Journal of Machine Engineering, Vol. 17, No. 3, 2017

Received: *20 February 2017* / Accepted: *12 August 2017* / Published online: *28 September 2017*

characteristic diagram, adaptive structured grids,
thermo-elastic deformation, grid optimization,
thermal error estimation

Matthias PUTZ[1]
Steffen IHLENFELDT[2]
Christian NAUMANN[1*]
Janine GLÄNZEL[1]

OPTIMIZED GRID STRUCTURES FOR THE CHARACTERISTIC DIAGRAM BASED ESTIMATION OF THERMO-ELASTIC TOOL CENTER POINT DISPLACEMENTS IN MACHINE TOOLS

It is a well-known problem of milling machines, that waste heat from motors, friction effects on guides and also the milling process itself greatly affect the positioning accuracy and thus the production quality. An economic and energy-efficient method of correcting this thermo-elastic positioning error is to gather sensor data from the machine tool and the process and to use that information to predict and correct the resulting tool center point displacement using high dimensional characteristic diagrams. The size of these characteristic diagrams depends on the number of input variables (sensors) and the fineness of the discretization of the grid. While the number of sensors can usually not be reduced without affecting the quality of the prediction, it is often possible to minimize the size of characteristic diagrams through the use of adaptive grid refinement. This ensures that the finest grid sections correspond with the load cases that have the largest local gradients. Through such adaptive refinement, it is possible to reduce storage capacity and computation time without significant loss of precision. The aim of this article is to examine, test and compare different methods of adaptive grid refinement. For this, simulation data from a machine tool is used.

1. INTRODUCTION

Thermo-elastic machine tool deformation occurs during operation due to waste heat from motors and frictional heat from guides, joints and the cutting process, while coolants are used to reduce this influx of heat. Additional thermal influences come from the machine tool's environment and foundation. This leads to inhomogeneous, transient temperature fields within the machine tool which displace the tool center point (TCP) and thus reduce the production accuracy and finally the product quality. Active cooling can reduce this effect but does, however, require considerable amounts of energy. Another possibility is to stabilize temperature fields by allowing for lead-time or to limit the waste heat by using additional cooling, see [1].

[1] Fraunhofer Institute for Machine Tools and Forming Technology IWU, Chemnitz, Germany
[2] Dresden University of Technology TUD, Dresden, Germany
[*] E-mail: Christian.Naumann@iwu.fraunhofer.de

This creates a conflict of objectives, since increasing productivity while maintaining quality increases the energy demand (cooling) and thus the production costs. Conversely, a reduction of the energy consumption usually reduces product quality or productivity. The Collaborative Research Centers/Transregio 96 (CRC/TR 96), a German Research Foundation (DFG) project, attempts to shift this balance between productivity, energy efficiency and product quality by reducing the impact of waste heat on the TCP positioning accuracy while reducing or even eliminating the need for lead-time, see [2]. This is done, eg., by correcting the TCP position through the prediction of its thermo-elastic displacement based on data gathered from various sensors and from the machine tool control. One such algorithm is the characteristic diagram based correction, which uses high-dimensional characteristic diagrams to directly map discrete points of the temperature field and the current axis configuration onto the resulting TCP displacement. Paper [3] explains the theory of thermo-elastic correction and demonstrates its effectiveness on a demonstration machine.

Characteristic diagrams can theoretically be created with an arbitrary number of input variables but with each additional variable the dimension of the resulting problem increases exponentially. Therefore it is necessary to:

- limit the number of input variables,
- limit the discretization fineness of each variable,
- use efficient solvers (for the resulting linear systems),
- use efficient, problem-specific grid structuring.

Despite the power of current computers, increasing sizes of working memory and the use of graphics processing units for faster computation, solving very large linear systems is still challenging. More problematic still, is that the online-application of characteristic diagram based correction on a machine tool control places tight limits on the size of the characteristic diagram and the time available for calculating the correction values. That is why it is essential to make the characteristic diagram grids only as large as absolutely necessary. The number of input variables can be reduced by selecting only the most sensitive points of the temperature field as inputs. This can be done with a sensitivity analysis (SA), as described in [4]. The SA chooses the temperature inputs from the FE nodes of the model in such a way that the influence of the temperature of the next node on the TCP displacement is as large as possible while that node’s temperature is also mostly independent of that of all previously chosen nodes.

The necessary fineness of the discretization of each variable depends on both the nature of the physical dependency of that variable on the output (e.g. TCP displacement) and the grid structure used. Since this is therefore not an isolated problem, it will be discussed later in the sections dealing with grid structures and adaptive grid refinement.

Large characteristic diagrams are almost always sparsely filled because either the available data or the time to gather data through measurements or simulations is limited. Likewise the coefficient matrix of the resulting linear system will be sparse. Unfortunately this matrix fills rapidly if a direct solver is applied. This fill-in slows down computation and requires large amounts of memory. In addition, it is usually inefficient to fill a sparse matrix due to its compressed storage method. Therefore an efficient iterative solver would be preferable. In most cases, iterative solvers only perform well if the matrices are properly preconditioned [5]. Paper [6] compares different iterative solvers and explains how characteristic diagram based correction can be achieved with an FEM based approach (see Chapter 2) that allows the employment of efficient preconditioned multigrid solvers.

Grid structuring is a complex topic. There is rarely ever the one optimal grid for any given problem. Different methods of grid structuring, such as equidistant, variable, transformed, substructured and sparse grids will be described in chapter 3. Along with a description of their construction and optimization, some advantages and disadvantages will be discussed. As can be expected, the grid structures that allow the greatest adaptivity can achieve any given accuracy with the least number of grid vertices. Chapter 4 explains the use of adaptive grid refinement in the context of characteristic diagram based correction. For this, suitable choices of error estimators will be given, along with advice on the proper insertion of additional nodes into the grid. This will demonstrate that while substructured grids usually make the most efficient use of grid nodes, they are also the most complex in their implementation, optimization and usage. Another problem connected to adaptivity is the risk of overfitting. Chapter 6 therefore compares the different grid structures by evaluating their performance in a sample application. This application is a machine tool column with a moving spindle slide which is described in chapter 5. Thermo-elastic simulations of this column are used to generate training and test datasets for the evaluation of each grid structure.

2. CHARACTERISTIC DIAGRAM BASED CORRECTION

CHARACTERISTIC DIAGRAMS are continuous maps of a set of input variables onto a single output variable. They consist of a rectangular grid of support points along with kernel functions which describe the interpolation in between. Equation 1 shows a one-dimensional linear kernel $K_j(T_i)$ between two support points $\hat{T}_j$ and $\hat{T}_{j+1}$ which depends on a temperature T_i (compare Fig. 1)

$$K_j(T_i) = \frac{T_i - \hat{T}_j}{\hat{T}_{j+1} - \hat{T}_j} \tag{1}$$

Characteristic diagrams are created by first discretizing each input variable in order to establish the grid, then choosing a type of kernel function adequate for describing the local dependency of the input variables on the output variable and finally calculating the parameters of the kernel functions for each support point based on training data from simulations or experiments, see [7]. The most common form of characteristic diagram uses multilinear interpolation between support points, where a scalar factor r_j equal to the output value at the support point acts as a weight multiplied to the pyramid shaped kernel K_j.

$$f(\vec{T}_i) = \sum_j r_j \cdot K_j(\vec{T}_i), \quad \vec{T}_i := (T_{i,1}, \cdots, T_{i,n}) \tag{2}$$

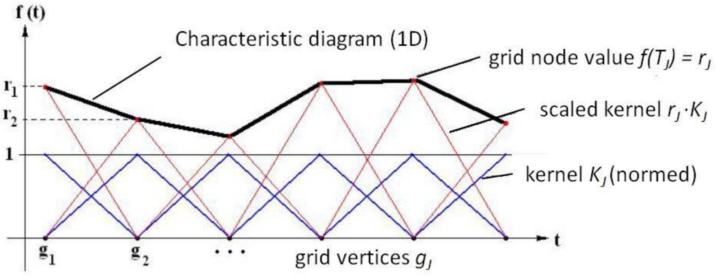
f (t)
Characteristic diagram (1D)
grid node value $f(T_J) = r_J$
r_1
r_2
1
scaled kernel $r_J \cdot K_J$
kernel K_J(normed)
g_1
g_2
. . .
grid vertices g_J
t

Fig. 1. 1D linear characteristic diagram interpolation

These higher dimensional kernels can be created by simply multiplying one-dimensional kernels. A more detailed account of characteristic diagrams and how they are calculated can be found in [8]. Here Priber describes a method called “Smoothed Grid Regression” (SGR) which adds smoothing terms to the data fitting terms described above in order to fit sparse or defective data. SGR is the basis for the characteristic diagram research presented here.

In the course of the CRC/TR 96 project, improvements to the characteristic diagram computation were achieved by employing multigrid solvers to efficiently solve the large sparse linear systems of high dimensional characteristic diagrams. This was done by developing a partial differential equation (PDE) that mimicked the common approach (s. eq. 1 and 2) and using it to derive the linear system via the finite element method (FEM). For piecewise multilinear kernels, this PDE in weak form reads:

$$\forall \overrightarrow{\delta u}: \quad \sum_{i=1}^{nData} (z_i - u(\vec{x}_i)) \cdot \delta u(\vec{x}_i) \quad + \quad \int_{\Omega} \lambda(\vec{x}) \cdot \nabla u(\vec{x}) \cdot \nabla \delta u(\vec{x})\, d\vec{x} \quad = \quad 0 \tag{3}$$

The equation comprises a data fitting term plus a smoothing term. Its derivation along with a more detailed explanation can be found in [2], under chapter “Correction Algorithms and High-Dimensional Characteristic Diagrams”. Where the common approach is limited to only 5 or 6 input variables due to limited memory, this new iterative solver can cope with 10 or more variables, as was shown in [6] using simulation data from a machine tool column. This development is of paramount importance when complex machine tool structures are to be modelled.

Many trials have shown that multilinear interpolation, given a fine enough grid, is best suited for the thermo-elastic problems in machine tools. Not only are they sufficiently accurate and easy to calculate, they also require comparatively few parameters and limit the risk of overfitting, see [6]. It was therefore used in the studies described in Chapter 6. Nevertheless other, more complex kernels such as B-splines or wavelets can also be combined with the various grid types described below.

3. METHODS OF GRID STRUCTURING

This chapter will introduce a number of different grid structuring techniques. Starting from simple equidistant grids, more complex grid structures will be derived until finally adaptive grids are discussed. All grid structures will be briefly described and evaluated in the context of their use in characteristic diagrams. A performance evaluation for thermo-elastic problems will be given in chapter 6.

3.1. EQUIDISTANT GRIDS

Equidistant grids present the simplest form of grid structures. Every grid axis is subdivided into equidistant intervals. The number of intervals in each dimension depends on the variability and the strength of the influence that the corresponding input variable has on the output variable. For many temperature variables describing the local expansion of a section of a machine tool such simple equidistant grids are, e.g., quite adequate. Their greatest advantage is their simplicity, which allows their implementation, calculation and real-time use with little effort. Equidistant grids are also very useful in combination with the FEM-based correction approach introduced in Chapter 2. By using 2^n intervals, it is easy to create the hierarchical grid needed for the multigrid solver. Equidistant grids are the largest grids in terms of the number of grid points and will therefore be used here as reference to measure the other sparser grid structures' performance.

3.2. GRIDS WITH VARIABLE INTERVALS

As with equidistant grids, all grid subdivisions are done on the individual axes. The difference is that now all intervals can vary in size. This allows greater variability depending on the states of individual input variables. An example could be the dependency of the vertical position of the spindle on the TCP displacement which is almost zero at the bottom and increases superlinearly towards the top. If done properly, variable grids will usually achieve better results than equidistant ones, even with the same number of grid points. The problem is that it is usually difficult to find the optimal grid intervals. In addition, the use of variably spaced grids requires a bit more effort because for any new measured data point the containing grid hypercube must be searched for.

Literature studies have not yet revealed an algorithm that will reliably compute the optimal variable grid for a given database. A simple iterative algorithm could work as follows:

1. Start with a coarse equidistant grid.
2. Increase the fineness of the equidistant grid until the approximation quality suffices (error < ε) or the computation effort exceeds a certain limit.
3. Shift the grid axis interval boundaries until no further improvement of the approximation quality greater than δ is possible. To do so, repeat the following steps:
   - Compute the maximum (or average) approximation error for each grid axis interval,
   - Find the two grid axis intervals that have the greatest difference in their respective approximation error,

   - Move the boundary between the two intervals towards the one with larger error by γ % of its interval length.
4. Is the solution of step 3 significantly worse than the previous solution?
   - YES: BREAK. Keep the previous solution and its corresponding grid.
   - NO: Compute the maximum (or average) regularisation (smoothness) error for each grid axis vertex. Remove the grid axis vertex with the smallest regularisation error. GO TO 3.

This algorithm requires computing a new characteristic diagram for each new grid structure. Even though this process is done offline and does therefore not affect the online correction, it can become very time consuming for large grids. Nevertheless, it has yielded good results in the computational tests described in chapter 6.

3.3. TRANSFORMED GRIDS

One very effective way of transforming a rectangular grid is through principal component analysis (PCA). It minimizes dependencies between the input variables by rotating the grids. A detailed description of the PCA can be found in [9]. Characteristic diagram grids are large hypercuboids created from the discretization of input variables. In practical applications it is often the case that some of the corners of this hypercube correspond to system states that are not relevant. For two neighbouring temperature sensors on a machine tool, e.g., one will never be very cold while the other is very hot. Rotating the grid using PCA allows cutting off these corners in order to preserve the limited number grid vertices for the grid sections that truly matter.

PCA computes the desired rotation (coefficient matrix) directly from the input data matrix by solving a small eigenvalue problem of the size of the input dimension. After the rotation has been applied to the input data, the grid is created with any of the structures listed in this chapter. The PCA also returns an indicator of the significance of each grid axis after the transformation. If this indicator is very low, hinting at a strong linear dependence in the input variables, then the grid dimension with the lowest significance can be omitted without affecting the characteristic diagram quality.

PCA can therefore be viewed as pre-processing step. It is a simple way of reducing the grid size with very little effort and without losing the advantages that the other grid structures offer. The only disadvantage is that it makes the interpretation of the characteristic diagram more difficult. This is, however, usually a moot point because the interpretation of any high-dimensional characteristic diagram is nearly impossible anyway.

3.4. GRIDS WITH SUBSTRUCTURES

Substructures can be added to any of the above grid structures. This is done by choosing a grid hypercube, usually one with a large approximation error, and subdividing it into smaller hypercuboids. This subdivision can once again be done with equidistant or variable intervals (see sections above). The most common form of substructure is created by splitting the hypercuboids in half, in some or all dimensions. If some of these new, smaller hypercuboids still produce a large error, they can be divided once more. This process is usually repeated until the local fineness of the grid matches the local variability of the function that is to be approximated. Such a sequence of bisections is often referred to as an octree, derived from the tree structures commonly used to construct them.

An example of when such substructures are useful can also be derived from the sample application in chapter 5. The temperature sensors at the top of the machine tool column have a large impact on the TCP displacement, but only if the spindle is in the upper half of the column. A fine discretization of these temperature variables in the lower spindle positions would therefore be a waste of precious grid vertices. This form of adaptivity can only be achieved with the substructures described in this section or the sparse grids from the following section.

The main advantage of grid substructures is the highly localized grid fineness which can be precisely tailored to the given application. The approximation error based refinement is simple, effective and arrives at the optimal grid in only a few iterations. The cost of this is the increasing complexity of the characteristic diagram computation and usage. One such complication is that an effective method of grid storage is required. Such a method needs to be easy to use in the implementation of the characteristic diagram method, it needs to allow easy determination of the grid hypercube belonging to a given data point and it needs to permit efficient solving of the resulting linear system. The latter requirement is due to fact that the order in which grid vertices are indexed has a great influence on the convergence behaviour of iterative solvers. Another complication is the fact that substructures lead to irregular triangulations which require a slightly different interpolation approach. In paper [10] an octree structure is used to solve PDEs on hypersurfaces with local refinement. It also demonstrates that optimal convergence can be achieved for irregular triangulations.

3.5. SPARSE (HIERARCHICAL) GRIDS

Sparse grids are derived from hierarchical grid structures. Hierarchical grids consist of overlapping grid layers where each subsequent layer is finer than the previous one. Figure 2 shows a complete hierarchical grid in 2D which is created via the tensor product of hierarchical subspaces W_{ij} as described in [11]. The number of grid points for such a complete grid is $(2^n - 1)^d$, with d the dimension and n the level of the grid. The grid in Fig. 2 thus contains $(2^3 - 1)^2 = 7^2 = 49$ grid points.

Sparse grids are based on the same principle, but they push certain subspaces onto higher grid levels. Compared to a full hierarchical grid of the same level, these subspaces are omitted and thereby the density of the grid reduced. As before, the selection of which subspaces are added on each subsequent level is determined by the approximation error. An example of such a sparse grid can be seen in Fig. 3, where the grid level is once again 3, but the number of grid points is now only

$$\sum_{i=1}^{n} 2^i \cdot \binom{d-1+i}{d-1} = 17.$$

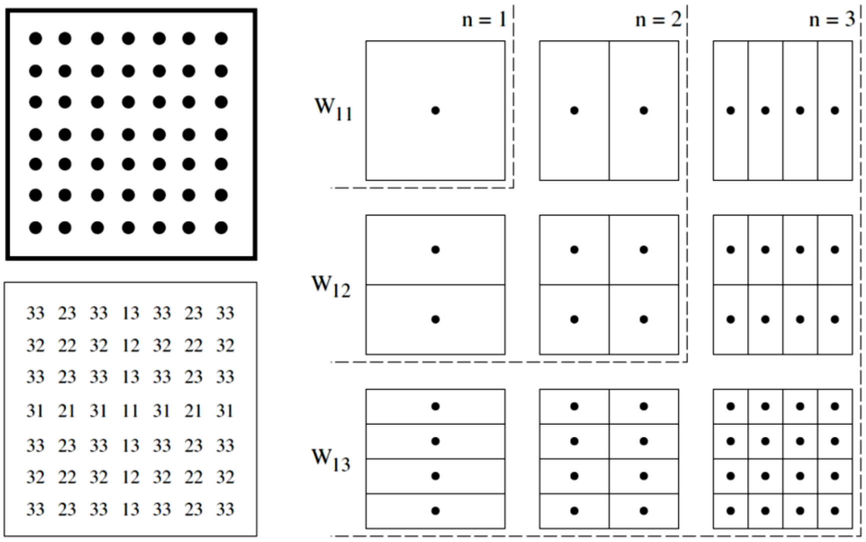

Fig. 2. Complete hierarchical grid in 2D [11]

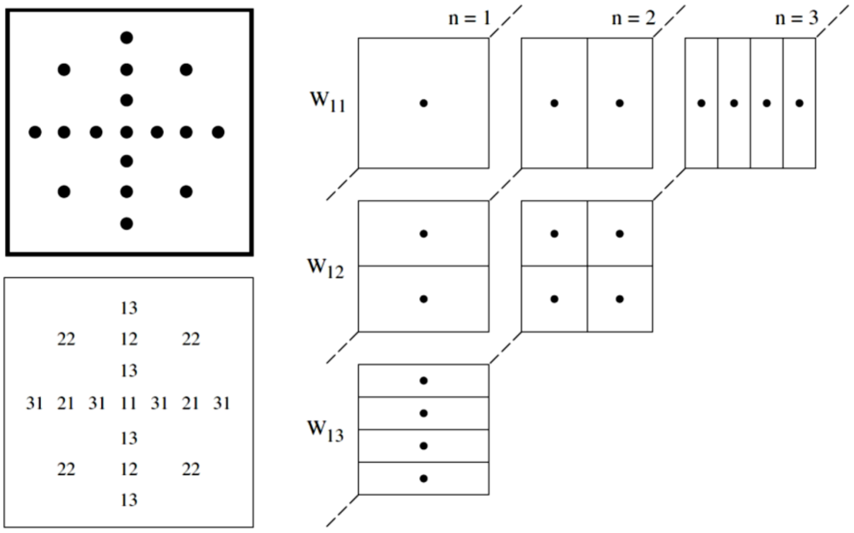

Fig. 3. Sample sparse hierarchical grid in 2D [11]

For a large number of problems, increasing the precision of the solution requires higher grid levels. Since increasing grid levels cause an exponential increase in the number of grid points, the complexity of the problem rises accordingly. The purpose of sparse grids is to limit the number of grid points added with each refinement while still obtaining the desired improvement of the solution. Paper [12] demonstrates how this reduction in complexity can be achieved without significant loss of precision for a number of sample applications.

As with the grid substructures, sparse grids can be precisely adapted to the local complexity of a given approximation problem and here too does this come at the cost of more complicated algorithms for the implementation, calculation and utilization of the resulting characteristic diagrams.

4. ADAPTIVE GRID REFINEMENT

Adaptive grid refinement can be a useful tool to either increase the accuracy of an approximation or reduce the problem size for a given approximation error. The way it is conducted depends heavily on the type of grid used and the problem to be solved. The most important issues to resolve in order to use adaptive grid refinement are:

- defining an error estimator,
- determining how the grid elements are subdivided,
- creating an efficient way to store and access the grid nodes and their values,
- adjusting assembler and solver to cope with new grid elements and nodes.

Defining an error estimator for smooth characteristic diagrams (SGR) is relatively straight forward. The SGR is created by minimizing the data approximation error and the smoothness error at the same time [7]. The adaptive grid refinement can therefore use either error or both to decide which grid elements to refine. In this investigation a combination of both data error and smoothness error has yielded the best results. As soon as the current solution has been calculated, the error is estimated for each element (or node) and then all nodes with an error above a predetermined threshold are refined.

Determining how to subdivide existing elements of a grid depends largely on the grid element type. Many three-dimensional FE grids use tetrahedral elements which are very easy to split. One only needs to place a new node inside an existing tetrahedron and connect this new node to all of its vertices to get four new tetrahedra. For the rectangular grids used for characteristic diagrams, the division is more complicated. The easiest way is to bisect the hypercuboid in each dimension, e.g. in 2D a rectangle becomes four new, equally large rectangles. Splitting the hypercuboid anywhere other than in the middle, splitting it into more than two parts or splitting it in only some dimensions would force the neighbouring elements to share that method if they too are split and is therefore generally avoided.

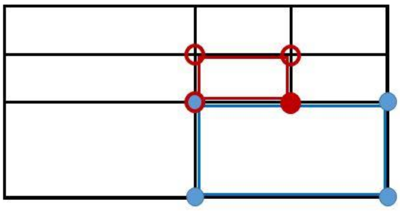

Fig. 4. Simple 2D adaptive grid with hanging node (red dot)

Figure 4 shows a simple 2D grid with a local refinement and already hints at the problem it creates. The newly refined grid is now a so called irregular triangulation because it contains a “hanging node” (red dot) that touches its neighbouring element (blue rectangle) on an edge rather than a vertex. There are two easy ways to resolve this. The first is to continue the bisection in each direction until the limits of the grid are reached. This method is simple and allows refining grid elements in selected dimensions. It is, however, impractical if the grid has a high dimension and a large number of grid vertices on each grid axis, because then each subdivision creates a lot of new grid elements of which most have little impact on the error. The second and better solution is to add restrictions to the hanging nodes. For multilinear interpolation one can add the requirement for the intersecting node to have half the value of the sum of the nodes whose edge it intersects:

r🔴 = ½ r◉ + ½ r🔵

Efficient data structures for characteristic diagrams are generally relatively simple to come by, due to their rectangular grids. Equidistant grids, e.g., only need the coordinates of one node, and both the grid interval length and the number of intervals in each dimension. Variable grids only need one grid axis vector per dimension to fully describe the grid. For grids with local refinement it is somewhat more complicated. Paper [13] describes an efficient data structure for adaptive triangular grids. It uses just the node and element matrices for the mesh and creates any auxiliary structures as they become necessary. For this investigation recursive data structures analogous to binary trees were used. Paper [11] uses similar structures with sparse grids.

Adjusting the assembler and solver to adaptive grids can likewise be challenging. This is mainly due to the hanging nodes described above. For the SGR method, as described in [7] and [8], the adjustment to the assembler simply requires adding one new data equation for each new grid point added through refinement. Refining a hypercuboid will create one new degree of freedom in the middle plus additional free vertices if neighboring vertices are also refined or the hypercube is on an outer hypersurface of the grid. Since this distinctions can be difficult to make for high-dimensional grids, an alternative is to create as many new vertices as possible and restrict those bound by coarser neighboring elements using additional smoothing equations with large weights, see [7]. Many direct and some iterative solvers will then solve the resulting linear system like that of an unrefined grid. The FEM based approach described in [6] becomes more complicated if hanging nodes are introduced. Paper [14] explains how hanging nodes can be handled for quadrilateral meshes and examines the resulting convergence. If adaptive refinement is used in combination with a multigrid solver, then the restriction operator will simply undo the last refinement steps to obtain its approximation for the solution on the fine grid, see [6].

Note that this section only described the so called “h-refinement” which uses grid element splitting to achieve adaptivity. There is also “r-refinement” which moves grid vertices and “p-refinement” which alters the kernel functions (e.g. polynomials of higher degree). Both of these refinement strategies can also be applied to the SGR method. The variable grids described in chapter 3b can, e.g., be considered an r-refinement. Previous investigations into the usage of polynomial kernel functions in the SGR have shown that multilinear kernels (usually with a slightly finer grid) are superior to higher-degree polynomial kernels because they are simpler to use and less prone to overfitting in large sparsely filled grids.

5. SAMPLE APPLICATION FOR GRID EVALUATION

The chosen sample application for the evaluation of the described grid structures is a simulation model of a machine tool column including the slide that would hold the spindle in a horizontal machining center. The thermo-elastic deformation of this assembly has a large impact on the TCP displacement. Figure 5 (left) shows a CAD-model of column and slide. In order to efficiently model the movement of slide on column and the moving frictional heat source it represents, the column was subdivided into 15 segments so that the slide is always in contact with at least 3 of the segments. This results in 11 discrete vertical positions between which linear interpolation can be applied.

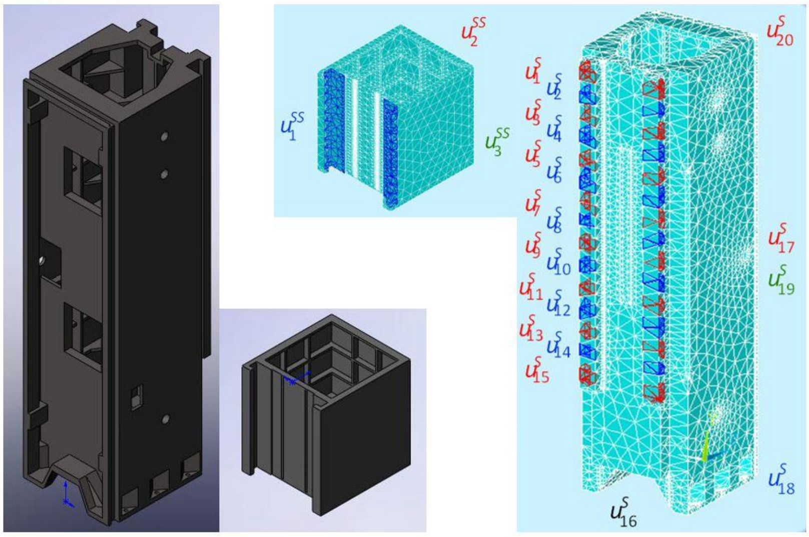

Fig. 5. CAD- and FE-model of machine tool column and spindle slide [15]

Figure 5 (right) shows the vertical segments and the FE-mesh of the components. The variables u_i therein indicate the locations of heat transfer. In addition, the model considers free convection and a vertical gradient of the surrounding air temperature. Forced convection which can have a significant effect for fast slide movements was neglected because it would require CFD simulations to determine the transient heat transfer coefficients.

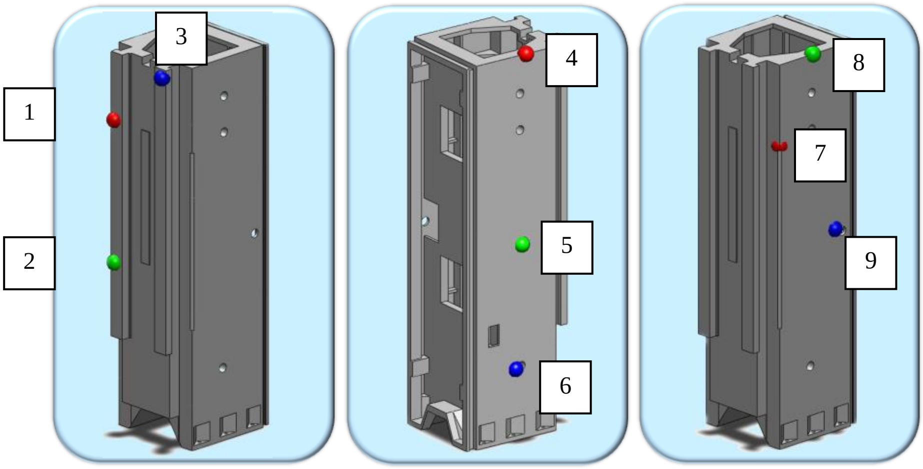

Fig. 6. Locations of virtual temperature sensors [15]

In order to ensure swift and accurate calculation of the temperature fields for the moving heat source, model order reduction was applied, as described in [13]. While the simplifications made in this scenario might limit the accuracy of the simulated temperature fields, they do not significantly affect the resulting characteristic diagrams. This is owed to the fact that these characteristic diagrams only imitate the mapping of temperature fields onto their respective deformations, which only depends on the correctness of the CAD model and its material parameters.

Figure 6 shows the locations of the virtual temperature sensors used to create the characteristic diagrams tested in the next chapter. These positions were not optimized by sensitivity analysis but rather chosen with the intention of creating diversity.

6. COMPARISON OF GRID STRUCTURING TECHNIQUES

Chapter 3 lists equidistant, variably spaced, transformed, substructured and sparse grids as the most common grid types for characteristic diagrams. Of these, substructured grids offer the greatest level of adaptivity and have therefore got the greatest potential for producing minimal grids. Sparse grids are basically just another kind of substructured grid and therefore not separately investigated. One problem with substructured grids is that they are far more complicated in their implementation, usage and optimization than many of the simpler grid structures. Most of this effort must, however, only be done once before adaptive grid structures are readily usable. More problematic is the fact that any local adaptivity risks overfitting. The risk of overfitting is especially great if:

- the training data contains errors,
- the training data is too sparse,
- the grid has too few input variables to fully represent the output,
- the kernels are fast-growing functions (e.g. higher-order polynomials / B-splines),
- the smoothing equations’ weights are too small compared to the data fitting weights.

Overfitting leads to larger errors in the estimation and can therefore make larger grids less effective than smaller ones. To summarize, more rigid grid structures, like equidistant grids, require more grid vertices but are less vulnerable to overfitting, while adaptive grid structures are much smaller at the cost of a greater risk of overfitting.

Using the simulations described in chapter 5, the various grid structures will now be evaluated and compared in order to gage their suitability for thermo-elastic problems. The simulation moved the spindle slide up and down with random motion profiles and a smoothly fluctuating heat source of up to 200W. For each test, one long sequence of heating and cooling cycles was simulated to generate data to train the characteristic diagrams and then another, shorter sequence was simulated to test them. In order to keep the time required for the grid optimization and the testing of different grid sizes manageable, only the first 4 temperature inputs were used for the grids and the maximum grid sizes were kept relatively small. An example of such a simulation for testing an equidistant characteristic diagram is shown in Fig. 7. The sequence of training and testing characteristic diagrams with different grids was done with five different simulations for each grid type in order to create some statistical reliability.

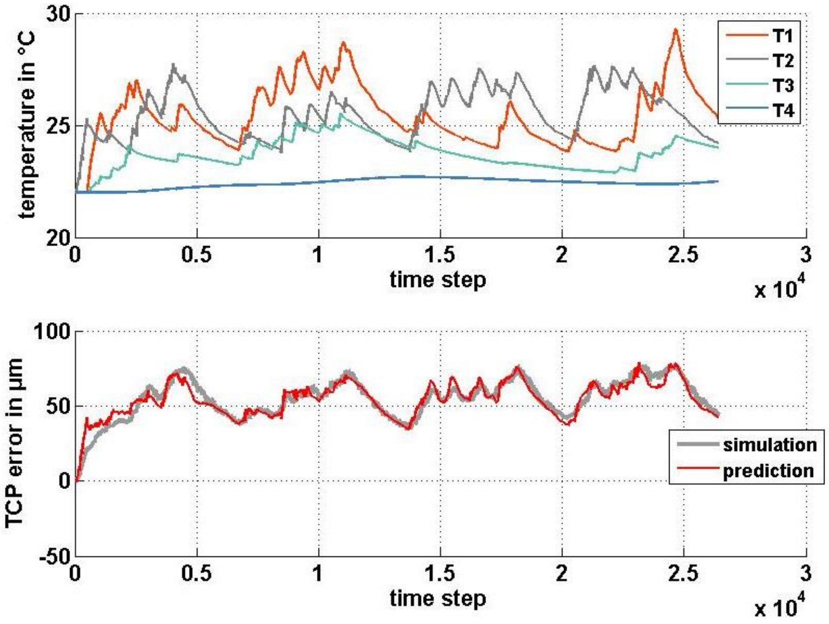

Fig. 7. Sample simulation data for characteristic diagram test with a corresponding estimation

For every tested grid (and thus every step of the grid optimization) the training data was randomly split into three equal sets, of which two were used for training and one for testing. For all three combinations of test and training sets, estimations were computed and their errors averaged. Depending on whether the recently tested grid performed better than the previous one, grid optimization (refinement) would continue. Finally, the data from the training simulation was used to test the last, finest grid after optimization was complete. The results of these grid evaluations are shown in Fig. 8.

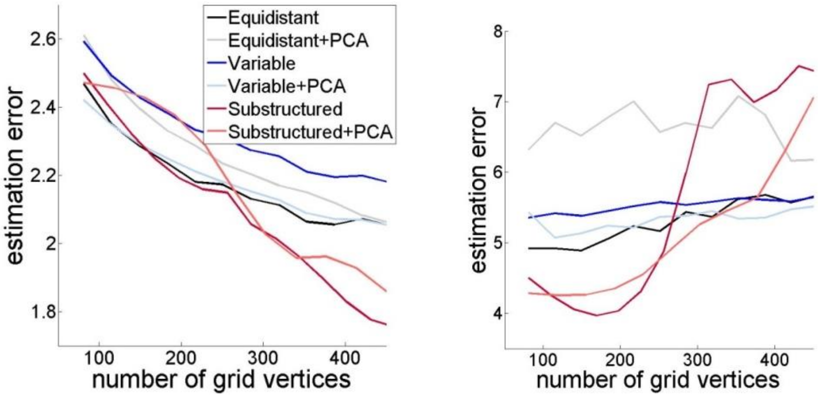

Fig. 8. Comparison of estimation error of thermal TCP displacement for training data (left) and test data (right)

The left chart shows the average absolute estimation error of the training data and the right chart that of the test data. The average absolute estimation error works better as an indicator of the grid quality than the maximum absolute estimation error because it is less susceptible to stochastic errors and thus enables a fairer comparison of the grid structures.

In this investigation, the estimation of training data can be compared to estimating error-free data which densely covers the grid structure. Under these circumstances overfitting is not an issue and thus adaptive grids are at an advantage. The estimation of the test data compares to the more realistic scenario where the data is limited and contains errors. Here, overfitting is a great problem so that finer grids are not always better than coarse ones and rigid grid structures often perform better than adaptive ones.
The following conclusions can be drawn from the charts in Fig. 8:

- PCA shows no significant improvement over untransformed grids,
- Substructured grids are slightly better at estimating training data,
- Substructured grids are more susceptible to overfitting test data,
- Coarse grids are most often better at fitting test data (overfitting),
- Equidistant and variably spaced grids perform similarly well,
- Four temperature sensors can already reduce the average thermal error from over 50µm to less than 5µm.

The most important lessons to be learned from these simulations is that the simplest grids often produce the best results and that grid optimization in the manner described in chapter 3 should not be done with limited or erroneous training data.

7. CONCLUSION

The paper builds on previous research on the characteristic diagram based correction of thermo-elastic tool center point displacements. After outlining the problem of thermo-elastic deformations in machine tools, the state of the art in characteristic diagram based correction is presented. Following this, equidistant, variably spaced, transformed, substructured and sparse grid structures are introduced and the implications of their usage in characteristic diagram based correction are discussed. The problems arising with the usage of adaptive grid refinement, such as irregular triangulations and efficient memory structures are discussed and solutions presented. A simulation model of a machine tool column with a moveable spindle slide is presented as a sample application to demonstrate and test the effectiveness of each of the aforementioned grids. The performed simulations have shown that adaptive grids are very efficient under ideal conditions but susceptible to overfitting if the training data is limited or erroneous. In these cases coarse, rigid grid structures are most reliable. Grid transformations using principal component analysis has no significant effect on the estimation quality. Across all grid types, characteristic diagram based estimation of thermal displacements has yielded good results with only four temperature inputs.

ACKNOWLEDGMENTS

This research was supported by a German Research Foundation (DFG) grant within the Collaborative Research Centers/Transregio 96, which is gratefully acknowledged.

REFERENCES

[1] BRYAN J., 1990, *International status of thermal error research*, CIRP Annals – Manufacturing Technology, 39/2, 645-656.

[2] GROSSMANN K. et al., 2015, *Thermo-energetic design of machine tools*, Lecture Notes in Production Engineering, Springer.

[3] IHLENFELDT S., KAUSCHINGER B., NAUMANN C., PUTZ M., THIEM X., RIEDEL M., 2016, *Implementation and demonstration of characteristic diagram as well as structure model based correction of thermo-elastic tool center point displacements*, Journal of Machine Engineering, 16/3, 88-101.

[4] HERZOG R., RIEDEL I., 2015, *Sequentially optimal sensor placement in thermo-elastic models for real time applications*, Optimization and Engineering, 1-30.

[5] SAAD Y., 2003, *Iterative methods for sparse linear systems*, 2nd Edition, Society for Industrial and Applied Mathematics.

[6] IHLENFELDT S., NAUMANN C., PRIBER U., RIEDEL I., 2015, *Characteristic diagram based correction algorithms for the thermo-elastic deformation of machine tools*, Proceedings 48th CIRP Conference on Manufacturing Systems.

[7] NAUMANN C., PRIBER U., 2012, *Modellierung des Thermo-Elastischen Verhaltens von Werkzeugmaschinen mittels Hochdimensionaler Kennfelder*, Proceedings Workshop Computational Intelligence.

[8] PRIBER U., 2003, *Smoothed grid regression*, Proceedings Workshop Fuzzy Systems.

[9] CHATFIELD C., COLLINS A.J., 1980, *Introduction to multivariate analysis*, Springer US, 57-81.

[10] CHERNYSHENKO A.Y., OLSHANSKII M.A., 2014, *An adaptive octree finite element method for PDEs on surfaces*, arXiv,1408.3891v1.

[11] BURGANTZ H.J., GRIEBEL M., 2004, *Sparse Grids*, Acta Numerica 13 147-269.

[12] PFLUEGER D., 2010, *Spatially adaptive sparse grids for high-dimensional problems*, Dissertation, TU Munich.

[13] CHEN L., ZHANG C., 2010, *A Coarsening algorithm on adaptive grids by newest vertex bisection and its applications*, IEEE Journal of Computational Mathematics, 28, 1-23.

[14] MAO S., SHI Z.-C., ZHAO X., 2010, *Adaptive quadrilateral and hexahedral finite element methods with hanging nodes and convergence analysis*, In: Journal of Computational Mathematics, 28/5, 621-644.

[15] BENNER P., LANG N., SAAK J., 2015, *Model order reduction for thermo-elastic assembly group models*, In: Thermo-Energetic Design of Machine Tools, Lecture Notes in Production Engineering, Springer, 85-93.