DoE for Scale-Up I

… using MODDE® and DoE-DiVa®

Session 1: Introduction to the Concepts -- 18.01. 2023

Prof. Dr. Andreas Orth

umesoft

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

Everyone knows ***MODDE®***, …

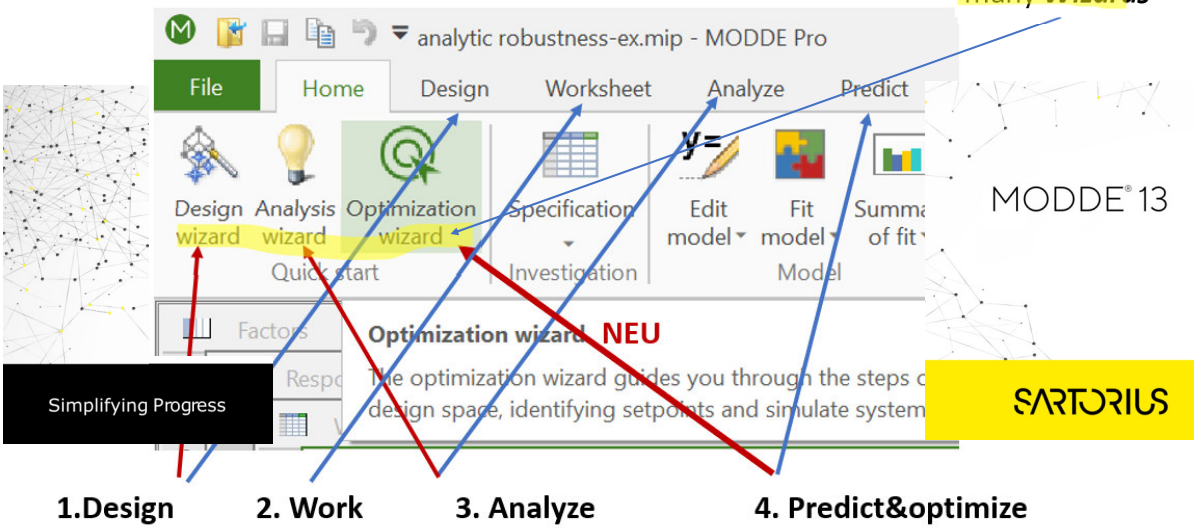
- ***MODDE®*** is *the* State-of-the-Art user-friendly Software for ***D****esign* ***o****f* ***E****xperiments* (***DoE***).
- ***MODDE®*** provides the complete ***D****esign* ***o****f* ***E****xperiments* workflow: *Design, Analyze* and *Predict*
- ***MODDE®*** is compatible with regulatory requirements from *american FDA, european EMA* and *japanese PMDA.*

..., but what's ***DoE-DiVa***?

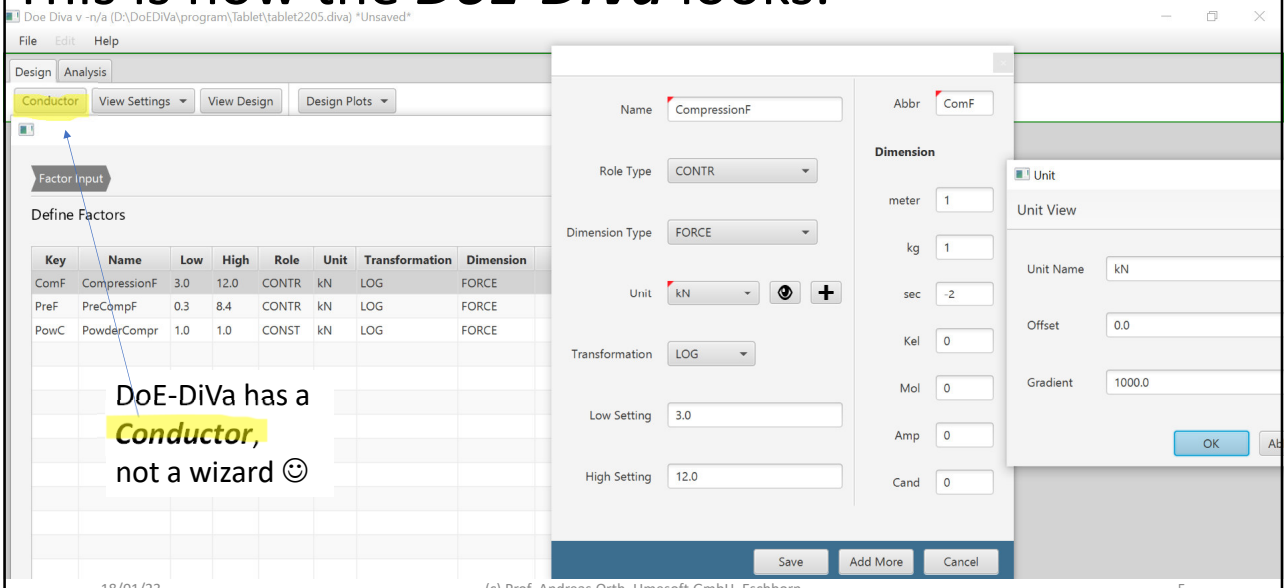
- ***DoE-DiVa*** is a user-friendly Software for Engineers and Scientists in R&D, developed in JAVA by umesoft
- ***DoE-DiVa*** *enhances* ***D****esign* ***o****f* ***E****xperiments and makes it more intelligent*
- ***DoE-DiVa*** *integrates Similarity Theory for* ***Di****mensionless* ***Va****riables and provides DoE's for* ***Scale-Up*** *and* ***Scale-Down***

This is how ***MODDE®*** looks.

MODDE® has many ***Wizards***

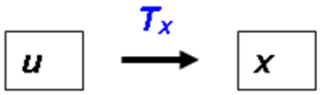


This is how the ***DoE-DiVa*** looks.

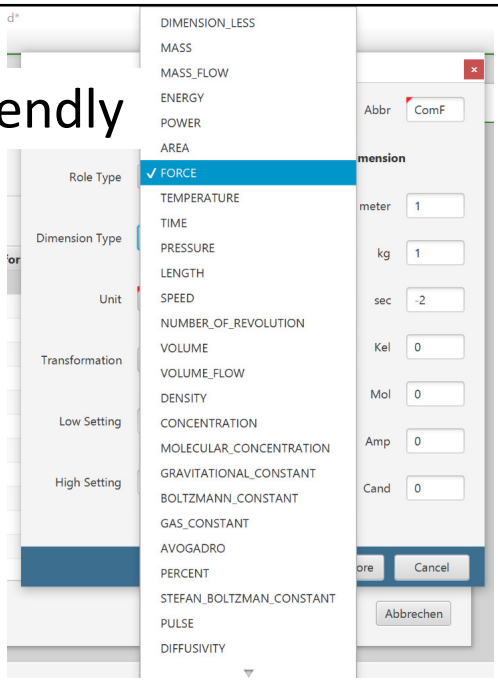


DoE-DiVa's conductor is friendly

- ***DoE-DiVa*** differentiates between
- ***User-factors and eXplaining-factors***

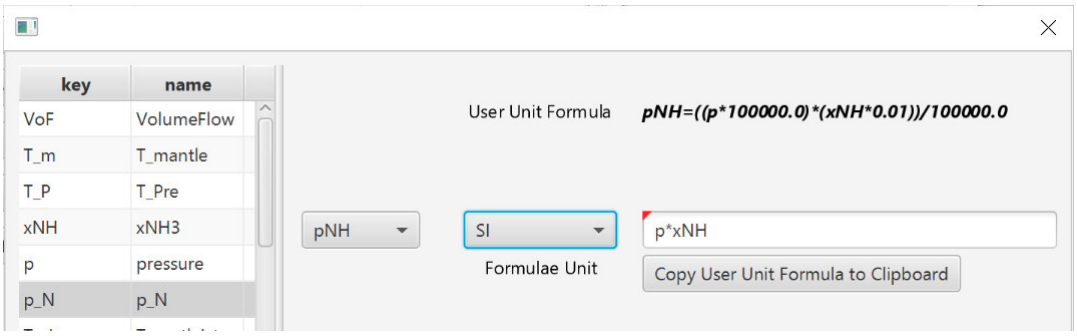


- ***DoE-DiVa*** let's the user choose his ***Dimensions, Units, Transforms*** *and* ***Scaling*** and carries them through all the User-Software work session



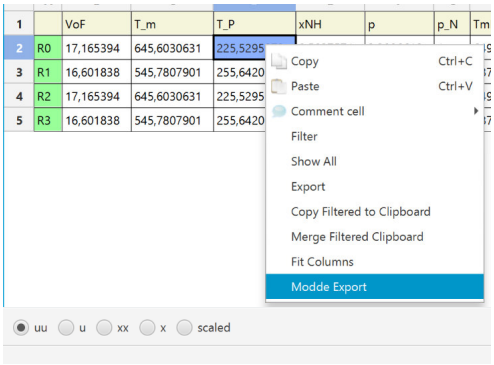
DoE-DiVa's conductor is flexible

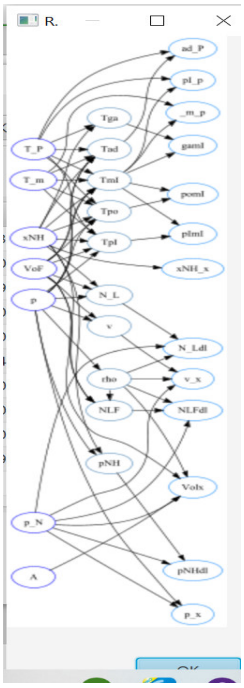
- ***DoE-DiVa*** also accepts *dependent U-factors* as formulae, data tables (from simulation results or VDI-Waermeatlas) or known coefficients (from gas law for example)



DoE-DiVa's conductor is informative

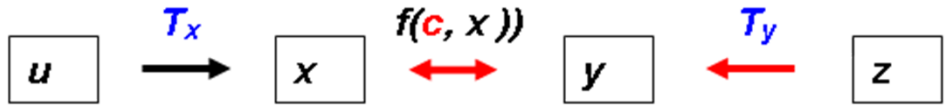
- ***DoE-DiVa*** generates ***factor-dependency plot*** *and ok-for-modelling plot, as well as classical interpretative and diagnostic plots*
- ***DoE-DiVa*** has a simple data/formula interface to MODDE® for ***process optimization*** and ***QbD (Quality by Design)***.





1. ***eXplaining factors vs. User factors***
2. Using DoE-DiVa for for an easy example
3. Future Plans and Ideas

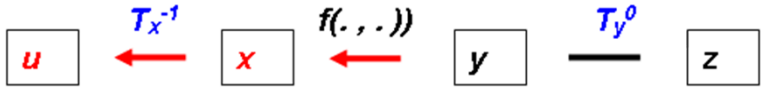
The ***DoE-DiVa***-approach



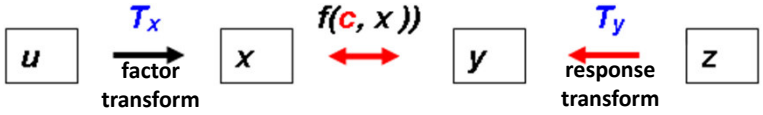
u: ***U****ser-factor,* to be set in the experiment, e.g. *Temp, pressure etc.*
x: *e****X****plaining-factor,* to be used in the model, e.g. *a force-ratio*
T_x***:*** *transformation* to get from ***u*** to ***x***, e.g. *ratio, dimensionless variable*
c: ***c****oefficients or parameters in the model,* ***f****, to be determined by model FIT*
z: *measured response value*
y: *transformed response value, e.g. ratio or product of a* ***z*** *and some* ***u***
T_y***:*** *transformation* to get from ***z*** to ***y***, *may also just be log or neg-log*

The role of the *Transformations*

Design: How to get *u* from *f* and the *tr*ansformation



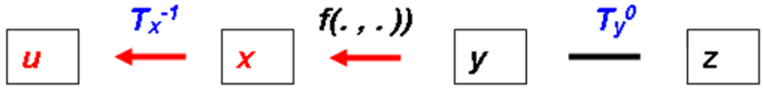
Model Fit: how to get coefficients, *c*, from *x* and *y*



Optimization: how to get the *u* from the specification for *z*

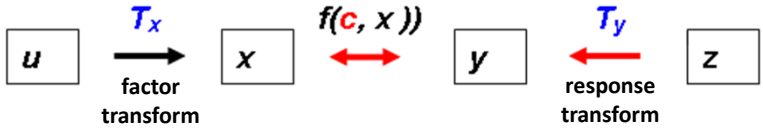
The *Approximation* Trick to get T_x^{-1}

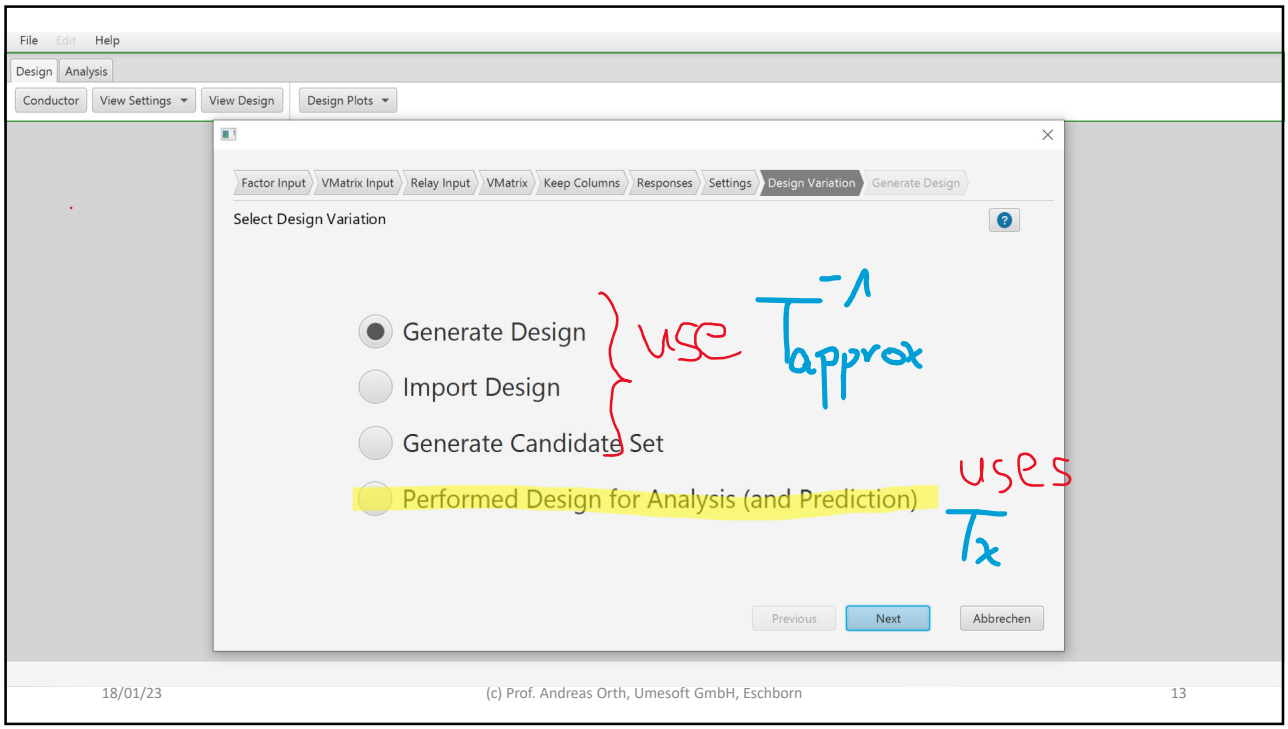
Design: To get u from *x*, T_x has to be inverted



Inverting a non-linear T_x may be difficult or impossible, so T_x is approximated by a linear T_{approx} and ***when we write*** T_x^{-1} ***we mean*** T_{approx}^{-1}

Model Fit: To get *x* from *u* for Fit, T_x need not be inverted, T_{approx} is not needed.





DoE-DiVa has some under cover mathematics

- ***DoE-DiVa solves the PI-theorem*** to find ***Di****mensionless* ***Va****riables* to use as ***X***-factors and lets the user adjust and edit these
- ***DoE-DiVa*** approximates and inverts the ***X****-factors-from-****U****-factors transform in order to recover* ***U****-factor settings from an* ***X****-factor design.*
- ***DoE-DiVa*** solves Karush-Kuhn-Tucker conditions to find inner and outer limits for ***X****-factors* from given user-limits for ***U****-factors*
- ***Generates optimal candidate sets*** for ***U****-constrained D-optimal* ***X****-designs* to satisfy equality and inequality contraints.

1. e**X**plaining factors vs. **U**ser factors
2. ***Using DoE-DiVa for an easy example***
3. Future Plans and Ideas

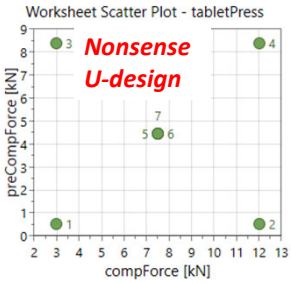
u_1 = compression force, 3 to 12 kN
u_2 = pre-compression force, 0.3 to 8.4 kN

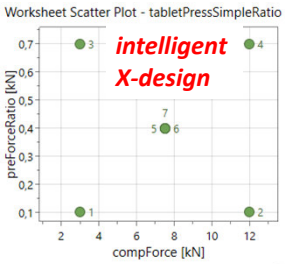
x_1 = compression force, 3 to 12 kN
x_2 = pre-force-ratio , 0.1 to 0,7

In modern presses, there is a pre-compression step where both punches are forced by pre-compression rollers to press the powder in the die into a tablet.

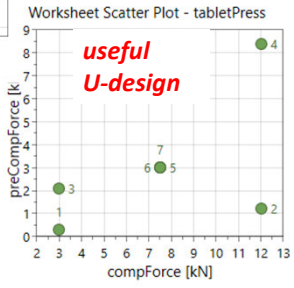
It makes no sense to set the pre-compression force already higher than the main compression force.

Easy example: Tablet pressing





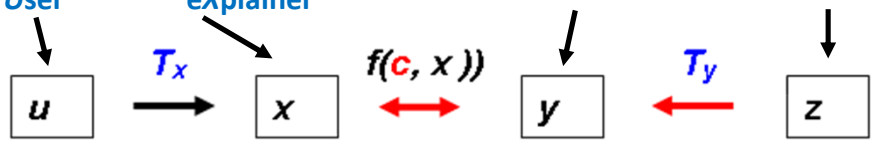
T_x^{-1}



Technical and explaining Factors

User *eXplainer*

Transformed and raw and Responses



u_1 = compression force
u_2 = pre-compression force
u_3 = compressibility (const)

x_1 = compression force,
x_2 = *pre-force-ratio*

$y_1 = z_1$
$y_2 = z_2$
$y_3 = z_3$
$y_4 = z_4 / u_1$

z_1 Medicament release
z_2 Dissolution, %
z_3 Friability, %
z_4 Hardness, kN

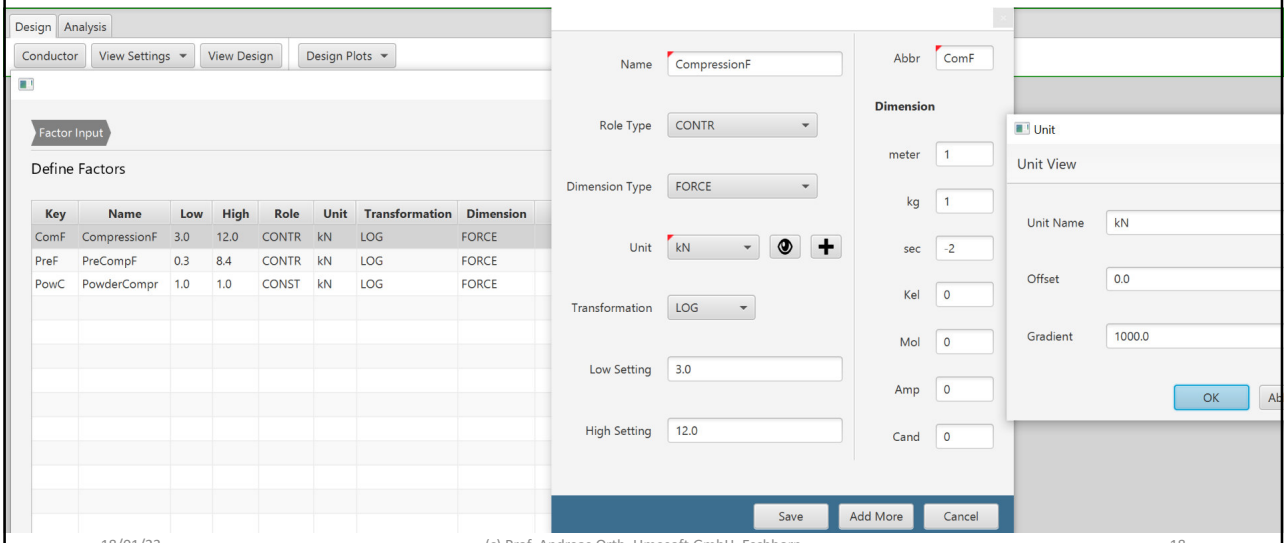
Easy example: Tablet pressing

T_x as a Exponent V-Matrix (for factors)

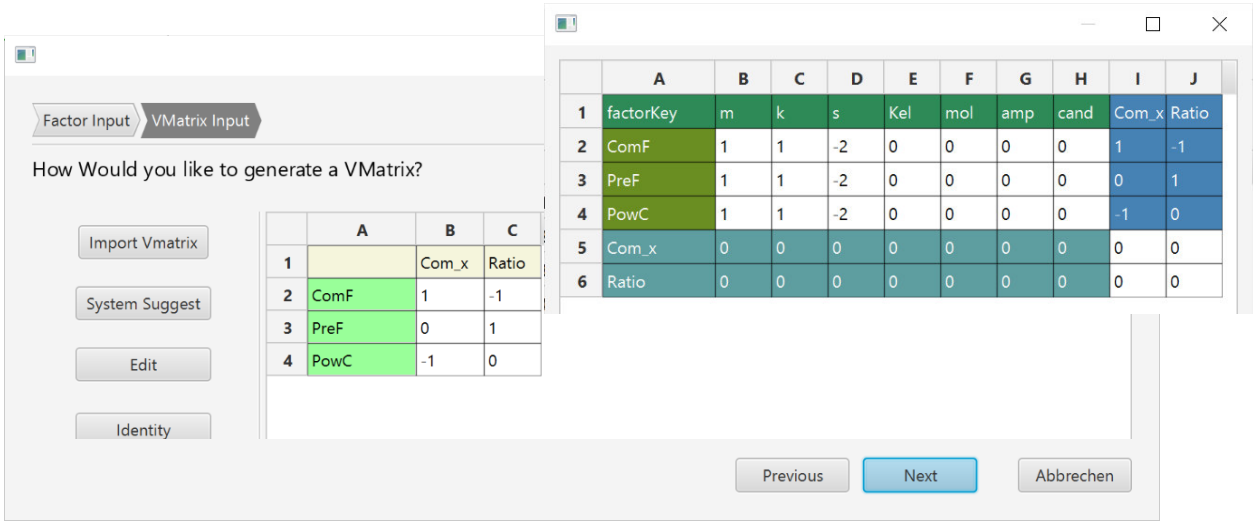
	x_1	x_2
u_1	1	-1
u_2	0	1
u_3	-1	0

No need for approximation, therefore : $T_x = T_{approx}$

Step 1: *u-factors*



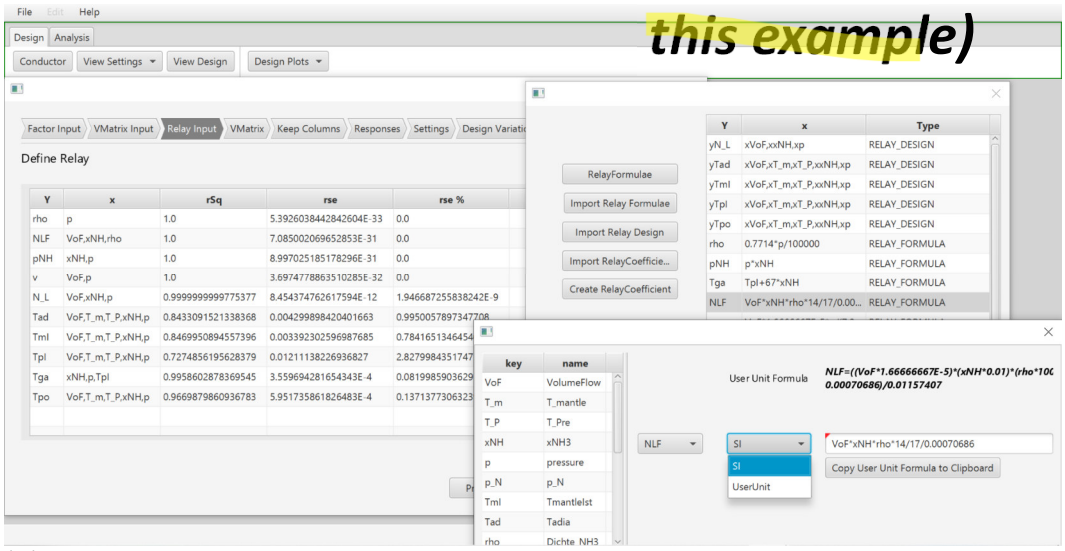
Step 2: ***x-factors***



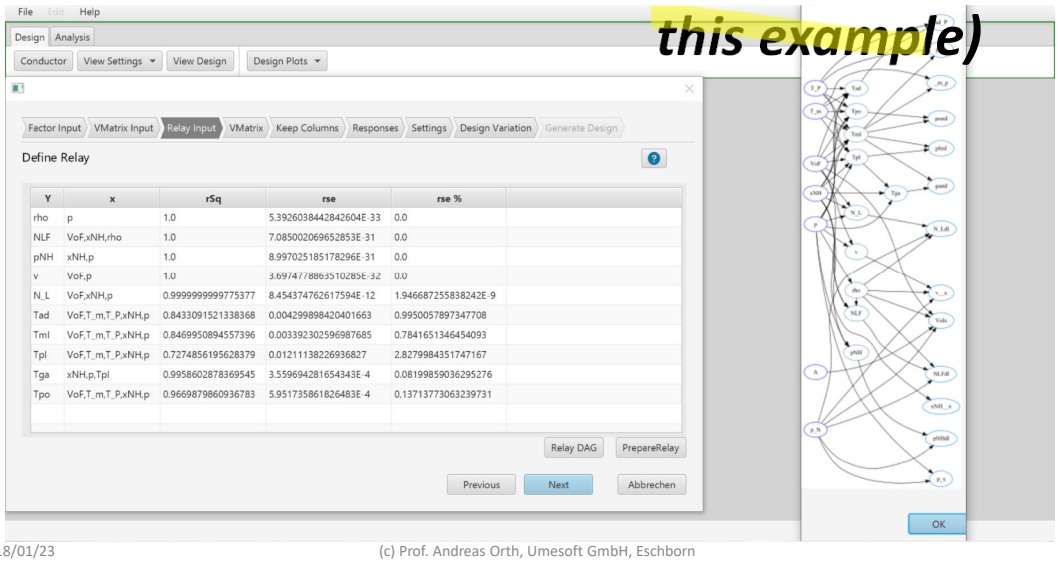
	A	B	C
1		Com_x	Ratio
2	ComF	1	-1
3	PreF	0	1
4	PowC	-1	0

	A	B	C	D	E	F	G	H	I	J
1	factorKey	m	k	s	Kel	mol	amp	cand	Com_x	Ratio
2	ComF	1	1	-2	0	0	0	0	1	-1
3	PreF	1	1	-2	0	0	0	0	0	1
4	PowC	1	1	-2	0	0	0	0	-1	0
5	Com_x	0	0	0	0	0	0	0	0	0
6	Ratio	0	0	0	0	0	0	0	0	0

Step 3: ***u-factor dependencies (not in this example)***



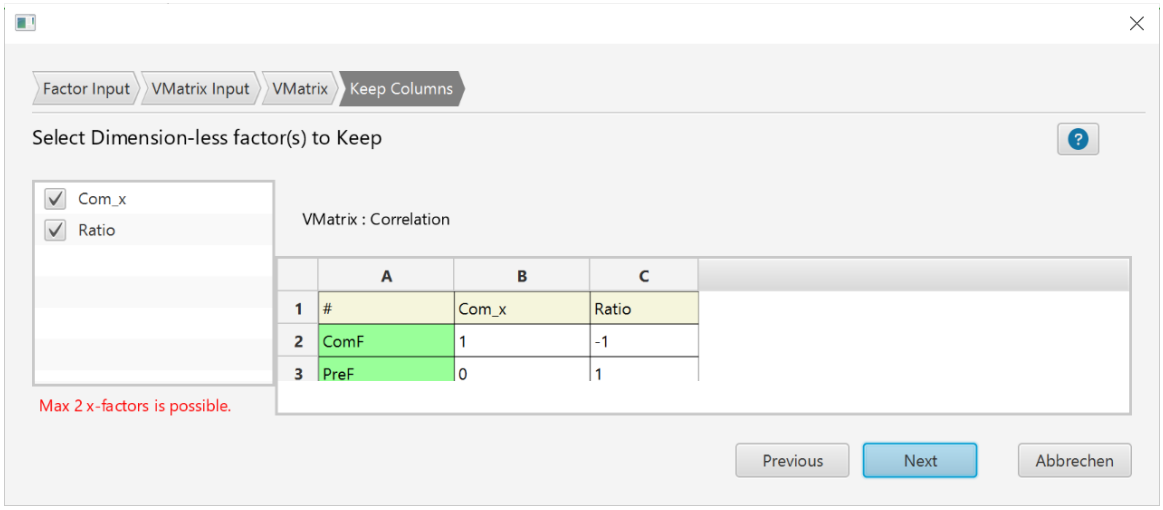
Step 3ff: *u-factor dependencies (not in this example)*



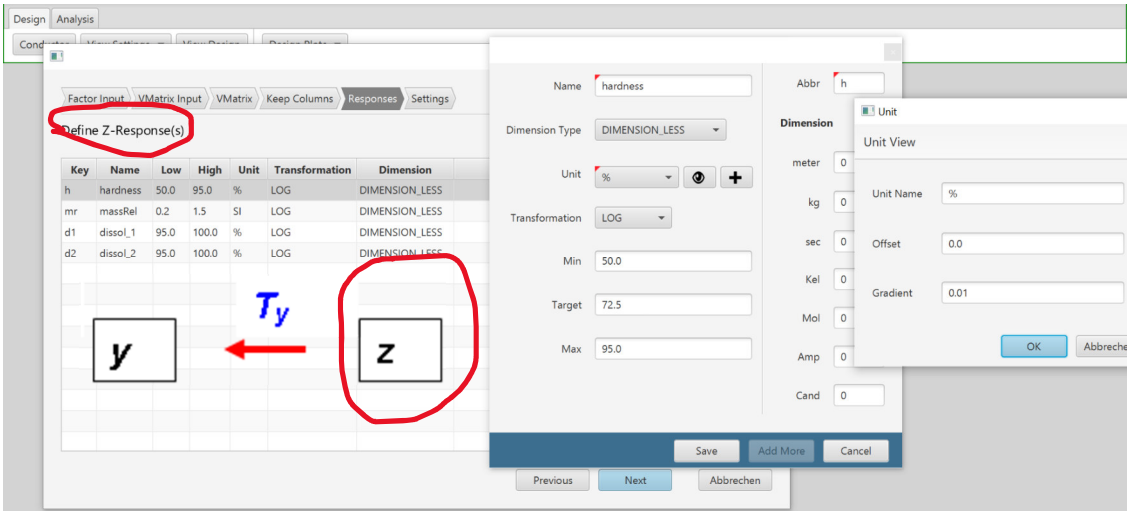
4: Info, T_{approx} – *transformation as a matrix*

In this example $T_{approx} = T$, and it is invertible

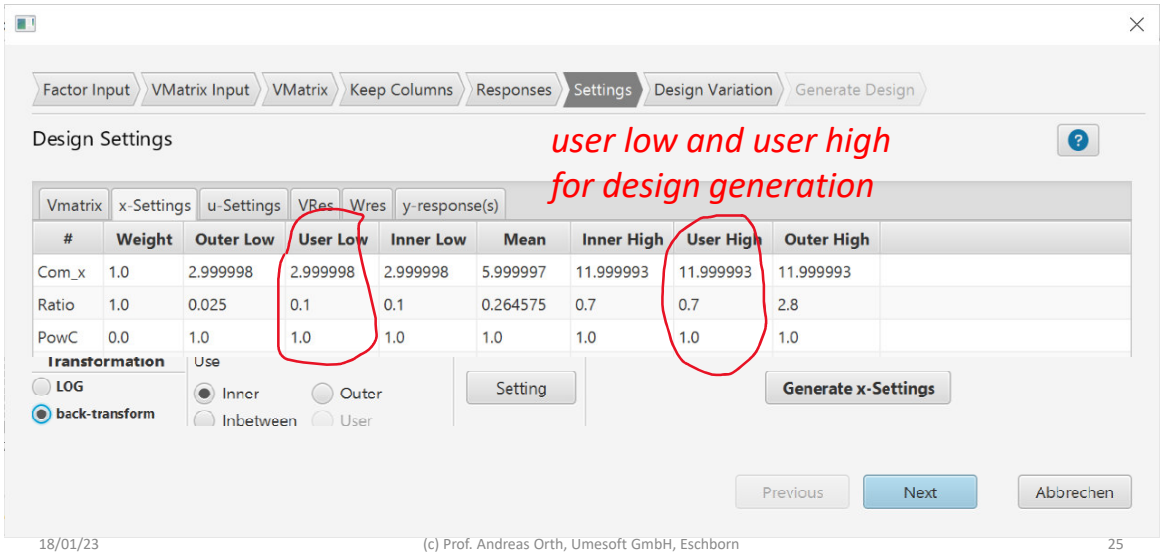
Step 5: ***choose x-factors to use***



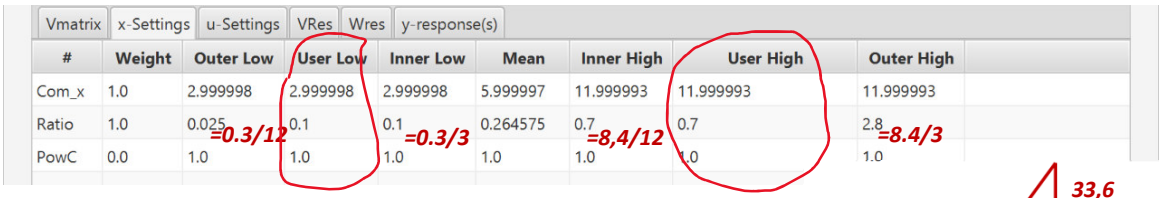
Step 6: ***define z-response(s)***



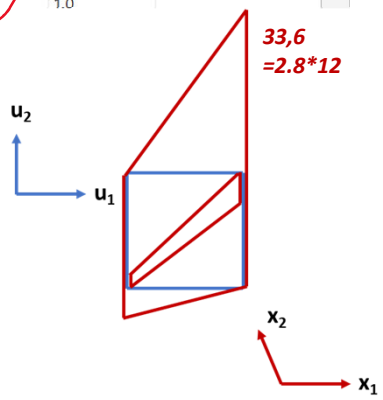
Step 7: *view and edit x-settings*



Step 7: Background



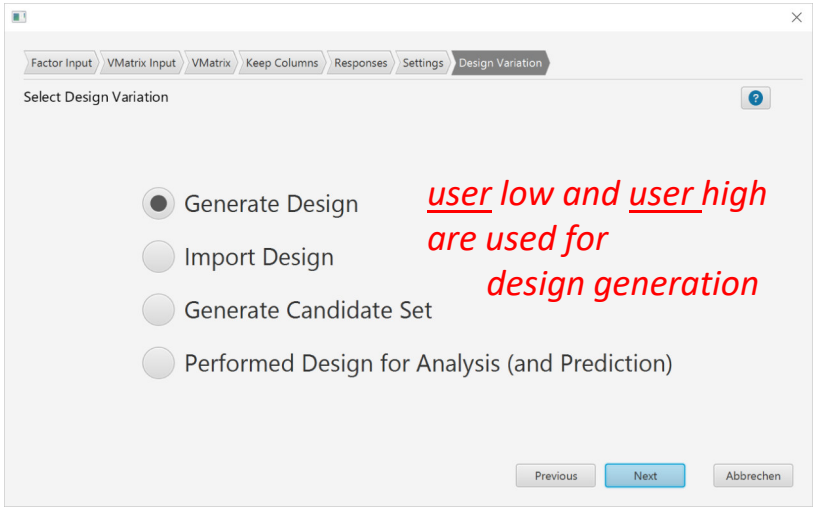
u_1 = compression force, 3 to 12 kN
u_2 = pre-compression force, 0.3 to 8.4 kN
$x_1 = u_1 / u_3$ (dimensionless)
$x_2 = u_2 / u_1$ (dimensionless)



We need *limits* for *x*-factors

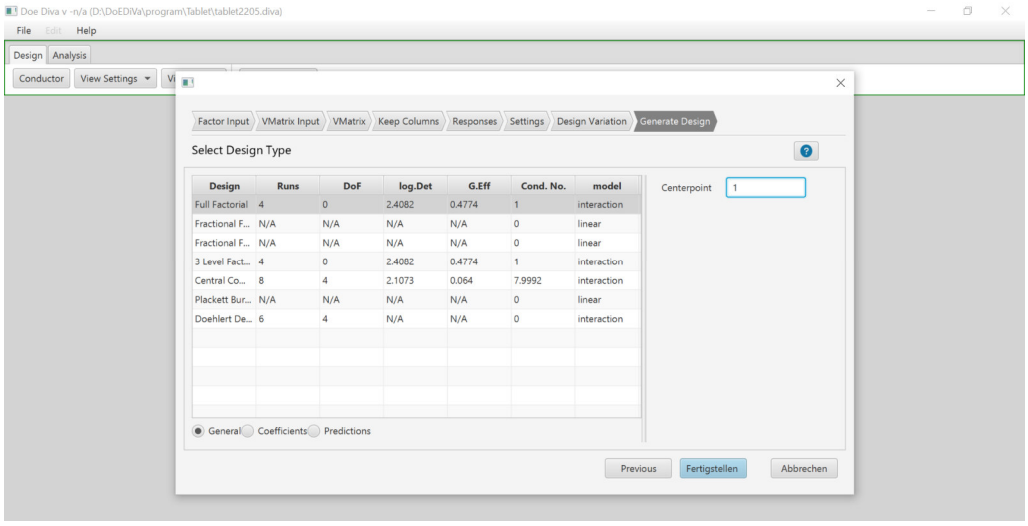
Outer: smallest ***x***-domain containing ***u***-domain
Inner: largest ***x***-domain contained in ***u***-domain

Step 8: ***select the design variation***

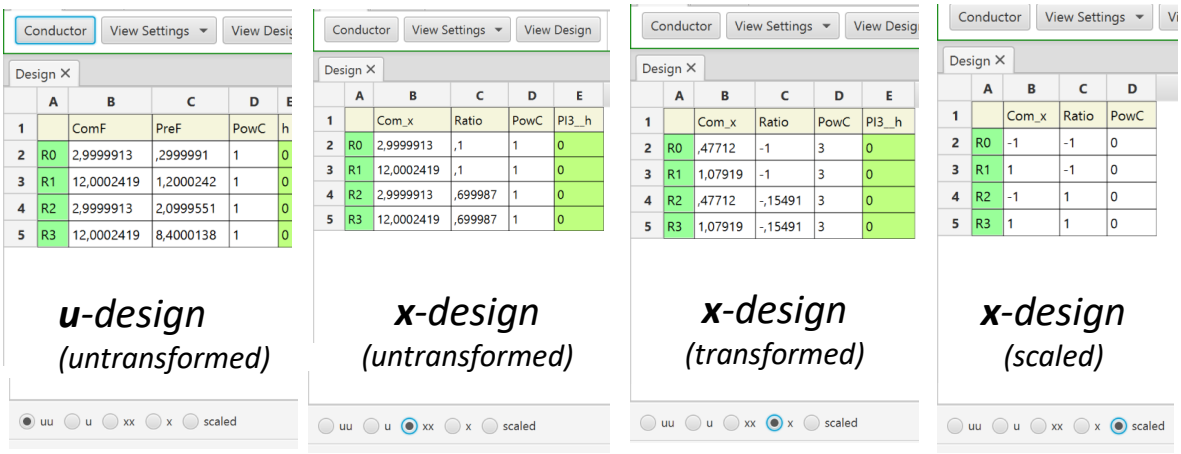


user low and user high are used for design generation

Step 8b: ***select the design***



Step 9: ***view design***



	A	B	C	D	E
1		ComF	PreF	PowC	h
2	R0	2,9999913	,2999991	1	0
3	R1	12,0002419	1,2000242	1	0
4	R2	2,9999913	2,0999551	1	0
5	R3	12,0002419	8,4000138	1	0

***u*-design**
(untransformed)

	A	B	C	D	E
1		Com_x	Ratio	PowC	PI3__h
2	R0	2,9999913	,1	1	0
3	R1	12,0002419	,1	1	0
4	R2	2,9999913	,699987	1	0
5	R3	12,0002419	,699987	1	0

***x*-design**
(untransformed)

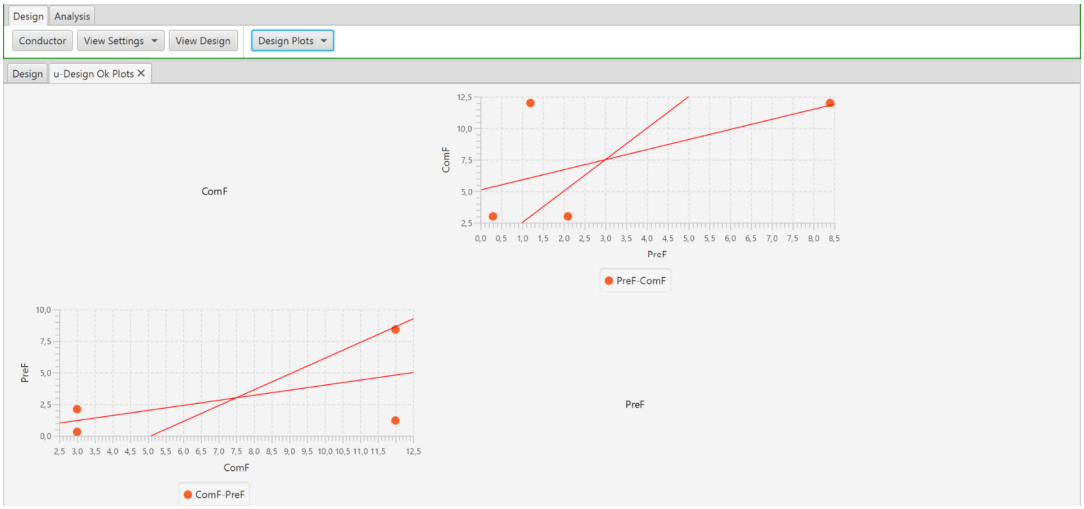
	A	B	C	D	E
1		Com_x	Ratio	PowC	PI3__h
2	R0	,47712	-1	3	0
3	R1	1,07919	-1	3	0
4	R2	,47712	-,15491	3	0
5	R3	1,07919	-,15491	3	0

***x*-design**
(transformed)

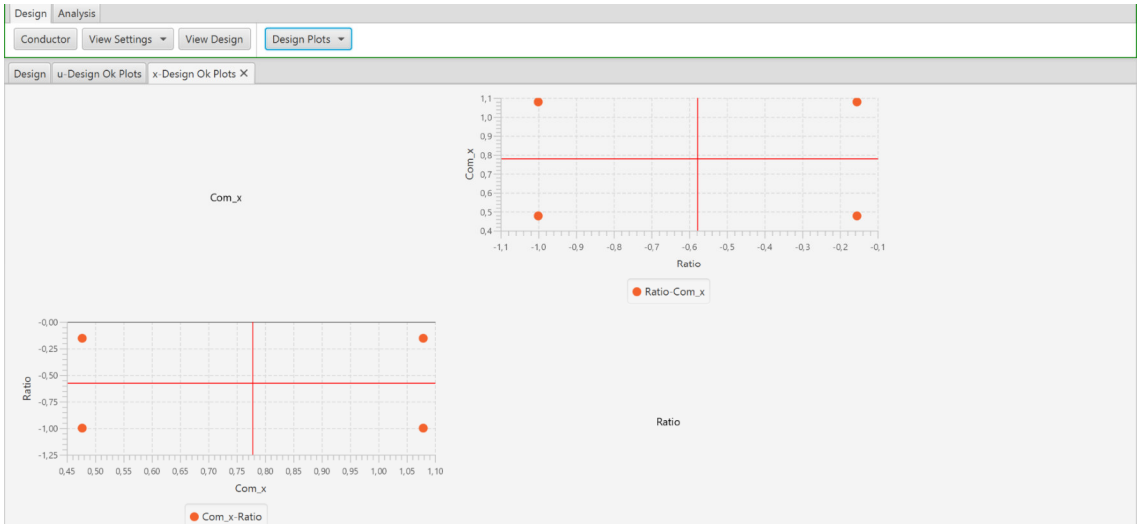
	A	B	C	D
1		Com_x	Ratio	PowC
2	R0	-1	-1	0
3	R1	1	-1	0
4	R2	-1	1	0
5	R3	1	1	0

***x*-design**
(scaled)

Step 10: ***check its structure (u-design)***



Step 10b: ***check its structure (x-design)***



1. e**X**plaining factors vs. **U**ser factors
2. Using DoE-DiVa for for an easy example
3. ***Future Plans and Ideas***

4. A: Existing Functionality

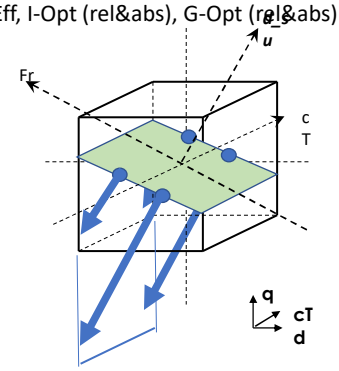
December 2022

- Entering ***u***-facors/u-factors, their dimensions and scales, roles, transformations, limits/settings
- Entering ***x***-factors and their relation to ***u***-factors (solution of Buckinghams Π-Theorem)
- Concept of qualitative dependency (incl. multi-quantitative multi-level factors)
- Concept of „relaying“ for quantitative dependencies from known formulae, linear coefficients or tables
- Approximate Inversion of ***u*** → ***x*** transformation
- Provision of outer and inner factor limits for ***x***-factors given those for ***u***-factors by solving KKT-conditions
- Support for Scale-Up resp. checking for Validity of the Similarity Principle of DA
- Generation of Candidate Sets from limits and constraints using Weyl-Minkowski double description
- Generation of Classical Designs for Inner/Outer or User settings of ***x***-factors
- Generation of Start-Designs for Exchange Algorithms (from classical inner-settings design)
- Display of ***x***-designs, ***u***-designs, ***uuD***-designs (***u***-designs in user units incl. dependent factors)
- ok-Correlation-buster-Plots for ***x***-designs, ***u***-designs
- Automatic creation of designs, constraints, candidates and nested formulae for transfer to MODDE® 13
- Analysis (MLR-fit, residual analysis) of ***y*** ~ ***x***, alternatively for designed ***x*** or ***performed*** ***x.***
- Rudimentary „Many Worlds“ approach – i.e. choosing ***x****-factors* to use – with automatic ***x****-design* update.
- ***Conductor*** to lead through the software Rudimentary Help-System

4. B: Immediate Plans

Next 3 months

- Debug
- X-design ok plots for performed design
- Display Design Quality Measures (VIF, coeff-var, D-Opt, G-Eff, I-Opt (rel&abs), G-Opt (rel&abs)
- Improve Help-System
- Use „slider technique“ to match qualitative settings
- Complete the current Use-Cases
- A runnable .exe will be ready in the near future



4. C: Future Ideas

Next 2-3 years

After Experiments have been performed:
- Improve Analysis and Diagnostics
- Compare and Update Relaying Information – this may affect optimality of designs
- Converting PLS or PCA loadings into V-matrix
- Coping with implicit equations in the u-factor dependencies (e. g. mass balances)

Coping with Partial Similarity in Dimensional Analysis:
- ?? Automisation of „Many-World“ approach: i.e. coping with multiple sets of ***x***-factors
- Compromise Designs for several sets of ***x***-factors (that belong to the same ***u***-design)
- Competitive Models for several sets of ***x***-factors

General Improvements
- Improve Help System
- Continue Debugging and Testing
- Increase the set of Use-Cases

4. D: Quality Assurance

ongoing

Correctness of Algorithms:
- All calculations are double checked against dim-doe R package being developed in parallel
- All kernel functions are subjected to J-unit testing in an IntelliJ-IDE
- User functions are subjected to Used-Cases from 25 years of DoE-experience

Help System:
- Help-System is being developped in parallel with emphasis on Used Cases

Integration and Deployment
- DoE-DiVa uses GIT-versioning and undergoes a Continuous Integration Process for all New Functionality
- DoE-DiVa is deployed as a JAR-File to be launched in JAVA 8 runtime environment
- It is considered to integrate *DoE-DiVA®* into *Umetrics Studio®* by *Sartorius Data Analytics* in future

Vielen Dank!

Gefördert durch:

Bundesministerium
für Wirtschaft
und Energie

aufgrund eines Beschlusses
des Deutschen Bundestages

ganz besonders an:

Chhawang Lama
Anthony Orth