

Robustes Design und Optimierung von HF-Komponenten mittels Sensitivitäten

Franz Hirtenfelder
Applications Engineer

CST-Computer Simulation Technology AG,
Bad Nauheimer Strasse 19, 64289
Darmstadt, Germany.

Tel: +49 89 2420 828 101, Mob: +49 170 9160 110, Fax: +49 89 2420 828 198
Email: franz.hirtenfelder@cst.com, Web: www.cst.com

--

Agenda

- Einführung
- Übersicht der verwendeten Optimierungsalgorithmen
 - Genetic Algorithm
 - Particle Swarm Optimization
 - Nelder-Mead Simplex
 - Interpolated Quasi-Newton
 - Trust Region Framework
- Sensitivität /Yield
- Beispiele
 - Cavity und Iris gekoppelte Bandpass Filter
 - Zweistufiger Impedanztrafo
 - Planar filter tuning
- Zusammenfassung

Agenda

- **Einführung**
- Übersicht der verwendeten Optimierungsalgorithmen
 - Genetic Algorithm
 - Particle Swarm Optimization
 - Nelder-Mead Simplex
 - Interpolated Quasi-Newton
 - Trust Region Framework
- **Sensitivität /Yield**
- **Beispiele**
 - Cavity und Iris gekoppelte Bandpass Filter
 - Zweistufiger Impedanztrafo
 - Planar filter tuning
- **Zusammenfassung**

Terminologie

Optimierungs-Prozess: $\underline{x}_f = \arg \min_x U(R(\underline{x}))$

Optimal design to
be determined

Objective (goal or
fitness) function

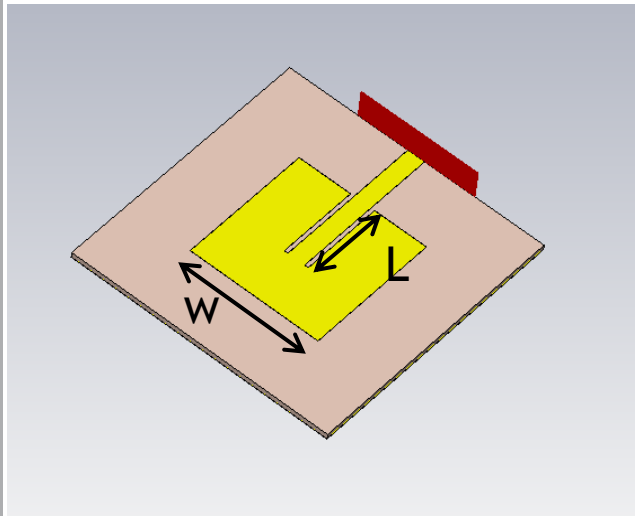
Response vector
e.g. S21 vs.
frequency
(3D Simulation)

Design parameters
e.g. dimensions,
material properties
etc.

x can vary over the
search space
(parameter space)

General Optimizer Setup

Parameterization



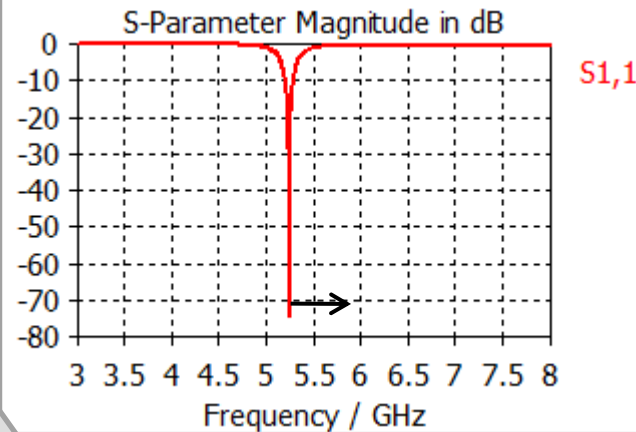
1. Define structure parameters

Name	Value	Description
w	15	patch width
L	10	gap length

2. Define parameter ranges

	Parameter	Min	Max
☒	L	9	11
☒	w	13.5	16.5

Definition of Goal function



Type	Operator	Target
1D Result: S11	move min	6

Optimizer Choice

- Trust Region Framework
- Genetic Algorithm
- Particle Swarm Optimization
- Nelder Mead Simplex Algorithm
- Interpolated Quasi Newton
- Trust Region Framework
- Classic Powell

Agenda

- Einführung
- **Übersicht der verwendeten Optimierungsalgorithmen**
 - Genetic Algorithm
 - Particle Swarm Optimization
 - Nelder-Mead Simplex
 - Interpolated Quasi-Newton
 - Trust Region Framework
- Sensitivität /Yield
- Beispiele
 - Cavity und Iris gekoppelte Bandpass Filter
 - Zweistufiger Impedanztrafo
 - Planar filter tuning
- Zusammenfassung

Local vs. Global Optimizers

Classic Powell

Interpolated Quasi Newton

Trust Region Framework

Nelder-Mead Simplex Algorithm

Particle Swarm Optimization

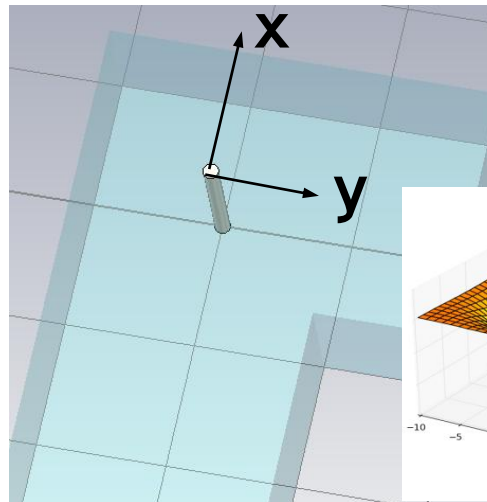
Genetic Algorithm

local

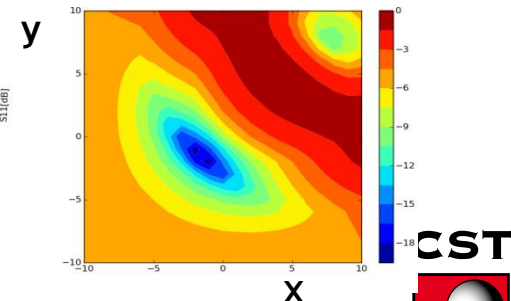
global

Initial parameters already
give a good estimate of the
optimum, parameter ranges
are small

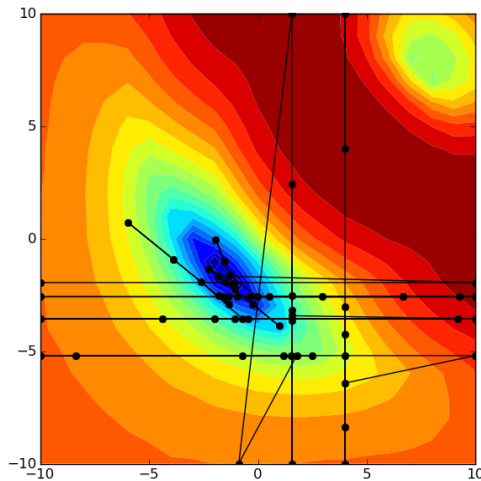
Beispiel:
Waveguide Corner Goal:
Minimize S_{11}



Initial parameters give a
poor estimate of the
optimum, parameter
ranges are large

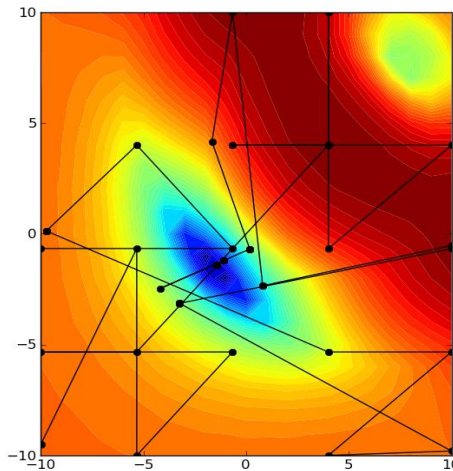


WG Corner: Landscape Visualization (1)



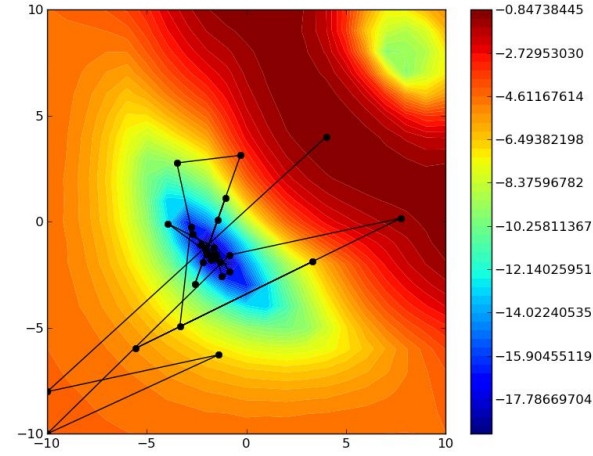
Classic Powell:

A local optimizer that robustly finds an optimum within the given parameter bounds. Sometimes, many iterations are necessary when closing in on the optimum.



Quasi Newton:

A Search algorithm for expensive problems: The parameter space is sampled in each variable direction. EM simulations are only performed for these discrete parameter space points. A model is created from these evaluations and used for optimization. During the search, the model is updated regularly by real evaluations.



Nelder Mead

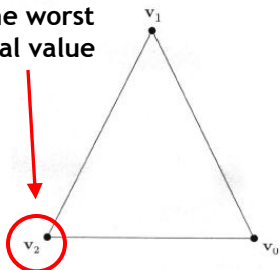
An optimizer for more complex problem domains with good convergence behavior: Uses relatively few evaluations if the problem has a low number of parameters (i.e., less than 5). In geometry, a simplex or n -simplex is an n -dimensional analogue of a triangle.

Nelder-Mead Simplex Method I.

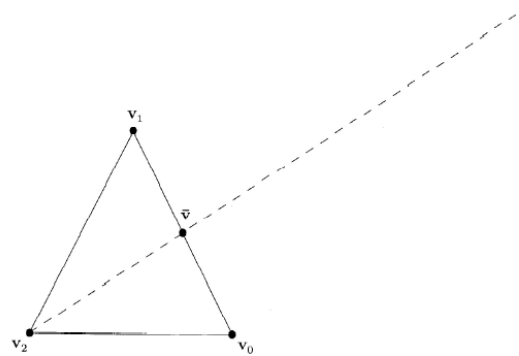


In geometry, a simplex or n -simplex is an n -dimensional analogue of a triangle. This optimization method has been invented by John Nelder & R. Mead (1965).

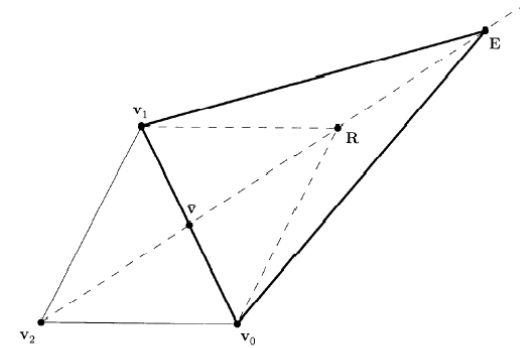
The worst goal value



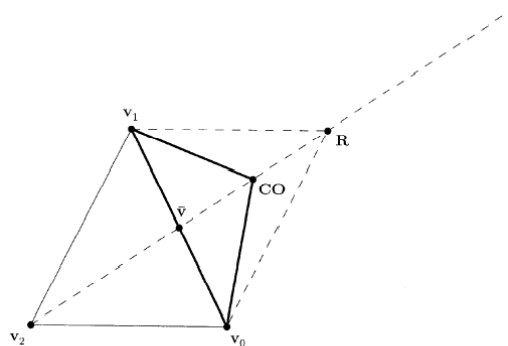
A simplex (2-d)



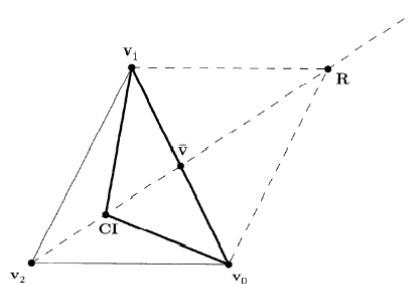
The search direction for the Nelder-Mead algorithm



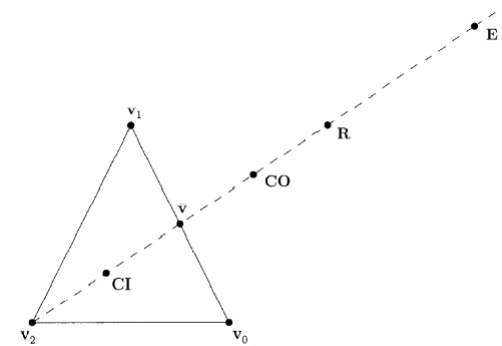
Expand step



Contract outside step



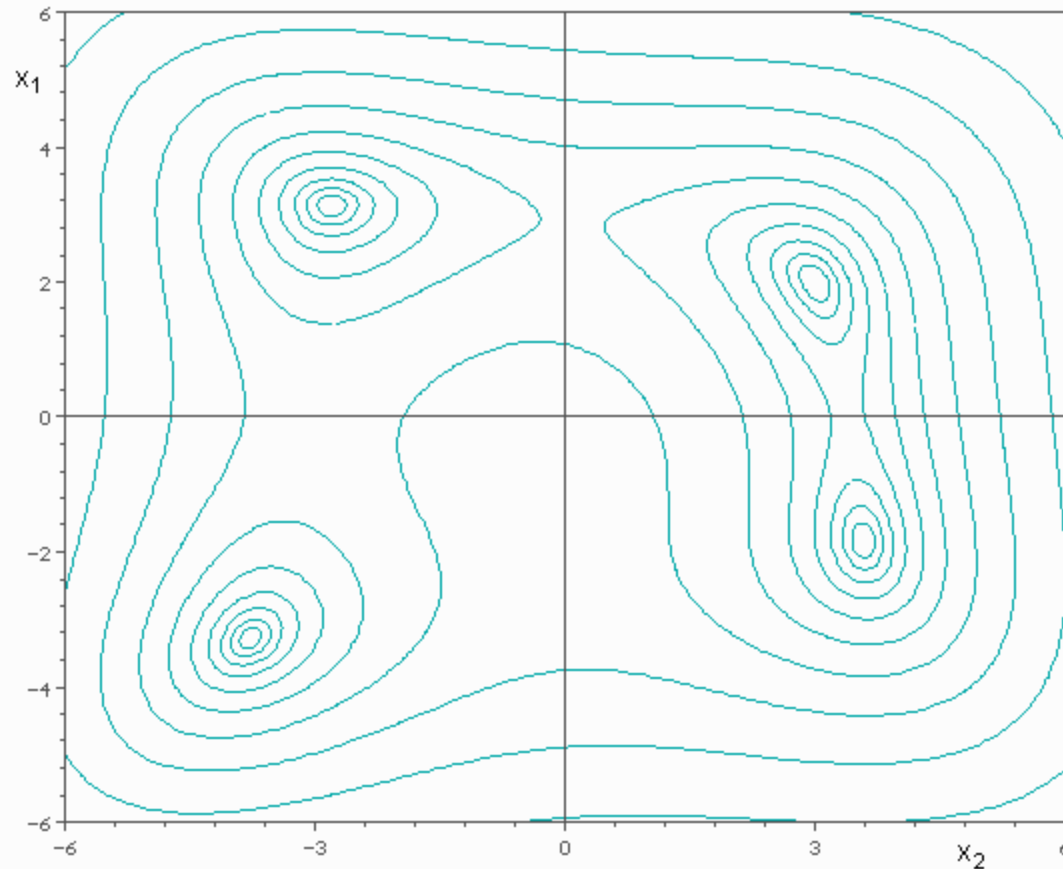
Contract inside step



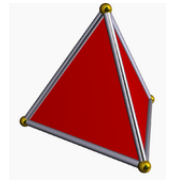
All of the trial points for the Nelder-Mead algorithm

Nelder-Mead Simplex Method II.

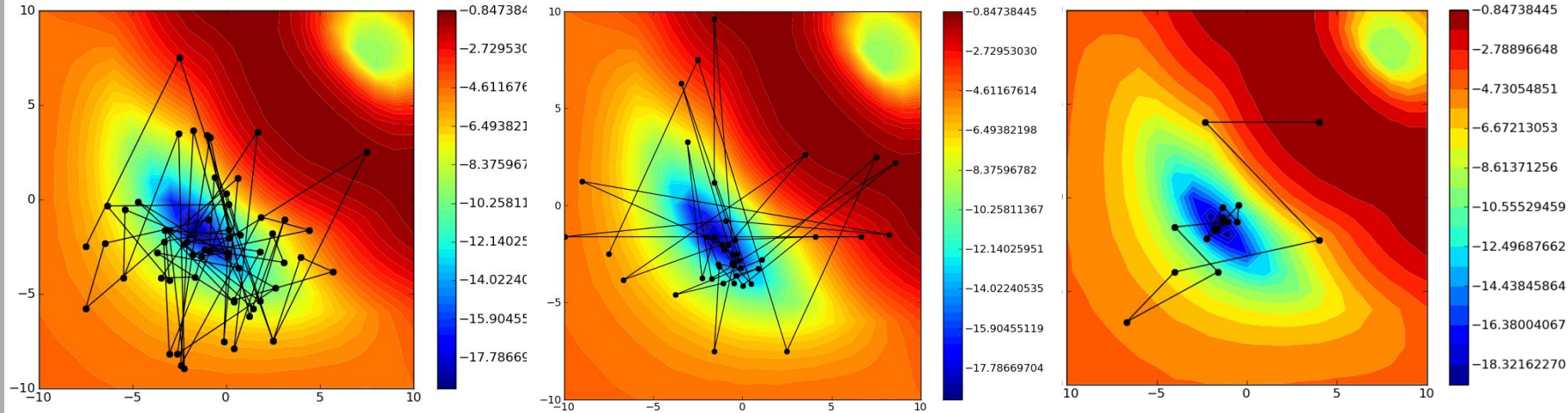
Nelder-Mead Simplex search over Himmelblau function



(c) P.A. Simionescu 2006



WG Corner: Landscape Visualization (2)



Particle Swarm

A global optimizer that uses a higher number of evaluations to explore the search space, also suited for larger numbers of parameters (hint: use distributed computing).

Genetic Algorithm

A global optimizer that uses a high number of evaluations to explore the search space, suited for large numbers of parameters or very complex problem domains (hint: use distributed computing).

Trust Region Framework (TRF)

A fast and accurate optimizer that converges robustly and finds an optimum within the given parameter bounds using a low number of evaluations.

Agenda

- Einführung
- Übersicht der verwendeten Optimierungsalgorithmen
 - Genetic Algorithm
 - Particle Swarm Optimization
 - Nelder-Mead Simplex
 - Interpolated Quasi-Newton
 - Trust Region Framework
- **Sensitivität /Yield**
- **Beispiele**
 - Cavity und Iris gekoppelte Bandpass Filter
 - Zweistufiger Impedanztrafo
 - Planar filter tuning
- **Zusammenfassung**

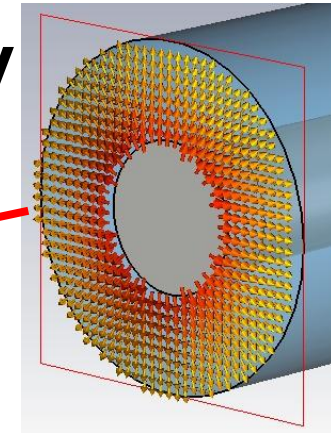
Introduction to Sensitivity

S-Parameters:

3D Fieldsolution

$$S(\omega, p) = \frac{1}{-j\omega\mu_0} E^T(\omega, p) K^T(\omega, p) E(\omega, p)$$

[K] ...left hand side, E (Fields at ports, p... any parameter)



Transient Solver...

Frequency Domain Solver...

Sensitivity of S-parameter vs. parameter change:

$$-j\omega\mu_0 \frac{\partial S}{\partial p_i} = E^T \frac{\partial K}{\partial p_i} E$$

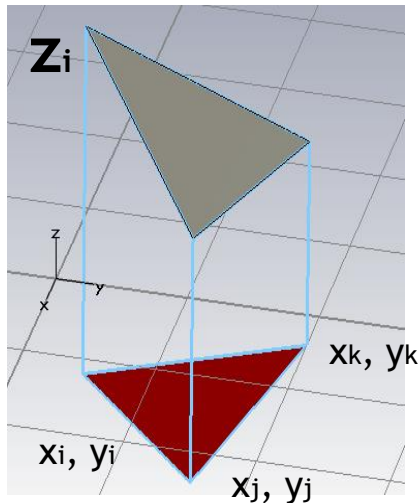
Same 3D Fieldsolution

Direct analytical derivation of K-matrix elements via e.g. [N]

Introduction to Sensitivity

Matrix to solve: $[K]\{E\} = \{Q\}$

$[K]$: symmetric, complex, contains geometry, material, frequency



Example: Linear Shape functions for a 2D element in xy

$$[N] = -\frac{1}{2\Delta} [1, x, y] \begin{bmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \\ c_i & c_j & c_k \end{bmatrix}; a_{ijk} = x_j y_k - y_j x_k; b_{ijk} = y_j - y_k; c_{ijk} = x_k - x_j$$

$$z = [N_i, N_j, N_k] \begin{Bmatrix} z_i \\ z_j \\ z_k \end{Bmatrix}$$

Example: electrostatic

$$k_{m,n} = \iint_{xy} (\epsilon_x \frac{\partial N_m}{\partial x} \frac{\partial N_n}{\partial x} + \epsilon_y \frac{\partial N_m}{\partial y} \frac{\partial N_n}{\partial y}) dx dy; m, n = i, j, k$$

$[E]$: unknowns z

$[Q]$: Sources

Introduction to Sensitivity

What is it good for?

- The sensitivity helps estimate „new“ S-parameters due to the (small) change of the parameter, at no extra cost

Suppose the parameter p changes by a quantity Δp :

$$S(x + \Delta p) \approx S(x) + \sum_i \frac{\partial S}{\partial p_i} \cdot \Delta p_i$$



exact computation of the Sensitivity

(Approximated by 1st order Taylor expansion)

- The various sensitivities are used in an optimizer to solve for Δp as variables to best fit the S-parameter goals.

$$S_{nm} \Leftarrow S_{nm(3D-MWS)} + \sum_i \frac{\partial S}{\partial p_i} \cdot \Delta p_i$$



Δp ... face constraints

Typical Approach vs. CST Approach

- How is yield calculated typically?
 - Parameters vary according to a known probability curve
 - Repeat
 - Change the value of all parameters
 - Simulate
 - Check if specification (in our case for S-params.) is met
 - Until the number of simulations is statistically relevant
 - This is a large number of EM simulations - typically hundreds or thousands!!!
- Knowing the sensitivity, there is **no need to perform 3D simulations**, at least if the **parameters vary in a small range**.
 - The efficiency of this new sensitivity analysis approach makes Monte-Carlo based yield analysis feasible even for complex multi-parametrical three-dimensional structures

Agenda

- Einführung
- Übersicht der verwendeten Optimierungsalgorithmen
 - Genetic Algorithm
 - Particle Swarm Optimization
 - Nelder-Mead Simplex
 - Interpolated Quasi-Newton
 - Trust Region Framework
- Sensitivität /Yield
- **Beispiele**
 - Cavity und Iris gekoppelte Bandpass Filter
 - Zweistufiger Impedanztrafo
 - Planar filter tuning
- Zusammenfassung

Beispiel 1: 2-Post Bandpass Filter



Filter Tutorial

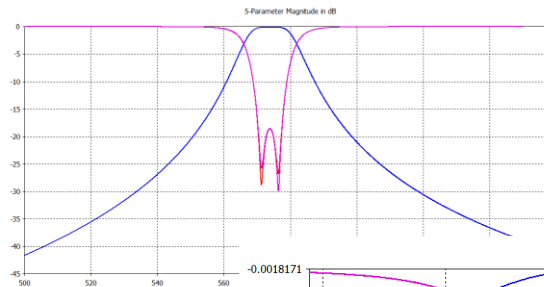
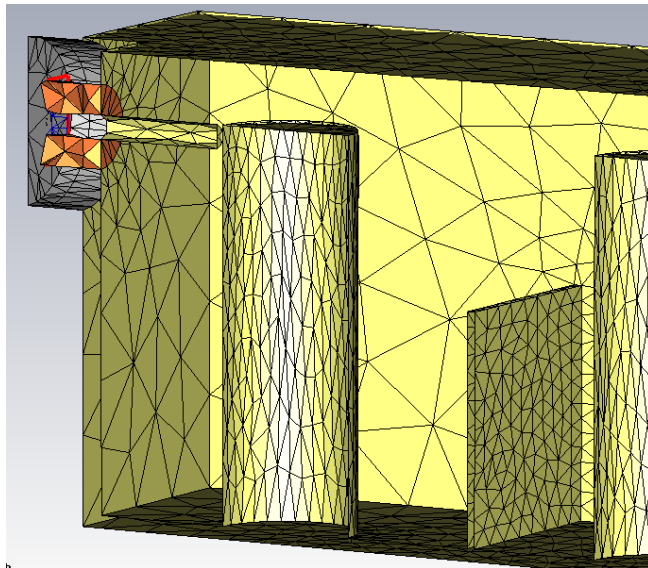
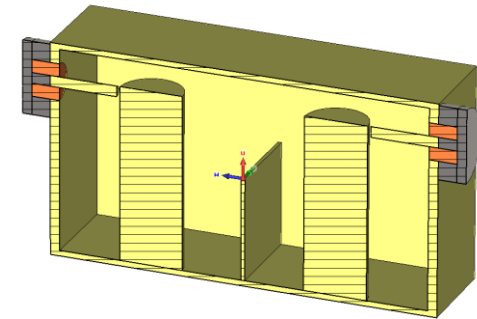
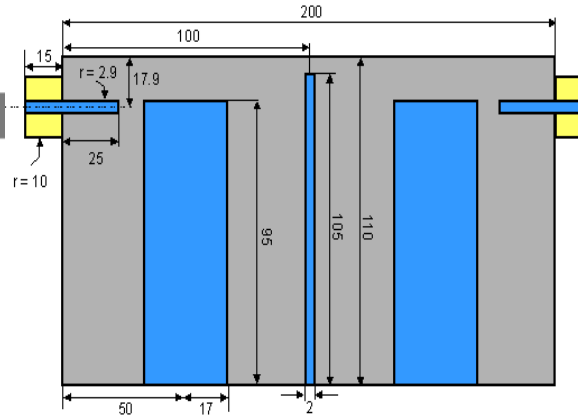
Tutorials

Frequency Domain Tetrahedral:

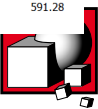
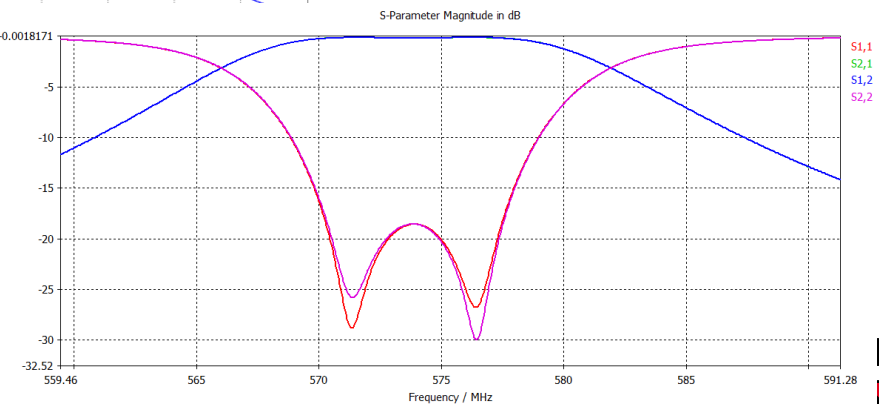
Run Example

Frequency Domain Resonant:

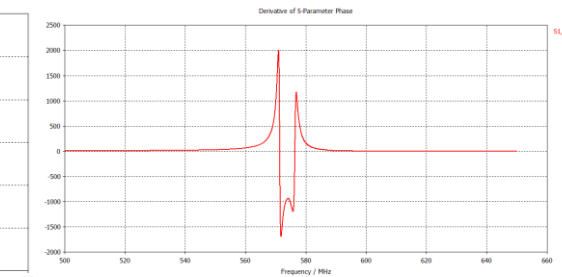
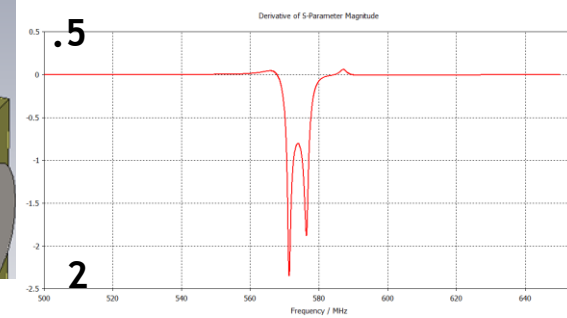
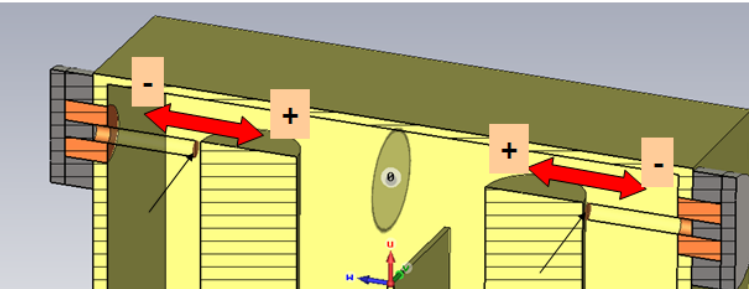
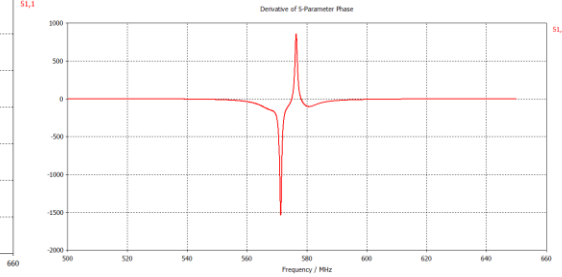
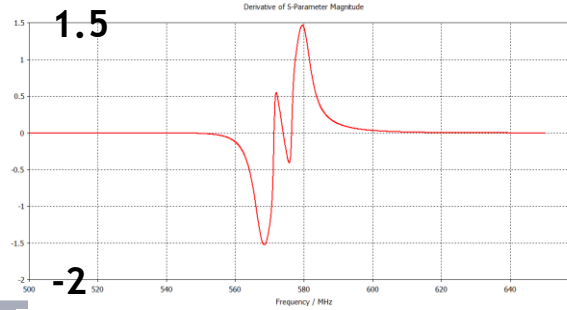
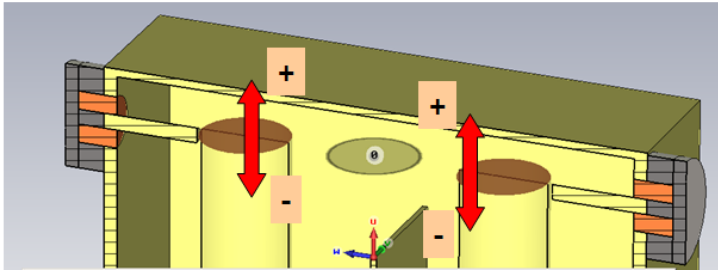
Run Example



Rel Bandwidth: 0.9 %

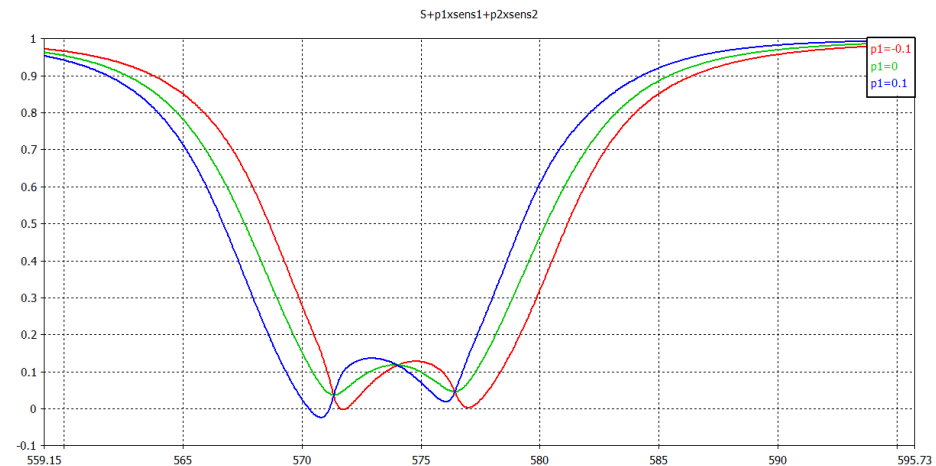
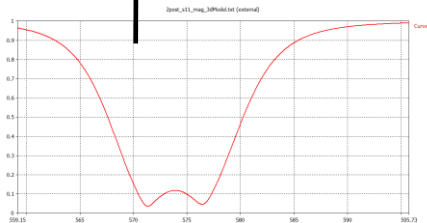


Bsp. 1: Optimization using Sensitivity Data



$$S_{11} = S_{11_3D_MWS} + \sum_p \text{sens}_p \cdot \Delta p$$

?



Bsp. 1: Perform Optimizations : Simplex

Initial parameters

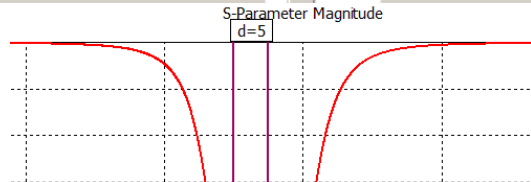
Reset min/max 15 % of initial value

Parameter	Min	Max
☒ p2	-0.15	0.15
☒ p1	-0.15	0.15

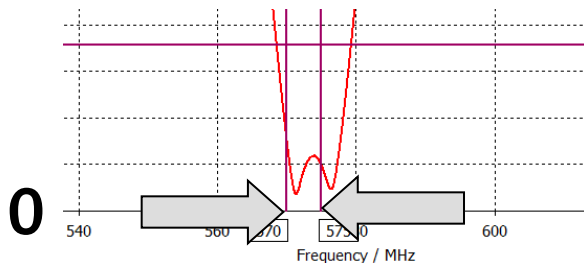
Parameters Goals Algorithm Info

Add new goal ... Edit... Remove All Remove

Type	Operator	Target	Range	Weight
☒ 0D Result: S+p1xsens1+p2xsens2_Max	=	0.0	-	1.0
☐ 0D Result: S+p1xsens1+p2xsens2_Max_lc	=	1	-	1.0
☐ 0D Result: S+p1xsens1+p2xsens2_Max_u	=	1	-	1.0



New Goal: minimize S11 in a somewhat lower frequency band



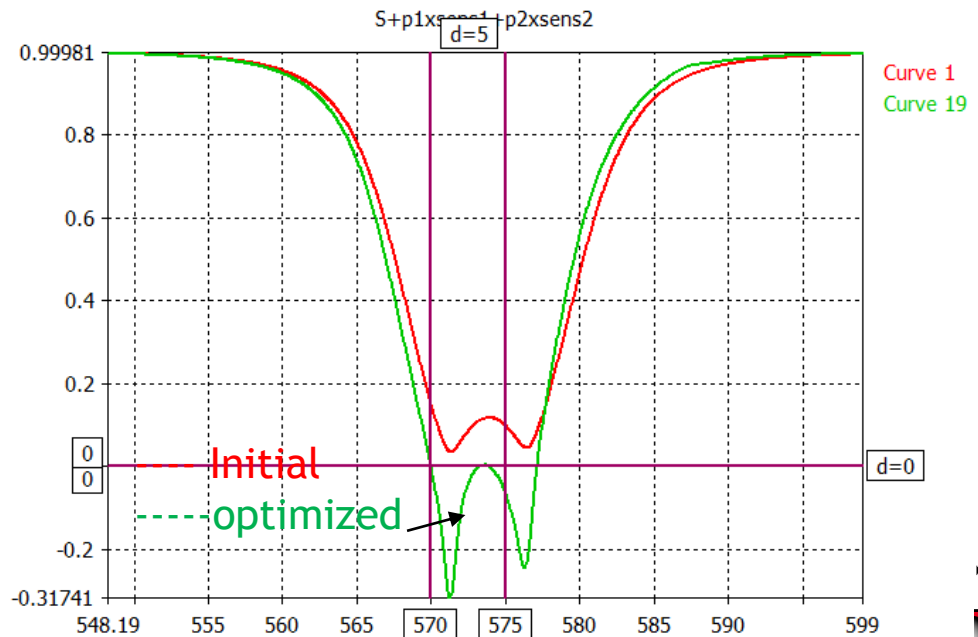
Optimizer

Parameters Goals Algorithm Info

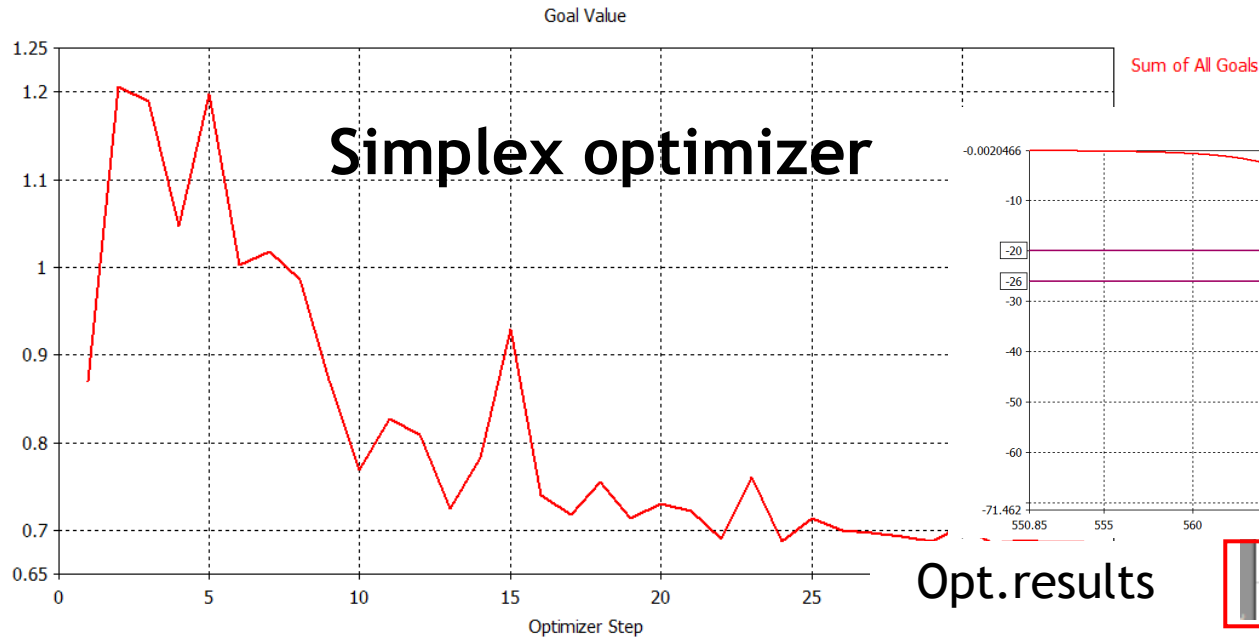
Algorithm: Nelder Mead Simplex Optimizer
 Number of evaluations: 20
 (solver: 20, interpolation: 0)
 Initial goal function value = 0.1526051
 Last goal function value = ** aborted by user **
 Best goal function value = 0.00349321934319

Best parameters so far:

p2 = 0.149987
 p1 = 0.0707552



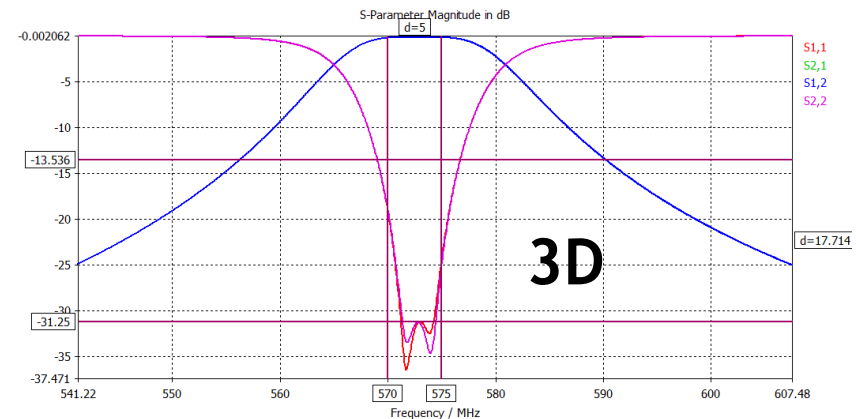
Bsp. 1: Results of the 2nd opt. loop



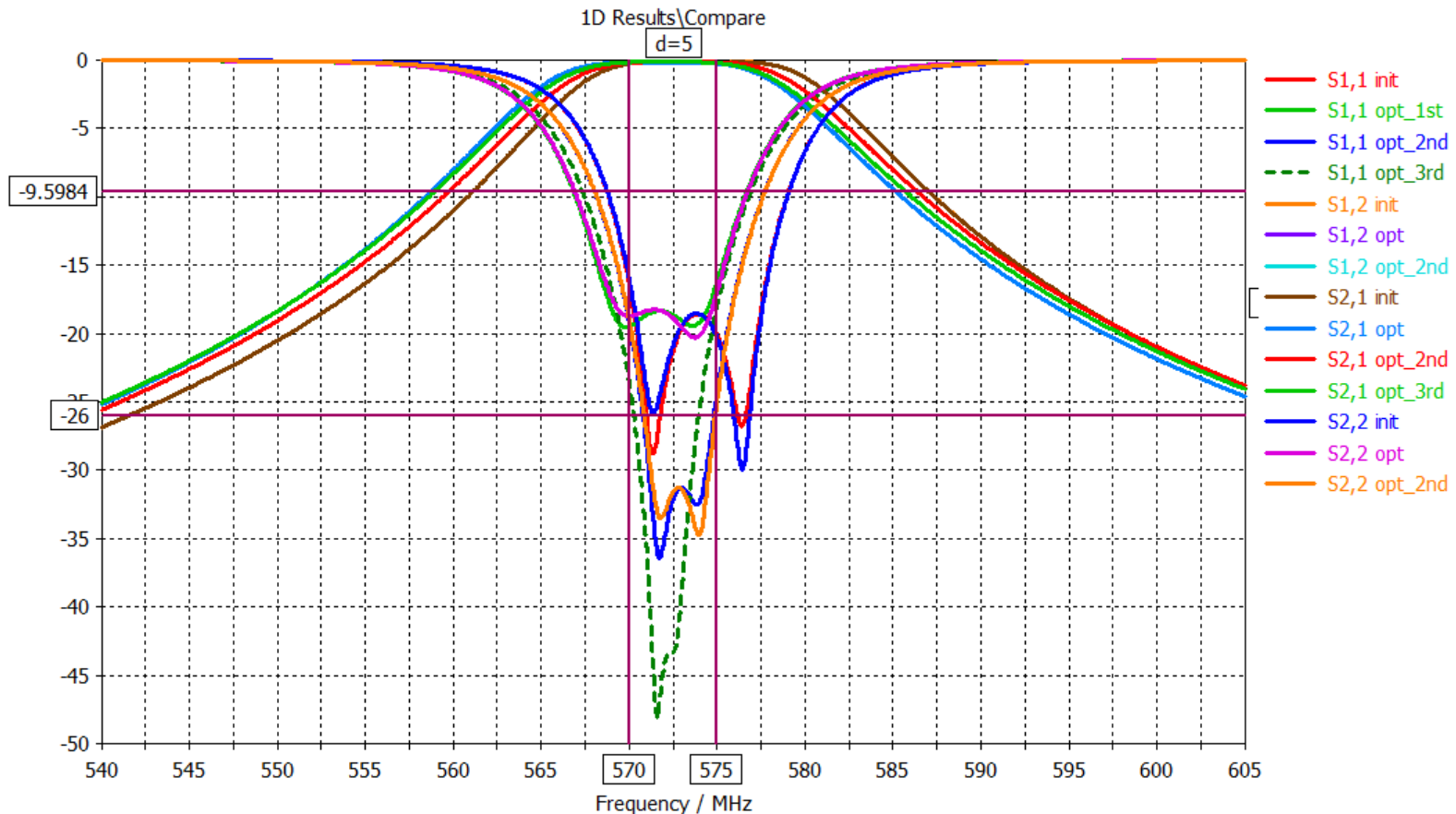
Opt.results

p1	-0.20237568682391
p2	0.046889685215927

Name	Value	Description
Input_coupling_1	6.3+p2	6.3
Input_coupling_2	6.3+p2	6.3
Res_tuner_1	0+p1	0, Varying post-high
Res_tuner_2	0+p1	0
aperture	-49.15	-52
facedistance_1	0	Parametrized face
facedistance_2	0	Parametrized face
p1	.071+0.2	
p2	.15+0.4700	



Bsp. 1: S-Parameter Results Comparison



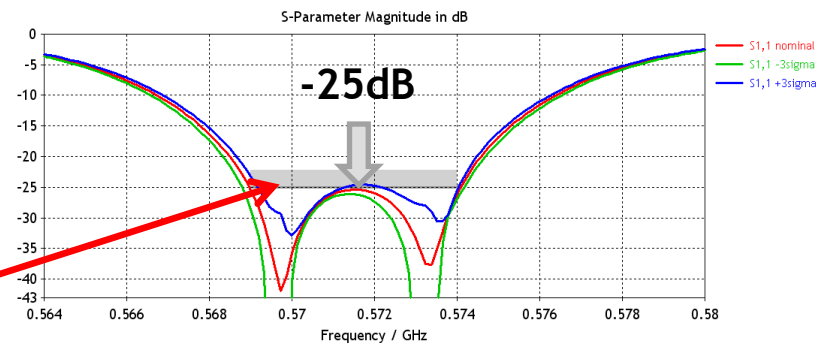
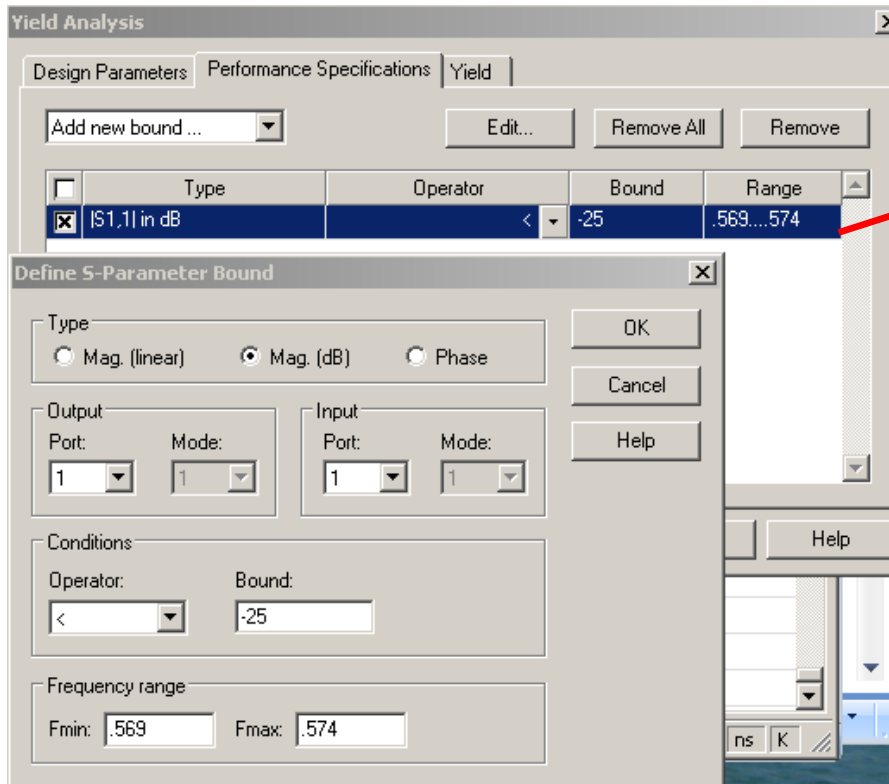
Several iteration steps are required since the linear sensitivity approach is applied to a nonlinear system (3D-filtermodel)

Bsp. 1: Yield Analysis

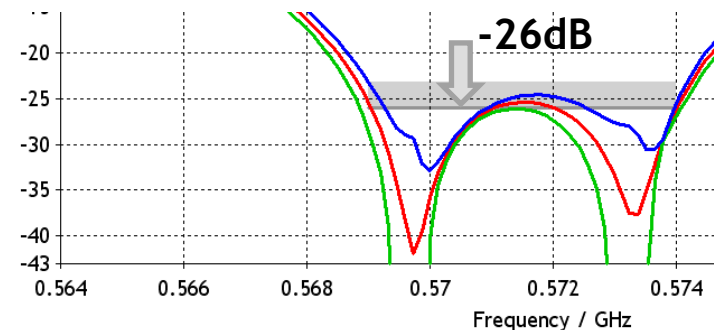
Yield-Specifications: **< -25dB**

In the range .569 to .574, S11 is under -25dB, the 3-sigma Lines are partially above and below -25dB

For this spec, the yield will tell us what percentage of the devices are within this limit of < -25 dB for the given frequency band.

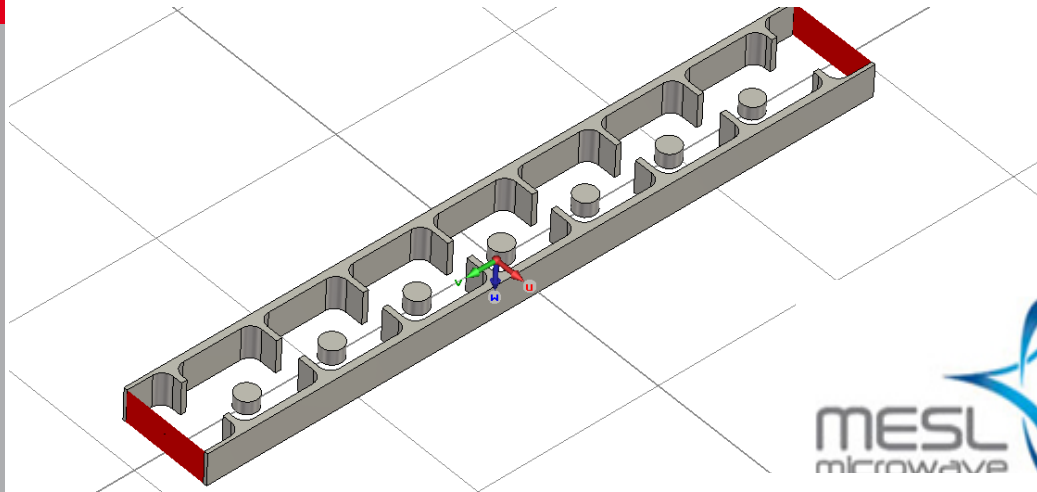


Calculated Yield: 91.66%

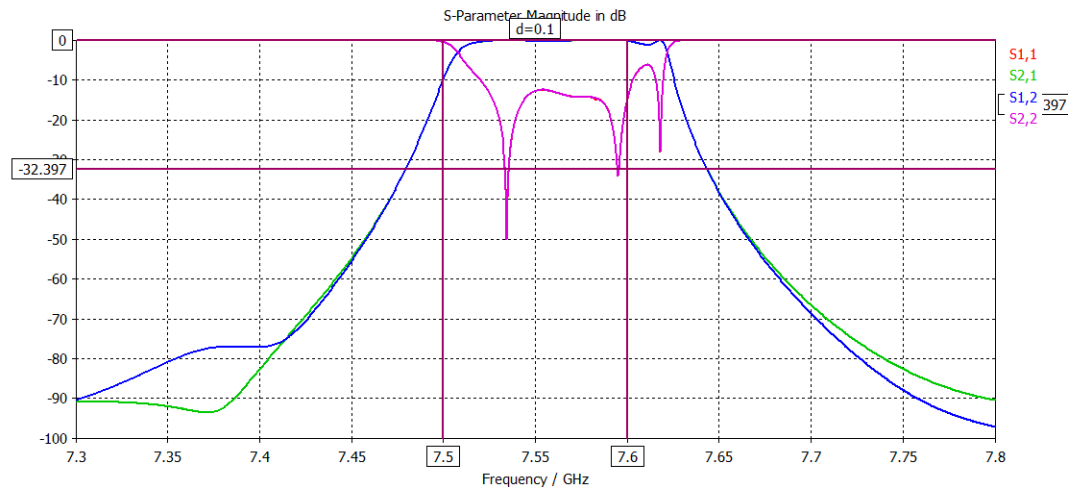
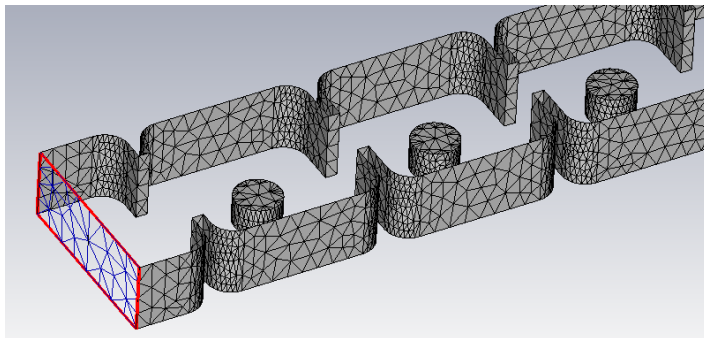


Calculated Yield: 0.71%

Beispiel 2: 7-Post Filter *)



Center frequency: 7.55 GHz
Bandwidth: 100 MHz
Rel Bandwidth: 1.3 %
Insertion Loss: -0.01 dB
Return Loss: - 26 dB

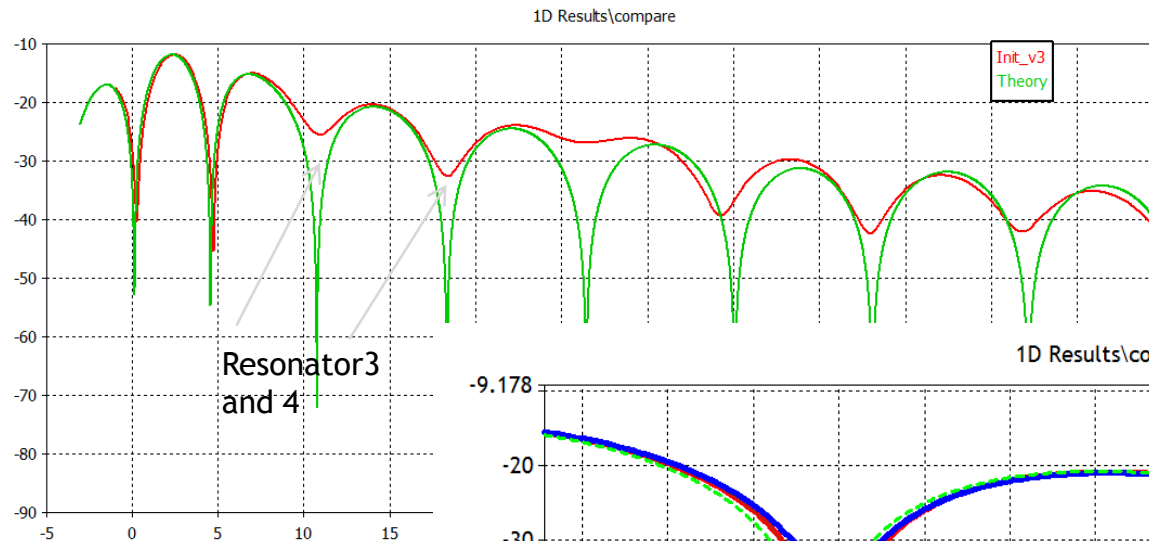


Best results obtained by Group delay Tuning

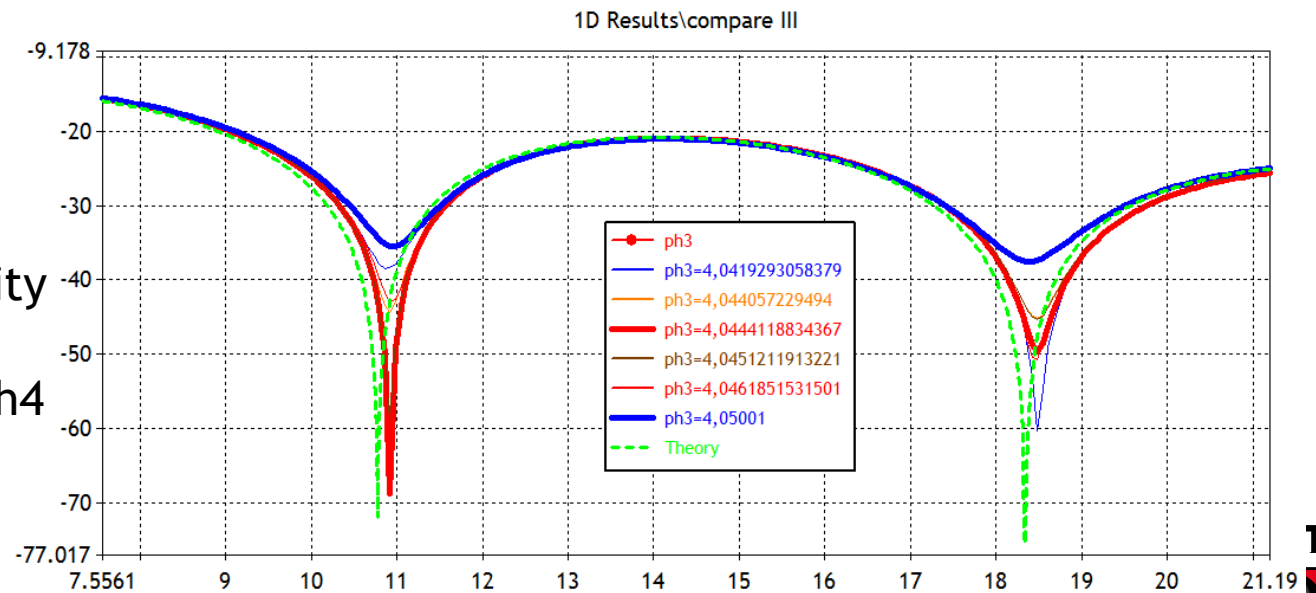
*) By courtesy of MESL Microwave Limited, Edinburgh

Bsp. 2: Method of InvChirpZ

Comparison of the InvChirpZ-resonse between groupdelayed tuned filter and Theory

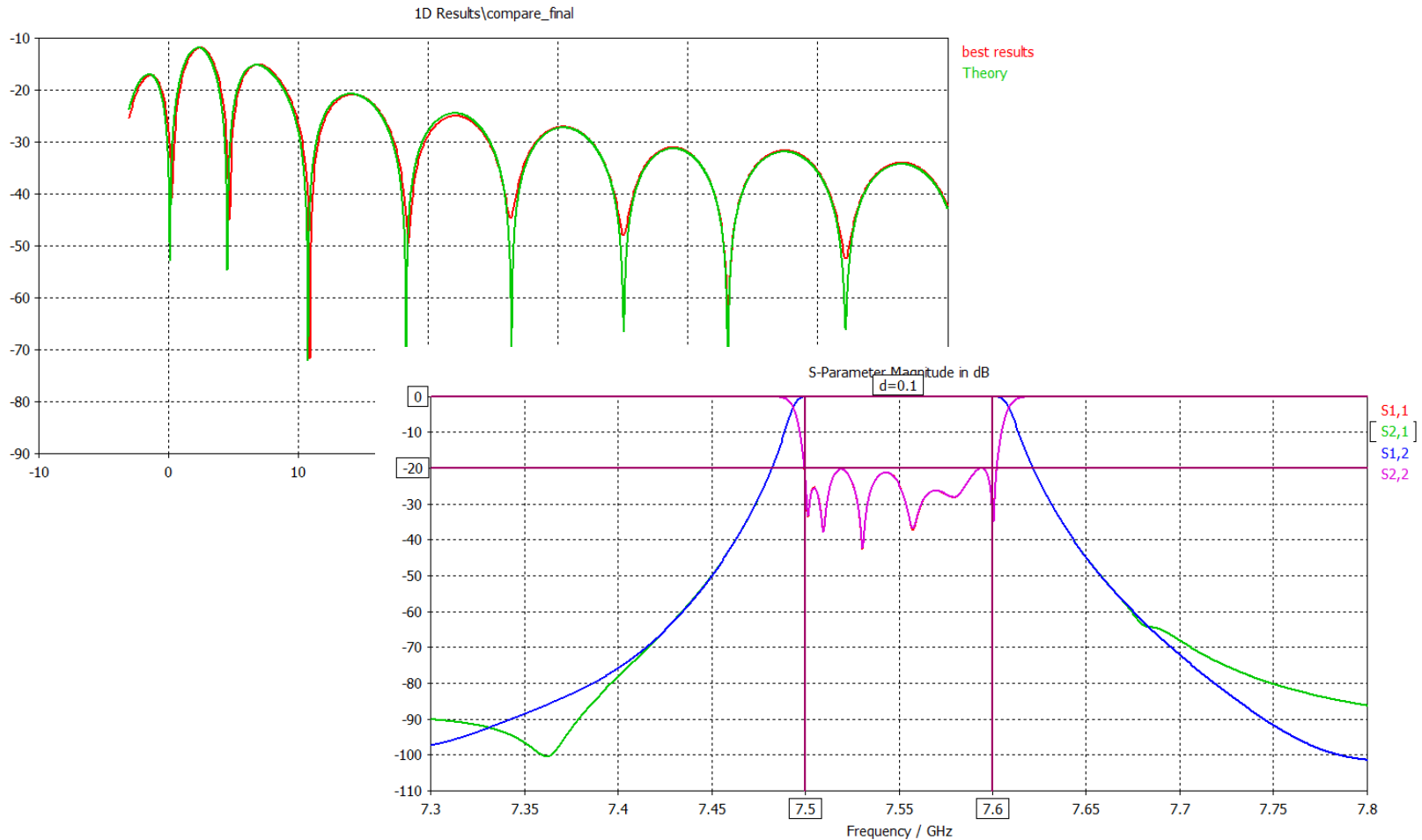


Note the high sensitivity at resonator 3 and 4 (parameter ph3 and ph4 respectively)



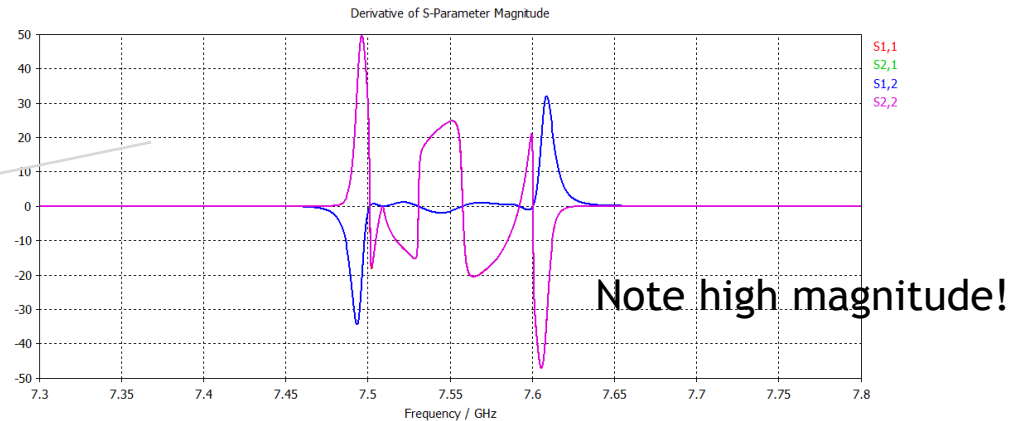
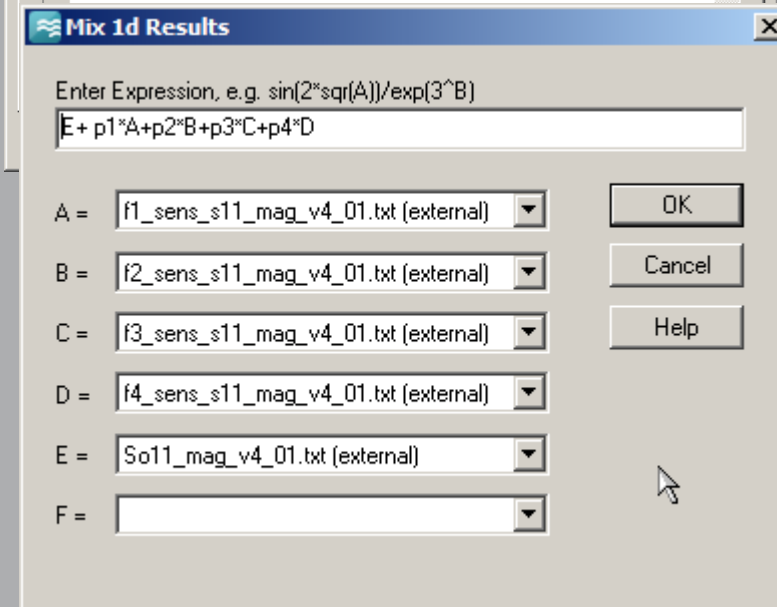
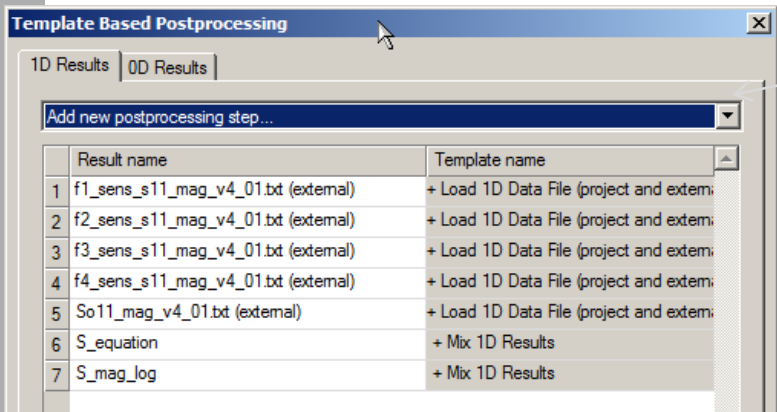
Bsp. 2: Method of InvChirpZ - final results

Final results by individually tuned posts

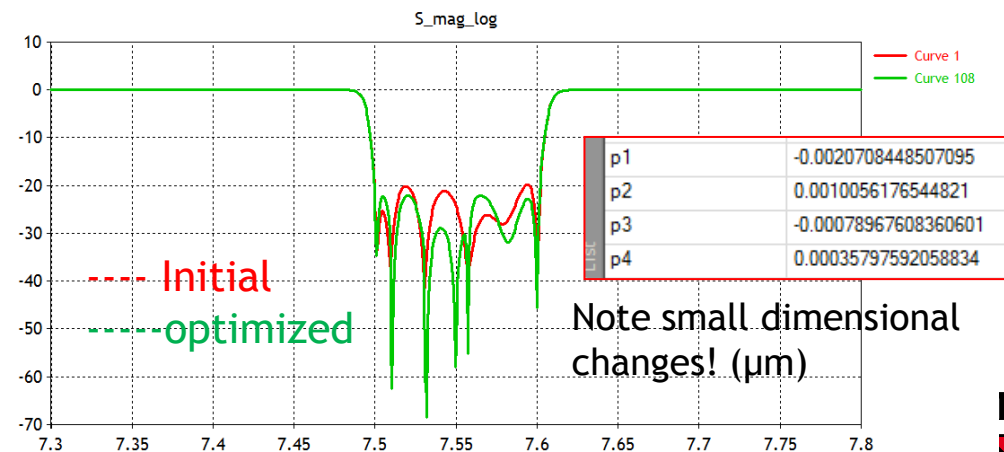


Bsp.2: Optimization using Sensitivity Data I

$$S_{11} = S_{11_3D_MWS} + \sum_p sens_p \cdot \Delta p$$



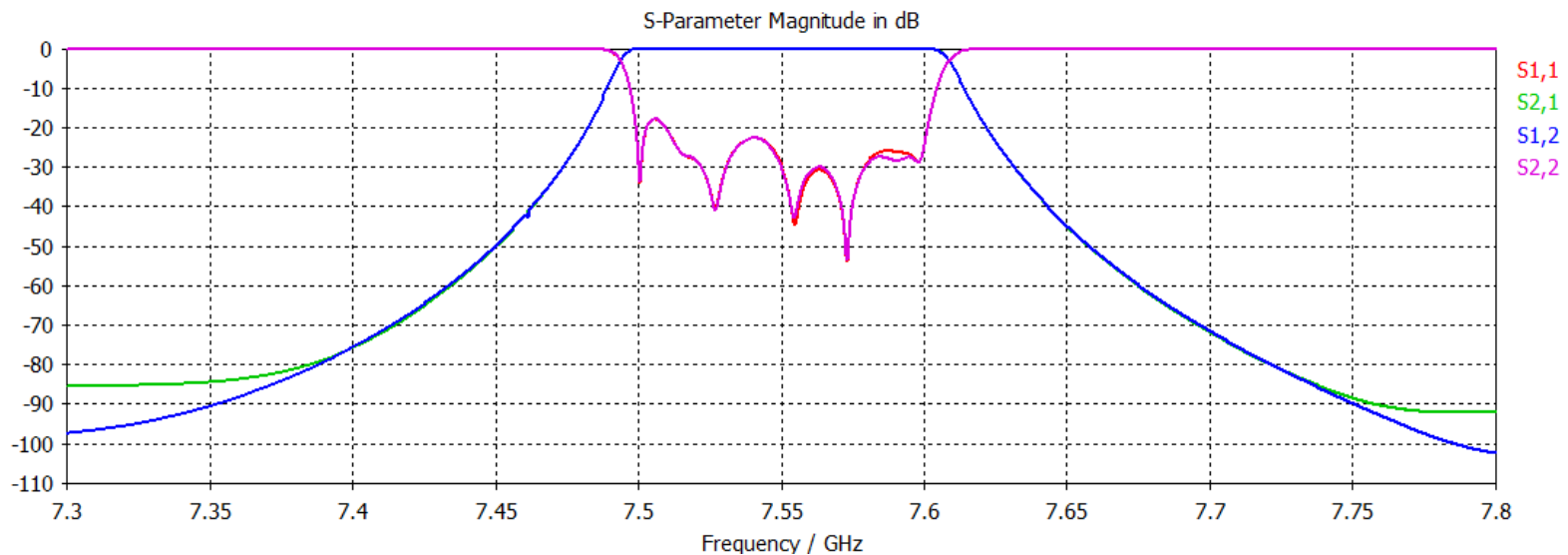
Results of the lin. optimization



Bsp. 2: Optimization using Sensitivity Data I

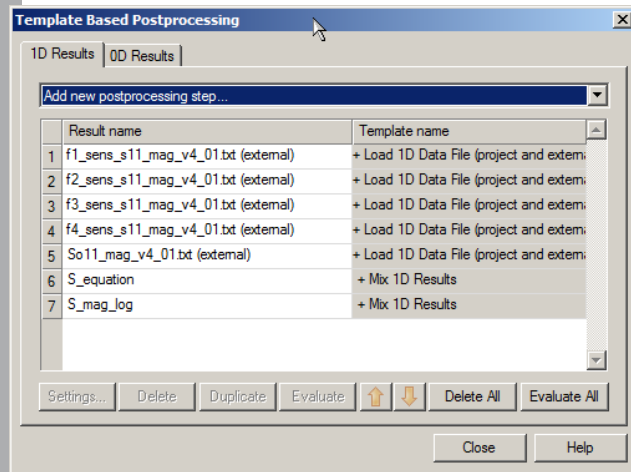
Modifying the geometry according to the found Delta-Ps of the lin. Opt. system:

facedistance_1	-3.5703713166821-0.0020708448507095
facedistance_2	-4.0022064900113+0.0010056176544821
facedistance_3	-4.0230879722653-0.00078967608360601
facedistance_4	-4.03+0.00035797592058834

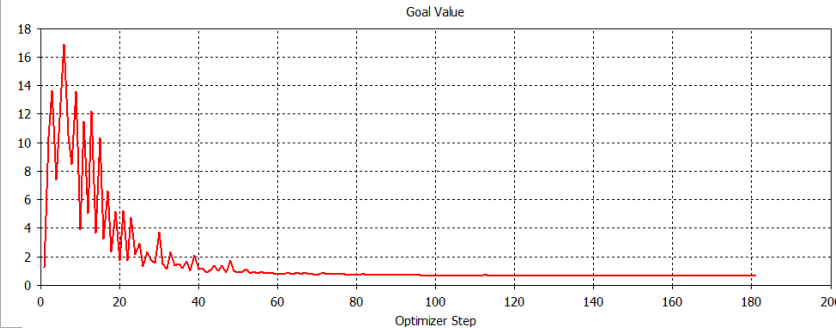
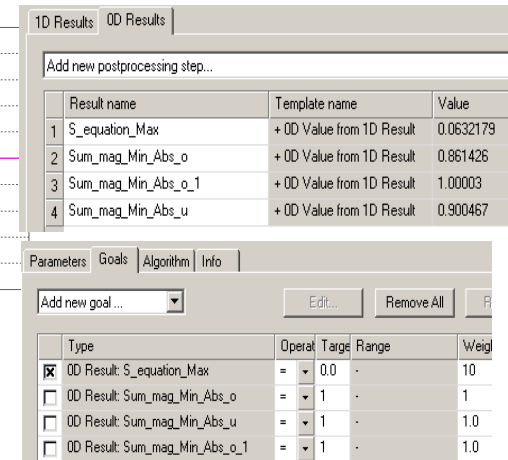
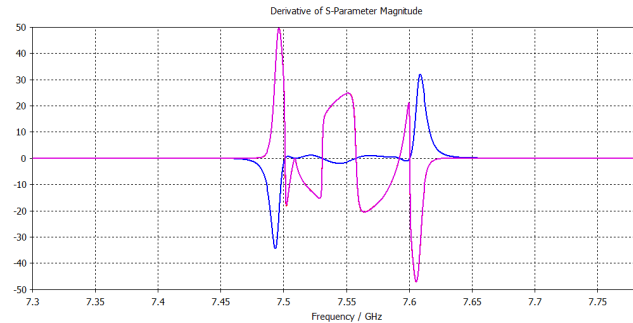


Result of the new 3D simulation

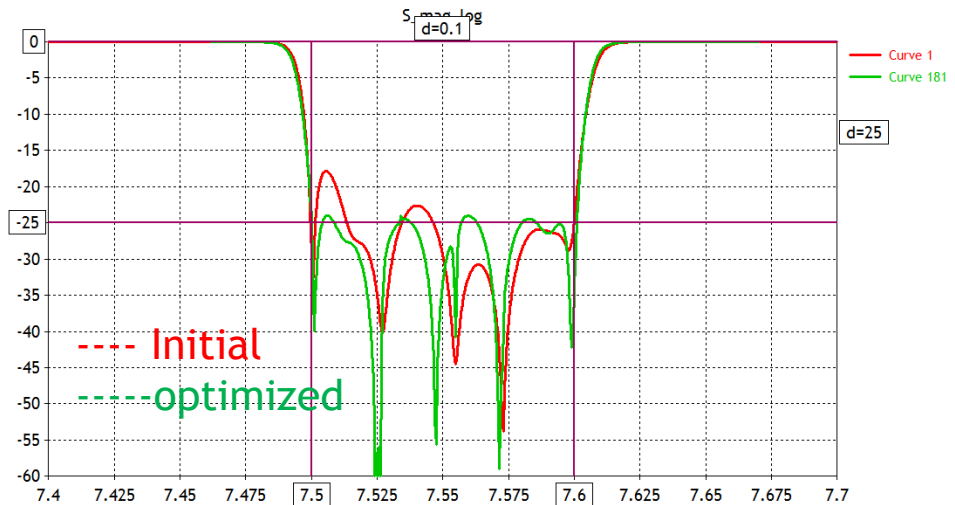
Bsp. 2: Optimization using Sensitivity Data II



Perform a 2nd optimisation loop



Sum of All Goals

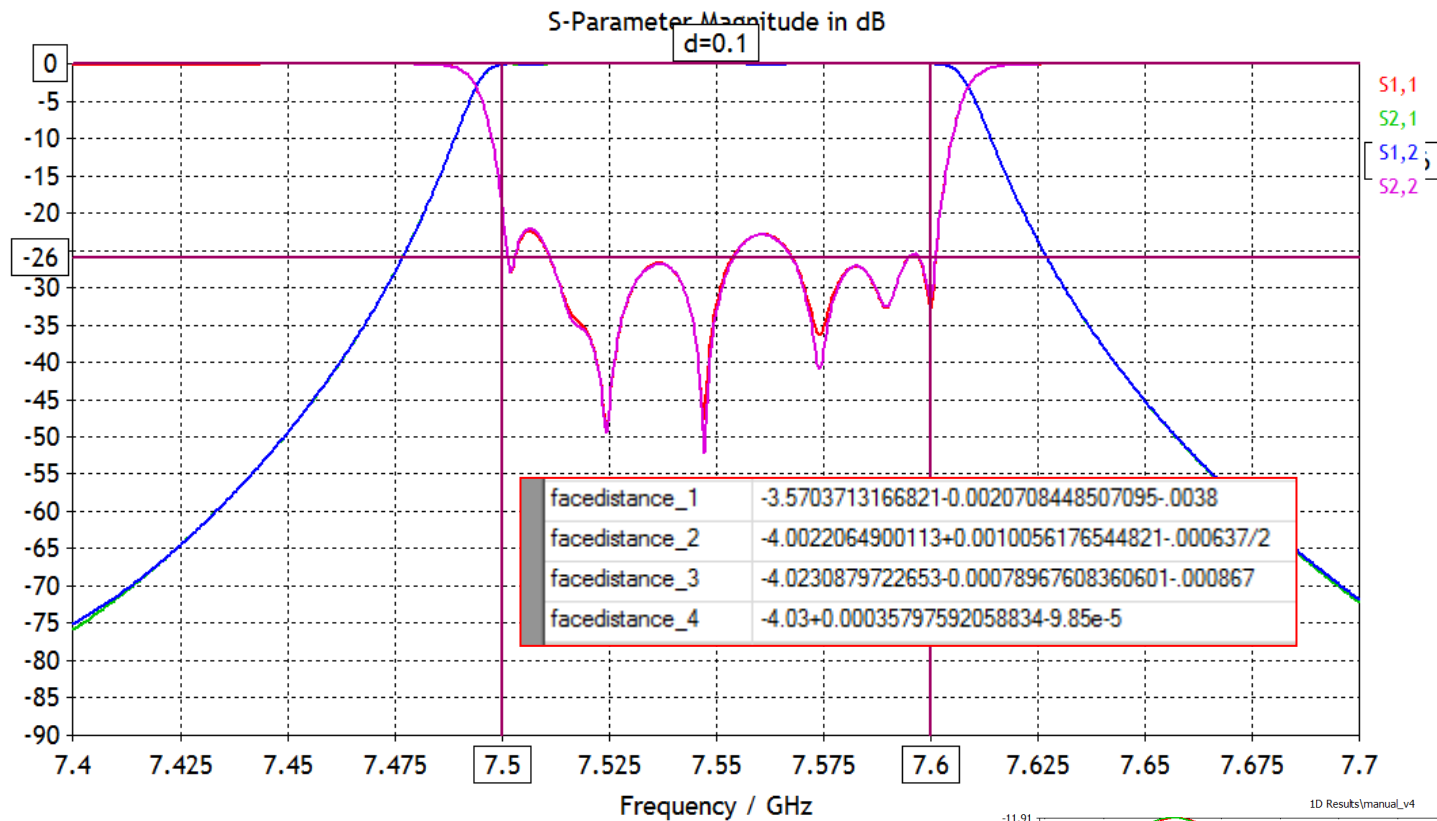


p1	-0.0073822379493485
p2	-0.001371471035572
p3	-0.0016749161719886
p4	-9.9118446524347e-005

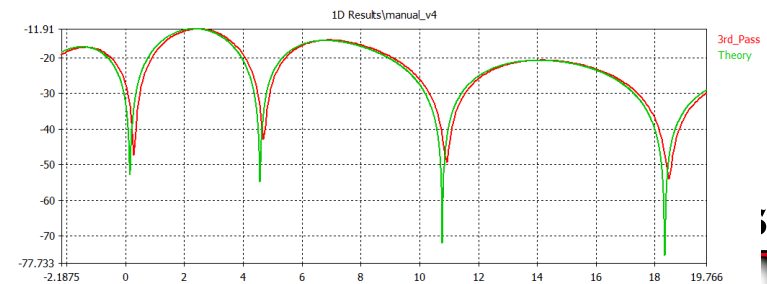
Result of new δP



Bsp.2: Optimization using Sensitivity Data II



Result of the new 3D simulation

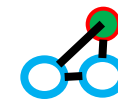
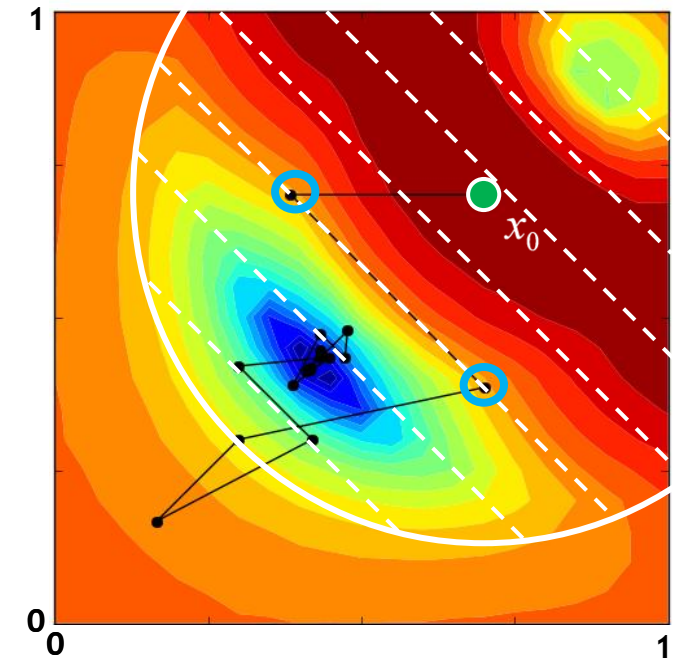


Trust Region Framework Algorithm (1)

- Choose initial point x_0
- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:

- Go to the minimizer (predicted optimum) of the model inside the trust-region
- Verify: Does the error decrease?
 - If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
 - If 'just' true, keep trust region radius and calculate new model
 - If not true, reduce size of trust region.



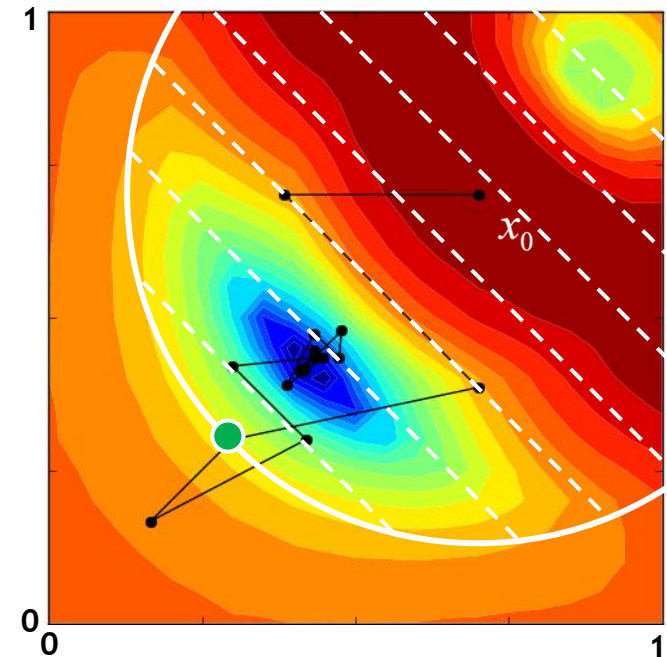
Define a plane by 3 points
(if no sensitivity data is
available)

Trust Region Framework Algorithm (2)

- Choose initial point x_0
- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:

- Go to the minimizer (predicted optimum) of the model inside the trust-region
- Verify: Does the error decrease?
 - If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
 - If 'just' true, keep trust region radius and calculate new model
 - If not true, reduce size of trust region.

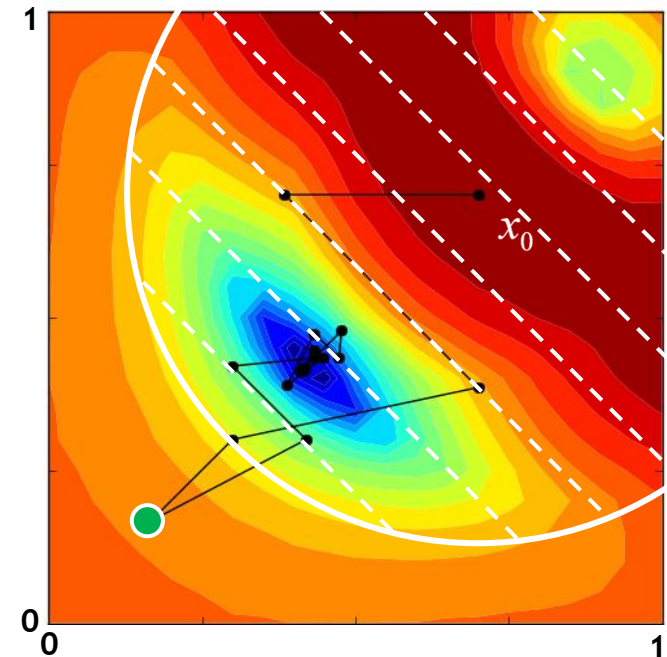


Trust Region Framework Algorithm (3)

- Choose initial point x_0
- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:

- Go to the minimizer (predicted optimum) of the model inside the trust-region
- **Verify: Does the error decrease?**
 - **If true, and if the model is very good, go further until quality gets worse,** take last point as new center. Reduce trust region radius and calculate new model
 - **If 'just' true,** keep trust region radius and calculate new model
 - **If not true,** reduce size of trust region.

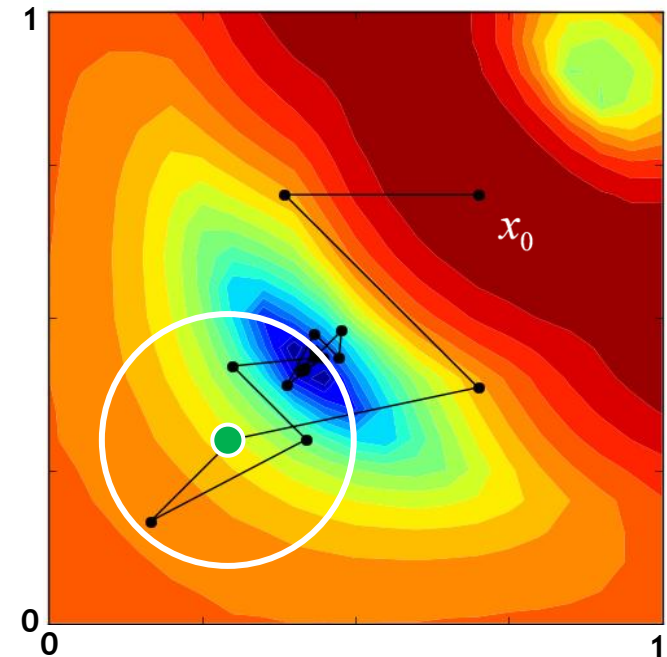


Trust Region Framework Algorithm (4)

- Choose initial point x_0
- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:

- Go to the minimizer (predicted optimum) of the model inside the trust-region
- Verify: Does the error decrease?
 - If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
 - If 'just' true, keep trust region radius and calculate new model
 - If not true, reduce size of trust region.

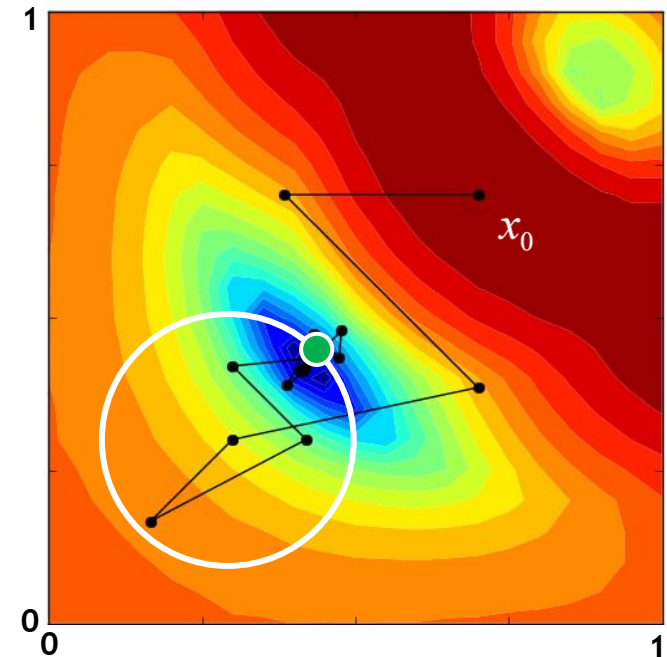


Trust Region Framework Algorithm (5)

- Choose initial point x_0
- Create a local linear model around that point, and define an initial 'trust region radius', an area in which we think the model is good.

Repeat:

- Go to the minimizer (predicted optimum) of the model inside the trust-region
- Verify: Does the error decrease?
 - If true, and if the model is very good, go further until quality gets worse, take last point as new center. Reduce trust region radius and calculate new model
 - If 'just' true, keep trust region radius and calculate new model
 - If not true, reduce size of trust region.



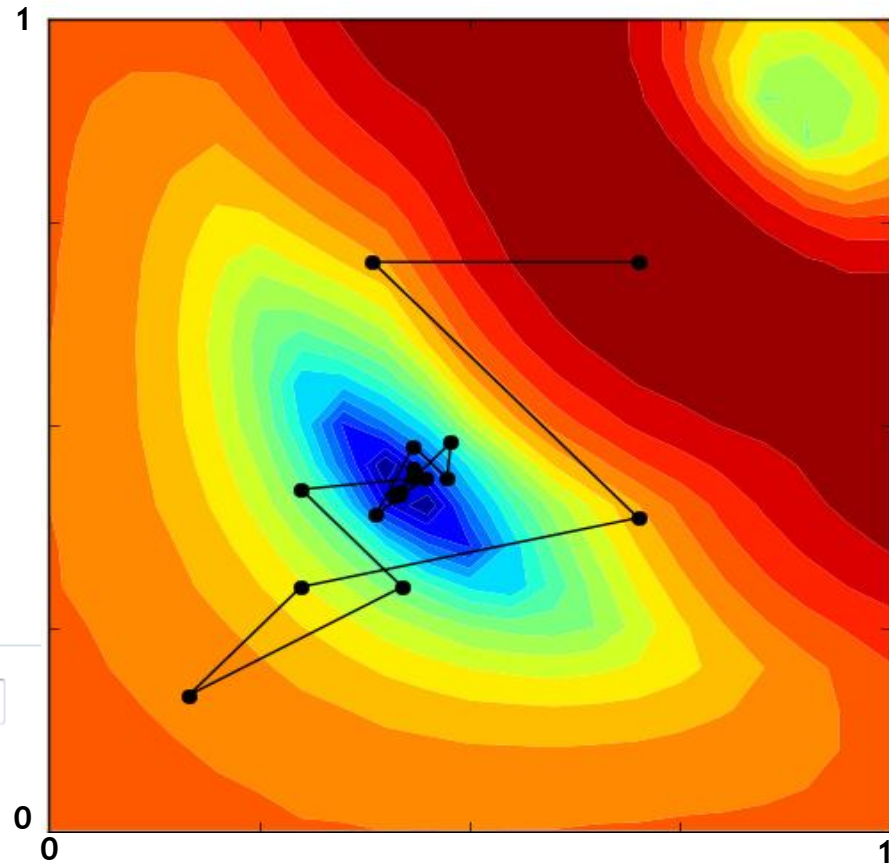
Trust Region Framework Algorithm (6)

... “The algorithm will be converged once the trust region radius or distance to the next predicted optimum becomes smaller than the specified domain accuracy.”

Algorithm: Trust Region Framework Properties...

Algorithm settings

Reset min/max 10 % of initial value Domain accuracy: 1.e-4

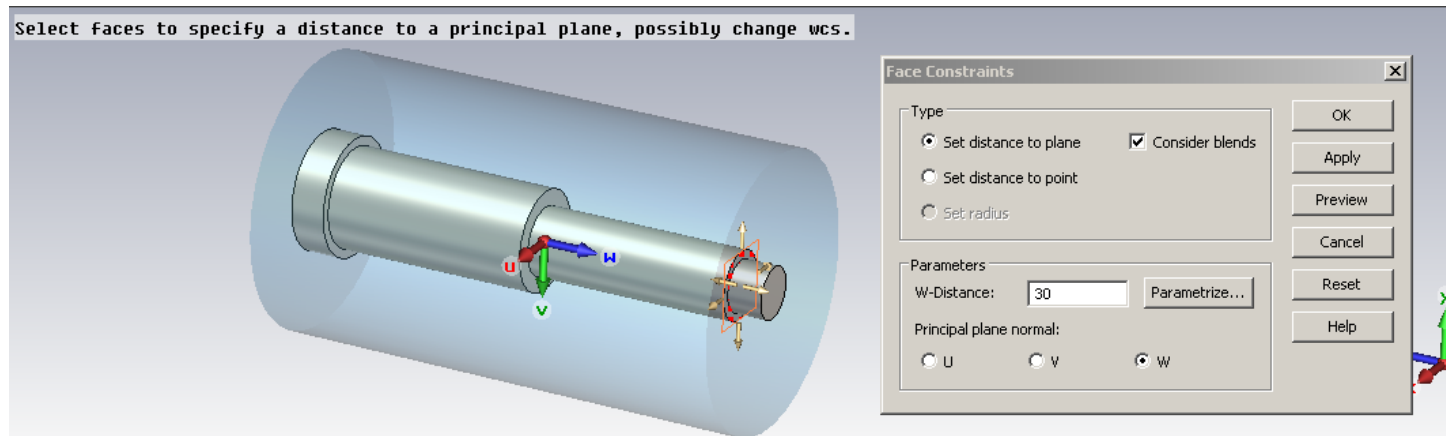


Agenda

- Einführung
- Übersicht der verwendeten Optimierungsalgorithmen
 - Genetic Algorithm
 - Particle Swarm Optimization
 - Nelder-Mead Simplex
 - Interpolated Quasi-Newton
 - Trust Region Framework
- Sensitivität /Yield
- **Beispiele**
 - Cavity und Iris gekoppelte Bandpass Filter
 - Zweistufiger Impedanztrafo
 - Planar filter tuning
- Zusammenfassung

Bsp. 3: a) Initial Sensitivity Model

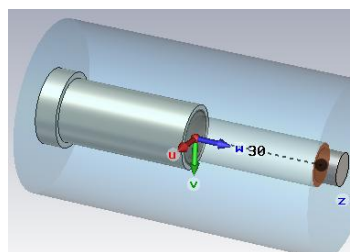
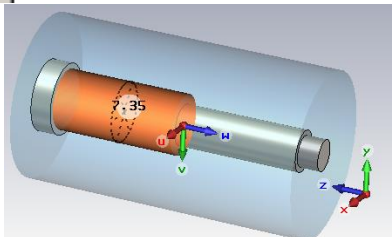
Setup of face-constraints required for the TrustRegionFramework Optimizer



Show Face Constraints

Face	Parameter	Value	Type
component1:50ohmstück;1	facedistance_1	-30	distance to plane
component1:Stufe1;2	faceradius_1	7.35	radius of face
component1:Stufe2;1	facedistance_2	30	distance to plane
component1:Stufe2;2	faceradius_2	5	radius of face

Name	Value	Description
L1	30	30
L2	30	30
R1	7.35	$R_a/\exp(60/60)$ (start 7.35)
R2	5	$R_a/\exp(82.44/60)$ (start 5)
Ra	20	Defined as initial Parameter
facedistance_1	-30	Parametrized face of component1:50ohmstück
facedistance_2	30	Parametrized face of component1:Stufe2
faceradius_1	7.35	Parametrized face of component1:Stufe1
faceradius_2	5	Parametrized face of component1:Stufe2
r100	$R_a/\exp(100/60)$	100 Ohm Port
r50	$R_a/\exp(50/60)$	50 Ohm Port



Bsp. 3: b) Initial Sensitivity Model

Frequency Domain Solver Setup + Sensitivity results

Frequency Domain Solver Parameters

Method

- ☒ General Purpose
- ☐ Resonant: Fast S-Parameter
- ☐ Resonant: S-Parameter, fields

Mesh type: **Tetrahedral Mesh**

Solver settings

- ☐ Save all field results
- ☐ Store result data in cache
- ☐ Calculate modes only
- Accuracy (tetrahedral mesh): **1e-4**

Excitation settings

Source type: **All Ports** Mode: **1**

S-parameter settings

- ☐ Normalize to fixed impedance
- 50** Ohms

Frequency samples

	Auto	Samples	From	To	Unit
Max. Range	☒	1	1.5	3.5	GHz
Adapt. Freq.	☒				GHz
Frequency	☒				GHz
Frequency	☐				GHz
Frequency	☐				GHz

☐ Do not calculate field monitors

☒ Use broadband frequency sweep

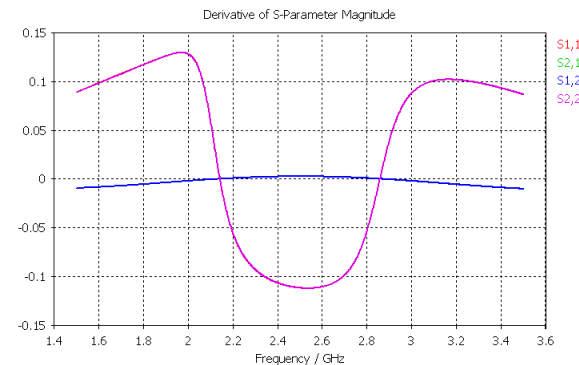
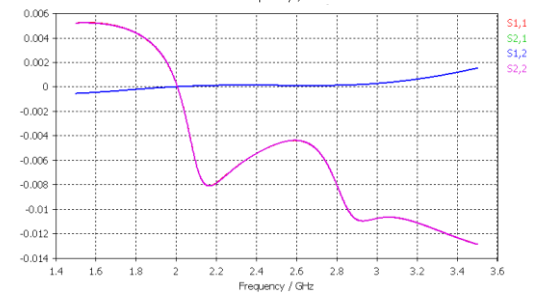
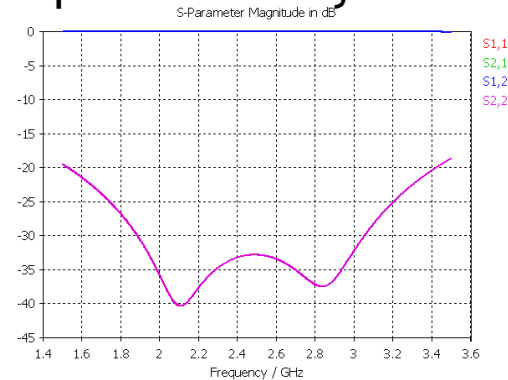
Adaptive mesh refinement

- ☐ Adaptive tetrahedral mesh refinement

Sensitivity analysis

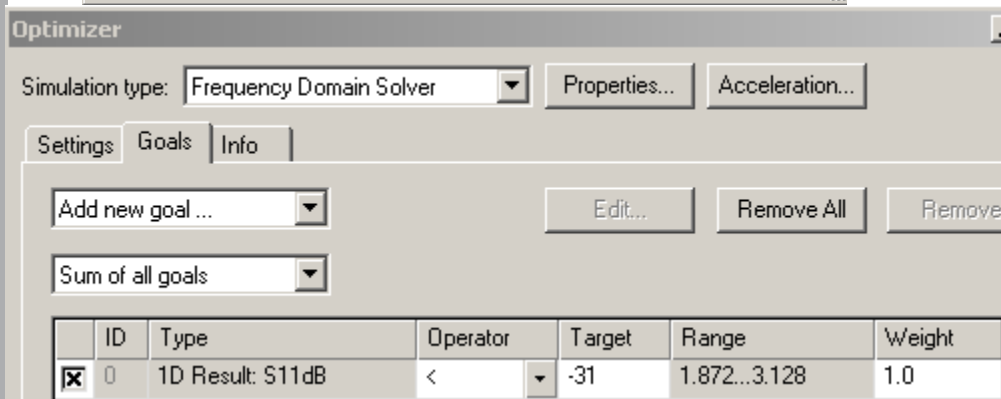
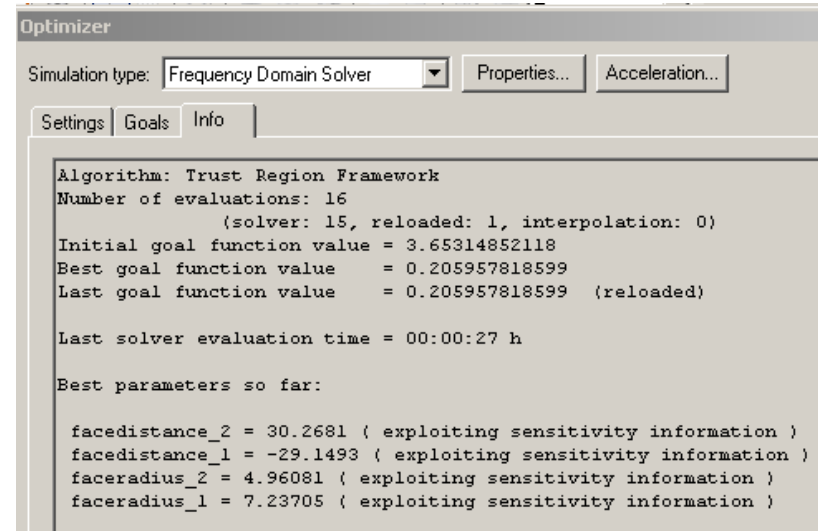
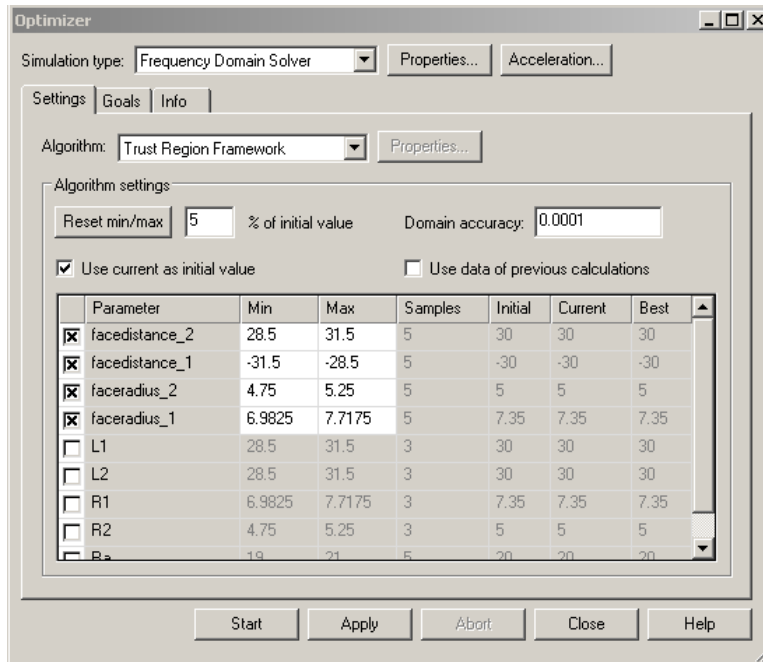
- ☒ Use sensitivity analysis

Buttons: Start, Optimize..., Par. Sweep..., Acceleration..., Specials..., Simplify Model..., Apply, Close, Help

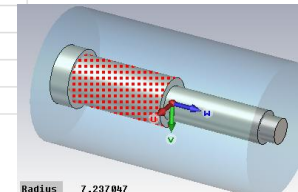


Bsp.3: c) Optimized Sens. Model

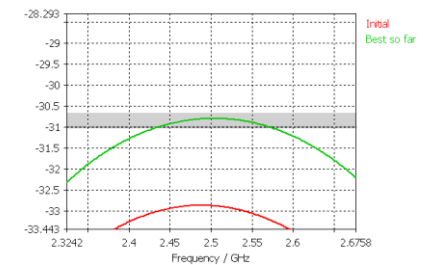
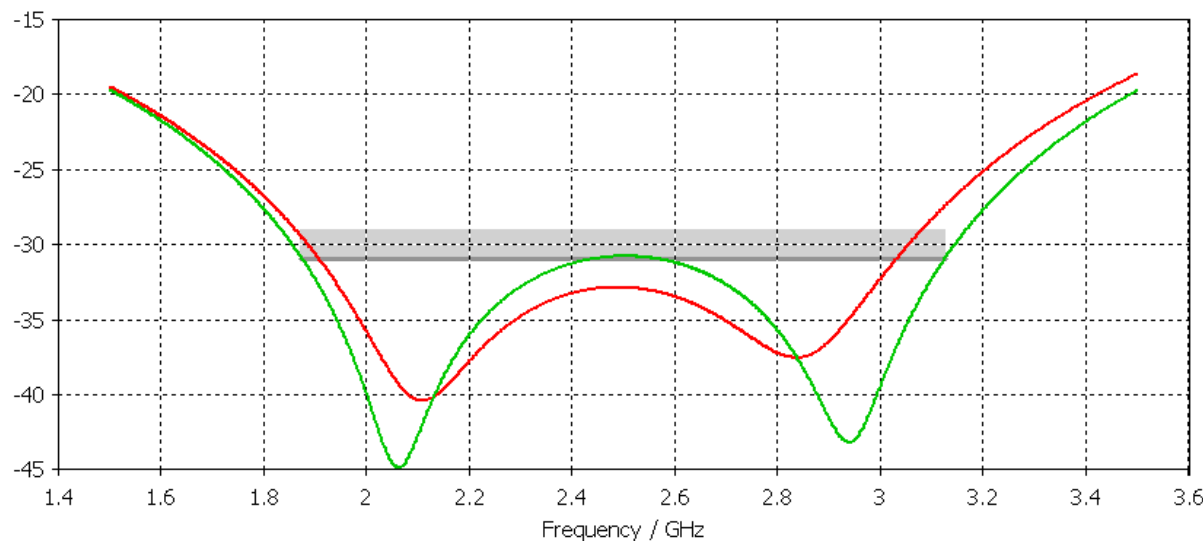
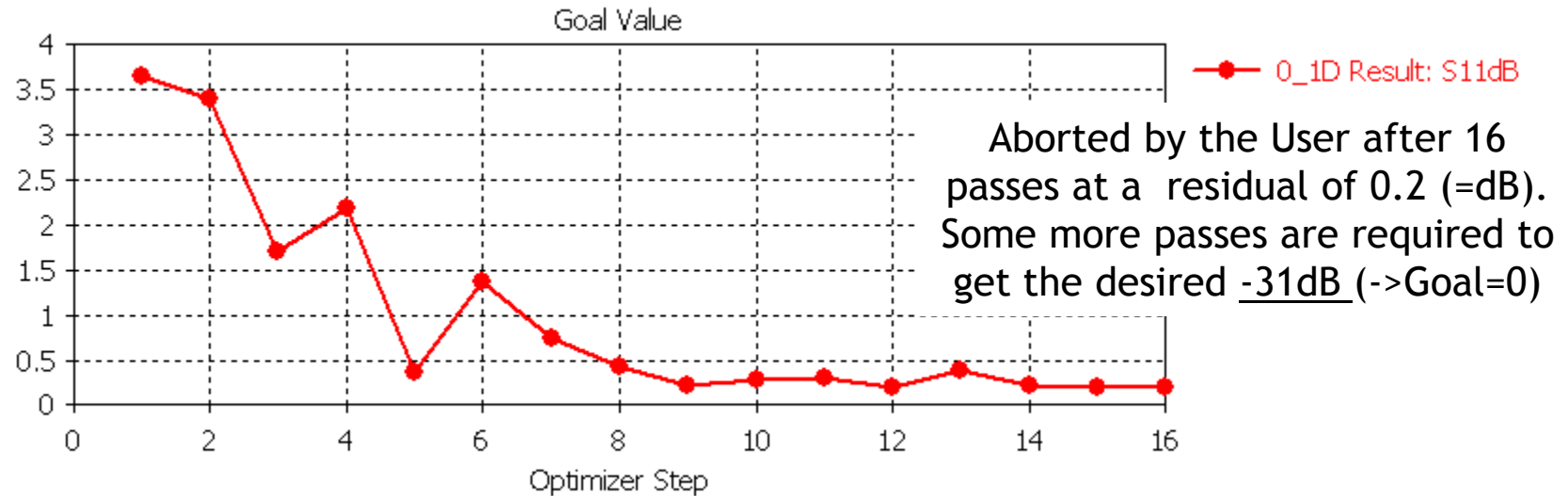
Using TRF Optimizer for a 3D Model-Optimization



Name	Value	Description
L1	30	30
L2	30	30
R1	7.35	Ra/exp(60/60) (start 7.35)
R2	5	Ra/exp(82.44/60) (start 5)
Ra	20	Defined as initial Parameter
facedistance_1	-29.149303182698	Parametrized face of component1:50ohmstück
facedistance_2	30.268117799564	Parametrized face of component1:Stufe2
faceradius_1	7.2370465181744	
faceradius_2	4.9608056191913	
r100	Ra/exp(100/60)	
r50	Ra/exp(50/60)	

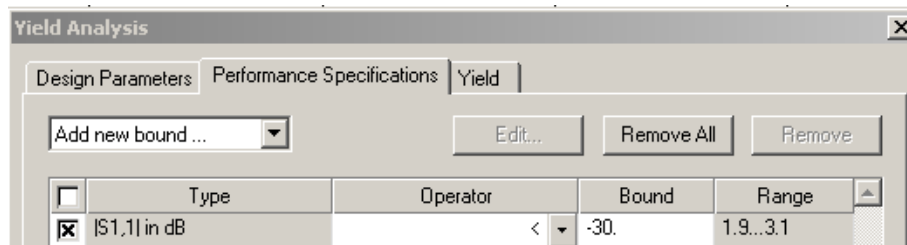


Bsp.3: d) Optimized Sens. Model

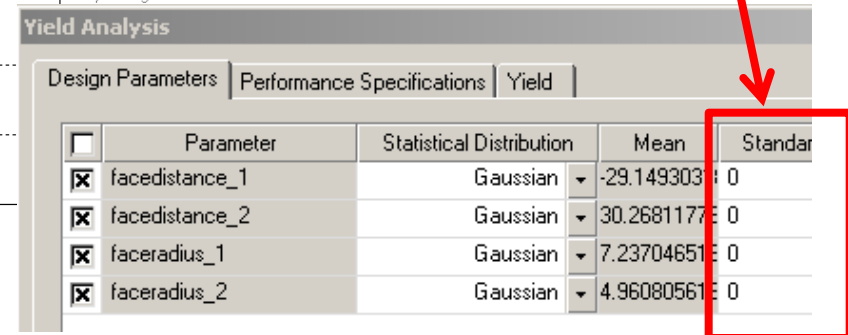
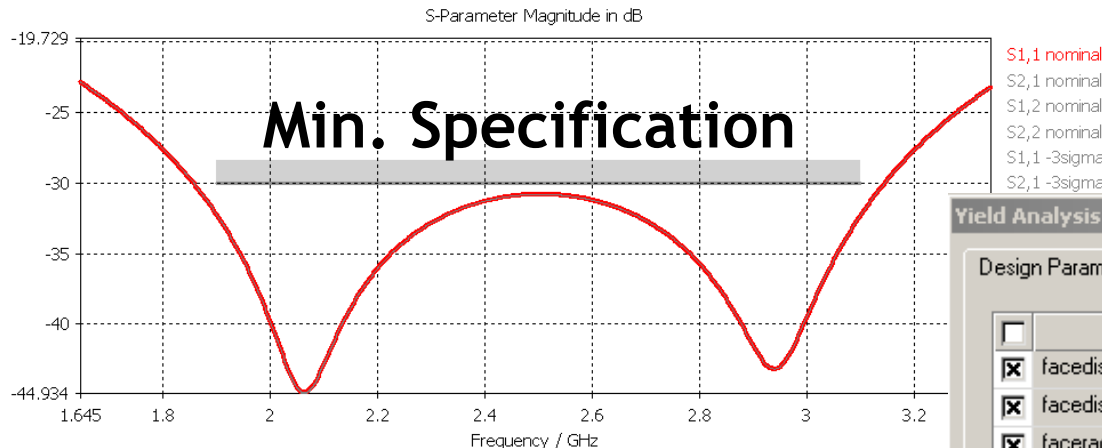


Bsp. 3: e) Initial PostProcessingTemplate Model

By releasing the specifications a bit (-30dB and a smaller bandwidth), standard-deviations > 0 can be optimized such, that the spec is still satisfied.



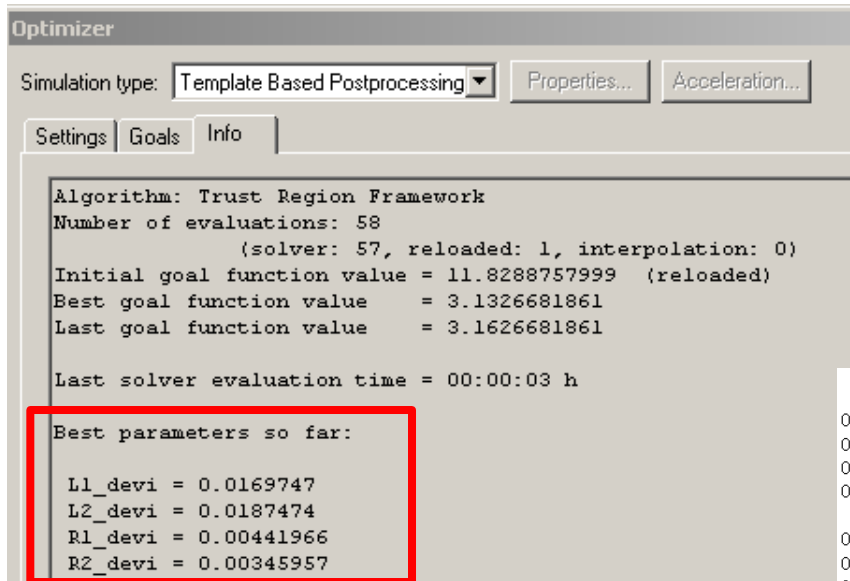
Goal is to find the maximal possible deviations?



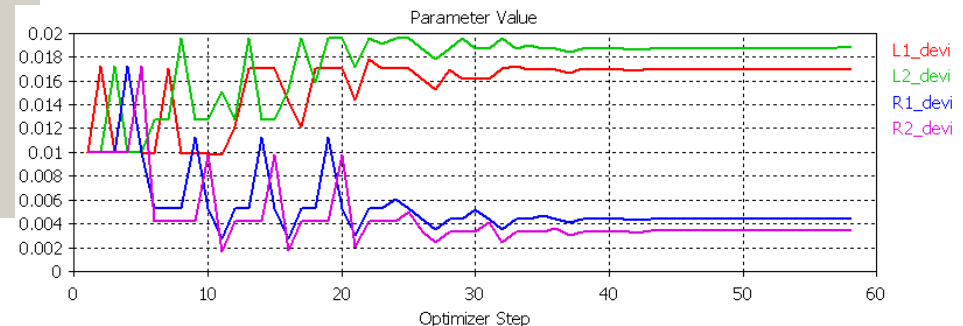
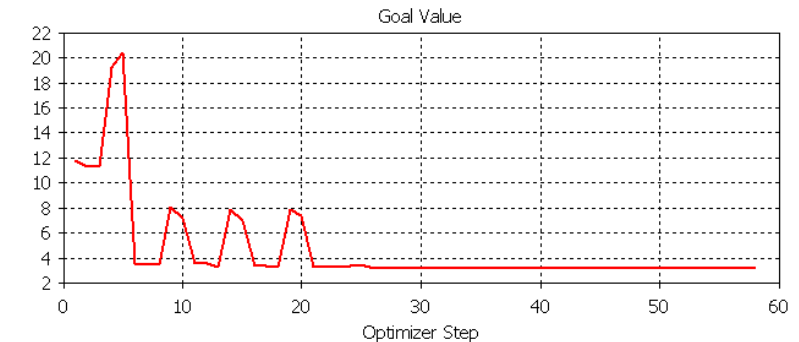
The 'Yield Analysis' dialog box shows the 'Yield' tab with a table of parameters. A red box highlights the 'Mean' and 'Standard' columns. A red arrow points from the text 'Goal is to find the maximal possible deviations?' to this box.

Parameter	Statistical Distribution	Mean	Standard
☒ facedistance_1	Gaussian	-29.149303	0
☒ facedistance_2	Gaussian	30.268117	0
☒ faceradius_1	Gaussian	7.2370465	0
☒ faceradius_2	Gaussian	4.9608056	0

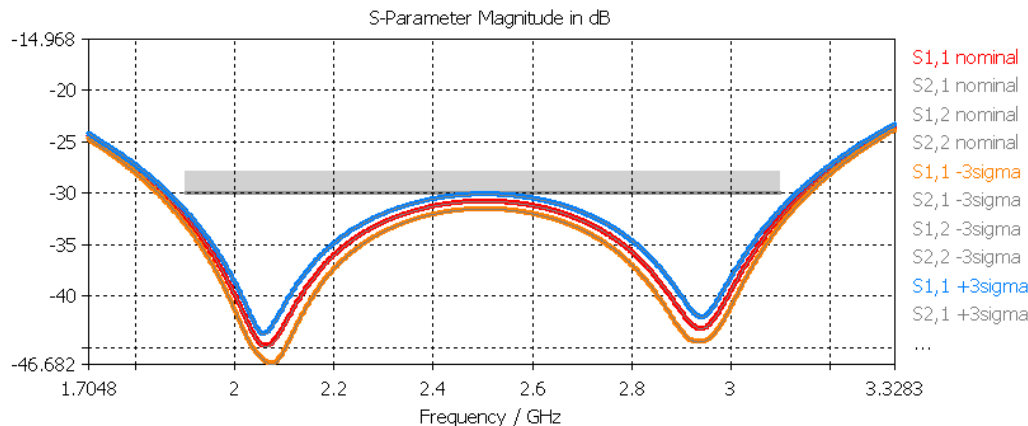
Bsp.3: f) Optim. PostProcessingTemplate Model



The radii are factor ~4 in sigma more critical than the pertinent lengths



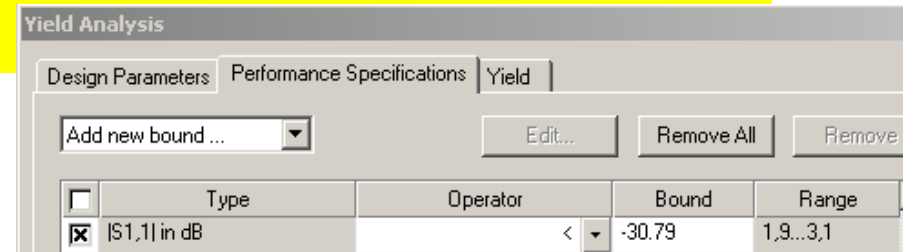
Check: Specification not violated



Bsp.3: a) Initial PostProcessingTemplate Model

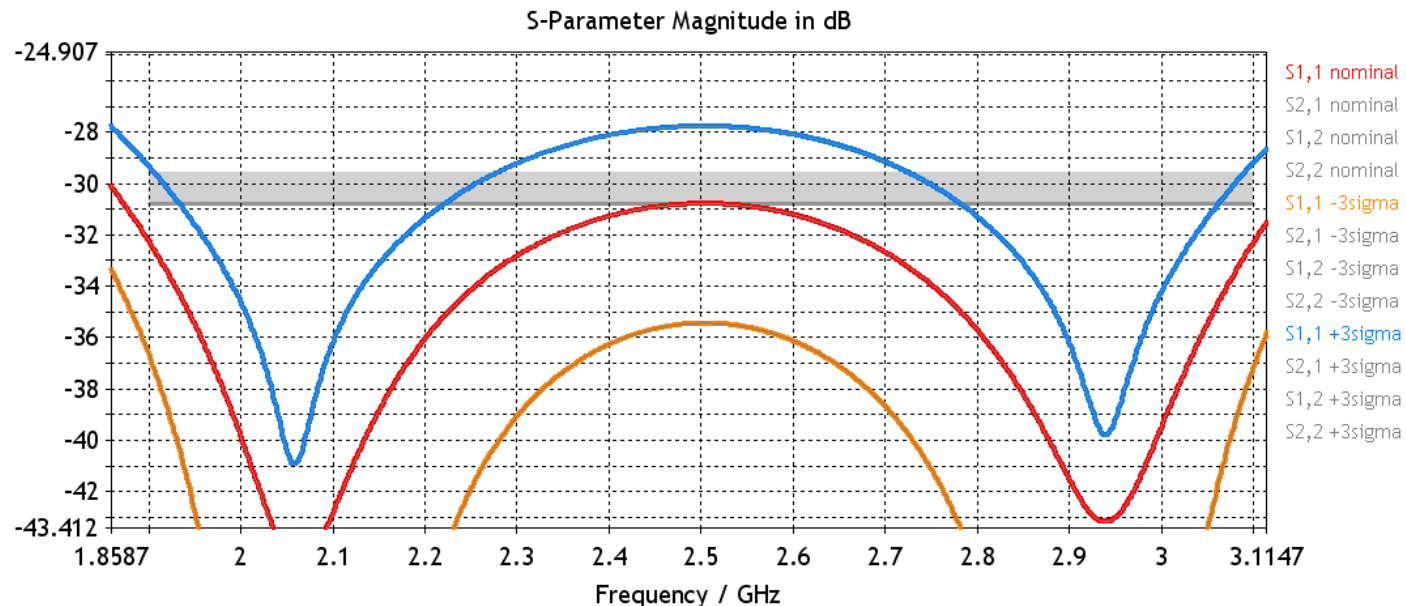
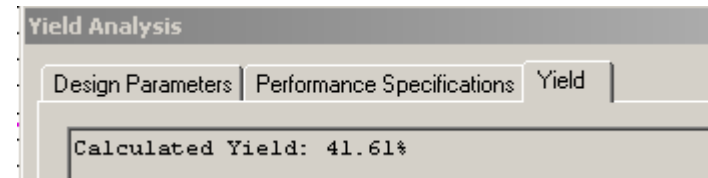
Goal is to find the best parameters to maximize the yield

Yield is calculated for the initial set of constraints (=to satisfy the S-Parameter specifications)



facedistance_1	-29.149303182698
facedistance_2	30.268117799564
faceradius_1	7.2370465181744
faceradius_2	4.9608056191913

Parametrized face of component1:50ohmstück
 Parametrized face of component1:Stufe2
 Parametrized face of component1:Stufe1
 Parametrized face of component1:Stufe2



Bsp. 3: b) Result of the TRF-Optimization

Optimizer

Simulation type: **Frequency Domain Solver** **Properties...** **Acceleration...**

Settings **Goals** **Info**

Algorithm: Trust Region Framework
 Number of evaluations: 48
 (solver: 46, reloaded: 2, interpolation: 0)
 Initial goal function value = 0.5856 (reloaded)
 Best goal function value = 0.487
 Last goal function value = 0.487 (reloaded)
 Last solver evaluation time = 00:00:30 h

Best parameters so far:

facedistance_1 = -29.1526
 facedistance_2 = 30.2477
 faceradius_1 = 7.2509
 faceradius_2 = 4.95522

The new dimensions to achieve the max yield:

Yield Analysis

Design Parameters | Performance Specifications | Yield

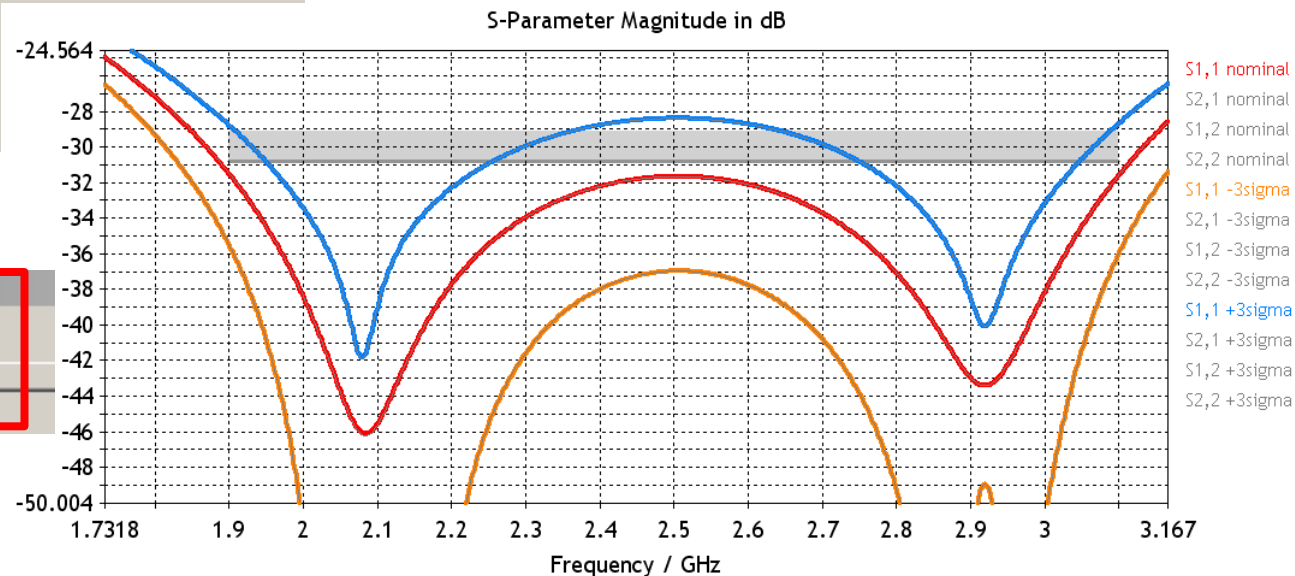
☐	Parameter	Statistical Distribution	Mean	Standard Dev
☒	facedistance_1	Gaussian	-29.1525845	0,02
☒	facedistance_2	Gaussian	30.24767343	0,02
☒	faceradius_1	Gaussian	7.250895486	0,02
☒	faceradius_2	Gaussian	4.955222310	0,02

Check: 1 - goal = yield:

Yield Analysis

Design Parameters | Performance Specifications | Yield

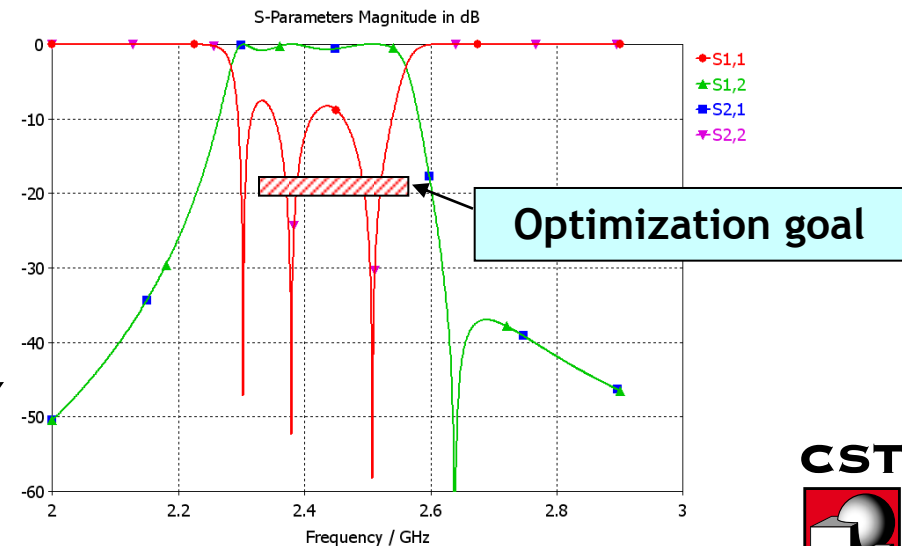
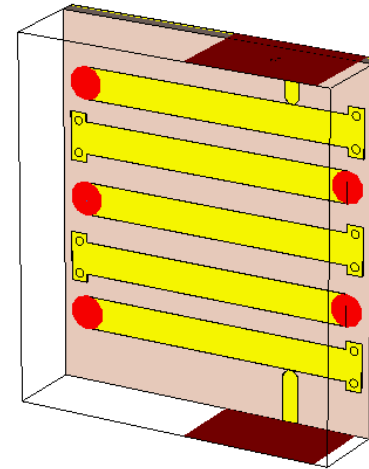
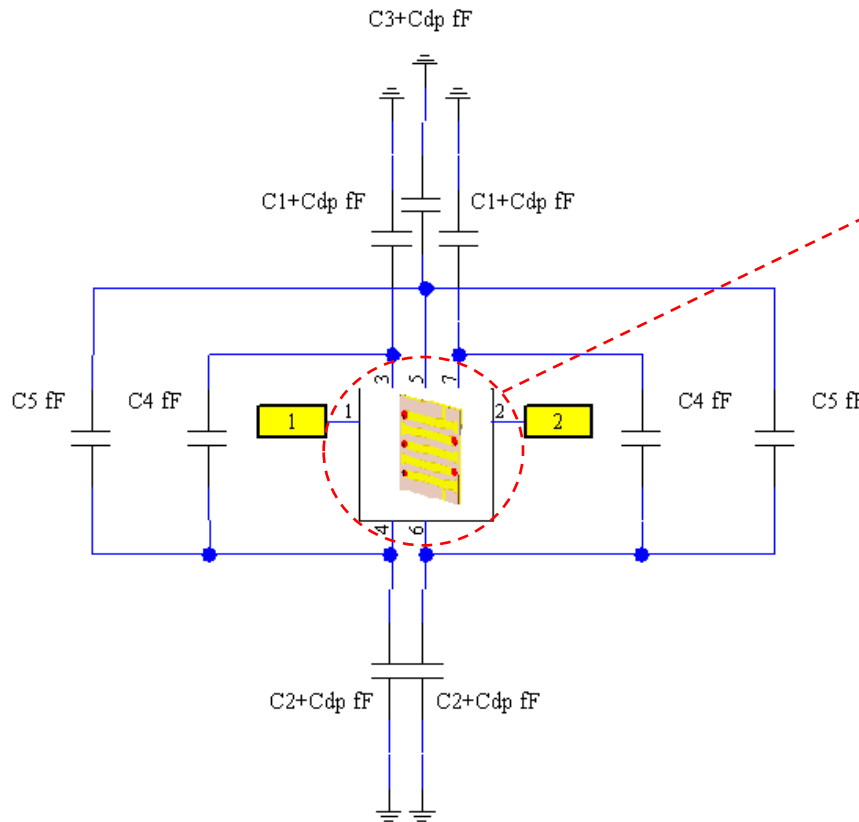
Calculated Yield: 50.94%



Agenda

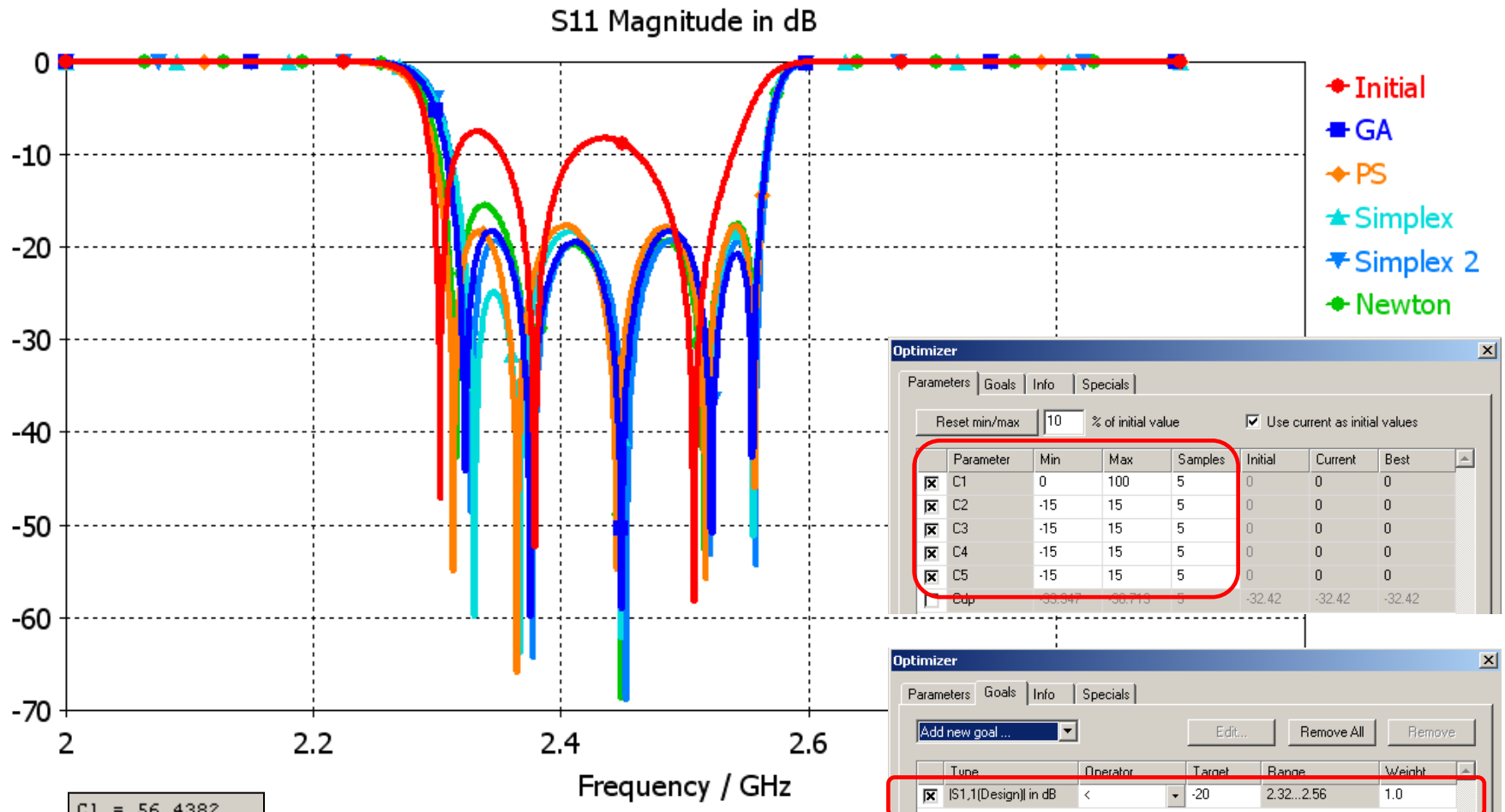
- Einführung
- Übersicht der verwendeten Optimierungsalgorithmen
 - Genetic Algorithm
 - Particle Swarm Optimization
 - Nelder-Mead Simplex
 - Interpolated Quasi-Newton
 - Trust Region Framework
- Sensitivität /Yield
- Beispiele
 - Cavity und Iris gekoppelte Bandpass Filter
 - Zweistufiger Impedanztrafo
 - Planar filter tuning
- Zusammenfassung

Beispiel 4: Planar filter tuning



Result for the initial value of tuning capacitors $C_{1-5} = 0$ fF.

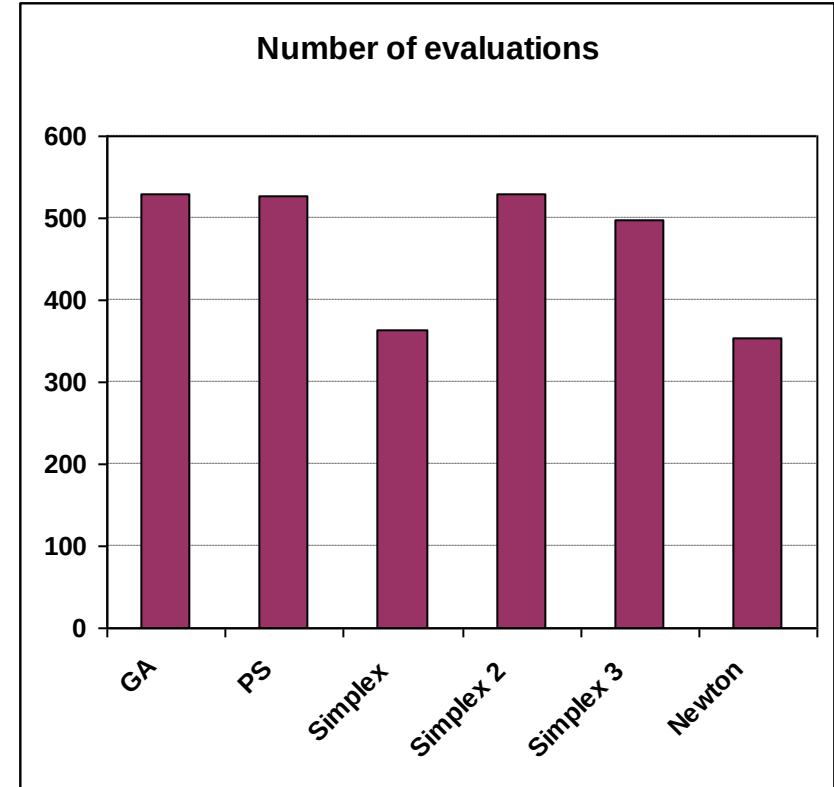
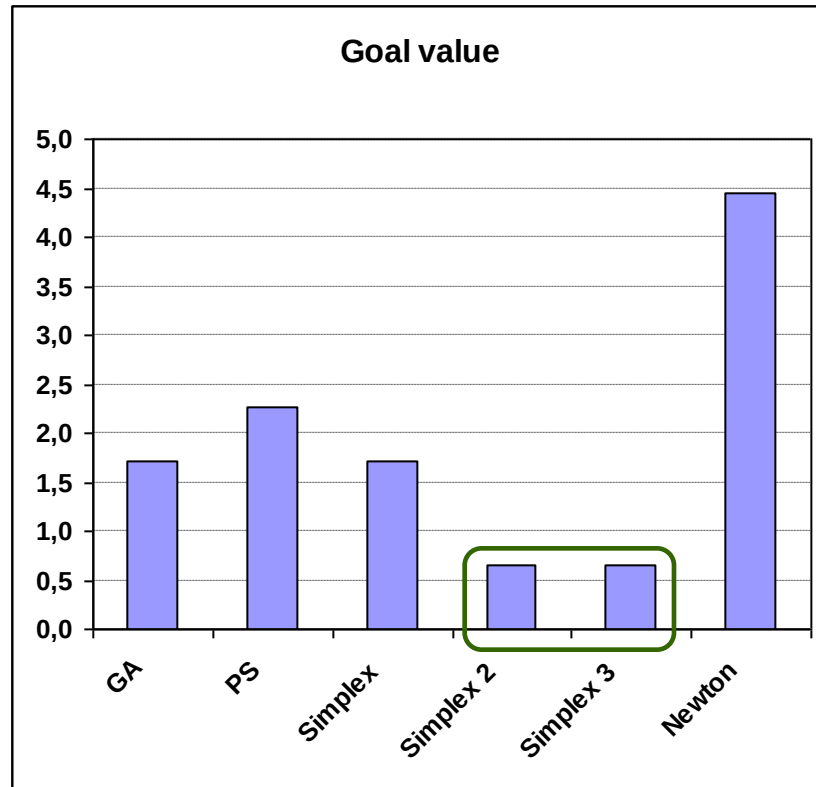
Bsp. 4: Optimizer setting and Results - S11 [dB]



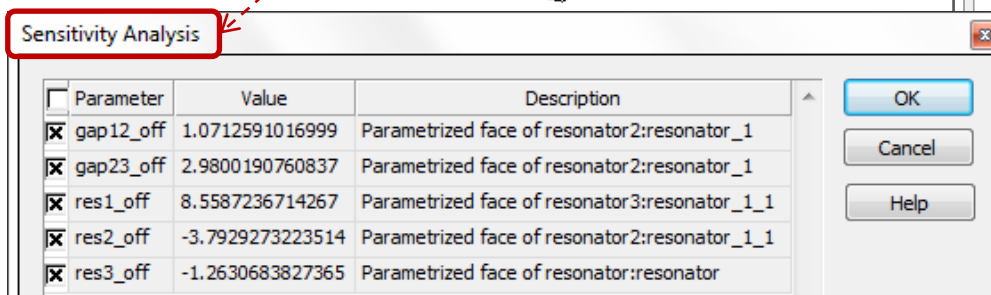
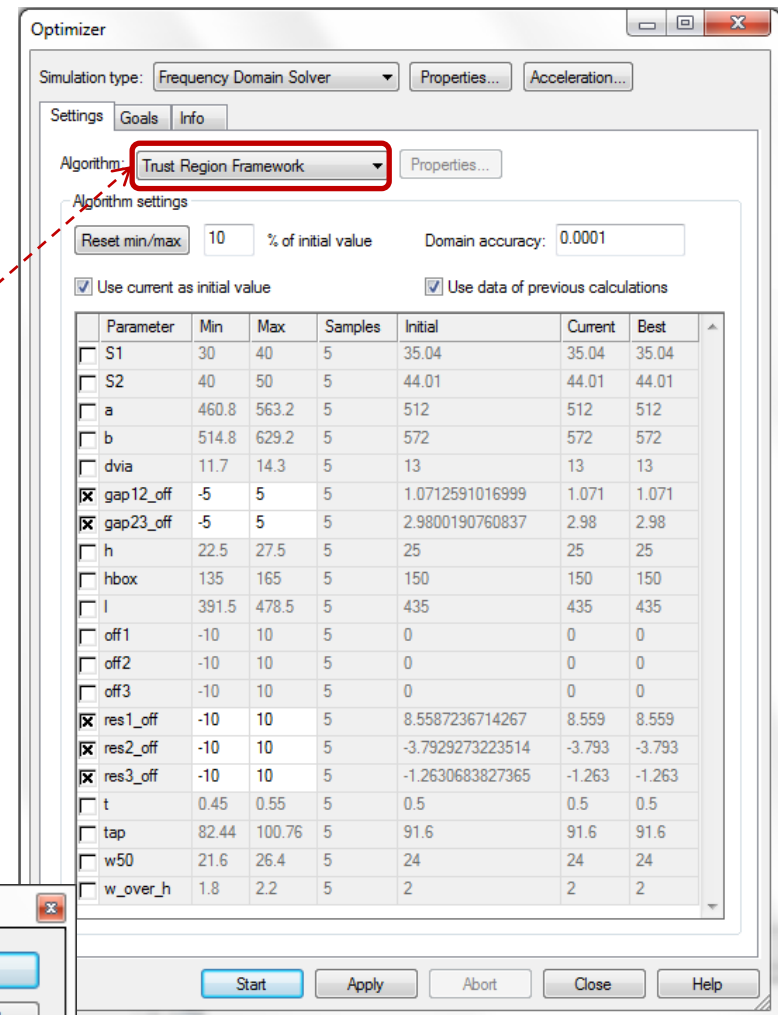
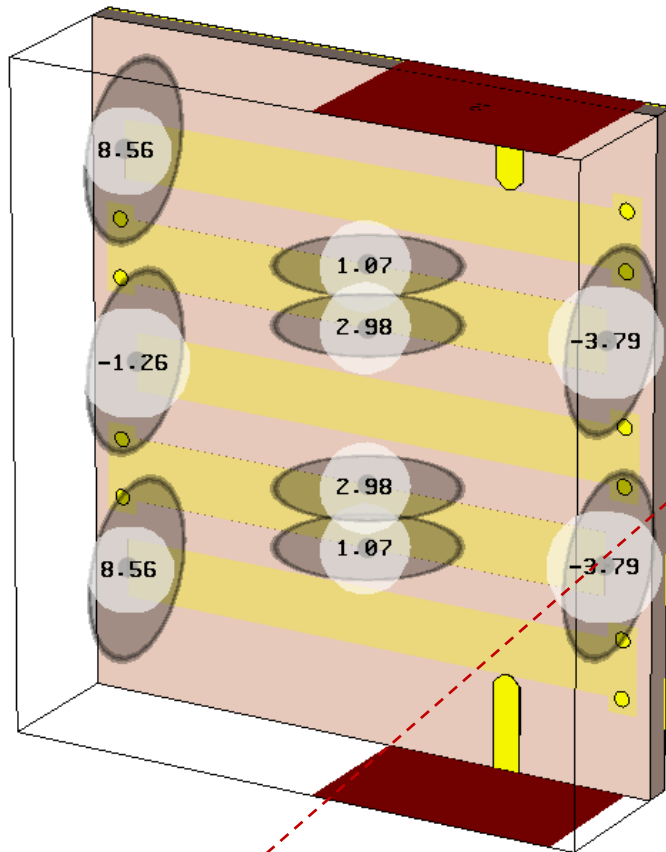
C1 = 56.4382
C2 = -8.21395
C3 = -14.4768
C4 = -9.20739
C5 = -0.940925

→ The values of the best result (Simplex method)

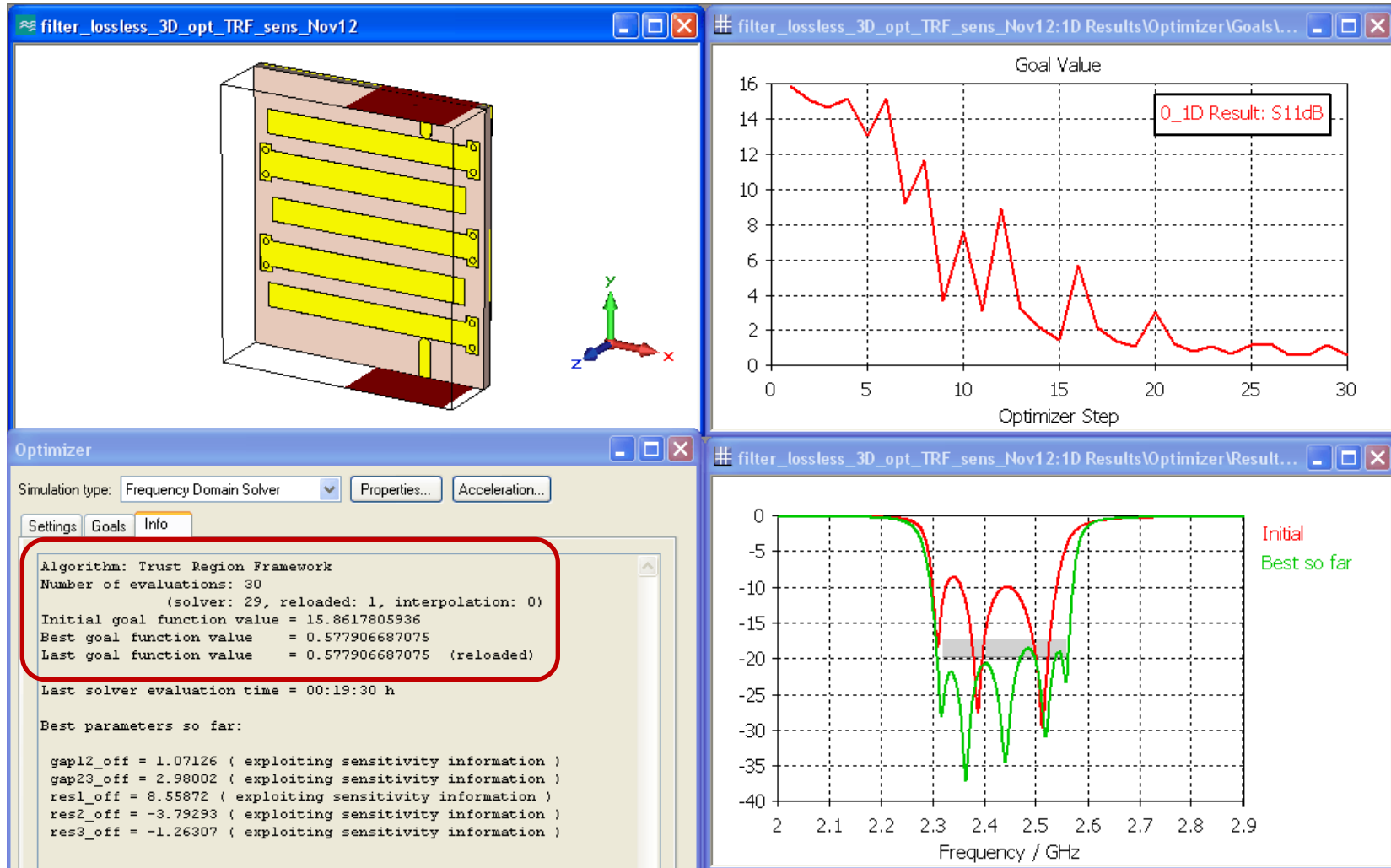
Bsp 4: Comparison of methods I.



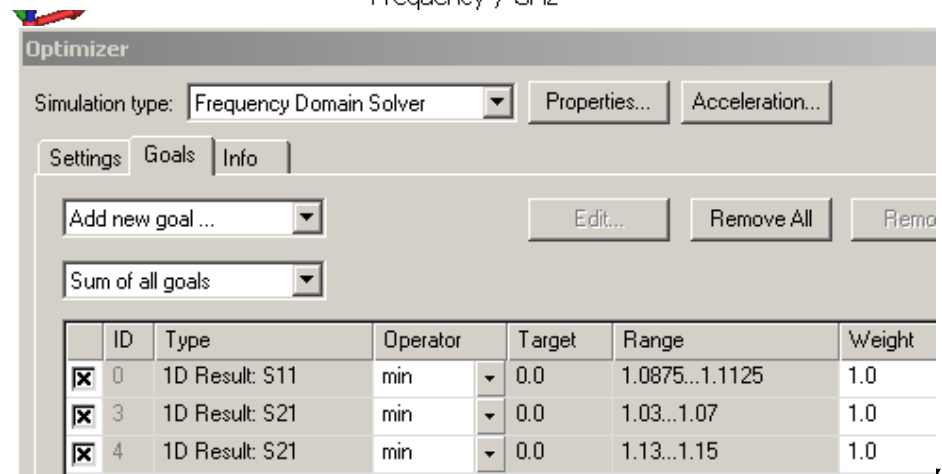
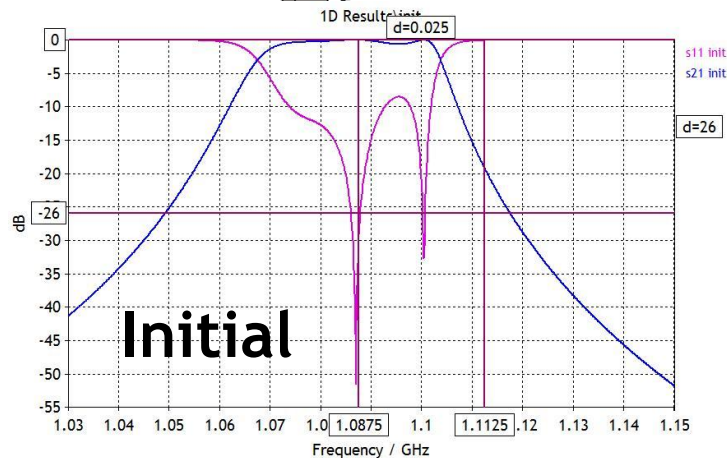
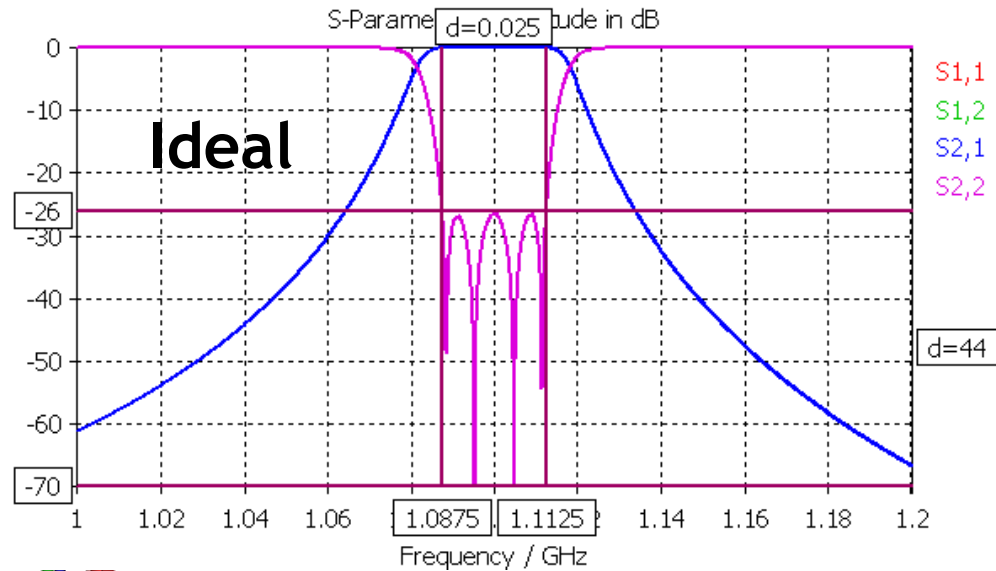
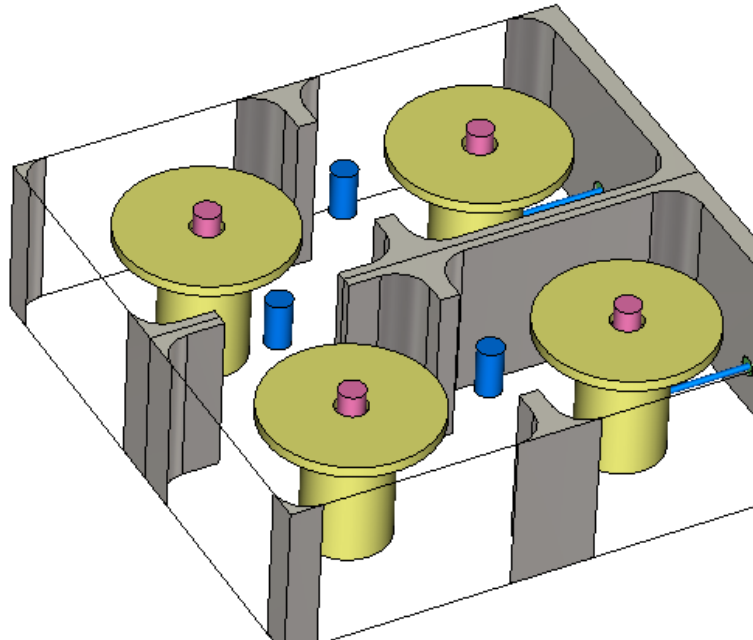
Bsp.4: Trust Region Framework + Sensitivity



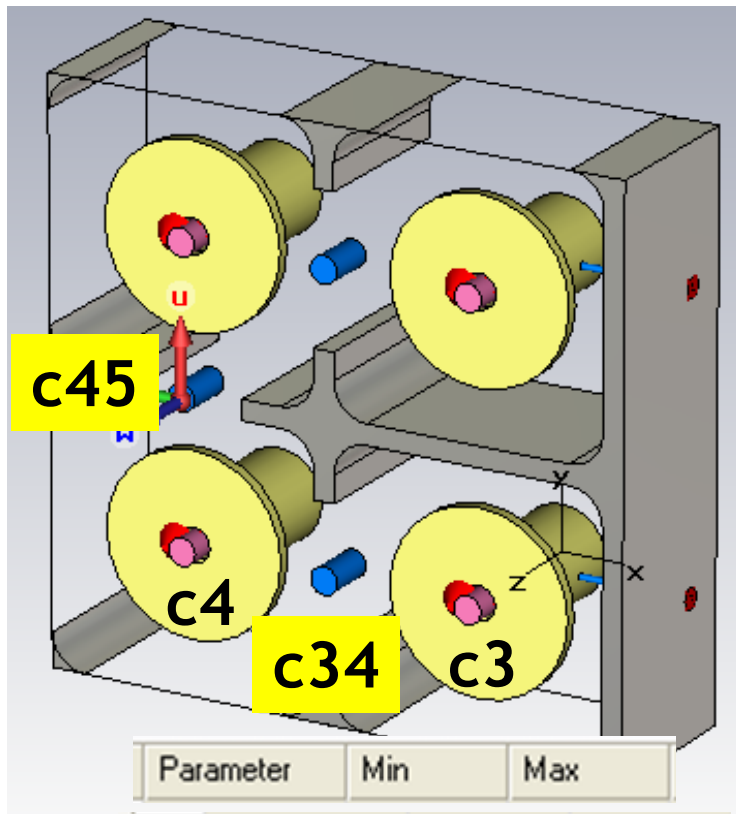
Bsp.4: TRF + Sensitivity: Results



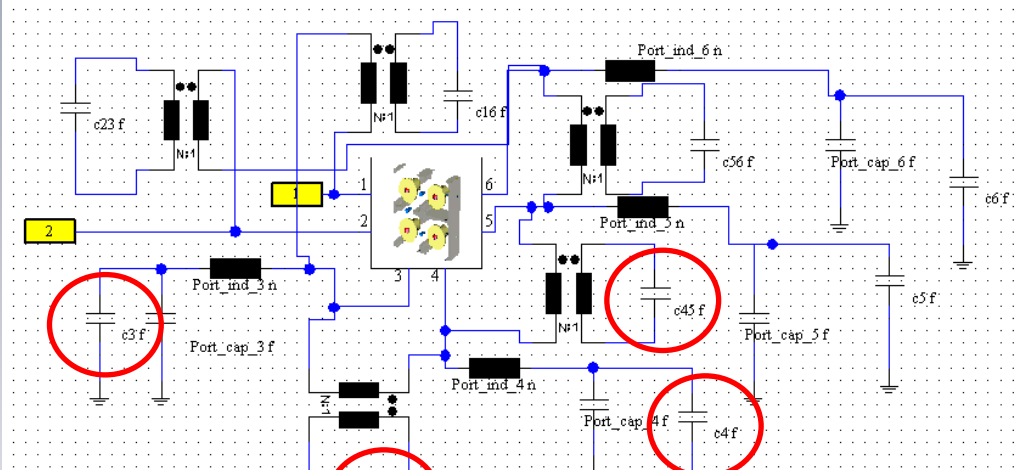
Bsp.5: 4-Pole Bandpass Filter (2.3% fractional Bandwidth)



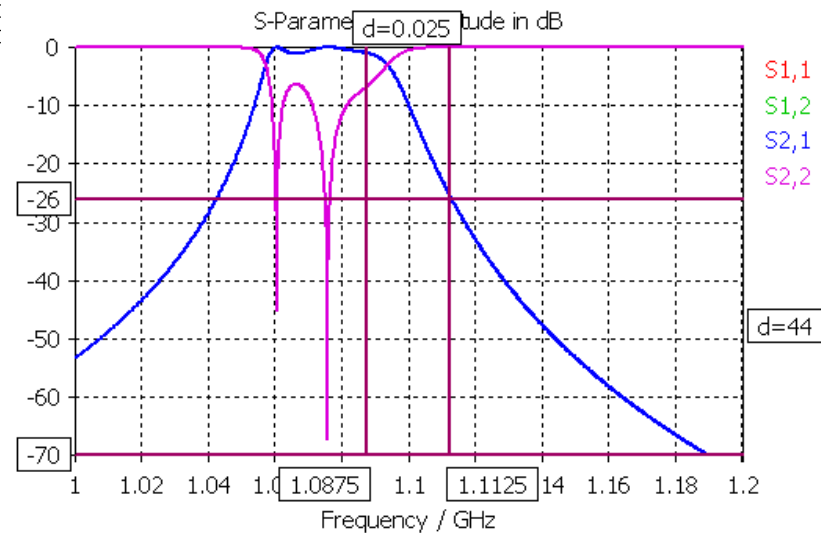
Bsp.5: Parameters:



Parameter	Min	Max
☒ c3	-300	300
☒ c34	-30	30
☒ c4	-300	300
☒ c45	-30	30

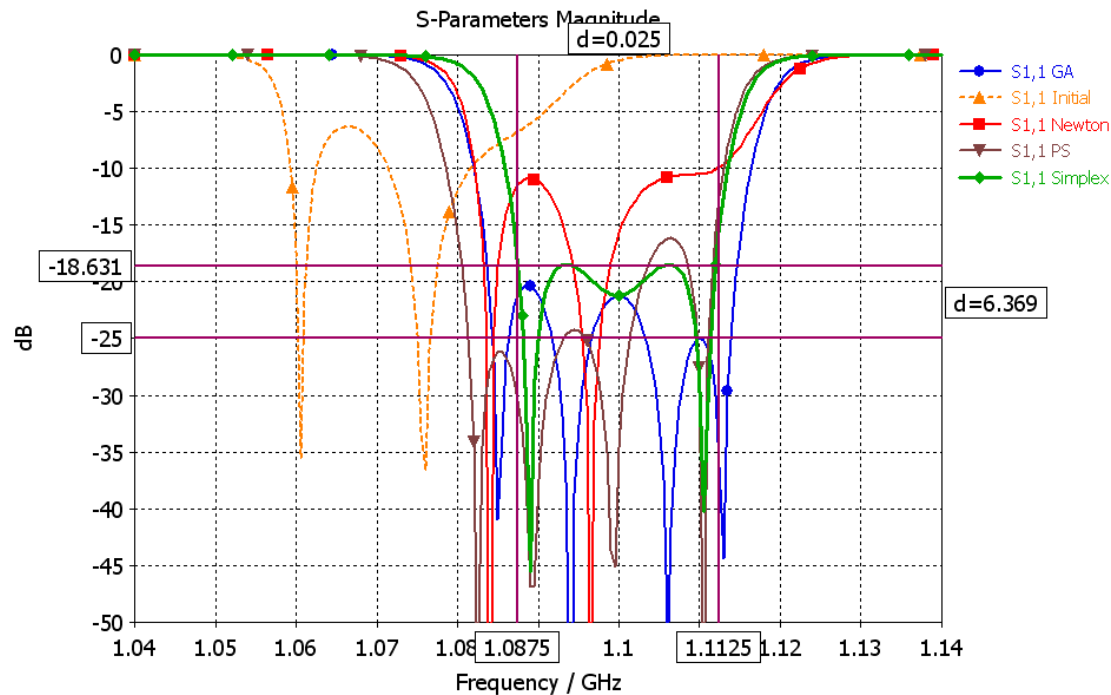


port1



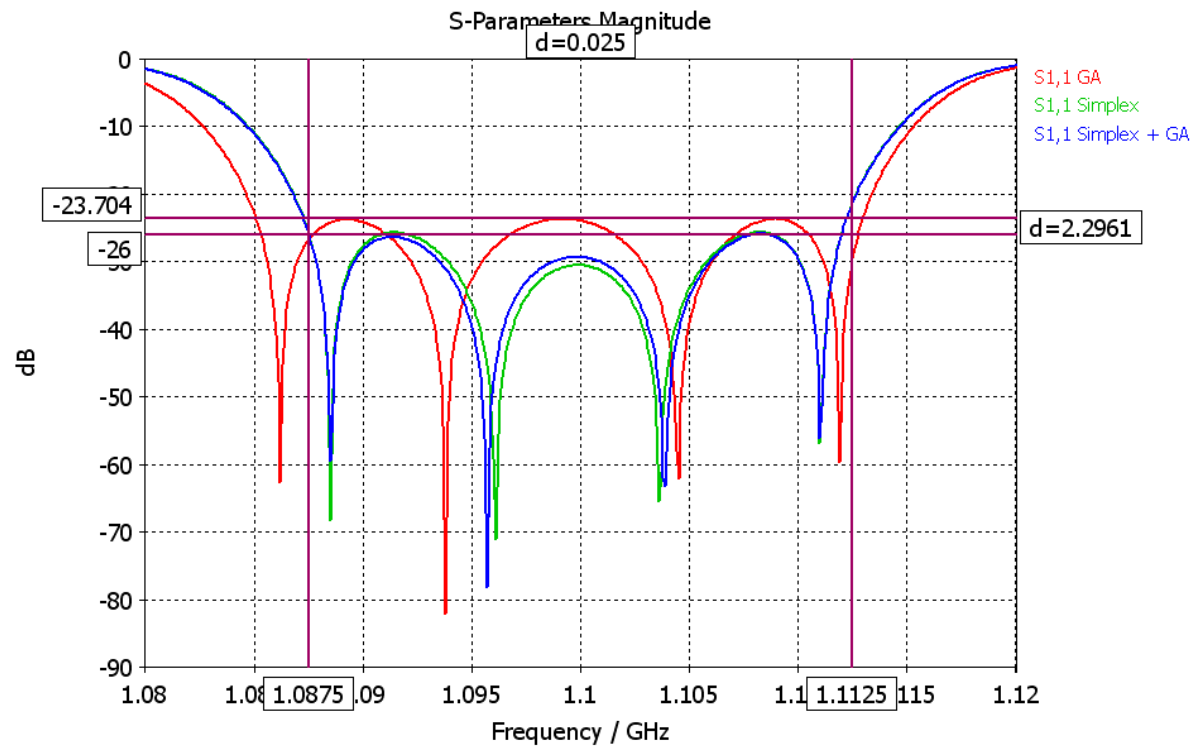
Bsp.5: Result Comparison

Optimization Algorithm	Goal value	Number of Iterations	C3	C34	C4	C45
Nelder - Mead Simplex	6.5304	284	-116.703	10.8331	-173.935	6.06555
Genetic Algorithm	4.6469	497	-101.677	-3.12807	-147.783	-3.7497
Particle Swarm	9.4008	451	-71.691	-2.31349	-125.868	-4.65223
Quasi-Newton	15.7412	66	-133.449	-5.62373	-147.671	-0.730258

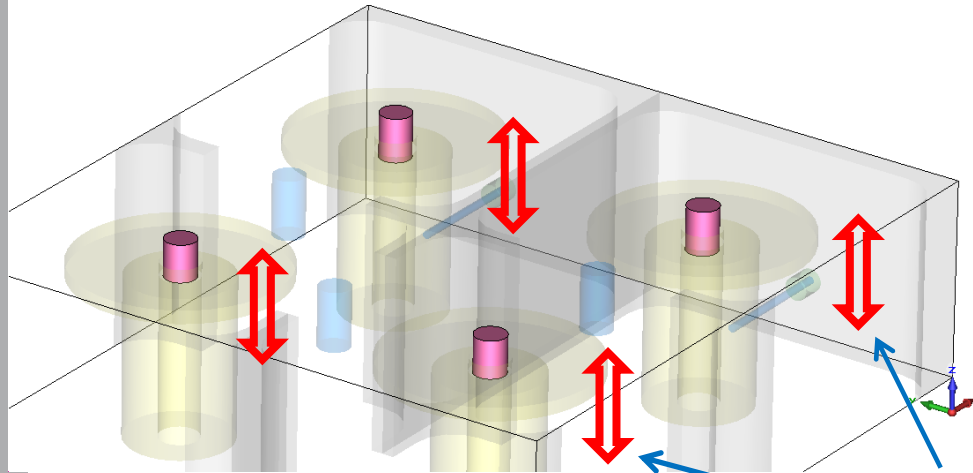


Bsp.5: 2nd Step - Fine Optimization

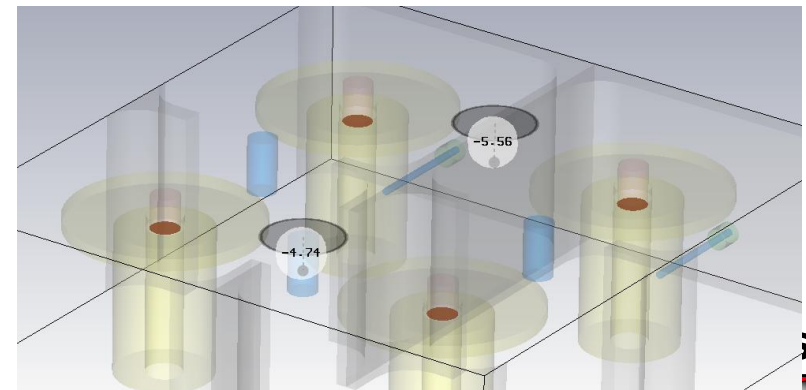
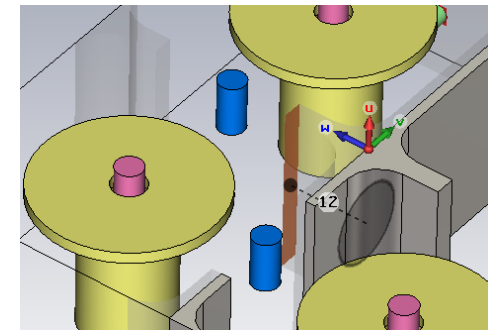
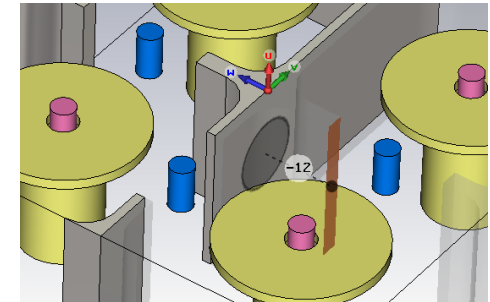
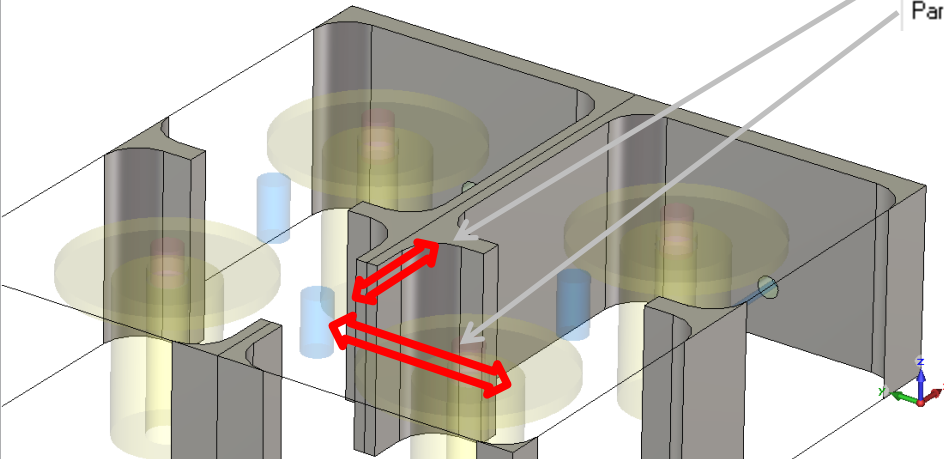
Optimization Algorithm	Goal value	Number of Iterations	C3	C34	C4	C45
Nelder - Mead Simplex	0.0517	352	-109.806	4.32482	-164.966	3.500
Genetic Algorithm	0.0653	1953	-100.589	-0.23677	-151.744	-0.920112
Simplex + GA	0.0508	1953	-109.279	4.09072	-164.547	3.43842



Bsp.5:Tuning Parameters/FaceConstraints

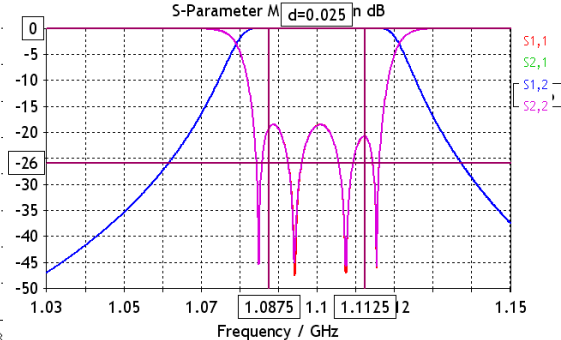
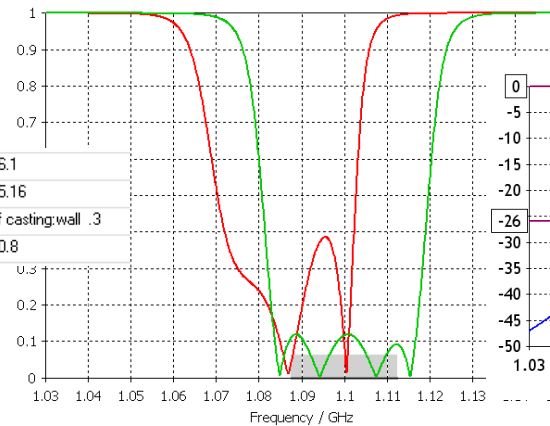
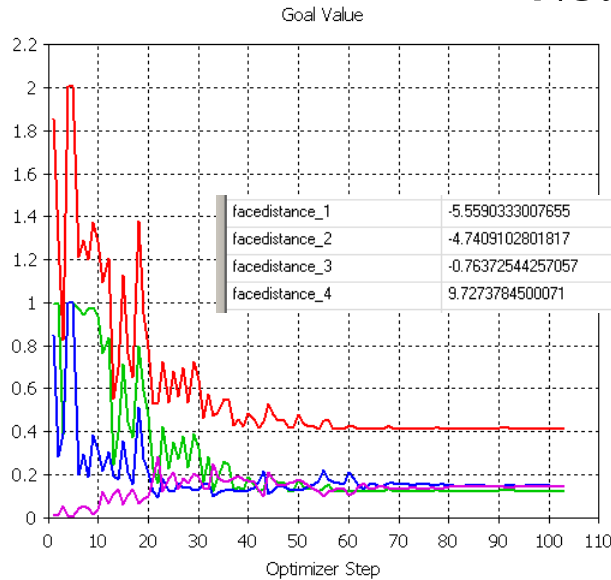


- Parametrized face -6.1
- Parametrized face -5.16
- Parametrized face of casting wall .3
- Parametrized face 10.8

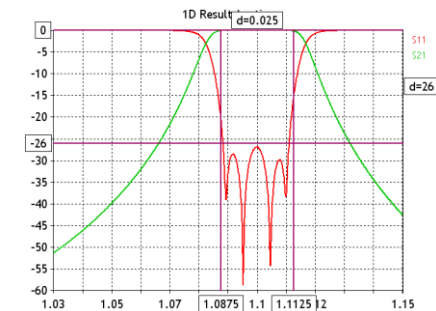
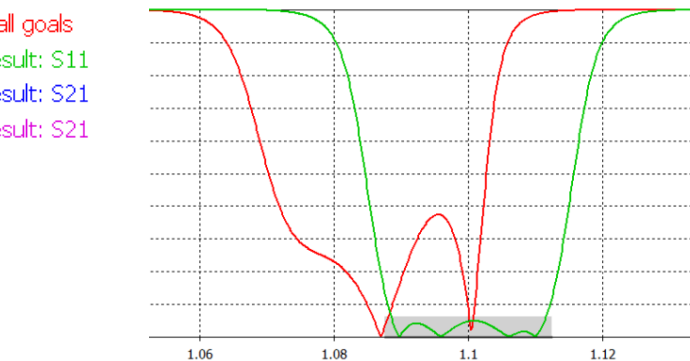
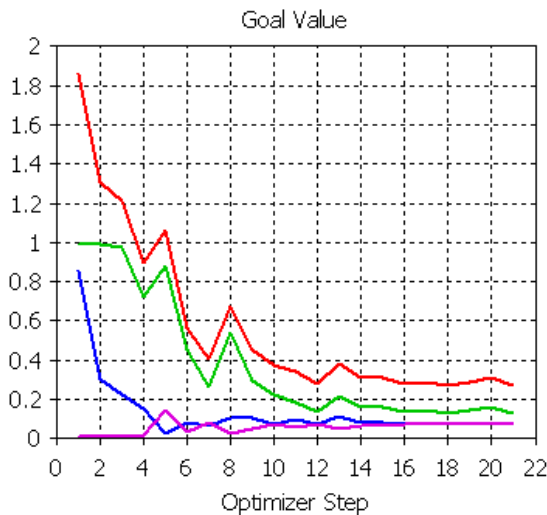


Bsp.5: Results

Nelder Mead: 70 Iterations

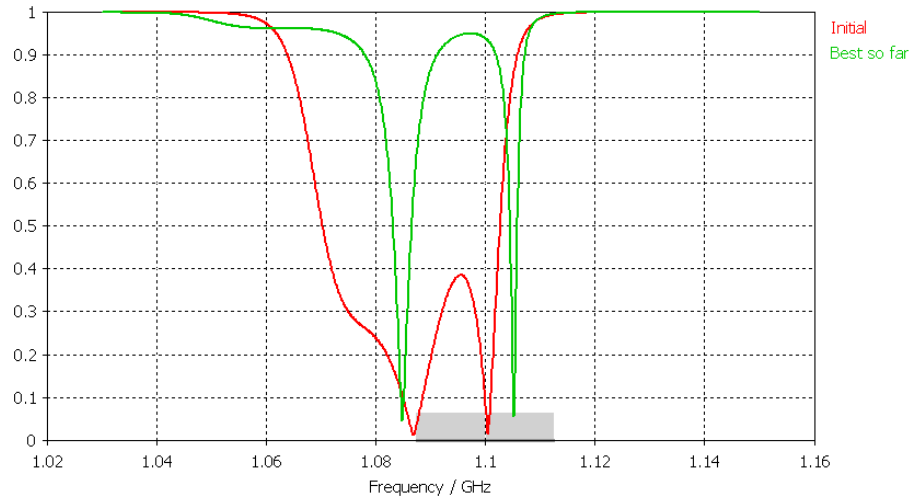


TRF: 15 Iterations



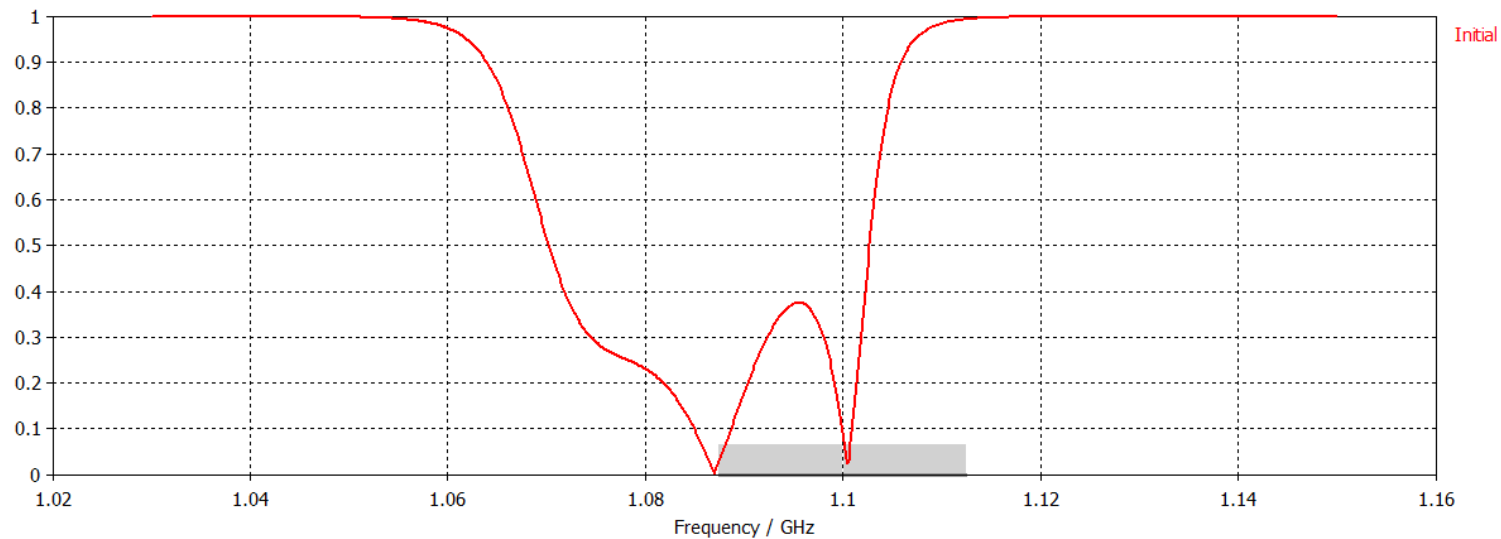
facedistance_1	-5.6140452203368	Parametrized face -6.1
facedistance_2	-4.8580310433415	Parametrized face -5.16
facedistance_3	0.89864122848554	Parametrized face of casting: wall .3
facedistance_4	11.323478857037	Parametrized face 10.8

Convergence Speed (animated)



NelderMead

TRF



Zusammenfassung



- **CST Studio Suite 2011 bietet**
 - Sensitivitäts- und Yield-Analysen
 - S-Parameter Sensitivität wird **exakt** analytisch berechnet (ohne zusätzlichen Aufwand), ohne Geometrie- oder Meshvariationen, aufwendige Parameterstudien entfallen
- **Sensitivitäts-Informationen beschleunigen Optimierungen**
 - Optimierung erfolgt an kleinen, linearen Matrix-Systemen und ist daher schnell
- **Neue Optimierungsstrategien basierend auf Sensitivitätsinformationen**
 - Trust Region Framework



Vielen Dank!