SIMULATION OF SEMICONDUCTOR DEVICES AND PROCESSES Vol. 4
Edited by W. Fichtner, D. Aemmer - Zurich (Switzerland) September 12-14, 1991 - Hartung-Gorre

Transformation Methods for Nonplanar Process Simulation

Karl Wimmer, Robert Bauer, Stefan Halama,
Gerhard Hobler*, and Siegfried Selberherr

Institut für Mikroelektronik,
**Institut für Allgemeine Elektrotechnik und Elektronik,*
Technical University of Vienna, Austria

Abstract

We present a transformation method for 2D process simulation in arbitrary structures. Several grid generation techniques have been investigated systematically on their influence on convergence properties. A mapping function strategy applicable to process simulation problems is introduced. New formulations of the discretized equations based on box integration are introduced in order to satisfy global conservation laws automatically.

1 Introduction

Transformation methods are appropriate for the numerical simulation in nonplanar 2D and 3D domains [1], [2], and are widely used in computational fluid dynamics [3]. The transformation is accomplished by specifying a curvilinear coordinate system which maps the nonplanar physical domain (x, y) onto a rectangular computational domain (u, v) (see Fig. 1). With this approach the simulation domain is simplified at the expense of complicating the equations and boundary conditions.

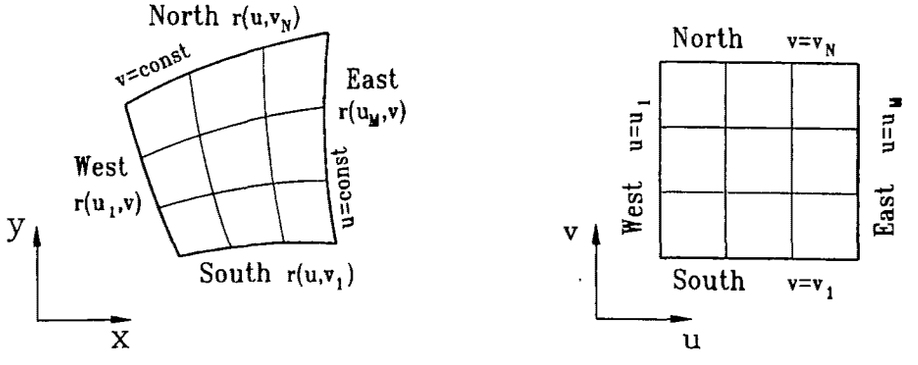

Figure 1: Transformation from physical space (x, y) to computational space (u, v) by a curvilinear coordinate system

For application in our 2D process simulator PROMIS [4], the methods must be suited for moving boundary problems. Additionally, an unknown variety of geometries has to be treated and adaptive transient grid strategies with heavy changes in grid spacing must be supported. The methods presented in this paper fulfill the above requirements.

2 Grid Generation and Adaption

The generation of the grid used for the solution of the physical equations is separated into two tasks. The first task is the generation of a mapping function. An approximately equidistant grid in the computational domain which resolves all geometric details (Fig. 2a) is mapped onto the physical domain (Fig. 2b). This grid ("geometric grid") is designed such that the points at the boundaries are approximately equidistributed along the arc length. It is only used to establish the mapping function $(u, v) \leftrightarrow (x, y)$, and has to be calculated just once for each geometry.

The second task is the generation and adaption of the grid for solving the physical equations ("physical grid"). Two criteria control the adaption of the grid according to the evolving dopant profiles. A gradient criterion guarantees the resolution of steep gradients in the dopant profiles. A dose conservation criterion minimizes the local dose error (1), which is expressed consistently with the discretization of the diffusion equation.

$$LDE_x = \frac{1}{12} \cdot \Delta x^3 \cdot \Delta y \cdot \frac{\partial^2 C(x,y)}{\partial x^2} \tag{1}$$

After each timestep during a transient simulation the adaption is achieved by inserting and deleting grid lines in the computational domain (Fig. 2c) based on the above criteria. The corresponding grid lines in the physical domain are obtained by interpolation in the geometric grid (Fig. 2d).

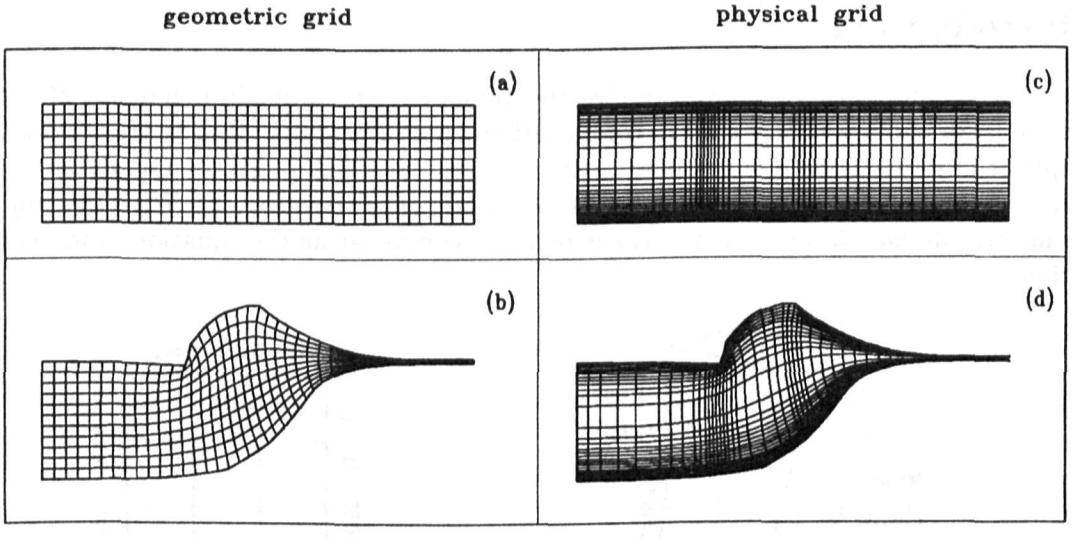

Figure 2: Grid generation and adaption: The "geometric grid" (left) is used to establish the mapping function from computational domain (a) to physical domain (b). The "physical grid" (right) is used for solving the physical equations.

For the generation of the mapping function, algebraic, elliptic or variational methods can be used.

Algebraic methods are based on an interpolation between the boundaries [5]. Using the boundary points and, if necessary, positions at internal interfaces, e.g. Si/SiO_2, we get the positions of the inner grid points from a transfinite interpolation scheme (2) with Lagrange polynomials (3).

Boundaries: South: $\vec{r}(u, v_1)$, North: $\vec{r}(u, v_N)$, West: $\vec{r}(u_1, v)$, East: $\vec{r}(u_M, v)$
Interfaces: $\vec{r}(u_I, v)$, $1 < I < M$, $\vec{r}(u, v_J)$, $1 < J < N$

$$\vec{r}(u,v) = \sum_{i=1}^{M} \Phi_i(u) \cdot \vec{r}(u_i, v) + \sum_{j=1}^{N} \Phi_j(v) \cdot \vec{r}(u, v_j) - \sum_{i=1}^{M} \sum_{j=1}^{N} \Phi_i(u) \cdot \Phi_j(v) \cdot \vec{r}(u_i, v_j) \tag{2}$$

$$\Phi_i(u) = \prod_{\substack{l=1 \\ l \neq i}}^{M} \frac{u - u_l}{u_i - u_l}, \qquad \Phi_j(v) = \prod_{\substack{l=1 \\ l \neq j}}^{N} \frac{v - v_l}{v_j - v_l} \tag{3}$$

Elliptic methods are based on the solution of the boundary value problem [2] $\Delta u = P$, $\Delta v = Q$. In the computational domain, the corresponding transformed equations (4)–(5) have to be solved. Subscripts denote partial derivatives, and $g = x_u y_v - x_v y_u$ is the Jacobian determinant.

$$a \cdot x_{uu} - b \cdot x_{uv} + c \cdot x_{vv} = -g \cdot (P \cdot x_u + Q \cdot x_v) \tag{4}$$

$$a \cdot y_{uu} - b \cdot y_{uv} + c \cdot y_{vv} = -g \cdot (P \cdot y_u + Q \cdot y_v) \tag{5}$$

$$a = x_v^2 + y_v^2, \quad b = x_u x_v + y_u y_v, \quad c = x_u^2 + y_u^2 \tag{6}$$

This system of nonlinear elliptic PDEs is discretized on the "geometric grid" using 9-point finite differences. The resulting system of coupled nonlinear algebraic equations is solved with a nonlinear SOR Algorithm [6].

A superimposed iteration scheme (7)–(9) determines the source functions P and Q at the boundaries in order to avoid clustering of gridlines at concave corners, and to allow orthogonality control at the boundaries.

$$P^0 = Q^0 = 0 \tag{7}$$

$$P^{k+1} = P^k \pm \epsilon_p \cdot \arctan\left(1 - \frac{\alpha}{\alpha_r}\right) \tag{8}$$

$$Q^{k+1} = Q^k \pm \epsilon_q \cdot \arctan\left(1 - \frac{\delta}{\delta_r}\right) \tag{9}$$

Here α denotes the angle of intersection of gridlines (10) (which should be equal to the required value $\alpha_r = \pi/2$), and δ resembles the gridspacing (11).

$$\alpha = \arccos\left(\frac{\vec{r_u} \cdot \vec{r_v}}{|\vec{r_u}||\vec{r_v}|}\right) \tag{10}$$

$$\delta = |\vec{r}(u, v_{i+1}) - \vec{r}(u, v_i)| \tag{11}$$

The upper signs in (8) and (9) relate to the South boundary, the lower signs to the North boundary. For the East and West boundaries, the correction terms for P^k and Q^k are exchanged. In the interior, P and Q are obtained by transfinite interpolation.

Variational methods offer a direct influence on certain grid properties [7]. We minimize a linear combination of three integrals (12) to control smoothness I_S, cell area I_A and orthogonality I_O of the numerically generated grid.

$$I = \lambda_S \cdot I_S + \lambda_O \cdot I_O + \lambda_A \cdot I_A \tag{12}$$

$$\lambda_S \geq 0, \quad \lambda_O \geq 0, \quad \lambda_A \geq 0, \qquad \lambda_S + \lambda_O + \lambda_A = 1 \tag{13}$$

$$I_S = \int ((\nabla x)^2 + (\nabla y)^2)/g \, du \, dv \tag{14}$$

$$I_A = -1/2 \int g^2 \, du \, dv \tag{15}$$

$$I_O = -1/2 \int w \, (x_u x_v + y_u y_v)^2 \, du \, dv \tag{16}$$

In the orthogonality functional I_O a weighting function $w(u, v)$ enables additional orthogonality control at the boundary.

The minimization problem for the overall criterion I is solved by calculating the Euler-Lagrange (EL) equations for the variational problem. The EL equations are given in the appendix. They are discretized using 9-point finite differences, the nonlinear algebraic system is solved with a nonlinear SOR algorithm [6].

Fig. 3 shows grids for a test geometry, generated with algebraic, elliptic and variational methods.

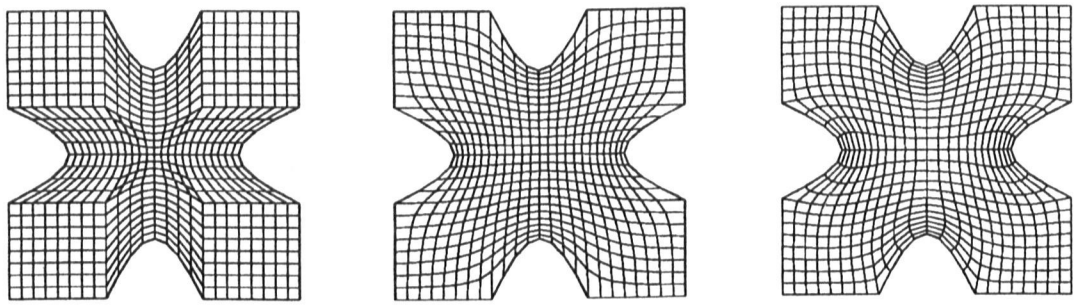

Figure 3: Grids for a test geometry generated with different methods:
left: transfinite interpolation
middle: elliptic grid with source function $P = Q = 0$
right: variational method, $\lambda_S = 0.3$, $\lambda_A = 0.7$, $\lambda_O = 0$;
$w = 10$ at the boundaries, $w = 1$ else.

3 Transformation of Physical Equations

The general structure of the diffusion equations (17)–(18) and the boundary conditions (19) in PROMIS allows the treatment of a wide variety of diffusion phenomena, including point defect assisted diffusion under oxidizing conditions [8], clustering effects [9], point defect and pair formation kinetics [10], [11], and grain boundary diffusion in polysilicon [12].

$$\sum_{j=1}^{N} \alpha_{ij} \cdot \frac{\partial C_j}{\partial t} + \operatorname{div} \vec{J}_i + \gamma_i = 0 \qquad i = 1, \ldots, N \tag{17}$$

$$\vec{J}_i = \sum_{j=1}^{N} \left(a_{ij} \cdot \operatorname{grad} C_j + b_{ij} \cdot C_j \cdot \operatorname{grad} \Psi + \vec{c}_{ij} \cdot C_j \right) + \vec{d}_i \tag{18}$$

$$\sum_{j=1}^{N} \beta_{ij} \vec{J}_j \cdot \vec{n} + \phi_i = 0 \tag{19}$$

A simple mathematical transformation from cartesian coordinates (x, y) to curvilinear coordinates (u, v) using the Jacobian matrix of the transformation leads to discretized equations which violate global conservation laws considerably. By means of box-integration we have derived a conservative form of the transformed equations (20)–(23). They satisfy global conservation properties automatically without any additional computational expense. The "conservative" transformation of the equations from physical to computational domain is performed automatically in the program.

In the most general case of a time variant physical domain (moving boundaries) we have to insert (20)–(23) into (17) to get the transformed equations in the fixed computational domain.

$$\left.\frac{\partial C_j}{\partial t}\right|_{(x,y)} = \left.\frac{\partial C_j}{\partial \tau}\right|_{(u,v)} - \frac{x_t}{g}\cdot\Big((y_v C_j)_u - (y_u C_j)_v\Big) - \frac{y_t}{g}\cdot\Big((x_u C_j)_v - (x_v C_j)_u\Big) \tag{20}$$

$$\operatorname{div}\vec{J}_i = \frac{1}{g}\cdot\Big((y_v J_i^x - x_v J_i^y)_u + (-y_u J_i^x + x_u J_i^y)_v\Big) \tag{21}$$

$$J_i^x = \sum_{j=1}^{N}\left[\frac{a_{ij}}{g}\Big((y_v C_j)_u - (y_u C_j)_v\Big) + \frac{b_{ij}}{g}C_j\Big((y_v \Psi_j)_u - (y_u \Psi_j)_v\Big) + c_{ij}^x C_j\right] + d_i^x \tag{22}$$

$$J_i^y = \sum_{j=1}^{N}\left[\frac{a_{ij}}{g}\Big((x_u C_j)_v - (x_v C_j)_u\Big) + \frac{b_{ij}}{g}C_j\Big((x_u \Psi_j)_v - (x_v \Psi_j)_u\Big) + c_{ij}^y C_j\right] + d_i^y \tag{23}$$

The boundary conditions for the North and South boundaries (v = const) and West and East boundaries (u = const) are transformed according to the normal vectors.

$$\vec{J}_j \cdot \vec{n} = \pm\frac{1}{\sqrt{x_u^2 + y_u^2}}\cdot(x_u J_j^y - y_u J_j^x) \qquad \text{North, South} \tag{24}$$

$$\vec{J}_j \cdot \vec{n} = \pm\frac{1}{\sqrt{x_v^2 + y_v^2}}\cdot(y_v J_j^x - x_v J_j^y) \qquad \text{West, East} \tag{25}$$

4 Evaluation of Grid Generation Strategies

A recessed local oxidation serves as a test example. A structure with 0.3 μm etched recess, 20 nm native oxide and a 0.1 μm nitride mask is oxidized for 90 min at 1000°C in wet ambient. Fig. 4 shows the initial structure and the final oxide shape with an algebraically generated grid. The same structure with a grid generated by a variational method ($\lambda_A = 0.5, \lambda_O = 0.2, \lambda_S = 0.3$) is shown in Fig. 2d.

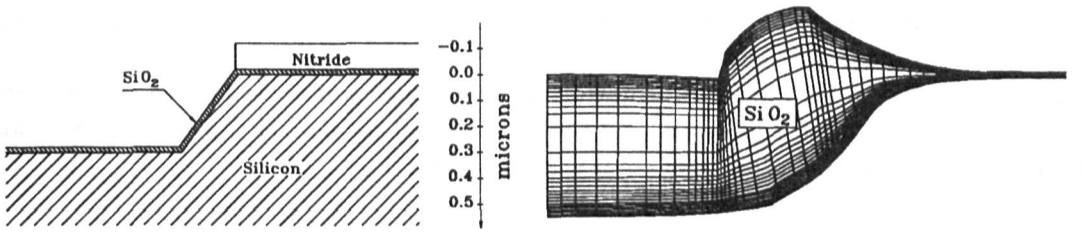

Figure 4: Recessed LOCOS: initial structure (left), and final oxide shape after 90 min oxidation at 1000°C in wet ambient (right).

The quality of the generated grid is judged upon the convergence properties of the solution of the discretized physical equations. The time for the grid generation is negligible in all cases. In Fig. 5 the relative computation time for the solution with respect to the algebraic case is shown. The elliptic grid saves about 50% of cpu time, if at least 5 iterations are used. With suitable parameters $\lambda_A, \lambda_O, \lambda_S$ a variational grid saves up to 70%.

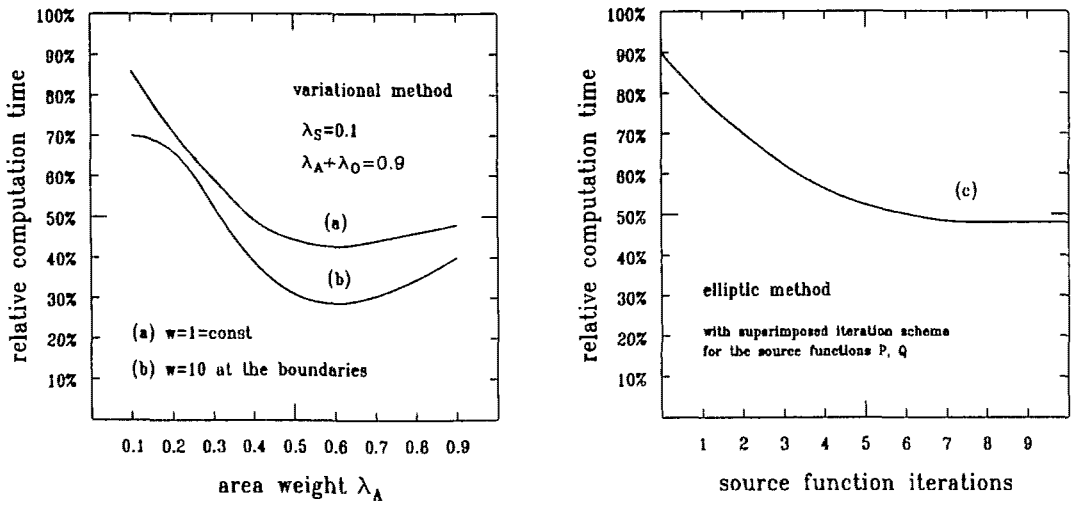

Figure 5: Relative computation time required for solving the physical equations on the different grid types (reference: algebraic grid $\hat{=}100\%$):
(a) variational grids with different area weighting parameters λ_A $(\lambda_A+\lambda_S+\lambda_O=1)$.
(b) same as (a) except for an additional orthogonality at the boundaries.
(c) elliptic grids obtained after different numbers of P, Q iterations.

From Fig. 5 and other examples it can be seen that orthogonality at the boundary and area control are most important, followed by some smoothness control, and then orthogonality in general. Note that, historically, orthogonality has received the most attention.

5 Conclusion

Transformation methods have been applied successfully to process simulation problems. The computational expense for the generation of a "high quality" grid is typically only a few seconds on commonly used workstations and results in a speed up as high as a factor 3 for the solution of the transformed physical equations.

An integral attempt for the transformation of the physical equations lead to conservative formulations of the discretized equations, and therefore global conservation properties are satisfied automatically without any additional computational effort.

Acknowledgement

This project is supported by the research laboratories of: AUSTRIAN INDUSTRIES - AMS International at Unterpremstätten, Austria; DIGITAL EQUIPMENT Corporation at Hudson, USA; SIEMENS Corporation at Munich, Germany; and SONY Corporation at Atsugi, Japan.

References

[1] H. Umimoto, and S. Odanaka, *Three-Dimensional Numerical Simulation of Local Oxidation of Silicon*, IEEE Trans. Electron Devices, ED-**38**, pp. 505–511, 1991

[2] A. Hilgenstock, *A Fast Method for the Elliptic Generation of Three-Dimensional Grids with Full Boundary Control*, On Numerical Grid Generation in Computational Fluid Dynamics, S. Senguta, J. Häuser, P.R. Eiseman, and J.F. Thompson (Eds.), Pineridge Press, Swansea, U.K., pp. 137–146, 1988

[3] J.F. Thompson, A.U.A. Warsi, and C.W. Mastin, *Numerical Grid Generation, Foundations and Applications*, North Holland, New York, 1985.

[4] P. Pichler, W. Jüngling, S. Selberherr, E. Guerrero, and H. Pötzl, *Simulation of Critical IC-Fabrication Processes*, IEEE Trans. Electron Devices, ED-**32**, pp. 1940–1953, 1985

[5] P.R. Eiseman, and G. Erlebacher, *Grid Generation for the Solution of Partial Differential Equations*, ICASE Report No. 87-57, NASA Langley Research Center, Hampton, Virginia, 1987

[6] J.M. Ortega, and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, San Diego, 1970.

[7] J.U. Brackbill, and J.S. Saltzman, *Adaptive Zoning for Singular Problems in Two Dimensions*, J. Comp. Phys., **46**, pp. 342–368, 1982

[8] M.D. Giles, Defect-Coupled Diffusion at High Concentration, IEEE Trans. Computer-Aided Design, CAD-**8**, pp. 460–467, 1989

[9] M.Y. Tsai, F.F. Morehead, J.E.E. Balgin, and A.E. Michel, *Shallow Junctions by High-Dose As Implants in Si: Experiments and Modeling*, J. Appl. Phys., **51**, pp. 3230–3235, 1980.

[10] T.L. Crandle, W.B. Richardson, and B.J. Mulvaney, *A Kinetic Model for Anomalous Diffusion during Post-Implant Annealing*, IEEE IEDM 1988 Technical Digest, pp. 636–639, 1988

[11] G. Hobler, S. Halama, K. Wimmer, S. Selberherr, and H. Pötzl, *RTA-Simulations with the 2-D Process Simulator PROMIS*, Nupad III Technical Digest, pp. 13–14, 1990

[12] F. Lau, *Modeling of Polysilicon Diffusion Sources*, IEEE IEDM 1990 Technical Digest, pp. 737–740, 1990

Appendix: Euler-Lagrange Equations

The Euler-Lagrange Equations for the *smoothness* functional I_S (14) are (26)–(27) using the abbreviations (28)–(29). For $A^2 - BC \neq 0$ they reduce to (30)–(31).

$$B(dx_{uu} - 2ex_{uv} + fx_{vv}) - A(dy_{uu} - 2ey_{uv} + fy_{vv}) = 0 \tag{26}$$

$$A(dx_{uu} - 2ex_{uv} + fx_{vv}) - C(dy_{uu} - 2ey_{uv} + fy_{vv}) = 0 \tag{27}$$

$$A = x_u y_u + x_v y_v \qquad B = y_u^2 + y_v^2 \qquad C = x_u^2 + x_v^2 \tag{28}$$

$$d = (x_v^2 + y_v^2)/g^3 \quad e = (x_u x_v + y_u y_v)/g^3 \quad f = (x_u^2 + y_u^2)/g^3 \tag{29}$$

$$dx_{uu} - 2ex_{uv} + fx_{vv} = 0 \tag{30}$$

$$dy_{uu} - 2ey_{uv} + fy_{vv} = 0 \tag{31}$$

As Euler-Lagrange equations for the *area* functional I_A (15) we obtain (32)–(33).

$$(x_u - x_v)g + g_v x_u - g_u x_v = 0 \tag{32}$$

$$(y_v - y_u)g + g_u y_v - g_v y_u = 0 \tag{33}$$

The Euler-Lagrange Equations for the *orthogonality* functional with included boundary control I_O (16) are (34)–(35) using the substitutions (36)–(40).

$$w \cdot (b_1 x_{uu} + b_2 x_{uv} + b_3 x_{vv} + a_1 y_{uu} + a_2 y_{uv} + a_3 y_{vv}) = -r_1 \tag{34}$$

$$w \cdot (a_1 x_{uu} + a_2 x_{uv} + a_3 x_{vv} + c_1 y_{uu} + c_2 y_{uv} + c_3 y_{vv}) = -r_2 \tag{35}$$

$$a_1 = x_v y_v \qquad b_1 = x_v^2 \qquad c_1 = y_v^2 \tag{36}$$

$$a_2 = x_u y_v + x_v y_u \quad b_2 = 2(2x_u x_v + y_u y_v) \quad c_2 = 2(x_u x_v + 2y_u y_v) \tag{37}$$

$$a_3 = x_u y_u \qquad b_3 = x_u^2 \qquad c_3 = y_u^2 \tag{38}$$

$$r_1 = (w_u y_v - w_v y_u) \cdot (x_u x_v + y_u y_v)^2 / 2g \tag{39}$$

$$r_2 = (w_v x_u - w_u x_v) \cdot (x_u x_v + y_u y_v)^2 / 2g \tag{40}$$