# Efficient Iterative Solvers for Angle Based Flattening

Rhaleb Zayer, Christian Rössl, Hans-Peter Seidel

Max-Planck-Institut für Informatik
Computer Graphics Group
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
Email: {zayer, roessl, hpseidel}@mpi-sb.mpg.de

## Abstract

In this paper, we derive an efficient approach for solving the optimization problem which arises in *Angle Based Flattening*. As the size of system matrix associated with the ABF blows up to a size approximately five times the number of faces, finding a solution for reasonably sized triangular meshes becomes a challenging problem. We propose two iterative approaches to overcome these limitations. The first approach is based on the fact that decoupling the system yields a much easier matrix equation. This enables an iterative approach for carrying out the computations. The second approach is based on the analysis of saddle point problems. We derive a modified Uzawa algorithm for efficiently addressing the numerical optimization problem.

## 1 Introduction

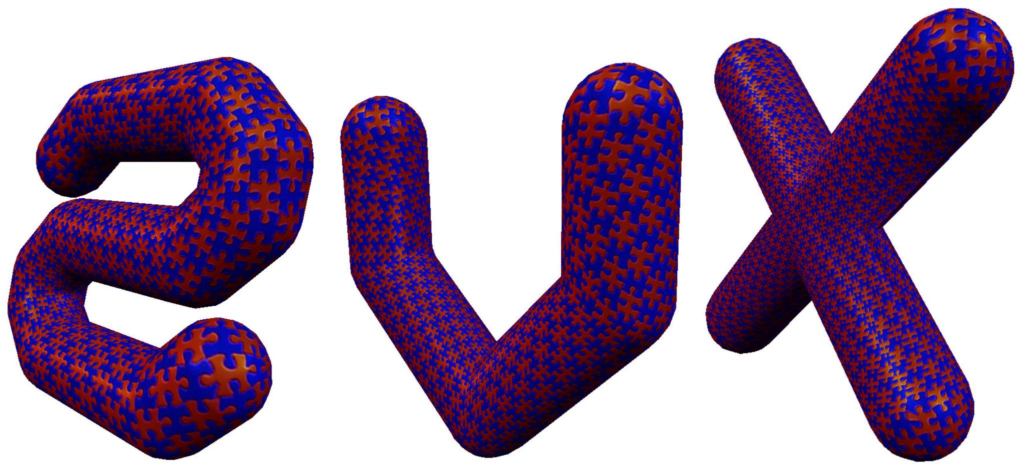
Surface parameterization studies the embedding of 3D surface meshes into the plane. In a typical setting, the given surface is homeomorphic to a disc. The rareness of isometric mappings induced a large focus on conformal maps. A wide range of the existing methods relies on the discretization of the Laplace equation in order to establish the embedding given appropriate boundary conditions, see [9] for a comprehensive survey. One major drawback of these methods is that the treatment of the boundary does not reflect the behavior of the original surface boundary. This may be of importance when the original boundary cannot be approximated by a convex polygon. While pseudo-conformal parameterizations such as [4, 6, 12, 13, 16] propose several schemes to minimize angular distortion, it seems natural to formulate the problem in terms of interior angles of the flat mesh. This leads to the *Angle Based Flattening* (ABF) method introduced by Sheffer and de Sturler [17]. The ABF algorithm constructs the parameterization by solving a constrained optimization problem where the objective function controls the angular distortion of the planar mesh w.r.t. the angles of the original mesh. A set of linear and non-linear equality constraints on the planar angles guarantees the validity of the parameterization.

## 2 Overview

In this paper, we derive an efficient approach for solving the optimization problem which arises in *Angle Based Flattening*. As the size of system matrix associated with the ABF blows up to a size approximately five times the number of faces, finding a solution for reasonably sized triangular meshes becomes a challenging problem. Several numerical schemes have been proposed thus far to speed up the convergence of the original algorithm [17] by using preconditioning [14], smoothing [18] and sparsity pattern improvement [21].

We note that the use of a hierarchical approach is excluded due to the fact that providing an initial valid solution as an initial guess would already satisfy the constraints and hence it is local minimum. So the system will not evolve or find another minimum. The main choice left is to develop efficient numerical methods to efficiently solve the arising optimization problem.

We propose two iterative approaches. The first approach is based on the fact that decoupling the system yields a much easier matrix equation. This enables an iterative approach for carrying out the computations. The second approach is based on the analysis of saddle point problems. We propose a modified Uzawa algorithm for efficiently addressing the numerical optimization problem. The key ingredient for both methods is the decoupling of the



(a) (b) (c)

Figure 1: Textured letter and number models: (a) 2, (b) V, and (c) X. Notice that the parameterization of these surfaces on a square or a circle may yield high distortion.

matrix equations.

The rest of the paper is organized as follows, after introducing the notation, we briefly review the setup of the ABF approach in section 4, the optimization problem in section 5, and the diagonalization of the sparsity pattern improvement of the system matrix in section 7. In section 7 we propose two efficient numerical methods for the solving the matrix equations as the main contribution. We discuss the results in 8.

## 3 Conventions

In order to simplify the ensuing discussion, we first introduce the following notation:

Given a triangular mesh

- $N$ is the total number of interior mesh angles.
- $\alpha_i^*$ ($i = 1, \ldots, N$) denote the angles of the *original* mesh.
- $\alpha_i$ are the corresponding angles of the *flat* mesh. As these are the variables of the optimization problem, then in this context, the more usual notation $x_i$ is used as an alternative.
- $v$ denotes the central vertex in a centered drawing of a *wheel*, i.e. of its 1-neighborhood. $d$ is the number of direct neighbors of $v$ or its *valence*. $\alpha_j$ ($j = 1, \ldots, d$) refer to the angles at $v$, while $\beta_j$ and $\gamma_j$ denote the opposite left and right angles of a face with central angle $a_j$, respectively. All faces are oriented counter-clockwise.
- Variables and functions without subscripts may refer to multivariate vectors as explained by the context.

## 4 Characterization of drawings of planar graphs

The problem of guaranteeing the validity of the planar embedding of a mesh solely based on angles was addressed by Sheffer and de Sturler in [17]. The set of positive angles of the planar mesh should meet the following consistency conditions:

- *Triangle consistency*
  For each triangular face with angles $\alpha$, $\beta$, $\gamma$ the face consistency:

$$\alpha + \beta + \gamma - \pi = 0 \tag{1}$$

- *Vertex consistency*

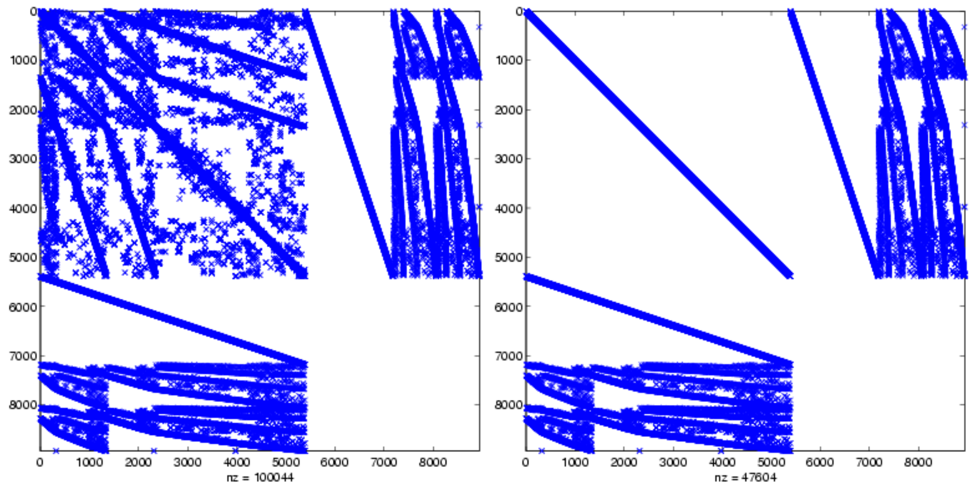



Figure 2: System matrices of equation (5) generated from the *ear* model using the (a) original wheel condition. (b) simplified wheel condition. The diagonal Hessian brought the number of nonzero elements from 100044 down to 47607.

For each internal vertex $v$, with central angles $\alpha_1, \dots, \alpha_d$:

$$\sum_{i=1}^{d} \alpha_i - 2\pi = 0 \qquad (2)$$

- *Wheel consistency*
  For each internal vertex $v$ with left angles $\beta_1$,..,$\beta_d$ and right angles $\gamma_1, \dots, \gamma_d$:

$$\prod_{i=1}^{d} \frac{\sin(\beta_i)}{\sin(\gamma_i)} = 1 \qquad (3)$$

These conditions ensure the centered embedding of internal vertices without overlapping of interior edges. However they do not prevent the overlapping of boundary edges. The problem of drawing angle based graphs is well-studied problem in graph theory [5, 11]. Di Battista and Vismara [5] provide a characterization of the convex planar straight line drawing of a tri-connected graph for a given set of positive angles. Their minimal constraints for the planarity of the graph includes (1),(2),(3) and require the additional condition that all exterior angles should be (less than $\pi$). Zayer et al. [21] discuss the solution of the ABF problem including this condition and allow direct control over the boundary behavior of the planar mesh using an active set approach to handle the additional inequality constraints.

## 5 Constrained optimization problem

A general approach to establish a surface parameterization consists of minimizing an objective function $f(x)$ that quantifies distortion w.r.t. certain quality criteria. As the validity of the flat mesh is guaranteed by the angle constraints of section 4. A typical choice of such function consists of establishing an angle based objective function [17]

$$f(x) = \sum_{i=1}^{N} w_i (x_i - a_i)^2$$

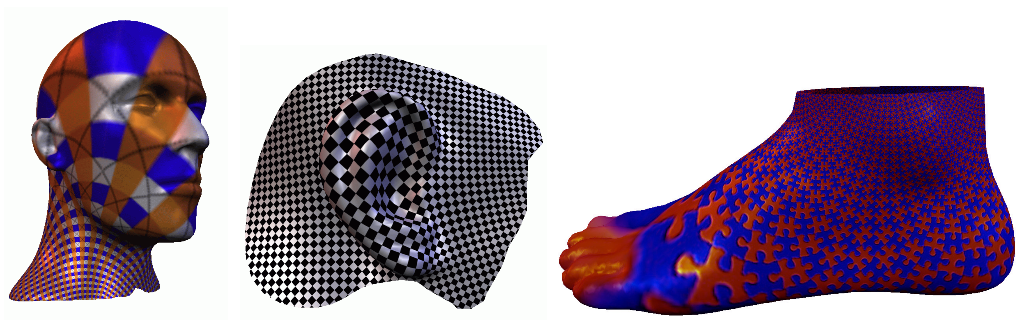
with the weights $w_i = \frac{1}{a_i^2}$ and the variables $a_i$ represent the *optimal angles* of the flat mesh, which are

$$a_i = \begin{cases} \alpha_i^* \frac{2\pi}{\sum_{i=1}^{d} \alpha_i^*} & \text{around an interior vertex} \\ \alpha_i^* & \text{around a boundary vertex} \end{cases}$$

We can now formulate the general optimization problem as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0 \end{array} \qquad (4)$$

where $h$ represents the multivariate functions of the equality constraints (1),(2),(3). It should be noted that if inequality constraints are used in order to



(a) (b) (c)

Figure 3: Textured models: (a) mannequin (c) ear (b) and (d) foot. Notice the pseudo-conformality of the parameterization.

control the boundary behavior, it is possible to convert them into equality constraints using an active set approach by altering the Lagrange multipliers associated with them. If a constraint does not figure in the active set, its associated multipliers are set to zero, otherwise it is treated as an equality constraint. The numerical advantage of this method is that as the iterates get closer to the solution, the active set becomes more and more stable. A detailed description of the active set method can be found in [1].

## 6 Optimization problem setup

The efficient solution of large constrained optimization systems of the form (4) is still an area of active research and offers several open problems in the field of non-linear optimization [3]. The adequacy of a minimization method depends on the properties of the objective function as well as on the constraints.

In order to solve the optimization problem we use the method of Lagrange multipliers as it guarantees the exact satisfaction of constraints. The arising optimization problem is solved using a Newton method in order to guarantee full satisfaction of the constraints. Additionally a line search is used to guide and enhance the iterative steps.

In every Newton iteration the following system is solved

$$\begin{bmatrix} \nabla^2_{xx} L & J_h^T \\ J_h & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \mu_h \end{bmatrix} = - \begin{bmatrix} \nabla_x L \\ h \end{bmatrix} \quad (5)$$

where the Lagrangian $L$ is given by

$$L = f(x) + \mu_h^T h(x).$$

In the classic ABF algorithm, the computation of the Hessian matrix $\nabla^2_{xx} L$ involves finding the second derivatives of the products involved in condition (3). The resulting matrix is sparse, but it still contains a considerable number of non-zero elements (cf. Fig. 2(a)). This number depends largely on the valences of the input mesh vertices.

Instead, we use the *modified wheel condition* (6) introduced in [21]. As the angles are strictly positive we can safely rewrite condition (3) as

$$\sum_{i=1}^{d} \log(\sin \beta_i) - \log(\sin \gamma_i) = 0. \quad (6)$$

The advantage of this modification resides in the fact that it yields a diagonal Hessian matrix.

$$\nabla^2_{xx} L = \text{diag}(f''(x_i) + m_i \frac{-1}{\sin^2(x_i)})$$

where $m_i$ is the linear combination of the Lagrange multipliers involved with $x_i$ in condition (6). The amount of computation and effort by the iterative solvers is hence reduced considerably. The Hessian can be computed efficiently as this reduction also avoids the estimation of complex derivatives with all the floating error they may induce. Fig. 2 illustrates the structure of a typical system matrix and the improvement induced by the *modified wheel condition*.

## 7 Solution of the matrix equation

In this section we introduce our main contribution. Our approach relies on the analysis of the structure of the system matrix $S$ in (5). In most previous approaches [17, 14, 18, 21] the problem is treated by feeding the matrix to an iterative solver in a black box fashion. Keeping the modified wheel condition in mind we take advantage of the underlying system structure in order to decouple the system into two simpler matrix equations. In fact, the optimization problem at hand can be restated as the following general matrix system

$$\begin{array}{lcl} Ax + B^t\lambda & = & f, \\ Bx & = & g. \end{array} \tag{7}$$

In our ABF setting (5) the matrix $A$ is the diagonal Hessian and $B$ is the sparse Jacobian matrix associated with the equality constraints. Then the system matrix $S$ can be expanded as

$$S = \begin{bmatrix} A & 0 \\ B & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} A & B^t \\ 0 & I \end{bmatrix}, \tag{8}$$

where $M = BA^{-1}B^t$. According to Sylvester's law of inertia the eigenvalues of $S$ are the same as the eigenvalues of $A$ and $M$. This suggests that the system is highly indefinite as both $A$ and $M$ are not positive definite. The alternative approach to directly dealing with $S$ is to decouple the system equations in order to reduce the computational effort.

### 7.1 First approach

Since the matrix $A$ is diagonal, its inverse is obtained easily. The problem (7) now reduces to solving the following system of equations

$$\begin{array}{lcl} x & = & A^{-1}(f - B^t\lambda), \\ BA^{-1}B^t\lambda & = & BA^{-1}f - g. \end{array}$$

The matrix $M = BA^{-1}B^t$ is symmetric, however it is not necessarily positive definite in general. This rules out the existence of a unique solution as the matrix might be singular. The system can be solved using the MINRES algorithm which is the equivalent of the conjugate gradient method for general symmetric matrices. The advantage of using MINRES over other existing iterative algorithms resides in the fact that it converges for definite, indefinite or singular cases and avoids break ups or stagnation [15]. Since the matrix size might be large when dealing with large meshes, it is not desirable to directly perform the matrix-matrix multiplication unless an efficient matrix package is accessible. The algorithm can be implemented without explicit computation of the matrix $M$, this is achieved by doing matrix-vector multiplication within the MINRES method especially that the matrix $A$ is diagonal.

This method reduces the size of the matrix problem to a smaller matrix problem of only approximately twice the number of triangles in the mesh.

### 7.2 Second approach

In the following we rely on the similarity of the system equation (7) with general saddle point problems. In saddle point problems, the matrix $A$ is in general positive definite which is not the case in our setting. However, we have the advantage that the inverse of $A$ is immediate. A widely used approach for solving saddle point problems relies on the Uzawa algorithm [19]. We cannot apply the Uzawa algorithm directly as it depends on the condition number of the matrix $M$. As an alternative we take advantage of direct inversion of matrix $A$ for using the conjugate directions [2]. This yields the following algorithm:

---

1. Initialize
   $x_0 = A^{-1}(f - B^t\lambda)$,
   $d_0 = Bu_0 - g, q_0 = -d_0$.
2. Repeat until convergence
   - $p_k = B^t d_k, h_k = A^{-1}p_k$,
   - $\alpha_k = \frac{<q_k, q_k>}{<p_k, h_k>}$,
   - $\lambda_{k+1} = \lambda_k + \alpha_k d_k$,
   - $x_{k+1} = x_k - \alpha_k h_k, q_{k+1} = g - Bx_{k+1}$
   - $\beta_k = \frac{<q_{k+1}, q_{k+1}>}{<q_k, q_k>}$,
   - $d_{k+1} = -q_{k+1} - \beta d_k$

---

It should be noted that the matrix multiplications

| | MINRES | Uzawa |
|---|---|---|
| Ear (1796△) | 3 | 1 |
| Man head (5420△) | 26 | 18 |
| twists (6K△) | 2 | 1 |
| Goldfeather (10K△) | 16 | 7 |
| Goldfeather (24K△) | 74 | 31 |
| foot (20K△) | 346 | 115 |

Table 1: Runtimes (in seconds) using different iterative solvers. For most models, 3 to 4 Newton iterations were needed.

involving the diagonal matrix $A$ should be treated as element by element multiplication, the matrix $A$ can be stored as a vector.

## 8 Results and Discussion

We applied our algorithm to a set of different triangular meshes. All experiments were conducted on a 1.7 GHz Intel Xeon CPU. The parameterization time depends on the number of triangles, on the geometry, as well as on the connectivity of the input mesh. For most meshes the parameterization takes few seconds to few minutes. We summarize results in table (8). Our measurements rely on a simple matrix library, and we expect our algorithms to perform much faster using highly optimized matrix packages.

We observe that the Uzawa algorithm outperforms the MINRES in our experiments. In general, the decoupling of the system equations provides an efficient mean for improving the convergence.

## 9 Conclusion

We presented and discussed the non-linear constrained optimization problem arising in in The ABF parameterization. We take advantage of the structure of the system matrix in order to decouple the problem and we end up with much easier matrix equations to solve. The use of a modified version of the Uzawa algorithm-typically used in saddle point problems allows to efficiently solve the matrix equations.

## References


[1] D.P. Bertsekas. *Constrained optimization and lagrange multiplier methods*. Athena Scientific, 1996.

[2] D. Braess. *Finite Elements. Theory, Fast Solvers and Applications in Solid Mechanics*. Cambridge University Press 2001.

[3] R.H. Byrd, J. Nocedal Active set and interior methods for nonlinear optimization *Doc. MATH*, Extra Volume ICM III, 1998, pp. 667–676.

[4] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of triangle meshes. *Proc. Eurographics 2002*, pp. 209-218.

[5] G. Di Battista and L. Vismara. Angles of planar triangular graphs. *SIAM Journal on Discrete Mathematics*, 9 (3), 1996, pp. 349–359.

[6] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Proc. SIGGRAPH 95*, pp. 173–182.

[7] M.S. Floater. Parametrization and smooth approximation of surface triangulations. *Comp. Aided Geom. Design*, (14), 1997, pp. 231-250.

[8] M. S. Floater, K. Hormann. Parameterization of triangulations and unorganized points. *Tutorials on Multiresolution in Geometric Modelling* Springer-Verlag, Heidelberg (2002), 287-315.

[9] M. S. Floater and K. Hormann, Surface Parameterization: a Tutorial and Survey, to appear in Advances on Multiresolution in Geometric Modelling, N. Dodgson, M. S. Floater, and M. Sabin (eds.), Springer-Verlag, Heidelberg.

[10] J. Goldfeather and V. Interrante A Novel Cubic-Order Algorithm for Approximating Principal Direction Vectors. ACM Transactions on Graphics, 23(1), pp. 45-63.

[11] A. Garg. New results on drawing angle graphs. *Computational Geometry* (9), (1-2), 1998, pp. 43–82.

[12] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics*, 6(2), 2000, pp. 181–189.

[13] B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic

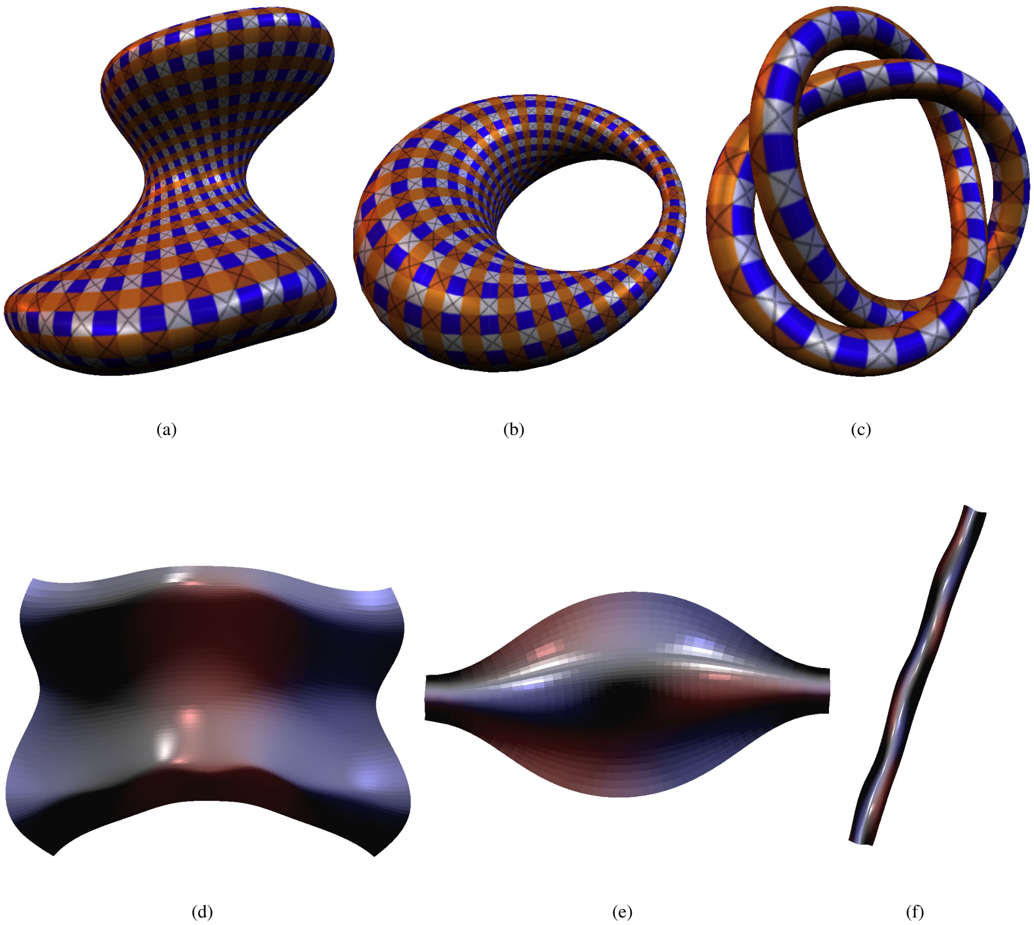



Figure 4: Textured implicit surfaces and their corresponding planar embedding: (a+d) Goldfeather surface [10], (b+e) ring cyclide, (c+f) twists.

texture atlas generation. *Proc. SIGGRAPH 2002*, pp. 362–371.

[14] J. Liesen, E. de Sturler, A. Sheffer, Y. Aydin, and C. Siefert. Preconditioners for indefinite linear systems arising in surface parameterization. *Proceedings of the 10th International Meshing Round Table*, 2001, pp. 71–81.

[15] C. Paige and M. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal*, 12, 1975, pp. 617–629.

[16] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(15), 1993, pp. 15-36.

[17] A. Sheffer and E. de Sturler. Parameterization of faceted surfaces for meshing using angle based flattening. *Engineering with Computers*, 17 (3), 2001, pp. 326–337.

[18] A. Sheffer, E. de Sturler. Smoothing an overlay grid to minimize linear distortion in texture mapping. *ACM Transactions on Graphics*, 21 (4), 2002.

[19] K. Arrow, L. Hurwicz, H. Uzawa, Studies in Nonlinear Programming, *Stanford University Press, Stanford, CA, 1958*.

[20] W. T. Tutte. How to draw a graph. *Proceed-*

*ings of the London Mathematical Society*, 13 (3), 1963, pp. 743–768.

[21] R. Zayer, C. Rössl, and H.-P. Seidel. Variations on angle based flattening. In Proceedings of Multiresolution in Geometric Modelling, 2003, pp 285-296.