Optics Minimization in LISE^{++}

Purpose:
Minimization of optics of existed separators
First step: quad fields

First version v.9.10.100 from 05/22/15
Update v.9.10.119 from 06/23/15
Update v.9.10.156 from 07/27/15

Recently the first stage of optics minimization procedure was introduced, based on the "levmar" package by M.I.A. Lourakis using the Levenberg-Marquardt nonlinear least square algorithm. At this stage only the quadrupole fields can be varied to minimize user constraints for matrix and beam ellipse elements. In the future this minimization procedure will be used to define curved profile shape, fragment spatial distributions in Monte Carlo mode, and optimize intensity/purity combination.

Based on

levmar: Levenberg-Marquardt nonlinear least squares algorithms in C/C++. M.I.A. Lourakis July 2004. http://users.ics.forth.gr/~lourakis/levmar

Minimization for

- *E-blocks (extended configurations)*
- *with non-linked matrices*
- *set the option "Allow remote matrices recalculation"*

Introduction: "levmar" package

users.ics.forth.gr/~lourakis/levmar/

levmar : Levenberg-Marquardt nonlinear least squares algorithms in C/C++

Home About FAQ Changelog Download Contact

If you are looking for a general-purpose sparse Levenberg-Marquardt C/C++ implementation, please have a look at sparseLM.

Introduction

This site provides GPL native ANSI C implementations of the Levenberg-Marquardt optimization algorithm, usable also from C++, Matlab, Perl, Python, Haskell and Tcl and explains their use. Both unconstrained and constrained (under linear equations, inequality and box constraints) Levenberg-Marquardt variants are included. The Levenberg-Marquardt (LM) algorithm is an iterative technique that finds a local minimum of a function that is expressed as the sum of squares of nonlinear functions. It has become a standard technique for nonlinear least-squares problems and can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one, the algorithm behaves like a steepest descent method: slow, but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method.

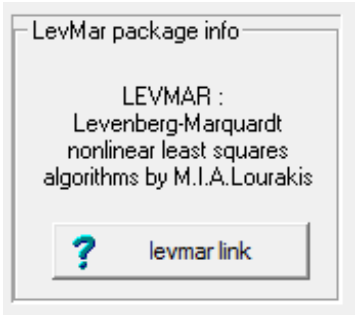
Technical Overview

`levmar` includes double and single precision LM C/C++ implementations, both with analytic and finite difference approximated Jacobians. It is provided free of charge, under the terms of the GNU General Public License. The mathematical theory behind unconstrained `levmar` is described in detail in the lecture notes entitled *Methods for Non-Linear Least Squares Problems*, by K. Madsen, H.B. Nielsen and O. Tingleff, Technical University of Denmark; Matlab implementations of the algorithms presented in the lecture notes are also available. Note however that the formulation of the minimization problem adopted here is slightly different from that described in the lecture notes. There is also a short note, providing a quick overview of the material in the lecture notes.

To deal with linear equation constraints, `levmar` employs variable elimination based on QR factorization, as described in ch. 15 of the book Numerical Optimization by Nocedal and Wright. For the box-constrained case, `levmar` implements the algorithm proposed by C. Kanzow, N. Yamashita and M. Fukushima, Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties, Journal of Computational and Applied Mathematics 172, 2004, pp. 375-397.

`levmar` provides the following two options regarding the solution of the linear systems formed by the augmented normal equations:

1. If you have LAPACK (or an equivalent vendor library such as Intel's MKL, AMD's AMCL, Sun's performance library, IBM's ESSL, SGI's SCSL, NAG, ...), the included LAPACK-based solvers can be used. This is the default option. The employed solver is based on the LU decomposition. Additionally, for experimenting with other approaches, linear solvers based on the Cholesky and QR decompositions have been supplied.
2. If LAPACK is unavailable, a LAPACK-free, LU-based linear systems solver can be used by undefining `HAVE_LAPACK` in `levmar.h`.



- Unconstrained optimization
  - `dlevmar_der()`: double precision, analytic Jacobian
  - `dlevmar_dif()`: double precision, finite difference approximated Jacobian
  - `slevmar_der()`: single precision, analytic Jacobian
  - `slevmar_dif()`: single precision, finite difference approximated Jacobian
- Constrained optimization
  - `dlevmar_lec_der()`: double precision, linear equation constraints, analytic Jacobian
  - `dlevmar_lec_dif()`: double precision, linear equation constraints, finite difference approximated Jacobian
  - `slevmar_lec_der()`: single precision, linear equation constraints, analytic Jacobian
  - `slevmar_lec_dif()`: single precision, linear equation constraints, finite difference approximated Jacobian
  - `dlevmar_bc_der()`: double precision, box constraints, analytic Jacobian
  - `dlevmar_bc_dif()`: double precision, box constraints, finite difference approximated Jacobian
  - `slevmar_bc_der()`: single precision, box constraints, analytic Jacobian
  - `slevmar_bc_dif()`: single precision, box constraints, finite difference approximated Jacobian
  - `dlevmar_blec_der()`: double precision, box & linear equation constraints, analytic Jacobian
  - `dlevmar_blec_dif()`: double precision, box & linear equation constraints, finite difference approximated Jacobian
  - `slevmar_blec_der()`: single precision, box & linear equation constraints, analytic Jacobian
  - `slevmar_blec_dif()`: single precision, box & linear equation constraints, finite difference approximated Jacobian
  - `dlevmar_bleic_der()`: double precision, box, linear equation & inequality constraints, analytic Jacobian
  - `dlevmar_bleic_dif()`: double precision, box, linear equation & inequality constraints, finite difference approximated Jacobian
  - `slevmar_bleic_der()`: single precision, box, linear equation & inequality constraints, analytic Jacobian
  - `slevmar_bleic_dif()`: single precision, box, linear equation & inequality constraints, finite difference approximated Jacobian
  - Convenience wrappers `xlevmar_blic_der()`/`xlevmar_blic_dif()`, `xlevmar_leic_der()`/`xlevmar_leic_dif()` & `xlevmar_lic_der()`/`xlev` are also provided.

Introduction: minimization scheme

LISE++

1. Select a optical block to minimize,
   Check in a parameter to minimize,
   Set bounds constraint

2. Create a block ”Fitting constraints”
   Set constraints

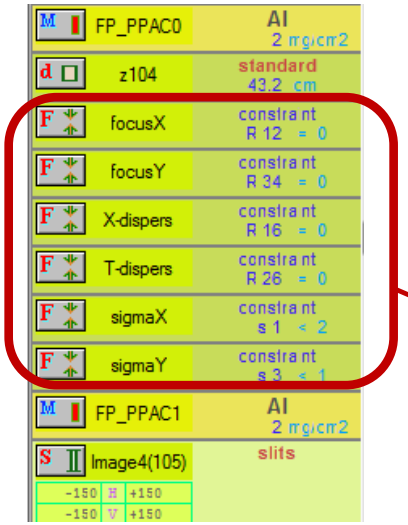
3. Run minimization

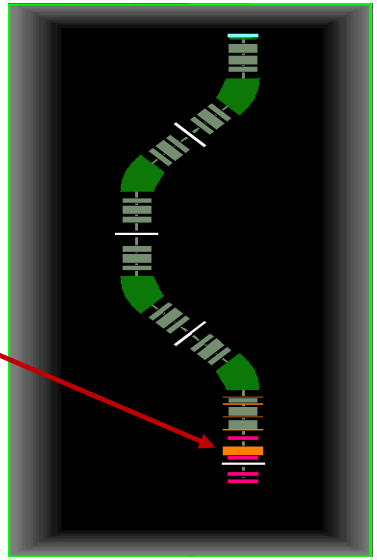
Transport

Command
5.01 "q1B " 0.7 -1.86164 9.75 ;

Command
10.0 "fit1" -2. 6. 0.0 .001 ;

3. Run minimization

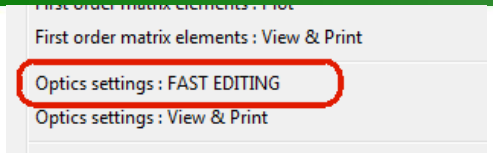




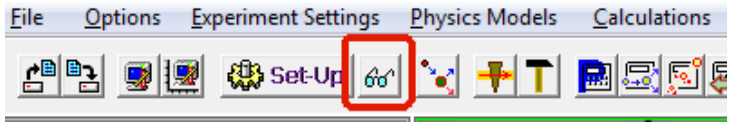
files\examples\NSCL\
A1900_extended_LISE_FIT.lpp

The next file is to append standard constraint blocks
files\examples\
FITconstraints.lpp

Introduction: information about blocks used for minimization



or



Optics settings (fast editing)

Block	Given Name	Start(m)	Length(m)	B0(kG)	Br(Tm)cor/*real	DriftM/*Angle	Rapp(cm)/*R(...	Leff(m)/*Ldip(m)	2 nd order	CalcMatr/*Z-Q	AngAcc,Apps,Slits	COSY | Fit	SE
drift	z089	28.783	0.5640			standard					-- HV --	-	e
= Dipole	D4	29.347	2.4300	+9.6965	* 3.0000	* +45.0	* 3.0939	* 2.4299	yes	* 0	-- HV --	-	S
drift	z097	31.777	0.5260			standard					-- HV --	-	e
Fit	sigY	32.303	0.0000								-- -- --	s3 < 50	e
<Quad>	Q098-8TA	32.303	0.4300	+7.0851	3.0000	QUAD	15.0000	0.4300	yes	1 R	-- HV --	FIT	e
Fit	sigY	32.733	0.0000								-- -- --	s3 < 50	e
drift	z099	32.733	0.1720			standard					-- HV --	-	e
<Quad>	Q100-8TB	32.905	0.7480	-8.1167	3.0000	QUAD	13.3000	0.7480	yes	1 R	-- HV --	FIT	e
Fit	sigY	33.653	0.0000								-- -- --	s3 < 50	e
drift	z101	33.653	0.1756			standard					-- HV --	-	e
<Quad>	Q102-8TC	33.828	0.7480	+4.2117	3.0000	QUAD	13.3000	0.7480	yes	1 R	-- HV --	FIT	e
Fit	sigY	34.576	0.0000								-- -- --	s3 < 50	e
drift	z103	34.576	0.3750			standard					-- HV --	-	e
drift	z104	34.951	0.4320			standard					-- HV --	-	e
Fit	focusX	35.383	0.0000								-- -- --	R12 = 0	e
Fit	focusY	35.383	0.0000								-- -- --	R34 = 0	e
Fit	X-dispers	35.383	0.0000								-- -- --	R16 = 0	e
Fit	T-dispers	35.383	0.0000								-- -- --	R26 = 0	e
Fit	sigmaX	35.383	0.0000								-- -- --	s1 < 2	e
Fit	sigmaY	35.383	0.0000								-- -- --	s3 < 1	e
slits	Image4(105)	35.383	0.0000			SLITS					-- -- HV	-	e
drift	z105	35.383	0.3490			standard					-- -- --	-	e
drift	z106	35.732	0.0890			standard					-- -- --	-	e

Selected block: Dispersive (M-dipole); Block Length [m] 0.0001; Let call automatically; Block name = tuning; Length after this block [m] 0.0001; Charge State (Z-Q) = 0; Selected Block Edit; Mutlipole Edit; Cuts (Acceptances); Optical Matrix

Angular acceptance (mrad): Use; Horizontal ±; Vertical ±; Shape: Rectangle / Ellipse

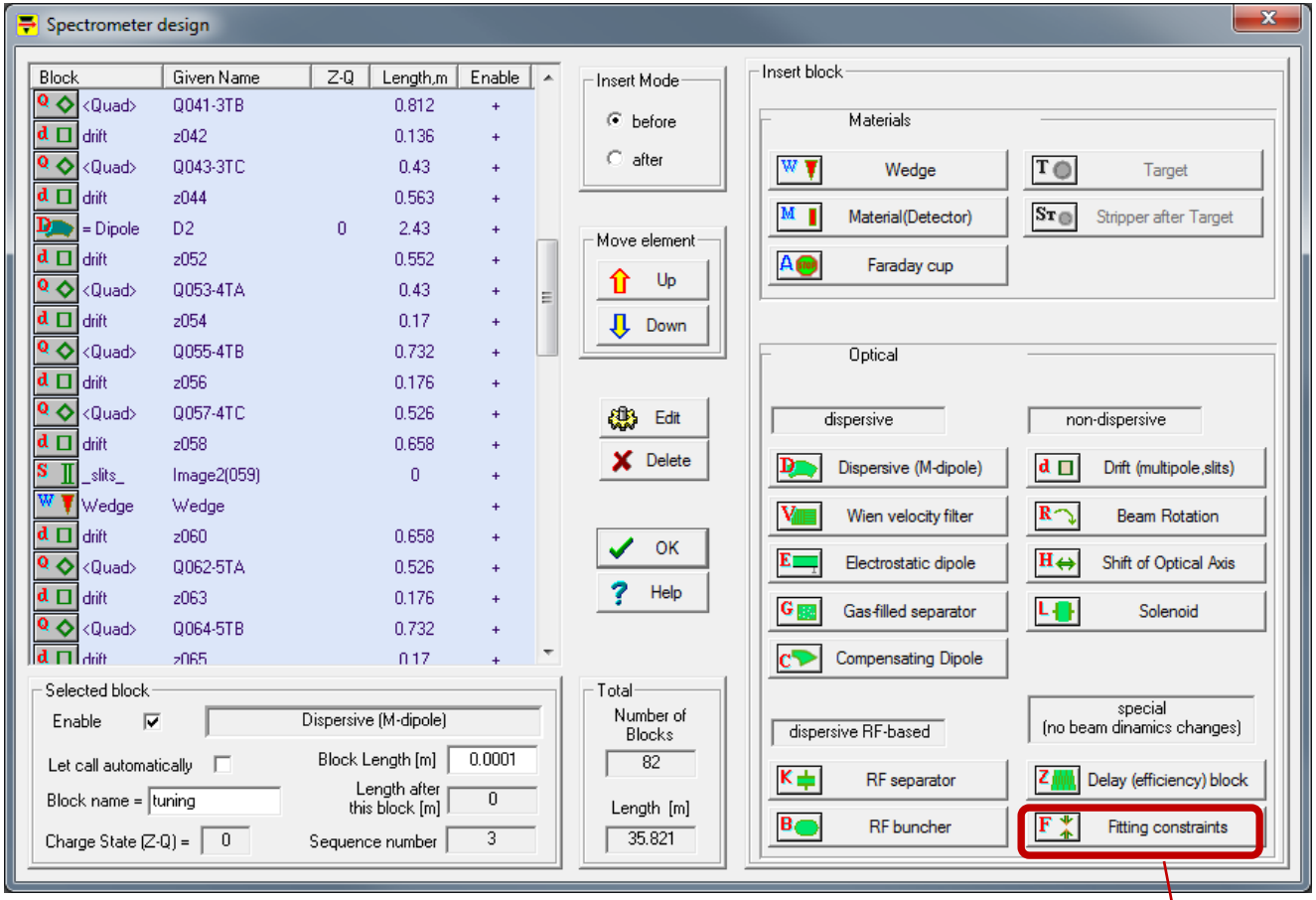
Inside Aperture (mm): min max Use; X = -50 50; Y = -50 50; Shape: Rectangle / Ellipse

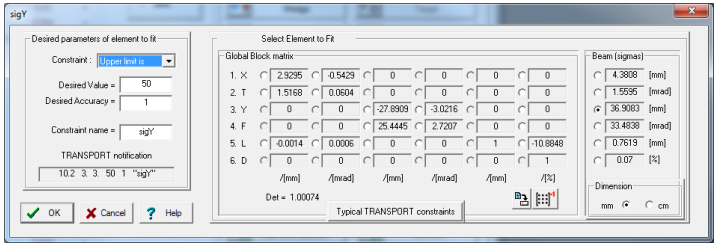
Slits (mm) after this BLOCK: min max Use; X =; Y =; Shape: Rectangle / Ellipse

1-st order Matrix Elements: Matrix Plot; Beam-Sigma Plot; View

Quit; Help

New Optic Block "Fitting Constraints"



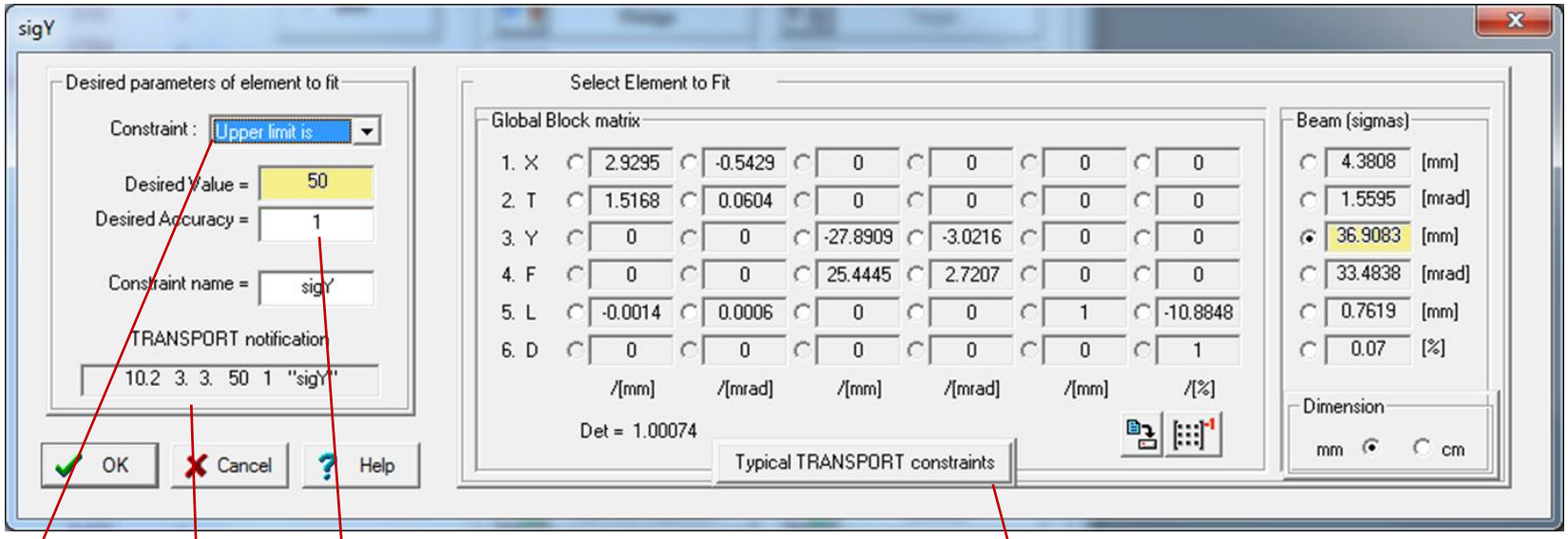


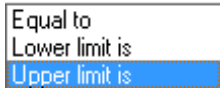
New Optic Block "Fitting Constraints"

42* possible selection for global matrix elements and beam sigma vector

(some matrix elements can be disabled if non rotation or solenoid blocks)

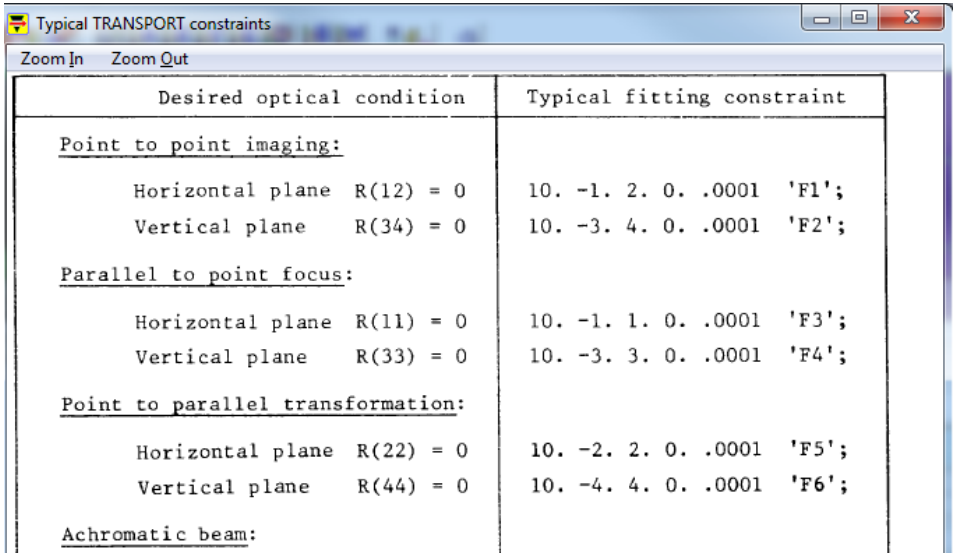
The "Fit constraint" dialog. For a constraint the user selects an element from an optical matrix or beam sigma vector, and set its desired value and precision (weight).



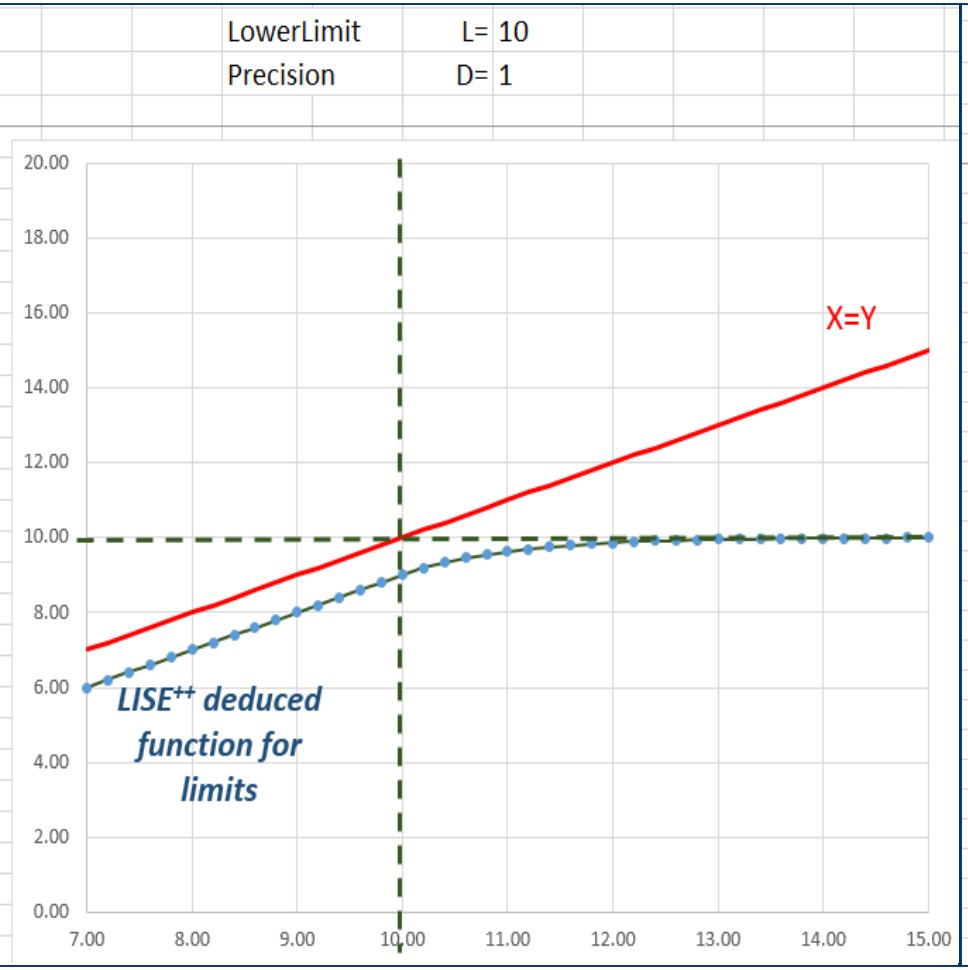


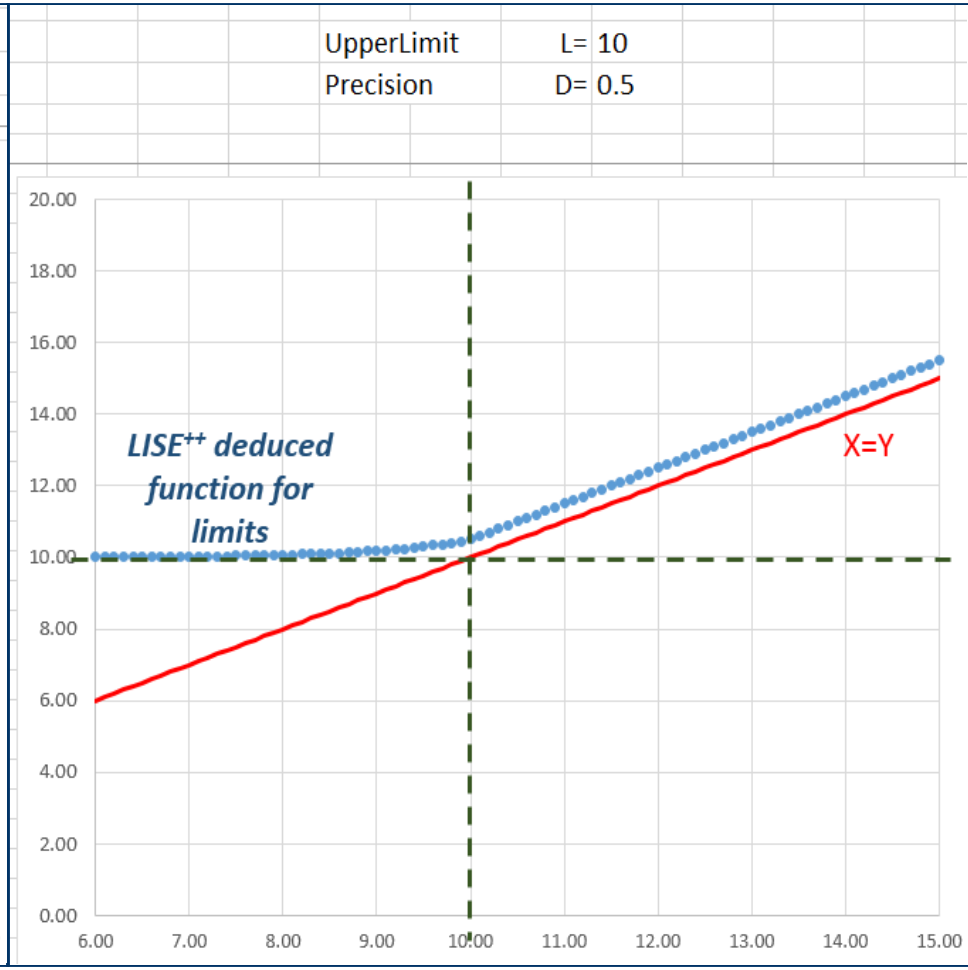
Inverse weight

TRANSPORT notification of selected constraint.
Second order constraint input under development



Special LISE^{++} functions fore limit constraints



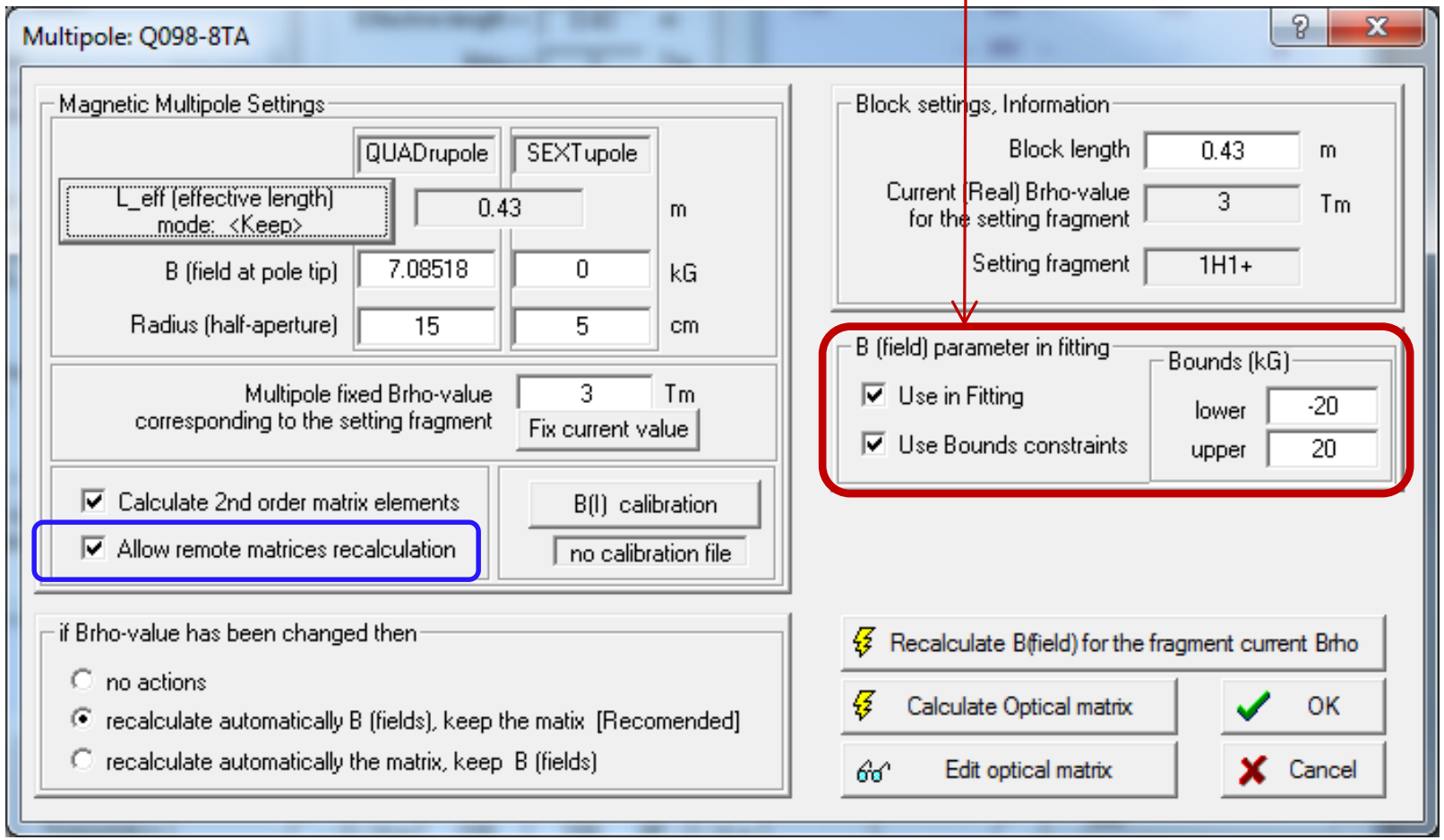


Levmar functions for "equal_to" constraints are used.
Important to have limit constraints in LISE^{++} for apertures
New Functions should continuous!

Selecting optic block and its parameters to use in the minimization

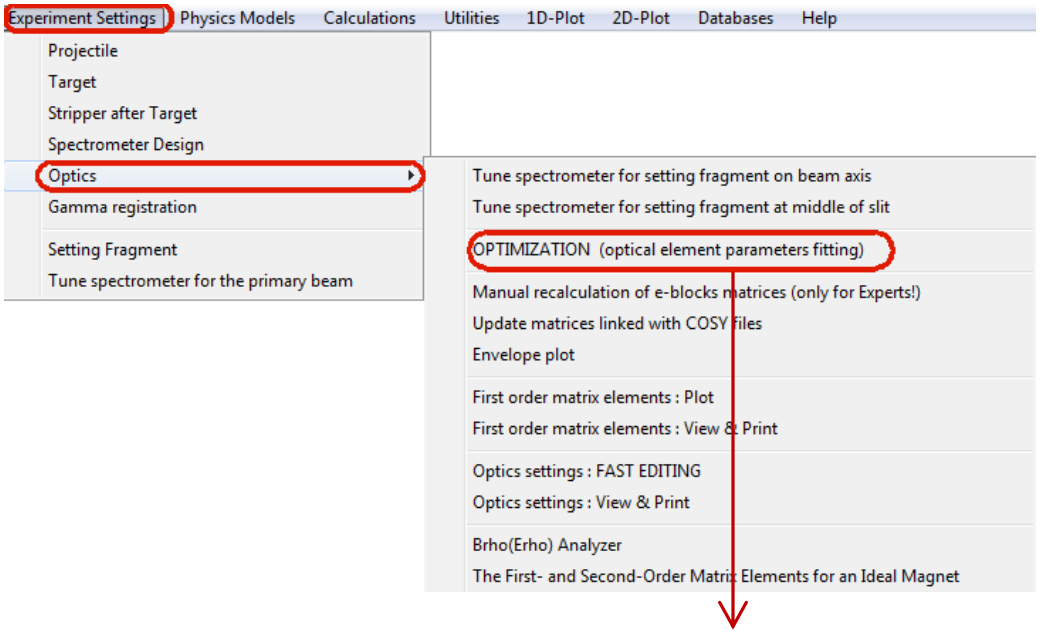
In current version only M-Quad B-fields and E-Quad voltages

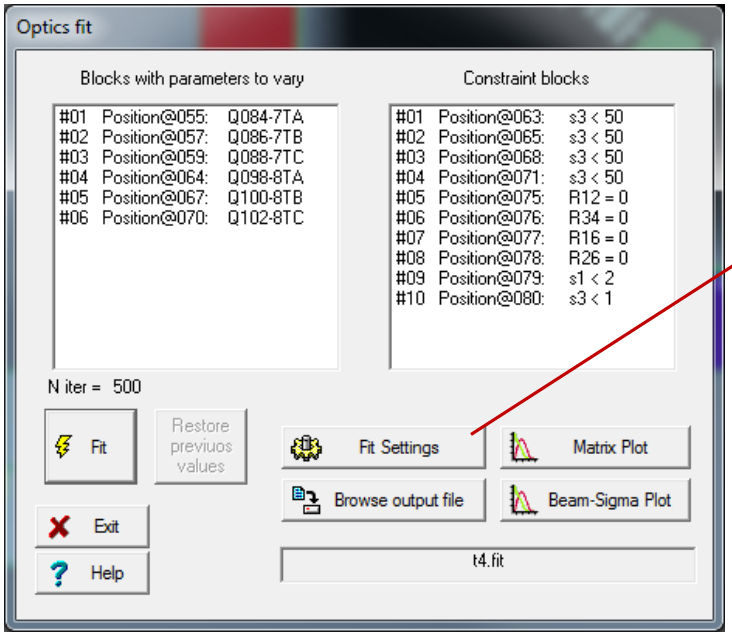
Set in it!

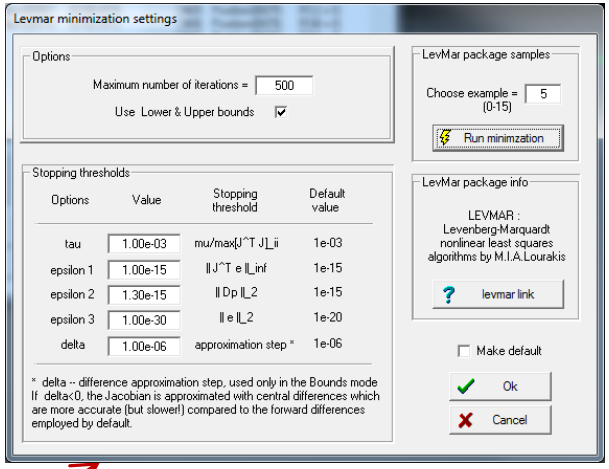


No matrix link to external file!

Optimization dialog





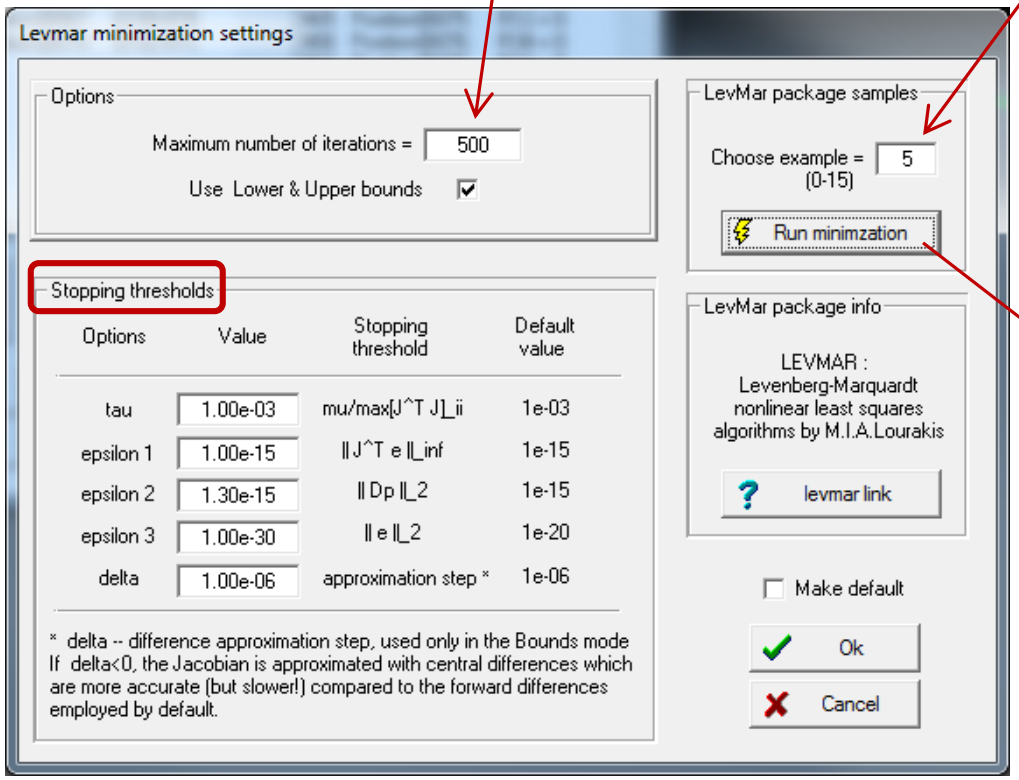


The "Optics Fit" dialog. The left panel shows optical blocks with varying parameters, whereas blocks with fitting constraints.

Fit Settings Dialog

For the first step use “50-100”

“levmar” package examples to play with settings



“see the next page

Levmar's examples (#5 – Osbrone data fitting)

```
case 5:
/* Osborne's data fitting problem */
{
  double x33[]={
    8.44E-1, 9.08E-1, 9.32E-1, 9.36E-1, 9.25E-1, 9.08E-1, 8.81E-1,
    8.5E-1, 8.18E-1, 7.84E-1, 7.51E-1, 7.18E-1, 6.85E-1, 6.58E-1,
    6.28E-1, 6.03E-1, 5.8E-1, 5.58E-1, 5.38E-1, 5.22E-1, 5.06E-1,
    4.9E-1, 4.78E-1, 4.67E-1, 4.57E-1, 4.48E-1, 4.38E-1, 4.31E-1,
    4.24E-1, 4.2E-1, 4.14E-1, 4.11E-1, 4.06E-1};

  m=5; n=33;
  p[0]=0.5; p[1]=1.5; p[2]=-1.0; p[3]=1.0E-2; p[4]=2.0E-2;


        work=malloc((LM_DIF_WORKSZ(m, n)+m*m)*sizeof(double));
        if(!work){
          fprintf(stderr, "memory allocation request failed in main()\n");
          exit(1);
          }
        covar=work+LM_DIF_WORKSZ(m, n);

  if(box)
    {
    lb[0]=lb[1]=lb[2]=lb[3]=lb[4]= -20;
    ub[0]=ub[1]=ub[2]=ub[3]=ub[4]=  20;

        if(jacob) ret=dlevmar_bc_der(osborne, jacosborne, p, x33, m, n, lb, ub, NULL,
        else      ret=dlevmar_bc_dif(osborne,             p, x33, m, n, lb, ub, NULL,
    }
  else
    {
        if(jacob) ret=dlevmar_der(osborne, jacosborne, p, x33, m, n, N, opts, info, w
        else      ret=dlevmar_dif(osborne,             p, x33, m, n, N, opts, info, w
    }
}
break;
```

```
//##############################################################################
//##############################################################################
/* Osborne's problem, minimum at (0.3754, 1.9358, -1.4647, 0.0129, 0.02
void osborne(double *p, double *x, int m, int n, void *data)
{
register int i;
double t;

      for(i=0; i<n; ++i){
    t=10*i;
    x[i]=p[0] + p[1]*exp(-p[3]*t) + p[2]*exp(-p[4]*t);
      }
}
//##############################################################################
void jacosborne(double *p, double *jac, int m, int n, void *data)
{
register int i, j;
double t, tmp1, tmp2;

  for(i=j=0; i<n; ++i){
    t=10*i;
        tmp1=exp(-p[3]*t);
    tmp2=exp(-p[4]*t);

        jac[j++]=1.0;
        jac[j++]=tmp1;
        jac[j++]=tmp2;
    jac[j++]=-p[1]*t*tmp1;
    jac[j++]=-p[2]*t*tmp2;
  }
}
//##############################################################################
```

Levmar's examples (#5) : fitting results

```
/* Osborne's problem, minimum at (0.3754, 1.9358, -1.4647, 0.0129, 0.0221) */
```

c:\program files (x86)\lise\results\LevMar 5 00.log

Print

```
Jacob: No,  Box: No

------------------------------------
==>  Results for Osborne's problem(5):
Levenberg-Marquardt returned 90 in 90 iter, reason 2
Solution: 0.3754101  1.935847  -1.464687  0.01286753  0.0221227

Options info:
0: 1.000e-03        mu
1: 1.000e-15        epsilon1  ||J^T e||_inf
2: 1.000e-15        epsilon2  ||Dp||_2
3: 1.000e-20        epsilon3  ||e||_2
4: 1.000e-06        delta       aprrox.step

Minimization info:
0: 8.790e-01        ||e||_2 at initial p
1: 5.465e-05        ||e||_2
2: 4.899e-10        ||J^T e||_inf
3: 5.068e-31        ||Dp||_2
4: 2.233e+01        mu/max[J^T J]_ii
5: 90               # iterations
6: 2                reason for terminating
7: 141              # function evaluations
8: 10               # Jacobian evaluations
9: 91               # linear systems solved, i.e. # attempts for reducing error

Termination reason: 2 - stopped by small Dp

Covariance of the fit :
+4.293184e-06 +4.168233e-04 -4.204934e-04 +8.751658e-07 -1.635539e-06
+4.168233e-04 +4.853246e-02 -4.884843e-02 +9.847883e-05 -1.962537e-04
```

c:\program files (x86)\lise\results\LevMar 5 01.log

Print

```
Jacob: No,  Box: Yes
Low Bounds:-20  -20  -20  -20  -20
Upp Bounds:+20  +20  +20  +20  +20

------------------------------------
==>  Results for Osborne's problem(5):
Levenberg-Marquardt returned 24 in 24 iter, reason 2
Solution: 0.3754098  1.935822  -1.464662  0.01286748  0.0221228

Options info:
0: 1.000e-03        mu
1: 1.000e-15        epsilon1  ||J^T e||_inf
2: 1.000e-15        epsilon2  ||Dp||_2
3: 1.000e-20        epsilon3  ||e||_2
4: 1.000e-06        delta       aprrox.step

Minimization info:
0: 8.790e-01        ||e||_2 at initial p
1: 5.465e-05        ||e||_2
2: 1.902e-07        ||J^T e||_inf
3: 3.009e-33        ||Dp||_2
4: 7.378e-11        mu/max[J^T J]_ii
5: 24               # iterations
6: 2                reason for terminating
7: 983              # function evaluations
8: 24               # Jacobian evaluations
9: 24               # linear systems solved, i.e. # attempts for reducing error

Termination reason: 2 - stopped by small Dp

Covariance of the fit :
```

Jacobian: No

Box: No

Box: Yes

c:\program files (x86)\lise\results\LevMar 5 10.log

Print

```
Jacob: Yes,  Box: No

------------------------------------
==>  Results for Osborne's problem(5):
Levenberg-Marquardt returned 31 in 31 iter, reason 2
Solution: 0.3754101  1.935847  -1.464687  0.01286753  0.0221227

Options info:
0: 1.000e-03        mu
1: 1.000e-15        epsilon1  ||J^T e||_inf
2: 1.000e-15        epsilon2  ||Dp||_2
3: 1.000e-20        epsilon3  ||e||_2
4: 1.000e-06        delta       aprrox.step

Minimization info:
0: 8.790e-01        ||e||_2 at initial p
1: 5.465e-05        ||e||_2
2: 4.522e-09        ||J^T e||_inf
3: 5.227e-30        ||Dp||_2
4: 4.923e+01        mu/max[J^T J]_ii
5: 31               # iterations
6: 2                reason for terminating
7: 40               # function evaluations
8: 31               # Jacobian evaluations
9: 40               # linear systems solved, i.e. # attempts for reducing error

Termination reason: 2 - stopped by small Dp

Covariance of the fit :
+4.294491e-06 +4.169215e-04 -4.205927e-04 +8.751991e-07 -1.636147e-06
+4.169215e-04 +4.853941e-02 -4.885547e-02 +9.847325e-05 -1.963069e-04
```

c:\program files (x86)\lise\results\LevMar 5 11.log

Print

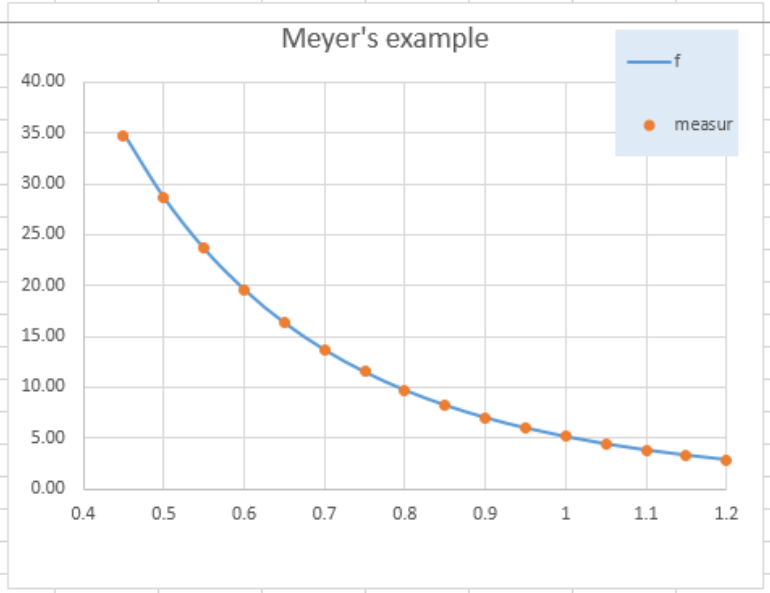
```
Jacob: Yes,  Box: Yes
Low Bounds:-20  -20  -20  -20  -20
Upp Bounds:+20  +20  +20  +20  +20

------------------------------------
==>  Results for Osborne's problem(5):
Levenberg-Marquardt returned 28 in 28 iter, reason 2
Solution: 0.3754101  1.935847  -1.464687  0.01286753  0.0221227

Options info:
0: 1.000e-03        mu
1: 1.000e-15        epsilon1  ||J^T e||_inf
2: 1.000e-15        epsilon2  ||Dp||_2
3: 1.000e-20        epsilon3  ||e||_2
4: 1.000e-06        delta       aprrox.step

Minimization info:
0: 8.790e-01        ||e||_2 at initial p
1: 5.465e-05        ||e||_2
2: 1.435e-09        ||J^T e||_inf
3: 0.000e+00        ||Dp||_2
4: 7.488e-11        mu/max[J^T J]_ii
5: 28               # iterations
6: 2                reason for terminating
7: 1359             # function evaluations
8: 28               # Jacobian evaluations
9: 28               # linear systems solved, i.e. # attempts for reducing error

Termination reason: 2 - stopped by small Dp

Covariance of the fit :
```

Jacobian: Yes

With Boxes is slower!!

Levmar's examples (#4) : Excel vs. Levmar

i	ui	f	measur	delta
0	0.45	34.92	34.78	5.49E-04
1	0.5	28.65	28.61	4.47E-05
2	0.55	23.63	23.65	2.15E-05
3	0.6	19.59	19.63	8.29E-05
4	0.65	16.32	16.37	1.35E-04
5	0.7	13.67	13.72	2.10E-04
6	0.75	11.49	11.54	1.77E-04
7	0.8	9.71	9.744	1.11E-04
8	0.85	8.24	8.261	5.74E-05
9	0.9	7.02	7.03	1.69E-05
10	0.95	6.00	6.005	4.68E-07
11	1	5.15	5.147	9.77E-06
12	1.05	4.44	4.427	4.58E-05
13	1.1	3.84	3.82	1.10E-04
14	1.15	3.33	3.307	1.95E-04
15	1.2	2.90	2.872	2.94E-04
				2.06E-03



f excel	f_levmar	delta	delta/f
34.92	34.78	0.14	0.39%
28.65	28.61	0.04	0.13%
23.63	23.64	-0.02	-0.07%
19.59	19.63	-0.04	-0.22%
16.32	16.38	-0.05	-0.32%
13.67	13.72	-0.05	-0.37%
11.49	11.54	-0.04	-0.38%
9.71	9.74	-0.03	-0.35%
8.24	8.26	-0.02	-0.28%
7.02	7.03	-0.01	-0.17%
6.00	6.01	0.00	-0.04%
5.15	5.15	0.01	0.13%
4.44	4.43	0.01	0.33%
3.84	3.82	0.02	0.55%
3.33	3.31	0.03	0.79%
2.90	2.87	0.03	1.06%

p0	9.460866	2.48	8.85
p1	5.096541	6.15	4
p2	3.112559	3.45	2.5

LevMar	5.72E-06	
		init
p0	2.48	8.85
p1	6.18	4
p2	3.50	2.5

Excel	2.06E-03	
		init
p0	9.46	8.85
p1	5.10	4
p2	3.11	2.5

x[i]=p[0]*exp(10.0*p[1]/(ui+p[2]) - 13.0)

Initial parameters for both cases

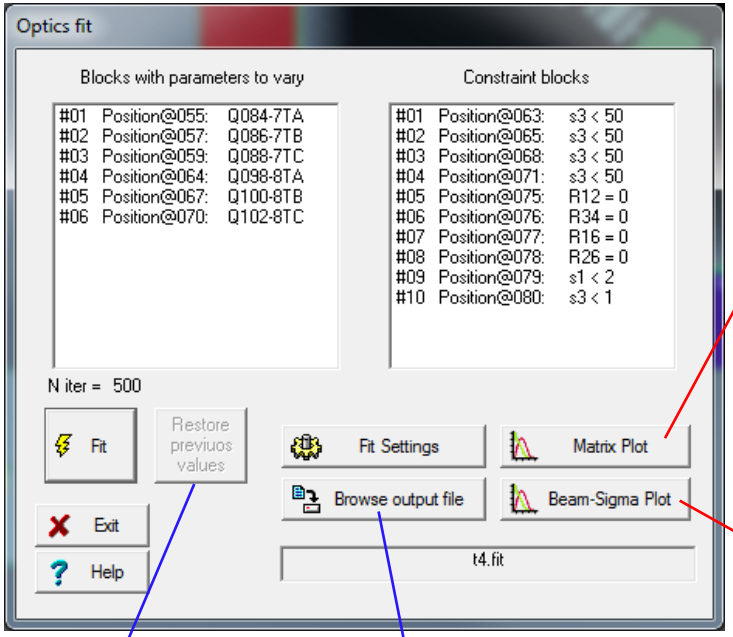
Levmar results

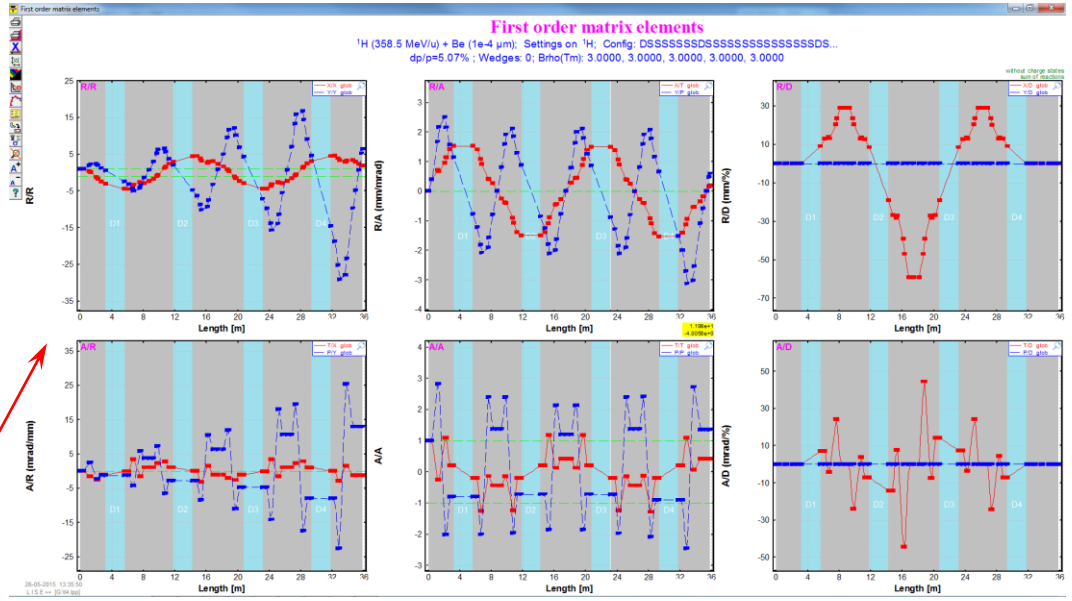
Excel results

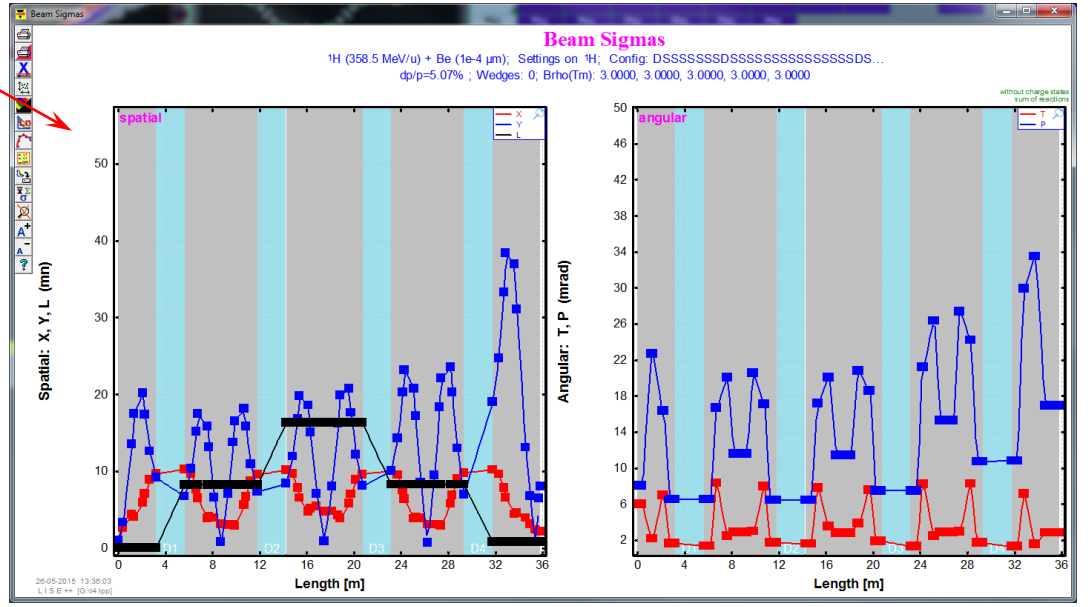
Levmar chi-square result by 3 orders of magnitude is lower, than Excel's result!!!

Matrix elements and Beam sigmas plot

You can get plots before fitting process and after to compare values







After fitting process it is possible to restore initial settings

Initial log-file name is LISE++ filename with the "fit" extension. Located by default in the directory "LISE\result"

Fitting information window (coming from log-file)

t4.fit

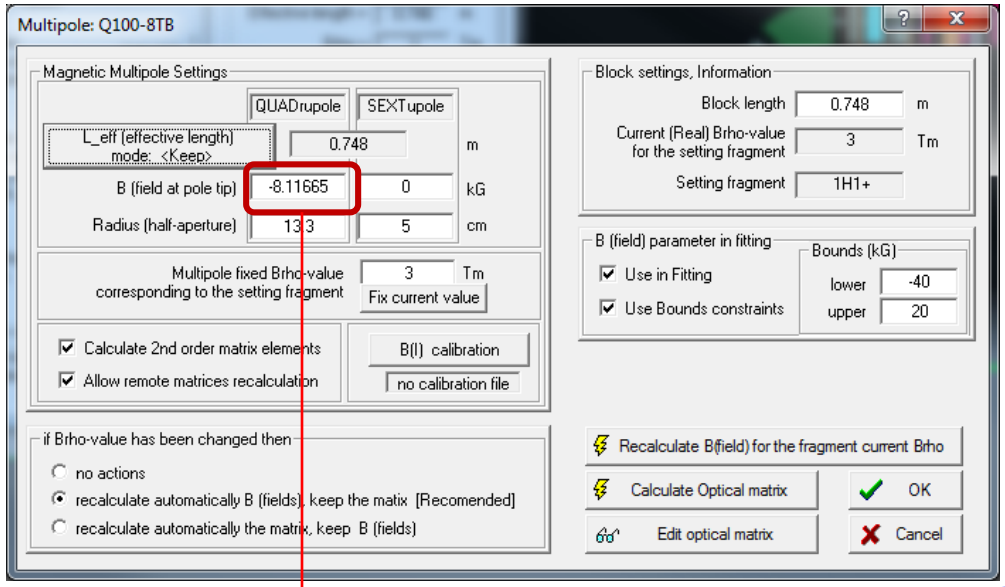
```
Initial  +641.499  and Final +641.173  LISE fit reduced values

Parameters:          LeftBound      Initial     RightBound   |   Final
#01: Q084-7TA      -1.000e+99 < +9.403e+00 < +1.000e+99   | +8.916e+00
#02: Q086-7TB      -1.000e+99 < -1.083e+01 < +1.000e+99   | -9.624e+00
#03: Q088-7TC      -1.000e+99 < +8.752e+00 < +1.000e+99   | +7.892e+00
#04: Q098-8TA      -2.000e+01 < +7.085e+00 < +2.000e+01   | +6.310e+00
#05: Q100-8TB      -4.000e+01 < -5.000e+00 < +2.000e+01   | -6.152e+00
#06: Q102-8TC      -1.000e+01 < +4.212e+00 < +4.000e+01   | +3.994e+00

-------------------------------------------------------------------------

Fitting values:       Initial       Final       Precision  (Fin-Des)/P |    Desired
#01: sigY           +2.469e+01   +1.648e+00    1.000e+00   0.000e+00   | < +5.00e+01
#02: sigY           +3.323e+01   +3.167e+00    1.000e+00   0.000e+00   | < +5.00e+01
#03: sigY           +4.555e+01   +6.177e+00    1.000e+00   0.000e+00   | < +5.00e+01
#04: sigY           +4.716e+01   +7.588e+00    1.000e+00   0.000e+00   | < +5.00e+01
#05: focusX         +6.022e-01   -1.913e-06    1.000e-03   1.913e-03   | = +0.00e+00
#06: focusY         -5.364e+00   -8.493e-01    5.000e-01   1.699e+00   | = +0.00e+00
#07: X-dispers      -1.287e-13   -1.767e-06    1.000e-03   1.767e-03   | = +0.00e+00
#08: T-dispers      +2.814e-13   -2.038e-05    1.000e-02   2.038e-03   | = +0.00e+00
#09: sigmaX         +3.618e+00   +1.991e+00    1.000e-01   9.911e-01   | < +2.00e+00
#10: sigmaY         +6.510e+01   +1.134e+01    1.000e-02   1.035e+03   | < +1.00e+00

-------------------------------------------------------------------------

==>  Results for t4.fit:
Levenberg-Marquardt returned 500 in 500 iter, reason 3
Solution: 8.916349  -9.623625  7.891567  6.309539  -6.152035  3.993751

Minimization info:
0: 4.147e+07        ||e||_2 at initial p
1: 4.111e+07        ||e||_2
2: 2.840e+00        ||J^T e||_inf
3: 2.275e-06        ||Dp||_2
4: 2.378e-04        mu/max[J^T J]_ii
5: 500              # iterations
6: 3                reason for terminating
7: 4499             # function evaluations
8: 500              # Jacobian evaluations
9: 500              # linear systems solved, i.e. # attempts for reducing error

Termination reason: 3 - stopped by itmax

-------------------------------------------------------------------------
```

Appears automatically after fitting process completed

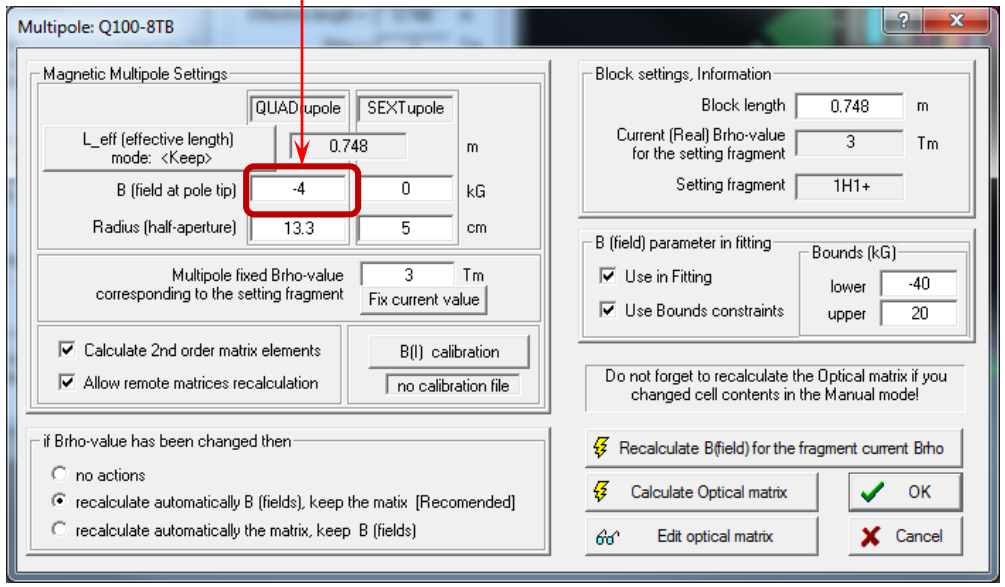
It is planning to use different colors and fonts to underline, to select key moments

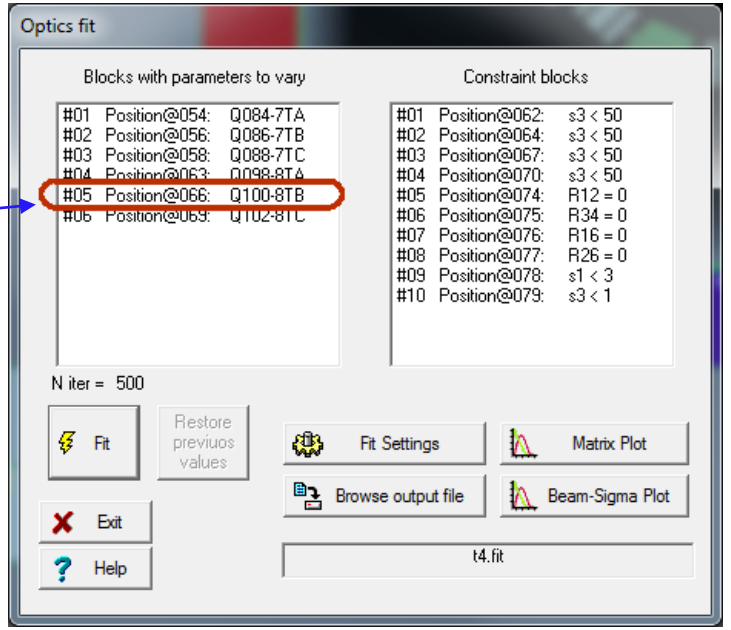
```
==> "sigmaY" : last fitting block global optical matrix and sigma vector

                                                    Format [mm-mrad]
=================== G L O B A L ====  matrix ==================   | Beam(sigma) |
   +1.991e+00  -1.913e-06    0           0          0    -1.767e-06   |    1.99e+00 |
   -1.482e+00  +5.025e-01    0           0          0    -2.038e-05   |    3.36e+00 |
    0           0          -9.083e+00  -8.493e-01   0     0           |    1.13e+01 |
    0           0          -3.325e+00  -4.210e-01   0     0           |    4.73e+00 |
   -1.428e-03  +5.792e-04    0           0          1.0  -1.088e+01   |    7.62e-01 |
    0           0           0           0          0    +1.000e+00   |    7.00e-02 |

-------------------------------------------------------------------------

Covariance of the fit :
+3.071318e+15 -7.383656e+15 +5.433857e+15 +6.802513e+16 -1.067550e+17 +2.799623e+15
-7.383595e+15 +1.775066e+16 -1.306325e+16 -1.635389e+17 +2.566550e+17 -6.742622e+15
+5.433800e+15 -1.306322e+16 +9.613624e+15 +1.203536e+17 -1.888818e+17 +4.964434e+15
+7.835533e+16 -1.883748e+17 +1.386313e+17 +1.182920e+18 -9.006090e+17 -1.998218e+18
-1.408364e+17 +3.385910e+17 -2.491815e+17 -1.296397e+18 -1.118888e+18 +6.699783e+18
+4.149371e+16 -9.976589e+16 +7.342309e+16 -1.150617e+18 +5.385893e+18 -7.714485e+18
```

Example for A1900 (1)

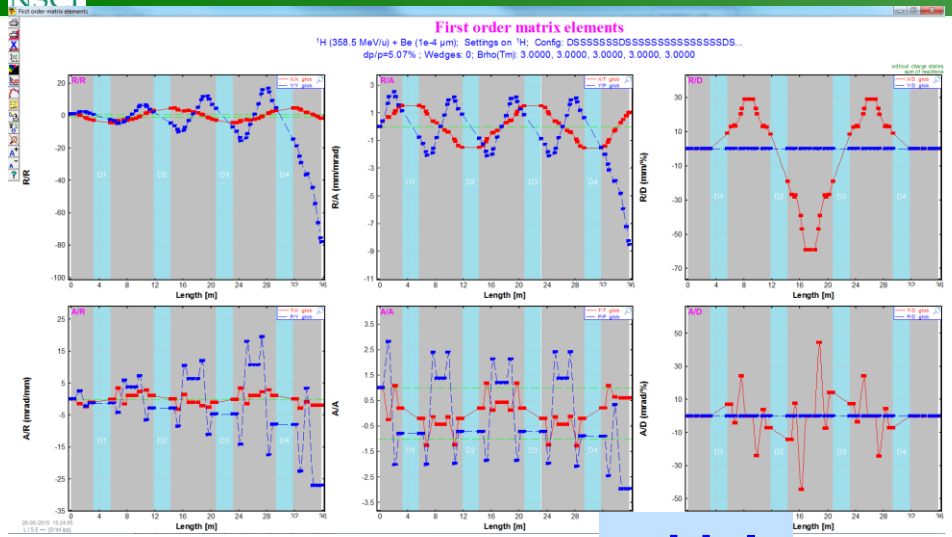


Let's destroy it manually

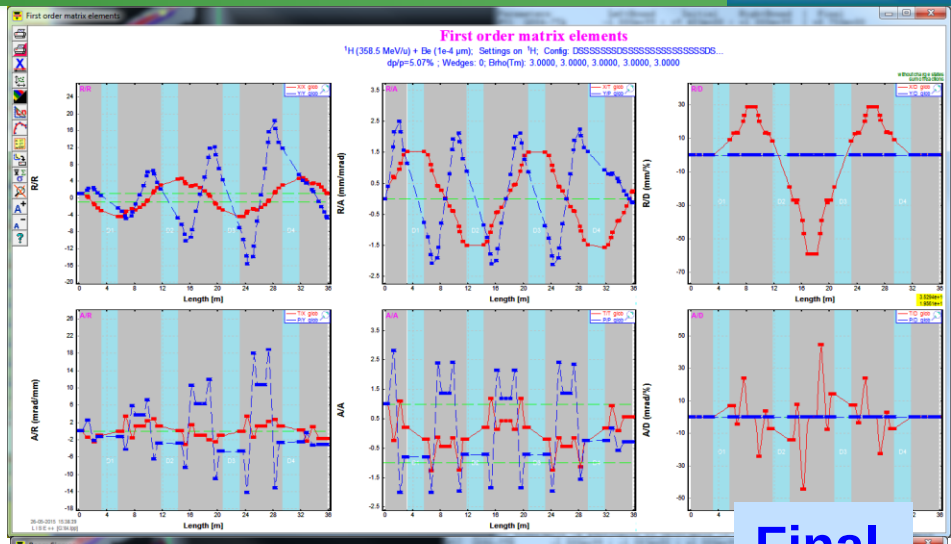




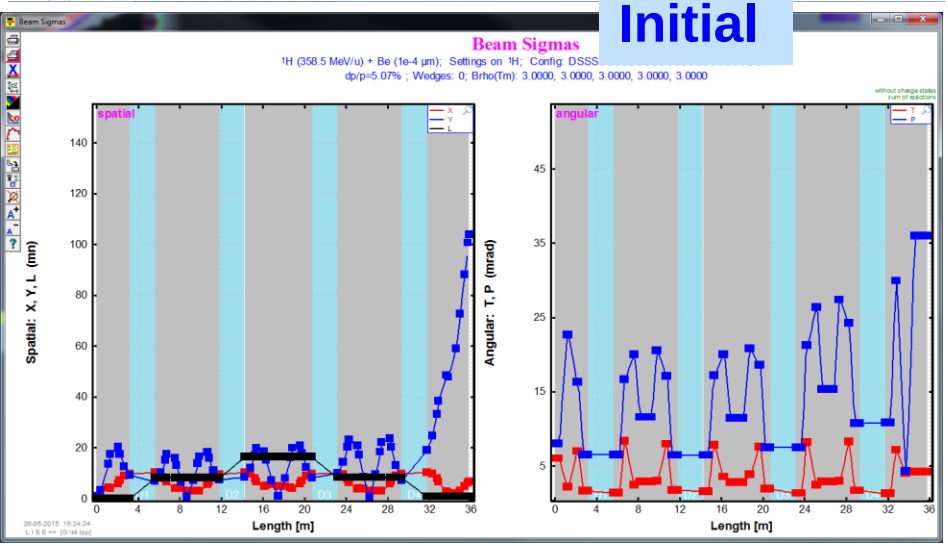
Example for A1900 (1)

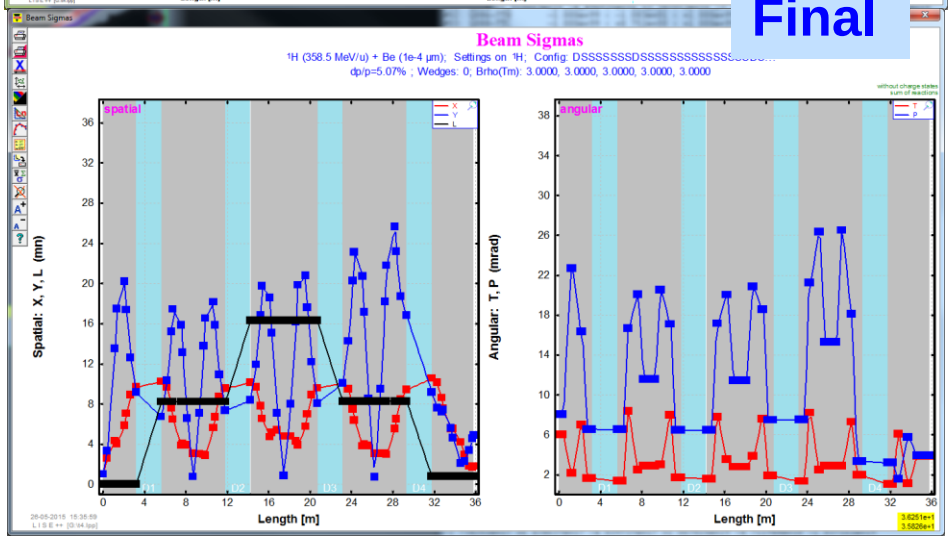


Initial



Final





Initial +870.782 and Final +870.318 LISE fit reduced values

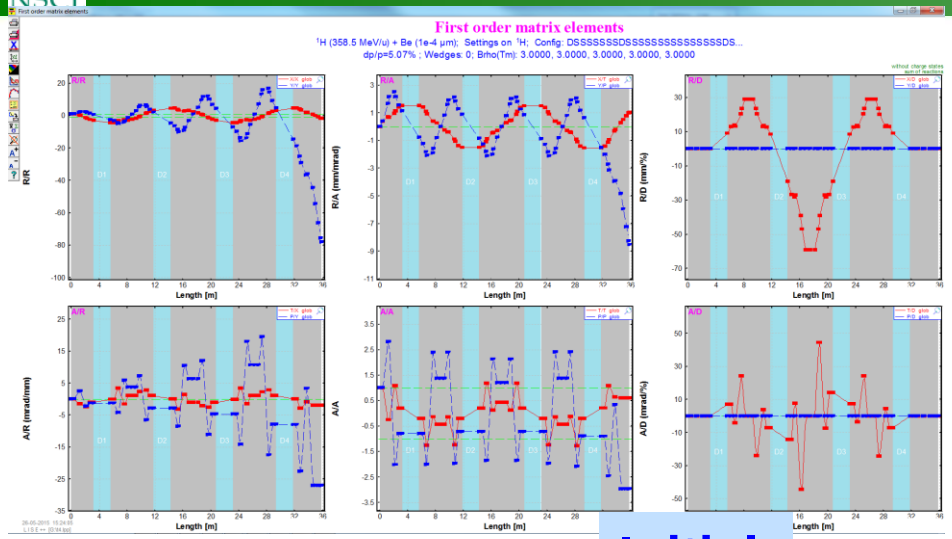
Parameters:	LeftBound	Initial	RightBound	Final
#01: Q084-7TA	-1.000e+99 <	+9.403e+00 <	+1.000e+99	+8.750e+00
#02: Q086-7TB	-1.000e+99 <	-1.083e+01 <	+1.000e+99	-9.228e+00
#03: Q088-7TC	-1.000e+99 <	+8.752e+00 <	+1.000e+99	+7.597e+00
#04: Q098-8TA	-2.000e+01 <	+7.085e+00 <	+2.000e+01	+5.738e+00
#05: Q100-8TB	-4.000e+01 <	-4.000e+00 <	+2.000e+01	-5.247e+00
#06: Q102-8TC	-1.000e+01 <	+4.212e+00 <	+4.000e+01	+4.103e+00

The Quad field value was not restored exactly

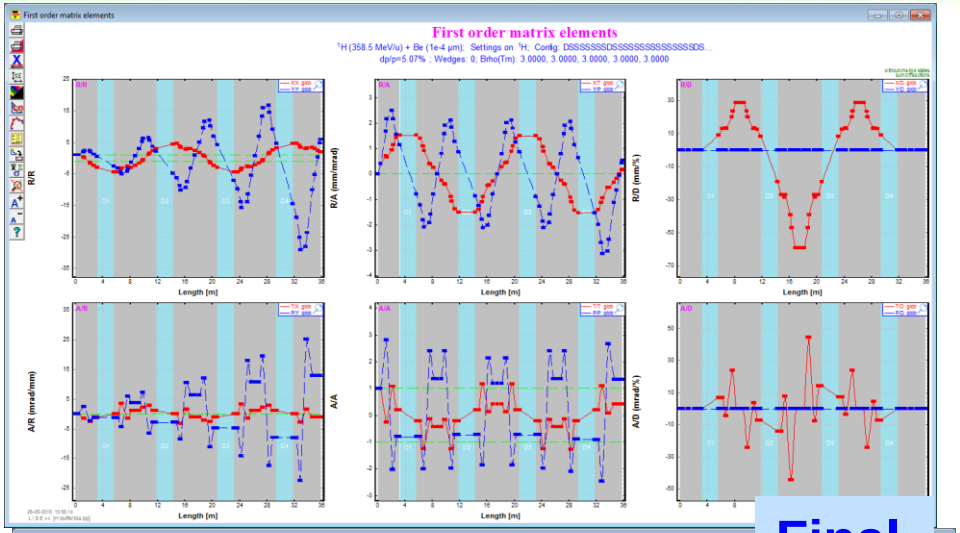
Fitting values:	Initial	Final	Precision	(Fin-Des)/P	Desired
#01: sigY	+2.469e+01	+7.589e+00	1.000e+00	0.000e+00	< +5.00e+01
#02: sigY	+3.323e+01	+7.188e+00	1.000e+00	0.000e+00	< +5.00e+01
#03: sigY	+4.846e+01	+5.595e+00	1.000e+00	0.000e+00	< +5.00e+01
#04: sigY	+5.902e+01	+2.098e+00	1.000e+00	0.000e+00	< +5.00e+01
#05: focusX	+7.758e-01	+5.198e-05	1.000e-03	5.198e-02	= +0.00e+00
#06: focusY	-7.236e+00	+3.918e-05	5.000e-01	7.836e-05	= +0.00e+00
#07: X-dispers	+7.190e-05	+3.171e-15	1.000e-03	3.171e-12	= +0.00e+00
#08: T-dispers	-1.354e-04	-3.960e-15	1.000e-02	3.960e-13	= +0.00e+00
#09: sigmaX	+4.744e+00	+1.780e+00	1.000e-01	2.953e-01	< +3.00e+00
#10: sigmaY	+8.802e+01	+3.395e+00	1.000e-02	2.405e+02	< +1.00e+00

The last constaint was not succesfull

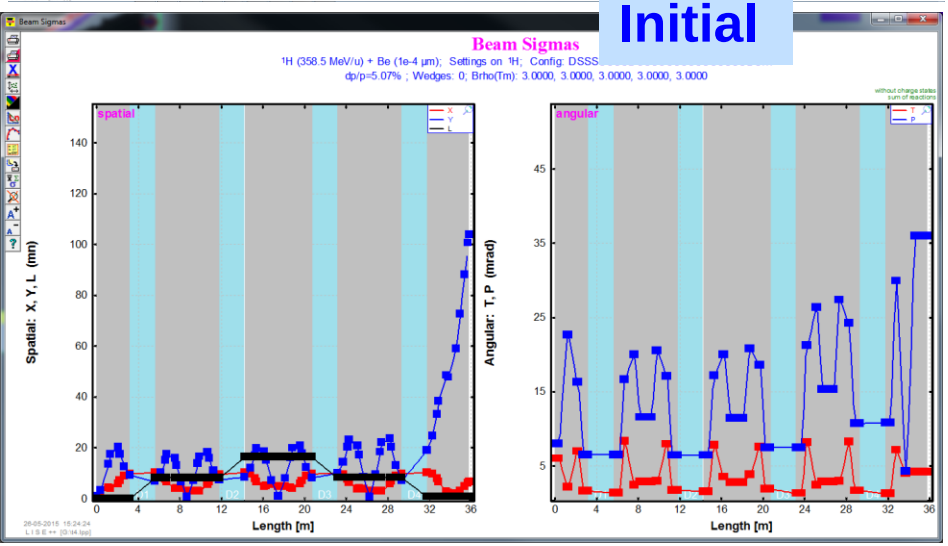
Example for A1900 (2) -- only last triplet to use in fit

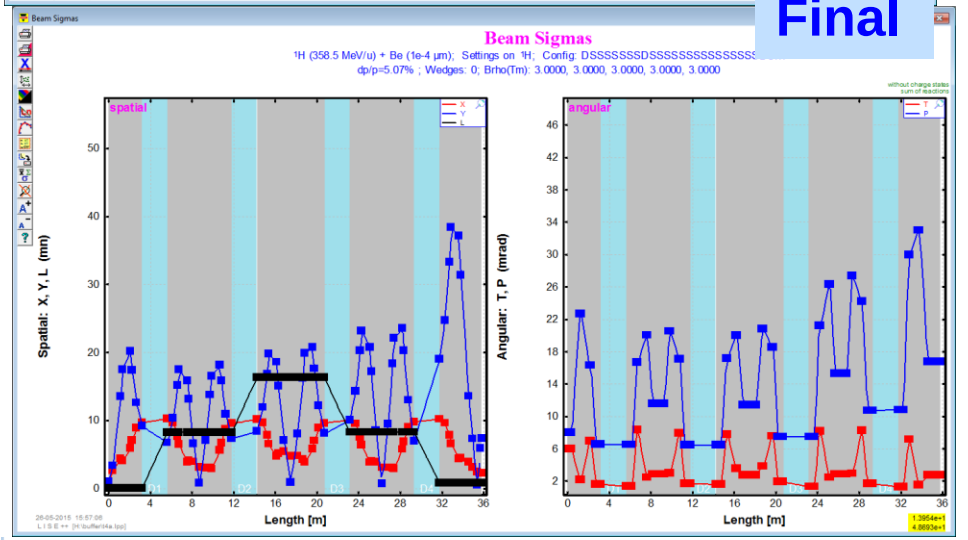


Initial



Final





Initial +870.782 and Final +870.376 LISE fit reduced values

Parameters:	LeftBound	Initial	RightBound	Final
#01: Q100-8TB	−4.000e+01 <	−4.000e+00 <	+2.000e+01	−8.040e+00
#02: Q102-8TC	−1.000e+01 <	+4.212e+00 <	+4.000e+01	+4.040e+00

The Quad field value was restored

Fitting values:	Initial	Final	Precision	(Fin−Des)/P	Desired
#01: sigY	+2.469e+01	+2.469e+01	1.000e+00	1.014e−11	< +5.00e+01
#02: sigY	+3.323e+01	+3.323e+01	1.000e+00	5.224e−08	< +5.00e+01
#03: sigY	+4.846e+01	+3.711e+01	1.000e+00	2.536e−06	< +5.00e+01
14: sigY	+5.902e+01	+1.355e+01	1.000e+00	0.000e+00	< +5.00e+01
15: focusX	+7.758e−01	−1.178e−06	1.000e−03	1.178e−03	= +0.00e+00
16: focusY	−7.236e+00	−4.717e−02	5.000e−01	9.434e−02	= +0.00e+00
17: X-dispers	+7.190e−05	+3.588e−04	1.000e−03	3.588e−01	= +0.00e+00
18: T-dispers	−1.354e−04	−5.358e−05	1.000e−02	5.358e−03	= +0.00e+00
19: sigmaX	+4.744e+00	+2.414e+00	1.000e−01	5.564e−01	< +3.00e+00
#10: sigmaY	+8.802e+01	+4.777e−01	1.000e−02	5.931e−01	< +1.00e+00

All constraints are good!

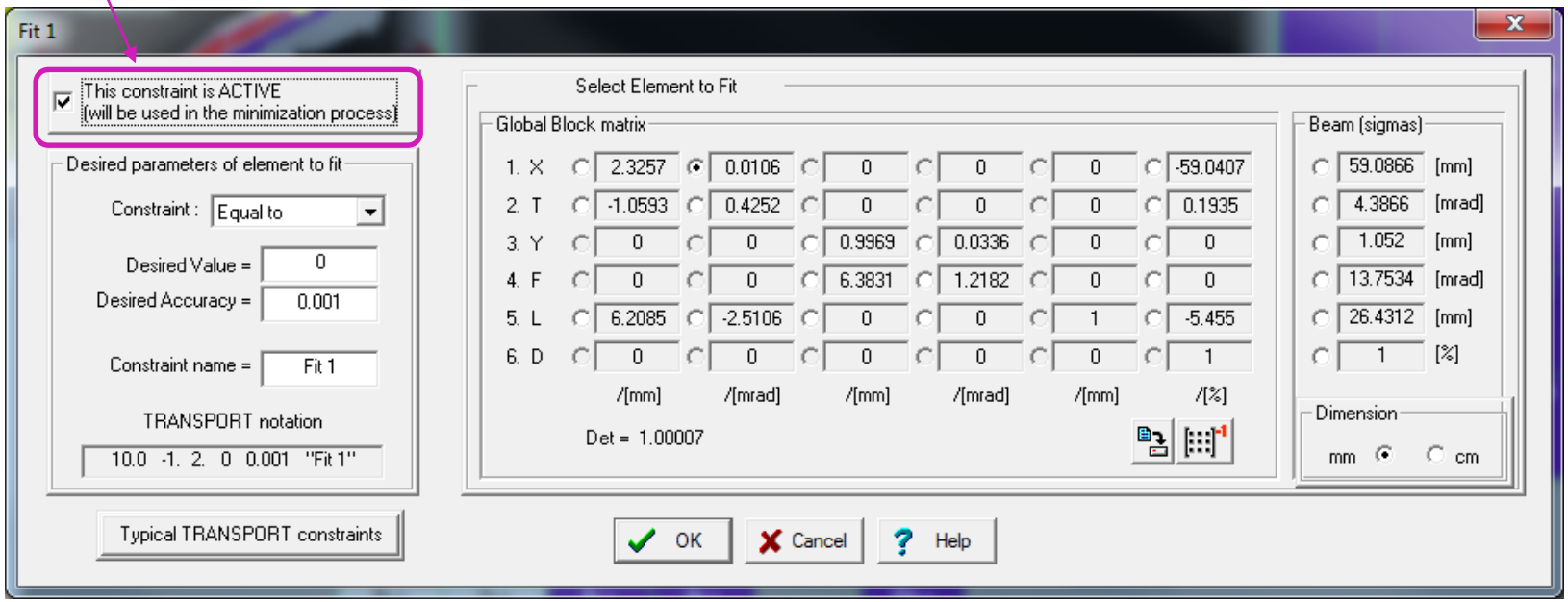
Optics Minimization in LISE^{++} : update

Update
v.9.10.119
from 06/23/15

1. Fitting constraint block : new option “Active”
2. Change the Fitting option "Active" in the “Fast Edit Optics” dialog
3. New Option in the Preferences dialog: show "Fit" blocks in the Scheme and Setup windows
4. Appearance of Fitting constraint blocks in Menus, Dialog, Windows
5. Call the "Fast Edit Optics" dialog from the Optics optimization dialog
6. User Break in the Minimization process
7. Miscellaneous for fitting procedure
8. Miscellaneous for v.9.10.119

Fitting constraint block : New Option “Active”

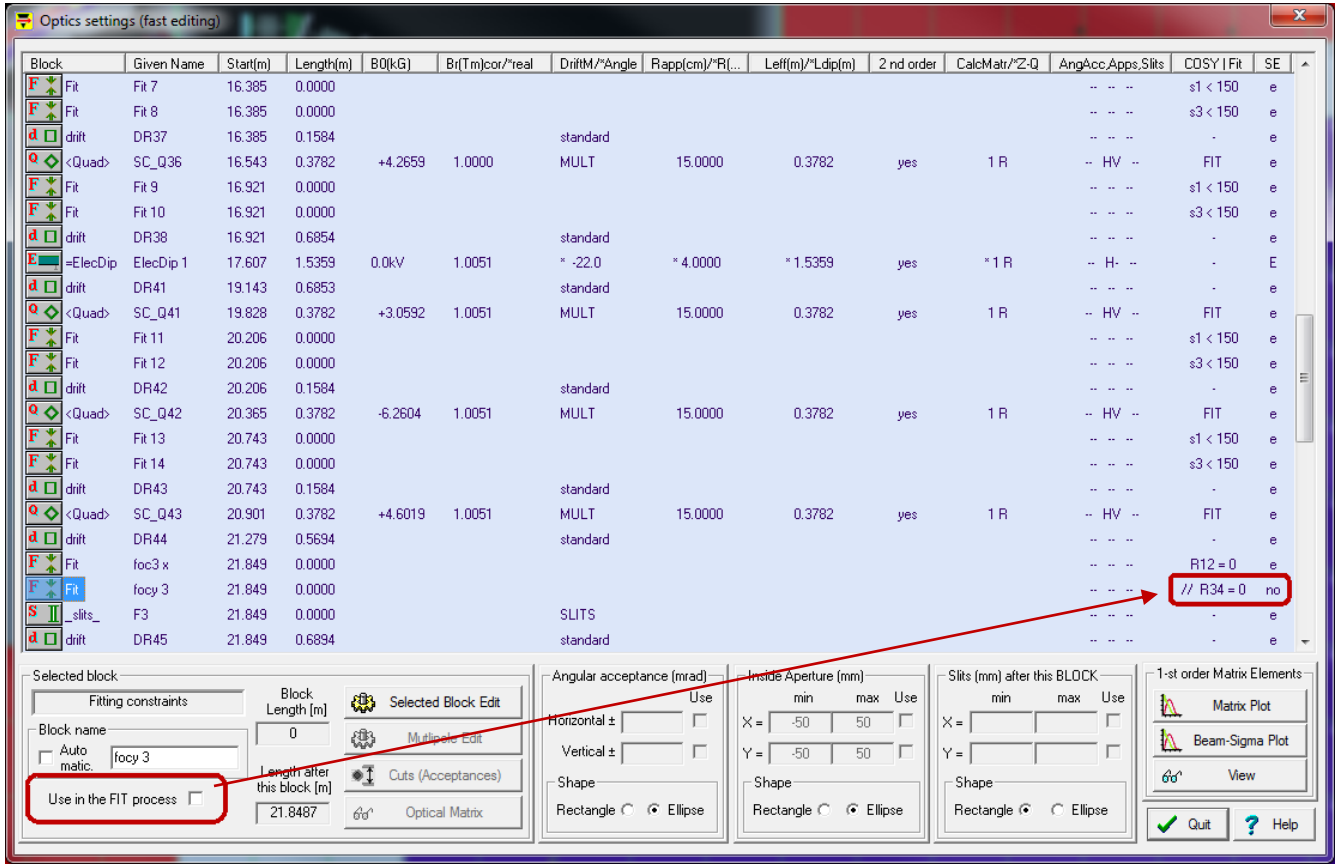
New option

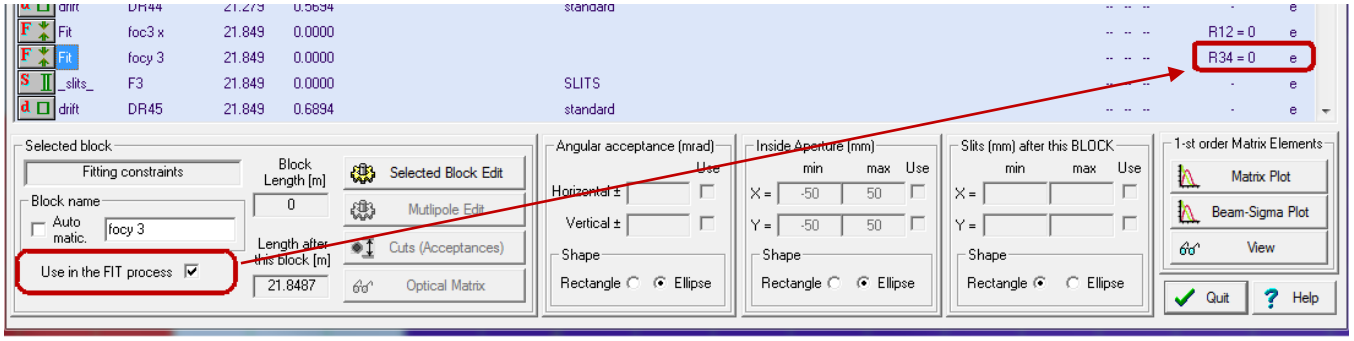


- If it is not “active”, then the block will still enable in the Setup and FastEditOptics dialogs, but it won’t be shown in The Scheme and Setup windows
- This property it can be easily changed from the “Fast Edit Optics” dialog (see the next slide)

Change the Fitting option "Active" in the “Fast Edit Optics” dialog

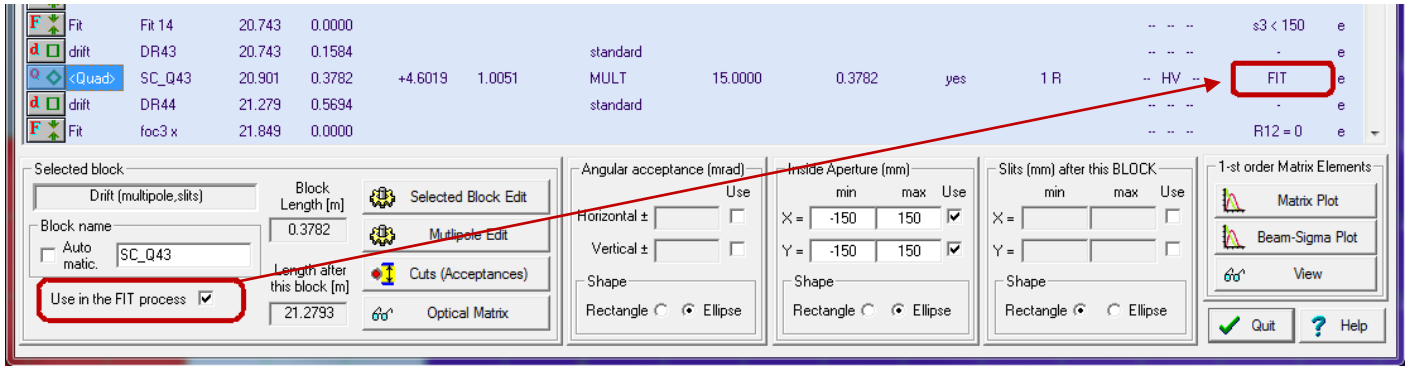
Application for [Fitting constraint blocks]

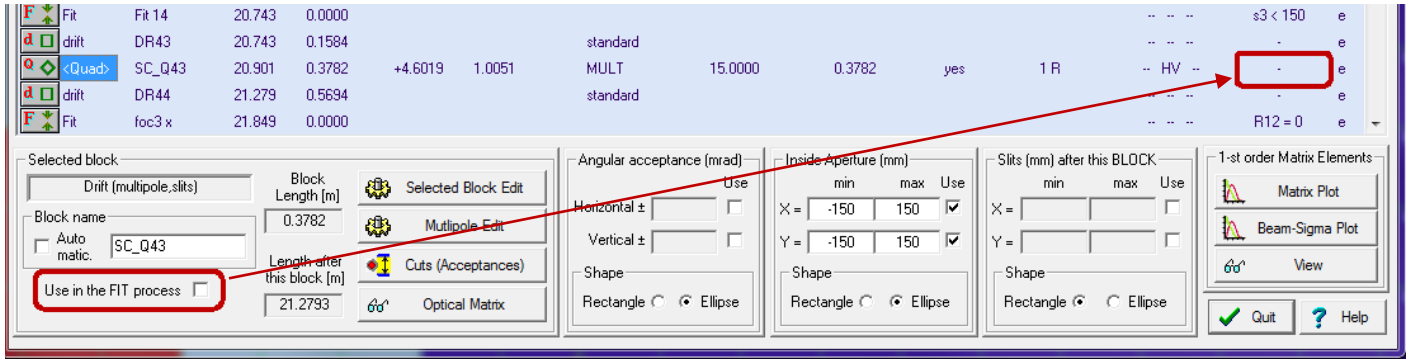




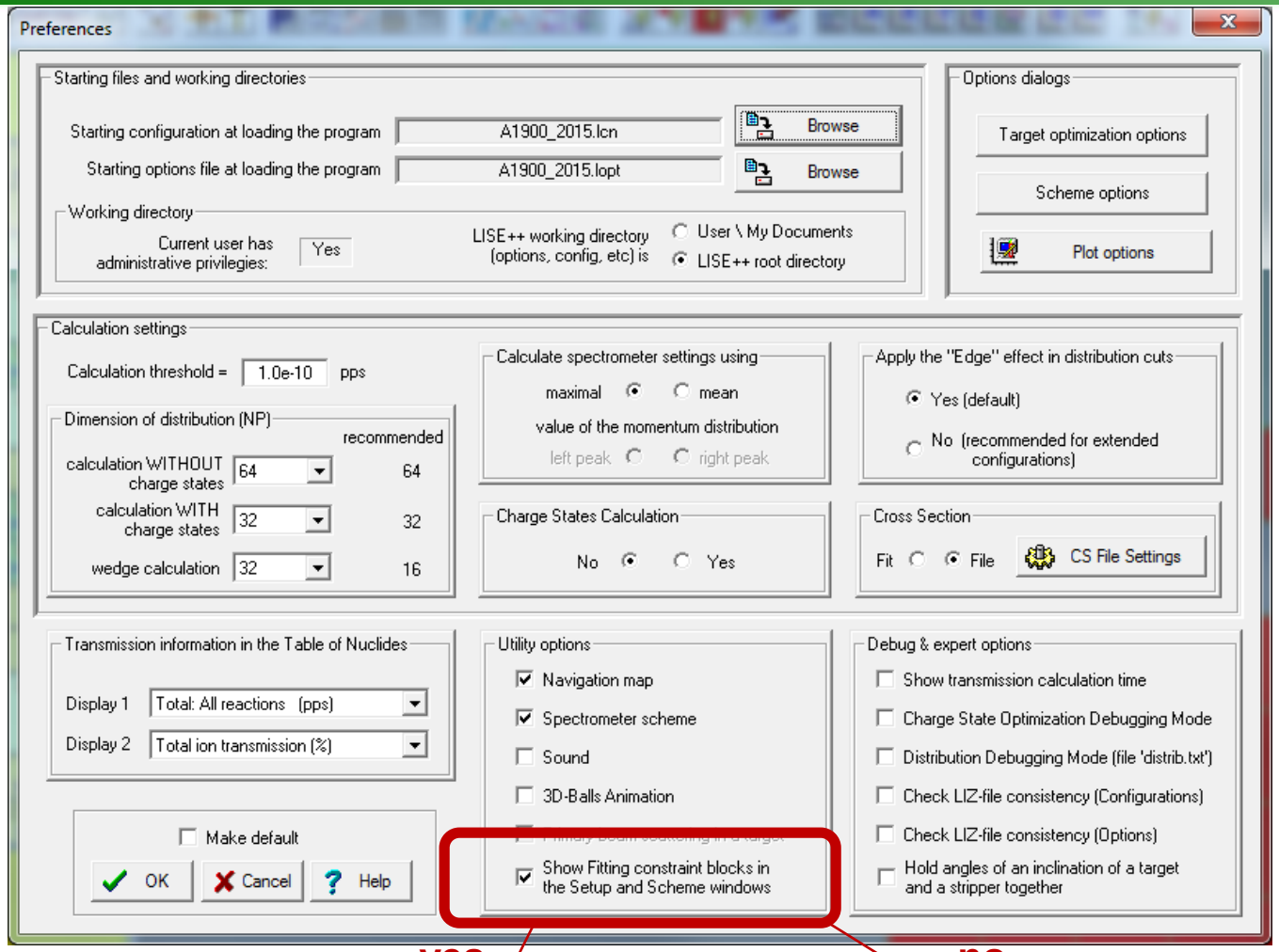
Change the option “Use in the Fit process" in the “Fast Edit Optics” dialog

Application for
M & E quadrupoles
blocks

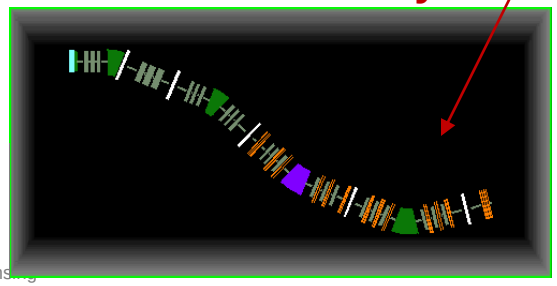




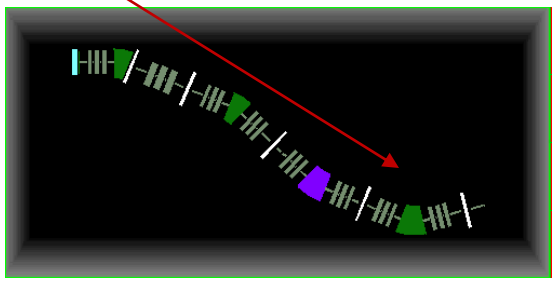
New Option in the Preferences dialog: show "Fit" blocks in the Scheme and Setup windows



yes



no

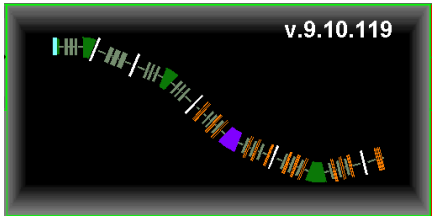


Appearance of Fitting constraint blocks in Menus, Dialog, Windows

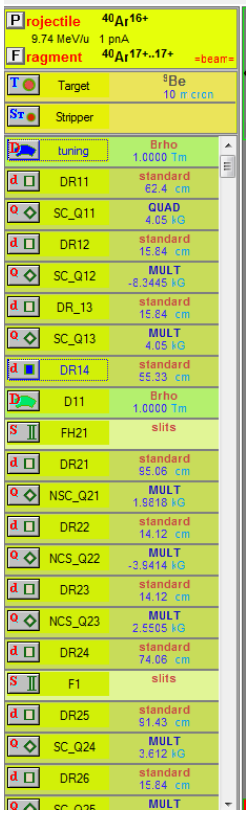
Window	Show
Setup window	only Active*
Setup scheme	only Active*
Setup dialog ("Spectrometer design")	always
Optics settings dialog (fast editing)	always
Menus	never
Monte Carlo calculation of fragment transmission	never

** - show only "Active", if the corresponding option is set the "Preferences"diaolg*

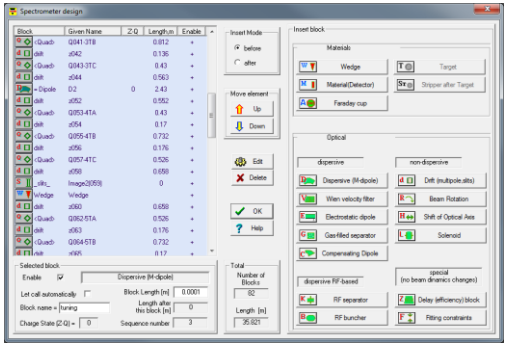
Setup scheme



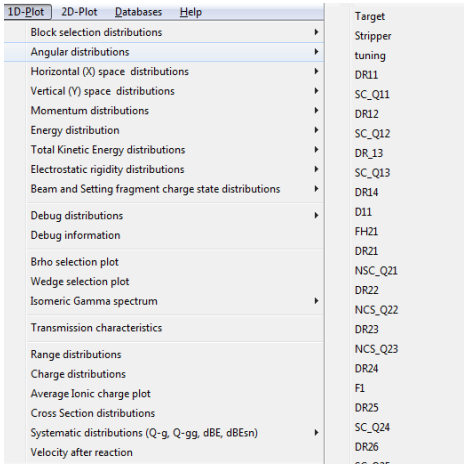
Setup window



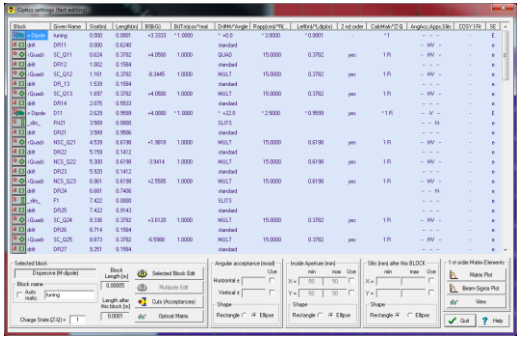
Setup dialog ("Spectrometer design")



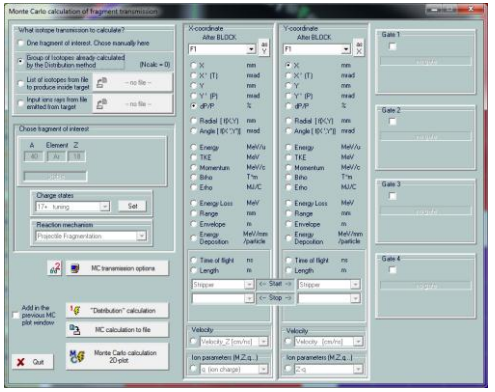
Menus



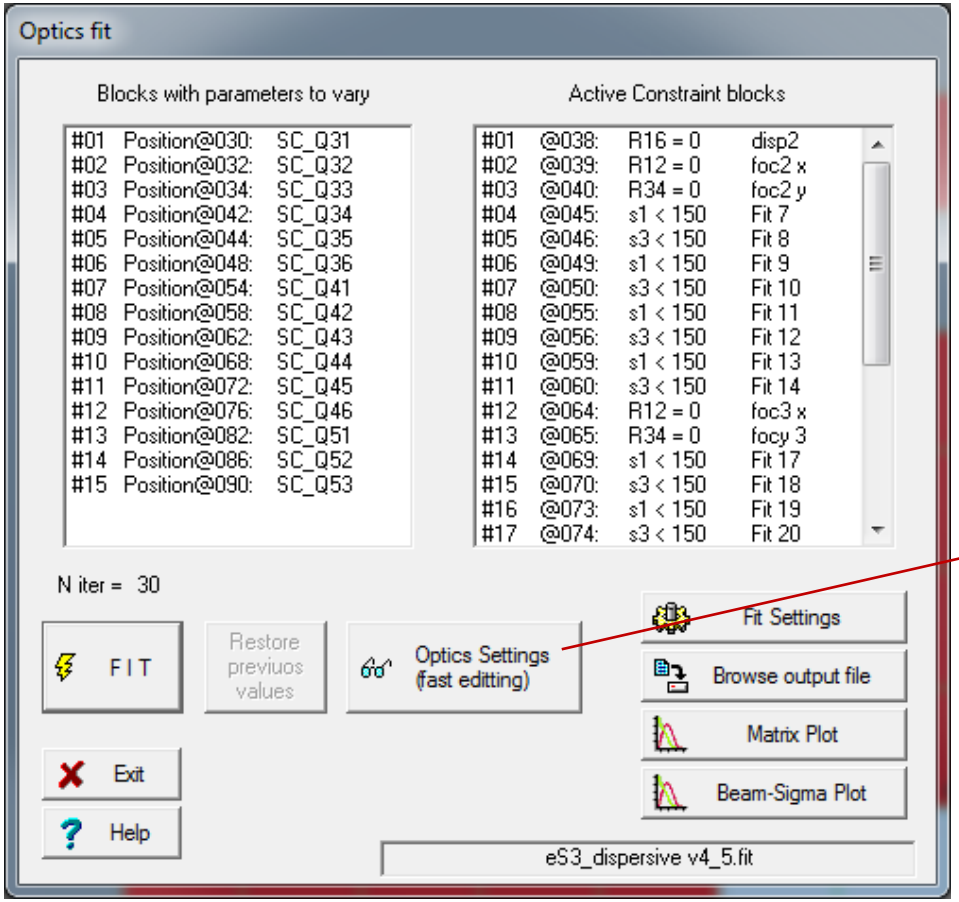
Optics settings (fast editing)

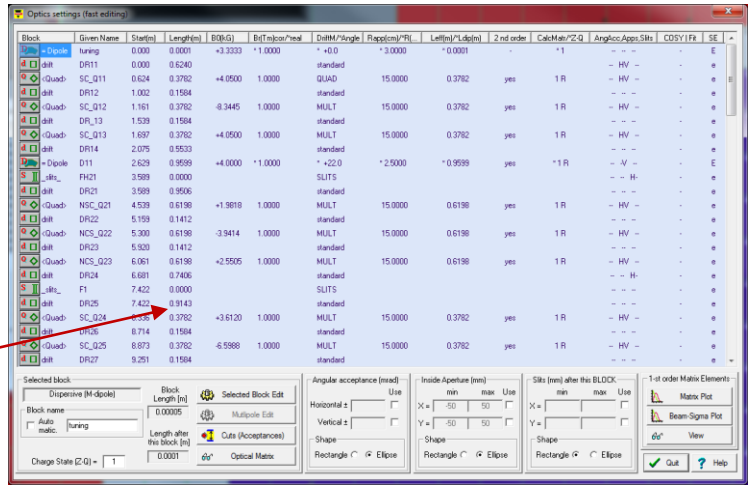


Monte Carlo calculation of fragment transmission



Call the "Fast Edit Optics" dialog from the Optics optimization dialog

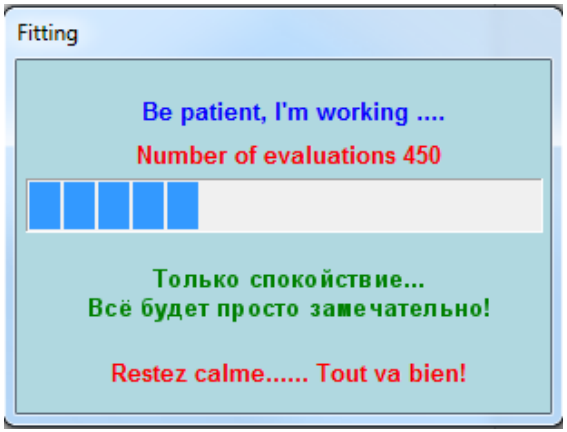




Without leaving the "Optics fit" dialog it is possible to load the Fast Optics Edit" dialog where you can set/unset "Active" properties, modify Fitting constraint block parameters, or enter a Quad field value.

User Break in the Minimization process

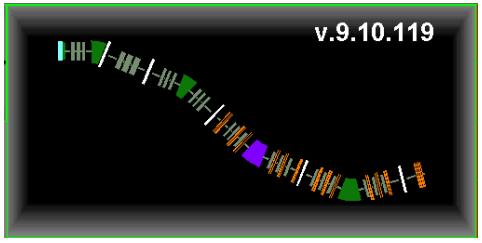
Press the "Escape" button to cancel the minimization procedure



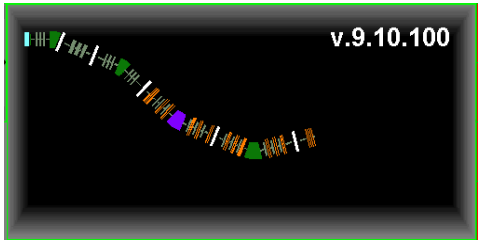

```
#26: foc4 x          +1.036e-02   +2.328e-03    1.000e-03   2.328e+00   | = +0.00e+00
#27: foc4 y          +7.946e-02   -3.101e-02    1.000e-01   3.101e-01   | = +0.00e+00
#28: disp4           +1.077e+01   +1.048e-02    1.000e-02   1.048e+00   | = +0.00e+00
#29: beamy4          +3.799e+00   +1.446e+01    1.000e-01   3.913e-03   | < +2.00e+01

------------------------------------------------------------------------------

==>  Results for eS3_dispersive v4_5.fit:
Termination reason: USER BREAK after 600 evaluations!

Minimization info:
0: 1.160e+06         ||e||_2 at initial p
1: 1.153e+02         ||e||_2
2: 3.298e+03         ||J^T e||_inf
3: 7.967e-29         ||Dp||_2
4: 1.377e-05         mu/max[J^T J]_ii
5: 300               # iterations
```

Miscellaneous for fitting procedure

- New functions BLOCKnext and BLOCKprevious with "noFit" options



- Corrections for Separator scheme in the case of Fitting block



- "SetFocus" back in the Fast Edit Optics dialog

- Minimization output modification

#12:	SC_Q46	+0.0e+00	<	+3.296e+00	<	+1.0e+01	+2.890e+00
#13:	SC_Q51	+0.0e+00	<	+2.517e+00	<	+1.0e+01	+2.798e+00
#14:	SC_Q52	-1.0e+01	<	-5.905e+00	<	+0.0e+00	-5.845e+00
#15:	SC_Q53	+0.0e+00	<	+3.202e+00	<	+1.0e+01	+3.456e+00

Fitting values:	Initial	Final	Precision	(Fin-Des)/P		Desired
#01:	disp2	+1.158e-01	+8.581e-02	1.0e-02	+8.581e+00	= 0
#02:	foc2 x	+3.150e-04	-3.143e-04	1.0e-03	+3.143e-01	= 0
#03:	foc2 y	+5.912e-01	+6.146e-01	1.0e-01	+6.146e+00	= 0
#04:	Fit 7	+2.487e+01	+3.303e+01	1.0e-01	0	< 150
#05:	Fit 8	+2.314e+01	+3.266e+01	1.0e-01	0	< 150
#06:	Fit 9	+3.827e+01	+5.153e+01	1.0e-01	0	< 150
#07:	Fit 10	+1.665e+01	+2.289e+01	1.0e-01	0	< 150
#08:	Fit 11	+1.014e+02	+8.777e+01	1.0e-01	0	< 150
#09:	Fit 12	+2.119e+01	+2.143e+01	1.0e-01	0	< 150
#10:	Fit 13	+1.433e+02	+1.278e+02	1.0e-01	0	< 150
#11:	Fit 14	+2.145e+01	+2.170e+01	1.0e-01	0	< 150
#12:	foc3 x	-5.376e-03	-2.484e-03	1.0e-02	+2.484e-01	= 0
#13:	focy 3	-5.088e-02	-4.954e-02	1.0e-01	+4.954e-01	= 0
#14:	Fit 17	+1.487e+02	+1.239e+02	1.0e-01	0	< 150
#15:	Fit 18	+2.367e+00	+2.228e+00	1.0e-01	0	< 150
#16:	Fit 19	+6.993e+01	+6.356e+01	1.0e-01	0	< 150
#17:	Fit 20	+5.935e+00	+6.123e-01	1.0e-01	0	< 150
#18:	Fit 21	+2.821e+01	+3.745e+01	1.0e-01	0	< 150

2nd Order Optics Minimization in LISE++

Purpose:
2nd order optics minimization of existed separators

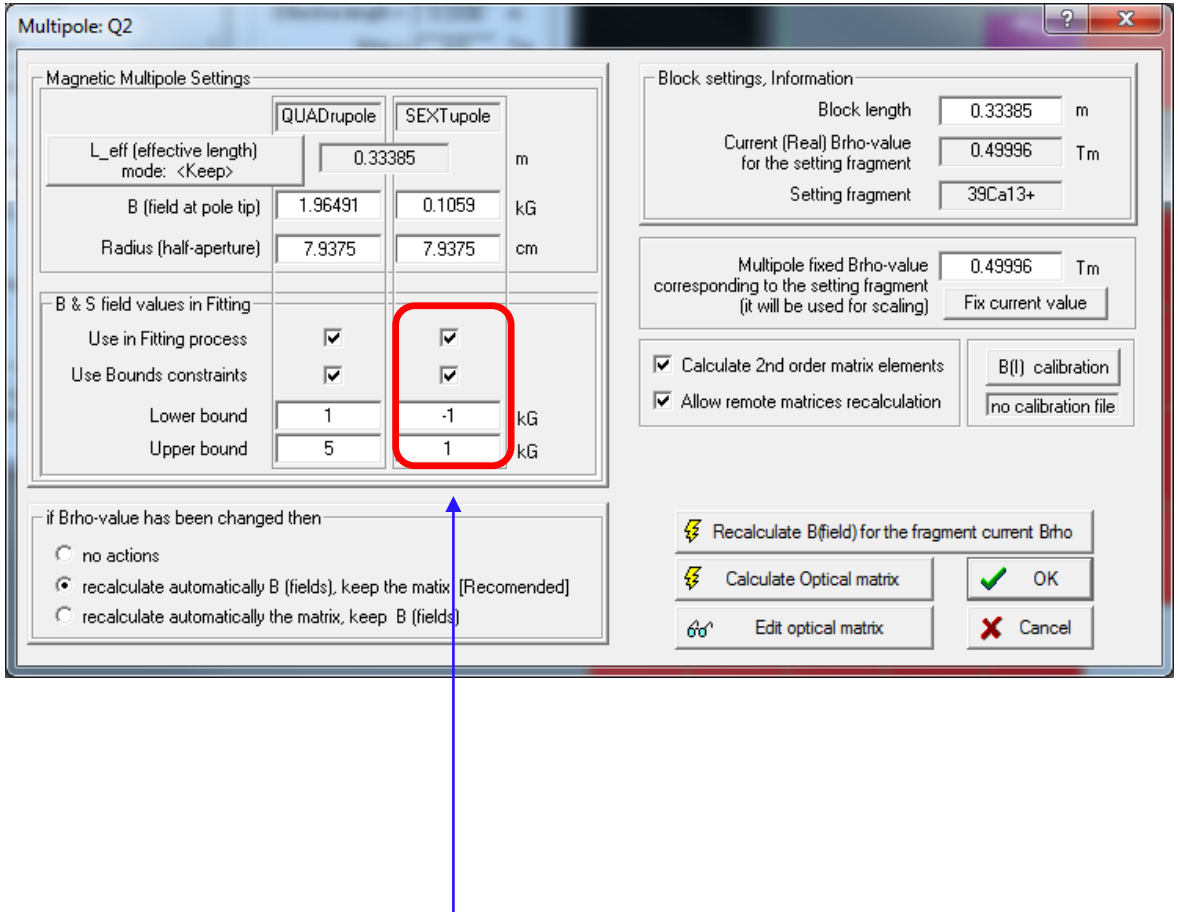
v.9.10.156
from 07/27/15

1. **Introduction: First order optics minimization in LISE++ (links)**
2. **Modification of M-multipole blocks for 2nd order optimization**
3. **Modification of the "Fitting constraints" block for 2nd order**
   - *Aberrations list*
4. **Run new minimization version**
   - *High order truncation problem*
5. **"OptBeam" : New Beam sigma vector used only in the Optics Optimization**

Additionally:

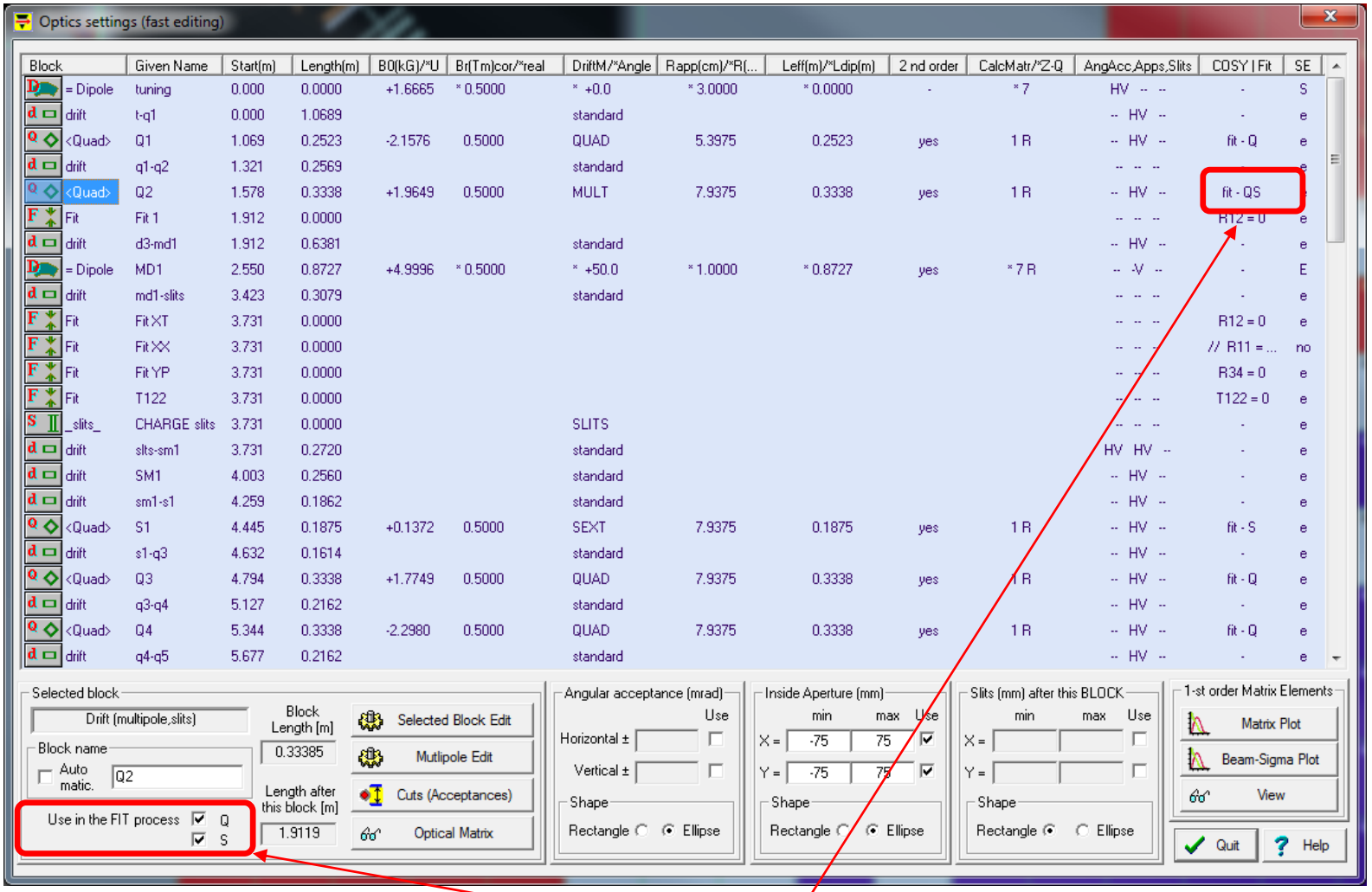
v.9.10.142	22-07-15	Using LISE++ optics minimization for the DRAGON2000 extended configuration development

Modification of M-multipole blocks for 2nd order optimization



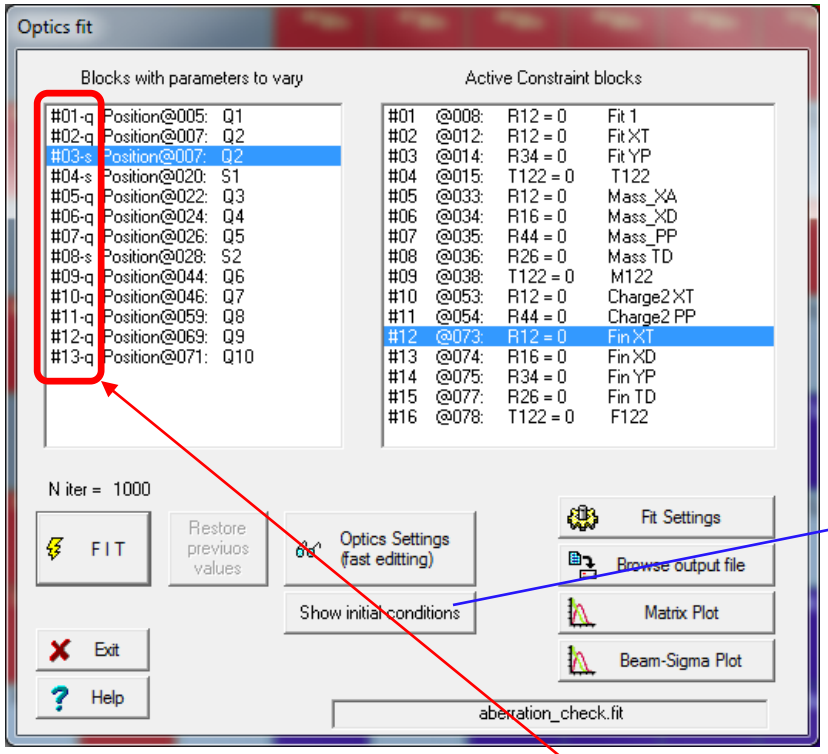
New feature: using sextupole fields in the Optimization process.
Evidently, it has an effect only in the 2nd order optimization

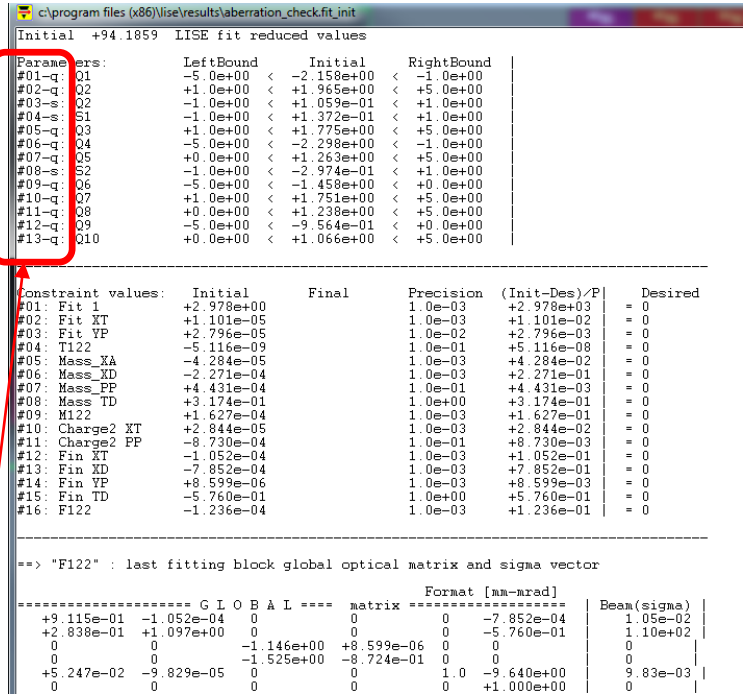
The "Optics settings" dialog and Sextupole fields for optimization



It is possible to change quickly the use of Quadrupole and/or Sextupole fields from the Optics settings dialog

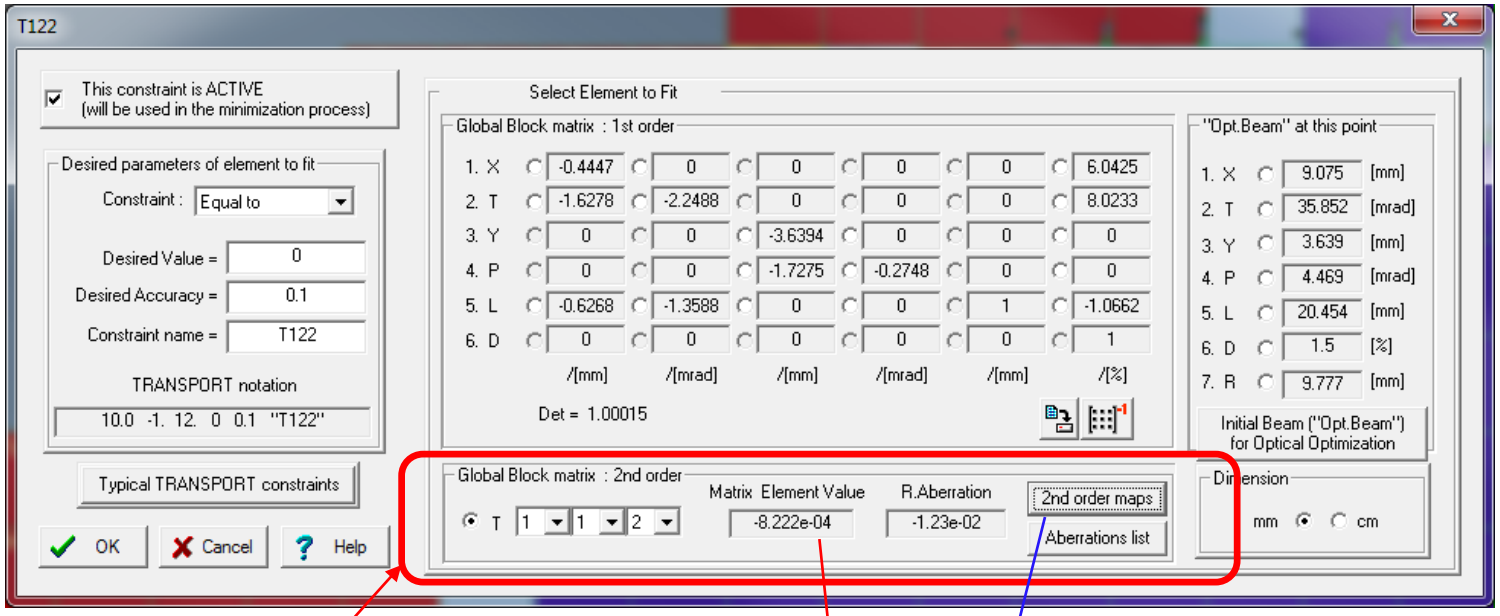
The “Optics settings” dialog and Sextupole fields for optimization



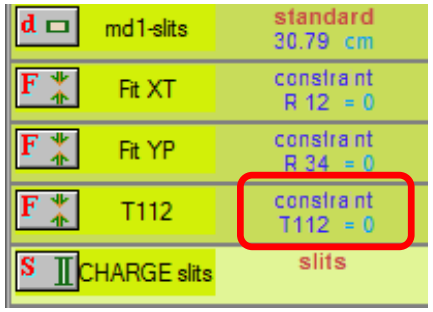


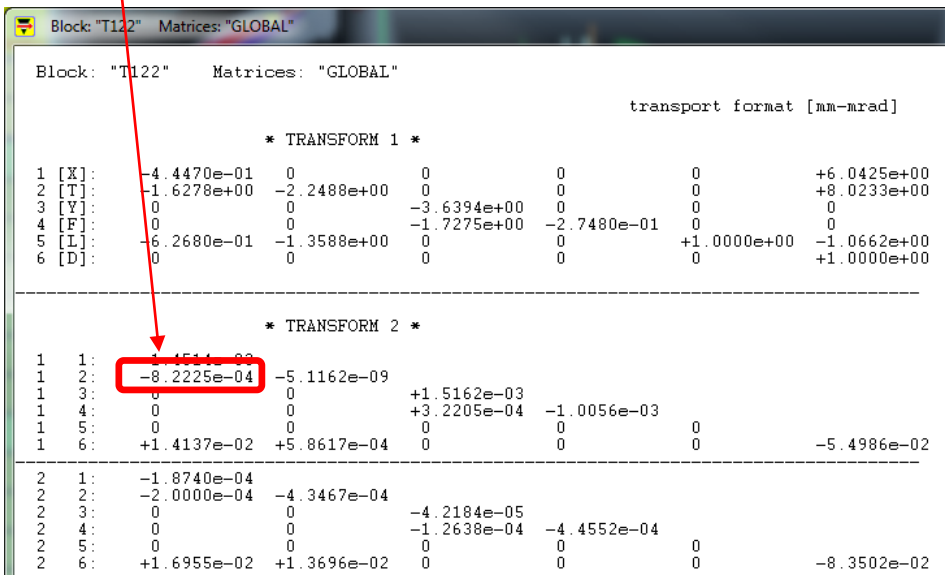
Chars “q” and “s” show what component (“q”uadrupole or “s”extupole) of multipole is used in the Optimization process

Modification of the "Fit constraint" block for 2nd order optimization

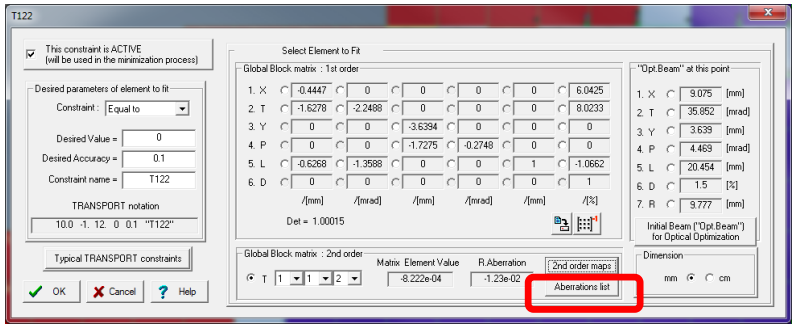


New feature: Selection of 2nd order matrix element for the "Optimization" process





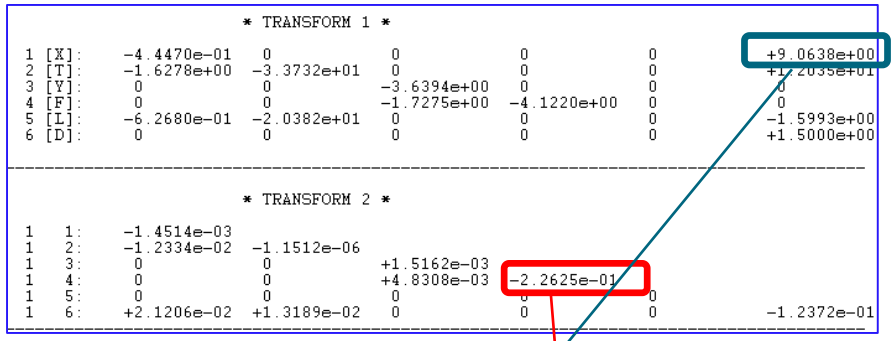
"Fit constraint" block for 2nd order optimization : Aberrations List



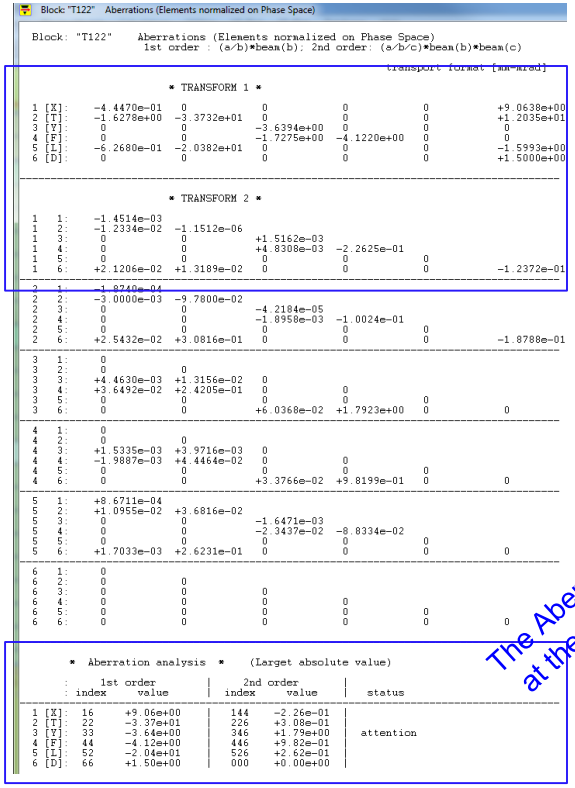
The Aberration list purpose : show a 2nd element to pay attention. Aberration value is:

1st order : $V_1 = (a|b)*s(b)$

2nd order : $V_2 = (a|bc)*s(b)*s(c)$

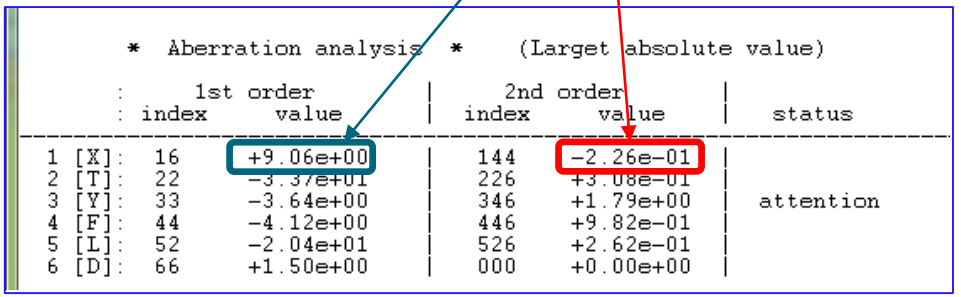


```
                    * TRANSFORM 1 *
1 [X]:   -4.4470e-01   0            0            0            0          +9.0638e+00
2 [T]:   -1.6278e+00  -3.3732e+01   0            0            0          +1.2035e+01
3 [Y]:    0            0           -3.6394e+00   0            0           0
4 [F]:    0            0           -1.7275e+00  -4.1220e+00   0           0
5 [L]:   -6.2680e-01  -2.0382e+01   0            0            0          -1.5993e+00
6 [D]:    0            0           0            0            0          +1.5000e+00

                    * TRANSFORM 2 *
1   1:   -1.4514e-03
1   2:   -1.2334e-02  -1.1512e-06
1   3:    0            0           +1.5162e-03
1   4:    0            0           +4.8308e-03  -2.2625e-01
1   5:    0            0           0            0            0
1   6:   +2.1206e-02  +1.3189e-02   0            0            0          -1.2372e-01
```



```
Block: "T122"   Aberrations (Elements normalized on Phase Space)
                    * TRANSFORM 1 *
1 [X]:   -4.4470e-01   0            0            0            0          +9.0638e+00
2 [T]:   -1.6278e+00  -3.3732e+01   0            0            0          +1.2035e+01
3 [Y]:    0            0           -3.6394e+00   0            0           0
4 [F]:    0            0           -1.7275e+00  -4.1220e+00   0           0
5 [L]:   -6.2680e-01  -2.0382e+01   0            0            0          -1.5993e+00
6 [D]:    0            0           0            0            0          +1.5000e+00
                    * TRANSFORM 2 *
1   1:   -1.4514e-03
1   2:   -1.2334e-02  -1.1512e-06
1   3:    0            0           +1.5162e-03
1   4:    0            0           +4.8308e-03  -2.2625e-01
1   5:    0            0           0            0            0
1   6:   +2.1206e-02  +1.3189e-02   0            0            0          -1.2372e-01
2   1:   -1.8740e-04
2   2:   -3.0000e-03  -9.7800e-02
2   3:    0            0           -4.2184e-05
2   4:    0            0           -1.8958e-03  -1.0024e-01
2   5:    0            0           0            0            0
2   6:   +2.5432e-02  +3.0816e-01   0            0            0          -1.8788e-01
3   1:    0
3   2:    0            0
3   3:   +4.4630e-03  +1.3156e-02   0
3   4:   +3.6492e-02  +2.4205e-01   0            0
3   5:    0            0           0            0            0
3   6:    0            0           +6.0368e-02  +1.7923e+00   0           0
4   1:    0
4   2:    0            0
4   3:   +1.5335e-03  +3.9716e-03   0
4   4:   -1.9887e-03  +4.4464e-02   0            0
4   5:    0            0           0            0            0
4   6:    0            0           +3.3766e-02  +9.8199e-01   0           0
5   1:   +8.6711e-04
5   2:   +1.0955e-02  +3.6816e-02
5   3:    0            0           -1.6471e-03
5   4:    0            0           -2.3437e-02  -8.8334e-02
5   5:    0            0           0            0            0
5   6:   +1.7033e-03  +2.6231e-01   0            0            0          0
6   1:    0
6   2:    0            0
6   3:    0            0           0
6   4:    0            0           0            0
6   5:    0            0           0            0            0
6   6:    0            0           0            0            0          0

    * Aberration analysis *    (Largest absolute value)
    :     1st order          |     2nd order          |
    : index     value        | index     value        |   status
1 [X]:  16     +9.06e+00     |  144     -2.26e-01     |
2 [T]:  22     -3.37e+01     |  226     +3.08e-01     |
3 [Y]:  33     -3.64e+00     |  346     +1.79e+00     |   attention
4 [F]:  44     -4.12e+00     |  446     +9.82e-01     |
5 [L]:  52     -2.04e+01     |  526     +2.62e-01     |
6 [D]:  66     +1.50e+00     |  000     +0.00e+00     |
```

The Aberration analysis table at the bottom of the window



```
    *  Aberration analysis  *    (Largest absolute value)
    :     1st order          |     2nd order          |
    : index     value        | index     value        |   status
1 [X]:  16     +9.06e+00     |  144     -2.26e-01     |
2 [T]:  22     -3.37e+01     |  226     +3.08e-01     |
3 [Y]:  33     -3.64e+00     |  346     +1.79e+00     |   attention
4 [F]:  44     -4.12e+00     |  446     +9.82e-01     |
5 [L]:  52     -2.04e+01     |  526     +2.62e-01     |
6 [D]:  66     +1.50e+00     |  000     +0.00e+00     |
```

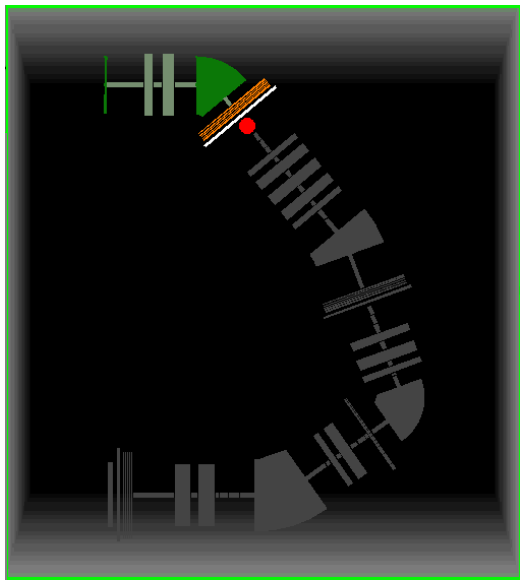
Comparison of absolute largest 1st and 2nd orders aberrations

Status messages

$V_1 < 3\ V_2$	attention
$V_1 < V_2$	Critical
$V_1 < V_2/3$	VERY critical

Run 2nd order optics minimization

The 1st part of DRAGON2000 : Target – Charge Slits



Initial settings

Q1

Magnetic Multipole Settings

	QUADrupole	SEXTupole	
L_eff (effective length) mode: <Keep>	0.2523		m
B (field at pole tip)	-2.1576	0	kG
Radius (half-aperture)	5.3975	5.3975	cm
B & S field values in Fitting			
Use in Fitting process	☑	☐	
Use Bounds constraints	☑	☐	
Lower bound	-5	-100	kG
Upper bound	-1	100	kG

Q2

Magnetic Multipole Settings

	QUADrupole	SEXTupole	
L_eff (effective length) mode: <Keep>	0.33385		m
B (field at pole tip)	1.96503	0.106	kG
Radius (half-aperture)	7.9375	7.9375	cm
B & S field values in Fitting			
Use in Fitting process	☑	☑	
Use Bounds constraints	☑	☑	
Lower bound	1	-1	kG
Upper bound	5	1	kG

Beam vector used for Optical Optimization

"Opt.Beam"

1. X	2	mm
2. T	20	mrad
3. Y	2	mm
4. P	20	mrad
5. L	0	mm
6. D	2	%

(•) mm ( ) cm

Ok

Cancel

Aberration list @ Charge Slits

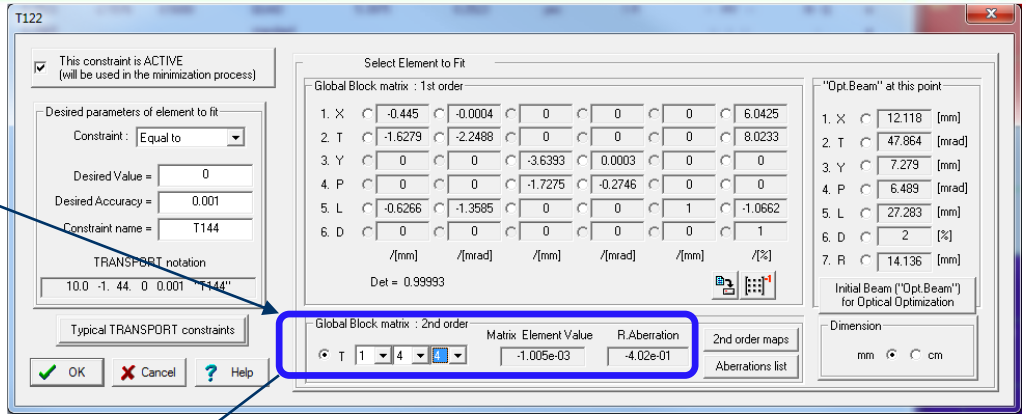
Block: "Fit YP" Aberrations (Elements normalized on Phase Space)

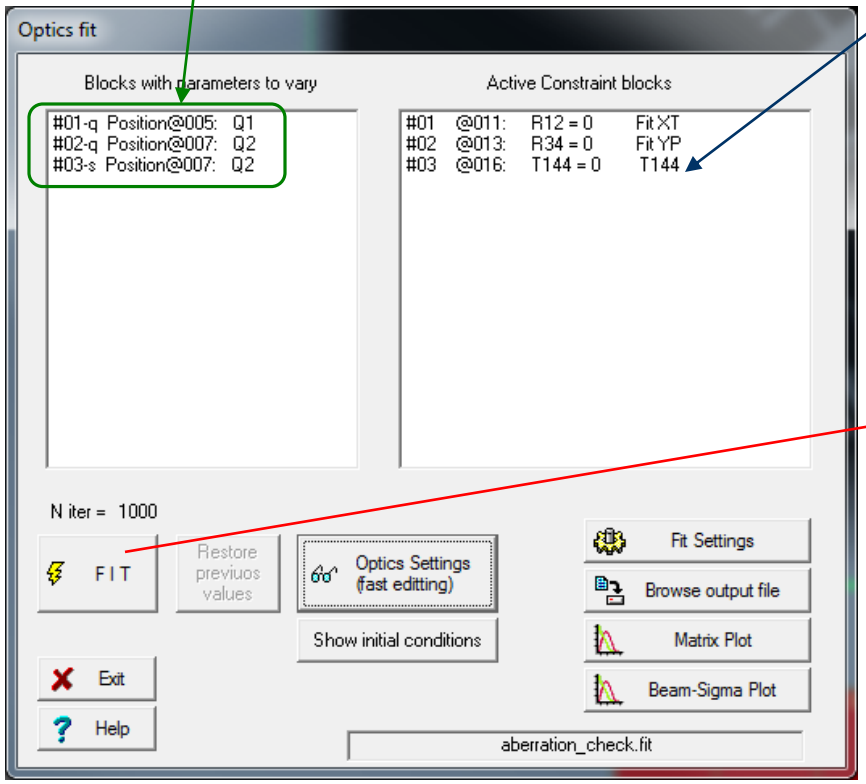
```
Block: "Fit YP"     Aberrations (Elements normalized on Phase Space)
                  1st order : (a/b)*beam(b); 2nd order: (a/b/c)*beam(b)*beam(c)

                                                          transport format [mm-mrad]

                          * TRANSFORM 1 *

1 [X]:    -8.9000e-01  -1.0000e-02   0            0            0            +1.2085e+01
2 [T]:    -3.2558e+00  -4.4978e+01   0            0            0            +1.6047e+01
3 [Y]:     0            0           -7.2784e+00  +4.0000e-03   0            0
4 [F]:     0            0           -3.4548e+00  -5.4940e+00   0            0
5 [L]:    -1.2532e+00  -2.7170e+01   0            0            0            -2.1324e+00
6 [D]:     0            0           0            0            0            +2.0000e+00

-------------------------------------------------------------------------------------

                          * TRANSFORM 2 *

1   1:    -5.8101e-03
1   2:    -3.2928e-02  -8.9324e-05
1   3:     0            0           +6.0714e-03
1   4:     0            0           +1.2939e-02  -4.0210e-01
1   5:     0            0           0            0            0
1   6:    +5.6559e-02  +2.3549e-02   0            0            0            -2.1994e-01
-------------------------------------------------------------------------------------
2   1:    -7.5022e-04
2   2:    -8.0057e-03  -1.7388e-01
2   3:     0            0           -1.6769e-04
2   4:     0            0           -5.0458e-03  -1.7819e-01
2   5:     0            0           0            0            0
2   6:    +6.7822e-02  +5.4786e-01   0            0            0            -3.3401e-01
-------------------------------------------------------------------------------------
3   1:     0
3   2:     0            0
3   3:    +1.7870e-02  +3.5159e-02   0
3   4:    +9.7387e-02  +4.3065e-01   0            0
3   5:     0            0           0            0            0
3   6:     0            0           +1.6097e-01  +3.1863e+00   0            0
-------------------------------------------------------------------------------------
4   1:     0
4   2:     0            0
4   3:    +6.1427e-03  +1.0628e-02   0
4   4:    -5.2666e-03  +7.9210e-02   0            0
4   5:     0            0           0            0            0
4   6:     0            0           +9.0040e-02  +1.7458e+00   0            0
-------------------------------------------------------------------------------------

        *  Aberration analysis  *     (Absolute largest value)

        :     1st order          |     2nd order          |
        : index     value        | index     value        |   status
-------------------------------------------------------------------------------
1 [X]:   16      +1.21e+01       |  144     -4.02e-01      |
2 [T]:   22      -4.50e+01       |  226     +3.46e-01      |
3 [Y]:   33      -7.28e+00       |  346     +3.19e+00      |   attention
4 [F]:   44      -5.49e+00       |  446     +1.75e+00      |
5 [L]:   52      -2.72e+01       |  526     +4.66e-01      |
6 [D]:   66      +2.00e+00       |  000     +0.00e+00      |
```

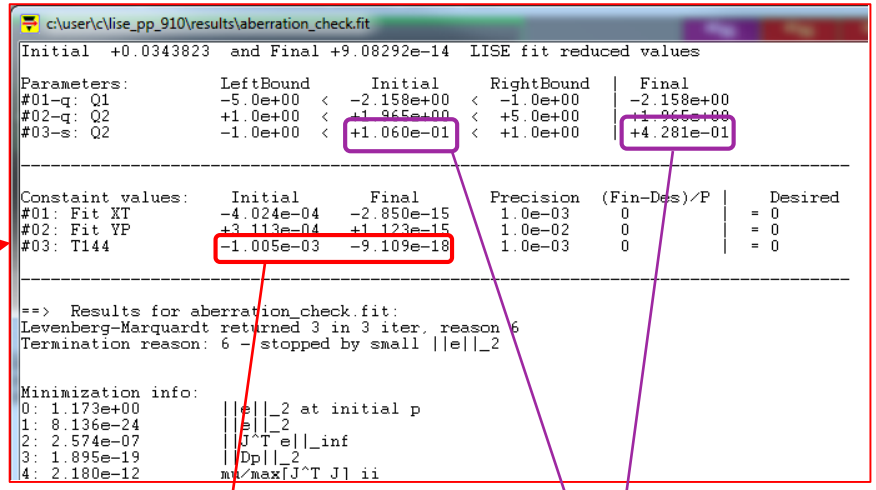
Let's try to neglect the element T144 in the optimization using Q1 & Q2 multipoles!

Run 2nd order optics minimization

Let's try to neglect the element T144 in the optimization using Q1 & Q2 multipoles!







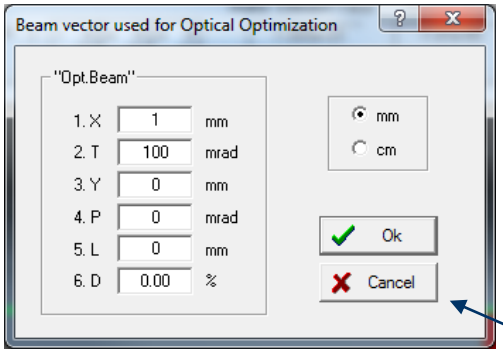
During the optimization the T144 element has been changed from 1.00e-3 to -9.1e-18

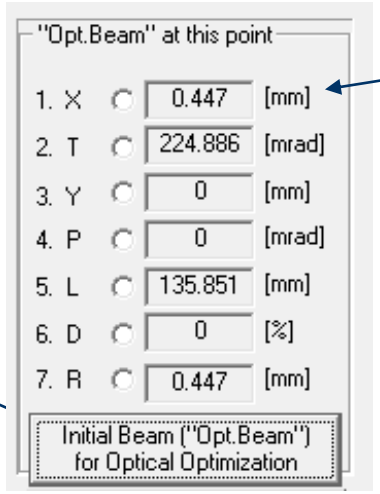
Sextupole field Q2-s has been changed from 0.106 kG to 0.428 kG

Quadrupole fields Q1-q & Q2-q have not been changed

High order truncation problem

Let's create the Beam vector for optimization with large X' component





First order X-sigma is small @ Charge Slits

```
Block: "T144"     Aberrations (Elements r
                    1st order : (a/b)*bear


                          * TRANSFORM 1 *

1 [X]:     -4.4500e-01  -4.0000e-02    0
2 [T]:     -1.6279e+00  -2.2488e+02    0
3 [Y]:      0             0            0
4 [F]:      0             0            0
5 [L]:     -6.2660e-01  -1.3585e+02    0
6 [D]:      0             0            0
-----------------------------------------

                          * TRANSFORM 2 *

1   1:     -1.4525e-03
1   2:     -8.2320e-02  -2.2331e-03
1   3:      0             0            0
1   4:      0             0            0
1   5:      0             0            0
1   6:      0             0            0
-----------------------------------------
2   1:     -1.8755e-04
2   2:     -2.0014e-02  -4.3470e+00
2   3:      0             0            0
2   4:      0             0            0
2   5:      0             0            0
2   6:      0             0            0
```

Aberration list @ Charge Slits

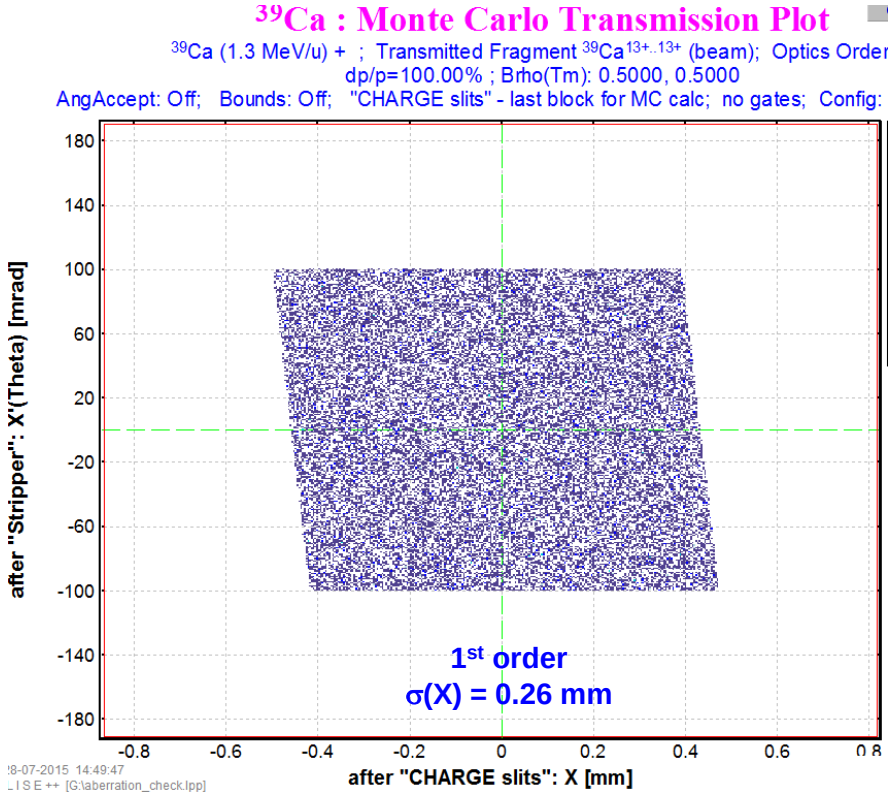
```
*  Aberration analysis  *    (Absolute largest value)

      :     1st order          |     2nd order          |
      : index     value        | index     value        |   status
-------------------------------------------------------------------
1 [X]:  11      -4.45e-01      |  112     -8.23e-02      |
2 [T]:  22      -2.25e+02      |  222     -4.35e+00      |
3 [Y]:  00      +0.00e+00      |  000     +0.00e+00      |
4 [F]:  00      +0.00e+00      |  000     +0.00e+00      |
5 [L]:  52      -1.36e+02      |  522     +1.64e+00      |
6 [D]:  00      +0.00e+00      |  000     +0.00e+00      |
```

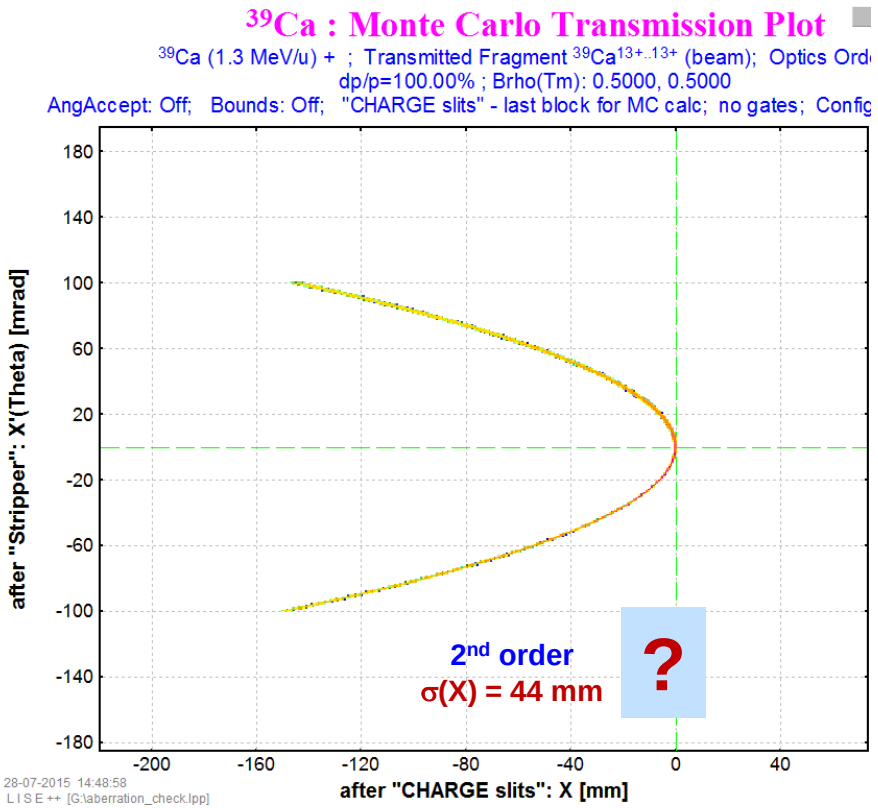
No large aberrations with using GLOBAL matrices!

High order truncation problem

But Let's run Monte Carlo calculation
the similar beam emittance

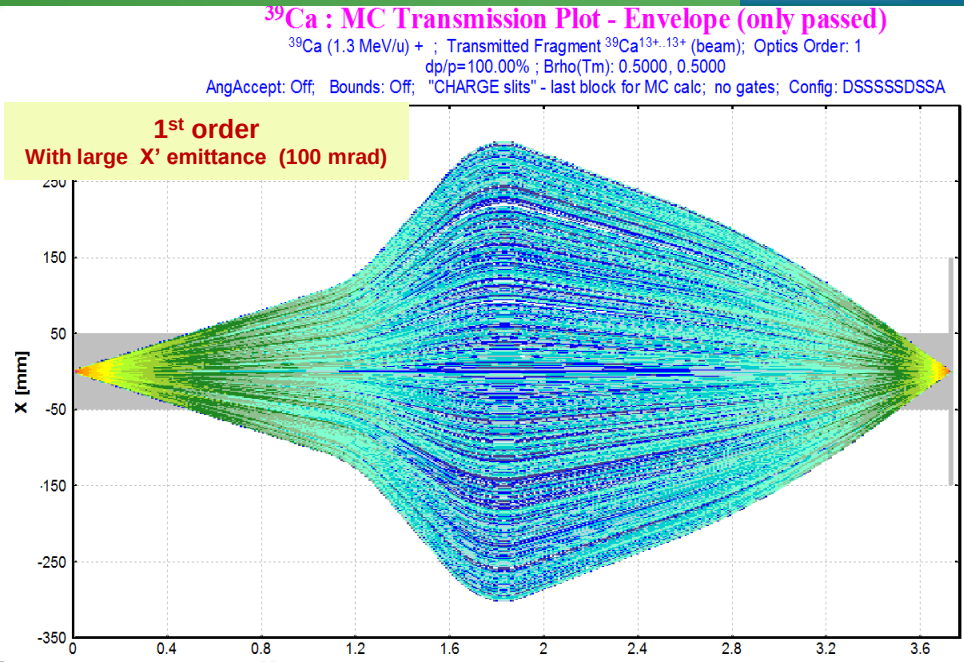
1. X	mm	1	Rectangle uniform
2. T	mrad	100	Rectangle uniform
3. Y	mm	0	Rectangle uniform
4. P	mrad	0	Rectangle uniform
5. L	mm	0	Gaussian
6. D	%	0	Rectangle uniform

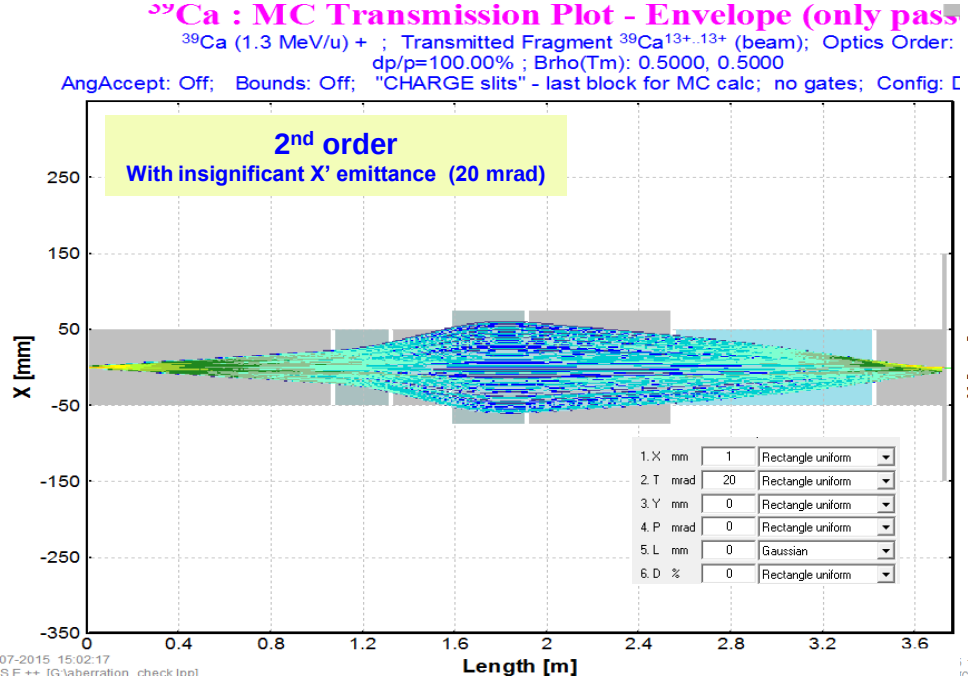


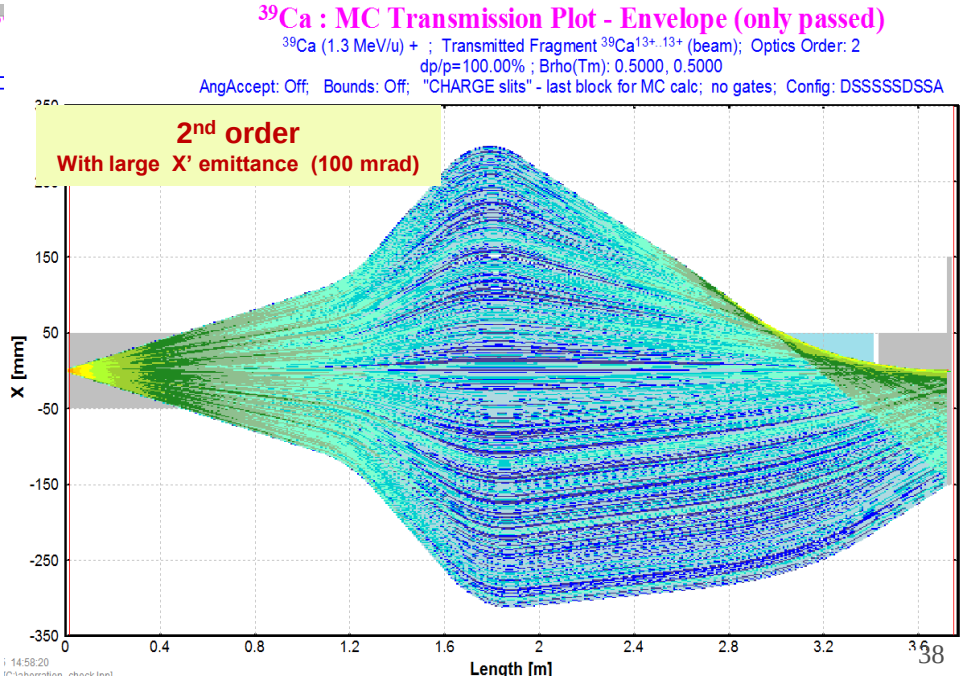


High order truncation problem

- We did not observe large values in the Aberration list calculated with the Global matrices
- Large aberrations have been observed in Monte Carlo calculations where local 2nd order matrices are used
- It is happen due to truncation of high order values with matrices multiplication
- Recommendation: not to use large Beam sigma values for optimization. 2nd order optic optimization operates with small emittances
- For large emittance the LISE++ optimization should be updated to work with Monte Carlo results.



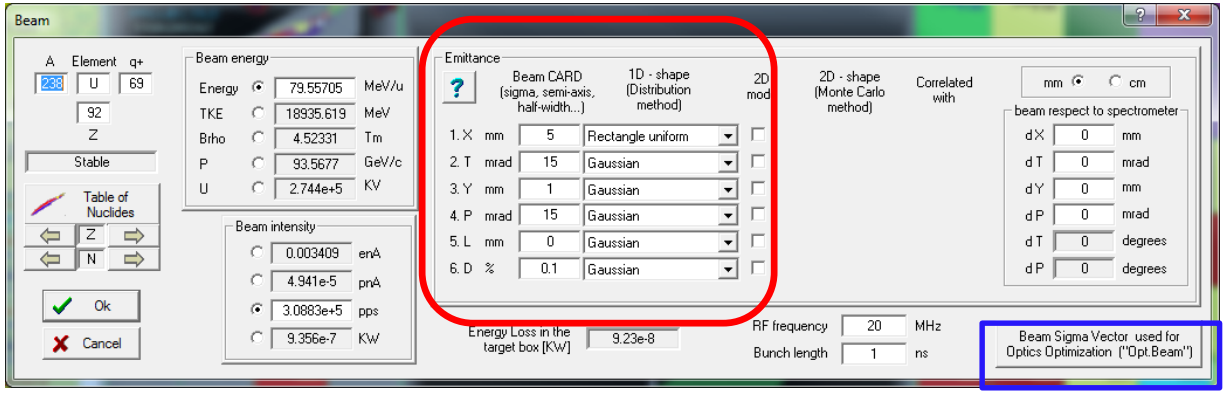




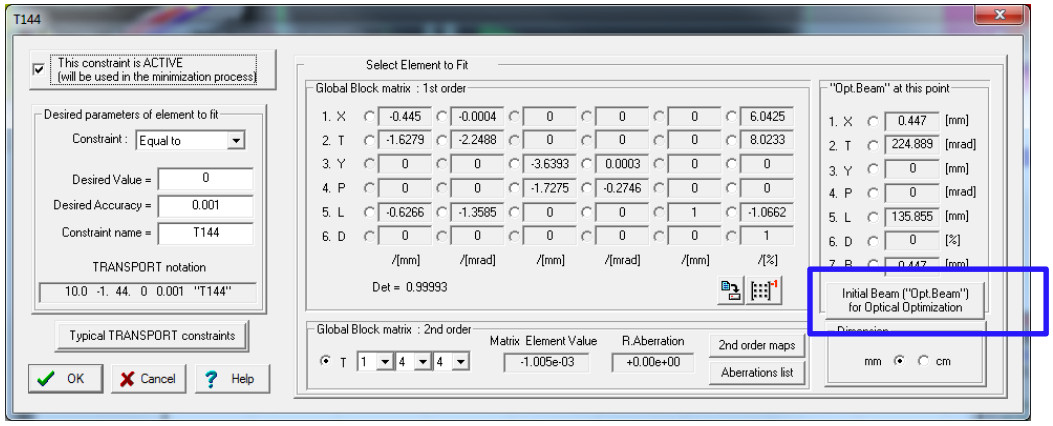
“OptBeam” : New Beam sigma vector used only in the Optics Optimization

“Main” Beam emittance is used for transmission calculations in both (“Distribution” and MC) modes, for matrices in Optical Block dialogs and so on.

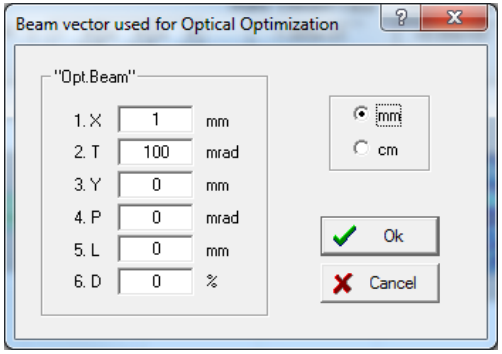
“Beam” dialog



“Fit constraint” dialog



New Beam sigma vector (“OptBeam”) used only for optics optimization in the Fit Constraint block dialog and the Optics Fit dialog.
Access to the dialog to this bema vector modification can be done through the Beam or Fit Constraint block dialogs



This new beam vector for optimization has been created to avoid additional beam emittance editing due to switching between fragment transmission calculations and the optimization mode, which done for the primary beam

Acknowledgements

to

Drs. M.Hausmann, and M.Portilio (NSCL/MSU),

for fruitful discussions.