A Grid Computing environment for Design and Analysis of Computer Experiments

Yann Richet[1], David Ginsbourger[2], Olivier Roustant[3], Yves Deville[4]

[1] Radioprotection and Nuclear Safety Institute, France
[2] Institute of Geology and Hydrogeology, University of Neuchatel, Switzerland
[3] Graduate School of Engineering, Saint-Etienne, France
[4] Statistical consultant, Chambery , France

Great thanks for **Rserve** package and support: **Simon Urbanek**

Overview

Few words about Research and Industry

Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Overview

Few words about Research and Industry

Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Few words about Research and Industry

Reasons to work together

- Industry needs
  - increase productivity
  - overtake competitors
- Applied research needs
  - industrial applications
  - funding

Few words
about Research and Industry

Reasons to work together
- Industry needs
  - increase productivity
  - overtake competitors
- Applied research needs
  - industrial applications
  - funding

Resiliency against partnership
- Industry needs
  - short term RoI
  - efficient productive integration over existing practice
- Applied research needs
  - "formal bridge" between theory and application
  - mid / long term & continuous partnership

Few words about Research and Industry

A well-suited partnership **DICE Consortium**
http://www.dice-consortium.fr (Deep Inside Computer Experiments)

Industrial partners

Research partners

Few words about Research and Industry

A well-suited partnership **DICE Consortium**
http://www.dice-consortium.fr (Deep Inside Computer Experiments)

Industrial partners: applications and testing
- "orthogonal" high tech fields:
  - automotive, oil, aerospace, nuclear plants & safety
- shared funding: 40 000 € / year.partner

Research partners: scientific and software deliverables
- supplementary skills
- contractual contribution and goals
- hold scientific organization (PhD, postdoc, ...)

Few words
about Research and Industry

A well-suited partnership **DICE Consortium**
http://www.dice-consortium.fr (Deep Inside Computer Experiments)

Industrial partners: applications and testing
- "orthogonal" high tech fields:
  - automotive, oil, aerospace, nuclear plants & safety
- shared funding: 40 000 € / year.partner

Research partners: scientific and software deliverables
- supplementary skills
- contractual contribution and goals
- hold scientific organization (PhD, postdoc, ...)

Finite term project
- 3 years long & every 6 month meeting focus on advances
- software deliverables to be released as OSS (GPL/LGPL) in the end
- scientific deliverables to be released in ~ public domain in the end

Overview

Few words about Research and Industry

Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Overview

Few words about Research and Industry

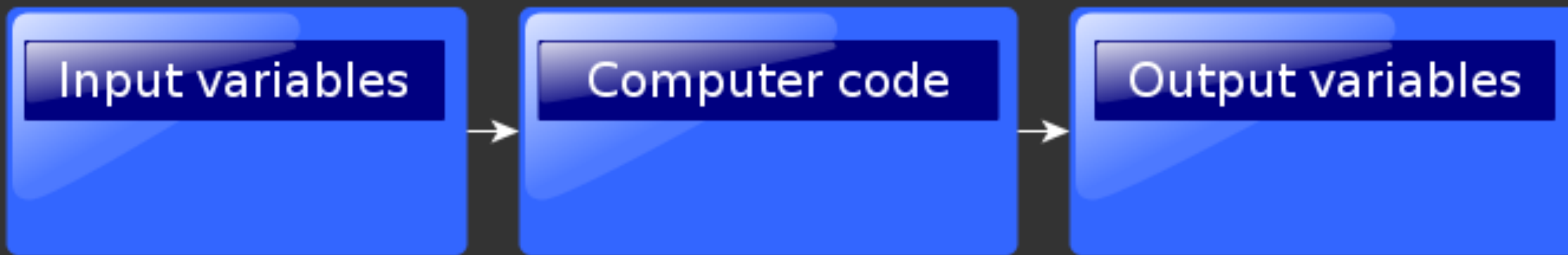
Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Computer Experiments framework



Computer code

- Used as an unknown function
- (Maybe) heavy CPU cost
- Represents any existing simulation solver: finite-elements, Monte Carlo, ...
  Fortran, C, close source, ...

Input variables

- Environment, control or simulation variables
- Scalar, vector, time sequences, ...

Output variables

- Interest values
- Scalar, vector, time sequences, ...

Computer Experiments framework

From math. tools ...

Design of experiments	DiceDesign, lhs, stats, ...
Surrogate modeling	DiceKriging, DiceEval, tgp, ...

Computer Experiments framework

From math. tools ...

Design of experiments	DiceDesign, lhs, stats, ...
Surrogate modeling	DiceKriging, DiceEval, tgp, ...

... To engineering issues

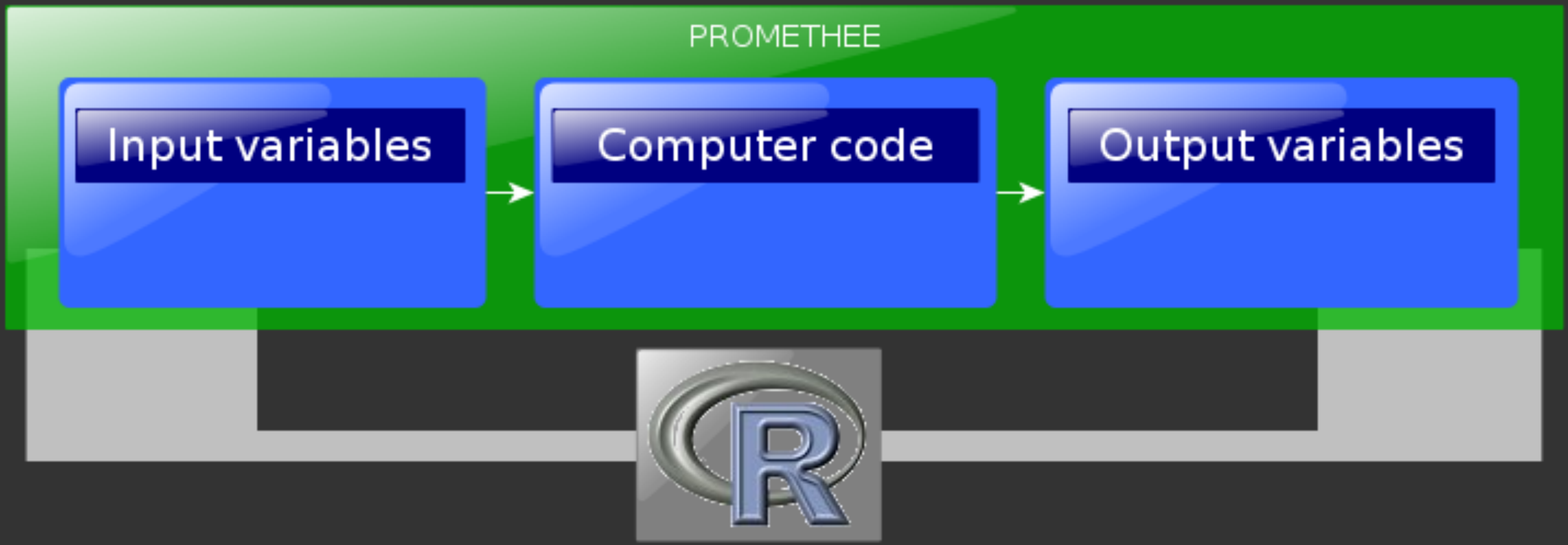
Sensitivity analysis	*DiceScreening*, sensitivity, ...
Uncertainties propagation	*DiceMRM*, lhs, boot, ...
Optimization	DiceOptim, ...
Inversion	...?

Computer Experiments framework

Software continuous integration: input / code / output

- Wrap "Computer code" as a [R] function
  - support computing environment (remote exec, network, grid load, ...)
  - integrate parallel capabilities of algorithms (primary issue !)

Computer Experiments framework



Software continuous integration: input / code / output

- Wrap "Computer code" as a [R] function
  - support computing environment (remote exec, network, grid load, ...)
  - integrate parallel capabilities of algorithms (primary issue !)

- Integrate [R] within grid computing environment
  - language interface & objects mapping [R] / {Java, C++, C#, Python, ...}
  - sequential access to algorithms (ask(...) & tell(...))

Overview

Few words about Research and Industry

Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Overview

Few words about Research and Industry

Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

PROMETHEE
Grid Computing environment

Software overview

- Engineering through "Computer Experiments"
  - Allows engineer to easily apply "brute" factorial design ...
  - ... then induces to formalize its model and goals in a DoE approach
  - Frequently needs for supplementary features (through dedicated code plugin)

PROMETHEE
Grid Computing environment

Software overview

- Engineering through "Computer Experiments"
  - Allows engineer to easily apply "brute" factorial design ...
  - ... then induces to formalize its model and goals in a DoE approach
  - Frequently needs for supplementary features (through dedicated code plugin)

- Distributed computing
  - Compatible with larger set of CPU boxes:
    - server, workstation, grid, cluster, ... and even (Windows) office desktop !
  - Easy dynamic merge of heterogeneous power

PROMETHEE
Grid Computing environment

Software overview

- Engineering through "Computer Experiments"
  - Allows engineer to easily apply "brute" factorial design ...
  - ... then induces to formalize its model and goals in a DoE approach
  - Frequently needs for supplementary features (through dedicated code plugin)

- Distributed computing
  - Compatible with larger set of CPU boxes:
    - server, workstation, grid, cluster, ... and even (Windows) office desktop !
  - Easy dynamic merge of heterogeneous power

- Application fields agnostic software
  - Any ASCII I/O software is compatible
  - All algorithms selectable for any computing software

PROMETHEE
Grid Computing environment

Software overview

- Engineering through "Computer Experiments"
  - Allows engineer to easily apply "brute" factorial design ...
  - ... then induces to formalize its model and goals in a DoE approach
  - Frequently needs for supplementary features (through dedicated code plugin)

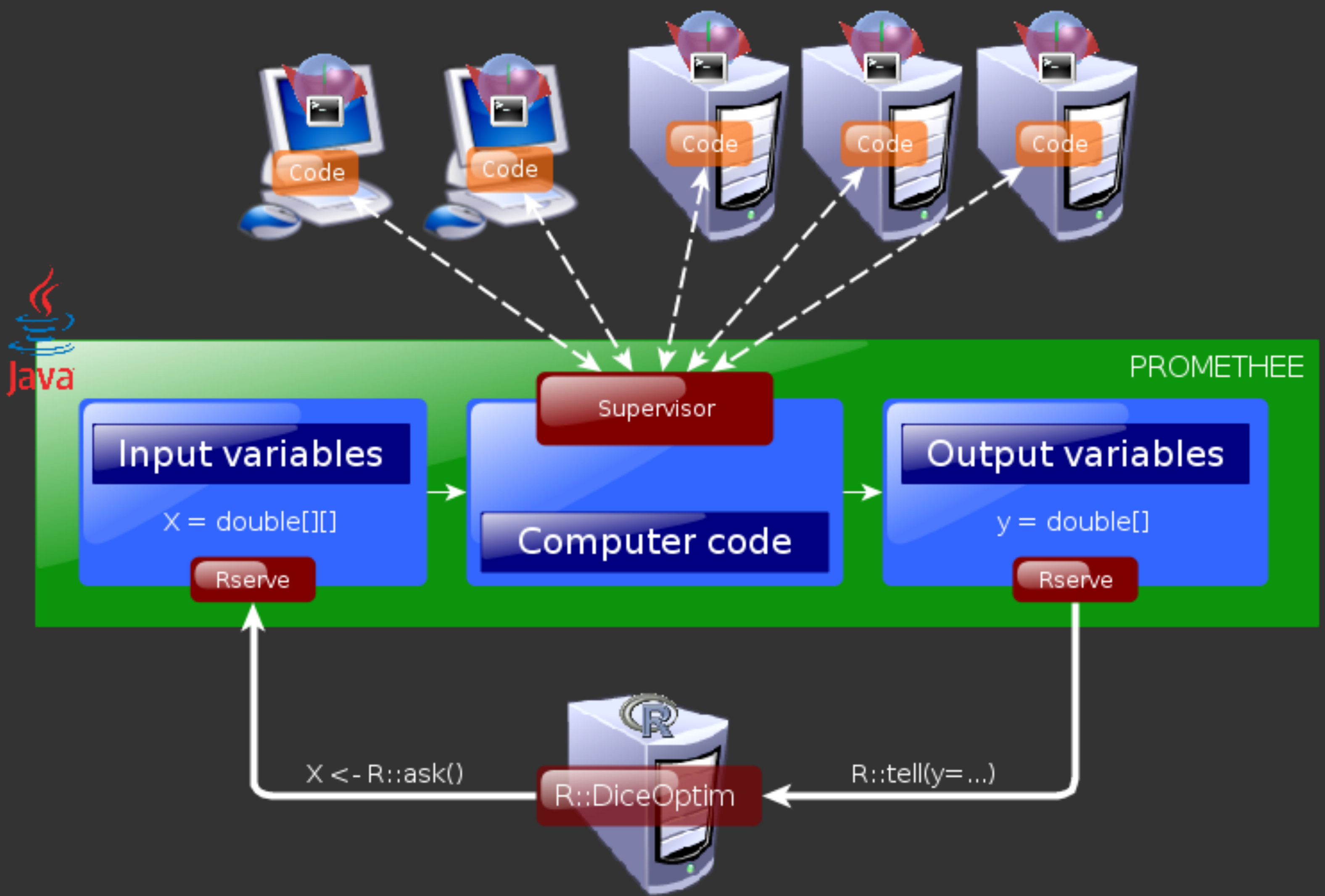
- Distributed computing
  - Compatible with larger set of CPU boxes:
    - server, workstation, grid, cluster, ... and even (Windows) office desktop !
  - Easy dynamic merge of heterogeneous power

- Application fields agnostic software
  - Any ASCII I/O software is compatible
  - All algorithms selectable for any computing software

- Extendability & wrapping
  - Basic (Groovy-DSL scripting) and extended (Java) plugins for computing code
  - Basic ([R]) and extended (Java::Rserve or Java::*) plugins for algorithms

PROMETHEE
Grid Computing environment

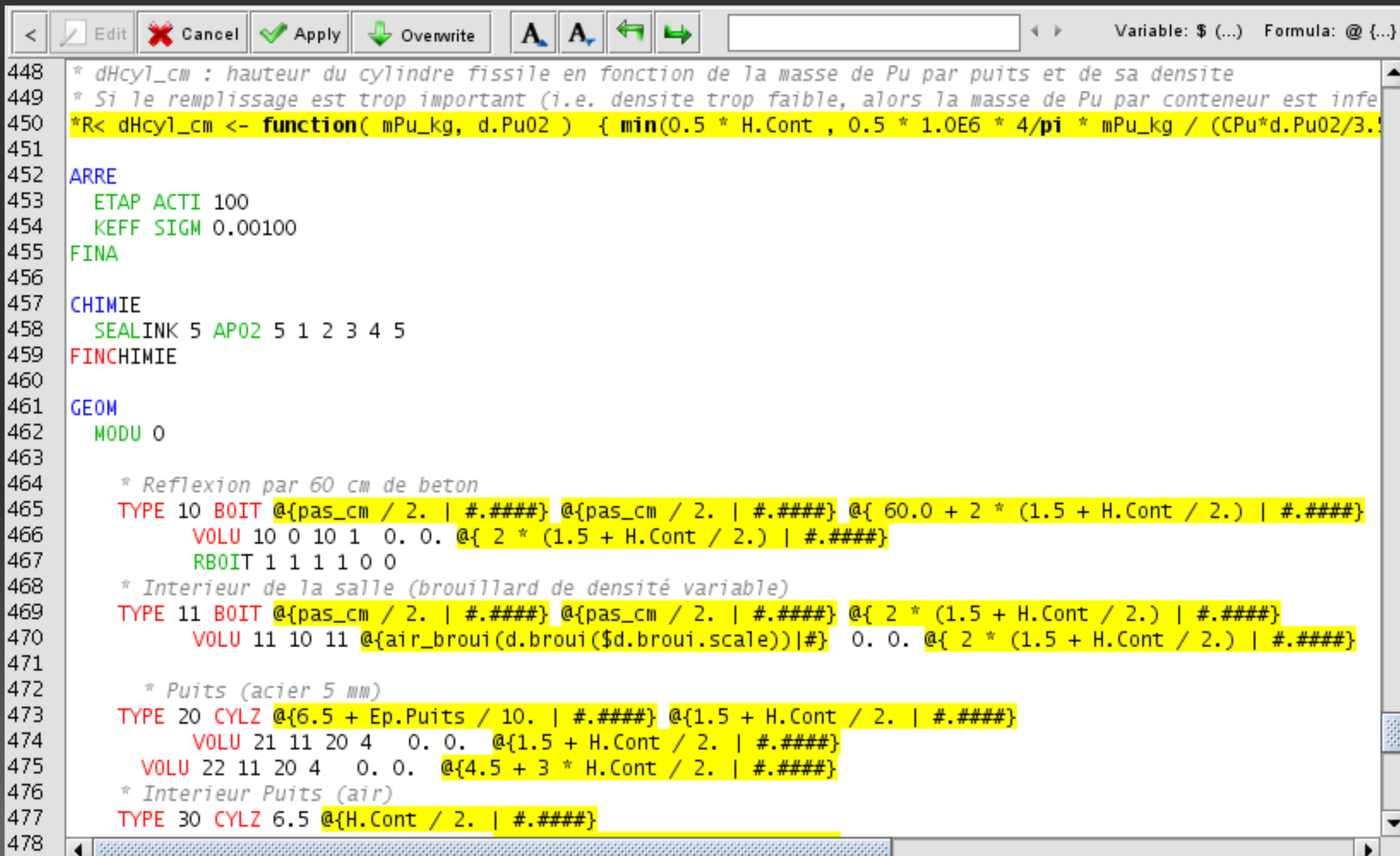
Network integration overview



PROMETHEE
Grid Computing environment

[R] tech. overview

- [R] used as a script engine for dataset parameterizing




```
* dHcyl_cm : hauteur du cylindre fissile en fonction de la masse de Pu par puits et de sa densite
* Si le remplissage est trop important (i.e. densite trop faible, alors la masse de Pu par conteneur est infe
*R< dHcyl_cm <- function( mPu_kg, d.PuO2 )  { min(0.5 * H.Cont , 0.5 * 1.0E6 * 4/pi * mPu_kg / (CPu*d.PuO2/3.!

ARRE
  ETAP ACTI 100
  KEFF SIGM 0.00100
FINA

CHIMIE
  SEALINK 5 APO2 5 1 2 3 4 5
FINCHIMIE

GEOM
  MODU 0

    * Reflexion par 60 cm de beton
    TYPE 10 BOIT @{pas_cm / 2. | #.####} @{pas_cm / 2. | #.####} @{ 60.0 + 2 * (1.5 + H.Cont / 2.) | #.####}
          VOLU 10 0 10 1  0. 0. @{ 2 * (1.5 + H.Cont / 2.) | #.####}
          RBOIT 1 1 1 1 0 0
    * Interieur de la salle (brouillard de densité variable)
    TYPE 11 BOIT @{pas_cm / 2. | #.####} @{pas_cm / 2. | #.####} @{ 2 * (1.5 + H.Cont / 2.) | #.####}
          VOLU 11 10 11 @{air_broui(d.broui($d.broui.scale))|#}  0. 0. @{ 2 * (1.5 + H.Cont / 2.) | #.####}

      * Puits (acier 5 mm)
    TYPE 20 CYLZ @{6.5 + Ep.Puits / 10. | #.####} @{1.5 + H.Cont / 2. | #.####}
          VOLU 21 11 20 4   0. 0.  @{1.5 + H.Cont / 2. | #.####}
      VOLU 22 11 20 4   0. 0.  @{4.5 + 3 * H.Cont / 2. | #.####}
    * Interieur Puits (air)
    TYPE 30 CYLZ 6.5 @{H.Cont / 2. | #.####}
```

PROMETHEE
Grid Computing environment

[R] tech. overview

- [R] used as a script engine for dataset parameterizing
- [R]/Rserve used as an API inside Java DoE algorithm plugin

```
R.voidEval("km" + currentiteration + "_" + hcode + " <- km(y~" + kmodel + ","
        + "optim.method='gen',"
        + "penalty = NULL,"
        + "covtype='" + covtype + "',"
        + nuggetnoise_str
        + "design=X" + currentiteration + "_" + hcode + ","
        + "response=Y" + currentiteration + "_" + hcode + ","
        + "control=list(" + control_km + "))");

REXP exists = R.eval("exists('km" + currentiteration + "_" + hcode + "')");
if (exists == null || !(exists.asInteger() == 1)) {
    return new Status(Decision.DESIGN_OVER, "No km object built.");
}

R.savels(new File(_repository, "km" + (currentiteration) + "_" + hcode + ".Rdata"), (currentiteration) + "_" + hcode);

R.voidEval("EGO" + currentiteration + "_" + hcode + " <- max_qEI.CL(model=km" + currentiteration + "_" + hcode + ","
        + "npoints=" + batchSize + ","
        + "L=c(" + liar + "(" + (search_min ? "" : "-") + "Y" + currentiteration + "_" + hcode + "$y)," + liar_noise + "),"
        + "lower=c(" + ASCII.cat(",", min) + "),"
        + "upper=c(" + ASCII.cat(",", max) + "),"
        + "control=list(" + control_ego + ")"
        + (expertfunction != null ? ",weight.EI=" + expertfunction : "") + ")");

/*REXP*/ exists = R.eval("exists('EGO" + currentiteration + "_" + hcode + "')");
if (exists == null || !(exists.asInteger() == 1)) {
    return new Status(Decision.DESIGN_OVER, "No EGO object built.");
}
```

PROMETHEE
Grid Computing environment

[R] tech. overview

- [R] used as a script engine for dataset parameterizing
- [R]/Rserve used as an API inside Java DoE algorithm plugin
- [R] DoE algorithm plugin

```
#help=Efficient Global Optimization (EGO) algorithm
#type=Optimization
#output=Optimum
#parameters=initBatchSize=10,batchSize=10,iterations=10,nugget.estim=f

## constructor and initializer of R session
init <- function() {
    ...
}

## first design building. All variables are set in [0,1].
## d is the dimension, or number of variables
## @param d number of variables
buildInitialDesign <- function(d) {
    ...
}

## iterated design building.
## @param X data frame of current doe variables (in [0,1])
## @param Y data frame of current results
## @return data frame or matrix of next doe step
prepareNextDesign <- function(X,Y) {
    ...
}

## final analysis. All variables are set in [0,1]. Return HTML string
## @param X data frame of doe variables (in [0,1])
## @param Y data frame of  results
## @return HTML string of analysis
analyseDesign <- function(X,Y) {
    ...
}
```

```
## first design building. All variables are set in [0,1]. d is the dimension, or r
## @param d number of variables
buildInitialDesign <- function(d) {
    lhs <- maximinLHS(n=initBatchSize,k=d)
    if (Xbounds) {
      e=c(0,1)
      id=1
      while(id<d){
        e=rbind(cbind(e,0),cbind(e,1))
        id=id+1
      }
      Xinit=rbind(as.matrix(lhs),as.matrix(e))
    } else { Xinit=as.matrix(lhs) }
    return(Xinit)
}

## iterated design building.
## @param X data frame of current doe variables (in [0,1])
## @param Y data frame of current results
## @return data frame or matrix of next doe step
prepareNextDesign <- function(X,Y) {
    if (iEGO > iterations) return();

    d = dim(X)[2]
    if (dim(Y)[2] == 2) { noise.var <<- as.array(Y[,2])
    } else { noise.var <<- NULL }

    if (Xautoscale=="" | Xautoscale=="FALSE" | Xautoscale=="false") { pEGO <<- re
    } else { pEGO <<- searchOptim(Xautoscale,init=pEGO,X=X,Y=Y) }
    cat("p=",pEGO,"\n")

    if (search_min) {y=Y[,1]} else {y=-Y[,1]}

    kmi <- km(control=list(trace=FALSE),trend,optim.method='gen',penalty = NULL,c

    EGOi <- max_qEI.CL(model=kmi,npoints=batchSize,L=liar(as.array(Y[,1])),lower=

    Xnext <- unscaleX(EGOi$par,pEGO)
```

Overview

Few words about Research and Industry

Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Overview

Few words about Research and Industry

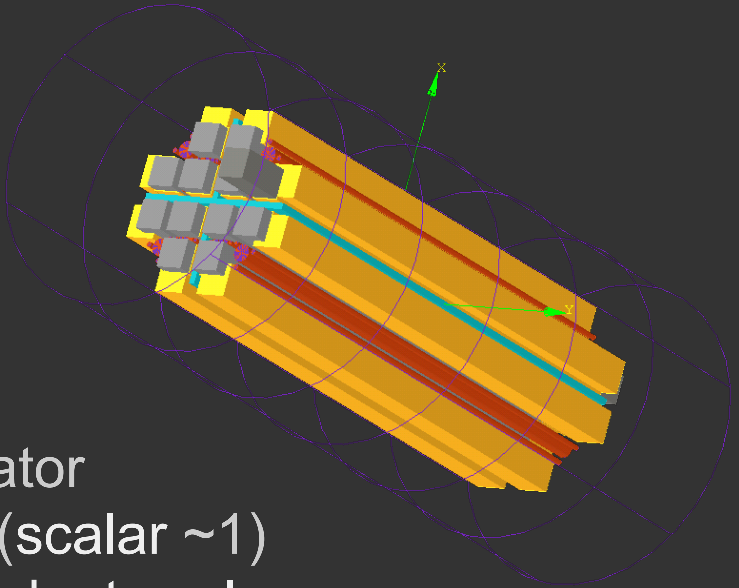
Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Real world example



Criticality safety assessment

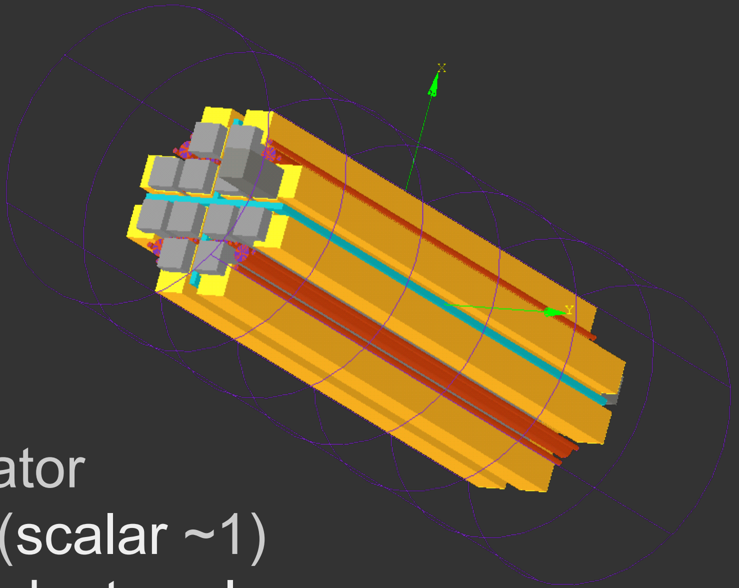
Computer code: Monte Carlo neutrons simulator

Output variables: neutron multiplication factor (scalar ~1)

Input variables: many hypothesis as independent scalar code input parameters

Engineering issue: find optimization (max) of output over hypothesis range

Real world example



Criticality safety assessment

Computer code: Monte Carlo neutrons simulator

Output variables: neutron multiplication factor (scalar ~1)

Input variables: many hypothesis as independent scalar code input parameters

Engineering issue: find optimization (max) of output over hypothesis range

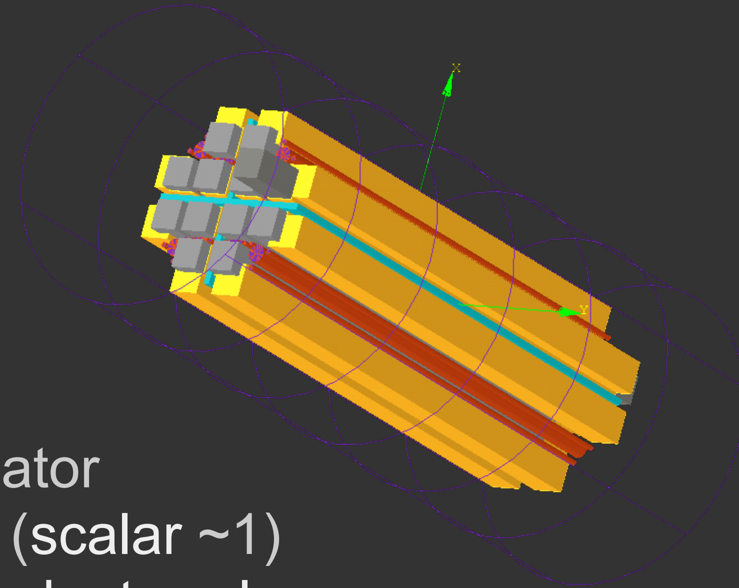
Old practical method (2 years ago)

Hierarchical (user's prior) selection of ~3 input variables

By-hand remote code launching (over interactive shell)

Iterative & orthogonal maximization search (<20 points of calculation)

Real world example



Criticality safety assessment

- Computer code: Monte Carlo neutrons simulator
- Output variables: neutron multiplication factor (scalar ~1)
- Input variables: many hypothesis as independent scalar code input parameters
- Engineering issue: find optimization (max) of output over hypothesis range

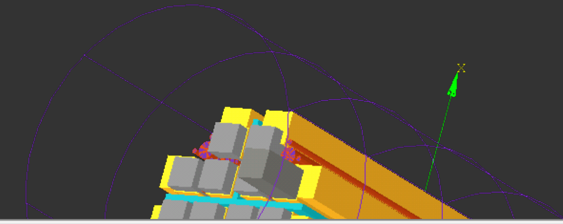
Old practical method (2 years ago)

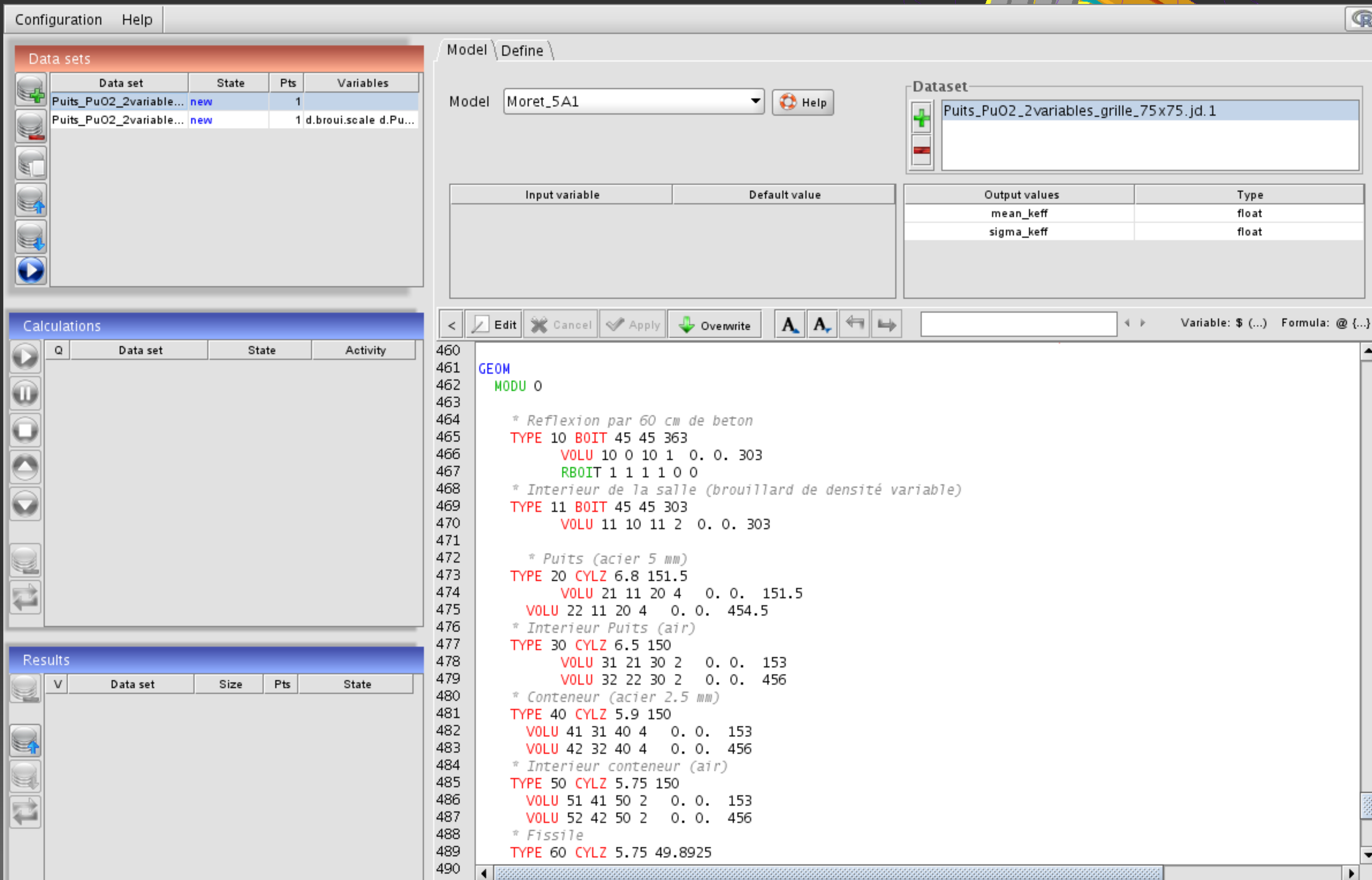
- Hierarchical (user's prior) selection of ~3 input variables
- By-hand remote code launching (over interactive shell)
- Iterative & orthogonal maximization search (<20 points of calculation)

Within Computer Experiments paradigm (PROMETHEE & R::DiceOptim / DiceKriging)

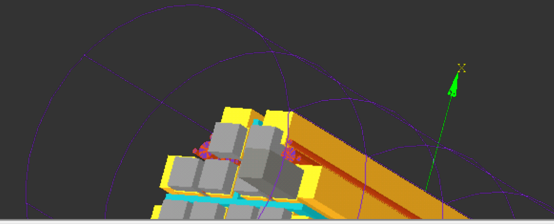
- No input variable ignored (no expert prior necessary)
- Automatic remote code launching & output parsing
- Global maximization of output (may support >1000 points of calculation)

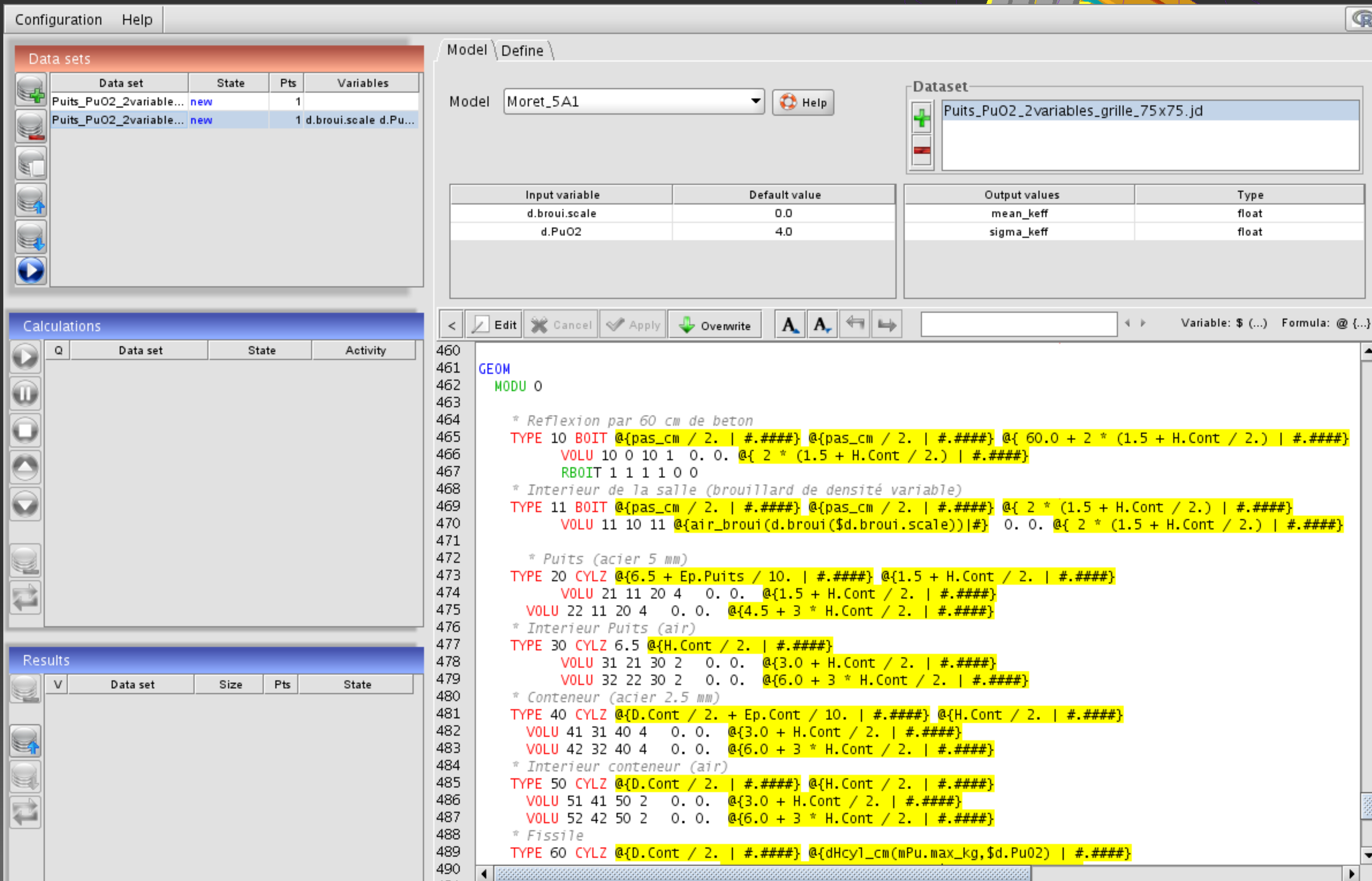
Real world example



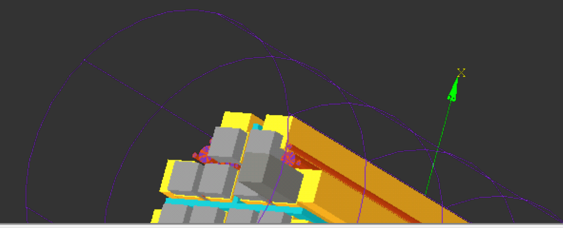


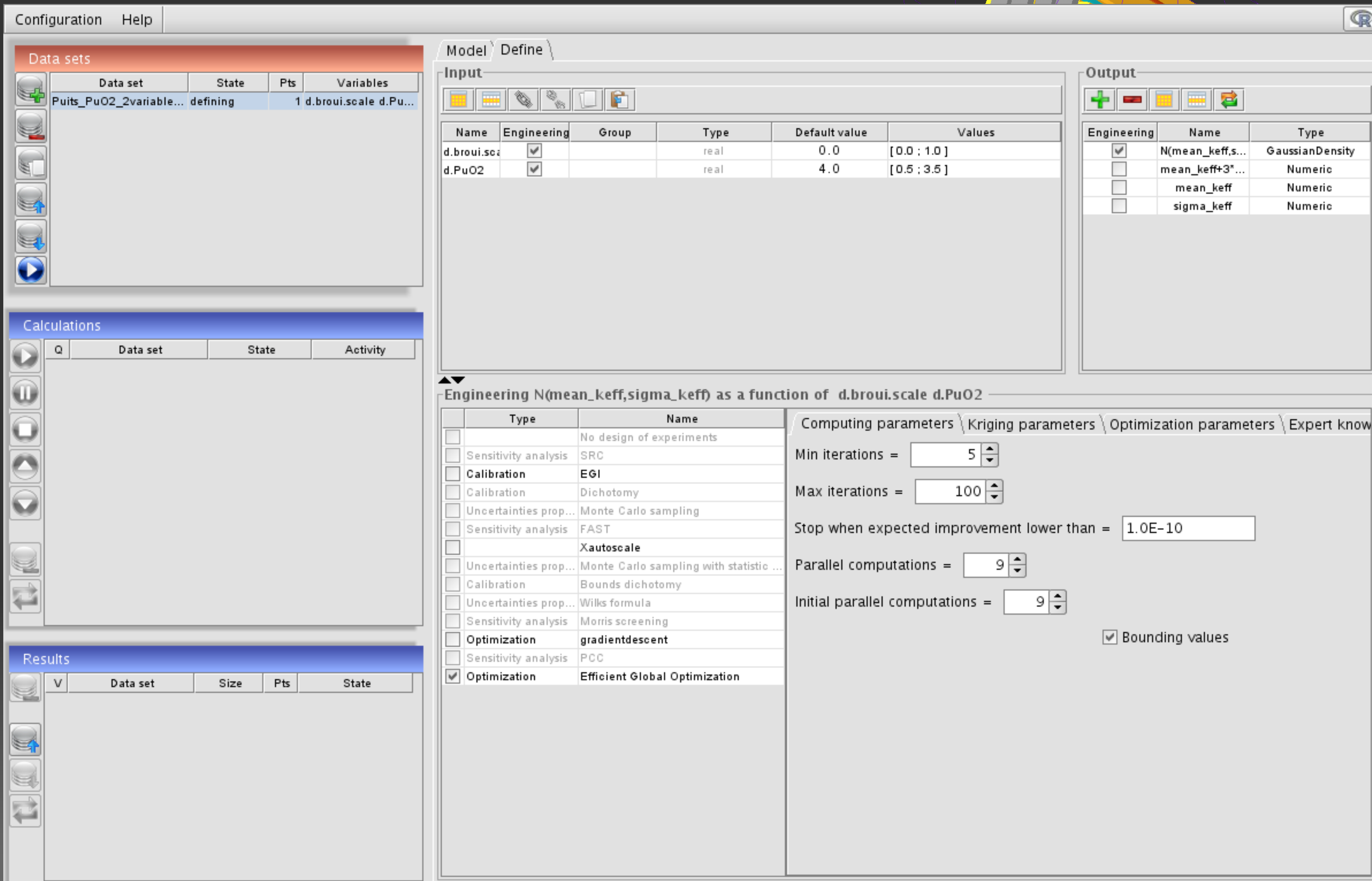
Real world example



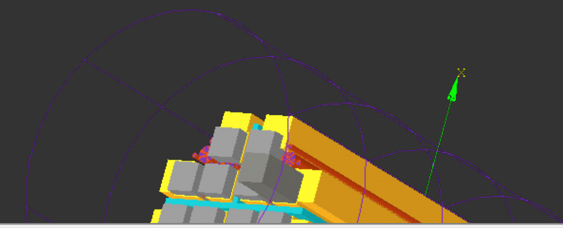


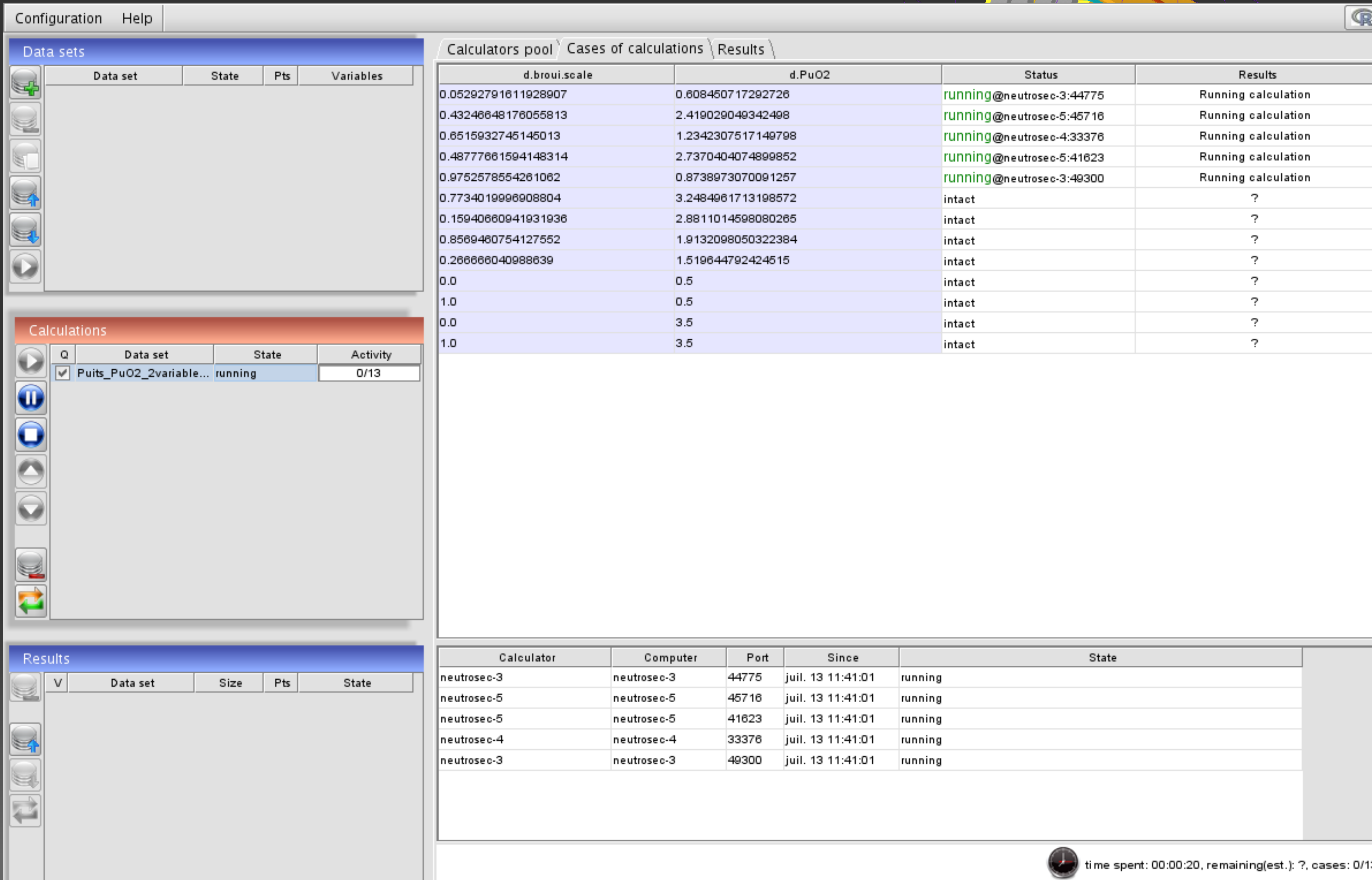
Real world example



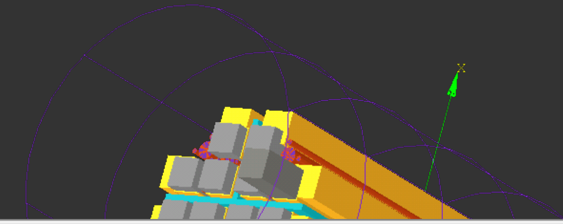


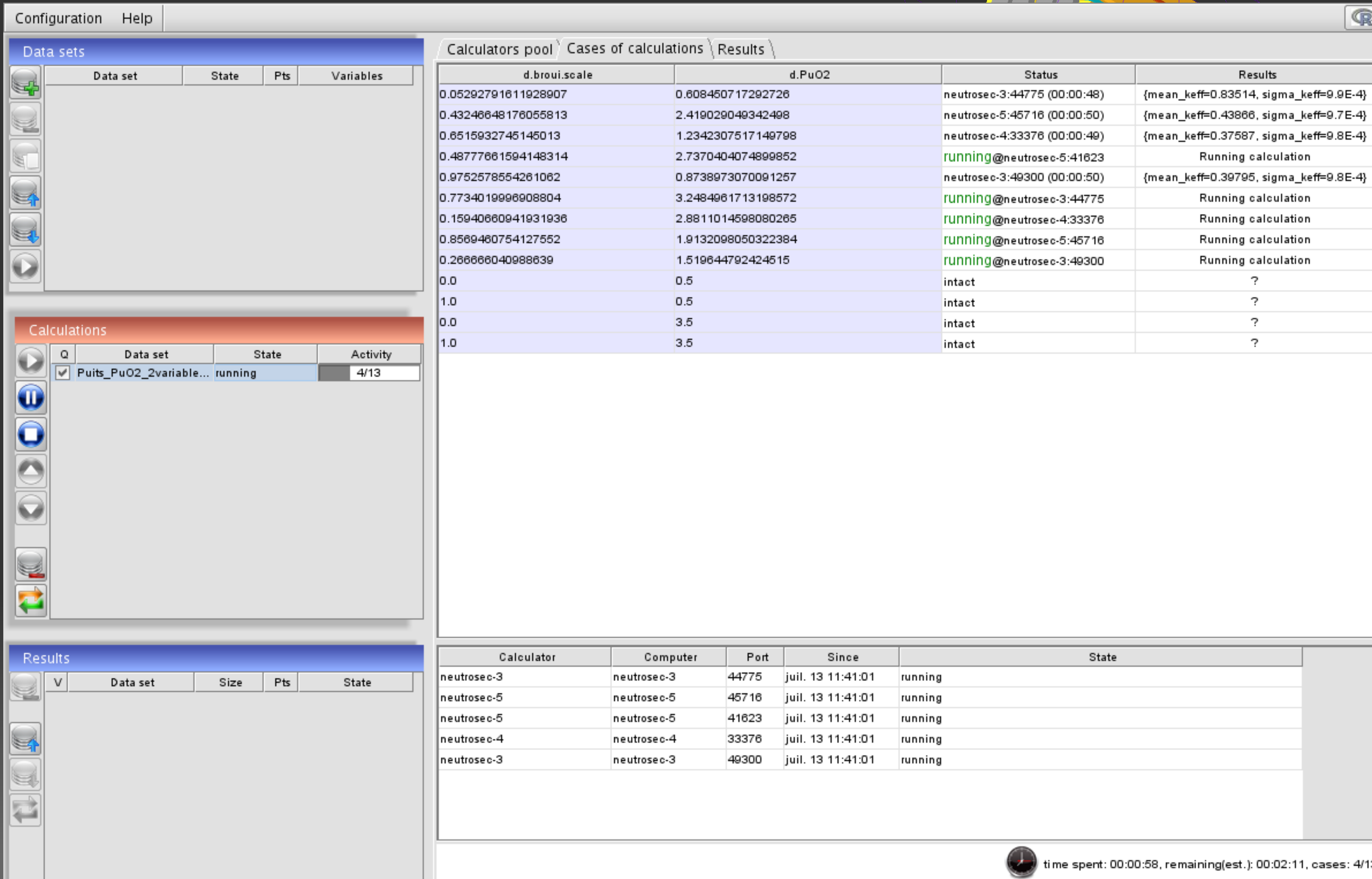
Real world example



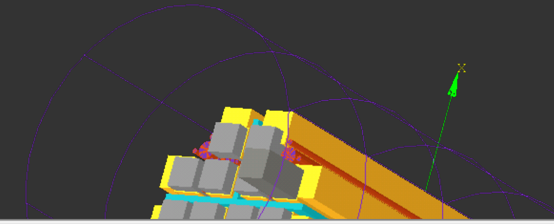


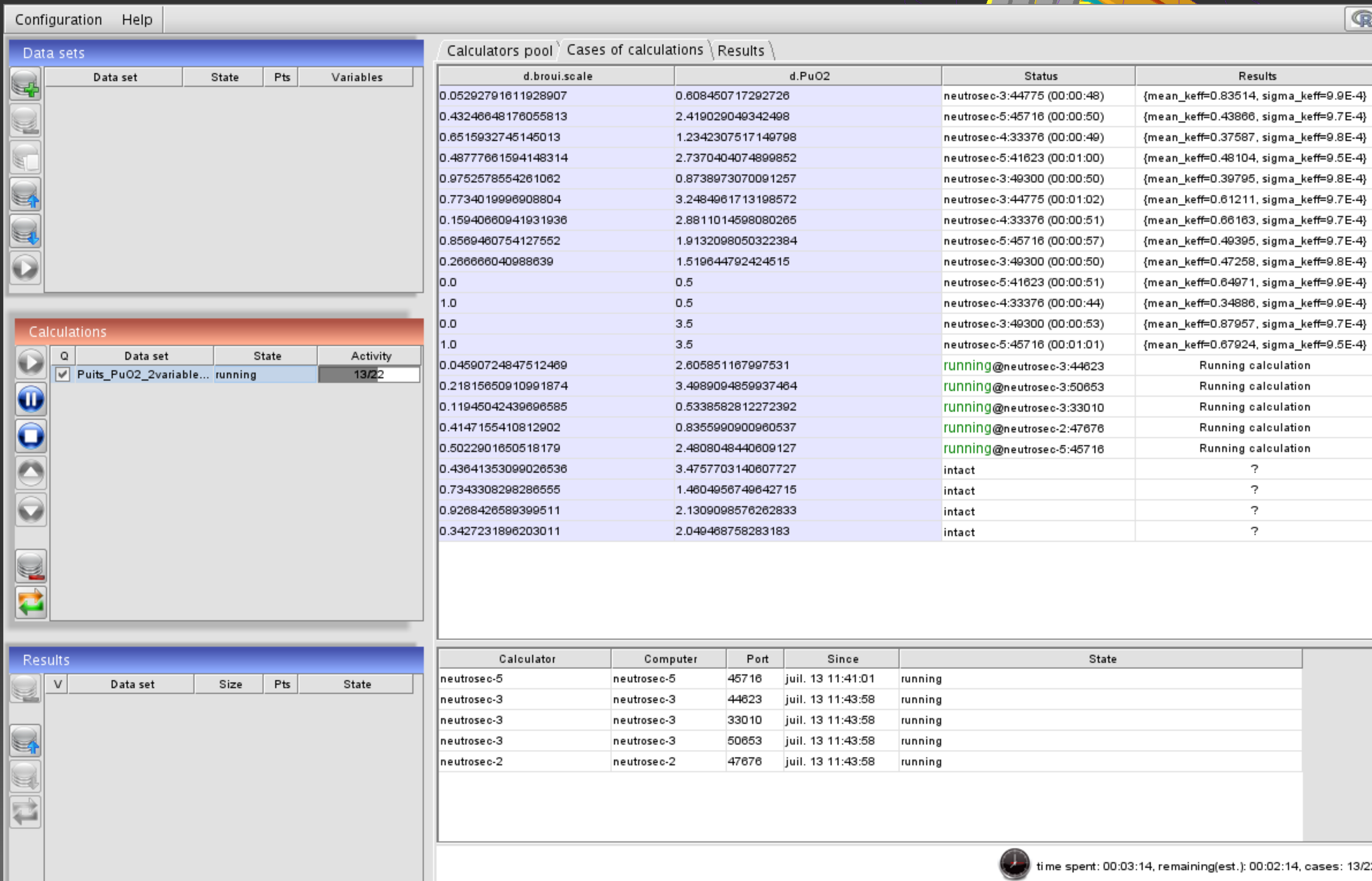
Real world example



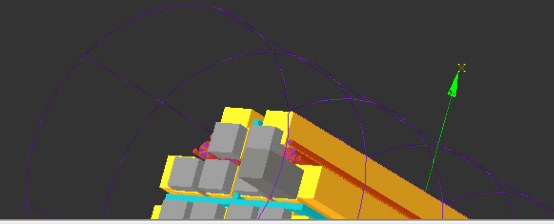


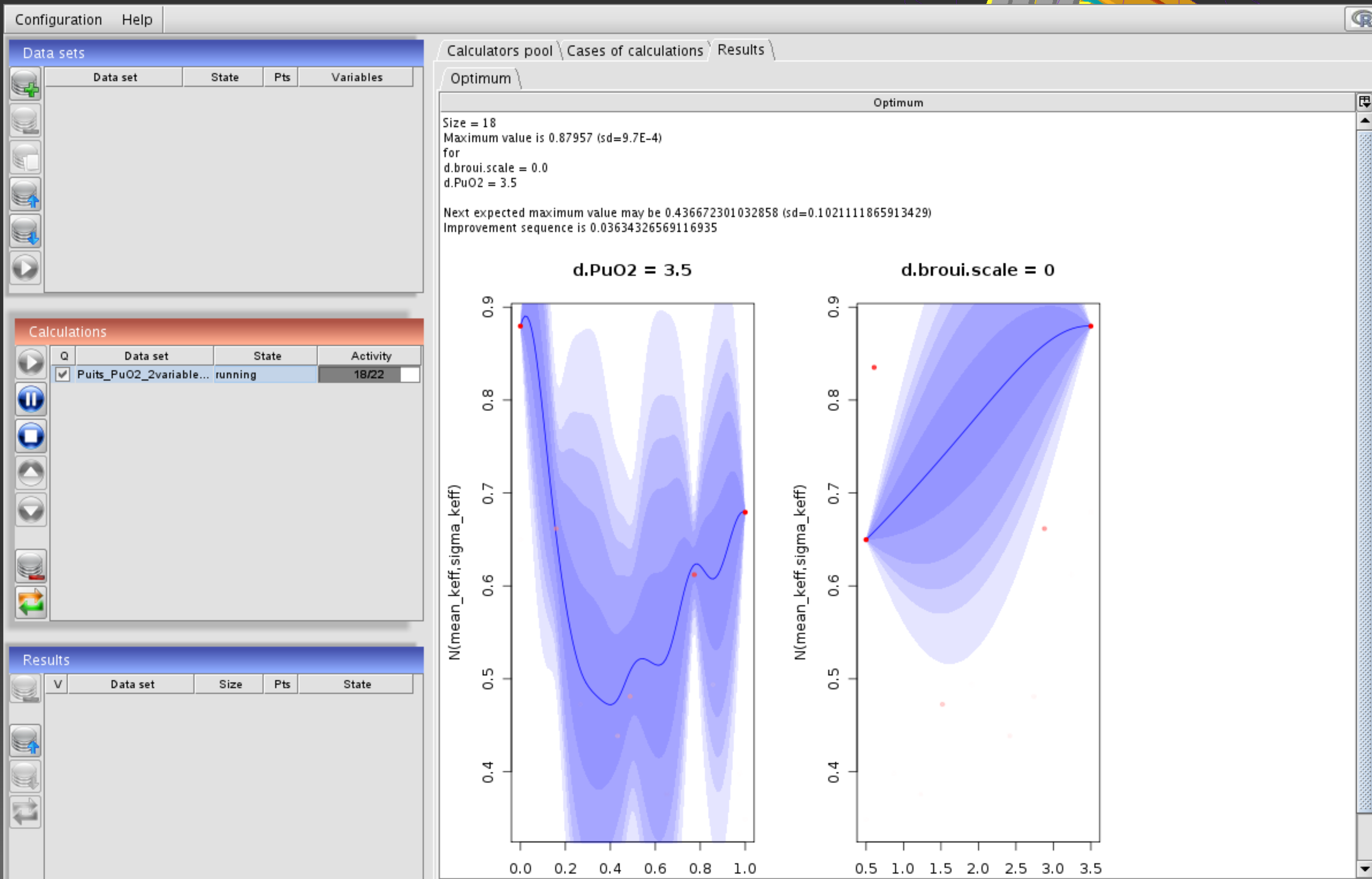
Real world example



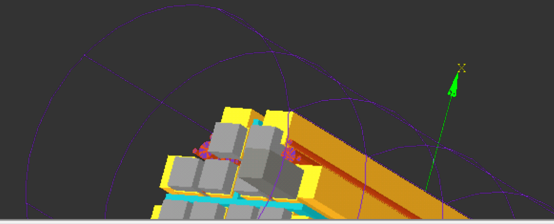


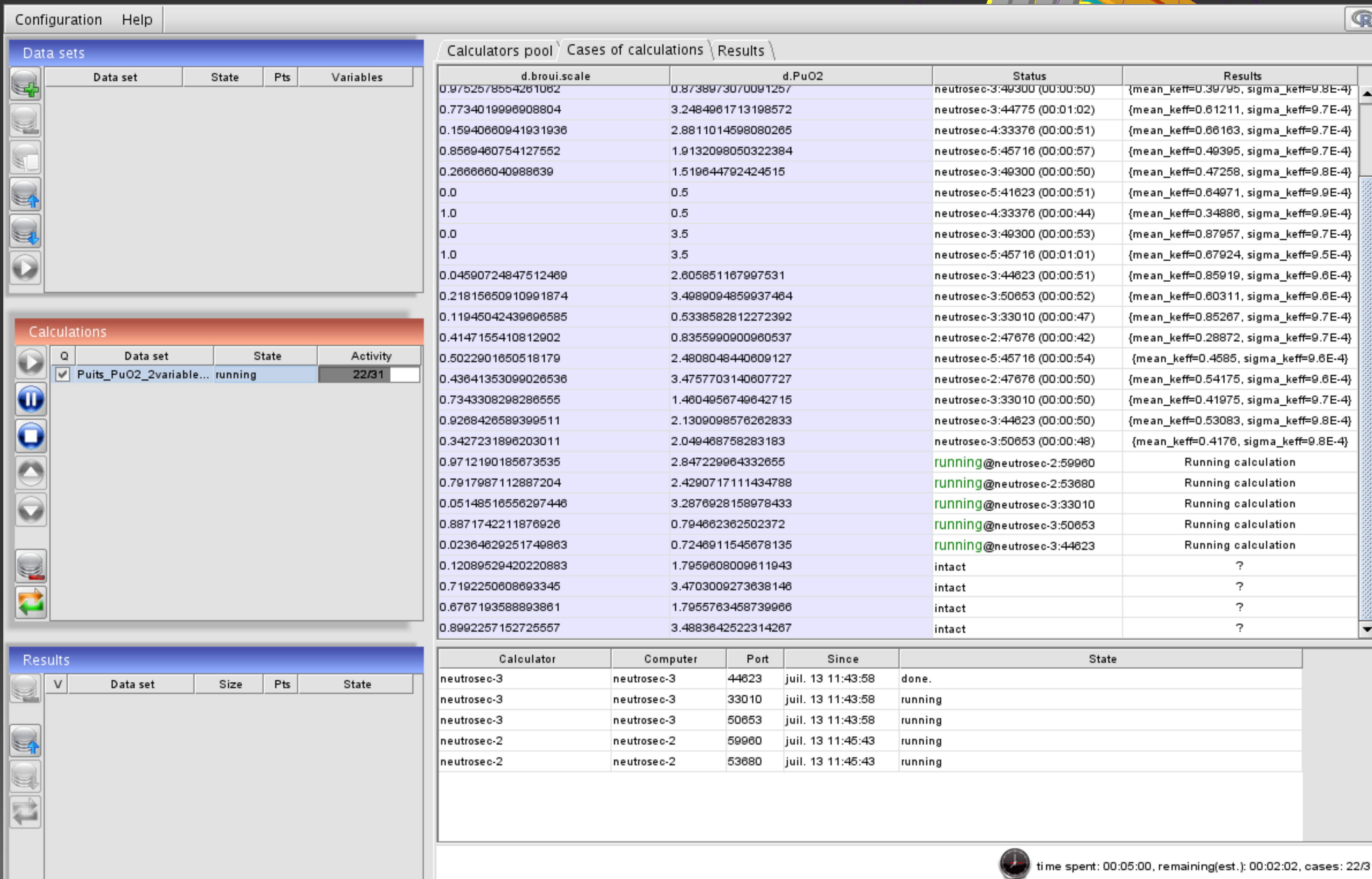
Real world example





Real world example





Configuration Help

Data sets

Data set	State	Pts	Variables

Calculations

Q	Data set	State	Activity
☑	Puits_PuO2_2variable...	running	22/31

Results

V	Data set	Size	Pts	State

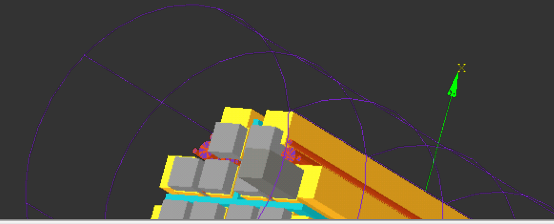
Calculators pool | Cases of calculations | Results

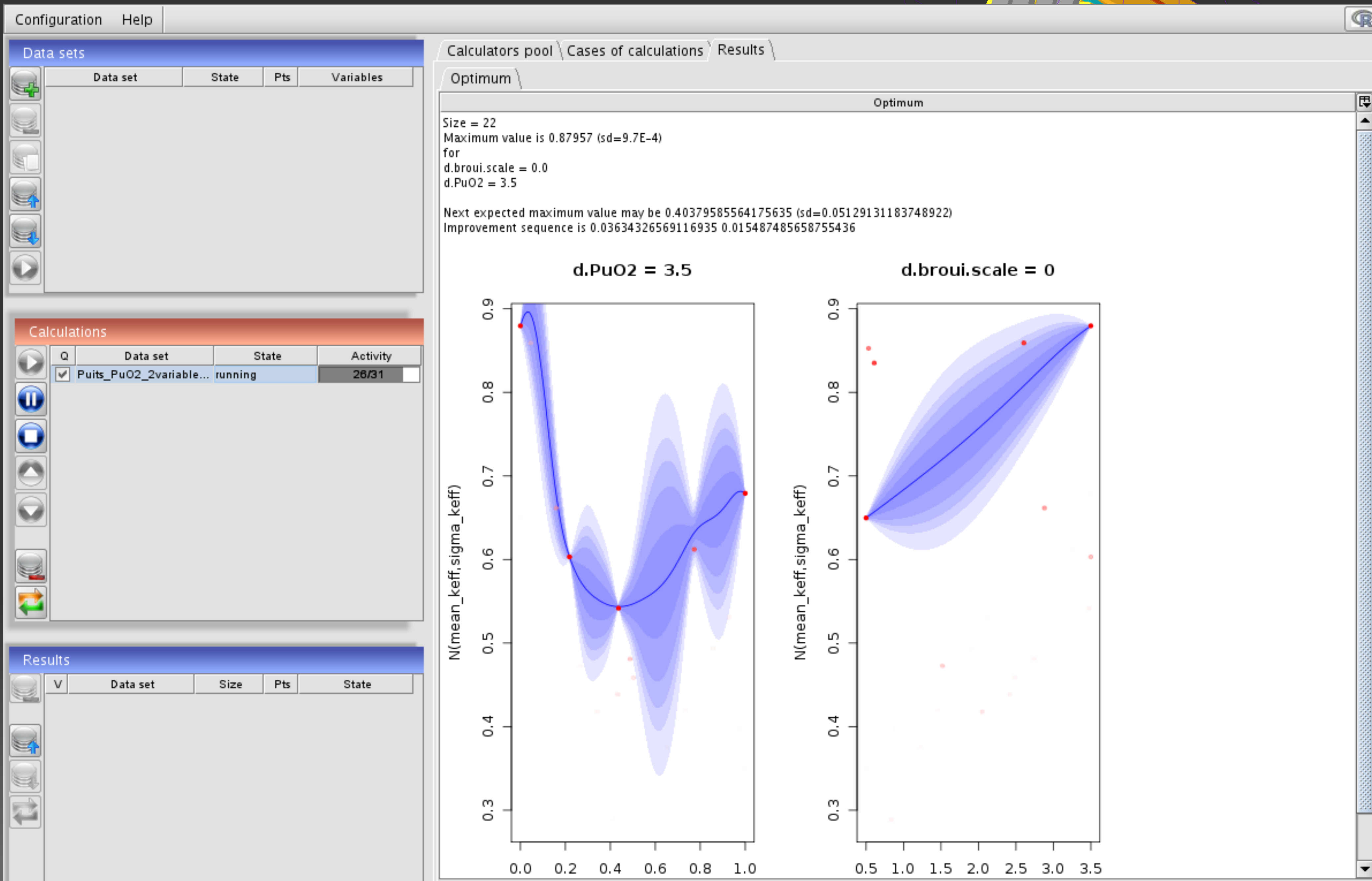
d.broui.scale	d.PuO2	Status	Results
0.9752578554261062	0.8738973070091257	neutrosec-3:49300 (00:00:50)	{mean_keff=0.39795, sigma_keff=9.8E-4}
0.7734019996908804	3.2484961713198572	neutrosec-3:44775 (00:01:02)	{mean_keff=0.61211, sigma_keff=9.7E-4}
0.15940660941931936	2.8811014598080265	neutrosec-4:33376 (00:00:51)	{mean_keff=0.66163, sigma_keff=9.7E-4}
0.8569460754127552	1.9132098050322384	neutrosec-5:45716 (00:00:57)	{mean_keff=0.49395, sigma_keff=9.7E-4}
0.266666040988639	1.519644792424515	neutrosec-3:49300 (00:00:50)	{mean_keff=0.47258, sigma_keff=9.8E-4}
0.0	0.5	neutrosec-5:41623 (00:00:51)	{mean_keff=0.64971, sigma_keff=9.9E-4}
1.0	0.5	neutrosec-4:33376 (00:00:44)	{mean_keff=0.34886, sigma_keff=9.9E-4}
0.0	3.5	neutrosec-3:49300 (00:00:53)	{mean_keff=0.87957, sigma_keff=9.7E-4}
1.0	3.5	neutrosec-5:45716 (00:01:01)	{mean_keff=0.67924, sigma_keff=9.5E-4}
0.04590724847512469	2.605851167997531	neutrosec-3:44623 (00:00:51)	{mean_keff=0.85919, sigma_keff=9.6E-4}
0.21815650910991874	3.4989094859937464	neutrosec-3:50653 (00:00:52)	{mean_keff=0.60311, sigma_keff=9.6E-4}
0.11945042439696585	0.5338582812272392	neutrosec-3:33010 (00:00:47)	{mean_keff=0.85267, sigma_keff=9.7E-4}
0.4147155410812902	0.8355990900960537	neutrosec-2:47676 (00:00:42)	{mean_keff=0.28872, sigma_keff=9.7E-4}
0.5022901650518179	2.4808048440609127	neutrosec-5:45716 (00:00:54)	{mean_keff=0.4585, sigma_keff=9.6E-4}
0.43641353099026536	3.4757703140607727	neutrosec-2:47676 (00:00:50)	{mean_keff=0.54175, sigma_keff=9.6E-4}
0.7343308298286555	1.46049567496642715	neutrosec-3:33010 (00:00:50)	{mean_keff=0.41975, sigma_keff=9.7E-4}
0.9268426589399511	2.1309098576262833	neutrosec-3:44623 (00:00:50)	{mean_keff=0.53083, sigma_keff=9.8E-4}
0.3427231896203011	2.049468758283183	neutrosec-3:50653 (00:00:48)	{mean_keff=0.4176, sigma_keff=9.8E-4}
0.9712190185673535	2.847229964332655	running@neutrosec-2:59960	Running calculation
0.7917987112887204	2.4290717111434788	running@neutrosec-2:53680	Running calculation
0.05148516556297446	3.2876928158978433	running@neutrosec-3:33010	Running calculation
0.8871742211876926	0.794662362502372	running@neutrosec-3:50653	Running calculation
0.02364629251749863	0.7246911545678135	running@neutrosec-3:44623	Running calculation
0.12089529420220883	1.7959608009611943	intact	?
0.7192250608693345	3.4703009273638146	intact	?
0.6767193588893861	1.7955763458739966	intact	?
0.8992257152725557	3.4883642522314267	intact	?

Calculator	Computer	Port	Since	State
neutrosec-3	neutrosec-3	44623	juil. 13 11:43:58	done.
neutrosec-3	neutrosec-3	33010	juil. 13 11:43:58	running
neutrosec-3	neutrosec-3	50653	juil. 13 11:43:58	running
neutrosec-2	neutrosec-2	59960	juil. 13 11:45:43	running
neutrosec-2	neutrosec-2	53680	juil. 13 11:45:43	running

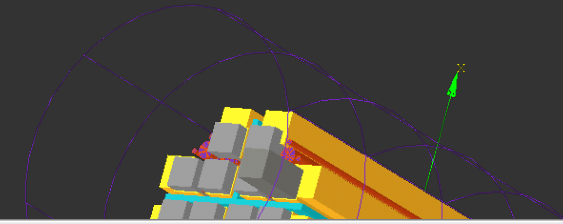
time spent: 00:05:00, remaining(est.): 00:02:02, cases: 22/31

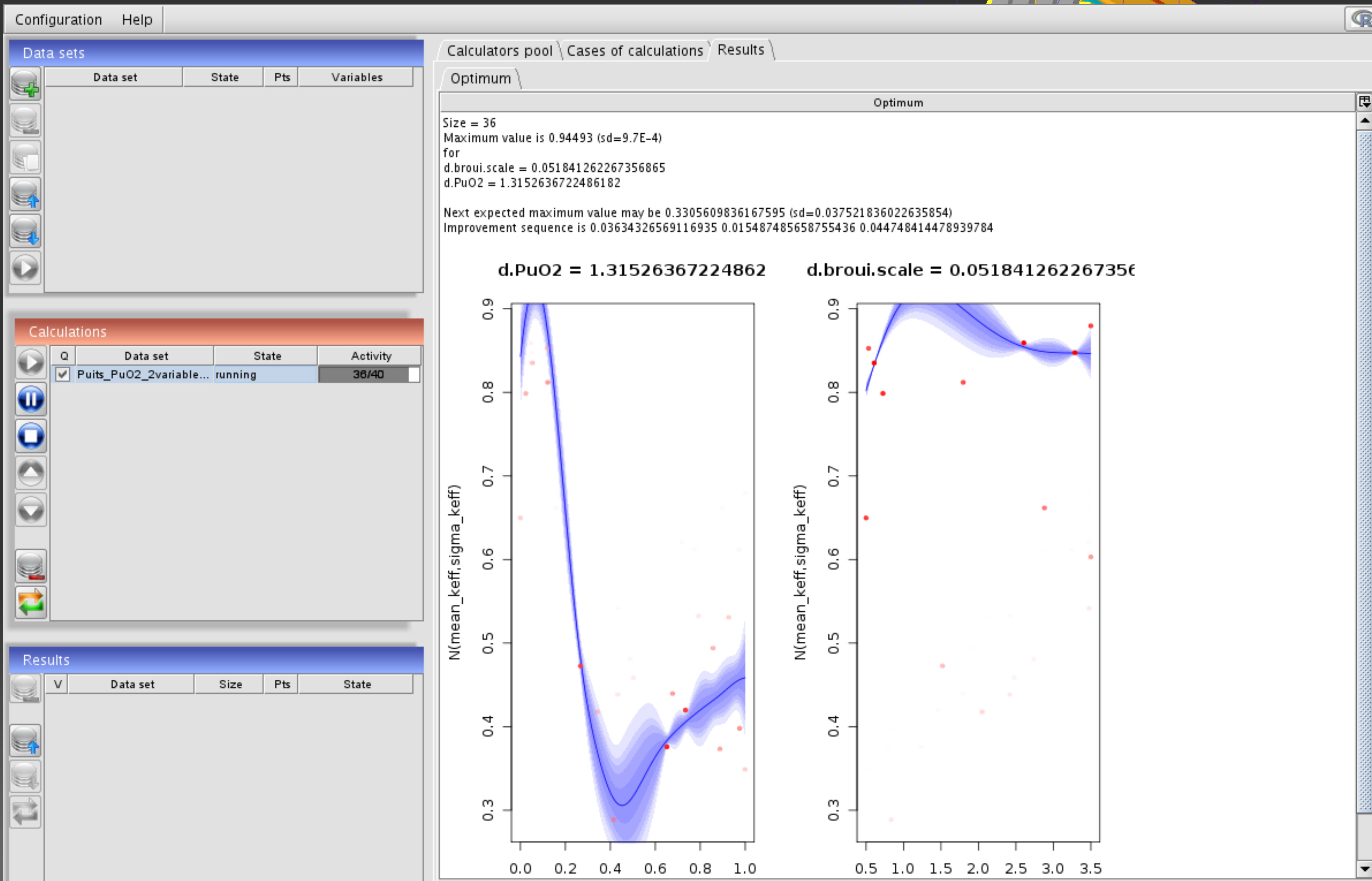
Real world example



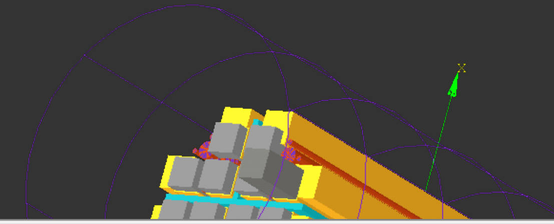


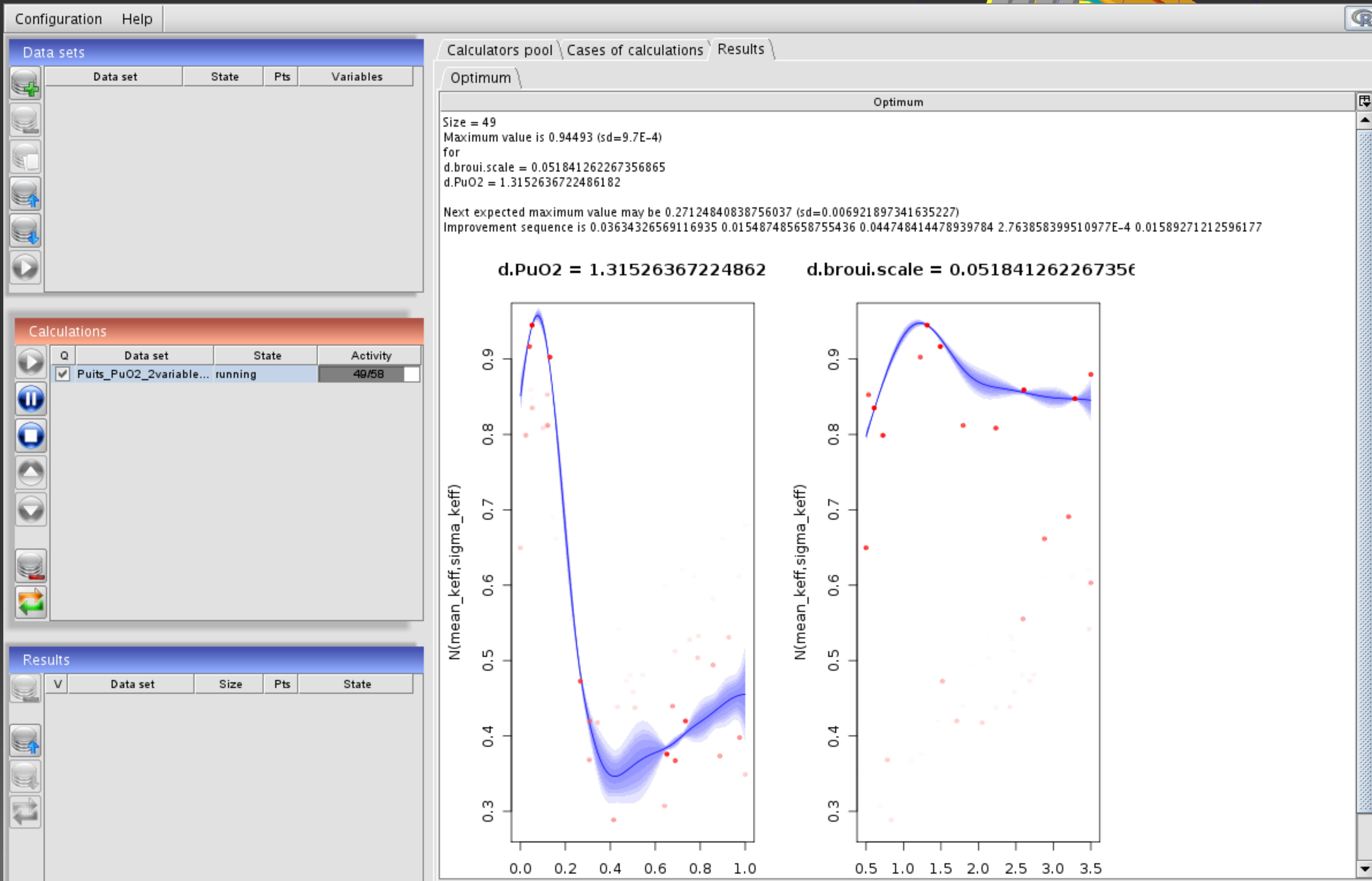
Real world example



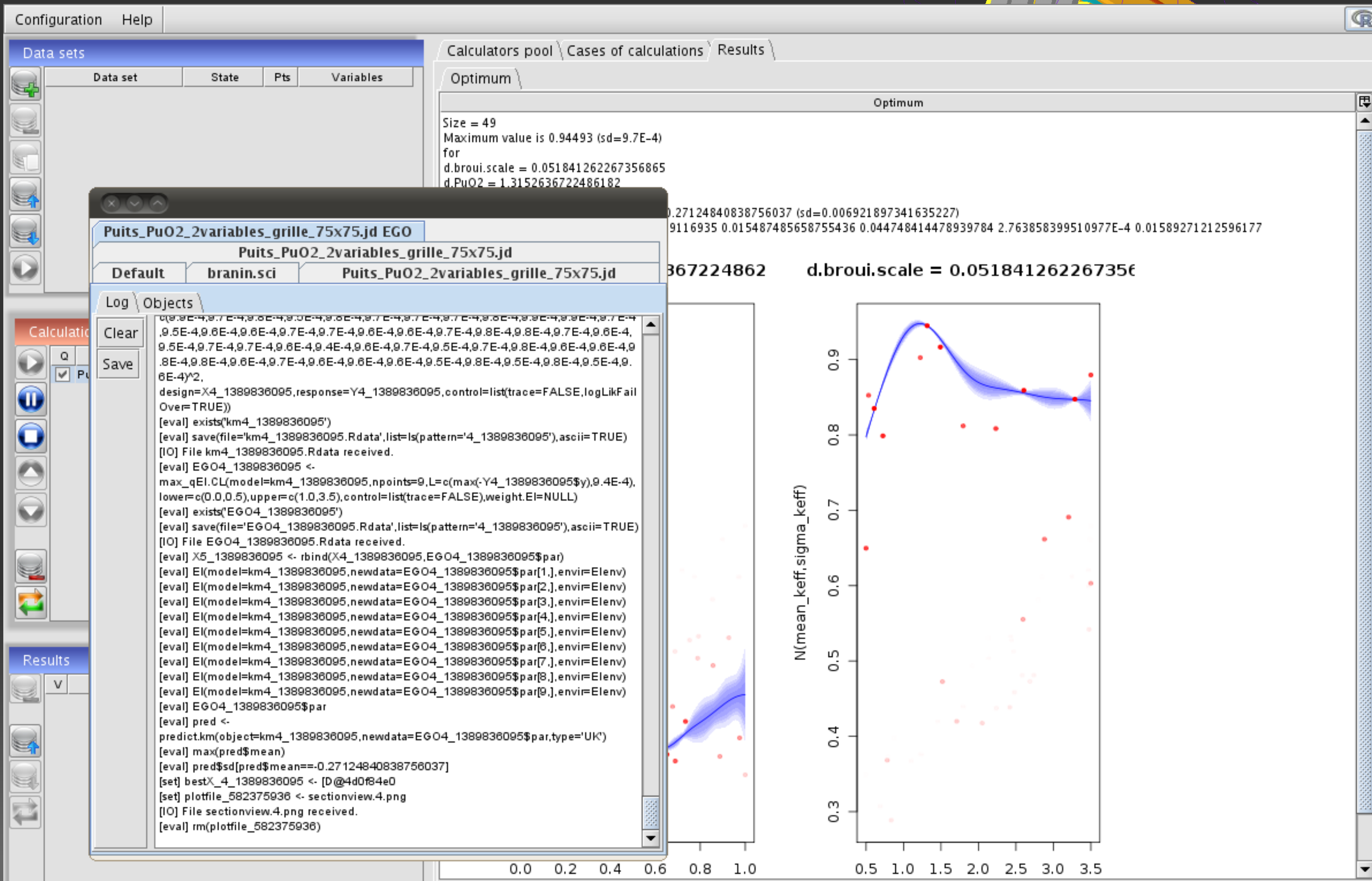


Real world example

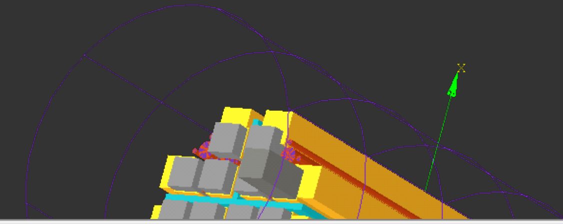


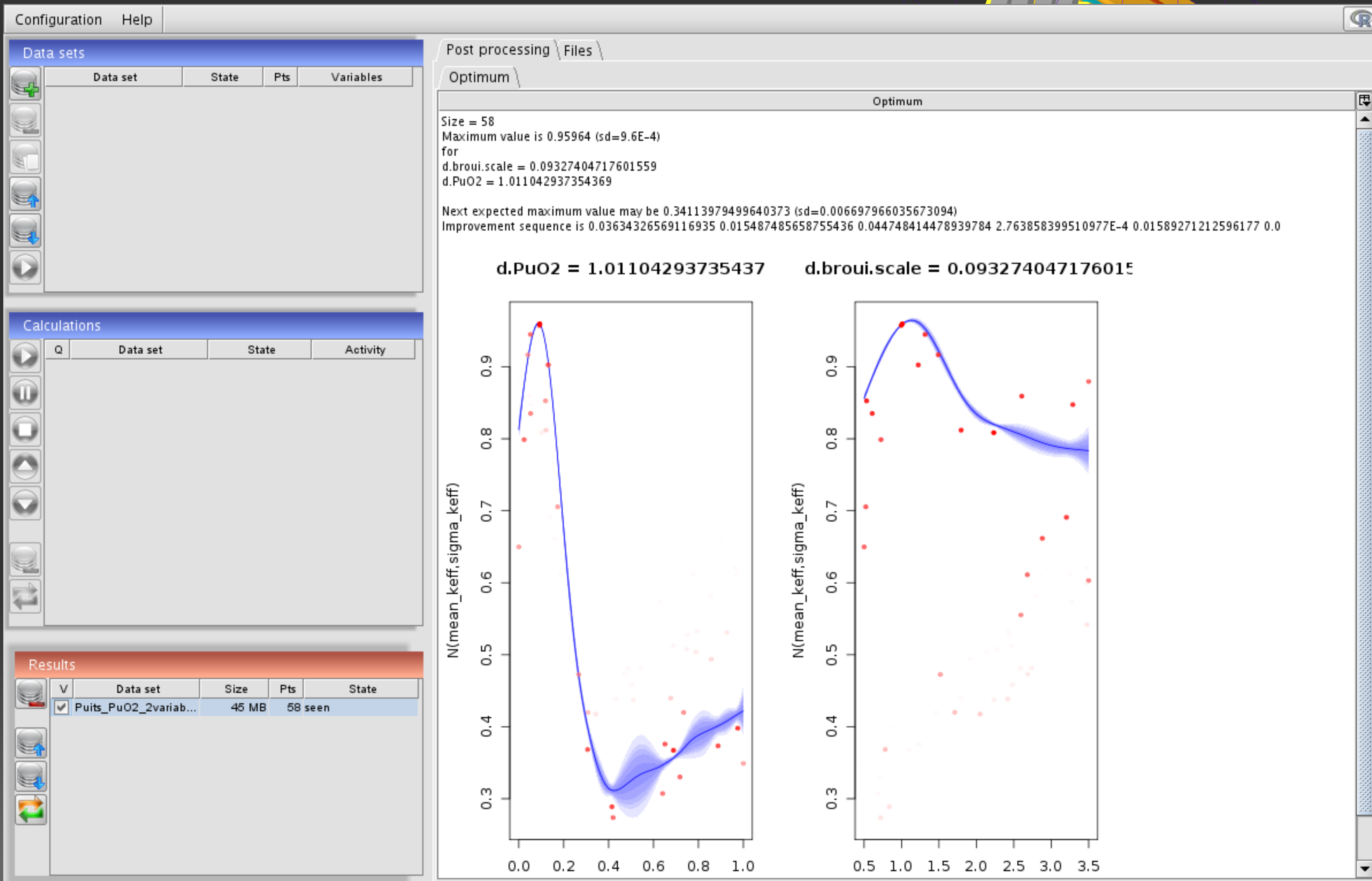


Real world example

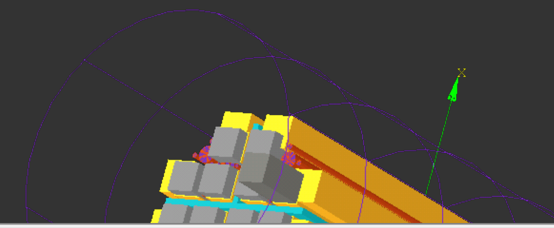


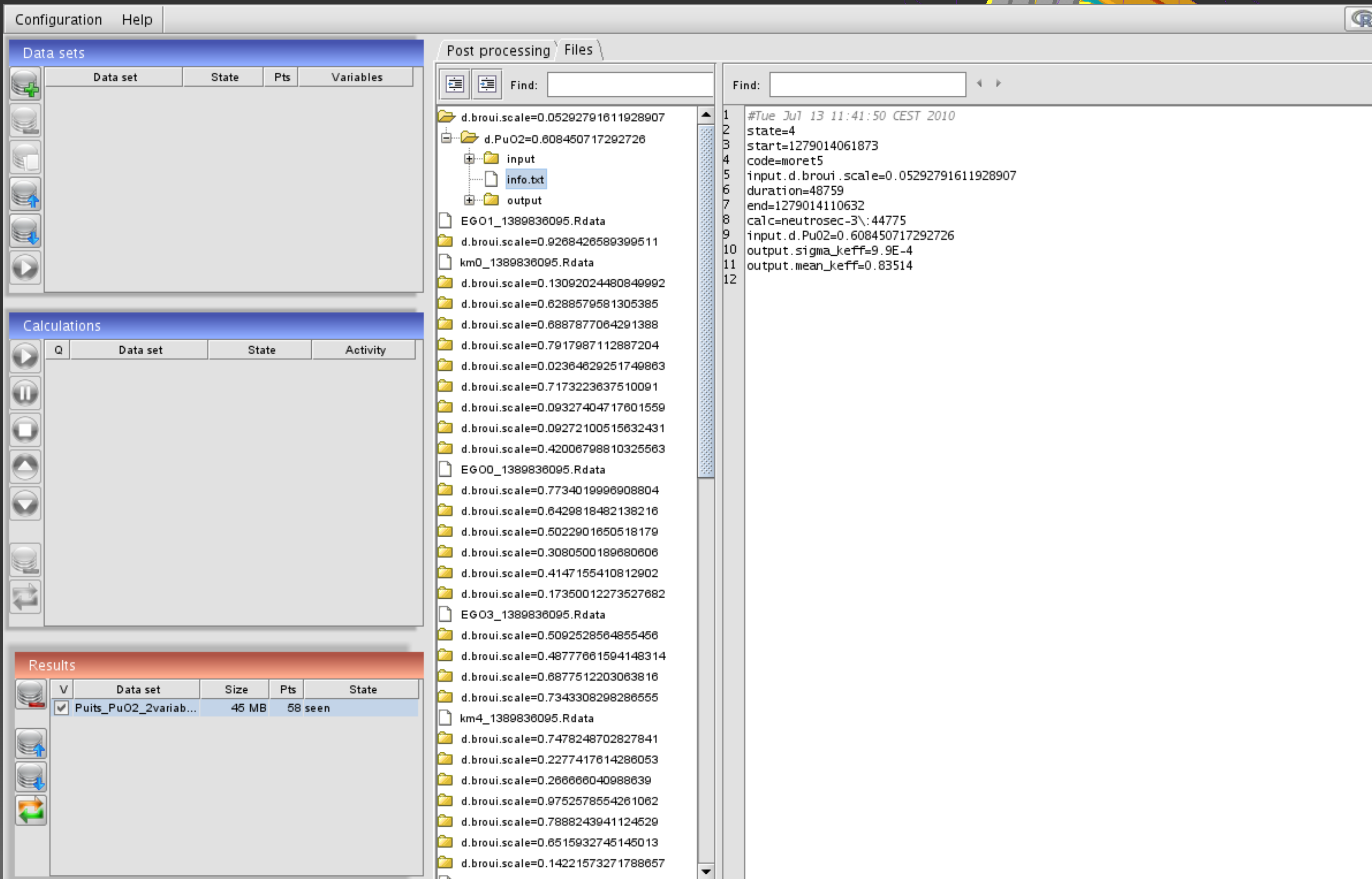
Real world example



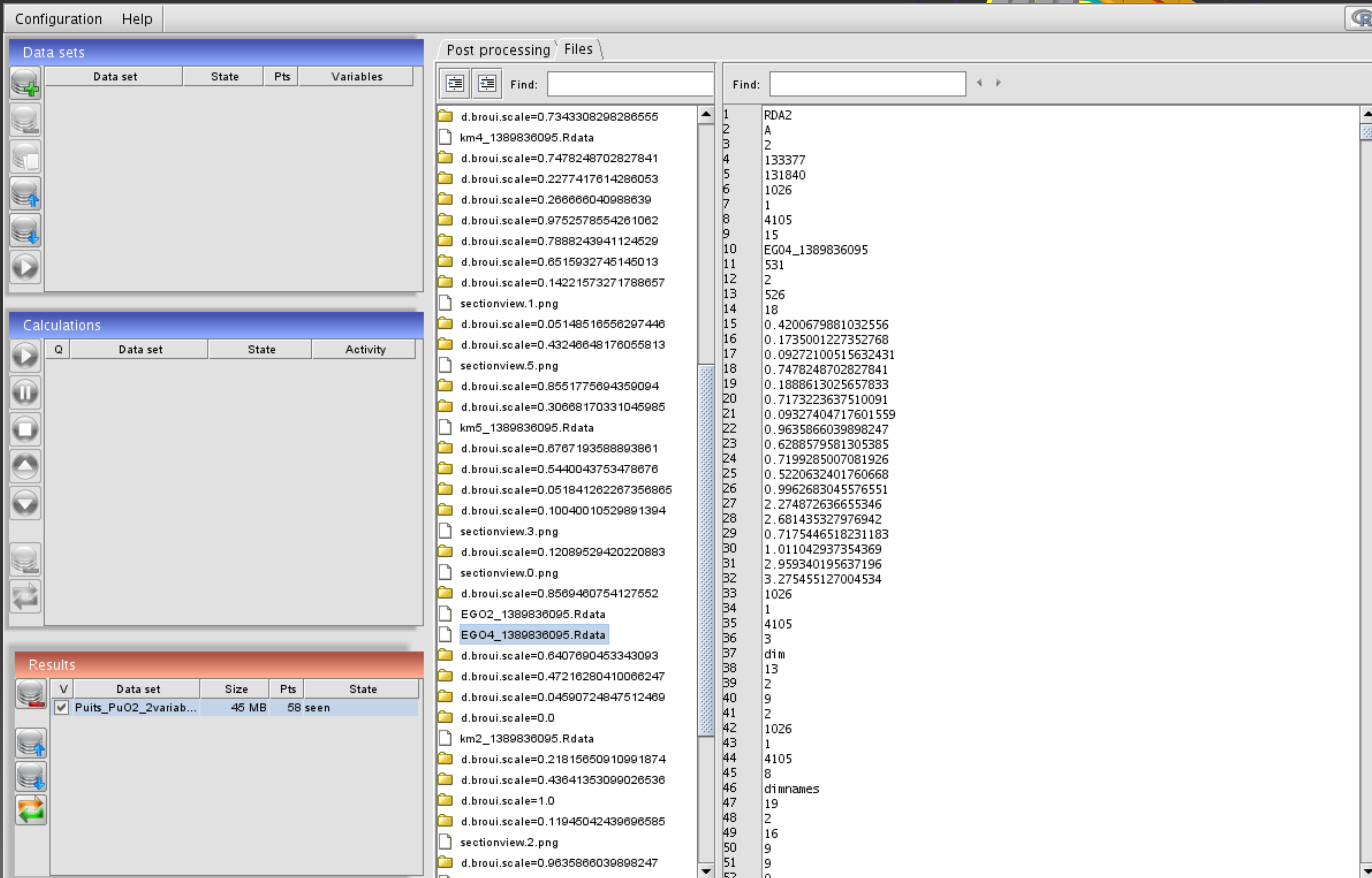


Real world example

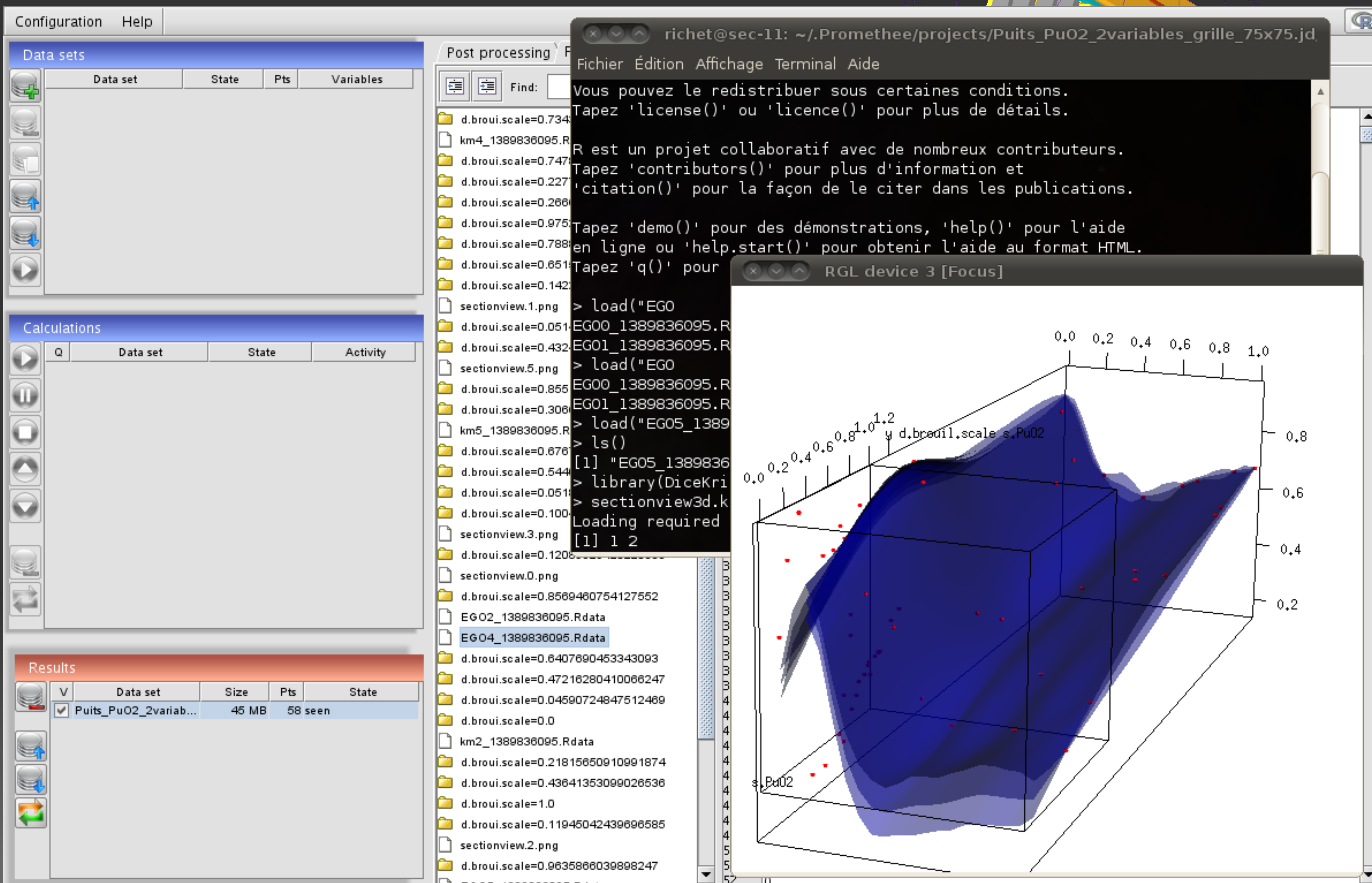




Real world example



Real world example



Overview

Few words about Research and Industry

Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Overview

Few words about Research and Industry

Computer Experiments framework

PROMETHEE Grid Computing environment

Real world example

Summary

Summary

Industry benefits: a five years leap

- Better/stronger day-to-day eng. conclusions
- Face new challenges: harder eng. issues *<u>now</u>* reachable
- New abstract & formalized approach of old engineering practices

Research support

- Lot of new industrial applications
- Lot of feedback on algorithms, underlying hypothesis, *ideas*
- New [R] users ...

... and a bit of (wholesome) money :)

Summary

... thanks to integration of

a <u>flexible</u> (technology, license & community)

research software: [R]

a <u>disruptive</u> (re-think *true* needs, use *true* resources)

industrial software: PROMETHEE

... available for free at http://promethee.irsn.fr