

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:



aufgrund eines Beschlusses
des Deutschen Bundestages

Everyone knows **MODDE®**, ...

- **MODDE®** is the State-of-the-Art user-friendly Software for **Design of Experiments (DoE)**.
- **MODDE®** provides the complete **Design of Experiments** 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 **Design of Experiments** and makes it more intelligent
- **DoE-DiVa** integrates **Similarity Theory for Dimensionless Variables** and provides **DoE's for Scale-Up and Scale-Down**

This is how **MODDE®** looks.

MODDE® has many **Wizards**

MODDE® 13

Simplifying Progress

Optimization wizard **NEU**

The optimization wizard guides you through the steps of design space, identifying setpoints and simulate system

SARTORIUS

1. Design 2. Work 3. Analyze 4. Predict&optimize

This is how the *DoE-DiVa* looks.

The screenshot shows the 'Define Factors' table with the following data:

Key	Name	Low	High	Role	Unit	Transformation	Dimension
ComF	CompressionF	3.0	12.0	CONTR	kN	LOG	FORCE
PreF	PreCompF	0.3	8.4	CONTR	kN	LOG	FORCE
PowC	PowderCompr	1.0	1.0	CONST	kN	LOG	FORCE

The configuration dialog for 'CompressionF' shows the following settings:

- Name: CompressionF
- Abbr: ComF
- Role Type: CONTR
- Dimension Type: FORCE
- Unit: kN
- Transformation: LOG
- Low Setting: 3.0
- High Setting: 12.0

Dimension list:

- meter: 1
- kg: 1
- sec: -2
- Kel: 0
- Mol: 0
- Amp: 0
- Cand: 0

Unit View:

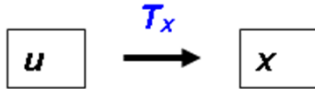
- Unit Name: kN
- Offset: 0.0
- Gradient: 1000.0

DoE-DiVa has a **Conductor**, not a wizard 😊

18/01/23 (c) Prof. Andreas Orth, Umesoft GmbH, Eschborn 5

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

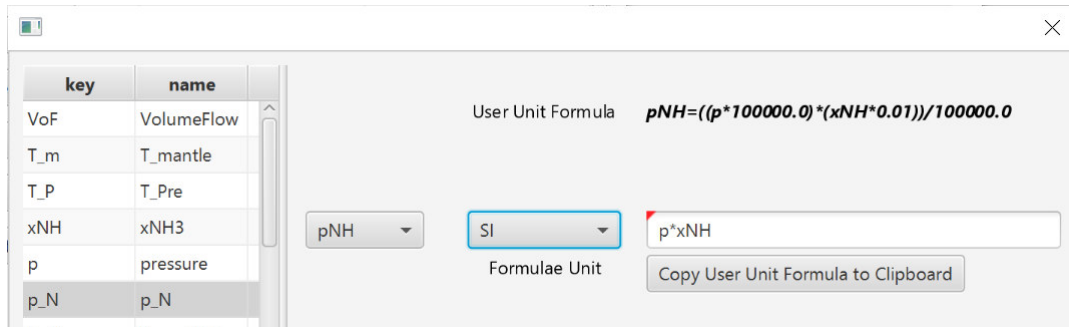
The screenshot shows a list of dimensions with 'FORCE' selected. The configuration dialog for 'ComF' shows the following settings:

- Abbr: ComF
- Dimension: FORCE
- meter: 1
- kg: 1
- sec: -2
- Kel: 0
- Mol: 0
- Amp: 0
- Cand: 0

18/01/23 (c) Prof. Andreas Orth, Umesoft GmbH, Eschborn 6

DoE-DiVa's conductor is flexible

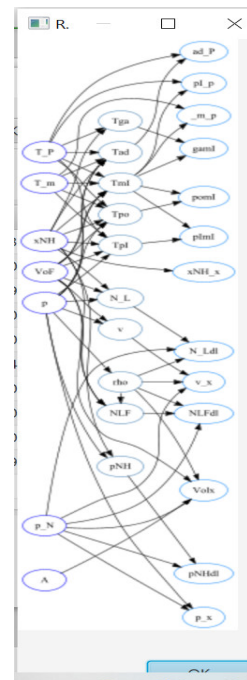
- **DoE-DiVa** also accepts *dependent U-factors* as formulae, data tables (from simulation results or VDI-Waermeatlas) or known coefficients (from gas law for example)



DoE-DiVa's conductor is informative

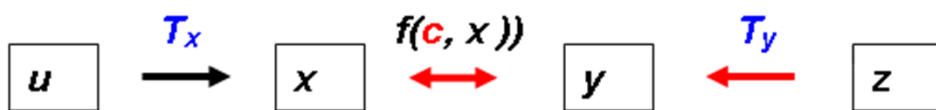
- **DoE-DiVa** generates *factor-dependency plot* and *ok-for-modelling plot*, as well as *classical interpretative and diagnostic plots*
- **DoE-DiVa** has a simple data/formula interface to MODDE® for *process optimization* and *QbD (Quality by Design)*.

	VoF	T_m	T_P	xNH	p	p_N	Tm
1							
2	RO	17,165394	645,6030631	225,5295			
3	R1	16,601838	545,7807901	255,6420			
4	R2	17,165394	645,6030631	225,5295			
5	R3	16,601838	545,7807901	255,6420			



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: *User-factor*, to be set in the experiment, e.g. *Temp, pressure etc.*

x: *eXplaining-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: *coefficients or parameters in the model, f* , to be determined by model FIT

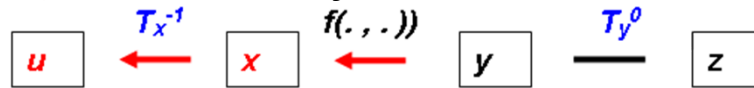
z: *measured response value*

y: *transformed response value, e.g. ratio or product of a z and some u*

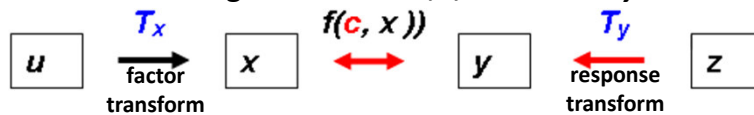
T_y : *transformation* to get from z to y , may also just be *log or neg-log*

The role of the *Transformations*

Design: How to get u from f and the transformation



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

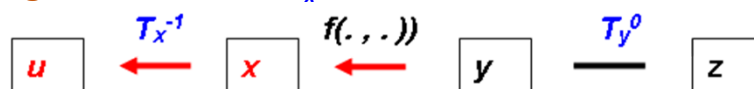


Optimization: how to get the u from the specification for z



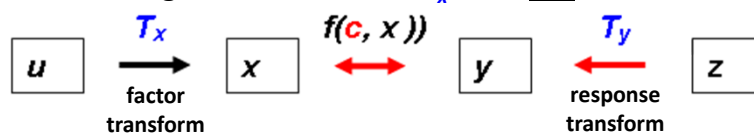
The *Approximation* Trick to get T_x^{-1}

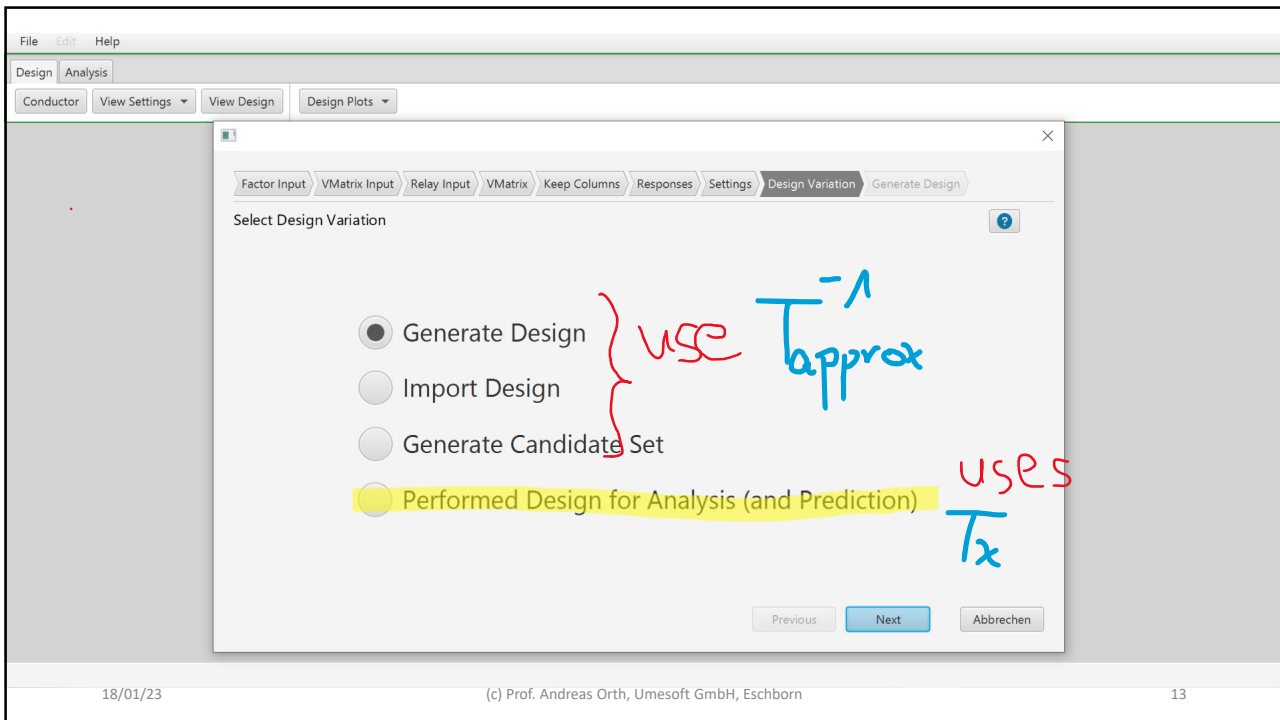
Design: To get u from x , T_x has to be inverted



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 **Dimensionless Variables** 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 constraints.

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

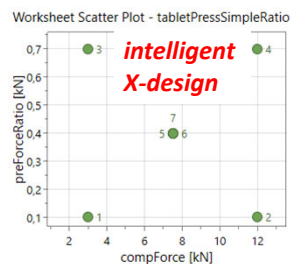
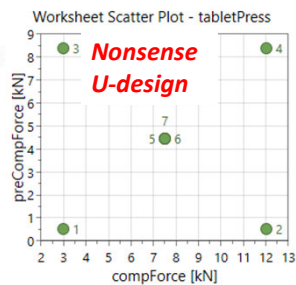
u_1 = compression force, 3 to 12 kN
 u_2 = pre-compression force, 0.3 to 8.4 kN

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

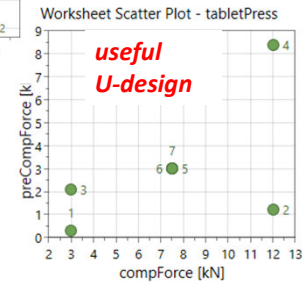
In modern presses, there is a pre-compression step where both punches are forced by pre-compression rollers to press the powder in the die into a tablet.

Easy example: Tablet pressing

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



T_x^{-1}



Technical and explaining Factors

User \rightarrow u $\xrightarrow{T_x}$ x $\xleftrightarrow{f(c, x)}$ y $\xleftarrow{T_y}$ z

Transformed and raw and Responses

Responses

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

Easy example: Tablet pressing

$u_1 = \text{compression force}$
 $u_2 = \text{pre-compression force}$
 $u_3 = \text{compressibility (const)}$

$x_1 = \text{compression force,}$
 $x_2 = \text{pre-force-ratio}$

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

$T_x = T_{approx}$

18/01/23 **umesoft**

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

17

Step 1: u -factors

Design Analysis

Conductor View Settings View Design Design Plots

Factor Input

Define Factors

Key	Name	Low	High	Role	Unit	Transformation	Dimension
ComF	CompressionF	3.0	12.0	CONTR	kN	LOG	FORCE
PreF	PreCompF	0.3	8.4	CONTR	kN	LOG	FORCE
PowC	PowderCompr	1.0	1.0	CONST	kN	LOG	FORCE

Name: CompressionF Abbr: ComF

Role Type: CONTR

Dimension Type: FORCE

Unit: kN

Transformation: LOG

Low Setting: 3.0

High Setting: 12.0

Unit View

Unit Name: kN

Offset: 0.0

Gradient: 1000.0

18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

18

Step 2: x-factors

Factor Input | VMatrix Input

How Would you like to generate a VMatrix?

Import Vmatrix

System Suggest

Edit

Identity

	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

Previous | Next | Abbrechen

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

File | Edit | Help

Design | Analysis

Conductor | View Settings | View Design | Design Plots

Factor Input | VMatrix Input | Relay Input | VMatrix | Keep Columns | Responses | Settings | Design Variat...

Define Relay

Y	x	rSq	rse	rse %
rho	p	1.0	5.3926038442842604E-33	0.0
NLF	VoF*xNH,rho	1.0	7.085002069652853E-31	0.0
pNH	xNH,p	1.0	8.997025185178296E-31	0.0
v	VoF,p	1.0	3.6974778863510285E-32	0.0
N_L	VoF,xNH,p	0.9999999999775377	8.454374762617594E-12	1.946687255838242E-9
Tad	VoF_T_m_T_P,xNH,p	0.8433091521338368	0.004299898420401663	0.9950057897347208
Tml	VoF_T_m_T_P,xNH,p	0.8469950894557396	0.003392302596987685	0.7841651346454
Tpl	VoF_T_m_T_P,xNH,p	0.7274856195628379	0.01211138226936827	2.8279984351747
Tga	xNH,p,Tpl	0.9958602878369545	3.559694281654343E-4	0.0819985903629
Tpo	VoF_T_m_T_P,xNH,p	0.9669879860936783	5.951735861826483E-4	0.1371377306323

RelayFormulae

Import Relay Formulae

Import Relay Design

Import RelayCoefficient...

Create RelayCoefficient

User Unit Formula: $NLF = (VoF * 1.66666667E-5) * (xNH * 0.01) * (rho * 100 / 0.00070686) / 0.01157407$

key | name

- VoF | VolumeFlow
- T_m | T_mantle
- T_P | T_Pre
- xNH | xNH3
- p | pressure
- p_N | p_N
- Tml | Tmantlelst
- Tad | Tadia
- rho | Dichte NH3

Y: NLF

x: SI

User Unit

VoF*xNH*rho*14/17/0.00070686

Copy User Unit Formula to Clipboard

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

The 'Define Relay' dialog box displays the following data:

Y	x	rSq	rse	rse %
rho	p	1.0	5.3926038442842604E-33	0.0
NLF	VoF,xNH,rho	1.0	7.085002069652853E-31	0.0
pNH	xNH,p	1.0	8.997025185178296E-31	0.0
v	VoF,p	1.0	3.6974778963510285E-32	0.0
N_L	VoF,xNH,p	0.999999999775377	8.454374762617594E-12	1.94668725583824E-9
Tad	VoF,T_m,T_P,xNH,p	0.8433091521338368	0.004299898420401663	0.9950057897347708
Tml	VoF,T_m,T_P,xNH,p	0.8469950894557396	0.003392302596987685	0.7841651346454093
Tpl	VoF,T_m,T_P,xNH,p	0.7274856195628379	0.01211138226936827	2.8279984351747167
Tga	xNH,p,Tpl	0.9958602878369545	3.559694281654343E-4	0.08199859036295276
Tpo	VoF,T_m,T_P,xNH,p	0.9669879860936783	5.951735861826483E-4	0.13713773063239731

The dialog also includes buttons for 'Relay DAG', 'PrepareRelay', 'Previous', 'Next', 'Abbrechen', and 'OK'.

18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

21

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

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

The 'Relayed VMatrix' dialog box displays the following data:

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

The dialog also includes buttons for 'Previous', 'Next', and 'Abbrechen'.

18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

22

Step 5: *choose x-factors to use*

Factor Input | VMatrix Input | VMatrix | **Keep Columns**

Select Dimension-less factor(s) to Keep

Com_x
 Ratio

VMatrix : Correlation

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

Max 2 x-factors is possible.

Previous Next Abbrechen

18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

23

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

Design | Analysis

Factor Input | VMatrix Input | VMatrix | Keep Columns | **Responses** | Settings

Define Z-Response(s)

Key	Name	Low	High	Unit	Transformation	Dimension
h	hardness	50.0	95.0	%	LOG	DIMENSION_LESS
mr	massRel	0.2	1.5	SI	LOG	DIMENSION_LESS
d1	dissoL1	95.0	100.0	%	LOG	DIMENSION_LESS
d2	dissoL2	95.0	100.0	%	LOG	DIMENSION_LESS

Unit

Name: hardness
Abbr: h
Dimension Type: DIMENSION_LESS
Unit: %
Transformation: LOG
Min: 50.0
Target: 72.5
Max: 95.0

Unit View
Unit Name: %
Offset: 0.0
Gradient: 0.01

Save Add More Cancel

Previous Next Abbrechen

18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

24

Step 7: *view and edit* x-settings

Design Settings

user low and user high for design generation

#	Weight	Outer Low	User Low	Inner Low	Mean	Inner High	User High	Outer High
Com_x	1.0	2.999998	2.999998	2.999998	5.999997	11.999993	11.999993	11.999993
Ratio	1.0	0.025	0.1	0.1	0.264575	0.7	0.7	2.8
PowC	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Transformation: Use
 LOG
 back-transform
 Inner
 Outer
 Inbetween
 User

Buttons: Previous, Next, Abrechnen, Generate x-Settings

18/01/23 (c) Prof. Andreas Orth, Umesoft GmbH, Eschborn 25

Step 7: Background

#	Weight	Outer Low	User Low	Inner Low	Mean	Inner High	User High	Outer High
Com_x	1.0	2.999998	2.999998	2.999998	5.999997	11.999993	11.999993	11.999993
Ratio	1.0	0.025	0.1	0.1	0.264575	0.7	0.7	2.8
PowC	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

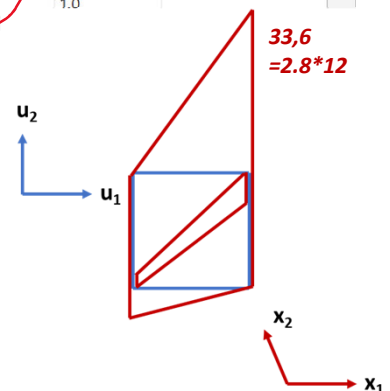
Handwritten annotations:
 Ratio: Outer Low = 0.3/12, Inner Low = 0.3/3, Inner High = 8.4/12, Outer High = 8.4/3
 User High: 33.6 = 2.8 * 12

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)

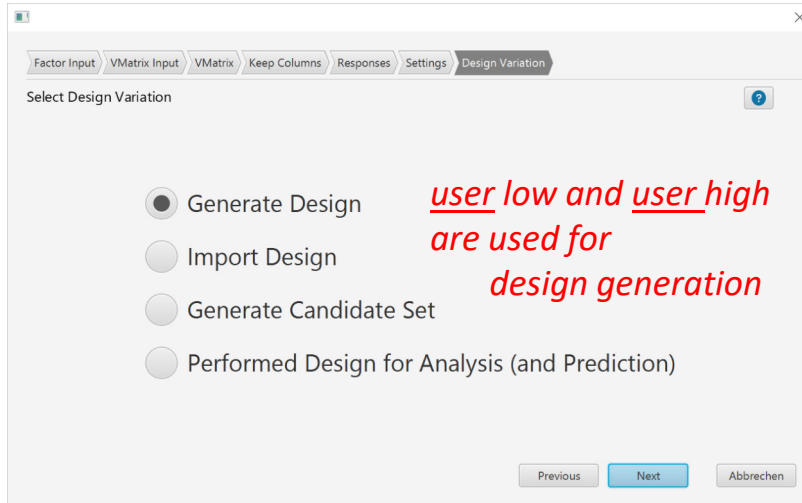
We need **limits** for x-factors

Outer: smallest x-domain containing u-domain

Inner: largest x-domain contained in u-domain



Step 8: *select the design variation*

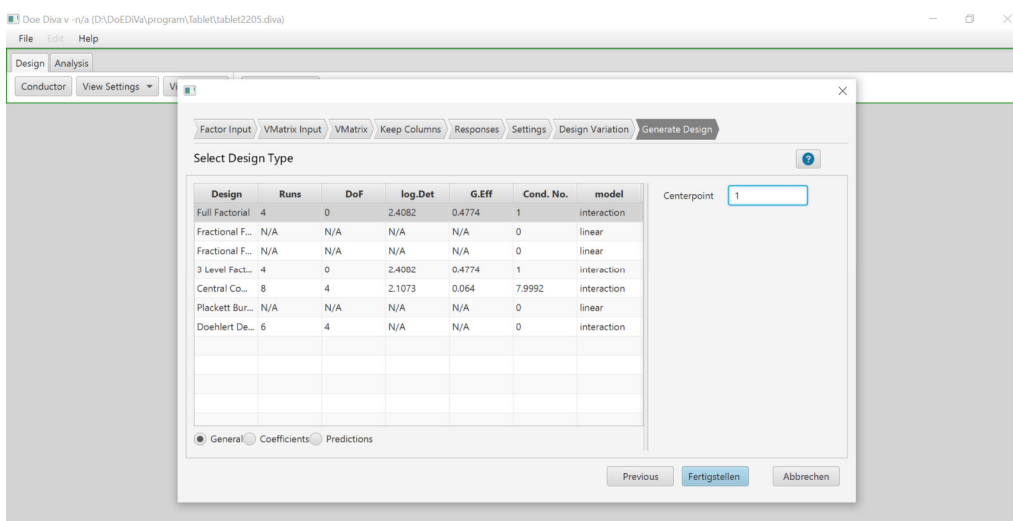


18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

27

Step 8b: *select the design*



18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

28

Step 9: *view design*

Design X					
	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)

uu u xx x scaled

Design X					
	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)

uu u xx x scaled

Design X					
	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)

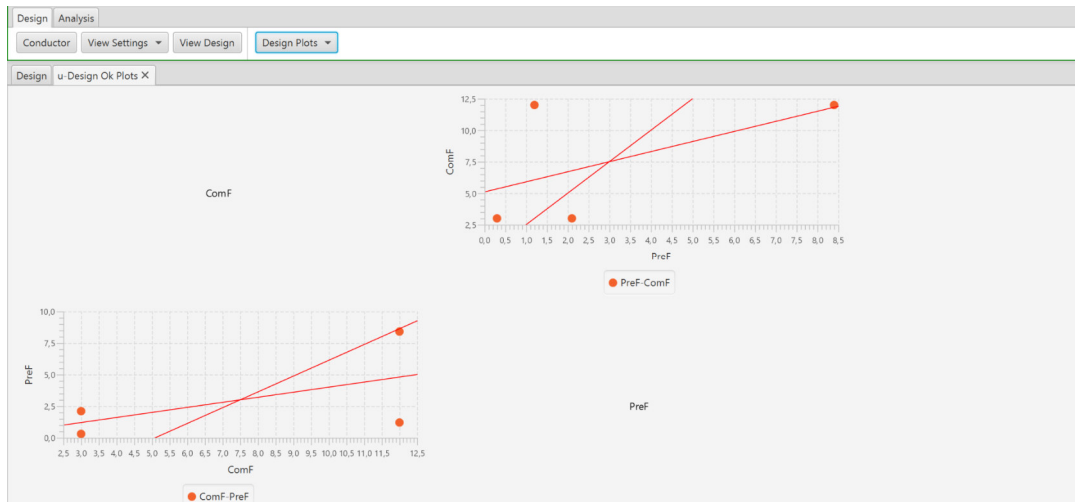
uu u xx x scaled

Design X				
	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

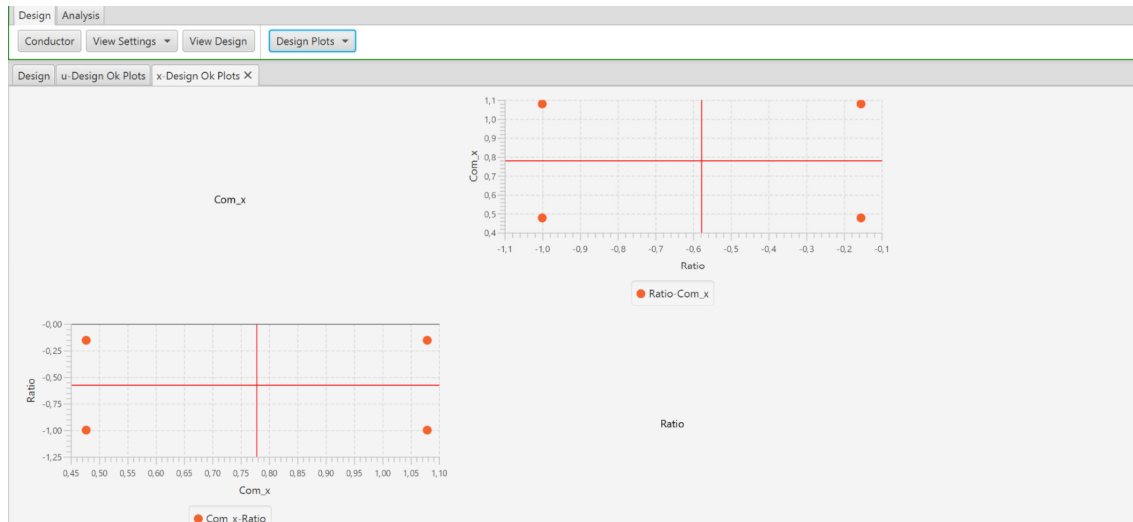
x-design
(scaled)

uu u xx x scaled

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



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



18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

31

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

18/01/23

(c) Prof. Andreas Orth, Umesoft GmbH, Eschborn

32

4. A: Existing Functionality

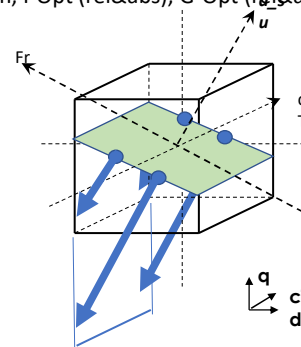
December 2022

- Entering u -factors/u-factors, their dimensions and scales, roles, transformations, limits/settings
- Entering x -factors and their relation to u -factors (solution of Buckingham's 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 $u \rightarrow x$ 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 $y \sim x$, 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 developed 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