Workshop on developing Python frameworks for earth system sciences

Setting the scene

Baudouin Raoult

ECMWF

Software@ecmwf

- In support of operations
- In support of research
- In support of our users

Software@ecmwf

ecCodes/ODB, MARS, Magics, Metview,

MIR, ecFlow, Verify, ObsStat,

etc

A bit of history - MARS

```
retrieve,
    type   = fc,
    step   = 12/24/36/48,
    date   = 2000-01-01,
    time   = 12,
    param  = z/t,
    level  = 1000/500,
    target = myfile.grib
```

1985

A bit of history – METVIEW (batch)

```
lincom,
    type    = fc,
    step    = 12/24,
    date    = 2000-01-01,
    time    = 12,
    param   = z,
    a       = -1,
    b       = 1,
    level   = 1000,
    target = myfile.grib
```

a * (step 24) – b * (step 12)

1989

A bit of history – MARS compute

```
retrieve,
   type   = fc,
   step   = 12,
   date   = 2000-01-01,
   time   = 12,
   param  = z,
   level  = 1000
   field  = a

retrieve,
   step   = 24,
   field  = b

compute,
   formula = “b-a”,
   field   = c

write,
   field   = c,
   target  = myfile.grib
```

1993

A bit of history – MARS compute

Lazy loading of data: max 3 fields in memory at a time

50 fields, 39 MiB

```
retrieve,
   type   = pf,
   step   = 24,
   number = 1/to/50,
   date   = -1,
   time   = 12,
   param  = u,
   stream = enfo
   level  = 850
   field  = u

retrieve,
   param  = v,
   field  = v
```

100 fields, 78 MiB

```
compute,
   formula = "sqrt(u*u + v*v)",
   field   = speed

compute,
   formula = "mean(speed > 10)",
   field   = proba,
```

1 field

```
write,
   field   = proba,
   target  = myfile.grib
```

1993

A bit of history – Metview macro

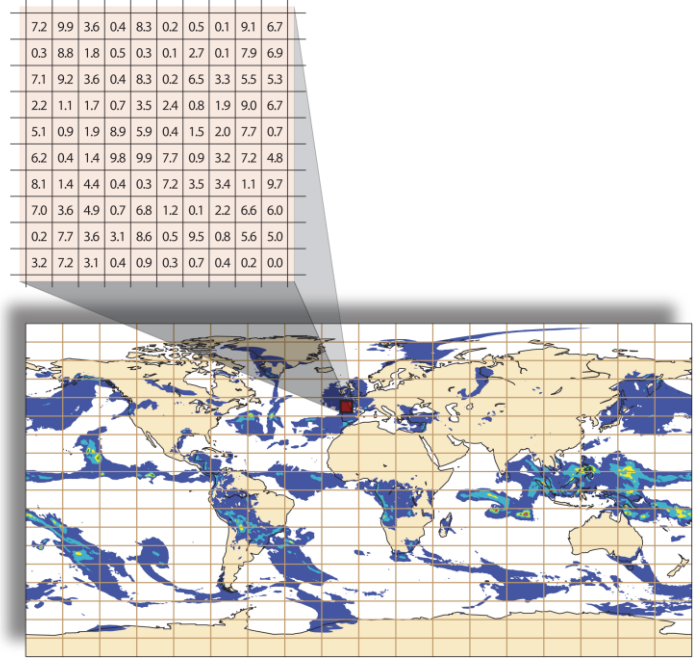
```
for step = 24 to 240 by 24

    u = retrieve(type: 'pf', step: step, number:  [1,'to',50], date: -1,
              time: 12, param: 'u', stream: 'enfo', level: 850)

    v = retrieve(type: 'pf', step: step, number:  [1,'to',50], date: -1,
                 time: 12, param: 'v', stream: 'enfo', level: 850)

    speed = sqrt(u*u + v*v)
    proba = mean(speed > 10)

    style = contour(contour_line_colour: 'red')
    plot(proba, style)

end for
```

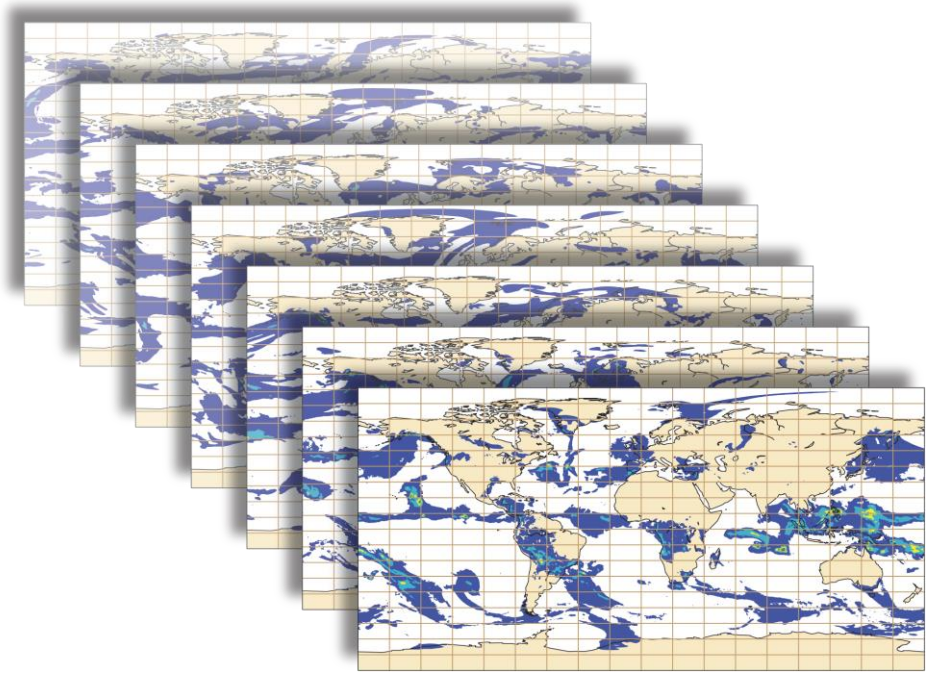
1994

TCL: 1988, Python2: 2000, Numpy: 2005

Metview

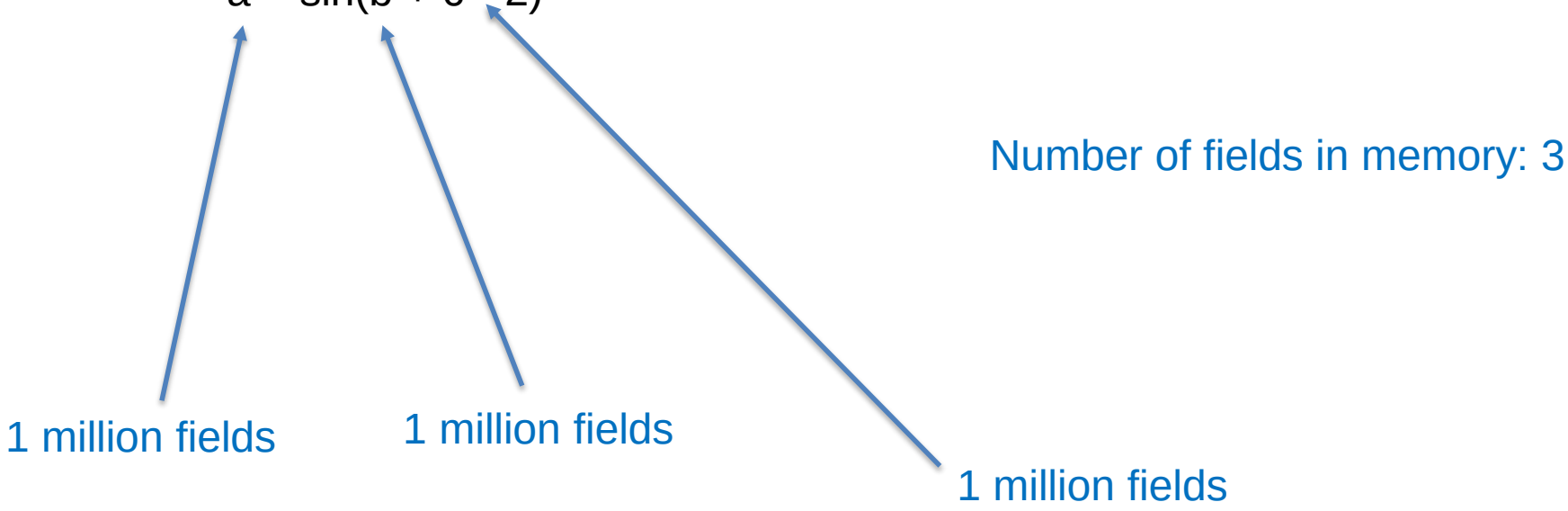
- Data access, manipulation and plotting
- Very high level abstraction (fieldset)
  - a = b + c
  - a = sin(b)
  - A fieldset is a collection of fields
  - All operations are done grid-point wise





Metview

- Lazy
  - Can process millions of fields in one function
  - a = sin(b + c * 2)



Number of fields in memory: 3

1994

Metview

- Fully featured
  - numbers, strings, dates, list, dictionaries, …
  - fields, observations, …
  - *if*, *while*, *function*, etc.
  - extendable (Fortran, C++, bash…)

1994

Metview

- Service oriented
  - Most functions are (internal) services
  - Multi-process using ’futures’

1994

So why Python?

- Joining a community
  - Documentation
  - Stack overflow
- Open Source
  - Code reuse
- Ecosystem
  - Contributing our domain knowledge

From Metview to Python

- Leverage *numpy*, *scipy*, *pandas*, *matplotlib*, *xarray*, etc

- Keep the high abstraction of Metview
- Remain lazy

- Fit for operations
  - Maintainability, robustness,
  - 24/7 support, monitoring, troubleshooting
- Fit for users
  - Feature rich
  - Multiple data format

Best practices: Python in operations

- Versioning
- Testing
- Deployments

Best practices : Python from a user point of view

- Site installations
- Virtual environments
- IDEs
- Debugging

Best practices : Contributing

- Packaging
- Documentation
- Incoming/outgoing contributions
- Licensing/IPR

Working groups

- Deploying and packaging Python frameworks
- Handling Big Data in Python
- (Code) Interoperability and common data structures

Thank you!

ECMWF