Adaptive Free-form Deformation for the Modification of CAD/CAM Data

Alexei Sacharow, Tobias Surmann, Dirk Biermann
Institute of Machining Technology
Technische Universitaet Dortmund
Dortmund, Germany
sacharow@isf.de, surmann@isf.de, biermann@isf.de

***Abstract*—In production engineering, the process of reverse engineering often requires modifications of CAD/CAM data. In general, CAD surfaces are modified according to a set of discrete displacement vectors. For this purpose, smoothing space-deformation techniques like free-form deformation (FFD) can be used. We present a B-spline-based adaptive FFD, which is able to ensure a user-defined shape accuracy. In an iterative process, the control-point lattice of the B-spline volume is automatically refined so that the approximation errors resulting from the direct free-form deformation decrease. Therefore, the areas inside the volume with the highest deformation are identified and subsequently refined by inserting new knots into the B-spline volume. Numerical studies have shown that the presented method improves the common B-spline-based FFD technique with respect to accuracy and efficiency.**

***Index Terms*—CAD; free-form deformation; reverse engineering**

I. Introduction

There are several applications in production engineering, in which the geometry of a designed part has to be modified. First of all, during product development, modifications of the product are often required in order to optimize or change the product properties. This process is generally called reverse engineering [1], or reengineering; it begins with a prototype being manufactured with respect to the designed CAD model (CAD - Computer-Aided Design), followed by manual modifications by the engineer. To incorporate these modifications, first, the manufactured workpiece is measured by an optical or tactile scanning device. Then, the digitized data is compared to the designed CAD model by a registration process [2]. Normally, the outcome of the registration is a discrete displacement field, which is used for the following modification of CAD/CAM data (CAM - Computer-Aided Manufacturing). Another important application is the compensation of springback in sheet metal forming. Here, the geometry of forming tools has to be modified with respect to a modification field [3], which is obtained by a finite element simulation of the forming process or by a registration of the digitized data.

Free-Form Deformation (FFD) is a technique for space deformation and is used for the modification of 3D objects [4] in computer graphics and geometric modeling. Due to its continuity and flexibility, this method was already applied for industrial applications, e.g., rapid manufacturing [5] or the compensation of form errors by modifying CAD/CAM data [6][7]. However, approximation errors always occur, thus, the desired shape accuracy cannot be ensured. In this paper, we present an adaptive FFD method, which allows the modification of 3D shapes with regard to a specified shape accuracy.

This paper is organized as follows. In Section II, a short overview of free-form deformation techniques is provided. Section III presents the new approach for the adaptive free-form deformation using B-splines. The presented method is validated in Section IV using two examples from sheet metal forming. Finally, a conclusion is given in Section V.

II. Related work

A first approach for free-form deformation was introduced by Barr [4]. He used differential affine transformations for regular global deformations, like scaling, tapering, bending, or twisting. Additionally, rules for the transformation of tangent and normal vectors were developed. Sederberg and Parry [8] presented a more general approach for spatial deformation. They defined the deformation function as a trivariate Bernstein polynomial tensor product (Bézier volume). The FFD volume is represented by a parallelepiped lattice of control points, and the space deformation is realized by moving the control points. This FFD technique proceeds as follows:

1) Define a lattice of control points, which encloses the object to deform
2) Calculate the local parameters of every point describing the embedded object
3) Deform the FFD volume by moving the control points
4) Displace the embedded object points.

Based on the work of Sederberg and Parry, Coquillart [9] developed an Extended Free-Form Deformation (EFFD) using arbitrarily shaped, non-parallelepipedical lattices. This method uses the Bézier technique, in which the lattice size depends on the degree of Bernstein polynomials, thus, only global deformations are possible. Extensions to B-splines [10] or NURBS (Non Uniform Rational B-spline) [11] increase the flexibility of FFD and provide the possibility for local deformations. Hsu et al. [12] developed a method for the direct manipulation of FFD, which allows to control the free-form deformation by displacing object points directly. They compute the repositioning of the control points in a sense of

 ISBN: 978-1-61208-172-4

least squares using the pseudo-inverse matrix. Hu et al. [13] introduced an explicit and more efficient solution for the direct manipulation problem. Sarraga [6] adopted the FFD-method for modifying CAD/CAM surfaces according to displacements prescribed at a finite set of points.

Biermann et al. [7] presented an approach for manufacturing modified workpieces, in which a B-spline volume is used for the direct deformation of original NC programs (NC - Numerical Control). The requirements for the FFD were accuracy and efficiency by deforming CAD/CAM data, which consist of several thousand points. Here, the accuracy increases with the number of control points, but it is still hard to ensure that the approximation error is within the required tolerance.

III. Adaptive FFD

In this section, an adaptive FFD method is presented that is able to ensure a user-defined shape accuracy by deforming an object in the described manner. The data at hand is a shape (e.g., mesh or CAD model of a workpiece) and a finite set of displacement vectors according to the desired shape modification. For this purpose, Biermann et al. [7] used a B-spline volume with a regular lattice of a fixed size. The innovation of the presented FFD method, which also uses a B-spline volume, is the iterative refinement of the lattice and recomputation of the control point positions so that the approximation error decreases in each iteration.

The adaptive FFD proceeds as follows:

1) Initialize a B-spline volume with a regular, paraxial lattice of minimal size
2) Compute the deformation according to the displacement vectors
3) Deform the object and update the displacement vectors
4) If the shape deviation (e.g., least squares distance between source and target shape) is not within the tolerance, refine the lattice and go to step 2.

In the following, the individual steps are explained in more detail.

A. Definition and initialization

The B-spline volume $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ of degree p, q, and r is defined by:

$$F(u,v,w) = \sum_{i,j,k=1}^{l,m,n} B_{i,p}(u) B_{j,q}(v) B_{k,r}(w) \boldsymbol{P}_{i,j,k}, \quad (1)$$

where $\boldsymbol{P}_{i,j,k}$ are the control points on a ($l \times m \times n$)-lattice, (u, v, w) are the local parameters of a point inside the B-spline volume, and $B_{i,p}(u)$, $B_{j,q}(v)$, and $B_{k,r}(w)$ are the nonrational B-spline basis functions defined on the knot vectors

$$U = \{\underbrace{u_{min}, \ldots, u_{min}}_{p+1}, u_{p+1}, \ldots, u_l, \underbrace{u_{max}, \ldots, u_{max}}_{p+1}\},$$
$$V = \{\underbrace{v_{min}, \ldots, v_{min}}_{q+1}, v_{q+1}, \ldots, v_m, \underbrace{v_{max}, \ldots, v_{max}}_{q+1}\},$$
$$W = \{\underbrace{w_{min}, \ldots, w_{min}}_{r+1}, w_{r+1}, \ldots, w_n, \underbrace{w_{max}, \ldots, w_{max}}_{r+1}\}.$$

In general, the parameter space and, therefore, the knot vectors are normalized to $[0, 1]$. This requires the embedding of the objects, where for every point $\boldsymbol{p}_i = (x_i, y_i, z_i)$ the local parameters (u_i, v_i, w_i) have to be found. In case of a Bézier volume, this can be easily done by solving linear equations [8]. In case of a B-spline or NURBS, due to the multiplicity of outer knots, the local parameters have to be found by numerical search [14]. Regarding the deformation of complex and large objects, this operation may be very costly. To overcome this problem, we define the initial B-spline volume as the identity so that the embedding operation vanishes. The initial B-spline volume has a regular, paraxial lattice of minimal size. This means $l = p + 1$, $m = q + 1$, and $n = r + 1$. For the normalized parameter space, the B-spline volume is equivalent to a Bézier volume, and for the embedding function $E : \mathbb{R}^3 \rightarrow [0, 1] \times [0, 1] \times [0, 1]$ yields:

$$E(x,y,z) = \left(\frac{x - x_1}{x_l - x_1}, \frac{y - y_1}{y_m - y_1}, \frac{z - z_1}{z_n - z_1} \right), \quad (2)$$

where $\boldsymbol{P}_{1,1,1} = (x_1, y_1, z_1)$ and $\boldsymbol{P}_{l,m,n} = (x_l, y_m, z_n)$. By reparameterizing the parameter space to $[x_1, x_l] \times [y_1, y_m] \times [z_1, z_n]$, the embedding function becomes identity: $E(x, y, z) = (x, y, z)$. This means that the local parameters of a point are equal to its coordinates.

B. Computing the deformation

The direct manipulation of an FFD volume requires the computation of control point displacements so that the distance between the deformed source point $\boldsymbol{p}_i$ and corresponding target point $\boldsymbol{q}_i$ is minimized for every pair $(\boldsymbol{p}_i, \boldsymbol{q}_i), 0 < i \leq d$, where d is the number of displacement vectors. This can be formulated as a least-squares problem:

$$R = \sum_{i=1}^{d} \| \boldsymbol{q}_i - F(\boldsymbol{p}_i) \|_2^2 \rightarrow \min . \quad (3)$$

The deformation of a point $\boldsymbol{p} = (x, y, z)$ is given by:

$$\begin{aligned} F(\boldsymbol{p}) &= F(x,y,z) \\ &= \sum_{i,j,k=1}^{l,m,n} B_{i,j,k}(x,y,z)(\boldsymbol{P}_{i,j,k} + \boldsymbol{\delta}_{i,j,k}) \\ &= \boldsymbol{p} + \sum_{i,j,k=1}^{l,m,n} B_{i,j,k}(x,y,z) \boldsymbol{\delta}_{i,j,k}, \end{aligned} \quad (4)$$

where $\boldsymbol{\delta}_{i,j,k}$ are the displacement vectors of the control points and $B_{i,j,k}(x, y, z) = B_{i,p}(x)B_{j,q}(y)B_{k,r}(z)$. The minimization problem (3) can be formulated as a set of linear equations and solved in a sense of least squares using the Singular Value Decomposition (SVD) [7].

In order to allow an iterative recalculation of control point displacements, equation (4) can be defined recursively:

$$F_s(\boldsymbol{p}) = F_{s-1}(\boldsymbol{p}) + \sum_{i,j,k=1}^{l,m,n} B_{i,j,k}(x,y,z) \boldsymbol{\delta}^s_{i,j,k}, \quad (5)$$

where $F_0(\boldsymbol{p}) = \boldsymbol{p}$ and $\boldsymbol{\delta}^s_{i,j,k}$ are the displacements of the control points in the iteration s, which updates the positions of the control points from the previous iteration. By considering only the control points which were involved in the refinement of the lattice in the previous iteration, the complexity of the least-squares problem (3) decreases and, thus, the computing time is reduced significantly. In doing so, the FFD volume is optimized locally in each iteration and the shape deviations decrease continuously.

C. Refinement of the lattice

In order to reduce the approximation errors, the flexibility of the FFD volume has to be increased. One well-known technique for increasing the degrees of freedom and thereby the flexibility of B-splines is knot insertion [14]. In case of a B-spline curves, adding a new knot does not change the shape of the curve. Therefore, inserting a new knot cause addition of a new control point and the displacing of some existing control points. In case of a B-spline volume with an $(l \times m \times n)$-lattice, e.g. inserting of a knot into the knot vector U increases the lattice size to $((l+1) \times m \times n)$ control points. By inserting knots successively into the knot vectors U, V, and W, the lattice can be expanded to an arbitrary size.

The challenge is to decide where the knots are to be inserted, so that the approximation error decreases as much as possible. Here, it is important to keep the lattice size small in order to not increase the computing time unnecessarily. Therefore, we subdivide the FFD volume into cells or subspaces and analyze the discrete deformation field inside each cell. Cells with strong deformation will be refined by knot insertion.

An intuitive subdivision of the FFD volume is given by the knot vectors U, V, and W. With respect to the initialization of the minimal B-spline volume, the parameter space is equal to the model space and the unequal knots inside a knot vector indicate the partitioning of the volume into the corresponding direction. For instance, the knot vector $U = \{\underbrace{u_{min}, \ldots, u_{min}}_{p+1}, \underbrace{u_{max}, \ldots, u_{max}}_{p+1}\}$ yields only one partition of the initial B-spline volume into x-direction, in particular, $[u_{min}, u_{max}]$ with respect to parameter space or, equivalently, $[x_{min}, x_{max}]$ with respect to model space. Furthermore, inserting a knot u_1 would subdivide the model space into cells $[x_{min}, u_1] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$ and $[u_1, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$. In this process, we consider three partitionings of the FFD volume, one in each direction.

In the next step, the deformation field inside each cell is analyzed in order to decide which cell has to be subdivided. For the comparison of the cells, an appropriate measure D is required, which describes the strength of space deformation inside a cell. The simplest one is the maximum length M of all deformation vectors. This will force the refinement of at least three cells, one for each direction, which is not always necessary. In fact, the disturbance of the vector field or, rather, its gradients into the corresponding direction, e.g., gradient into x-direction for u-cells, should be taken into account by the measure D. Additionally, large cells should be preferred for the refinement. These requirements are combined into one measure, e.g., for the u-cells, by:

$$D_x = M \cdot L_x \cdot \frac{1}{d} \sum_{i=1}^{d} \delta_x(\vec{\boldsymbol{v}}_i), \quad (6)$$

where L_x is the length of the cell into x-direction and $\delta_x(\vec{\boldsymbol{v}}_i)$ is the partial derivative of the vector field at vector $\vec{\boldsymbol{v}}_i = (\boldsymbol{p}_i, \boldsymbol{q}_i)$. We approximate $\delta_x(\vec{\boldsymbol{v}}_i)$ for the k-neighborhood by:

$$\delta_x(\vec{\boldsymbol{v}}_i) = \frac{1}{k} \sum_{j=1}^{k} \frac{||F_s(\vec{\boldsymbol{v}}_j) - F_s(\vec{\boldsymbol{v}}_i)||_2 \cdot |(\boldsymbol{p}_j - \boldsymbol{p}_i) \cdot \boldsymbol{b}_x|}{||\boldsymbol{p}_j - \boldsymbol{p}_i||_2^2}, \quad (7)$$

where $\boldsymbol{b}_x = (1,0,0)^T$ is one of the canonical basis vector in $\mathbb{R}^3$ and $F_s(\vec{\boldsymbol{v}}_i) = (\boldsymbol{q}_i - F_s(\boldsymbol{p}_i))$ is a residual deviation vector after deformation with F_s. Note, $F_0(\boldsymbol{p}_i) = \boldsymbol{p}_i$. The deformation measures D_y and D_z for v- and w-cells are calculated analogously. Among all cells we first determine the maximum value D_{max} and then refine the cells with $D \geq \alpha D_{max}$, $0 \leq \alpha \leq 1$, by inserting a knot into the middle of the cell. By varying the scalar α, it is possible to control the number of subdivided cells: for $\alpha = 1$, only one cell is refined and, for $\alpha = 0$, all cells are refined. Due to the cubic time complexity of SVD, inserting only one knot per iteration results in the shortest total computing time, although the number of iterations and the lattice size are not necessarily minimal. Thus, we set $\alpha = 1$ in the following numerical studies.

Fig. 1. Source and target shape of the hat profile.

IV. Results

The presented method is validated on the basis of two examples from sheet metal forming. The first example is a simple hat profile, see Fig. 1. It is a 2.5D object, i.e., the object has no variation of the shape along the x-axis, and the corresponding mesh has $5,859$ vertices. The target shape is the reference shape of the workpiece and the source shape was obtained from the target shape by bending. Thus, the displacements of the mesh vertices are known.

We use FFD for the adjustment of the source shape to the target shape with respect to the displacement vectors. Initially, the B-spline volume has a lattice with $(2 \times 4 \times 4)$ control points. The adjustment is proceeded by the adaptive FFD and the process terminates if the normalized residual $R_N = \frac{1}{d}R$ is below a user-defined threshold ϵ, which we set to 0.0001. For this simple example from Fig. 1, ten iterations were required to achieve the desired approximation quality. In this process the lattice has been refined to the size of $(2 \times 13 \times 4)$ control points in approx. 7 seconds. The calculated B-spline volume

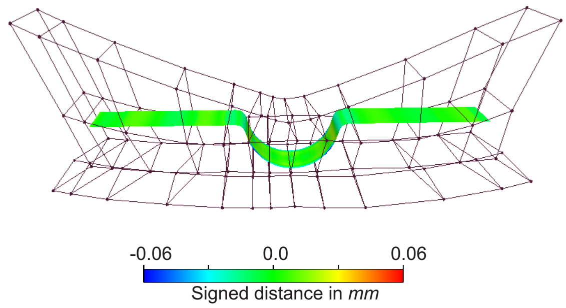

Fig. 2. Adjustment of the source shape to the target shape by adaptive FFD for the hat profile.

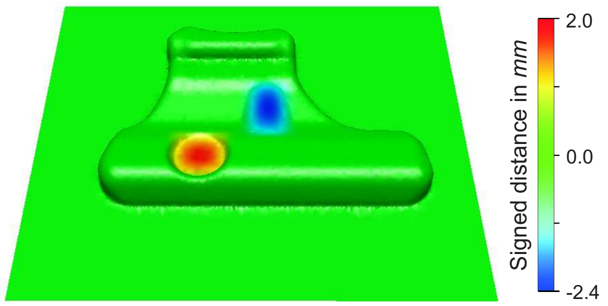

Fig. 3. Initial shape deviation of the tank.

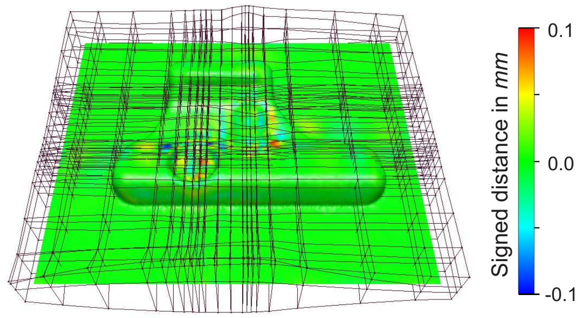

Fig. 4. Shape deviation of the tank after the adjustment by FFD.

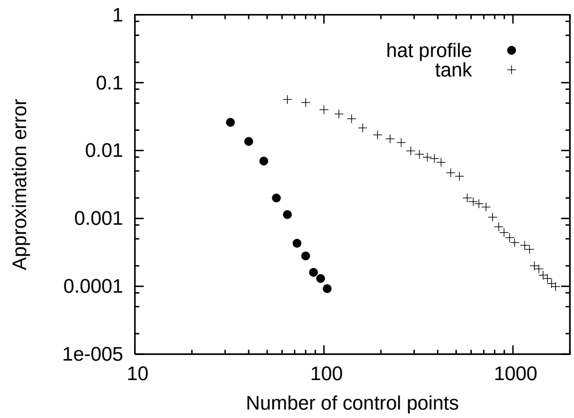

Fig. 5. Development of the approximation error R_N with regard to the number of control points.

and the deformed shape are presented in Fig. 2. It visualizes the Euclidean distance between the source and the target shape with respect to the displacement vectors after the adjustment with FFD. The maximum and average distances between the deformed source shape and the target shape were lowered from 9.0 mm and 4.958 mm to 0.06 mm and 0.007 mm, respectively. As expected, the lattice was refined into y-direction, which has the most space disturbance. For comparison, using a common, uniform B-spline volume with $(2 \times 13 \times 4)$ control points for this adjustment, the maximum and average distance were reduced to 0.08 mm and 0.014 mm, respectively, in approx. 4 seconds. For this example, the presented FFD approach outperforms the common method with respect to accuracy.

The second example is the geometry of the die for a tank wall, which we simply call the tank. The die was designed in a CAD environment and exported to a mesh with $5,577$ vertices. The target shape was generated from the source shape by displacing some of the vertices, see Fig. 3. Here, the deviations of the target shape are local in nature and vary from -2.4 mm to 2 mm. This is a more challenging and practically relevant shape. Again, the displacements of vertices are known and we fit the source shape into the target shape by using the adaptive FFD.

The initial B-spline volume has a lattice with $(4 \times 4 \times 4)$ control points, which encloses both shapes. The termination threshold was set to 0.0001. Since the deformation of the tank is complex in shape, the adjustment process needs 34 iterations and a computing time of approx. 8 minutes. The lattice was refined to the size of $(20 \times 21 \times 4)$ control points. The deviation of the source shape from the target shape after adjustment is shown in Fig. 4. The maximum and average distances have been reduced to 0.11 mm and 0.005 mm, respectively. Furthermore, it can be observed that the FFD lattice has a high density of control points in the deformed areas. The calculation of the uniform B-spline volume with the same size for $5,577$ displacement vectors takes approx. 23 minutes. Thereby the maximum and average distance were lowered to 0.24 mm and 0.012 mm, respectively.

Additionally, we analyze the convergence behavior of our algorithm. Fig. 5 presents the development of the approximation error R_N with regard to number of control points. Considering the logarithmic scales, it can be seen that, in this case, the convergence behavior of the presented method is superlinear.

V. Conclusion

In this paper, an adaptive free-form deformation technique was presented, which is used for modifying a workpiece geometry according to a set of displacement vectors. The novelty of this method is the iterative approach, in which the lattice of the FFD volume is automatically refined with regard to the current shape deviations. Due to the locality of B-splines, only a small number of control points is recomputed in each iteration, thus, the computing time was decreased significantly. Another advantage is the possibility to control the approximation quality of the FFD. This increases the accuracy and the efficiency of the manufacturing process, in which the FFD volume is used for shape modification of the workpiece.

The numerical studies have shown that the desired shape modifications can be obtained by deforming the object using the adaptive FFD. The presented method is more efficient than the common FFD method using a B-spline volume with a uniform control point lattice. The rate of convergence appears to be superlinear with respect to the lattice size. Further improvements can be achieved by a nonuniform subdivision of cells. For practical use, the initial lattice of FFD can be roughly adapted to the object shape by the user in order to reduce the number of iterations and speed up the calculation.

ACKNOWLEDGMENT

This work has been funded as subproject C4 of the Collaborative Research Center ”3D-Surface Engineering” (SFB 708) by the German Research Foundation (DFG).

REFERENCES

[1] T. Várady, R. R. Martin, and J. Cox, “Reverse engineering of geometric models – an introduction,” *Computer-Aided Design*, vol. 29, no. 4, pp. 255–268, 1997.

[2] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[3] W. Gan and R. H. Wagoner, “Die design method for sheet springback,” *International Journal of Mechanical Sciences*, vol. 46, no. 7, pp. 1097–1113, 2004.

[4] A. H. Barr, “Global and local deformations of solid primitives,” *SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 21–30, 1984.

[5] L. Orazi, “Constrained free form deformation as a tool for rapid manufacturing,” *Computers in Industry*, vol. 58, pp. 12–20, 2007.

[6] R. F. Sarraga, “Modifying cad/cam surfaces according to displacements prescribed at a finite set of points,” *Computer-Aided Design*, vol. 36, no. 4, pp. 343–349, 2004.

[7] D. Biermann, A. Sacharow, T. Surmann, and T. Wagner, “Direct free-form deformation of nc programs for surface reconstruction and form-error compensation,” *Production Engineering – Research and Development*, vol. 4, pp. 501–507, 2010.

[8] T. W. Sederberg and S. R. Parry, “Free-form deformation of solid geometric models,” *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 151–160, 1986.

[9] S. Coquillart, “Extended free-form deformation: a sculpturing tool for 3d geometric modeling,” *SIGGRAPH Comput. Graph.*, vol. 24, no. 4, pp. 187–196, 1990.

[10] J. Griessmair and W. Purgathofer, “Deformation of solids with trivariate b-splines,” in *Proceedings of Eurographics*, 1989, pp. 137–148.

[11] H. J. Lamousin and W. N. J. Waggenspack, “Nurbs-based free-form deformations,” *IEEE Computer Graphics and Applications*, vol. 14, pp. 59–65, 1994.

[12] W. M. Hsu, J. F. Hughes, and H. Kaufman, “Direct manipulation of free-form deformations,” in *SIGGRAPH ’92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1992, pp. 177–184.

[13] S.-M. Hu, H. Zhang, C.-L. Tai, and J.-G. Sun, “Direct manipulation of ffd: efficient explicit solutions and decomposible multiple point constraints,” *The Visual Computer*, vol. 17, no. 6, pp. 370–379, 2001.

[14] L. Piegl and W. Tiller, *The NURBS book (2nd ed.)*. New York, NY, USA: Springer-Verlag, 1997.

 ISBN: 978-1-61208-172-4