Optimization of Optical Imaging Systems

Florian Bociort

Delft University of Technology

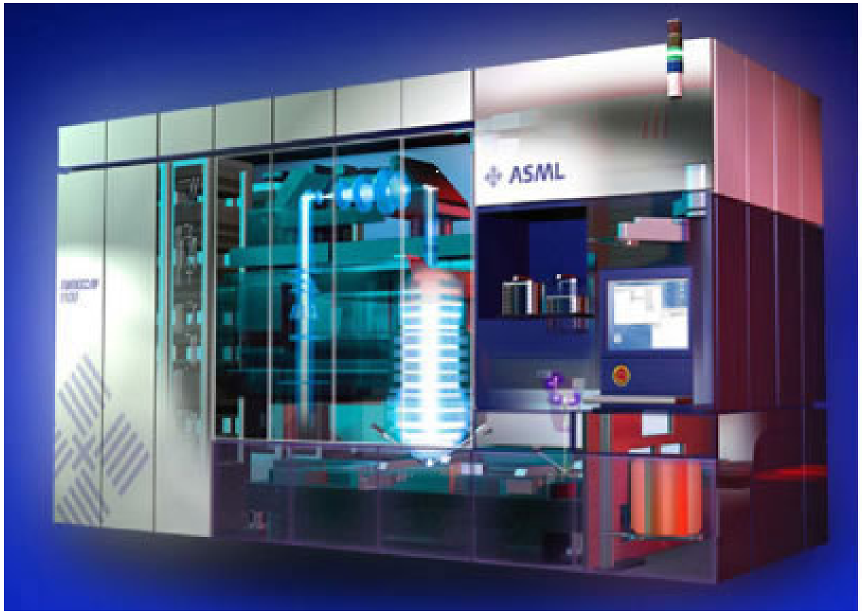
The Netherlands

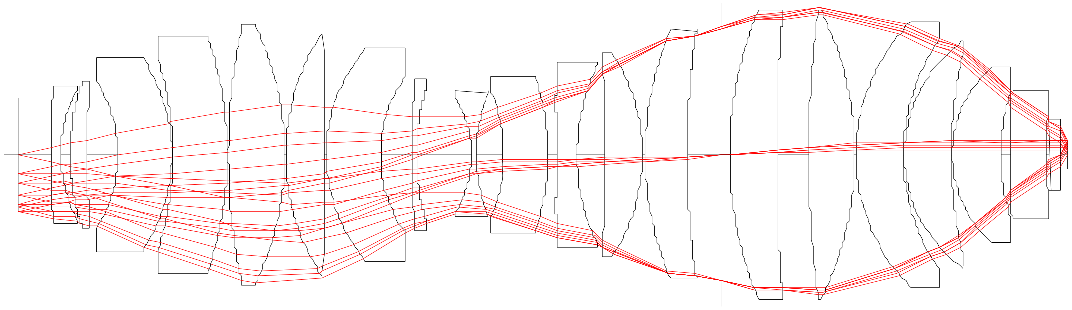
TU Delft

Delft University of Technology

1. Introduction

wafer scanner & optical projection objective





How are such complex systems designed?

- mathematical model -> geometrical optics

- Direct problem

  -> imaging quality as function of the constructional parameters of the system

- optical design = inverse problem
  - Given the required image quality
    - zero image defects within the specified tolerances
  - find a set of system parameters that leads to the required quality

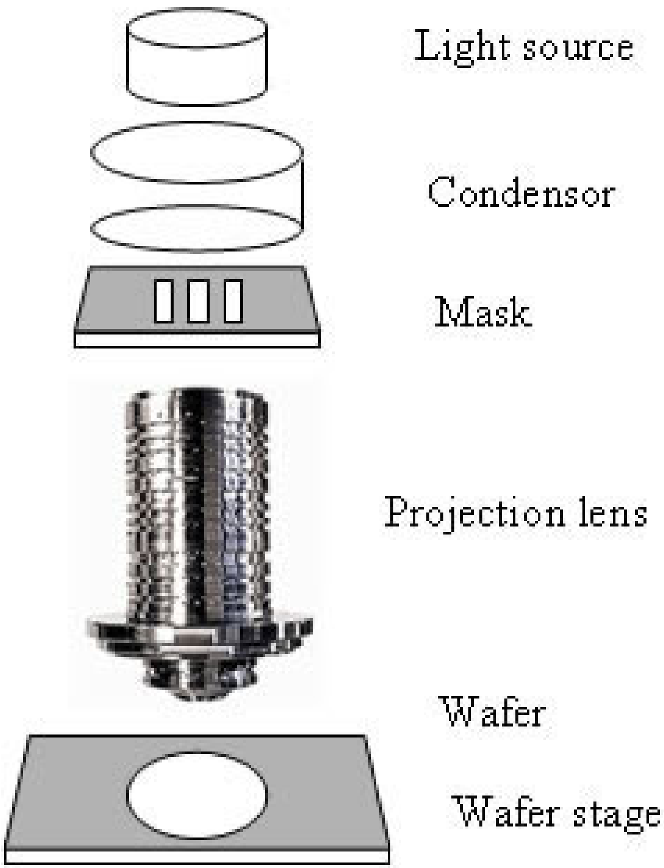
- In optical system design
    - No "receipt" for how to design a good system
    - Design process = "partly a science, partly an art“
- Essential design tool: **optimization** of the model on the computer

- In this talk: emphasis on the general principles of optical systems optimization
- For lithographic objectives
  - basic optimization techniques = the same as for simpler systems
  - But: because of high complexity, all difficulties appear in an extreme form!

Outline

1. Introduction
2. Reason for complexity of lithographic objectives
3. Basic ideas of optimization
4. Simple example of optimization: triplet design
5. Minimizing the error function
6. Constraints
7. Damped least-squares algorithm for local optimization

2. Reason for complexity of lithographic objectives



- Operation similar to that of a slide projector
- Circuit patterns → on the mask
- → imaged on a silicon wafer

Extremely high rate of information transfer: ~ 300 DVD's per second

- 24 seconds per wafer with 30 cm diameter (ASML)
- Smallest features ~ 45 nm

Requirements for lithographic objectives

- Resolution = $\mathbf{k_1\ \lambda/NA}$ for printing small features:
  - Low if
  - **numerical aperture NA is high** (DUV: 193nm)
  - λ is low (Next generation EUV: λ=13nm)
  - Low k_1 only for **high imaging quality** (low RMS Wavefront)

- Large number of wafers per hour
  → **field sufficiently large** (max image height)
- **Telecentricity** on both sides
- **Distortion** kept very low

- Difficult set of requirements → many lenses are needed
  - Number of possible design shapes is very large
  - There is always the danger to miss good design solutions

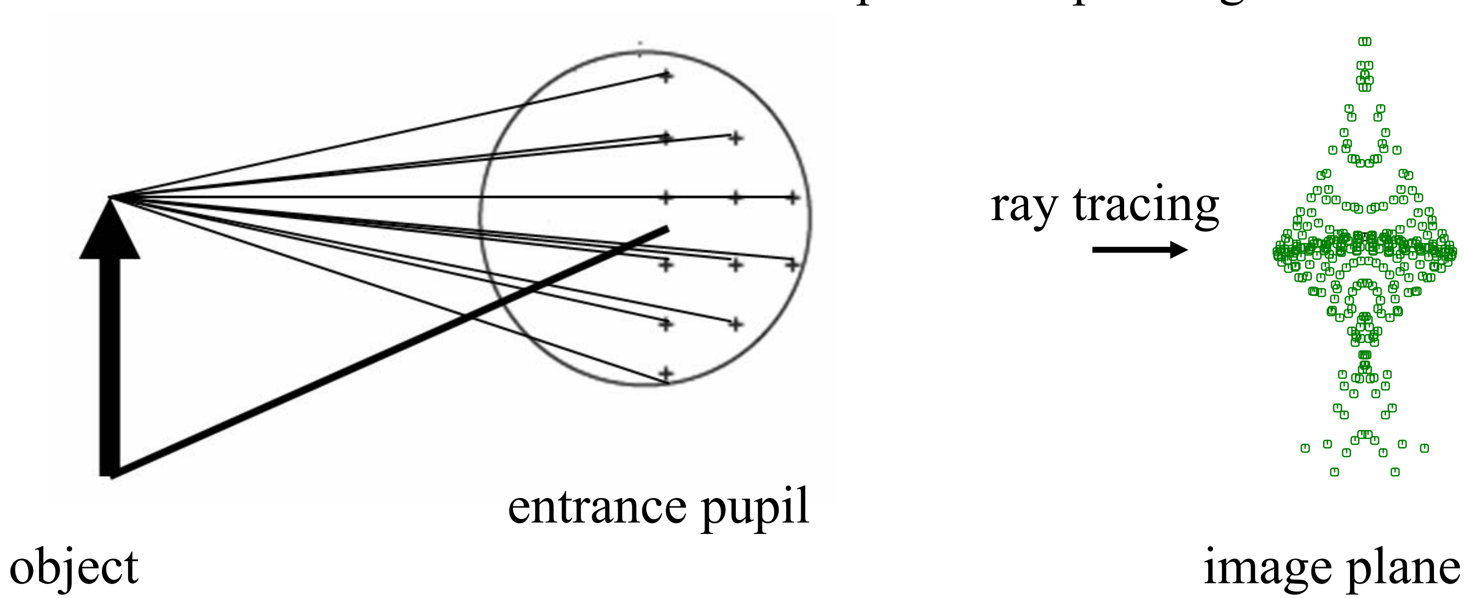
3. Basic ideas of optimization

- Initial configuration
  - does not yet fulfil the design requirements
- The software changes automatically the system parameters
  in order to make the system as good as possible

- Simple example of a measure of imaging quality = spot size
  - ideal imaging: image of a point is a point
  - real imaging: because of aberrations -> spot

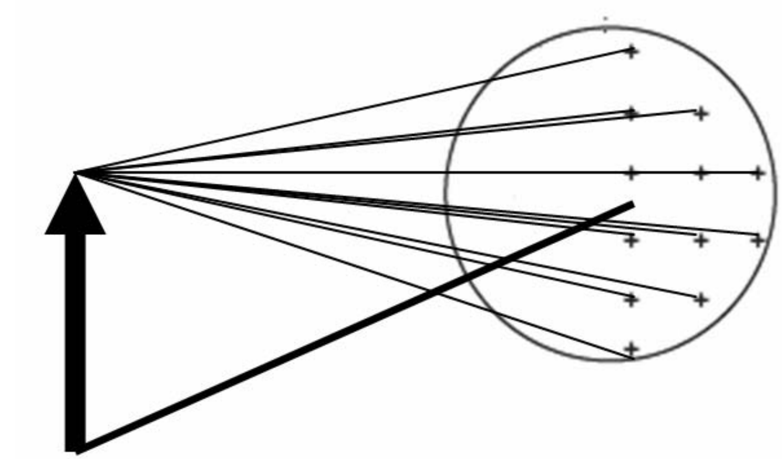
Spot diagram

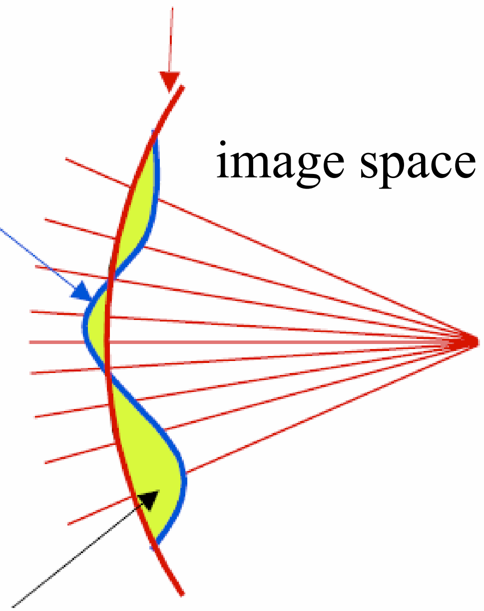
- Grid at the aperture stop (or at the entrance pupil)
    - Trace the rays from the object that pass through all grid points
- At the image plane
    - Collection of intersection points = Spot diagram



Wavefront aberration

- ideal imaging: wavefront in image space is spherical
- **Wavefront aberration=** wavefront deviation from spherical shape
- along all rays, compute the the optical path length between
  - ideal (reference) sphere
  - actual (aberrated) wavefront





Wavefront aberration

Lithography: acceptable wavefront aberration = a few mλ

For optimization the designer must specify:

i) a starting system configuration

ii) a merit function (or error function)

- a function of the system parameters
- at each stage -> measures the quality of the system

e.g. RMS spot size, RMS wavefront aberration

iii) the optimization variables

- a subset of system parameters
- are automatically changed during optimization

iv) a set of constraints

- limit the variation domain of the optimization variables
    - such that certain specifications are met

        e.g. focal length

    - the system must remain manufacturable

Optimization problem with *N* variables

Mathematically: each possible optical system configuration = point in a *N*-dimensional space.

Optimization -> implemented as a numerical algorithm

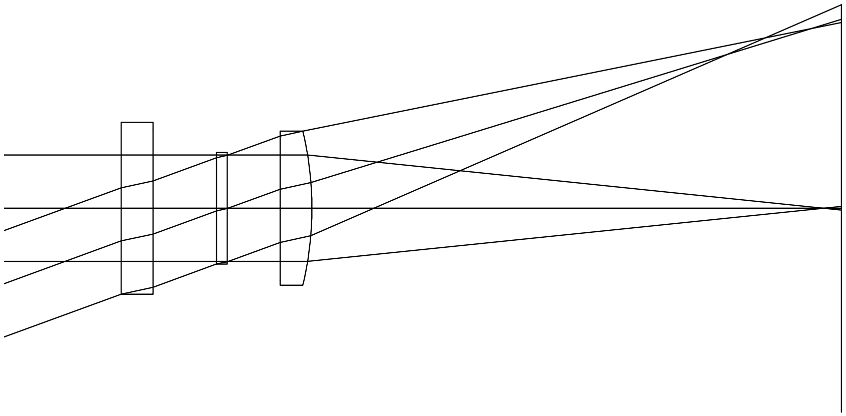
- attempts to find iteratively the (constrained) minimum of the error function

- starting from the initial configuration.

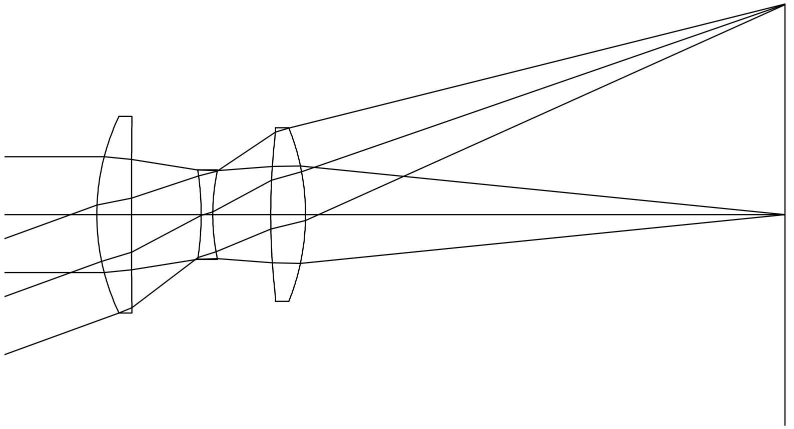
Optical system description

- number of surfaces (depends on the expected image quality)
- surface data
  - radius of curvature
  - distance to the next surface (called "surface thickness")
  - type of material after the surface (glass type, air, mirror)
  - other data if necessary (such as aspheric or gradient index coefficients)
- whole system specification data
  - aperture (entrance pupil diameter, numerical aperture, f/number)
  - field (object height, image height, field angle)
  - wavelength range
- position of the aperture stop

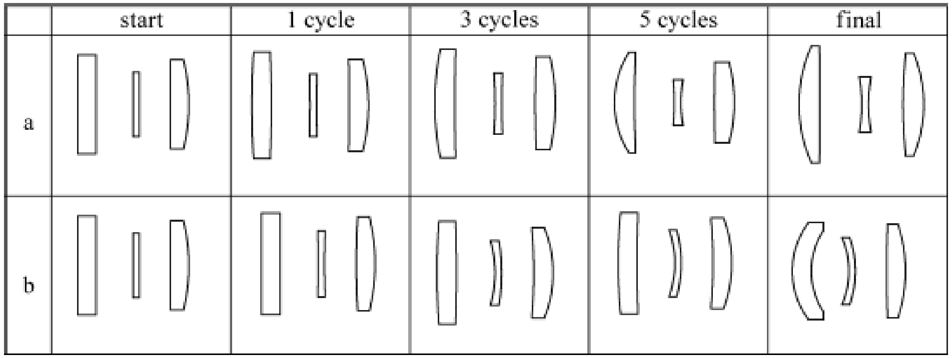
4 Simple example of optimization: triplet design



Starting point for the optimization of a three-lens system



Optimized three-lens system (Cooke triplet)



The evolution of the surface shapes of a triplet during optimization
Two different optimization strategies lead to different solutions

5. Minimizing the error function

$\mathbf{x} = (x_1, x_2,\ldots, x_N)$

$$f(\mathbf{x}) = \frac{\sum w_i \left(a_i(\mathbf{x}) - \tilde{a}_i\right)^2}{\sum w_i}$$

a_i: operands

- characteristics of the specific design task
- have target values $\tilde{a}_i$ and weights w_i
- examples:
  - transverse ray aberrations of individual rays
  - wavefront aberrations of individual rays
  - aberrations (e.g. distortion)
  - ray angles (e.g. for enforcing telecentricity)

$$f\left(\mathbf{x}\right)=\frac{\sum w_i\left(a_i\left(\mathbf{x}\right)-\tilde{a}_i\right)^2}{\sum w_i}$$

Obviously, in an ideal situation when all operands are equal to their targets then $f=0$

Why is an error function needed?

Because are there are typically much more operands than optimization variables

- a large number of rays must be traced

-> it cannot be expected that all operands reach their target values exactly

- only in a least-square sense

Local & Global Optimization

local optimization (LO)

- starting configuration
- LO= finding a minimum without ever increasing *f*
- the algorithm reduces *f* by changing the vector **x**
  - in several steps
- until the solution arrives at (or comes close enough to) a minimum of *f*
  - small changes of the optimization variables can only lead to an increase of *f* (i.e. there is no descent direction for **x**)
  - If there are no constraints, the gradient of *f* vanishes there

$$\nabla f\left(\mathbf{x}\right) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_N}\right) = 0$$

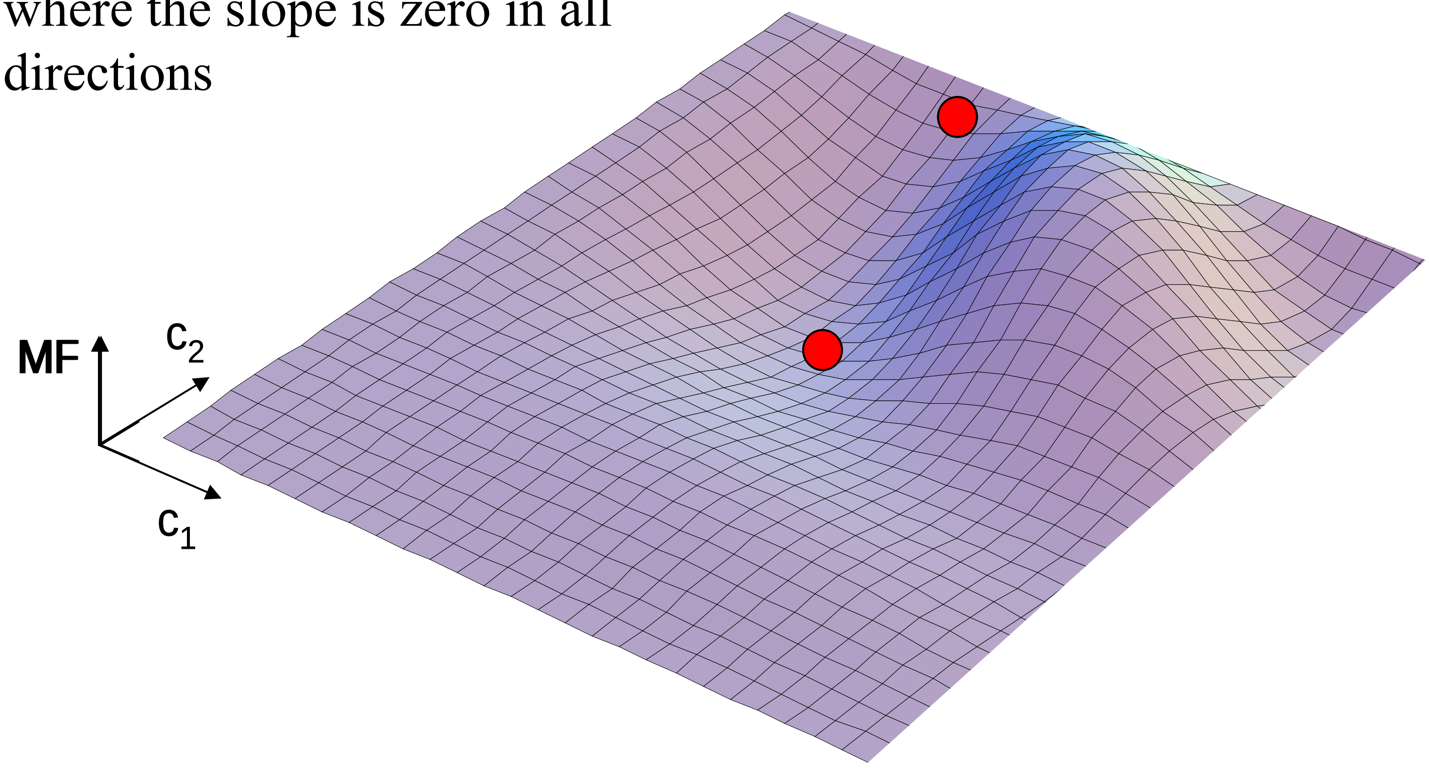
Two variables: *f* visualized as the elevation of a landscape

starting point = a ball placed at a specific location

optimization : the ball rolling downhill until it reaches a point where the slope is zero in all directions

There are many local minima in the optical design landscape!

global optimization = attempt to find the best among the local minima



<u>the steepest descent method</u>

- one of the simplest LO methods
- more powerful method discussed later, but SDM illustrates the essence of LO

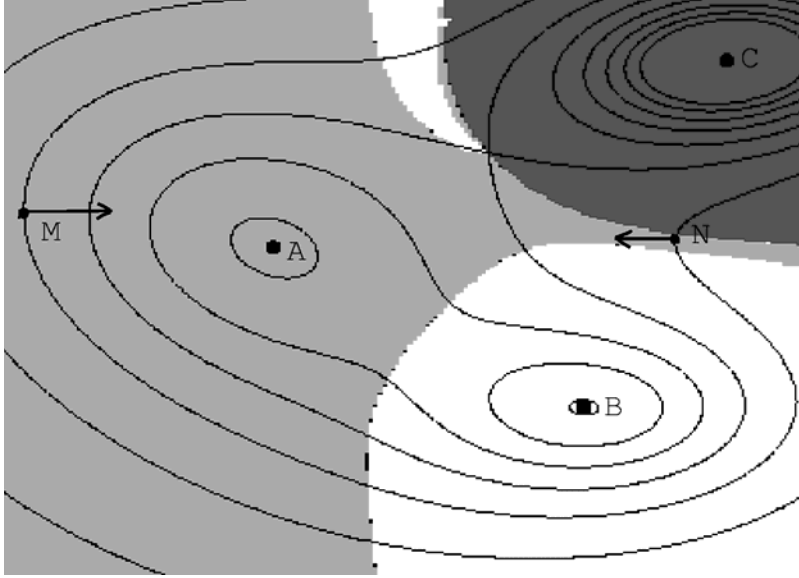


Figure: error function landscape with three local minima, A, B and C

- C = global minimum
  - equimagnitude contours

    = contours along which the error function has a constant value

- At each point : direction of <u>steepest descent</u> is that of $-\nabla f$
  - always perpendicular to the equimagnitude contours

    (the arrows in points M and N)

The **<u>steepest descent method</u>** can be summarized as follows:

i) at the starting point determine numerically the direction of $-\nabla f$

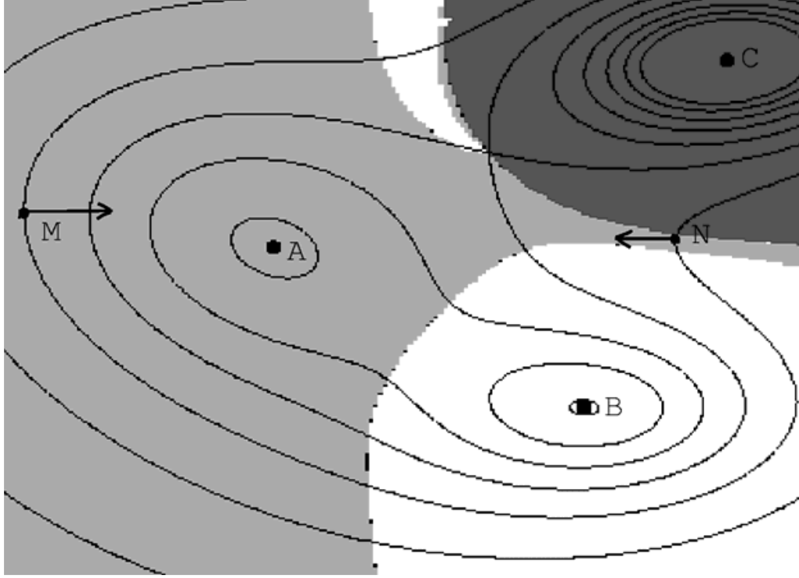
ii) along this direction determine the minimum of f

iii) make the minimum found in this way the new "starting point"

iv) repeat i), ii), iii)

- until the slope along the direction of steepest descent is close enough to zero

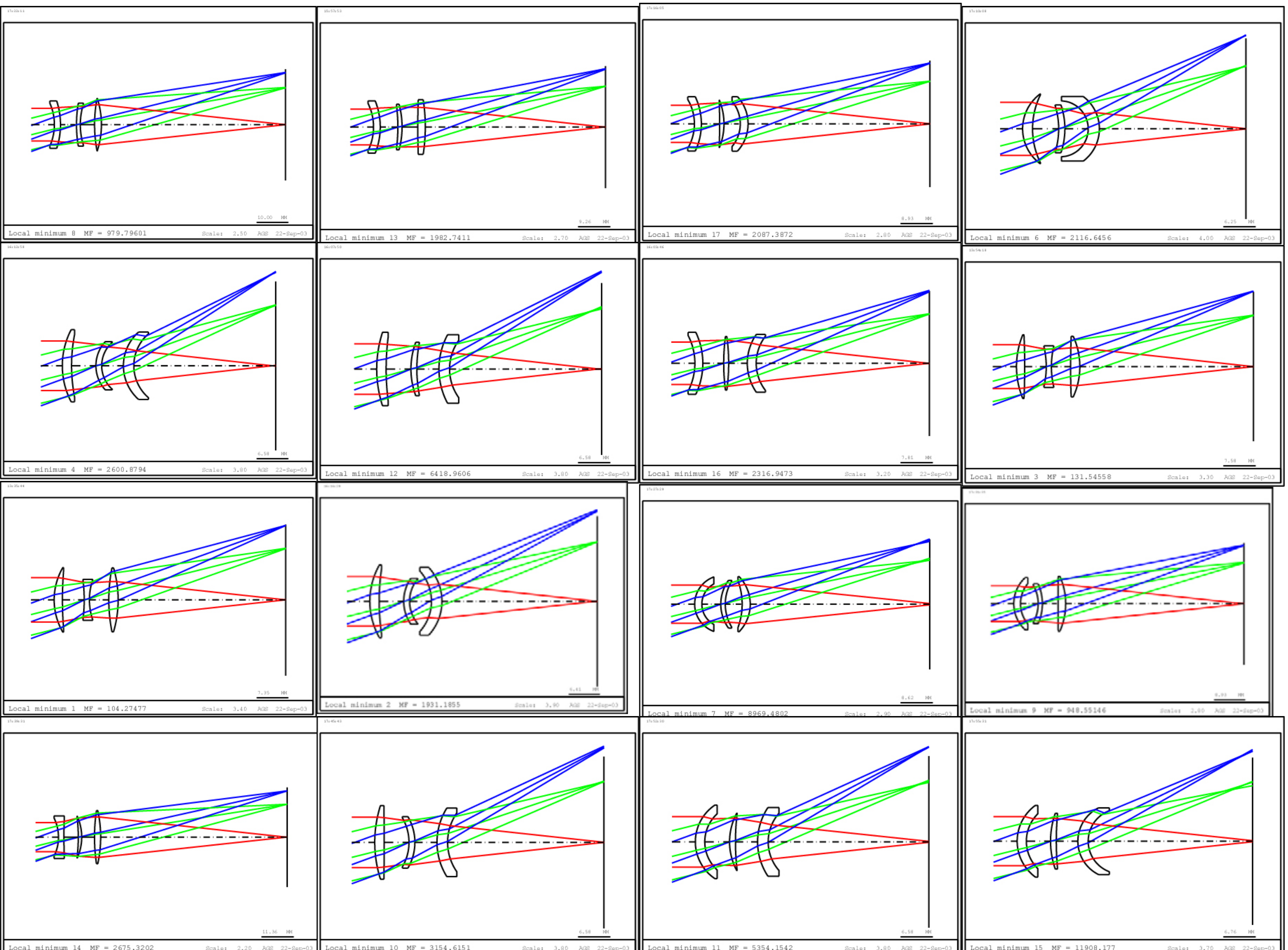
Regions	Local minimum
grey	A
white	B
black	C



- Key feature of LO in lens design:
  f is highly nonlinear -> many local minima
  -> result of LO is critically dependent on the choice of the starting point!

- Finding a good starting point for LO is the **most difficult** task in Lens design!
- Traditionally done on the basis of
  - previous experience, (e.g. patents or lens databases)
  - first-order layouts corrected for primary aberrations
  - intuition
  - often a substantial amount of trial and error
- Recent progress in global optimization alleviates this difficulty

Local minima of the 5D-triplet global search (16 out of 17)



6. Constraints

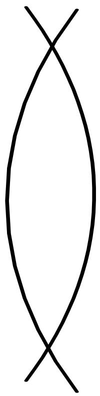
Minimize $f(\mathbf{x})$ subject to the conditions

$c_i(\mathbf{x}) = p_i \quad i=1,2,\ldots,m < N$ (equality constraints)

$d_j(\mathbf{x}) \geq q_j \quad j=1,2,\ldots,n$ (inequality constraints)

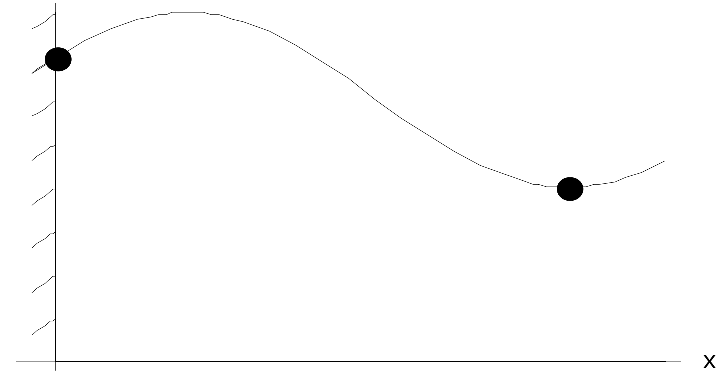
Examples:

- equality constraint
  - keeping the focal length constant during optimization
- inequality constraints
  - optimization variables are allowed to vary only within a given range
  - preventing the edge thickness of a biconvex lens to become negative when the surface curvatures tend to increase too much



When variables are restricted to a given interval
-> two types of minima are possible

i) minima where the tangent to the curve is horizontal (right dot)
ii) endpoints of the interval (left dot)



Only x > 0 is allowed
e.g. an axial air or glass thickness

- active constraints:
    - when a minimum is on the boundary of the allowed region (e.g. left dot)
    - one or more inequality constraints $d_j(\mathbf{x}) \geq q_j$ turn into equality constraints

$$d_j(\mathbf{x}) = q_j$$

-inactive constraints
  - If the set of active constraints is known, the remaining ones are inactive
  - can be ignored in the close neighborhood the given local minimum
    - have no influence on the optimization (e.g. x > 0 for the right dot)

-Locally, the situation becomes then an equality constraint situation

methods for enforcing equality constraints

1. "solves"
   - Certain lens parameters become "solutions" of equations
   - are eliminated from the list of variables
     - e.g. the last curvature, or the last air space in the system
   - when possible -> best option

$$f(\mathbf{x}) = \frac{\sum w_i \left(a_i(\mathbf{x}) - \tilde{a}_i\right)^2}{\sum w_i}$$

2. Adding "penalty" terms in f
   - The constraints $c_i(\mathbf{x}) = p_i$ become operands
   - optimization drives them towards their target values
   - constraints are not rigorously enforced
     - increasing w_i -> more rigorous enforcement

$$a_i = c_i$$
$$\tilde{a}_i = p_i$$

3. Lagrange multiplier method

- gives a better control over the constraints

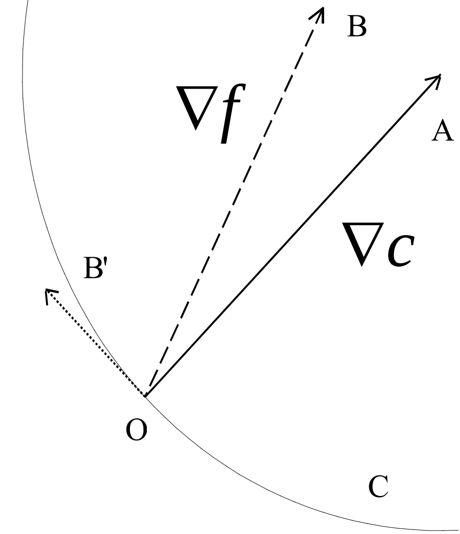
- 2D case:

constrained minimum = no descent direction *along C*

-> component of ∇f along the tangent must vanish

OB’= 0

-> gradients of f and c are parallel



$c(x_1, x_2) = p$

2D case, one constraint C

Condition for **constrained** minimum

$$\nabla f = \lambda \nabla c$$

- general case, m independent constraint functions :

$$\nabla f = \lambda_1 \nabla c_1 + \lambda_2 \nabla c_2 + \ldots + \lambda_m \nabla c_m$$

The Lagrange function transforms the problem into an unconstrained one

$$f_L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \lambda_1 c_1(\mathbf{x}) + \lambda_2 c_2(\mathbf{x}) + \ldots + \lambda_m c_m(\mathbf{x})$$

$x_1, x_2, \ldots, x_N$ and the Lagrange multipliers $\lambda_1, \lambda_2, \ldots \lambda_m$ must be determined

m constraint equations + annulling the components of ∇f_L -> $N+m$ equations

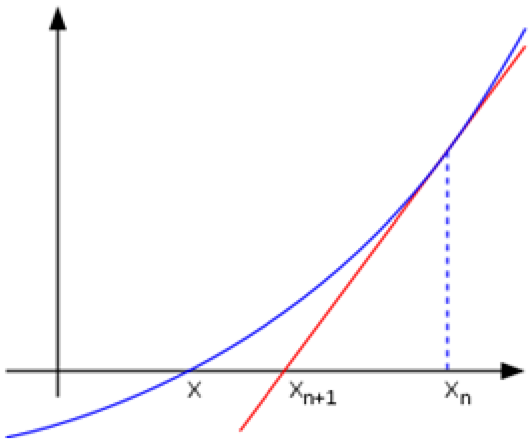
7. Damped least-squares algorithm for local optimization

Simple 1D problem: find the root *x* of a function, *f(x) =0*

- close to an initial guess x_0

Newton's method : iterative method

- finds successively better approximations to the root
- the function is approximated by its tangent line



$$f\left(x_0+\Delta x\right)\approx f\left(x_0\right)+f'\left(x_0\right)\Delta x \quad , \Delta x = x - x_0$$

$$f\left(x_0+\Delta x\right)=0 \rightarrow solve \;\; f'\left(x_0\right)\Delta x = -f\left(x_0\right)$$

- replace x_0 by the solution for x
- repeat the process until it converges to the root

after n iterations:

$$\Delta x = x_{n+1} - x_n = -\left(f'\left(x_n\right)\right)^{-1} f\left(x_n\right)$$

To find a minimum of *f(x)* , *f '(x)=0*
-> apply Newton's method to the derivative

$$f''\left(x_0\right)\Delta x = -f'\left(x_0\right)$$

$$\Delta x = x_{n+1} - x_n = -\left(f''\left(x_n\right)\right)^{-1} f'\left(x_n\right)$$

To find a minimum of the error function *f(**x**)* with N variables

- “prime”->”nabla”
- solve

$$\nabla^2 f\left(\mathbf{x_0}\right)\mathbf{\Delta x} = -\nabla f\left(\mathbf{x_0}\right) , \; \mathbf{\Delta x = x - x_0}$$

$$\mathbf{\Delta x = x_{n+1} - x_n} = -\left(\nabla^2 f\left(\mathbf{x_n}\right)\right)^{-1} \nabla f\left(\mathbf{x_n}\right)$$

1. A linear approximation for $\nabla f(\mathbf{x})$ is valid when the resulting $\mathbf{\Delta x}$ is small

To enforce a small $\mathbf{\Delta x}$, add at each iteration a "penalty term" to $f(\mathbf{x})$ that will "discourage" large values of $\mathbf{\Delta x}$ $p = damping\ factor$

$$f_D(\mathbf{x}) = f(\mathbf{x}) + p\Delta\mathbf{x}^T.\Delta\mathbf{x}$$

When the solution has converged to the minium, $\Delta\mathbf{x} = 0$ and $f_D(\mathbf{x}) = f(\mathbf{x})$

The solution is the same and we can minimize $f_D(\mathbf{x})$ instead of $f(\mathbf{x})$

2. The numerical evaluation of 2nd order derivatives in

$$\mathbf{\Delta x = x_{n+1} - x_n} = -\left(\nabla^2 f(\mathbf{x_n})\right)^{-1} \nabla f(\mathbf{x_n})$$

is computatinally expensive!

Goal: use the special (least-square) structure of $f(\mathbf{x})$ in order to simplify the computation of $\nabla^2 f$

- The error function with m operands

$$f(\mathbf{x}) = \frac{\sum_{j=1}^{m} w_i \left(a_i(\mathbf{x}) - \tilde{a}_i\right)^2}{\sum_{j=1}^{m} w_i}$$

- Rewrite $f(\mathbf{x})$ as

$$f(\mathbf{x}) = \sum_{j=1}^{m} A_j(\mathbf{x})^2 = \mathbf{A}^T(\mathbf{x}).\mathbf{A}(\mathbf{x})$$

$$\mathbf{A}(\mathbf{x}) = \left(A_1(\mathbf{x}), A_2(\mathbf{x}), \ldots, A_m(\mathbf{x})\right)^T$$

$$A_i(\mathbf{x}) = \sqrt{\frac{w_i}{T}}\left(a_i(\mathbf{x}) - \tilde{a}_i\right),\ T = \sum w_i$$

$\mathbf{A}(\mathbf{x}) -\!>$ residual vector

- Compute $\nabla f_D(\mathbf{x_0})$ and $\nabla^2 f_D(\mathbf{x_0})$ in $\nabla^2 f_D(\mathbf{x_0})\mathbf{\Delta x} = -\nabla f_D(\mathbf{x_0})$

$$\nabla f_D(\mathbf{x}) = 2\sum_{j=1}^{m} A_j(\mathbf{x})\nabla A_j(\mathbf{x}) + 2p\Delta\mathbf{x} = 2\mathbf{J}(\mathbf{x})^T\mathbf{A}(\mathbf{x}) + 2p\Delta\mathbf{x}$$

$$\mathbf{J}(\mathbf{x}) = \mathit{Jacobian\,matrix}$$

$$J_{ij}(\mathbf{x}) = \frac{\partial A_i(\mathbf{x})}{\partial x_j}$$

$$\nabla f_D(\mathbf{x_0}) = 2\mathbf{J}(\mathbf{x_0})^T\mathbf{A}(\mathbf{x_0}) \qquad \nabla f_D(\mathbf{x_0}) = \nabla f(\mathbf{x_0})$$

$$\nabla^2 f_D(\mathbf{x}) = 2\sum_{j=1}^{m}\nabla A_j(\mathbf{x})\nabla A_j(\mathbf{x})^T + 2\sum_{j=1}^{m} A_j(\mathbf{x})\nabla^2 A_j(\mathbf{x}) + 2p\mathbf{I}$$

$$\nabla^2 f_D(\mathbf{x_0}) = 2\mathbf{J}(\mathbf{x_0})^T\mathbf{J}(\mathbf{x_0}) + 2\sum_{j=1}^{m} A_j(\mathbf{x_0})\nabla^2 A_j(\mathbf{x_0}) + 2p\mathbf{I}$$

Levenberg Marquardt method: neglect the term $\sum_{j=1}^{m} A_j(\mathbf{x_0})\nabla^2 A_j(\mathbf{x_0})$

only the computation of $\mathbf{J}$ is then necessary at each iteration

$$\nabla^2 f_D(\mathbf{x_0}) = 2\mathbf{J}(\mathbf{x_0})^T\mathbf{J}(\mathbf{x_0}) + 2p\mathbf{I}$$

Denote $\mathbf{J}(\mathbf{x_0}) = \mathbf{J};\ \mathbf{A}(\mathbf{x_0}) = \mathbf{A}$

Solution of $\nabla^2 f_D(\mathbf{x_0})\mathbf{\Delta x} = -\nabla f_D(\mathbf{x_0})$

$$\Delta\mathbf{x} = -\left(\mathbf{J}^T\mathbf{J} + p\mathbf{I}\right)^{-1}\mathbf{J}^T\mathbf{A}$$

Observation: If damping p is large

$$\nabla^2 f_D(\mathbf{x_0}) = 2p\mathbf{I}$$

(other terms $\nabla^2 f_D(\mathbf{x_0})$ in are negligible)

$$\mathbf{\Delta x = x_{n+1} - x_n} = -\nabla f(\mathbf{x_n})/2p$$

$->$ direction of steepest descent

Final remark

The basic principles discussed here are applicable in many other optimization problems with continuous variables

Supplementary Reading

1. F. Bociort, *Optical System Optimization*, Encyclopedia of Optical Engineering, 1843-1850, Marcel Dekker, New York (2003)
http://wwwoptica.tn.tudelft.nl/users/bociort/optsysopt_ency.pdf

2. J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, (1999), mainly Chapter 10

3. For our work at TU Delft on optimization see our web pages
- *Networks, local minima and saddle points in optical system optimization*
http://wwwoptica.tn.tudelft.nl/users/bociort/networks.html

- *Fractals and chaos in optical system optimization*
http://wwwoptica.tn.tudelft.nl/users/bociort/fractals.html