Elliptic Grid Generation with B-Spline Collocation

Philipp Lamby
Karl-Heinz Brakhage
Institute for Geometry and Applied Mathematics
RWTH Aachen University
Templergraben 55, 52078 Aachen, Germany
{lamby,brakhage}@igpm.rwth-aachen.de

Abstract

We revisit the classical technique of elliptic grid generation with harmonic mappings. For the determination of the control functions we use the framework developed by Spekreijse [1]. However, instead with finite differences we discretize the underlying partial differential equation with a B-spline collocation method in order to work directly with the native data representations of our CAGD system. This way we can make use of the sparsity and accuracy of the B-spline boundary representations and guarantee the geometric consistency of our CAD models. In this paper we will summarize the underlying algorithms and present some first application examples.

Introduction

In the context of the development of a new adaptive Navier-Stokes solver QUADFLOW which aims at the simulation of fluid-structure interaction at airplane wings, cf. [3], a new grid generation module has been implemented which is based on the representation of the geometry with parametric mappings. As in many common CAD systems free form curves and surfaces are represented by B-splines. From this parametric representations adaptive grids are computed by function evaluation. Concretely, within this system curves are represented by B-splines of the form

$$\mathbf{x}(u) = \sum_{i=0}^{N} \mathbf{p}_i \, N_{i,p,U}(u), \tag{1}$$

planar grids and surfaces are modeled by bivariate B-spline tensor products

$$\mathbf{x}(u,v) = \sum_{i=0}^{N} \sum_{j=0}^{M} \mathbf{p}_{i,j} \, N_{i,p,U}(u) \, N_{j,q,V}(v) \tag{2}$$

and volume grids are represented by trivariate mappings

$$\mathbf{x}(u,v,w) = \sum_{i,j,k=0}^{N,M,L} \mathbf{p}_{i,j,k}\, N_{i,p,U}(u)\, N_{j,q,V}(v)\, N_{k,r,W}(w). \tag{3}$$

Here U, V, W are non-decreasing and non-stationary *knot sequences*, i.e.,

$$U = (u_i)_{i=0}^{N+p-2}: \quad u_0 \le u_1 \le \ldots \le u_{N+p-2}, \quad u_{i+p} < u_i, \tag{4}$$

and $N_{i,p,U}$ denotes the i-th normalized B-spline-function of order p defined by the recursion

$$N_{i,1,U}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

$$N_{i,p,U}(u) = \frac{u - u_i}{u_{i+p-1} - u_i} N_{i,p-1}(u) + \frac{u_{i+p} - u}{u_{i+p} - u_{i+1}} N_{i+1,p-1}(u). \tag{6}$$

B-spline curves are piecewise polynomials of degree $p-1$. Usually we choose $p = 4$, i.e. cubic splines, and knot sequences with p-fold knots at the interval ends. This has the advantage, that the knot sequence interval coincides with the parameter interval of the curve, that the first and last control point coincide with the start and end point of the curve, and that the first and last span of the control polygon are tangential to the curve at the start and end point of the curve. In many practical cases non-uniform knot sequences are constructed, for instance in the course of an adaptive approximation or interpolation procedure. Multiple interior knots can be used to model non-smooth features of an object.

Grid Generation Equations

In order to generate smooth grids, all the common techniques of structured grid generation, namely transfinite interpolation and methods based on the solution of partial differential equations are applied. In this paper we want to integrate an elliptic grid generation technique into our CAGD system. Our choice was for Spekreijse's approach which can be very briefly summarized as follows: let $\mathbf{x}(\mathbf{s})$ be a harmonic mapping from the d-dimensional parameter space $\mathcal{P}$ onto the physical domain $\mathcal{C}$ and $\mathbf{s}(\boldsymbol{\xi})$ be a so-called control mapping from the computational domain $\mathcal{C}$ onto the parameter domain $\mathcal{P}$. Then the composite mapping

$$\mathbf{x}(\mathbf{s}(\boldsymbol{\xi})): \quad \mathcal{C} \longrightarrow \mathcal{D} \tag{7}$$

fulfills a differential equation of the form

$$L(\mathbf{x}) = \sum_{i,j=1}^{d} g_{ij} \frac{\partial^2 \mathbf{x}}{\partial \xi_i \xi_j} + \sum_{k=1}^{d} P_k \frac{\partial \mathbf{x}}{\partial \xi_k} = 0 \tag{8}$$

where

$$P_k = \sum_{i,j=1}^{d} J^2 g^{ij} P_{ij}^k, \tag{9}$$

$J = \det \mathbf{x}'(\boldsymbol{\xi})$ is the Jacobian of the composite mapping, the g_{ij} and g^{ij} are the covariant and contravariant metric tensors defined by

$$g_{ij} = \frac{\partial \mathbf{x}}{\partial \xi_i} \cdot \frac{\partial \mathbf{x}}{\partial \xi_j}, \quad \sum_{k=1}^{d} g_{ik} g^{kj} = \delta_{ij}, \tag{10}$$

the P_{ij}^k are the components of the vector

$$\mathbf{P}_{ij} = -T^{-1} \frac{\partial^2 \mathbf{s}}{\partial \xi_i \partial \xi_j}, \tag{11}$$

and $T = \mathbf{s}'(\boldsymbol{\xi})$ is the Jacobian matrix of the control mapping. The aim of this paper is to solve this PDE with a B-spline collocation method.

B-Spline Collocation

The general idea of collocation is to determine a function, so that it exactly satisfies the differential equation at certain points, the collocation points. In a way collocation is similar to interpolation, but in contrast to interpolation we do not match function values but certain combination of function and derivative values. In order to simplify the notation we concentrate onto the bivariate case from now on and denote the Cartesian coordinates of the computational domain, the unit square, with $\boldsymbol{\xi} = (u, v)$ and of the parameter domain with $\mathbf{s} = (s, t)$. Hence, we search a function of the form (2) which fulfills

$$L\mathbf{x}(\hat{u}_i, \hat{v}_j) = 0, \quad i = 1, \ldots, N-1, \quad j = 1, \ldots, M-1 \tag{12}$$

at certain collocation points $\hat{u}_i$, $\hat{v}_j$ and Dirichlet boundary conditions for the control points $p_{0,j}$, $p_{N,j}$, $j = 0, \ldots, M$ and $p_{i,0}$, $p_{i,M}$, $i = 0, \ldots, N$. The task is now to choose appropriate collocation points for the configuration at hand.

The most popular B-spline collocation scheme is Gauß collocation with cubic Hermite-splines. Its application to elliptic grid generation has already been investigated by Manke [2]. Here in each knot interval the collocation points are the abscissae of Gaussian quadrature rules. An obvious caveat is that in a preprocessing step one has to make all interior knots two-fold by knot-insertion, and that therefore the resulting grid will be only C^1. Our preference is for a scheme that works for splines with arbitrary knot

sequences and uses the *Greville abscissae*, which are defined by

$$\breve{u}_i = \sum_{k=i+1}^{i+p} u_k, \tag{13}$$

as collocation points. This choice is motivated by the Schoenberg-Whitney theorem, see reference [4], which says that the interpolation problem $x(\hat{u}_i) = f_i$ is well posed if, and only if, every $\hat{u}_i$ lies in the support of the $i-th$ B-spline function, i.e., if $N_i(\hat{u}_i) > 0$. As one can easily verify, the Greville abscissae always give a set of as many distinct points, as the spline has control points and they fulfill the conditions of the Schoenberg-Whitney theorem. This collocation scheme frees us from the necessity to insert additional knots in our tensor product representations. A disadvantage, however, is that collocation at the Greville abscissae does not have the optimal consistency order.

The Schoenberg-Whitney theorem is also the reason why we do not use a standard finite difference code followed by an interpolation algorithm in order to generate elliptic B-spline grids: a typical finite difference code is based on the assumption that the grid points $\mathbf{x}_{i,j}$ are numerical approximations of regular spaced values $\mathbf{x}(ih_u, jh_v)$. However, depending on the structure of the underlying spline it could become necessary to work with unevenly spaced grids in order to fulfill the stipulations of the Schoenberg-Whitney theorem during the interpolation process.

Application Example

The afore-mentioned collocation schemes have been implemented and tested for planar grids, surfaces and volume grids. In order to solve the systems we just follow the standard approach and use a fixed point iteration, freezing the metric coefficients in Equation (8) in order to get a linear system in every single iteration. Then we apply the collocation scheme to the linearized equations. The arising sparse linear systems are solved with a direct solver. This kind of implementation is not well suited to solve big systems with maximum efficiency, but the aim of the current study was not to compare the efficiency of the implementation (this has already be done in [2]), but to study the principal validity of the method.

As a first application we present the grid in a block that is taken from a grid for a dual-bell configuration, see Figure 1. The boundaries are approximatively parameterized by arclength, so that we can use the identic mapping as control mapping. Hence, all control functions P_{ij}^k are zero and the resulting grid mapping is harmonic. However, the spacing of the control points, that can be seen in the upper plot is rather irregular. This irregularity

stems from an adaptive B-spline approximation algorithm which tried to resolve the different features of the nozzle contour and from the necessity to mutually insert the knots which are not present in the representation of the opposite boundary in order to build a tensor product. However, the resulting numerical grid, which is computed by evaluation of the B-spline function has the desired smoothness properties.

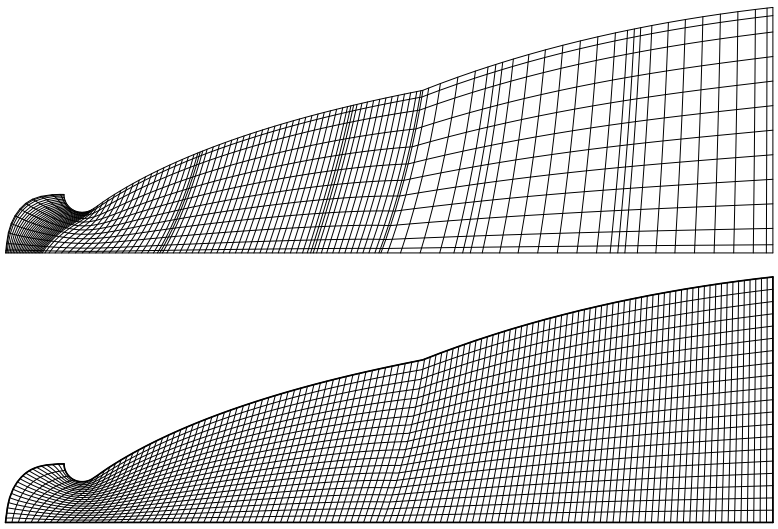

Figure 1. Control points and harmonic mesh for the dual bell.

Boundary Orthogonality

In Spekreijse's approach there remains the problem to determine suitable control functions in order to incorporate desired features into the grids. In order to find a control mapping that ensures boundary orthogonality Spekreijse proposes to proceed as follows. Let us assume that a folding-free grid $\mathbf{x}(\boldsymbol{\xi})$ is already available. This grid may be, for instance, a transfinite interpolant or the solution of the purely harmonic grid generation system. Then in the first step we solve the transformed Laplace equation

$$\operatorname{div}(A \operatorname{grad} s) = 0 \tag{14}$$

where

$$A = J \begin{pmatrix} g^{11} & g^{12} \\ g^{12} & g^{22} \end{pmatrix} = \frac{1}{J} \begin{pmatrix} g_{22} & -g_{12} \\ -g_{12} & g_{11} \end{pmatrix}. \tag{15}$$

This equation is supplied by mixed Dirichlet and Neumann boundary conditions, in particular we require $\partial s/\partial \mathbf{n} = 0$ at the boundaries $\mathbf{x}(u,0)$ and $\mathbf{x}(u,1)$ and $\partial t/\partial \mathbf{n} = 0$ at the boundaries $\mathbf{x}(0,v)$ and $\mathbf{x}(1,v)$ of the physical domain. The solution of this problem gives us a one-to-one boundary mapping $\partial\mathcal{C} \longrightarrow \partial\mathcal{P}$. In the second step we complete this boundary mapping to a suitable control mapping that fulfills the orthogonality conditions $\partial t/\partial u = 0$ along the boundaries $u = 0$ and $u = 1$ and $\partial s/\partial v = 0$ along the boundaries $v = 0$ and $v = 1$ using an algebraic grid generation method. For the details of this method we have to refer to [1].

Whereas the discretization of Equation 8 by collocation is straightforward it is more convenient to discretize equation 14 with a finite volume method. For this we observe that for any control volume Ω in the computation domain the equation

$$\int_{\partial\Omega} (\mathrm{grad} s, A\mathbf{n})\, d\sigma = 0 \tag{16}$$

holds. Of course, as control volumes we will choose intervals of the form $[u_i, u_{i+1}] \times [v_j, v_{j+1}]$. Again we want to represent the control mapping as tensor product B-spline. Therefore, the integral over the boundary of the control volume is composed of segments of the form

$$\begin{aligned}\int_{u_i}^{u_{i+1}} (\mathrm{grad} s, A\mathbf{n})\, d\sigma &= \int_{u_i}^{u_{i+1}} \left(\begin{pmatrix} s_u \\ s_v \end{pmatrix}, A \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) d\sigma \\ &= \sum_{i,j} p_{ij} \left[-N_j(v) \int_{u_i}^{u_{i+1}} N_i'(u) \frac{g_{12}}{J} du + N_j'(v) \int_{u_1}^{u_2} N_i(u) \frac{g_{11}}{J} du \right]\end{aligned}$$

and

$$\begin{aligned}\int_{v_j}^{v_{j+1}} (\mathrm{grad} s, A\mathbf{n})\, d\sigma &= \int_{v_j}^{v_{j+1}} \left(\begin{pmatrix} s_u \\ s_v \end{pmatrix}, A \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) d\sigma \\ &= \sum_{i,j} p_{ij} \left[-N_i(u) \int_{v_j}^{v_{j+1}} N_j'(v) \frac{g_{12}}{J} dv + N_i'(u) \int_{v_j}^{v_{j+1}} N_j(v) \frac{g_{22}}{J} du \right].\end{aligned}$$

The integrals in the brackets enter the matrix of the discretized problem and can cheaply be evaluated by quadrature formulas. In order to get as many equations as control points we center the control volumina around the Greville abscissae by choosing

$$\begin{aligned} u_i &= \frac{1}{2}(\check{u}_{i-1} + \check{u}_i), \quad i = 1, \ldots, N-1, \quad u_0 = 0, \quad u_N = 1 \\ v_i &= \frac{1}{2}(\check{v}_{j-1} + \check{v}_j), \quad i = 1, \ldots, N-1, \quad v_0 = 0, \quad v_M = 1 \end{aligned} \tag{17}$$

The boundary condition $\partial s/\partial \mathbf{n}$ transforms to $(\mathrm{grad} s, A\mathbf{n}) = 0$ at the corresponding boundary in $\mathcal{C}$, so that the discretization at boundary grid points

is also straightforward. Figure 2 shows the control points and the smooth evaluation of the resulting orthogonal grid, Figure 3 shows the corresponding control mapping and a detail view at the nozzle throat.

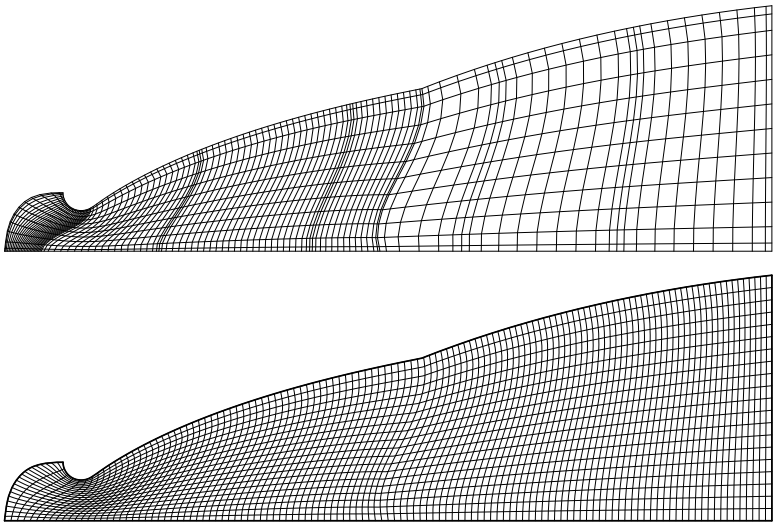

Figure 2. Control points and orthogonal mesh for the dual bell.

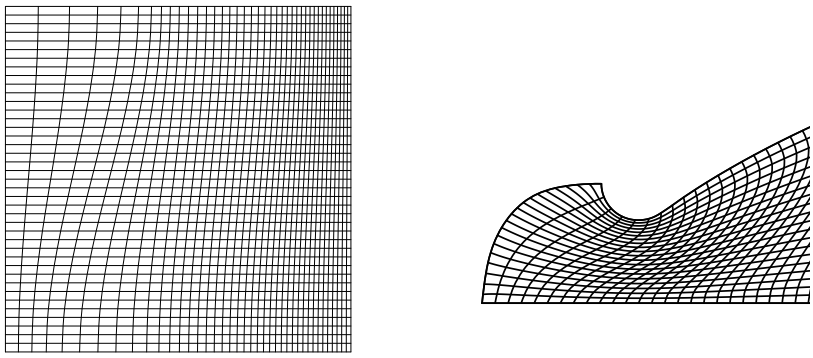

Figure 3. Control Map and Detail View

Convergence and Stability Matters

The above example shows that in principle the collocation method presents a viable method to integrate elliptic grid generation strategy into a spline based CAGD system. However, it turns out that there are also some complications. First signs for these problems already reveal themselves in the following convergence study.

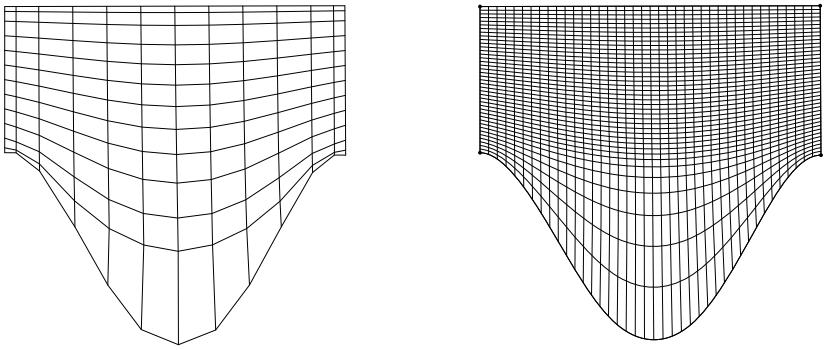

Figure 4. Cosine Testcase

We consider a simple rectangle where the lower side has been replaced by a cosine-like arc, see Figure 4, and compare the L_2-residual of the fully converged solutions, i.e., the solution we get when the fixed point iteration does not improve the solution any more.

	Gauß		Greville	
N	$r_N := \|L(x)\|_2$	$r_{N/2}/r_N$	$r_N := \|L(x)\|_2$	$r_{N/2}/r_N$
10	4.832e+2		4.757e+2	
20	3.138e+2	1.54	2.916e+2	1.63
40	1.233e+1	2.55	1.480e+1	1.97
80	6.146	2.01	5.939	2.49
160	2.570	2.39	1.988	2.99

Figure 5. Convergence behavior of the cosine test case.

First of all we observe that the Gauß collocation scheme does not produce better convergence faster than the Greville collocation scheme, although theory predicts fourth order convergence for the former and only second order for the latter. (We can indeed observe these rates for the linear Laplace equation with our implementation). The Hermite scheme even converges

slower than the Greville scheme, what somehow justifies our preference for the Greville scheme. One reason for this might be the very bad condition of the collocation matrices which typically goes with N^o where o is the order of the scheme. However, from the geometric point of view the result is not entirely satisfactory either. As is well known, harmonic grid generation systems have the tendency to push away the grid lines from concave boundaries. Especially if one tries to apply the collocation scheme with very coarse control nets this tendency seems to be even accentuated. Figure 4, for example, shows the resulting 10×10 control point grid and the corresponding grid evaluation. This defect disappears in the course of extensive grid refinement, but only very slowly. For instance, in the even more extreme example of Figure 6 one needs more than 160 control points in each coordinate direction before the grid lines start to converge towards the boundary. The Figure itself shows the result of the discretization based on 40×40 control points.

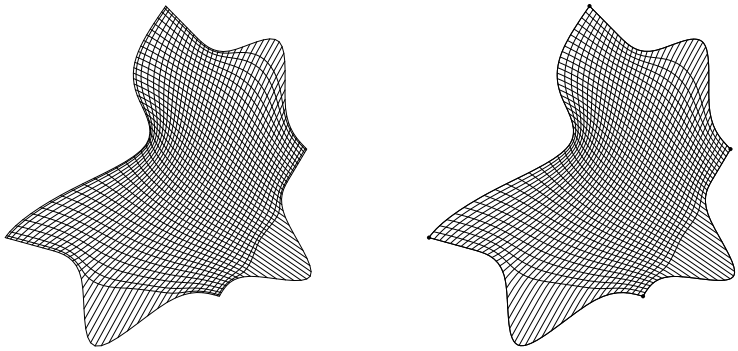

Figure 6. Discretization with 40×40 control points

At this place it is interesting to note, that the solution of the transformed Laplace equation (14) can also be used to compute for any given grid mapping $\mathbf{x}(u, v)$ a corresponding control mapping $s(u, v)$. This can be done by replacing the Neumann boundary conditions by standard Dirichlet conditions. Figure 7 shows the control mapping that corresponds to the grids in Figure 6. The aberration of this grid from the identic mapping is obviously caused by the discretization error.

Conclusion

The here proposed collocation scheme presents a useful method to realize elliptic grid generation methods directly on B-spline representations. However, one needs additional measures to control the boundary spacing.

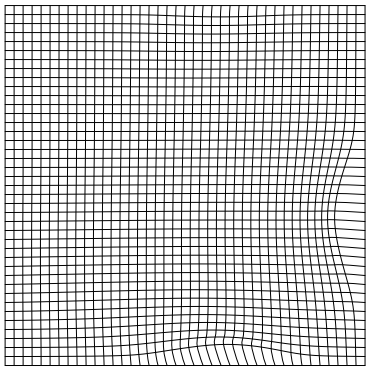

Figure 7. Control mapping corresponding to the grid in Figure 6.

Following Spekreijse this would afford the computation of appropriate control mappings via solution of the biharmonic equation. This, however, has not yet been implemented for B-spline grids. Since in [2] Manke does not report similar problems, it might well be that methods which compute the control functions iteratively are a reasonable alternative in this context, too.

Acknowledgments

This work has been and performed with funding by the Deutsche Forschungsgemeinschaft in the Collaborative Research Center SFB401 "Flow Modulation and Fluid-Structure Interaction at Airplane Wings" of the RWTH Aachen, University of Technology, Aachen, Germany.

References

[1] Stefan Spekreijse. Elliptic Grid Generation Based on Laplace-Equations and Algebraic Transformations. *J. Comp. Phys.*, 118:38–61, 1995.

[2] J. W. Manke. A Tensor Product B-Spline Method for Numerical Grid Generation. *J. Comp. Phys.*, 108:15–26, 1993.

[3] Frank Bramkamp, Philipp Lamby, and Siegfried Müller. An adaptive multiscale finite volume solver for unsteady and steady state flow computations. *J. Comp. Phys.*, 197:460–490, 2004.

[4] Carl de Boor. *A Practical Guide to Splines.* Springer, 1978.