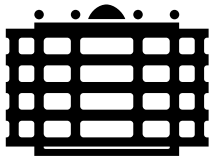
Optimal Layer Design

Jens Eller*
Computational Science
Chemnitz University of Technology

Christoph Sohrmann†
Computational Science
Chemnitz University of Technology

under joint supervision of
Prof. Dr. Karl Heinz Hoffmann (Computational Physics)
Prof. Dr. Frank Richter (Solid State Physics)

September 12th, 2003



CHEMNITZ UNIVERSITY
OF TECHNOLOGY

*jens.eller(at)s2000.tu-chemnitz.de
†c.sohrmann(at)warwick.ac.uk

Acknowledgement

We wish to express our sincere appreciation to all people who contributed to this thesis in one way or another. Particular gratitude for many helpful discussions and comments goes to the following people, without whom this work would not have been possible: Frank Heilmann, Dr. Thomas Chudoba, Prof. Dr. Karl Heinz Hoffmann, Prof. Dr. Frank Richter, Dr. Norbert Schwarzer, the computer center staff, Bianca Schott, Torsten Thau, and, last but not least, Birgit Weigold.

Furthermore we especially want to thank our families and friends for supporting us. Jens Eller is indebted to Solveig Hösel for her endless patience and love.

Contents

List of Figures

List of Tables

Material	Young's modulus E / GPa	Poisson's ratio η / GPa	Yield strength σ_M^{crit} / GPa	Critical tensile stength σ_{xx}^{crit} / GPa
Steel1	200	0.3	3	2
Steel2	215	0.3	3	2
Nickel	200	0.3	1.0	4
TiN1	400	0.25	13	3
TiN2	450	0.25	13	3
GaAs	118	0.24	12.3	0.8
DLC1	150	0.25	10	2
Si_3N_4	300	0.25	11.6	1.5
CrN	320	0.25	11	2
Cr	190	0.3	2	5
Al_2O_3	160	0.25	6	1
cBN	800	0.13	20	8
Al	70	0.3	0.4	2
Ni	210	0.3	1.5	4
Diamond	1100	0.3	-	-

Table 1: Materials used

σ_M^{max}	Maximum von Mises stress
σ_{xx}^{max}	Maximum tensile stress
σ_M^{crit}	Yield strength (critical von Mises stress)
σ_{xx}^{crit}	Critical tensile stress
$\mathbb{L}^i$	Spatial domain of layer i
R_{con}	Contact radius
Ω	State space of coatings
$\hat{\mathbb{R}}$	Range of the thicknesses
$\vec{\omega}$	State within the state space
Θ	Heavysite function
f	Objective function
f_{opt}	Optimal objective function value
$\vec{\sigma}$	Stress state
E	Young's modulus
ν	Poisson's ratio
d_{opt}	Optimal thickness

Table 2: Notations

1 Introduction

In dynamic systems mechanical contacts are often formed with or without purpose. Since these parts are subject to wearout, it is the engineer's task to protect the surface providing sufficient reliability and durability.
In the last decades it turned out that thin layered coatings are able to provide a convenient solution. By using layered thin film coatings with suitable properties one can increase lifetime and resistance to wearout of critical components significantly. Thus, engineers aim at designing the layer structure in an optimal way satisfying the growing needs of contemporary technical applications. With regard to the expenses of experimental approaches, computer simulations are highly prefered. Material is spared and results are obtained within comparably short time.

The goal of this work is to generate an appropriate strategy for finding optimal layer designs with respect to mechanical failure of the compound. To investigate mechanical durablity, indentation experiments used for reproducing real load situations, have proved adequate. Based upon the commercial simulation package ELASTICA [1], indentation experiments can be investigated theoretically and studied under aspects of internal stress states. Since mechanical failure is indicated by internal stresses exceeding their critical limits, stress maximums within the layers of the compound have to be determined. Several algorithms will be studied for their eligibility.

The durability and reliability of the compound can then be assessed by comparing the maximum stresses to their critical counterparts. Using this relation, a measure for optimality will be stated. For a particular load situation and substrate, materials and thicknesses of the single layers can be optimized with respect to this measure. Again, different ways using established optimization methods will be analyzed. Moreover, an esituation stimation of the quality of a compound will be presented, making predictions about how to combine the materials properly.

To gain insight into the structure of the distinct problems, as well as to obtain reference solutions, a huge number of compounds are necessarily to be reviewed. Due to the remarkably time-consuming evaluation of the stress fields, a parallel approach will be of vital importance for gaining results within the available time. Therefore, a facile tool for clustering arbitrary computers of a network has to be developed.

For some basic understanding, a short introduction into properties of thin films is given in chapter 2. The protective effect is pointed out and the modelling of contact problems is outlined.

Chapter 3 explains the fundamentals of optimization along general lines, and introduces the algorithms used within this work.

The main part is covered in chapter 4. Reviewing a vast number of coating systems, a strategy for finding the maximum stresses within the material domains is deduced from numerical investigations. Using the maximum stresses, the compounds are assessed with respect to their quality and a way for finding optimal thicknesses and

materials is developed.

Since the parallel calculation takes an utterly important part of this thesis, chapter 5 presents an overview about parallel concepts and classifications.

Chapter 6 presents an overview over the structure of source code needed during this work. Finally, the results are summarized within chapter 7.

2 Thin Films

In this preluding section, necessary steps for describing contact problems will be explained. The protecting effects of load carrying coatings are presented, and how material properties of thin films are experimentally determined. Realistic simulations of contact problems demand precise knowledge of the properties of the materials involved and the arising stress fields within the compound. Failure mechanisms and their theoretical analyzation will be introduced.

2.1 Protective Thin Film Coatings

Thin film coatings are used under several intentions as functional components. They serve as optical refinements, electrical contacts in switches and plugs, or to reduce friction. With suitably chosen properties, they are capable of protecting mechanical components from destruction. A protective coating is able to avoid the intrusion of high stresses into the depth to be protected. With properly chosen layer materials stress maximums can be 'displaced' into load carrying protection layers. Unfortunately, the protecting effect cannot be achieved in every case (see, e.g., [2]).
Figure 1 shows the effect of a 3 μm thick TiN coating on a steel substrate (material properties see, table 1).

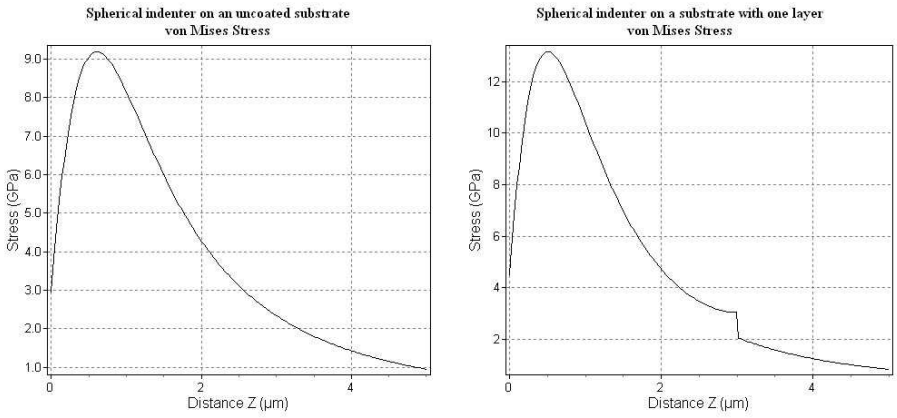


Figure 1: Protecting effect of a TiN coating: von Mises stress along the depth axis (Z axis) of uncoated (left) and 3 μm thick TiN coated steel (right) under a normal load of 50mN, and a 10μm spherical indenter.

The contact situation was chosen to be 10 μm spherical indenter pushing into the compound with a force of 50 mN. Steel has a critical value of 3 GPa, which is extensively exceeded without a coating. By using a TiN coating, the maximum von Mises stress is lowered to 2 GPa and remains below the critical limit. The TiN layer itself has a critical value of about 13 GPa. Hence, the substrate is protected by the thin film and will not be damaged.

2.2 Physical Properties

Thin films can be characterized in a multitude of ways. Most important properties are *Youngs's modulus, Poisson's ratio, yield strength, tensile strength, fracture thoughness, film adhesion, fatigue, friction coefficient, internal stresses and their temperature dependencies*. The basic properties which will be used later to simulate the elastic behavior, are Young's modulus and Poisson's ratio. For elastic, isotropic materials these two parameters are sufficient [3] for mathematical description.
Their determination still demands a lot of experience and technical effort, nowadays. One promising approach is *nanoindentation*. It is the most important source of satisfactory data. Though its accuracy is not completely satisfying, the achievable results are unprecedented. Indentation experiments are carried out by loading the object of interest and measuring the load-displacement curves. With the information obtained by the response of the material one can deduce different properties. Thereby, the influence of the substrate on the measuring process is of great importance. This problem can be solved by wholly elastic measurements with spherical indentation.

Young's Modulus. The load-displacement curve is measured both of the compound and the substrate. Afterwards, the curves are fitted to theoretically obtained data. The exact value of Young's modulus should then coincide with the value theoretically used. Due to material inhomogenties, results obtained by other approaches such as tensile tests or ultrasonic wave techniques often differs slightly.
Other important properties are more difficult to obtain. For the determination of the load carrying capacity, values for yield strength and tensile strength have to be found.

Yield Strength. The yield strength indicates the onset of plastic deformation and can be measured by cyclic indentation (loading and unloading) [4, 5] with continuously increasing indentation force. The load-displacement curves of the loading and unloading agree as long as the material stays within the elastic regime. The failure of the material starts when nonreversible deformation occurs. The onset can be found where the two curves are tearing apart. Knowing this critical indentation force, the yield strength can again be gained by comparision to theoretical data. For very thin layers, a compromise for the indenter radius has to be found. A small radius only allows a small force which is complicated to measure. On the other hand, a large radius imposes plastic deformation in the substrate which makes the measured values unusable since the substrate has to stay within the elastic regime.

Tensile Strength. For tensile strength, indentation experiments are not suitable since cracking and delamination does not clearly show such as plastic behaviour. The determination of internal breaks is carried out by a combination of bending tests and X-ray diffraction for instance.

2.3 The Stress Tensor

For understanding the behavior of a solid body, the internal stress states have to be examined. Stresses are defined as forces acting on planes within the body. For studying a stress state, we examine an arbitrary plane at a point P within the 3-dimensional body. This plane A with its plan normal $\vec{n}$ can be described as

$$\vec{dA} = \vec{n} \cdot dA = (dA_x, dA_y, dA_z)^T. \tag{1}$$

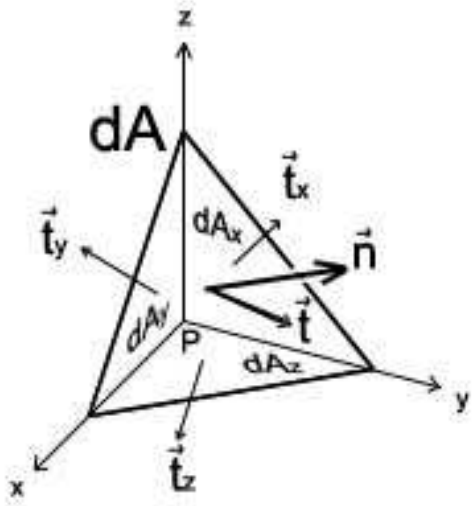


Figure 2: Illustration of the stress vectors.

In the static case, the sum of all forces acting on the point vanishes, otherwise it would start moving. The stress vectors, defined as the force per area, acting on the partial faces, have to compensate the force acting on the plane $\vec{dA}$

$$\begin{aligned} \vec{t} \cdot dA &= \vec{t_x} \cdot dA_x + \vec{t_y} \cdot dA_y + \vec{t_z} \cdot dA_z \\ &= \vec{t_x} \cdot n_x dA + \vec{t_y} \cdot n_y dA + \vec{t_z} \cdot n_z dA \end{aligned} \tag{2}$$

so

$$\begin{aligned} \vec{t} &= \vec{t_x} \cdot n_x + \vec{t_y} \cdot n_y + \vec{t_z} \cdot n_z \\ &= \mathcal{T} \cdot \vec{n}. \end{aligned} \tag{3}$$

These 3 vectors are combined to a tensor $\mathcal{T}$, called stresstensor

$$\mathcal{T} = (\vec{t_x}, \vec{t_y}, \vec{t_z}) = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix}.$$

The components of the stress tensor are the stresses due to forces on the 3 planes dA_x, dA_y und dA_z.

2.4 Failure of Compounds

For a reliable modelling and predicting of durability of layer compounds, knowledge of failure criteria is of paramount importance. The mechanism of wearout is twofold: On the one hand, *plastic deformation* can occur, and on the other hand *cracking and delamination* appears. Both mechanisms are caused by stresses exceeding their critical values and can be investigated both experimentally and theoretically.
Plastic deformation sets in whenever the *von Mises comparison stress* σ_M exceeds the *yield strength* σ_M^{crit}. Cracks will occur if the maximum tensile stress (here: corresponding to σ_{xx}) exceeds the value of *tensile strength* σ_{xx}^{crit}. The maximum tensile stress always appears within the σ_{xx} stress behind the indenter (regarding the force direction) since the force is supposed to act into the x-direction (see, [6]). Without lateral force, σ_{xx} and σ_{yy} are equal for spherical indentation due to the radial symmetry. The important issue is to avoid these failure mechanisms by the use of suitable protecting layers. The theoretical analysis uses the von Mises comparison stress defined as

$$\sigma_M = \sqrt{\frac{1}{2}\left[(\sigma_{xx}-\sigma_{yy})^2+(\sigma_{zz}-\sigma_{yy})^2+(\sigma_{xx}-\sigma_{zz})^2+6(\sigma_{xy}^2+\sigma_{xz}^2+\sigma_{zy}^2)\right]} \quad (4)$$

2.5 Modelling a Contact Problem

In 1966, Hamilton and Goodman [7] presented the first analytic solution for spherical Hertzian load and sliding friction. More convenient and usable solutions were attained by Hanson and Johnson in 1992 [8] and 1993 [9] using newly gained results of Fabrikant [10]. Layered systems could only be treated by Finite or Boundary Element methods, at first. In 2000, Schwarzer [11] published an entirely analytic solution for layered systems and Hertzian pressure distribution, for the first time, including sliding friction by utilizing the method of image charges of electrostatics. Finally, this theory was implemented in the commercial simulation software Elastica. It is based on an analytic approach and calculates the stress fields of layered systems by means of potential theory and the theory of image charges of electrostatics. For our investigations, we make use of ELASTICA as a Windows Dynamic Link Library (DLL). An outline on how this is done is presented in chapter 6.
For the modelling of a contact problem with ELASTICA, several parameters have to be specified:

- **Substrate:** Young's modulus and Poisson's ratio
- **Number of layers**
- **Layer materials:** Young's modulus, Poisson's ratio and thickness
- **Indenter:** radius, normal force, lateral force and friction coefficient
- **Calculation accuracy**

After the contact situation has been determined by the parameters mentioned above, stress fields and displacements can be investigated pointwise or via 1D, 2D or contour plot graphics.

3 Optimization Methods

This chapter gives an introduction to the fundamentals of optimization, the optimization methods and their basic ideas.
The following methods are mainly used within this work: Golden Section Search, Simplex, BFGS, and Gradient. They are all well known and often explained (see, e.g., [13, 14]). Therefore, only the basic ideas and important facts for this work are presented.

3.1 Fundamentals

The goal of optimization is to minimize a given function f that might depend on more than one independent variable. If the actual goal is to find the maximum of f, one has to minimize $-f$ instead.

$$f_{min} = \min f(\vec{x}) \quad \vec{x} \in \mathbb{R}^n, \tag{5}$$

where $\vec{x}$ is a vector containing the n independent variables x_i

$$\vec{x} = (x_1, x_2, \ldots, x_n)^T \tag{6}$$

and

$$f_{min} = f(\vec{x}^*). \tag{7}$$

A minimum can be *global* (truly the lowest function value) or *local* (the lowest function value in a finite neighborhood but not on the neighborhoods boundary).

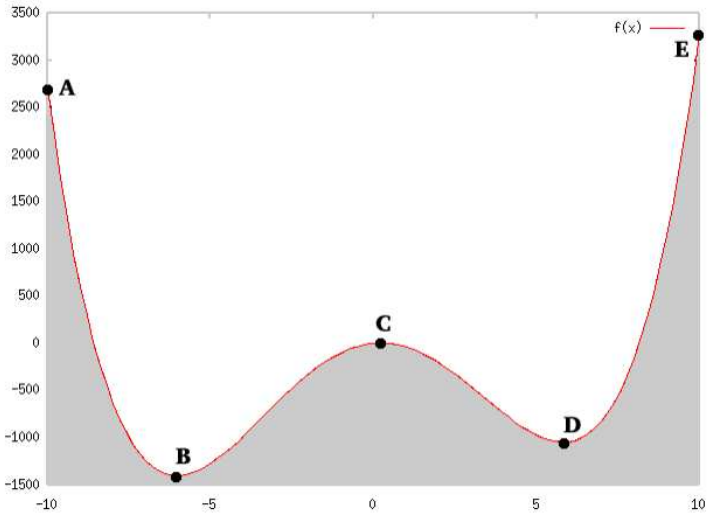


Figure 3: Minima and Maxima of a function in an interval. A, C and E are local maxima, where E is also *global* maximum. B and D are local minima, where B is *global* minimum.

Apparently, minimizing functions with only one local mininum is easier than the minimization of functions with more than one local minimum.

Often there are *a priori* known restrictions to the allowed values of the independent variables, then the optimization is called *constrained*.

3.2 Golden Section Search

The golden section search is an 1-dimensional method for bracketing a minimum $f_{min} = f(x^*)$ of a unimodal funtcion $f(x)$ in a given interval (a, b) by excluding parts of the interval.
By definition, a function $f(x)$ is called *unimodal* in $a \leq x \leq b$, if it has only one stationary point (either a minimum or a maximum) in (a, b).

The idea is to divide the initial interval $(a^{(0)}, b^{(0)})$ into 3 intervals $(a^{(i)}, x_1^{(i)})$, $(x_1^{(i)}, x_2^{(i)})$ and$(x_2^{(i)}, b^{(i)})$, where $i = 0$ in the first iteration, so that

$$\frac{x_2^{(i)} - a^{(i)}}{b^{(i)} - a^{(i)}} = \frac{x_1^{(i)} - a^{(i)}}{x_2^{(i)} - a^{(i)}} = \frac{b^{(i)} - x_2^{(i)}}{b^{(i)} - x_1^{(i)}} = \tau, \tag{8}$$

where $x_1^{(i)} < x_2^{(i)}$ and τ is constant.
Comparing $f(x_1^{(i)})$ and $f(x_2^{(i)})$, the interval with the bigger function value is excluded from the initial interval. If $f(x_1^{(i)}) < f(x_2^{(i)})$ then

$$a^{(i+1)} = a^{(i)}, b^{(i+1)} = x_2^{(i)} \text{ and } x_2^{(i+1)} = x_1^{(i)}$$

otherwise

$$a^{(i+1)} = x_1^{(i)}, b^{(i+1)} = b^{(i)} \text{ and } x_1^{(i+1)} = x_2^{(i)},$$

where $x_1^{(i)}$ or $x_2^{(i)}$ are chosen so that equation(8) is satisfied.
In the next iteration i is set to $i = i + 1$ and this is repeated until

$$b^{(i)} - a^{(i)} \leq \varepsilon,$$

where ε is the previously chosen accurracy of boxing in x^*. At least x^* is approximated by

$$x^* = \frac{a^{(i)} + b^{(i)}}{2}.$$

The initial conditions of equation(8) ensure, that after the initial iteration, only one function value $f(x)$ has to be calculated in each iteration.
For dividing the interval the best way, it can be shown [13] that τ has to be chosen as

$$\tau = \frac{\sqrt{5} - 1}{2} \approx 0.618, \tag{9}$$

what is called the *golden section* or *golden mean*.

3.3 Quasi Newton Methods

One of the most important search directions to locate a minimum is the *Newton direction* $\vec{n}$. It is derived from second-order Taylor series approximation

$$f(\vec{x}_k + \vec{p}) \approx f_k + \vec{p}^T \nabla f_k + \frac{1}{2}\vec{p}^T \nabla^2 f_k \vec{p} \stackrel{def}{=} m_k(\vec{p}), \tag{10}$$

where k is the index of the current iteration. Assuming that $\nabla^2 f_k$ is positive definite, the Newton direction, minimizing $m_k(\vec{p})$, can be obtained by setting the derivative of $m_k(\vec{p})$ to zero

$$\vec{n}_k = -\nabla^2 f_k^{-1} \nabla f_k. \tag{11}$$

The difficulties in numerical handling and calculation of $\nabla^2 f_k$ and $\nabla^2 f_k^{-1}$ (e.g. $\nabla^2 f_k$ may not allways be positive definite and so $\nabla^2 f_k^{-1}$ does not exist) were the intentions for the so called *Quasi-Newton* search methods. In these methods, the true Hessian $\nabla^2 f_k$ is approximated by B_k, using the informations about the second derivative of f provided by the changes of ∇f_k to update B_k. B_{k+1} has to satisfy the following condition, known as *secant equation* [14]:

$$B_{k+1}\vec{s}_k = \vec{y}_k,$$

where

$$\vec{s}_k = \vec{x}_{k+1} - \vec{x}_k, \qquad \vec{y}_k = \nabla f_{k+1} - \nabla f_k$$

One of the most popular formula to update the Haessian approximation B_k is the *BFGS formula*, named after its inventors, Broyden, Fletcher, Goldfarb, and Shanno, which is defined by

$$B_{k+1} = B_k - \frac{B_k \vec{s}_k \vec{s}_k^T B_k}{\vec{s}_k^T B_k \vec{s}_k} + \frac{\vec{y}_k \vec{y}_k^T}{\vec{y}_k^T \vec{s}_k}, \tag{12}$$

So the quasi-Newton direction is given by

$$\vec{n}_k = -B_k^{-1} \nabla f_k. \tag{13}$$

Using the direction of $\vec{n}_k$ in every iteration a line search has to be realized.

3.4 Conjugate Gradients

Another way of using derivative information is the method of *Conjugate Gradients*, which was originally developed for minimizing functions, which are supposed to be of quadratic form

$$f(\vec{x}) \;=\; \frac{1}{2}\vec{x} \cdot A \cdot \vec{x} - \vec{b} \cdot \vec{x}. \tag{14}$$

Nevertheless, the method of CG is also suitable for non-quadratic funtions because near the minimum almost every non-quadratic funtion behaves like a quadratically.
The starting position $\vec{x_o}$ is updated by moving into a direction $\vec{d}$ until a minimum is reached

$$\vec{x}_{i+1} \;=\; \vec{x}_i + \alpha_i \vec{d}_i, \tag{15}$$

where, in the first step, $\vec{d_0} = -\nabla f(\vec{x})$.
For quadratic functions, where A is known, α_i can be determined by a formula. In our case, α_i is unknown. It is found by a line minimization. After the first step into gradient's direction, later steps are into conjugate directions. Two directions are said to be conjugate, if

$$\vec{d_i} \cdot A \cdot \vec{d_j} \quad = \quad 0 \quad \text{for} \quad i \neq j \tag{16}$$

The new direction is found by Polak's formula

$$\vec{d}_{i+1} \quad = \quad -\nabla f(\vec{x}_{i+1}) + \frac{\nabla f(\vec{x}_{i+1}) \cdot (\nabla f(\vec{x}_{i+1}) - \nabla f(\vec{x}_i))}{\nabla f(\vec{x}_i) \cdot \nabla f(\vec{x}_i)} \vec{d}_i. \tag{17}$$

In the case of quadratic functions in n dimensions, every direction is taken only once and after n steps the optimum is reached. Because most functions, particularly in real applications, do not have quadratic form, the minimum will normally not be reached within n steps. The algorithm has to be restarted. Fortunately, the update formula of Polak tends to reconstruct the local gradient after n steps, which is equivalent to restarting.

3.5 Simplex Strategy

The simplex strategy provides an optimization method in multidimensional case without derivatives. It might not be confused with the simplex method of linear programming of Dantzig.

If n is the number of dimensions, then the convex hull of $n + 1$ points is called a simplex. For example, in the case of 2 dimensions a simplex is a triangle and in 3 dimensions a tetrahedron. For all vertices the objective function is evaluated. There are many variations of the simplex strategy and the most common one is that of Nelder and Mead [13]. In this case, starting at point $\vec{p}_0$, the vertices $\vec{p}_i$ are constructed by

$$\vec{p}_i = \vec{p}_0 + \lambda \vec{e}_i, \tag{18}$$

where $\vec{e}_i$ are the unit vectors and λ is determined by the problem's characteristic length scale.
In the used implementation, the vertex with the largest function value is reflected at the centroid of the other vertices. If the reflection was successful, an extention in the same direction is tried. If the reflection failed, it is checked, if another vertex has a bigger funtion value than the one of the newly created vertex. If this is true, the new vertex is taken, and the next iteration is started. Otherwise a contraction is performed on the reflection side (were the worst vertex is located) and also one on the opposite side. If the latter contraction was not successful, the simplex contracts in all directions. This forces the simplex climbing downhills. The algorithm is aborted, if the last step was smaller then a determined tolerance. Restarting the simplex at the found minimum is useful for better convergence.
The simplex method is costly but robust.

4 The Optimization Issue

This chapter marks the core of this work. It is outlined how a coating is assessed with regard to the expected durablity and reliability. Upon this basis, an optimum can be sought for. Thereby, the strategy of a seperate treatment of the optimal materials and optimal thicknesses is followed.

Remark: The material properties are mostly extracted from literature. Many critical tensile stresses are assumptions following similar materials, since literature lacks sufficient experimental data. The compounds emerging from randomly combinating the materials are regarded as examples and without the claim of depositability. They only serve as a space for investigations.

4.1 Basic Considerations

To accomplish an optimization among thin film coatings, a concrete contact problem has to be connected to an abstract mathematical method. The problem of finding the optimal coating is confined to the search for most suitable materials and degrees of thickness for a particular substrate and load condition. The load condition demands some reasoning about loads appearing within the prospective application. A representative load state should be chosen reflecting averaged or maximum load conditions. This clearly results in a compromise between covering as many contingencies as possible, and the capability of creating the coating. For setting up an adequate load state describing the application best, several parameters have to be specified properly. Substrate and layer materials have to be determined by their Young's modulus, Poisson's ration, yield strength and tensile strength. The maximum thickness of the layer compound should be fixed as well in order to ease the computation.

Now, a connection to a mathematical optimization model has to be drawn. Therefore, the coating is assessed with respect to the failure criteria mentioned above in section 2.4. This leads to a sub-optimization task. The arising stresses inside the layers and the substrate are analyzed as described within the next chapter. The maximum stress values for von Mises stress and tensile stress are retained for every material domain. By comparing these values to their critical limits yield strength and tensile strength, a measure of quality can be stated. From this point it is a mere mathematical task finding the very coating, matching the objectives best. Therefore, a seperated treatment of materials and their thicknesses is considered since they appear mathematically different: Discrete materials and continuous thicknesses.

The mathematical treatment of the optimization is based on a *state space* and an *objective function*. The state space contains the set of all depositable materials as specified by the user. A state represents a particular coating, including the information about the materials and thicknesses. For any state within the state space, section 4.2 illustrates how to proceed on finding the maximum stress values, refered to as sub-optimization task. The next section will describe the structure of the state space in a more comprehensive way.

4.2 Determining Stress Maximums

4.2.1 The Black Box DLL

To let later decissions become more clear, at first a summary of the unexpected behaviour of the stress data supplied by the ELASTICA DLL is presented.

At the beginning of this work it was known that ELASTICA could provide data with scatter and many obviously unreal peaks. It might appear when the difference of elastic properties of the materials is to big or the calculation accuracy is chosen wrong. Since ELASTICA calculates points equal distributed in a domain, it can use statistics to warn if the data seems to contain scatter of data. Because of the uncertainity about the distribution of the calcultated points in the used optimization methods, in this work no investigations to detect automaticly scatter in the data were made. One has to refelect if the calculated material combinations might tend to produce scatter(e.g. because of too different Young's modulus, see figure 4).

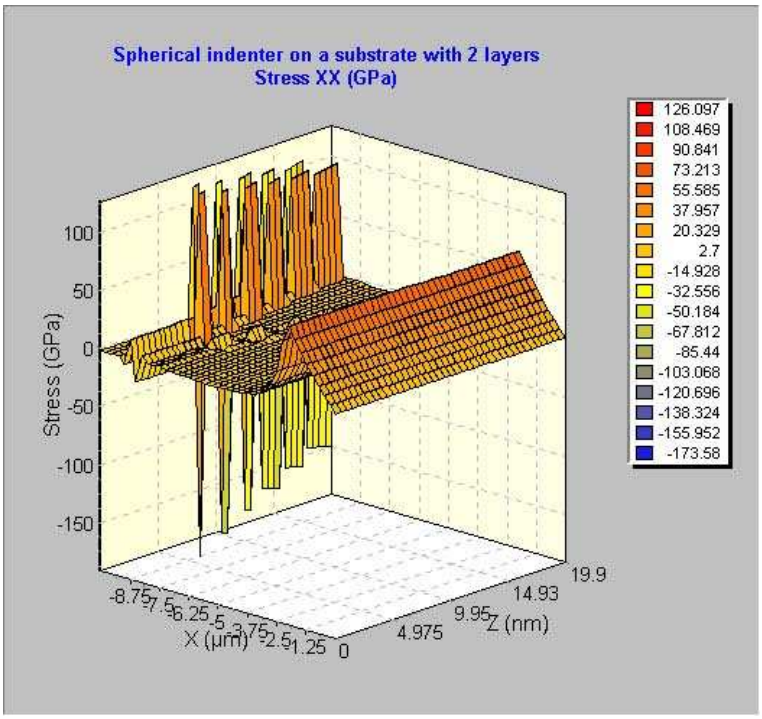


Figure 4: Scatter in data of ELASTICA.

In some cases special scatter, unexpected in this appearance, was noticed. In these cases the normal presentation of ELASTICA produced smooth stress distributions. But investigated in detail there were inexplicable sharp peaks in the data. The maximum-stress-detection methods tended to converge there and so the values of the later introduced objective function of layered systems with neighboroughed thickness distributions had an unsmooth relation.

Furthermore, instead of continous behavior very near the surface or direct under the indenter a discrete behavior of the stress values was sometimes noticed. There, changing the x- or z-coordinate up to approximatly 0.007 μm the returned stress values were the

same (see figure 5).

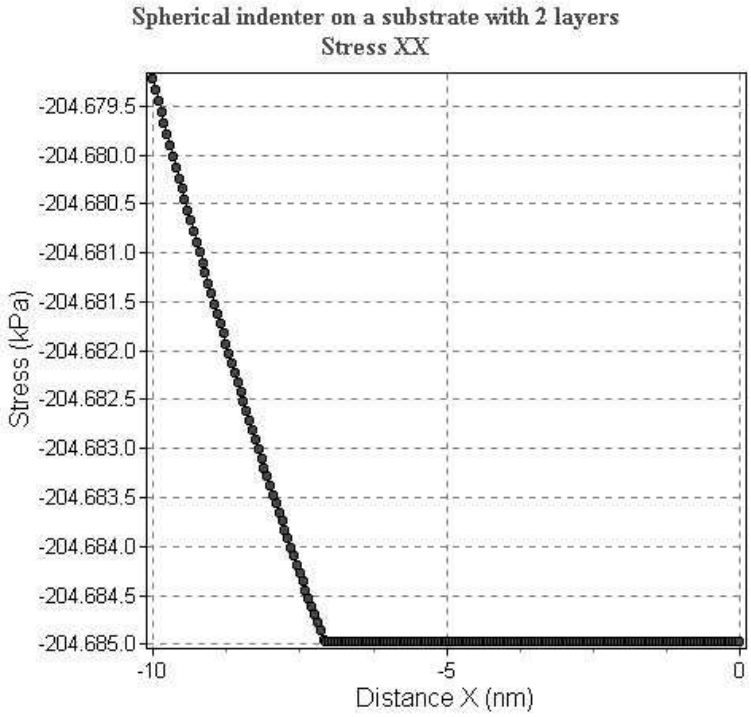


Figure 5: Plateau, complicating the numerical handling. Picture taken from ELASTICA.

Due to the posibility of this behavior, using finite-difference derivative approximation (e.g. see [14], p. 168)

$$\frac{\partial f}{\partial x_i}(\vec{x}) \approx \frac{f(\vec{x} + \varepsilon e_i) - f(\vec{x})}{\varepsilon},$$

the finite difference ε was restricted to $0.02 \leq \varepsilon \leq \infty$, if the evaluated point was near the surface, the interfaces or direct under the indenter. Therewith functionality of optimization methods using derivatives could be ensured.

Normaly the boundary between two layers is located at the toplayers thickness (e.g. if a toplayer is 1 μm thick, the next layer starts at z = 1μm and the toplayer ranges $0\mu m \leq z < 1\mu m$). But in some cases the stress values of a layer were extended behind the boundary into the deeper layer.

4.2.2 Typical Stress Distributions

General. Spherical indentations, only with normal load, produces radialsymmetric stressdristributions. So the dimension of the search area can be reduced from 3 downto 2. In cartesian coordinates with x parallel and z perpendicular to the layerboundaries (depth). The x-coordinate can also be constraint to values higher or lower than zero because of the radial symmetry.

Adding a lateral force in direction of x the symmetry for y still exists but no more for x. But considering about the direction of the lateral force and the consequences to the material one can realize, that the maximum stress will be behind the indenter [6].

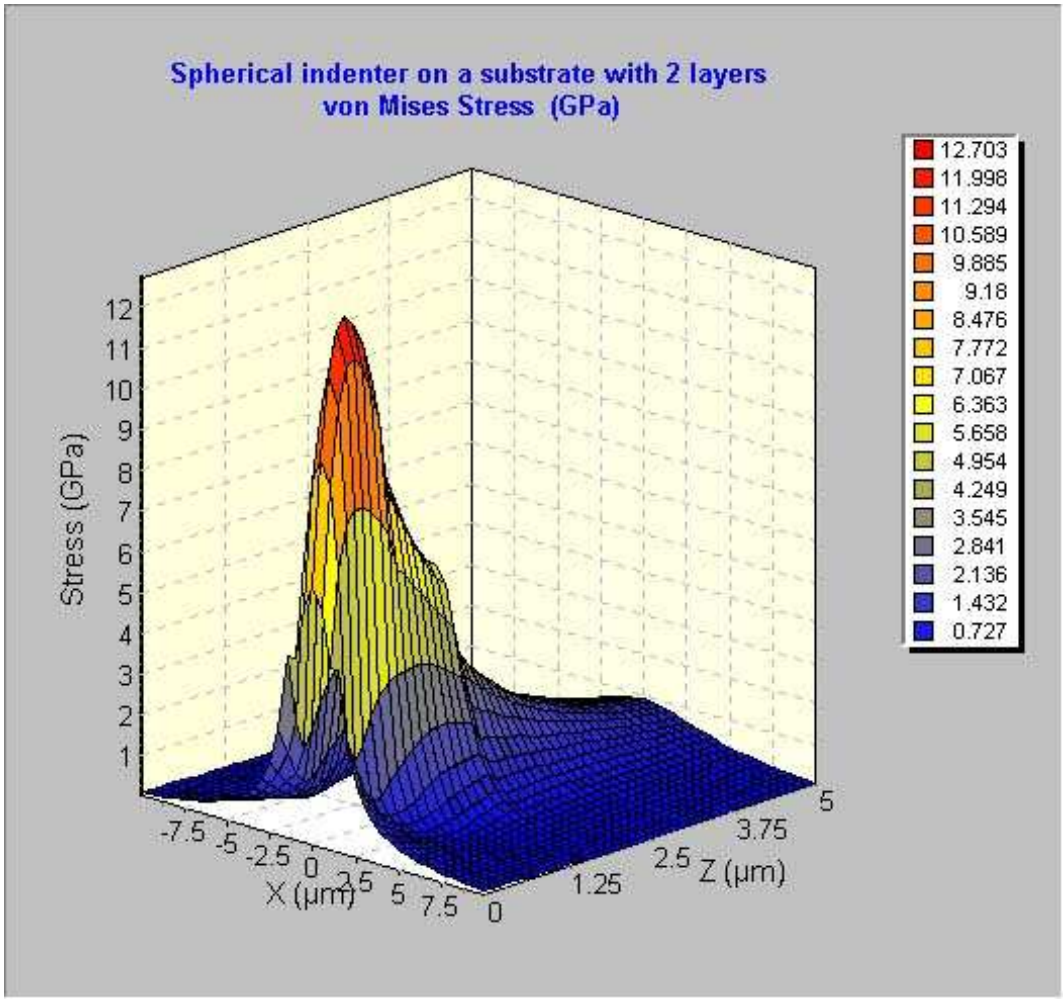


Figure 6: Stress distribution as illustrated by ELASTICA. The von Mises field for a 2-layer system is shown.

Von Mises Stress. In case of only normal loads the maximum of the von Mises stress is always located perpendicular to the layers under the indenter [2]. In every layer one finds not more than 3 maximums, where one is the global for this layer. The second might have its reason in the different mechanical properties of the neighbouring layers. The last one can occure in a layer as a result of a specific combination load-, indenter- and material properities, so that, unexpected, the von Mises stress tends to zero near the surface and a local maximum can be found at the surface [2]. Neverless a linesearch is sufficent to locate the global maximum.
If lateral forces occur, the maximum drifts, as aforementioned, behind the indenter. At the surface this drift is bigger than in the deeper materials. Therefore the search has to be 2-dimensional.

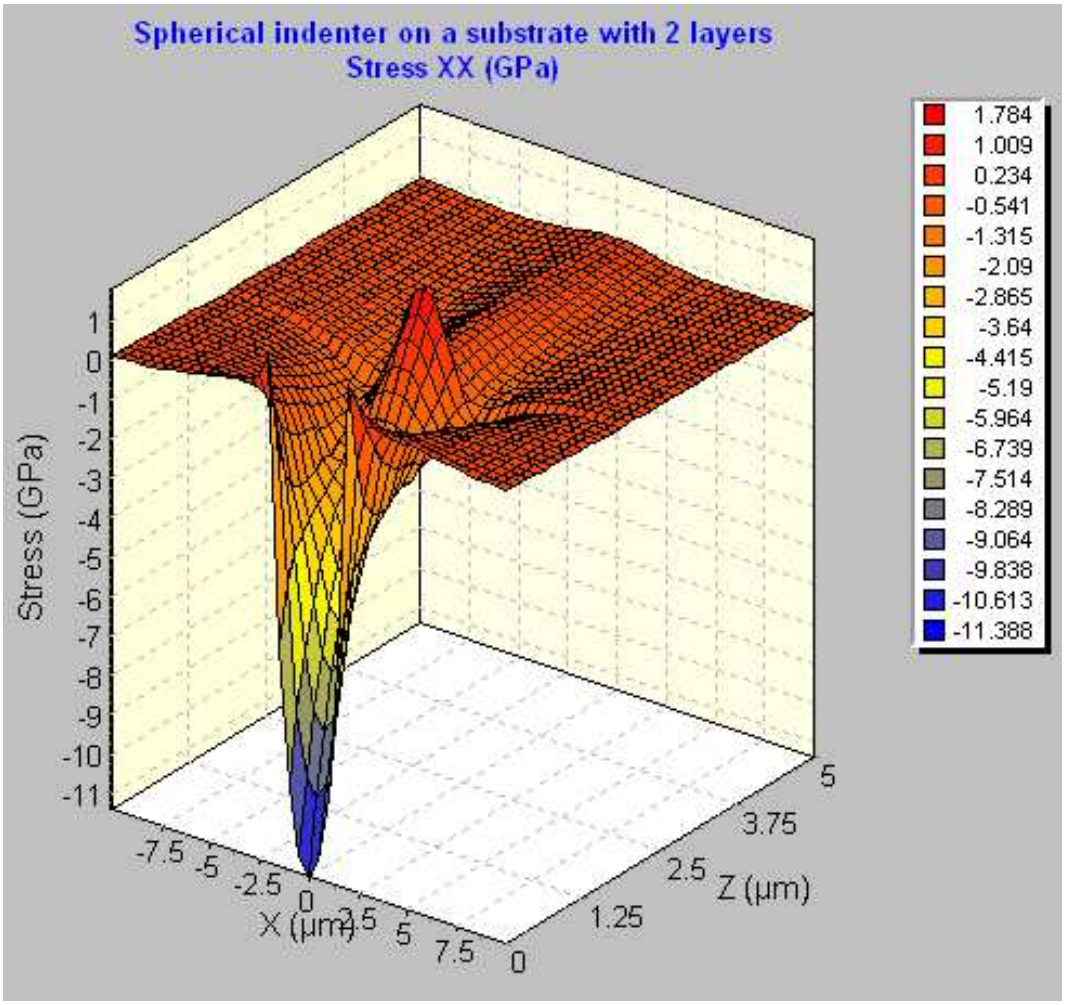


Figure 7: Stress distribution as illustrated by ELASTICA. The σ_{xx} field for a 2-layer system is shown.

Tensile Stress. In view of the tensile stress, reducing the dimensions of the searcharea is not possible. Even with only normal load the maximum is located in the x-z-plane, but often along the layer boundaries. With lateral force, additionally, the maximum drifts away from the boundaries. For normal force only, one might think about checking the boundariers only. Often the maxima are of sharp kind, so they must be located very precisely to ensure a correct stress value.

Contact Radius The contactradius R_{con} is an important information to reduce the effort of the search. The maximum tensile stress is often located near the value of R_{con} on the upper layer boundary. In deeper layers the maximums are widely spread around R_{con}, sometimes many times bigger. In unlayered media the maximum stress is located not deeper than R_{con} and in layered media approximately not deeper than 2 times R_{con} [6].

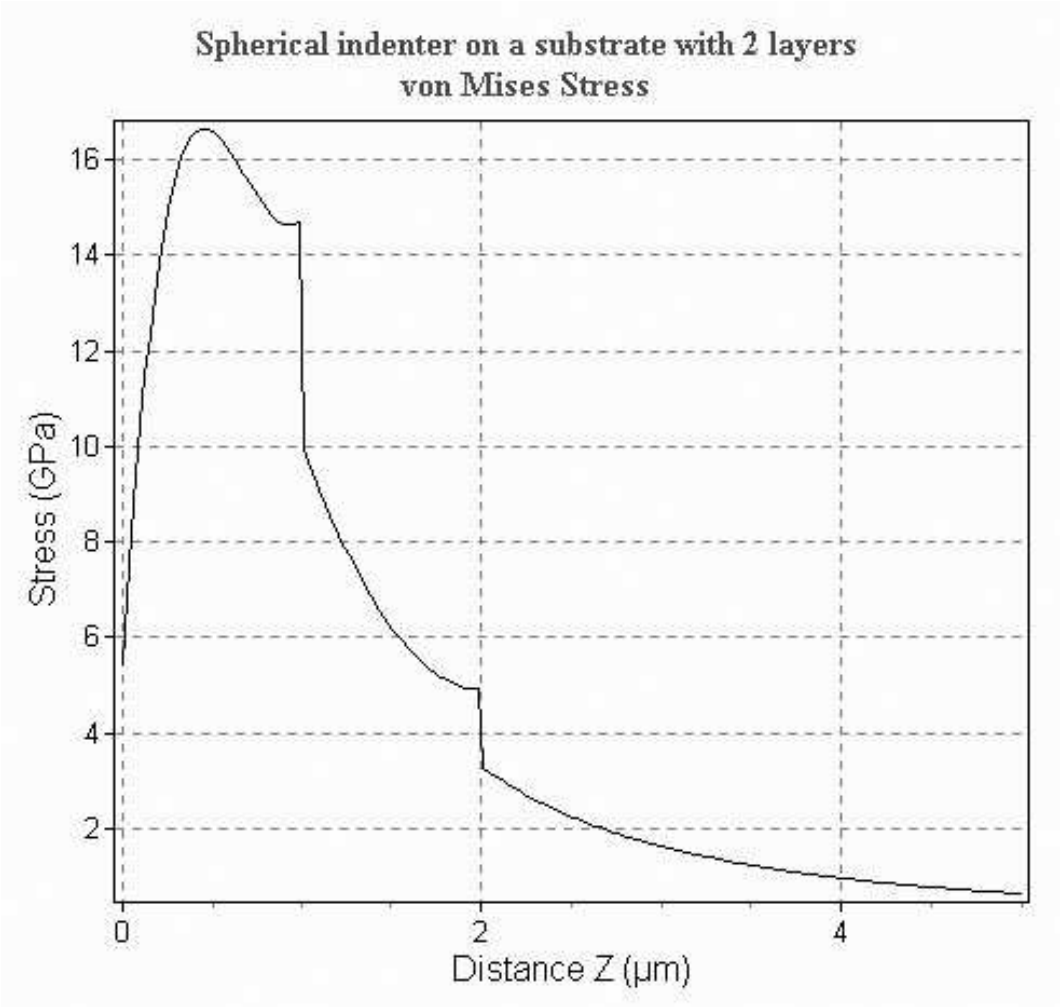


Figure 8: Von Mises stress distribution as illustrated by ELASTICA.

4.2.3 Layer Boundaries

Getting a stress value from the used DLL one sends the DLL the coordinates of the point to evaluate and the stress value is returned. The origin of the coordinate system is fixed at the surface directly under the indenter (see Figure 9).

Even though one does not need to specify which layer should be investigated, the DLL gives no hint in which layer the evaluated point lays. The domain of a single layer $\mathbb{L}^{(i)}$ used in ELASTICA can be given as

$$\mathbb{L}^{(i)} = \{z \in \mathbb{R} : z^{(i)} \leq z < z^{(i)} + d^{(i)}\} \times \{x \in \mathbb{R}\} \times \{y \in \mathbb{R}\}, \tag{19}$$

where i specifies the layer and a point $\vec{p}$ is given by $\vec{p} = (x, z, y)^T$.
Due to the force always acting into the x-direction, only the negative x-axis has to be looked at. For the z-direction, the domain is trimmed, the layer boundaries are not always sticked to by ELASTICA.

$$\tilde{\mathbb{L}}^{(i)} = \{z \in \mathbb{R} : z^{(i)} + \varepsilon \leq z < z^{(i)} + d^{(i)} - \varepsilon\} \times \{x \in \mathbb{R} : x \leq 0\} \tag{20}$$

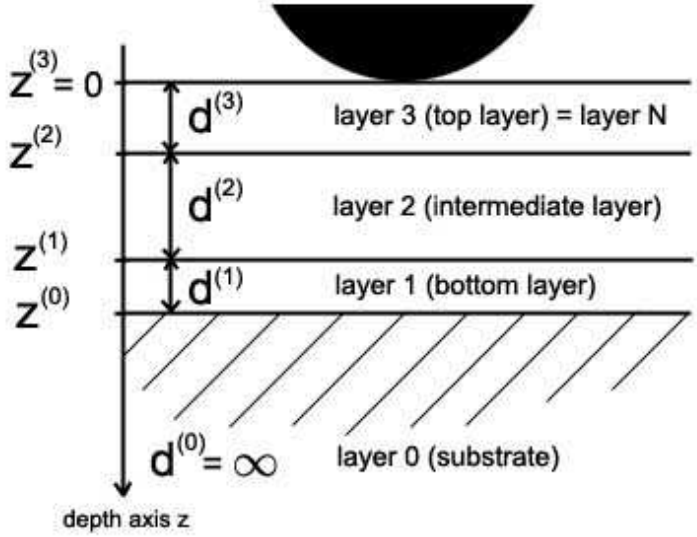


Figure 9: Layers of the compound.

Where ε was chosen as $\varepsilon = 0.00001$.
These restrictions of $\tilde{\mathbb{L}}^{(i)}$ are realized with an outer-penalty-method, since there are kinks behind the boundary of $\tilde{\mathbb{L}}^{(i)}$. For methods using derivatives the penalty was chosen slightly increasing

$$\sigma(\vec{p}) = \begin{cases} \sigma(\vec{p}) & \text{if } \vec{p} \in \mathbb{L}^{(i)} \\ \sigma(\vec{p}_{bound}) - |\vec{p}_{bound} - \vec{p}| & \text{otherwise} \end{cases}, \tag{21}$$

where $\vec{p}_{bound}$ is the nearest point in $\tilde{\mathbb{L}}^{(i)}$ wiht respect to $\vec{p}$.
Since the simplex method does not require the calculation of derivatives, the stress values can get a hard penalty with no relationship to the boundary of the searcharea. So leaving the searcharea using simplex method a stress value about 10^5 was returned. This spares function evaluations using the DLL and calculationtime. Further the simplex can not 'sit' over the boundary, even though the minimum is located there, as it is possible with a penalty.

$$\sigma(\vec{p}) = \begin{cases} \sigma(\vec{p}) & \text{if } \vec{p} \in \mathbb{L}^{(i)} \\ 10^5 & \text{otherwise} \end{cases}. \tag{22}$$

4.2.4 Numerical Investigations

To find out which optimization methods to use, and where they should start, a huge number of numerical of investigations have been carried out. The investigated layer systems and the load conitions were randomly chosen:

- **Number of layers:** 2
- **Layer thickness,** $d^{(i)}$**:** 0.01 - 2.00 μm

- **Young's modulus,** E**:** 50 - 1300 GPa
- **Poisson's ratio,** ν**:** 0.05 - 0.49
- **Indenter radius,** R_{ind}**:** 1 - 1000 μm
- **Normal indenter force,** F_{normal}**:** 1 - 1000 mN
- **Lateral indenter force,** $F_{lateral}$**:** 1 - 1000 mN

The accuracy was set to 4, in order to speed up the calculations and to reduce cases of scattered data. This restriction were necessary to keep the effort of calculating the reference stress values acceptable. Because of the similarity of the stress dirstibutions, the chosen restrictions should produce representative stress distributions, also for 1 and 3-layer systems.
The reference maximum stress value σ^{ref} was determined with a fine mesh for every system. The maximum distance between the mesh points in both x and z-direction was chosen to be 0.05 μm, but at least 21 stress values should be calculated in every layer. The mesh size was $6 \cdot R_{con}$ in x-direction. For the z-direction, the mesh size was the layer thickness or in the substrate $3 \cdot R_{con}$.

Comparing the values of the mesh with the values found by optimization, convergence was reached, if the found stress value σ^{max} approached the reference value by 10MPa.

$$(\sigma^{ref} - 10MPa) \leq \sigma^{max} < \infty \tag{23}$$

This was due the fact, that critical stress values are only available with a precision of 100MPa.
Elastica has some restrictions what kinds of layered systems it can handle appropriately. For instance, might the quotient of two neighbouring layer Young's moduli lay between 0.25 and 8. For this section, this does not matter, since the task is to locate the maximum stress value in a given stress distribution. Otherwise, systems failing these limitations, the probability of getting scatter in the data is increased.
To compare different optimization methods the number of function evaluations are considered, because of their costs to calculate. For example, the evaluation of 1000 points of a 3-layer-system with normal load only, takes 13,4 seconds on a Intel Pentium III (700 MHz) computer.

Many of those calculations were made on AMD Athlon(tm) XP 2400+ computers located in a computer pool at the TU Chemnitz. Sequential applications were running lone standing on each computer. Altogether, in mean 20 of these computers had a calculation time of approximately one week each. This shows the important role of even low-level use of parallel processing for this work.

Von Mises stress

Number of investigated compounds	500000
Number of investigated layers	1500000
Convergence reached	1499955
Convergence not reached	45
Convergence [%]	99.997
Mean failure of non-converg. [%]	~ 1
Function evaluations for 1 golden sect. search	12

Table 3: Systems investigated with golden section search; $F_n = 1 \ldots 999$ mN, $F_l = 0$ N and R_{ind}= $1 \ldots 999$ μm

Normal Load. In the case of normal load only, the maximum von Mises stress can always be found under the indenter. So a line minimization along z-axis under the indenter is needed. The method of golden section search is chosen. Even though the convergence of this method is proofed only for unimodal functions, with an extension good results could be reached.
If there is more than one maximum in one layer, only one might be located between the layer boundaries and the others are always at the layer bounderies. Golden section search tends to find the inner maximum, only if an outer maximum is much bigger than the inner one, otherwise the outer will be found. After the maximum stress has been found, it should be compared to the boundary values. Therewith, global convergence could be reached almost everytime.

Lateral and Normal Load. To locate the maximum stress value in the case of both lateral and normal load, 2-dimesional search methods have been used. BFGS and simplex were started from different positions to ensure convergence everytime. It was found that using only one optimization method starting from different positions, in some case, from no starting position convergence can be reached. Starting positions were typically -0.0, -0.5, -1.0, -1.5 and -5.0 times R_{con} for x and $z^{(i)} + \varepsilon, z^{(i)} + d^{(i)}/2$ or $z^{(i)} + d^{(i)} - \varepsilon$ for z-coordinate.
At this point, a further optimization procedure is introduced :

Rastering the boundary of the serach domain $\tilde{\mathbb{L}}^{(i)}$ is denoted as *boundary-raster*. In detail, stress values are calcutated within the range $-6 \cdot R_{con} \leq x \leq 0$ in direction of x and within $z^{(i)} + \varepsilon \leq z \leq z^{(i)} + d^{(i)} - \varepsilon$ in direction of z. The distance between two calculated values is normally set on 0.05 μm, but at least 20 stress values are evaluated on each edge. The highest stress value found and its position is returned.

Combinations of BFGS and simplex failed in only 0.5 percent of the investigated layers. Performing an additional boundary-raster the convergence was increased to almost 100. Due to the use of the boundary-raster, the starting positions of BFGS and simplex on the layer boundaries can be spared and the number of stress value evaluations is reduced. To reach the maximums more precisely, starting from the best position found, BFGS and simplex are used the second time.
Since boundary-raster converged in 99.86% of investigated cases, omitting the 2-dimensional search methods is not possible.

test series	Range Fn [mN]	Range Fl [mN]	Range R_{ind} [μm]
A	1 ... 999	1 ... 99	1 ... 99
B	1 ... 999	1 ... 99	1 ... 999
C	1 ... 99	100 ... 999	1 ... 99

Table 4: Test series investigating von Mises stress

There were only 4 compounds where convergence was not reached in a single layer. In these cases, the toplayer thickness was only up to 0.02 μm and there was scatter in the data. Because of the scatter in the data these cases can be neglected.

The number of stress values evaluated by boundary-raster increases with the thickness of the layers thicker than 1μm and contact radii larger than 0.166μm. Therefore another procedure to get the maximum stress on the has been developed. It will be denoted as *boundary-check*. Boundary-check is also rastering within the same ranges as boundary-raster but it makes only 100 stress value evaluations in direction at the layer boundaries and 20 stress value evaluations in direction of z under the indenter. At each position of the largest stress value of the two layer boundaries an one-dimensional simplex is started. Starting from x=0 in the rastered points at each layer boundary the first porsition where the stress values decrease is located. Around this position a golden section search is started. Finaly, at the position of the largest stress value of the search domain edges a last one-dimensional simplex is started.

Boundary-check reached convergence in 99.74% and is regarded as an suitable surrogate for boundary-raster additional to BFGS and simplex.

test series	SIMPLEX	BFGS	MARQUARDT	bnd.-raster	bnd.-check
A	60,9	107,0	139,2	897,6	355,3
B	69,9	102.0	160,0	1890,9	363,1
C	65,3	110.8	101,1	428.89	350,7

Table 5: Mean stress value evaluations of the optimization methods

Tensile Stress

Normal Load with or without Lateral Force. At the begining of this investigations, loads with and without lateral force were handled as same kind of problem. The choosen optimization methods and their starting positions were the same as for von Mises stress with normal and lateral load, because of the 2-dimesional search. Espacially for σ_{xx} starting points for x near R_{con} were very successful.
But having more data, it was found, that in case of only normal load, boundary-raster and boundary-check converged in 99.993% or 99.989%, respectively. In the 0.007% of non-convergence at boundary-raster a soft toplayer (Young's modulus near 100 GPa) was on a much harder intermediate layer (Young's modulus about 600-800 GPa). The substrate was harder than the intermediate layer. This difference of Young's moduli was too high, leading to non-convergence. Increasing the accuracy of ELASTICA to

test series	Range Fn [mN]	Range Fl [mN]	Range R_{ind} [μm]
D	1 ... 999	1 ... 99	1 ... 99
E	1 ... 999	1 ... 99	1 ... 999
F	1 ... 999	0	1 ... 99
G	1 ... 99	100 ... 999	1 ... 99

Table 6: Test series investigating stress XX

5, convergence could be reached, due to better stress field given by ELASTICA with accuracy 4. In these, it is sticked to the limitations of ELASTICA.

But turning the focus back on optimization, avoiding the shown failure reasons one can conclude, that boundary-check is able to reach 100 percent convergence. So the search can be reduced to a generalized 1-D coordinate-search and the calculation time is massively reduced.

If also lateral forces occur, boundary-raster reaches convergence at high levels, about 99.7 %. The 0.3 % of non-convergence are due to high differences of Young's modulus. In these cases, the lateral force has displaced the maximum stress behind the indenter. But the displacement is nature, that a 2-dimensional search had to be performed.

Test series	Investigated layers	combination	bnd.-raster	bnd.-check
A	59577	0	42	53
B	6693	0	0	1
C	20523	0	7	10
D	9000	15	54	126
E	96000	3	104	188
F	97911	0	5	3
G	98250	1	9	32

Table 7: Number of insvestigated layers and failed investigations of the combination of methods, boundary-raster and boundary-check

4.3 Finding the Best Coating

4.3.1 The State Space

The user's input is mathematically covered by the state space Ω. It represents the set of all possible material and thickness combinations. For specifying the possible coatings, the user determines a number of materials which are combined in any possible way. For a N-layer system, this results in a N-dimensional material subspace M. Even though the materials are determined by the 4 different parameters Young's modulus, Poisson's ratiom, yield strength and tensile strength, they are regarded as single points within the 1-dimensional material subspace.
Moreover, the thicknesses might vary within the real numbers, spanning another continuous subspace $\hat{\mathbb{R}}$. Consequently, the state space itself is a mixture of descrete and continuously varying variables. If we denote the number of layers by N, the spate space can be defined as

$$\Omega = M^N \times \hat{\mathbb{R}}^N.$$

$\hat{\mathbb{R}}$ is chosen not to be equal to $\mathbb{R}$ for leaving open the possibility of confining the thicknesses to a particular range.
The choice for discrete determined materials, is of practical purpose, and is based on two reasons: The experimental creation of a proposed material with particular Young's modulus and Poisson's ratio goes still beyond feasibility, nowadays. But the main problem, however, is the lack of knowledge about the corresponding critical stress values, which are not directly correlated to Young's modulus or Poisson's ratio. Without properly specifying the stress tolerance by the critical stress values, no useful prediction about durability can be stated. Unfortunately, treating the materials within a single space proves complicated for finding the best material combination. An *effective Young's modulus* has been taken into consideration, leading to a ranking of the materials and even the whole compounds. An outline is presented section 4.4.

While the materials do not allow for formal optimization, an optimal thickness for a determined coating can be obtained in a straightforward manner. This is discussed in chapter 4.3.4.
Furhtermore, one might think of including other paramaters such as indenter radius, forces, or substrate material, for instance, into the state space. As long as they vary continously, they are apt for being optimized by the methods used in this work.

4.3.2 The Objective Function

For a particular state, the maximum stresses are obtained by the suboptimization of section 4.2. This can be regarded as a mapping $\vec{\sigma}$ from the state space to a stress state containing the 2(N+1) maximum stresses as appearing within compound domains.

$$\vec{\sigma} : \Omega \to \mathbb{R}^{2(N+1)} \tag{24}$$

Now, a measure for the coating quality can be stated. Thereby the quality measure takes into account the failure criteria of section 2.4. It has to be taken care that not a single stress value may exceed its critical limit since the failure of one layer means

the failure of the whole compound. Additionally should all stress values be kept at a moderate level. In optimization terms, this quality measure is called *objective function* mapping the stress state into the real numbers. We will call this function f.

$$f : \vec{\sigma} \to \mathbb{R} \tag{25}$$

Different coatings can be compared with respect to their objective function value.
Our optimization task can now be stated mathematically as finding the optimal state $\vec{\omega}_{opt}$ within the state space whose stress state minimizes the objective function.

$$\vec{\omega}_{opt} \in \Omega : f(\vec{\sigma}(\vec{\omega}_{opt})) = \min_{\vec{\omega} \in \Omega} f(\vec{\sigma}(\vec{\omega})) \tag{26}$$

Besides the important failure criteria, the particular choise of f depends on practical intentions: The more information is covered, the more complexity the topology will exhibit. This means a more complicated search for the minimum. Thus, a compromise has to be made. Already by the fact that several objectives are merged into a single real number one can guess the big deal of designing the function properly.

Multi-objectives Optimization. The state which meets all objectives best is only a trade-off, since the goals are conflicting. Apparently, this is a *multi-objective optimization* (also multi-criteria or vector optimization). Normally, only so called *Pareto optimality* can be achieved. No solution exists in the conventional sense, but a set of solutions, where the degrees of optimality can neither be distinguished nor improved by another state with respect to all objectives. The user has still the choice. However, the optimization will be confine to searching for the best state with respect to the objective function.

Designing the Function. Fitting several objectives to an appropriate mathematical expression is a quite subtle task. For the calculations of this work, the two objectives mentioned above are included. Therefore, a *stress measure* κ is defined, indicating how far the critical stresses are actually approached.

$$\kappa = \sigma^{max}/\sigma^{crit} \tag{27}$$

Further on, the stress measure for each layer is distinguish by a superscript, e.g. $\kappa^{(i)}$. Thereby the substrate is denoted by 0 and the layers from 1 through N.
Obviously, κ may not take values above 1. This is treated by the use of the *Heavyside step function* Θ defined as

$$\Theta(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{28}$$

Using this function, the exceeding of the criticial stress values can be punished with the *failure measure*.

$$\Theta(\kappa - 1) = \begin{cases} 0 & \text{if } \sigma^{max} < \sigma^{crit} \\ 1 & \text{if } \sigma^{max} \geq \sigma^{crit} \end{cases} \tag{29}$$

Since the Θ-function lacks a closed representation, at best as a limit, for instance,

$$\Theta(x) = \lim_{a \to \infty} \frac{\tanh ax + 1}{2}$$

the approximation

$$\tilde{\Theta}(x, a) = \frac{\tanh ax + 1}{2}$$

has been used, leading to a soften transition from 0 to 1. By the parameter a, the degree of smoothness is tuned. It serves as a control mechanism for switching the objective function from 'rich in information' to 'easy to optimize'. A large a gives a more step-like Θ-function meaning a more step-like objective function as depicted in figure 10.

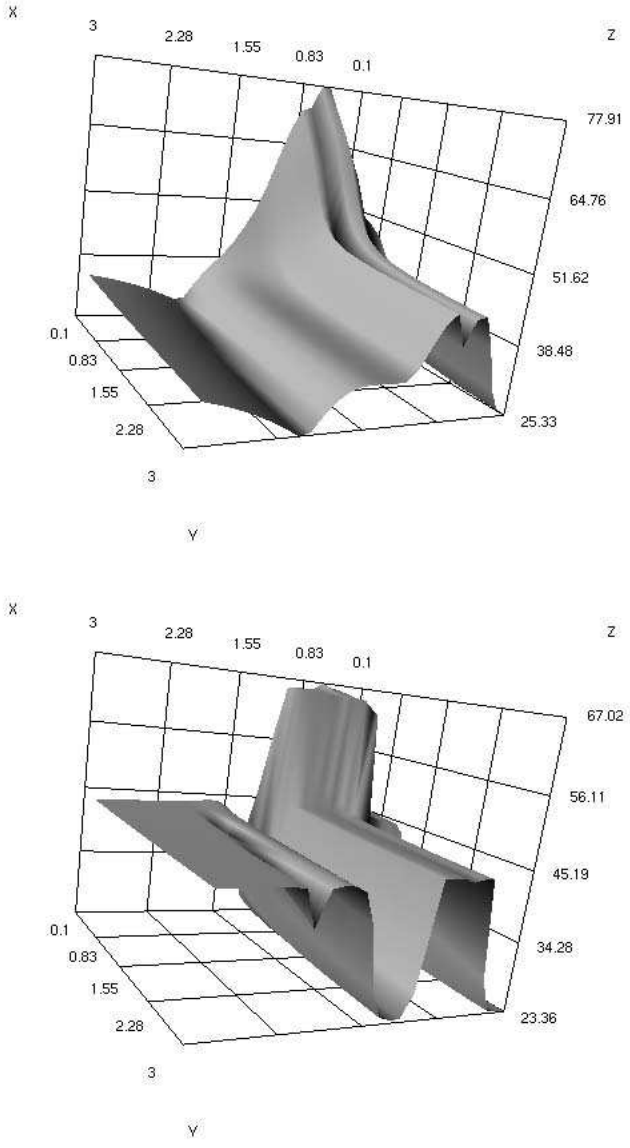


Figure 10: Influence of the smoothness of the Θ-function involved in an objective function. Different degrees of smoothness (top: a=5, bottom: a=50) result in differently smooth objective functions, even different optimums.

Depending on how the Θ-function is involved into the objective function, even the optimum can be drifted since a different objective function means different objectives met by the optimum. The smaller a, the worse the intended objectives are met. The

optimum will be reached easier, but the goals aimed at are not represented within the optimum, anymore. Further, states of overall-low stress maximums have to be found. Therefore, the stress measure κ can simply be added.

$$f = \Theta(\kappa - 1) + \kappa \tag{30}$$

With which the two objectives are considered, though not sufficiently. The single parts have to be weightened in an appropriate way. The Θ-function should be multiplied by a constant. Additionally, it has to take into account both the von Mises stress and the tensile stress of each layer, including the substrate.
This leads to the final objective function.

$$f = \sum_{i=0}^{N} \left[t \cdot \Theta(\kappa_M^{(i)} - 1) + \kappa_M^{(i)} + t \cdot \Theta(\kappa_{xx}^{(i)} - 1) + \kappa_{xx}^{(i)} \right] \tag{31}$$

Where κ_M denotes the von Mises stress measure $\sigma_M^{max}/\sigma_M^{crit}$ inside the distinct layer and substrate domains, and κ_{xx} the tensile stress measure $\sigma_{xx}^{max}/\sigma_{xx}^{crit}$. The parameter t can again tune the importance of the failure measure and has strong influence on the topology.
An appropriate expression can be built in a lot of different ways. For instance, a simpler constructions can be use as well. Another function might involve the maximum stress measure, only.

$$f = \max_{i \in [0,N]} \left(\kappa_M^{(i)}, \kappa_{xx}^{(i)} \right) \tag{32}$$

Despite the fact, that proper stress states are not seperated in the most suitable fashion, a function constructed this way indicates the exceeding, and in the end the failure of the compound, best. All objective function values above 1 implicate the ineligibility of the coating. Furthermore, it allows to compare between coatings of different kind.
One could also think of constructing a function, *indicating the critical load*. Therefore, the objective function should have a minimum where the stress measure takes 1.

$$f = \min_{i \in [0,N]} \left(|\kappa_M^{(i)} - 1|, |\kappa_{xx}^{(i)} - 1| \right) \tag{33}$$

Which function to chose depends only on user's intention. With our choice we tried to point the way towards constructing the function properly. Generally, using the real valued objective function is a very convenient and flexible way for meeting different objectives, although it demands due consideration. The outcome is quite sensible to its construction, but at the same time it holds plenty of capabilities. Newly included objectives are being met 'automatically' by the optimum. So, for instance, a monetary aspect could be involved easily. Unfortunately, the problem can become arbitraryly complicated by the increasing complexity of finding the optimum.

4.3.3 Coarse-graining the State Space

After an objective function has been set up, reflecting the relevant goals, a strategy for obtaining the optimal state has to be found. Two aspects should especially be taken care of: The method ougth to be fast and the optimum should be found with

sufficient certainty. To design a method, best adapted to the problem, the structure of a variety of problems has to be reviewed. But since the evaluation of a single point of the state space takes approximately half a minute, real-time investigations cannot be performed within reasonable time. Therefore, the online calculation has been replaced by the pre-sampling of the state space along a coarse mesh. Using this pre-calculated state space, investigations could then be carried out extensively without any temporal effort. However, for obtaining a dense mesh, a single personal computer is not sufficient. The number of points to evaluate goes beyond the available time. A parallel approach was of immanent importance. To give an example, for a 3-layer system, with varying thicknesses between 0.1 and 3.0 μm, and a raster point distance of 0.1 μm, 27000 points have to be evaluated. Studying compounds consisting of 3 different materials only, yielding 27 different coatings, leads to 250 days overall running time. This computative burden could only be managed by the help of the supercluster CLiC (Chemnitzer Linux Cluster, section 5.4). Up to 400 cluster nodes were able to supply the results within one day. A vast number of pre-samplings have been carried out, accumulating roughly 1 GB of data. On basis of the pre-sampled state space points, thorough investigations could be accomplished. The analysis presented within the next chapter rests on this pre-cached data.

Interpolation. To extract continuous data out of discrete ones, points laying inbetween the mesh points have to be interpolated. In the case of 3 dimensions, the method of *trilinear interpolation* has been utilized. Thereby, the data v is given at the vertices of a unity cube $\mathcal{C}$.

$$\mathcal{C} = [0,1] \times [0,1] \times [0,1] \tag{34}$$

The interpolation point $(x,y,z)^T$ must lay inside.

$$(x,y,z)^T \in \mathcal{C} \tag{35}$$

Then, the value of the point is evaluated as

$$v(x,y,z) = \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} v(i,j,k)|i-x||j-y||k-z| \tag{36}$$

For non-unity cubes, a simple translation and scaling has to be done. With this formula, the point inside is fitted to a trilinear function. For 2 dimensions, interpolation in the plane (*bilinear interpolation*) can easily be derived from setting $z = 0$.

4.3.4 Finding the Best Thicknesses

As already mentioned, the optimization will be treated for the materials and the thicknesses, seperately. Within this section, the materials of the coating are fixed and investigations into optimal thicknesses are presented. In terms of section 4.3.1, a point in the subspace M^N is fixed, and only the subspace of thicknesses $\hat{\mathbb{R}}^N$ is searched for the optimum.
Most investigations are based upon the pre-sampled data (see section 4.3.3), because the real-time evaluation of state space points takes too long. This way, the algorithms can be tested extensively without any expensive calculation time. Furthermore, it gives

insight into the structure of the problem by the possibility of plotting the objective function for the whole subspace of thicknesses.

The 2-layer systems.
Considering a 2-layer system with fixed materials, apparently 2 independent thicknesses can be varied. This gives a 2-dimensional subspace of thicknesses.
Mainly the following systems have been analysed:

Materials (# = 12)	**Situation**	$\mathbf{F_{normal}}$	$\mathbf{F_{lateral}}$	$\mathbf{R_{ind}}$	$\mathbf{d_{max}}$	$\mathbf{d_{min}}$
Layers:	1	20 mN	0 mN	10 μm	0.1 μm	4 μm
Steel1,Steel2,Ni,TiN1,	2	50 mN	0 mN			
TiN2,GaAs,DLC,Si_3N_4,	3	50 mN	10 mN			
CrN,Cr,Al_2O_3,cBN	4	50 mN	50 mN			
Substrate:	5	100 mN	10 mN			
Steel1	6	100 mN	50 mN			
Indenter:	7	50 mN	0 mN	30 μm		
Diamond	8	100 mN	0 mN	30 μm		

Table 8: 2-layer systems under investigation.

The material properties can be looked up in table 1.
Coatings are composed of all possible permutations of the materials. Some coatings arisen this way, are of no meaning for technical applications. They only serve as example systems for testing purposes.
Making use of the ability of plotting the objective function for these coatings, figure 12 shows a typical graph.

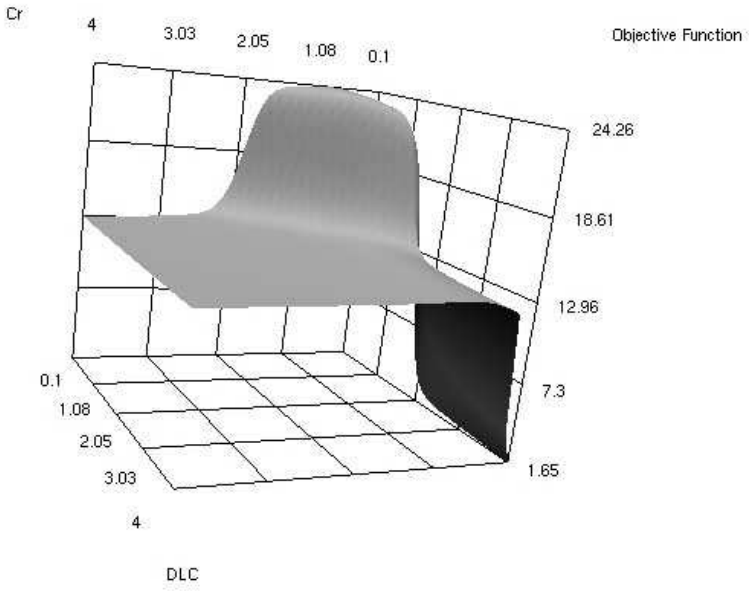


Figure 11: Objective function of a 2-layer system taken from situation 1 of table 8. Top layer Cr, bottom layer DLC.

Depicted is the objective function of a Cr on DLC 2-layer system depending on the thicknesses. The load situation was specified as system 1 from the table 8. The objective function indicates proper coatings by a function value below 5. Another characteristic graph descends from a CrN on GaAs system, again from situation 1.

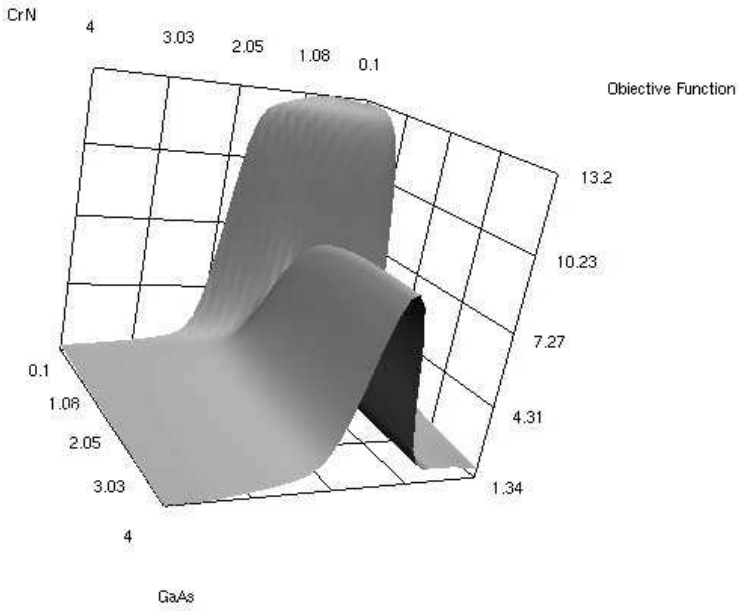


Figure 12: Objective function of a 2-layer system taken from situation 1 of table 8. Top layer CrN, bottom layer GaAs.

Most systems under investigation appear similarly. Only the use of lateral forces changes the nature noticably. What can be assumed is, that the optimization might be straightforward for this smooth surfaces. For tackling the optimization, different ways of using CG and the Simplex method has been put to use. Their capability of finding the optimum is statistically evaluated. Therefore, the real optimum f_{opt} has been determined reviewing the sampled state space points. The found optimums of the algorithms could then be compared to the real values. Table 9 shows the succeeding for several approaches.

Algorithm	**Avg. eval.**	**Mean(f-f$_{opt}$)**	**Max(f-f$_{opt}$)**	**Stddev(f-f$_{opt}$)**
4 x CG	79.6181	0.2131	10.9127	1.2198
9 x CG	311.5694	**0.1938**	**10.9127**	1.2107
4 x Simplex (inside)	168.7222	0.0433	3.1743	0.288
4 x Simplex (vertices)	129.9236	0.001	0.0297	0.0043
9 x Simplex	321.5069	**0.00075**	**0.0226**	0.0031

Table 9: Thickness optimization for a 2-layer system. Comparison between objective function value found by optimization and exact optimum f_{opt}. Averaged over 144 different trials.

Few points, equally distributed over the space, served as starting points for Simplex

or CG algorithms, indicated as '4 x Simplex', '9 x Simplex' and so on. The best point achieved was taken as the minimum. With a maximum number of about 300 iterations, the optimum is found with high accuracy.

Therewith, the evaluation time for state space points in 2-layer systems could be lowered in such a way, that 2-layer optimizations can perfectly be carried out on a single personal computer in reasonable time. For finding the best thicknesses of the 144 combinations using the materials presented in table 8, only about 6 hours of calculation time is needed.

Finally, table 10 shows some of the best coatings among the test composites of system 1.

The 3-layer systems.
For investigating systems of 3 layers, more effort had to be raised. The subspace of thicknesses contains 3 dimensions, from now on. Moreover will the complexity of the objective function increase noticably. Systems investigated are presented in table 11.

Unlike situation 1, the investigation of situation 2 could no longer be carried out based on interpolation of the pre-calculated state space points. The coarse-graining of the state space as done for situation 1 would by far exceed reasonable effort. The estimated calculation time is about 7 days for a parallel machine with 300 nodes. However, using situation 1 as reference values, situation 2 can be verified. The material properties are again extracted from table 1.

For the coarse-graining of the state space of situation 1, a distance of 0.1 μm has been chosen, leading to 729000 overall mesh points. Assuming the calculation time for a single point to be roughly half a minute, the total time needed with a personal computer exceeds 250 days. Using the massively parallel supercluster, the results have been obtained within 1 day only. (situation 2 can easily be calculated!)

For the sake of clarity, only 2-dimensional slices of the 3-dimensional objective function can be depicted. Figure 13 shows a 3-layer system of CrN on DLC on TiN as obtained from situation 1 of table 11. The slice is taken for fixed top layer at 0.1 μm.

Describing the increased complexity more clearly, figure 14 shows another graph of the same system.

Presented is a slice among the thickness of the intermediate layer at 3 μm. For the case of normal load, many systems revealed such a complex structure. To cope with these complex functions, the strategy from the 2-layer systems has to be extended. The 3-dimensional state space cube is being covered by a finer mesh than for 2-layer systems. These points are then chosen as starting points for the optimization algorithms. As already mentioned, the temporal effort for evaluating the objective function for 3-layer systems is significantly increased.
For keeping the computative effort as low as possible, it has furthermore been tried to compare the function at those mesh points, and only to launch the algorithms at the best point obtained. This proved to be the best strategy. The statistically viewed succeeding of several approaches is illustrated within table 12. Algorithms denoted by

Layer 2 / d_{opt}	**Layer 1 / d_{opt}**	**Objective Function Value**
DLC / 4 μm	cBN / 4 μm	1.3414334
GaAs / 4 μm	cBN / 4 μm	1.3523791
cBN / 4 μm	GaAs / 0.2 μm	1.3633127
TiN1 / 4 μm	cBN / 4 μm	1.3784352
cBN / 4 μm	cBN / 4 μm	1.3792775
...		
DLC / 4 μm	DLC / 4 μm	1.399854
GaAs / 4 μm	DLC / 4 μm	1.4107715
Steel1 / 4 μm	cBN / 4 μm	1.4385284
Steel2 / 4 μm	cBN / 4 μm	1.4462684
Si_3N_4 / 4 μm	DLC / 4 μm	1.4505223
TiN1 / 4 μm	DLC / 4 μm	1.4505915
...		
DLC / 4 μm	TiN1 / 3.4 μm	1.6508576
GaAs / 4 μm	TiN1 / 4 μm	1.655811
Nickel / 4 μm	cBN / 4 μm	1.6706522
Al_2O_3 / 4 μm	TiN1 / 4 μm	1.7232803
CrN / 4 μm	TiN1 / 4 μm	1.7356573
...		
TiN2 / 4 μm	GaAs / 0.1 μm	1.8203163
Cr / 4 μm	TiN1 / 4 μm	1.8558246
Al_2O_3 / 4 μm	TiN2 / 4 μm	1.9158362
Si_3N_4 / 4 μm	TiN2 / 4 μm	1.9496992
CrN / 4 μm	TiN2 / 4 μm	1.9511829
...		
cBN / 4 μm	Al_2O_3 / 0.1 μm	2.0199223
cBN / 1.2 μm	TiN2 / 4 μm	2.0397836
Cr / 4 μm	TiN2 / 4 μm	2.0449049
DLC / 4 μm	CrN / 1.5 μm	2.073129
DLC / 4 μm	CrN / 1.5 μm	2.073129
GaAs / 4 μm	CrN / 1.8 μm	2.1088569
...		
GaAs / 4 μm	Cr / 0.1 μm	6.1534639
...		

Table 10: Extract from best coatings and thicknesses for system 1 of table 8. (Layer 2 = Top Layer, Layer 1 = Bottom Layer)

Materials	Situation	$\mathbf{F_{normal}}$	$\mathbf{F_{lateral}}$	$\mathbf{r_{indenter}}$	$\mathbf{d_{max}}$	$\mathbf{d_{min}}$
Layers: TiN,DLC,CrN **Substrate:** Steel1 **Indenter:** Diamond	1	50 mN	0 mN	10 μm	0.1 μm	3 μm
Layers: cBN,Cr,Ni,CrN, TiN,DLC **Substrate:** Steel1 **Indenter:** Diamond	2	50 mN	0 mN	10 μm	0.1 μm	3 μm

Table 11: 3-layer systems investigated.

125 pts. + 1 Simplex, for instance, indicate the 'mesh points only' strategy, followed by a Simplex method.

Algorithm	Avg. eval.	Mean(f-$\mathbf{f_{opt}}$)	Max(f-$\mathbf{f_{opt}}$)	Stddev(f-$\mathbf{f_{opt}}$)
125 x CG + 1 CG	4049.8	0.2082	2.2443	0.5215
125 x CG + 1 Simplex	4168.4	**0.0313**	**0.5318**	0.1070
125 x 1 Simplices + 1 Simplex	2494.9	0.0144	0.1820	0.0417
27 x 2 Simplex + 1 Simplex	928.33	0.0125	0.1820	0.0393
125 x 2 Simplex +1 Simplex	4215.6	**0.0065**	**0.0993**	0.0203
125 pts. + 1 x Simplex	125+85.9	0.1654	1.6028	0.4267
729 pts. + 1 x Simplex	729+82.74	**0.0070**	**0.0769**	0.0163

Table 12: Thickness optimization for a 3-layer system: Comparison between objective function value found by optimization and exact optimum f_{opt}. Averaged over 144 different trials.

The reason for the few iterations necessary, is mainly the fact, that optimal points are very often to be found at the faces or vertices of the cube, where the mesh points match the optimum quite close.
For cases similar to that being analyzed, an optimum can be found with sufficient accuracy. But generally, one has to be cautious with the mesh points approache, since odd material combinations can produce odd objective functions.
What finally turned out, is, that a combination of slight rasterization and the Simplex method achieves the best results. An optimum can be found with roughly 800 iterations. This corresponds to approximately 3 hours for a single personal computer.

Table 13 shows the best and some selected coatings found by the above optimal strategy applied to situation 1 of table 11.

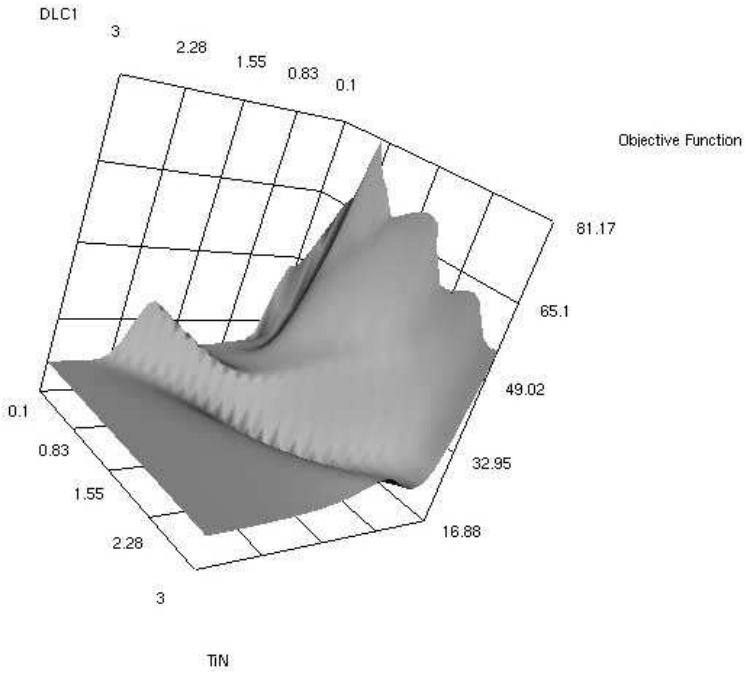


Figure 13: A slice of an objective function for a 3-layer system. CrN on DLC on TiN, with fixed CrN at 0.1 μm depending on the thicknesses of the intermediate and the bottom layer.

The results of situation 2 are presented in table 14 and can partially be compared to the reference values.

Layer 3 / d_{opt}	**Layer 2 / d_{opt}**	**Layer 1 / d_{opt}**	**Objective Function Value**
CrN / 3 μm	TiN / 3 μm	DLC / 0.1 μm	2.46361947333521
TiN / 3 μm	TiN / 3 μm	DLC / 0.1 μm	2.47960283373466
DLC / 2.7 μm	TiN / 3 μm	DLC / 0.1 μm	2.50432600318796
TiN / 0.4 μm	DLC / 1.4 μm	DLC / 3 μm	3.40777378051681
CrN / 0.3 μm	DLC / 1 μm	DLC / 3 μm	3.75475032148235
DLC / 3 μm	CrN / 3 μm	DLC / 0.1 μm	4.6125140681004
CrN / 2.9 μm	CrN / 3 μm	DLC / 0.1 μm	4.88780276378715
TiN / 2.1 μm	CrN / 3 μm	DLC / 0.1 μm	5.03842879519236
DLC / 3 μm	DLC / 3 μm	DLC / 3 μm	5.18742495072021
TiN / 1.4 μm	DLC / 1.2 μm	TiN / 0.1 μm	8.73584178485022
CrN / 1.9 μm	DLC / 0.7 μm	TiN / 0.1 μm	16.5987219217151
TiN / 1.6 μm	DLC / 1.1 μm	CrN / 0.1 μm	16.8774802779885
DLC / 3 μm	DLC / 3 μm	TiN / 2.4 μm	17.0905380537534
TiN / 0.1 μm	CrN / 1.7 μm	TiN / 1.2 μm	17.9473178159378
CrN / 0.5 μm	CrN / 1.1 μm	TiN / 1.2 μm	18.2223200030387
CrN / 0.8 μm	DLC / 0.7 μm	CrN / 2 μm	19.0200024859194
DLC / 3 μm	CrN / 1.3 μm	TiN / 1.1 μm	19.630133048272
TiN / 0.4 μm	TiN / 0.1 μm	TiN / 2.2 μm	22.2353730257833
DLC / 3 μm	DLC / 3 μm	CrN / 0.1 μm	23.3772995717815
DLC / 3 μm	TiN / 0.1 μm	TiN / 0.1 μm	23.6062779465379
DLC / 3 μm	TiN / 0.1 μm	CrN / 0.1 μm	24.1292009884777
DLC / 3 μm	CrN / 0.1 μm	CrN / 0.1 μm	24.3233635979436
CrN / 0.1 μm	TiN / 0.1 μm	TiN / 2.4 μm	28.1357864632204
CrN / 1.6 μm	TiN / 1 μm	CrN / 0.1 μm	32.9351584332568
TiN / 1.8 μm	CrN / 0.8 μm	CrN / 0.1 μm	35.4118718563443
TiN / 0.8 μm	TiN / 1.7 μm	CrN / 0.1 μm	37.093542613421
CrN / 3 μm	CrN / 3 μm	CrN / 3 μm	41.0290184065862

Table 13: Best thicknesses for coatings as calculated for system 1 of table 11. (Layer 3 = Top Layer, Layer 1 = Bottom Layer)

Layer 3 / d_{opt}	**Layer 2 / d_{opt}**	**Layer 1 / d_{opt}**	**Objective Function Value**
DLC / 2.87 μm	cBN / 2.99 μm	DLC / 0.13 μm	1.46252977848053
CrN / 2.85 μm	cBN / 2.98 μm	DLC / 0.12 μm	1.56785833835602
Cr / 2.98 μm	cBN / 3.00 μm	DLC / 0.12 μm	1.57687604427338
TiN / 2.94 μm	cBN / 2.95 μm	DLC / 0.12 μm	1.58111214637756
cBN / 3.00 μm	cBN / 2.99 μm	DLC / 0.13 μm	1.61190319061279
Nickel / 3.00 μm	cBN / 3.00 μm	DLC / 0.10 μm	1.78000211715698
cBN / 3.00 μm	TiN / 3.00 μm	DLC / 0.10 μm	2.45177173614502
CrN / 3.00 μm	**TiN / 3.00 μm**	**DLC / 0.10 μm**	**2.47271919250488**
TiN / 3.00 μm	**TiN / 3.00 μm**	**DLC / 0.10 μm**	**2.47945857048035**
DLC / 3.00 μm	**TiN / 3.00 μm**	**DLC / 0.10 μm**	**2.50644207000732**
Cr / 3.00 μm	TiN / 3.00 μm	DLC / 0.10 μm	2.84126210212708
cBN / 1.55 μm	DLC / 2.27 μm	cBN / 0.10 μm	3.03178524971008
cBN / 0.33 μm	DLC / 1.68 μm	DLC / 2.98 μm	3.18769812583923
DLC / 3.00 μm	TiN / 0.10 μm	cBN / 0.10 μm	3.28217101097107
TiN / 0.43 μm	**DLC / 1.37 μm**	**DLC / 3.00 μm**	**3.38497352600098**
...			
Nickel / 3.00 μm	Cr / 3.00 μm	CrN / 3.00 μm	56.3245582580566
Nickel / 3.00 μm	Nickel / 3.00 μm	CrN / 3.00 μm	62.1966705322266

Table 14: Extract of best thicknesses for coatings as 'real-time' optimized for system 2 of table 11. Real-time optimized meaning without interpolation. Bold lines can be compared to table 13. The differences between the objective function values is due to the interpolation used in table 13. (Layer 3 = Top Layer, Layer 1 = Bottom Layer)

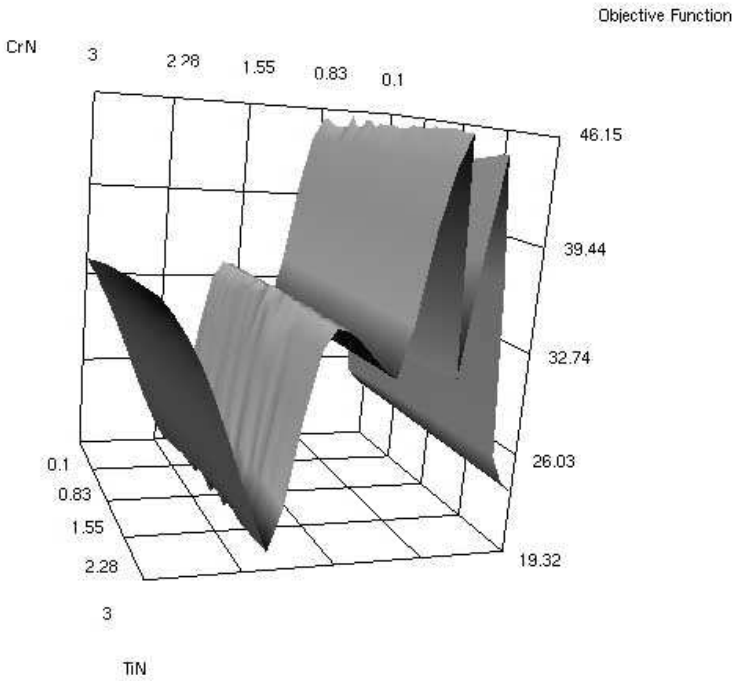


Figure 14: Slice of the objective function of the 3-layer system CrN on DLC on TiN. Intermediate layer DLC is fixed at 3 μm.

4.4 Finding the Best Material Combinations

The toughest task of the whole optimization is to find the best material combinations. First considerations about using Simulated Annealing had to be put down, because the discrete 'material points' within the state space do not provide a suitable neighbourhood relation.

Instead, an estimation about the quality of the compounds using the material properties, solely, is considered. Therefore, an *effective Young's modulus* as

$$E_{eff} = \frac{E}{\sigma_{xx}^{crit} \cdot \sigma_M^{crit}} \tag{37}$$

is introduced. This happens to make sense since a low Young's modulus induces only low stresses. Whereas soft materials are not very robust. Therefore, in the denominator, the critical values have to be taken into account. For multi-layers-system, several runs with different functional dependencies had been accomplished. Thereby it turned out, that the deeper the layer is located within the compound, the less it effects the overall stress state. A weighted summation over the layer's effective Young's modulus, obtaining an *estimation value* $\mathcal{E}$ for the quality of the whole compound, has been carried out.

$$\mathcal{E} = \sum_{i=1}^{N} w(i) \cdot E_{eff}^{(i)} \tag{38}$$

($i = 1$: bottom layer, $i = N$: top layer)

The substrate can be omitted, because it is the same for all coatings. For the weighting $w(i)$, a linear function with a slope of the magnitude of E_{eff} proved appropriate. Obvious choices like $w(i) = 1$ gained no convincing success.
Now the correlation between the estimation value and the actual objective function can be depicted. To do so, the estimation is used for a ranking and enumeration of the coatings with respect to the attained value since its magnitude is not expressive at all. Figure 15 and 16 show the relations for situation 2 of table 11. Omitting coatings with same adjacent materials, the number of compound decreases to 150.

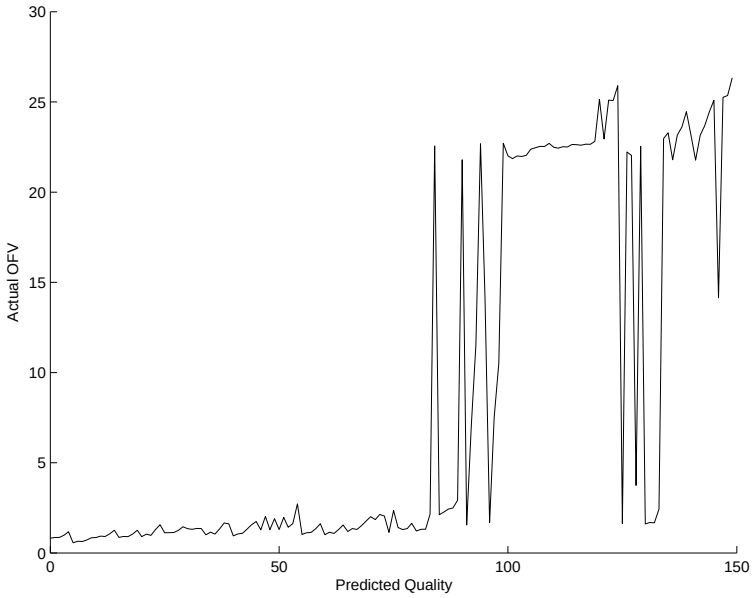


Figure 15: Estimating the best objective function value (OFV) of a 3-layer system. X-axis: Ranking with respect to the estimation value, y-axis: the actual objective function value.

Coatings with an objective function value above 5 are inadequate. What can be seen is the bold tendency towards the desired monotonicity. Moreover are the improper compounds satisfactorily characterized by bad estimation values. The large part of the undamaged compounds is found within the first half of the ordering. Only a few runaways do not fit at all. The strong fluctuation is due to the punishment for exceeding critical stress limits. Objective functions with a more gentle behaviour will turn out a more gentle appearance. For a better illustration of the ability to seperate proper and improper coatings, the Θ-punishment does the better job.
The method has also been tested with systems of two layers, leading to similar results. Figure 17 shows a magnification for situation 1 from table 8.

Using the estimation value as a preselection of the coatings to be optimized for the thicknesses, is by all means helpful and saves a lot of time. Further investigation are stongly suggested.

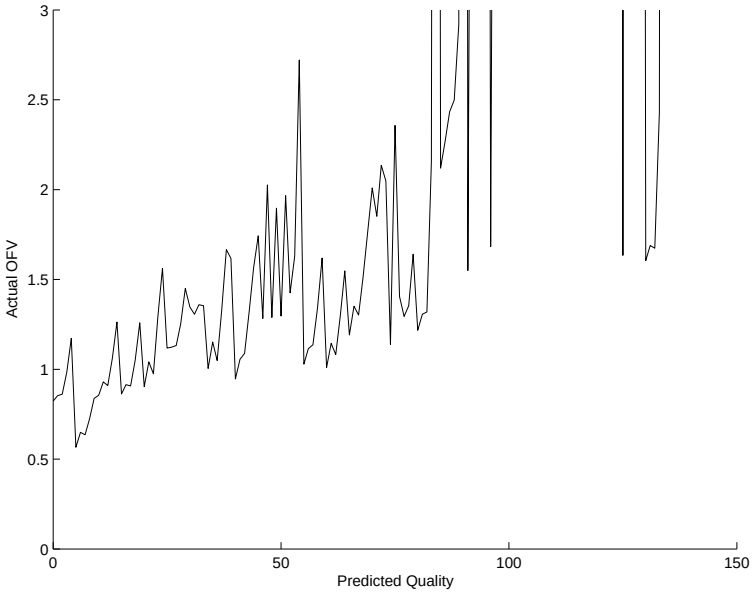


Figure 16: Estimating the best objective function value (OFV) of a 3-layer system. Magnification of figure 15.

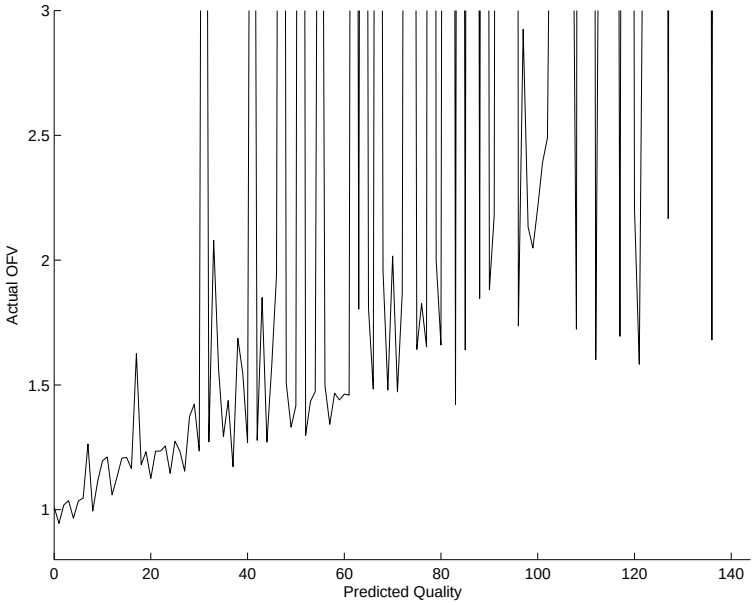


Figure 17: Estimating the best objective function value (OFV) of a 2-layer system.

5 Parallel Computation

The search for the optimal solution can easily go beyond the scope of available time as the number of materials, layers and range of thickness increases. The more possibilities the user searches the optimum in, the longer the runs take. In view of companies, computational power exists sufficiently, though often spread over many single personal computers. Combining this loose power to an efficient cluster reveals new levels of performance. Within this chapter, parallel concepts and the Chemnitzer Linux Cluster, on which our investigations have mainly been carried out, are presented shortly.

5.1 About Parallel Concepts

The most popular general clasification of parallel architectures has been proposed by Flynn [15] in 1972. The four architecture classes are depicted in figure 18.

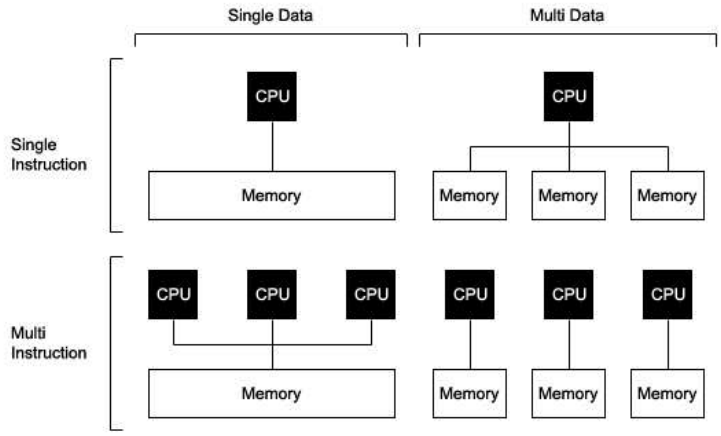


Figure 18: Architecture classes of parallel computers by Flynn.

For short, they are called

- **SISD:** (Single Instruction Single Data) one CPU working with one memory
- **SIMD:** (Single Instruction Multi Data) same instruction on several memories
- **MISD:** (Multi Instruction Single Data) several CPUs sharing one memory
- **MIMD:** (Multi Instruction Multi Data) seperated CPUs with their own memory

Personal computers are most often SISD, except multiprocessor PCs which are MISD. SIMD architectures are called vector-computers because one job can be done simultaniously for various data, similar to vector and matrix operations (which is in fact the big advantage of SIMD computers). Many Cray computers are SIMD based.
The greater part of parallel machines are MIMD where data and instructions are distributed by so called *message passing*, then stored and processed locally and finally sent back. Another approach for providing the data is MISD where data does not

need to be propagated since all nodes can easily access it as global memory. However, the increase in complexity leads to increasing administrative effort simply because accessing the same memory at the same time by two different CPUs will complicate the maintenance of data consistency and has to be strictly avoided by reliable software or hardware components.

Furthermore, parallel computers can be classified by

- **Type and number of processors:**
  loosely coupled, of arbitrary type: *coarse-grained parallelism*
  huge number of similar processors: *massively parallel*
- **Presence or absence of a global control mechanism:**
  to what extend a contolling node determines the single processes;
  related to classification by Flynn
- **Synchronous or asynchronous operation:**
  presence or absence of a common global clock synchronizing the operations
  (e.g. SIMD machines synchronized in a natural way)

Another important distinction in high-perfomance computing has to be made between *parallel* and *distributed* computation. They differ in several aspects regarding the connection among the nodes. The communication of a parallel system is supposed to be predictable, reliable, and formed by close-standing similar processors. Unlike the distributed system, where processors might be far apart and equipped with different performances, resulting in a more problematic communication. The links between nodes are variable and topology may change with time. The nodes are mostly not for exclusive parallel computing purpose. Mostly they are single-standing personal computers. There is surely no clear line between parallel and distributed computing, but what we are intending to make use of rather focuses on distributed computing.

5.2 Parallel Processing

In the first place, algorithmic operations on parallel computers can be distinguished by the way in which the parallel nodes are utilized, that is, whether they are working together in a *synchronous* or an *asynchronous* manner [16]. The striking advantage gained by asynchronous implementation is the high performance improve. The overall performance will, for the ideal case, be the sum over the single performances. Whereas the synchronous execution demands *phases* after which the processes have to interact for task and data propagation. Since the faster nodes have to wait for the slowest before the next phase can be initiated, the final perfomance is the number of computers times the performance of the slowest.
Different algorithms are possible at different levels for asynchronous implementation. Gradient-like optimization algorithms, for instance, might struggle againts fully asynchronous implementation, since the second step cannot be done before the first. On the other hand is a coarse-graining such as described in section 4.3.3 highly appropriate. What to aim at for the highest performance impove, is the most possible degree of asynchronous treatment.

5.3 Towards Parallel Implementation

The intention was to connect computers in a local network as a cluster computer to reduce the calculation time. The administrative effort of this cluster should not demand installed software (e.g. parallel communication libraries like MPI or MPIch [17]). So it has been decided to develop a TCP-IP based communication protocol for distributing the task, since every network-computer should provide TCP-IP communication and parallel communication libraries are not required.

In a client-server-hierachy a server process divides the task in similary smaller parts, distributes them to the clients and manages the results. The comunication effort could be kept low and estimated to approximately 100 Bytes every 10 seconds, which is regarded as neglectable. When the client applications are started, whether they automatically connect to a prescibed and running server, or they are connected by the server, which needs a list of available clients. Using only one computer, both applications have to be started. The subtasks are chosen in such a manner, that assynchronous processing is possible. A typical example for a subtask is the determination of the stress maximums within a layer system.

Due to the ELASTICA DLL, the application has been developed for Mircosoft Windows operating systems. On the other hand, huge computation capacity is available under Linux operating system at Chemnitz University of Technology (e.g. the CLiC and various computer pools). Using the Windows emulation software *Wine* [18], the door to this ressources has been opened.

5.4 The Chemnitzer Linux Cluster CLiC

The *Chemnitzer Linux Cluster CLiC* [19] is a massively parallel system (MIMD) of currently 528 computing nodes with a Intel PIII processor (800 MHz) and 512 MByte RAM each, 2 similar server nodes and a compound of Gigabit- and Fast-Ethernet.

Within this work it has been managed, for the first time, to run a Windows program with a self-made communication device on the CLiC using the Windows emulation software Wine (further details see [20]). Approximately 200 nodes were used for at least 2 weeks calculation time each and up to 420 nodes at the same time.

The communication and managment effort is depicted in figure 19, measured by switching off the calculations. The sharp bend at 150 nodes is due to other subcluster-sizes when the amount of nodes was increased. It is a matter of how the nodes are clustered by the administrative server process of CLiC, causing the response time to differ slightly. We also recognized an influence of how CLiC was utilized by other users.

The speedup is illustrated in figure 20, and the calculation time in figures 21, where the average subtasks calculation time amounts 14.7 seconds. Due to the long calculation time using only one calculation node the amount of subtasks was fixed to 1764 or 4900, respectively. Using hundreds of nodes, the chosen number of subtasks was too small in relation to the number of nodes. When the last subtasks were distributed, some nodes were not involved into the calculation and so the speedup was reduced. However, most of the calculations done in this work had a number of hundreds of thousands subtasks

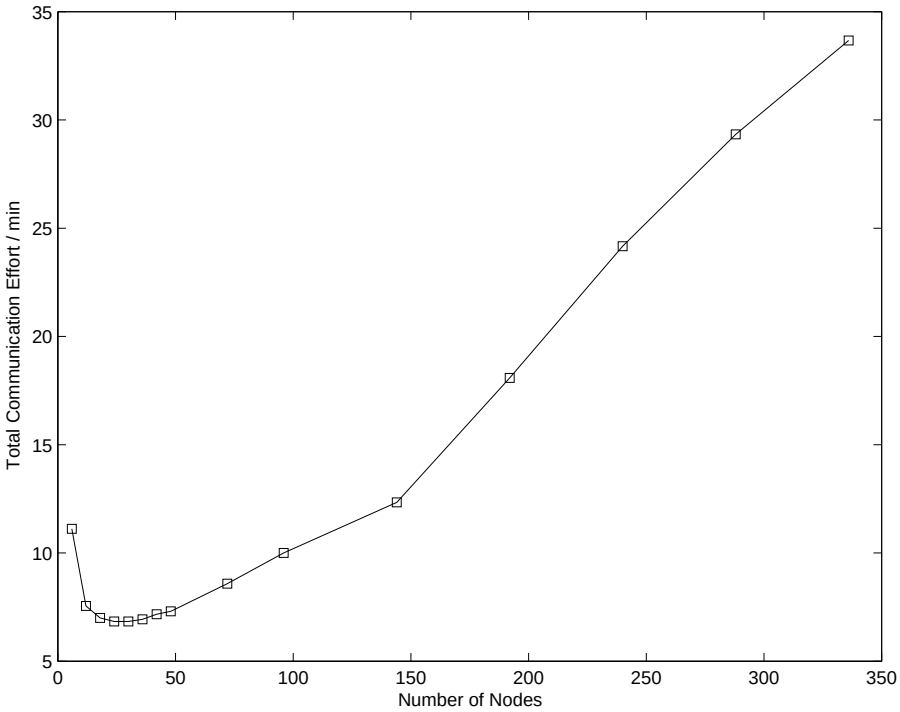


Figure 19: The overall communication and managment effort for the exchange of 1458000 data packets on CLiC (corresponds to calculation effort of a roughly 300 days optimization on a single node)

and a better speedup should have been realized.

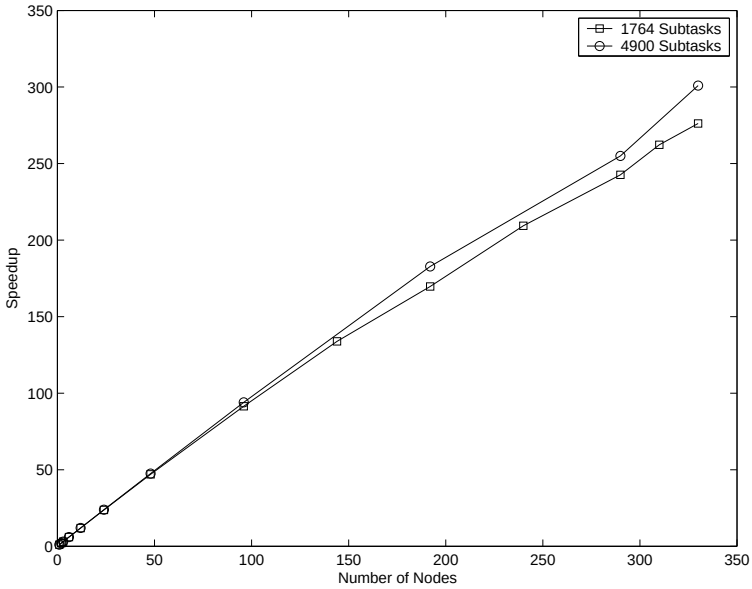


Figure 20: The speedup for 2 different problems.

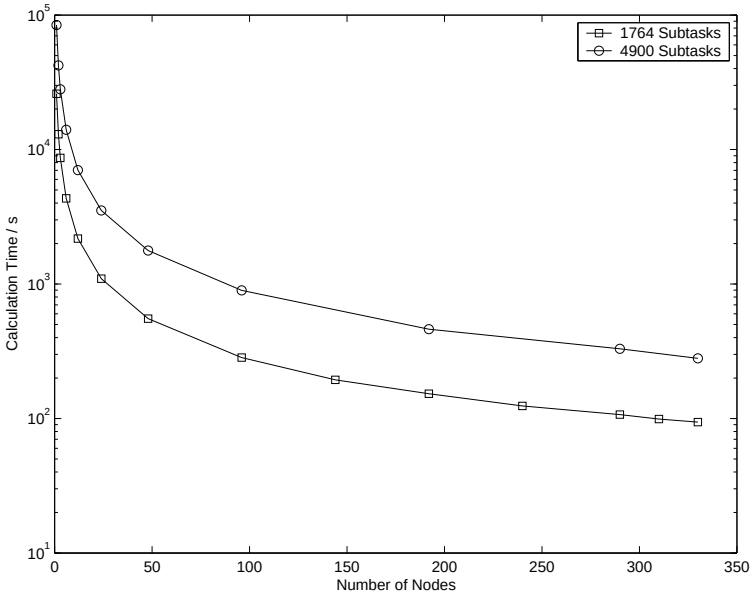


Figure 21: Calculation time in relation to the number of nodes for two different numbers of subtasks.

6 Implementation

This chapter is dedicated to the implementation in Delphi and will shortly outline the source code which came into being as part of this work. Furthermore, we intend to impart the ability to make use of the Delphi components for setting up and controlling an optimization problem. We opt for Delphi because its component approach and its integrated developement environment (IDE) is one of the most convenient and most easy to use, experienced by the authors.

6.1 Delphi Components

During this work we created a couple of components for managing the optimization comfortably.

- **TElastica:** main component, basic features
- **TOptimize:** implementation of optimization algorithms
- **TElasticaServer:** TCP/IP server component
- **TElasticaClient:** corresponding TCP/IP client
- **TRaster:** convenient component for coarse-graining the state space

TElastica is the basic component necessary for setting up the contact situation and retrieving the raw stress data from the DLL. A symbolic description looks like:

```
type TElastica=class(TComponent)
...
   Layers     : Array of TMaterial;
   LayerCount : Integer;
   Indenter   : TIndenter;
   procedure SetIndenter  (Material:TMaterial;
                           Radius,Fn,Ft:Double;
                           IndenterType:TIndenterType);
   procedure SetLayer     (Index:Integer; Layer:TMaterial);
   function Initialize    (Accuracy:Integer):Double;
   function GetStressValue(FieldValue:TFieldValue; x,y,z:Double):Double;
   function GetObjectiveFunctionValue:Double;
   procedure DetermineStressMaximums(FromLayer,ToLayer:Integer);
   function EstimateCompound:Double;
   function DetermineOptimalThicknesses(Min,Max:TVector;
                                        Accuracy:Integer):Double;
...
end;
```

The number of layers is determined by the last call of SetLayer, wherefore the setting of the layers should be started with 0, the substrate, and consecutively the others. The kind of indenter and stress is determined by a value of the range of TIndenterType or TFieldValue, respectively.

```
type TIndenterType = (itSpherical,itCircular,itConical);
     TFieldValue =   (fvStressXX,fvStressYY,fvStressZZ,fvStressXY,
                      fvStressXZ,fvStressYZ,fvDeformU,fvDeformV,
                      fvDeformW,fvPrStressS1,fvPrStressS2,
                      fvPrStressS3,fvVonMisesStress,
                      fvHydrostaticStress,fvMaxShearStress);
```

The TMaterial structure looks like:

```
type TMaterial = record
      Name: string[80];
      E,n,d:Double;
      CriticalVonMises,CriticalTensile,
      MaxVonMises,MaxTensile: Double;
      MaxVonMisesPosition,MaxTensilePosition: TXYZ;
     End;
```

To set up a contact situation one should, at first, specify the indenter, then the layers (starting with number 0), and then initializing. Furthermore, an individual objective function can be implemented into a Windows Dynamic Link Library (DLL) giving the user the ability of meeting his own goals. Therefore one has to specify the following function within the DLL:

```
function ObjectiveFunction(Layers:PMaterials; LayerCount:Integer):Double;
```

where PMaterial is a pointer to the array of Elastica's layer data.

6.2 Distributed Computation

TElasticaServer and TElasticaClient are derived from the classes TSimpleTCPClient and TSimpleTCPServer, obtainable free of charge on the internet. For distributing the computation, a couple of commands can be sent between them. Then, the event handler is executed and the user can process the request. A short discription follows:

```
type TElasticaServer = class(TSimpleTCPServer)
   ...
   procedure SendToClient      (ID:Integer; Command:Byte;
                                Info:LongInt; Data:PChar; Size:Integer);
   procedure BroadcastToClients(Command:Byte; Info:LongInt;
                                Data:PChar; Size:Integer);
   procedure StartSynchron;
   function WaitSynchron       (ms:longint):Boolean;
```

```
   function WaitForSingleResult(ms:longint):Boolean;
   ...
   property OnResult:TResultEvent;
   ...
end;
```

There are two ways to establish a connection between them: The server holds a list of client IP addresses and connect them directly, or, the other way around, each client knows the server IP address. Afterwards, the server sends command bytes and data packets the either one or all clients. They process the query and send the results back.

```
type TElasticaClient = class(TSimpleTCPClient)
   ...
   procedure SendToServer(Command:Byte; Info:LongInt; Data:PChar;
                          Size:Integer);
   ...
   property OnClientRequest:TClientEvent;
   ...
end;
```

The functions OnResult and OnClientRequest are the event handlers for data or result requests executed on any data event. The TElasticaClient components have a TElastica component internally. This can be used to process the queries. Finally, the SendToServer-procedure is used to hand back the results.

7 Conclusions

The purpose of this work was to find a strategy for optimizing layered thin film coatings with respect to their mechanical failure.

Therefore, the von Mises comparison stress and the tensile stress have to be considered theoretically within simulated, spherical indentation experiments.

By using the comercial simulation software ELASTICA, 2 and 3-layer systems have been extensively examined under particular load situations.

In order to determine the stress maximums exhibited within the single layers, established algorithms (Simplex, BFGS, and Golden Section Search) have been applied to 2-layer systems with layer thicknesses up to 2 μm each. Load states up to 1N have been investigated. Thereby, the 3-dimensional search was reduced to 2 dimensions, because the stress maximums were always located within the plane of the applied force. Since normal forces are acting into z-direction, and lateral forces into x-direction, the x-z-plane had to be studied only.

On the basis of a large number of examined load-material-combinations, the following general statements can be given:
In the case of normal load only, the von Mises stress was found directly under the indenter along the depth axis. The tensile stress was always located along the layer boundaries or under the indenter.

For additional lateral load, the stress maximums have to be searched within the entire faces into force direction. But mainly they are located at the boundaries of the search domains.

In both cases, a fast and convenient procedure has been developed, whereas cases of lateral force need noticably increased effort. Since the stress distributions are similar, 3-layer systems can be analyzed as well. Moreover, the method scales with the system size. The time needed for determining the maximum stresses are almost independent of the external parameters. However, thicknesses below 0.05 μm show unpredictable appearance and the method should not be applied. Using a Pentium IV computer, results can be obtained within 0.5 seconds for a 2-layer system, and about 7 seconds for 3-layer systems, what is even faster than ELASTICA plots the stress fields. But if also lateral load is applied, evaluation time raises up to 100 seconds for a 3-layer system.

Having access to the maximum stresses, the different compounds could then be compared for a particular load state, using a quality measure with regard to the critical stress limits. The quality measure was introduced and discussed comprehensively.

With respect to this quality measure, 2 optimization algorithms have been studied intensively. Thereby, the aforementioned evaluation time was very inconvenient. Out of pratical purposes, the optimization of the materials and thicknesses was chosen to be treated seperately. Using a massively parallel cluster, the state space of 2 and 3-layer systems was pre-sampled. The pre-cached data has been used as reference values for the developement of the optimization strategy. Additionally, graphs of the objective function for the whole state space could be plotted. In course of the Windows based

ELASTICA, it has been managed, for the first time, to run a Windows program on the Linux based supercluster CLiC.

The time necessary for a thickness optimization of a 3-layer system with a fixed material combination, using a Pentium IV computer, could finally be lowered to 2 hours. Such an automatic optimization has been realized for the first time.

To find proper material combinations, no formal optimization method is easily applicable. The compounds lack usable relations among each other. Therefore, an estimation value was proposed leading to a ranking of material combinations due to their properties only.

The significant correlation to the real quality obtained by calculation could be shown.

The method has been investigated on 2 and 3-layer systems with 12 and 6 materials, respectively. The eligiblity could even by shown for different substrates and load states.

In view of further investigations, the problem of ideal layer design offers a wide range of possibilities. A more comprehensive analysis of the occuring stress fields could further decrease the computational effort for finding the maximums. Moreover, varying thicknesses, loads and indenter could be used.

The thickness optimization has only be investigated within a small range. The extension by monetary aspects, for instance, is perfectly possible.

The highest potential is presumably contained within the search for optimal materials. More knowledge about the behavior of thin films could be involved leading to a more precise classification of the compounds. For instance, a parameter fit with the method of *least squares*, or, since the functional dependency should be analyzed in more detail, a *neural network* approach is proposed. Additionally, assumption over how the Young's moduli should be distributed among the layers could be imposed. Investigation had already been carried out by Schwarzer [21].

References

[1] T. Chudoba, ELASTICA, http://www.asmec.de

[2] O. Wänstrand, N. Schwarzer, T. Chudoba, Å. Kassman-Rudolphi, Load Carrying Capacity of Ni Plated Media in Spherical Indentation: Experimental and Theoretical Results, Surface Engineering 2002 Vol. 18 No. 2

[3] Gottstein, Physikalische Grundlagen der Materialkunde, 2. Aufl., Springer, 2001

[4] T. Chudoba, N. Schwarzer, F. Richter, Thin Solid Films 355-356 (1999) 284.

[5] T. Chudoba, N. Schwarzer, F. Richter, U. Beck, Thin Solid Films 377-378 (2000) 366.

[6] T. Chudoba, N. Schwarzer, F. Richter, Steps towards mechanical modeling of layered systems, Surf. Coat. Technol. 154 (2002) 140-151

[7] G.M. Hamilton, L.E. Goodman, ASME J. Appl. Mech. 33 (1966) 371.

[8] M.T. Hanson, ASME J. Tribol. 114 (1992) 606.

[9] M.T. Hanson, ASME J. Tribol. 115 (1993) 327.

[10] V.I. Fabrikant, Application of potential theory in mechanics: a Selection of New Results, Kluver Academic Publishers, The Netherlands, 1989.

[11] N. Schwarzer, ASME J. Tribol. 122 (2000) 672-681.

[12] E.R. Krl, K. Komvopoulos, D.B. Bogy, ASME J. Tribol. Trans. 118 (1996) 1.

[13] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Second Edition, Cambridge University Press, 1992

[14] Nocedal, J. and Wright, S. J., 1999, Numerical Optimization, Springer-Verlag, New York, Berlin, Heidelberg, Barcelona, Hong Kong, London, Milan, Paris, Singapore,Tokyo

[15] M. Flynn, IEEE Trans. Computers **21**, 948 (1972)

[16] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Inc., 1989

[17] http://www.mpi-forum.org

[18] http://www.winehq.org

[19] http://www.tu-chemnitz.de/urz/clic

[20] J. Eller, C. Sohrmann, Ein Windows-Programm auf dem CLiC, http://archiv.tu-chemnitz.de/pub/2003/0107/data/

[21] N. Schwarzer, Surf. Coat. Technol. 133-134 (2000) 397-402

[22] D. Roylance, Mechanics of Materials, John Wiley & Sons, Inc., 1996

[23] K.H. Hoffmann, M. Schreiber, Computational Physics, Springer, 1996

[24] Pons, English Learner's Dictionary, Klett, 1990

[25] Pons, Globalwörterbuch Deutch Englisch, Klett, 1990

Declaration of Independent Work

Hiermit versichern wir, dass wir die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel genutzt haben.

Chemnitz, den 12.09.2003

Christoph Sohrmann

Jens Eller