error-Correlation



- 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

$$\boldsymbol{S = U\,R\,U^T}$$

🤷 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

Exercise 2 & 3

- Please go to https://www.comet-toolkit.org/user-guide/training/aria/

IDEAS-QA4EO

NPL National Physical Laboratory

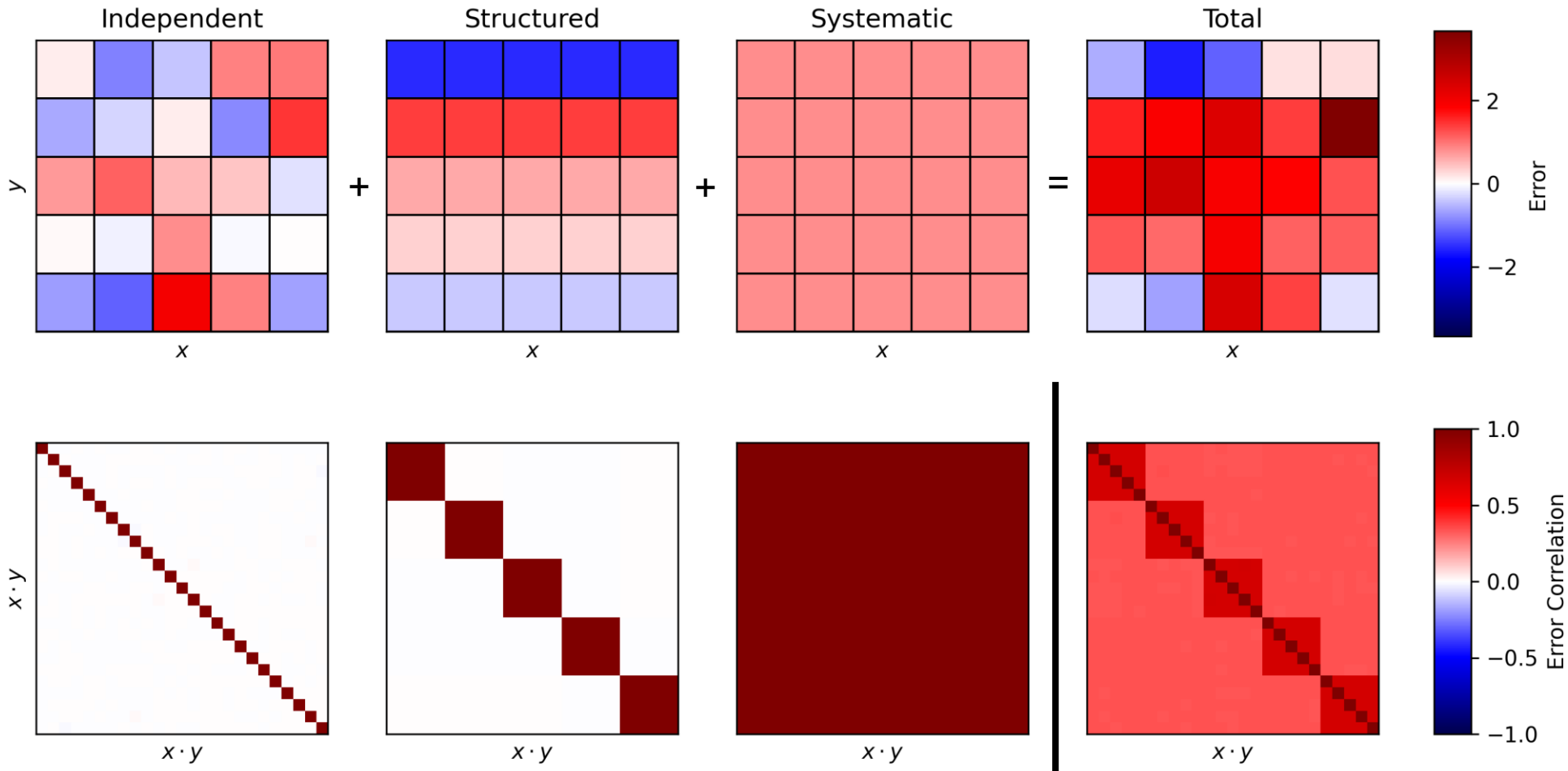
The CoMet Toolkit – UNC and obsarray

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

ARIA workshop - 15/04/2026

CoMet Packages:

ND Datasets & Error-Correlation



- Simply parameterisable
- One term might dominate at different scales

- Parameterisable via other terms
- Same complexity at all scales

Uncertainty metadata Naming Convention (UNC)

- Provide a standardised metadata format for uncertainty/error-covariance information.
- Enable compact representation of error-covariance matrices using parameterised metadata.

Interaction with Other Standards

The specification is intended to contribute to and build upon an existing ecosystem of standards and best practices. In particular, the following are adhered to, to the extent possible:

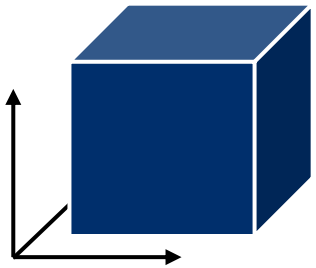
- The understanding of uncertainty concepts defined in the JCGM GUM (Guide to the expression of uncertainty in measurement) suite of documents.
- The definition of uncertainty-related terminology defined in the JGCM VIM(International Vocabulary for Metrology).
- The NetCDF data model for creating self-describing, array-oriented scientific datasets.
- The Climate and Forecast (CF) conventions on metadata for weather and climate data.

UNC Specification

obsarray

Uncertainty Variable Metadata

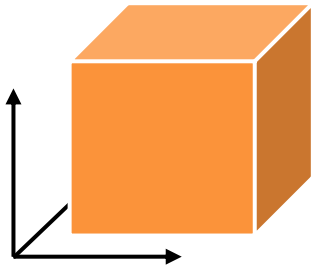
Observation Variables

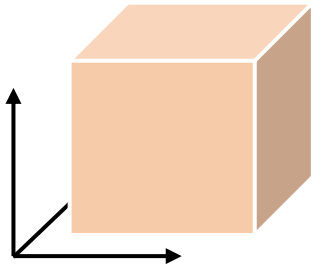


Metadata:
- Uncertainty Components

associated with

Uncertainty Variables





...

Metadata:
- **PDF Shape** (gaussian, …)
- **Units** (abs. or rel.)
- **Error-Correlation**…

IDEAS-QA4EO

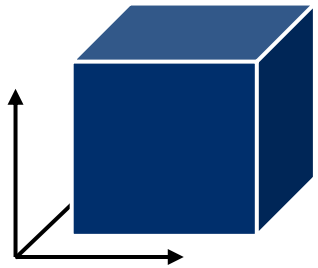
NPL National Physical Laboratory

UNC Specification

obsarray

🌡️ Example – Temperature Dataset

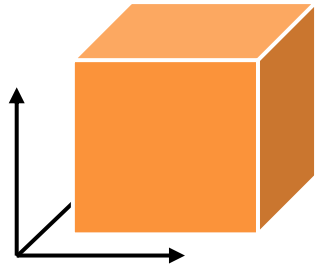
Observation Variables



temperature

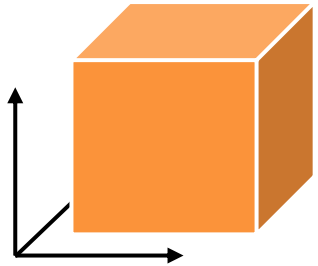
associated with

Uncertainty Variables



u_noise

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



u_calibration

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

IDEAS-QA4EO

NPL National Physical Laboratory

Summary of Three Forms of Error-Correlation with Examples

obsarray

FORM	DESCRIPTION	EXAMPLE
Form 1	Explicit error-correlation matrix	`"time: correlations (time×time matrix)"`
Form 2	Parameterized structure with optional parameters	`"lon: bell_shaped_relative (sigma: 5, n: 15)"`
Form 3	Qualitative non-standardized description	`"time: (temporal correlated on daily scales)"`

UNC Progress – Towards CF Integration

obsarray

```
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);
```

```
variables:
    float sst(time, lat, lon) ;
        sst:standard_name = "sea_surface_temperature" ;
        sst:units = "K" ;
        sst:units_metadata = "temperature: on_scale" ;
        sst:uncertainty_variables = "unc" ;
    float unc(lat, lon) ;
        unc:error_correlation = "
                    lat: correlations
                    time: uncorrelated
                    lon: bell_shaped_relative (n: 15 sigma: sigma)" ;
    float correlations(lat, lat1) ;
    float sigma ;
        sigma:standard_name = "sea_surface_temperature" ;
        sigma:units = "K" ;
        sigma:units_metadata = "temperature: difference" ;
```

UNC Spec v0.1

- Original proposal ~2023
- Implemented in e.g., Hypernets

Towards CF-UNC Spec v1.0

- Update with support from Reading, NCAS
  - More idiomatic CF-style syntax

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!

IDEAS-QA4EO

NPL National Physical Laboratory

obsarray

NPL

Example Usage

First we build an example dataset that represents a time series of temperatures (for more on how do this see the xarray documentation).

```
In [1]: import numpy as np

In [2]: import xarray as xr

In [3]: import obsarray

# build an xarray to represents a time series of temperatures
In [4]: temps = np.array([20.2, 21.1, 20.8])

In [5]: times = np.array([0, 30, 60])

In [6]: ds = xr.Dataset(
   ...:     {"temperature": (["time"], temps, {"units": "degC"})},
   ...:     coords = {"time": (["time"], times, {"units": "s"})}
   ...: )
   ...:
```

obsarray

NPL

Example Usage

Uncertainty and error-covariance information for observation variables can be defined using the dataset's `unc` accessor, which is provided by **obsarray**.

```
# add random component uncertainty
In [7]: ds.unc["temperature"]["u_r_temperature"] = (
   ...:     ["time"],
   ...:     np.array([0.5, 0.5, 0.6]),
   ...:     {"err_corr": [{"dim": "time", "form": "random"}]}
   ...: )
   ...:

# add systematic component uncertainty
In [8]: ds.unc["temperature"]["u_s_temperature"] = (
   ...:     ["time"],
   ...:     np.array([0.3, 0.3, 0.3]),
   ...:     {"err_corr": [{"dim": "time", "form": "systematic"}]}
   ...: )
   ...:
```

obsarray

NPL

Example Usage

The defined uncertainty information then can be interfaced with, for example:

```
# get total combined uncertainty of all components
In [9]: ds.unc["temperature"].total_unc()
Out[9]:
<xarray.DataArray (time: 3)> Size: 24B
array([0.58309519, 0.58309519, 0.67082039])
Coordinates:
  * time     (time) int64 24B 0 30 60

# get total error-covariance matrix for all components
In [10]: ds.unc["temperature"].total_err_cov_matrix()
Out[10]:
<xarray.DataArray (time: 3)> Size: 72B
array([[0.34, 0.09, 0.09],
       [0.09, 0.34, 0.09],
       [0.09, 0.09, 0.45]])
Dimensions without coordinates: time, time
```

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

```python
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)
```

IDEAS-QA4EO

NPL National Physical Laboratory

UNC Roadmap

obsarray

Alpha Testing – ends March 2026

- Iteration of UNC v0. 1 -> CF-UNC v1.0
- Test implementation by Hypernets (updated) and CCI SST

Beta Testing – ends July 2026

- Open testing

  Do you want to participate?

Proposal to CF ~September 2026

Exercise 4

- Please go to https://www.comet-toolkit.org/user-guide/training/aria/

IDEAS-QA4EO

NPL National Physical Laboratory

The CoMet Toolkit – Wrap-up

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

ARIA workshop - 15/04/2026

CoMet Application Examples

IDEAS-QA4EO

NPL National Physical Laboratory

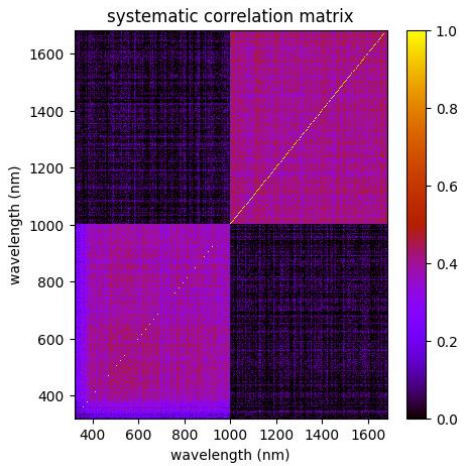
CoMet Toolkit in Action

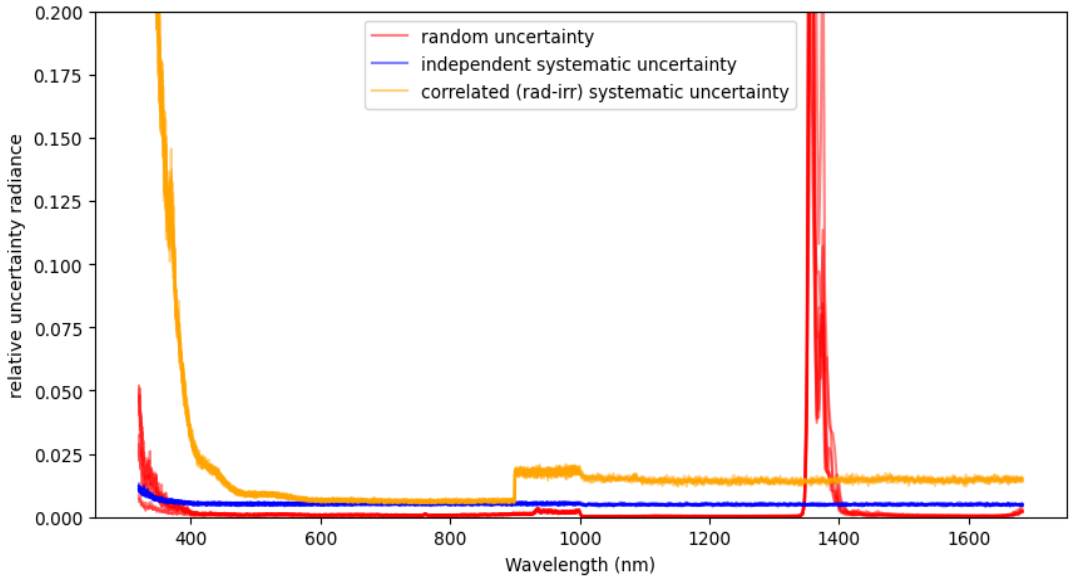
- Validated against **NIST** uncertainty engine:
  https://colab.research.google.com/github/comet-toolkit/comet_training/blob/main/NIST_example.ipynb

- *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)

IDEAS-QA4EO

NPL National Physical Laboratory