

TUG “standalone” benchmarks (and some hopefully useful metrics)

1 Benchmark description

All benchmarks are specified and can be modified (e.g., number of iterations) in the `eval/bench_defs.hpp` file.

1.1 barite_200

At a glance:

Grid	200x200
Size	1x1 m ²
Timestep	1000 s
Iterations	50
α_x, α_y	heter., aniso.
Species #	7
Init	homog.

Table 1: Summary of parameters for the barite_200 benchmark

Initial Conditions (IC): all concentrations are initially homogeneous, refer to the **IC** values in table 2. The actual numerical values read by the benchmark from file `eval/barite_200/barite_200_init.csv` have higher significant digits.

Boundary conditions (BC): the top left corner (first 5 element boundaries in both N and W sides) are set to constant value of 0.1 molal BaCl₂. All other boundaries are closed.

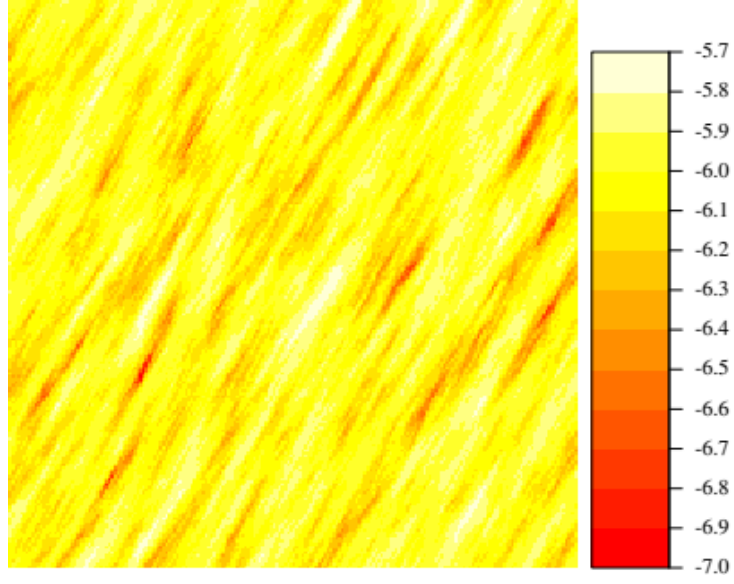
	H	O	Charge	Ba	Cl	S.6_	Sr
IC	110.0124	55.5086	-1.2163e-09	4.4553e-07	2.0e-12	6.1516e-5	6.1472e-5
BC	111.0124	55.5062	-3.3370e-08	0.1	0.2	0	0

Table 2: Initial and boundary values of all transported variables in the barite_200 benchmark.

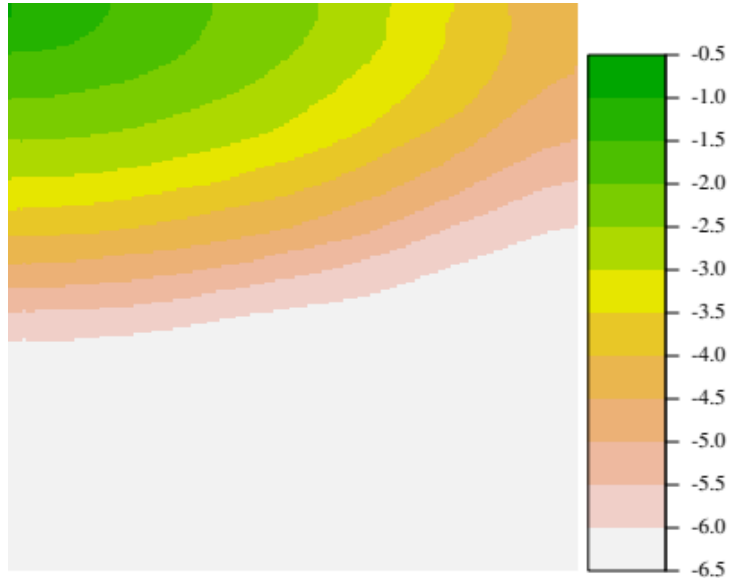
Spatially heterogeneous values for α_x and α_y are read from `eval/barite_200/alpha_[xy].csv`. They result from a single geostatistical simulation $\mathcal{F}$ of a $\mathcal{N}(0, 1)$ variable (anisotropic spherical variogram of correlation length 5 at -30° and 20 at 60°, and sill 1). This $\mathcal{F}$ field was scaled with an approximate order of magnitude for α_x and α_y respectively:

$$\begin{cases} \alpha_x &= 10^{-7} + 10^{-6} \frac{\mathcal{F} - \min(\mathcal{F})}{\max(\mathcal{F})} \\ \alpha_y &= 10^{-7} + 10^{-7} \frac{\mathcal{F} - \min(\mathcal{F})}{\max(\mathcal{F})} \end{cases}$$

Figure 1a displays the picture for α_x and 1b the results for Ba after the 50 iterations in the benchmark. This benchmark runs in ~ 1.8 s on 18 CPUs on a System @ PERFACCT.



(a) α_x field



(b) $\log_{10}\text{Ba}$ after 50 iterations

Figure 1: Diffusivity field and endresult for Ba in the `barite_200` benchmark

1.2 barite_large

Larger grid version of the **barite_200** benchmark, this time with heterogeneous initial conditions, closed boundaries everywhere and homogeneous diffusion coefficients. The sense of this benchmark is to check for mass conservation.

At a glance:

Grid	1000x1000
Size	10x10 m
Timestep	100 s
Iterations	50
α	homog. 1E-6
Species #	7
Init	heter.

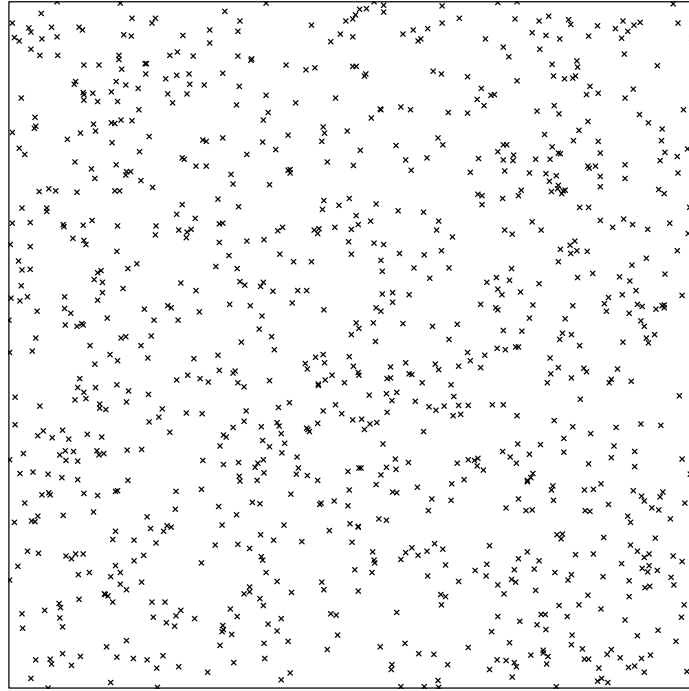
Table 3: Summary of parameters for the **barite_large** benchmark

Boundary conditions (BC): all boundaries are closed. As for initial conditions, background concentrations are set in the whole grid (**All** record in table 4). 1000 randomly selected grid cells (cfr figure 2a for their position) are assigned initial values of the **Locations** record. The non-rounded values are read from file `eval/barite_large/barite_large_init.csv`.

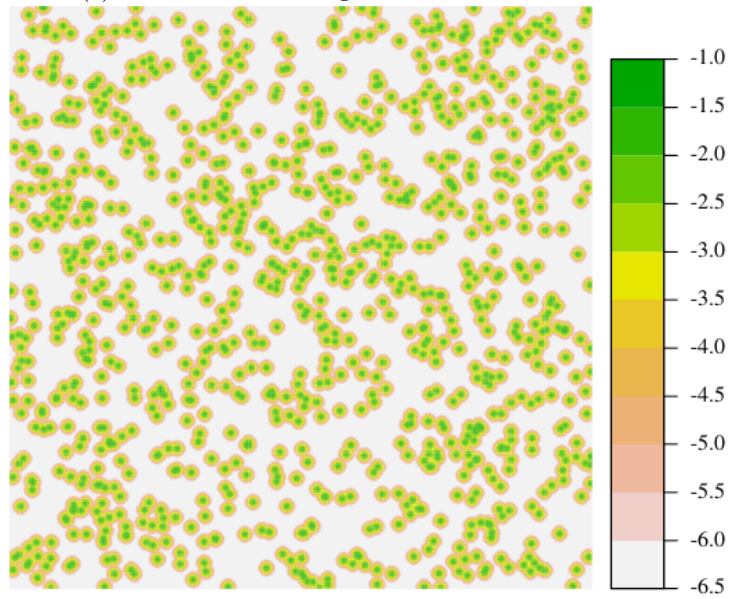
	H	O	Charge	Ba	Cl	S_6_	Sr
All	110.0124	55.50868	-1.2163e-09	4.4553e-07	0	0.0006152	0.00061
Locations	111.0124	55.50622	-3.0000e-07	1	2	0.01	0.001

Table 4: **Barite_large** benchmark, initial conditions: the whole grid has the values in the **All** record, while 1000 cells as displayed in figure 2a are assigned the values of **Locations** record.

This benchmark runs in ~ 6.4 s on my desktop using 18 CPUs.



(a) Locations of “heterogeneities” in initial conditions



(b) $\log_{10} \text{Ba}$ after 5 iterations

Figure 2: Diffusivity field and endresult for Ba in the `barite_200` benchmark

1.3 surfex

Homogeneous benchmark with values inspired from POET’s **surfex**, transporting 21 species. Here we use actual physically true values for α (isotropic and homogeneous set to 1.1E-12) and simulate a rectangular domain of 2x1 cm². At a glance:

Grid	200x100
Size	0.02x0.01 m
Timestep	3600 s (1 h)
Iterations	20
α	homog. 1.1E-12
Species #	21
Init	homog.

Table 5: Summary of parameters for the **surfex** benchmark

Boundary conditions (BC): all domain boundaries are set to constant **BC** values. **Initial conditions (IC):** homogeneous, cfr table 6.

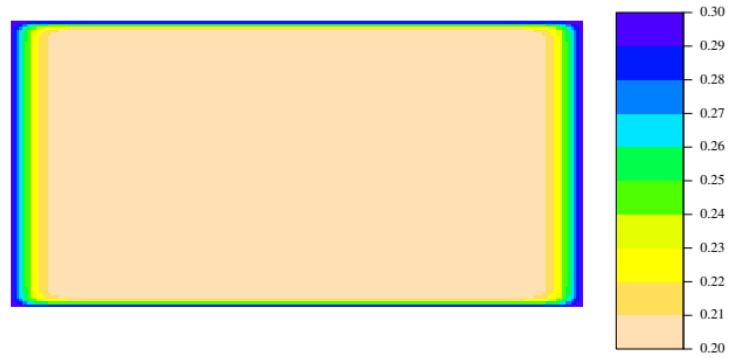
	IC	BC
H	1.11e+02	120.0
O	5.55e+01	55.1
Charge	-2.0e-13	8.0e-17
C	2.0e-16	2.0e-15
CH4	2.0e-03	0.2
Ca	2.0e-01	0.03
Cl	3.0e-01	0.5
Fe2	1.4e-04	0.0002
Fe3	1.3e-09	2.0e-08
H0	6.0e-12	2.0e-11
K	2.0e-03	1.0e-05
Mg	1.0e-02	0.2
Na	2.0e-01	0.3
HS2	5.9e-10	0
S2	8.3e-15	8.3e-12
S4	2.1e-14	5.1e-14
S6	1.6e-02	0.026
Sr	4.5e-04	0.045
U4	2.5e-09	2.5e-08
U5	1.6e-10	1.6e-10
U6	2.3e-07	1.0e-05

Table 6: **surfex** benchmark, homogeneous initial conditions **IC** and boundary values **BC**

This benchmark runs in ~ 1.1 s on my desktop using 18 CPUs.



(a) $\log_{10}(U6)$ after 20 iterations



(b) Na (linear scale) after 5 iterations

Figure 3: Results for U6 and Na in the **surfex** benchmark

2 Some hopefully useful metrics

The problem with the *classical, central* measures such as MAE (Mean Absolute Error) or RMSE (Root Mean Square Error), even when scaled using a constant (e.g., the range of true variable, its mean, or standard deviation) is that the measure does underestimate discrepancies for very small values and conversely is only sensible to large values. In practice these measures only work if the distribution of the variable of interest is uniform on a small range or symmetric distributed, e.g., gaussian. However, many variables we deal with in geochemical models are however rather *lognormally* distributed or *uniformly distributed on a logarithmic scale*. In any case, it is common that variables we try to regress or match with models span many orders of magnitude, and the error measures defined above are biased towards large values.

This problem can be partially solved operating on logarithms of the variables. Plugging the logarithms of the true and predicted variables y and $\hat{y}$ in the usual MAE and RMSE formula we get, **assuming both variables strictly larger than 0**:

$$\text{MAE}_{\log} = \frac{1}{N} \sum |\log y_i - \log \hat{y}_i| = \frac{1}{N} \sum \left| \log \frac{y_i}{\hat{y}_i} \right| \quad (1)$$

$$\text{RMSE}_{\log} = \sqrt{\frac{1}{N} \sum (\log y_i - \log \hat{y}_i)^2} = \sqrt{\frac{1}{N} \sum \left(\log \frac{y_i}{\hat{y}_i} \right)^2} \quad (2)$$

It is usual to define a slightly different variant of $\text{RMSE}_{\log}$, called RMSLE (Root Mean Square Logarithmic Error) by adding 1 to both the predicted and the true value to avoid dividing by 0:

$$\text{RMSLE} = \sqrt{\frac{1}{N} \sum [\log(y_i + 1) - \log(\hat{y}_i + 1)]^2} = \sqrt{\frac{1}{N} \sum \left(\log \frac{y_i + 1}{\hat{y}_i + 1} \right)^2} \quad (3)$$

All these measures yield 0 if y and $\hat{y}$ are identical. Note that the $\text{MAE}_{\log}$ resembles a *geometric mean of the absolute values of the quotients* of y and $\hat{y}$ per observation (putting either y or $\hat{y}$ at denominator is equivalent):

$$\text{Geometric Mean of Absolute Quotients} = \left(\prod \left| \frac{\hat{y}_i}{y_i} \right| \right)^{\frac{1}{N}} \quad (4)$$

The geometric mean of the quotients would be 1 if the two variables are identical. The connection with the $\text{MAE}_{\log}$ is easy to see since by definition a geometric mean is the N -th root of the product of the terms:

$$\exp \left[\frac{1}{N} \sum \log a_i \right] = \left(\prod a_i \right)^{\frac{1}{N}} \quad (5)$$

So the $\text{MAE}_{\log}$ is the logarithm of the actual geometric mean of the (absolute) logarithms of the quotients between the variables. Instead of the simple quotient of the true and predicted values y and $\hat{y}$ we can define a relative (**per observation**) error α_i as:

$$\alpha_i = \begin{cases} \frac{y_i - \hat{y}_i}{y_i} & \text{if } y_i, \hat{y}_i \neq 0 \\ 1 & \text{if } y_i = 0 \text{ and } \hat{y}_i \neq 0 \\ 0 & \text{if } y_i = 0 \text{ and } \hat{y}_i = 0 \end{cases} \quad (6)$$

The same treatment of the case when a variable is 0 of course also applies to the above introduced relative measures. Without using logarithms we can then define some relative measures such as the Mean Absolute Percentage Error (**MAPE**) and Relative RMSE (**RRMSE**):

$$\text{MAPE} = \frac{100\%}{N} \sum |\alpha_i| \quad (7)$$

$$\text{RRMSE} = \sqrt{\frac{1}{N} \sum (\alpha_i)^2} \quad (8)$$

These relative measures account for discrepancies across all magnitudes of the y and $\hat{y}$ variables while preserving the physical meaning of 0. An implementation of all these metrics in R is given in the **Metrics.R** file in this same directory.