Engineering with Computers (1999) 15: 263–268

Engineering with Computers

Winslow Smoothing on Two-Dimensional Unstructured Meshes

P. M. Knupp

Parallel Computing Sciences Department, Sandia National Laboratories[1], Albuquerque, NM, USA

Abstract. *The Winslow equations from structured elliptic grid generation are adapted to smoothing of two-dimensional unstructured meshes using a finite difference approach. We use a local mapping from a uniform N-valent logical mesh to a local physical subdomain. Taylor Series expansions are then applied to compute the derivatives which appear in the Winslow equations. The resulting algorithm for Winslow smoothing on unstructured triangular and quadrilateral meshes gives generally superior qualilty than traditional Laplacian smoothing, while retaining the resistance to mesh folding on structured quadrilateral meshes.*

Keywords. Elliptic smoothing; Laplacian smoothing; Quadrilateral meshing; Unstructured grid generation; Winslow equations

1. Introduction

The Winslow elliptic smoother has been used for many years in two-dimensional structured mesh generation because of its connection to harmonic maps between manifolds for which a one-to-one guarantee is proven. The smoother is based on solving the well known partial differential equations

$$g_{22}x_{\xi\xi} - 2\, g_{12}\, x_{\xi\eta} + g_{11}\, x_{\eta\eta} = 0 \tag{1}$$

$$g_{22}\, y_{\xi\xi} - 2\, g_{12}\, y_{\xi\eta} + g_{11}\, y_{\eta\eta} = 0 \tag{2}$$

as a boundary-value problem [1,2]. The equations are obtained by requiring the logical variables ξ and η to be harmonic functions, and by interchanging the dependent and independent variables in the corresponding Laplace equations.

The purpose of this paper is not to compare Winslow and Laplacian smoothing, but rather to present a logical extension of Winslow smoothing to unstructured meshes. Many comparisons between Winslow smoothing and Laplacian smoothing on structured meshes have been performed, and there is a widely held concensus that the former is the smoother of choice, mainly due to its resistance to grid folding. Despite the effectiveness of this mesh generator in structured meshing, few attempts have been made to extend its applicability to the realm of unstructured meshes. Instead, Laplacian smoothing predominates on unstructured meshes due to its generality and to its ease of implementation. It is the author's belief that the additional work of implementing Winslow smoothing is worth the effort to achieve robustness against mesh folding.

In Laplacian smoothing, node positions are the average of positions of the M neighboring nodes:

$$\mathbf{x}_n = \frac{1}{M}\sum_{m=0}^{M-1}\mathbf{x}_m \tag{3}$$

Laplacian smoothing on structured meshes can be shown to arise from solving the following pair of partial differential equations [3]:

$$x_{\xi\xi} + x_{\eta\eta} = 0 \tag{4}$$

$$y_{\xi\xi} + y_{\eta\eta} = 0 \tag{5}$$

Although Laplacian smoothing is easy to implement, its usefulness is limited by the fact that it sometimes results in mesh folding/spillover. No guarantee against folding can be constructed for Eqs (4) and (5). The fact that Laplacian smoothing on structured meshes can be extended to unstructured meshes strongly suggests that it should be possible to extend Winslow smoothing to unstructured meshes as well.

In his original paper, Winslow [1] solved Eqs (1) and (2) on a structured triangular mesh having six-valent nodes. Others soon extended the procedure to structured quadrilateral meshes having four-valent nodes [2]. To extend Winslow to unstructured

---

[1]Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-ACO4–94AL85000.

Correspondence and offprint requests to: Dr P. M. Knupp, Parallel Computing Sciences Department, Sandia National Laboratories, M/S 0441, PO Box 5800, Alburquerque, NM 87185, USA. Email: pknupp@sandia.gov

meshes clearly requires letting go of the idea of a global mapping. This point was grasped by mesh researchers such as Tipton [4], Allievi [5] and Hagemeier [6]. In spite of their insight, Winslow remains largely unused by the unstructured meshing community, although some *ad hoc* attempts at extending Winslow have been made [7].

Winslow smoothing on unstructured *quadrilateral* meshes is emphasised due to the author's present association with Sandia's CUBIT project. However, an extension to smoothing of unstructured triangular meshes is briefly outlined.

In Section 2 of this paper, equations are derived that give the local discrete approximation of the Winslow equations on an unstructured mesh. Section 3 presents implementation issues, demonstrates results on several CUBIT test problems, and discusses the extension to triangular meshes. Conclusions are given in Section 4.

2. Winslow Local Discretisation on an Unstructured Quadrilateral Mesh

The natural approach to solving Eqs (1) and (2) on unstructured meshes is to apply the Finite-Element Method (FEM). Tipton does this for 3D Winslow in a proprietary DOE paper [4]. An alternative approach outlined in this paper is to use finite differences, expanding derivatives about each mesh node in a Taylor Series and solving the resulting non-linear coupled system of equations. This is not meant to suggest that the finite difference approach given here is superior to a FEM approach. The choice merely reflects the fact that the author is more familiar with the former method.

For unstructured meshes, the most commonly used smoother is Laplacian smoothing. For a given mesh node $\mathbf{x}_n$ a discrete operator D_n from $R^3 \to R^3$ can be formulated

$$D_n\mathbf{x} = M\mathbf{x}_n - \sum_{m=0}^{M-1} \mathbf{x}_m \tag{6}$$

where the sum is over the M neighboring nodes. To smooth the mesh, one iterates on the set of equations $D_n\mathbf{x} = 0$ until some tolerance is satisfied. A similar operator is sought such that in the limit of small mesh size, Eqs (1) and (2) will be locally approximated at each node of the mesh.

To derive a discrete operator for Eqs (1) and (2) using finite differences, the two first derivatives and the three second derivatives at each node $\mathbf{x}_n$ with valency $M \geq 3$ must be approximated. First consider only the 'edge' nodes, i.e. the nodes $\mathbf{x}_m$ adjacent to $\mathbf{x}_n$. Let a local, discrete, uniform logical space (ξ_m, η_m) be given by

$$\xi_m = \cos\,\theta_m \tag{7}$$

$$\eta_m = \sin\,\theta_m \tag{8}$$

where

$$\theta_m = \frac{2\pi m}{M} \tag{9}$$

with $m = 0,1,\ldots M-1$. The following well known formula involving non-negative integral powers of the roots of unity is applied:

$$\sum_{m=0}^{M-1} (\exp^{i\theta_m})^k = \begin{cases} M & \text{k = 0,M,2M,...} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Explicit results for this formula are given for $k = 1,2,3,$ and 4 in the Appendix.

Assume that there exists a smooth, C^∞ function $f(\xi,\eta)$ on the local logical space, and that one can approximate $f(\xi_m,\eta_m)$ about $f(0,0)$ by the Taylor Series Expansion:

$$\begin{aligned} f(\cos\,\theta_m,\, \sin\,\theta_m) = \sum_{l=1}^{L-1} \frac{1}{(l-1)!} \bigg(&\cos\,\theta_m \frac{\partial}{\partial\xi} \\ &+ \sin\,\theta_m \frac{\partial}{\partial\eta}\bigg)^{l-1} f\Big|_{(0,0)} \\ &+ \frac{1}{L!}\bigg(\cos\,\theta_m \frac{\partial}{\partial\xi} \\ &+ \sin\,\theta_m \frac{\partial}{\partial\eta}\bigg)^{L} f\Big|_{(\tau\cos\theta_m,\tau\sin\theta_m)} \end{aligned} \tag{11}$$

for some $0 \leq \tau \leq 1$.

To arrive at expressions for the first and second derivatives of f at (0,0), multiply the Taylor Series expansion above by various combinations of $\cos\,\theta_m$ and $\sin\,\theta_m$, sum from $m = 0$ to $m = M-1$, and use the identities in the Appendix. Consider only $k \leq 4$ in order to neglect higher-order terms involving third and higher derivatives. For $M \geq 5$ the resulting approximations are:

$$f_\xi = \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_0)\cos\,\theta_m \tag{12}$$

$$f_\eta = \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_0)\sin\,\theta_m \tag{13}$$

$$f_{\xi\xi} = \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_0)(4\cos^2\theta_m - 1) \tag{14}$$

$$f_{\xi\eta} = \frac{8}{M} \sum_{m=0}^{M-1} (f_m - f_0) \cos\theta_m \sin\theta_m \tag{15}$$

$$f_{\eta\eta} = \frac{2}{\mathrm{M}} \sum_{m=0}^{M-1} (f_m - \mathrm{f}_0)(4\sin^2\theta_m - 1) \tag{16}$$

For $\mathrm{M} = 3$ and 4 there is not enough information in the 'edge' nodes to uniquely determine some of the required derivatives. For example, when $M = 4$, $f_{\xi\eta}$ cannot be approximated by the 'edge' nodes, and so the 'diagonal' nodes (those four nodes $\hat{\mathbf{x}}_m$ lying on the opposite corners of the four quadrilaterals) are included, increasing the valence to eight. To construct a logical space for the diagonal nodes assume that the logical quadrilaterals form rhombii. Then $\xi_m = L_M \cos\hat{\theta}_m$, $\eta_m = L_M \sin\hat{\theta}_m$, where $L_M^2 = 2(1 + \cos 2\pi/M)$ and

$$\hat{\theta}_m = \frac{2\pi(m + \frac{1}{2})}{M} \tag{17}$$

The motivation for this construction is, of course, to get the usual second-order accurate centered-difference approximation to $f_{\xi\eta}$ when $M = 4$. The result for $M = 4$ is then

$$f_\xi = \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_0) \cos\theta_m \tag{18}$$

$$f_\eta = \frac{2}{M} \sum_{m=0}^{M-1} (f_m - f_0) \sin\theta_m \tag{19}$$

$$f_{\xi\xi} = \frac{4}{M} \sum_{m=0}^{M-1} (f_m - f_0) \cos^2\theta_m \tag{20}$$

$$f_{\xi\eta} = \frac{2}{M} \sum_{m=0}^{M-1} (\hat{f}_m - \hat{f}_0) \cos\hat{\theta}_m \sin\hat{\theta}_m \tag{21}$$

$$f_{\eta\eta} = \frac{4}{M} \sum_{m=0}^{M-1} (f_m - f_0) \sin^2\theta_m \tag{22}$$

where the 'hat' notation refers to evaluations on the diagonal nodes.

For $\mathrm{M} = 3$, it is impossible to get the derivatives $f_{\xi\xi}$ and $f_{\eta\eta}$ if only the 'edge' nodes are used. For this case, use the diagonal nodes of the three adjacent quadrilaterals to augment the nodes to a set of six nodes. Then Eqs (12)–(16) can be applied but with $\mathrm{M} = 6$. The result is exactly the set of formulas used by Winslow [1]. In summary, the first and second derivatives of f may be approximated on an unstructured, quadrilateral mesh. Low-valent nodes with $M = 3$ and $M = 4$ require use of the 'diagonal' nodes, while higher valent nodes can be approximated without using the 'diagonal' nodes.

The discrete Winslow operator can now be constructed. Let

$$D_n\mathbf{x} = G_{22} D_{\xi\xi}\mathbf{x}_n - 2G_{12} D_{\xi\eta}\mathbf{x}_n + G_{11} D_{\eta\eta}\mathbf{x}_n \tag{23}$$

with

$$G_{11} = D_\xi\mathbf{x}_n \cdot D_\xi\mathbf{x}_n \tag{24}$$

$$G_{12} = D_\xi\mathbf{x}_n \cdot D_\eta\mathbf{x}_n \tag{25}$$

$$G_{22} = D_\eta\mathbf{x}_n \cdot D_\eta\mathbf{x}_n \tag{26}$$

and

$$D_\xi\mathbf{x}_n = \frac{2}{M} \sum_{m=0}^{M-1} (\mathbf{x}_m - \mathbf{x}_n) \cos\theta_m \tag{27}$$

$$\mathrm{D}_\eta\mathbf{x}_n = \frac{2}{\mathrm{M}} \sum_{m=0}^{M-1} (\mathbf{x}_m - \mathbf{x}_n) \sin\theta_m \tag{28}$$

The second derivatives are constructed similarly, but depend on whether or not $M = 4$.

3. Implementation and Applications

The method outlined in the previous section was implemented in C++ within the CUBIT code [8]. Because CUBIT uses solid model geometry, the equations were solved using vectors in R^3 and the nodes moved to the 'owning' surfaces. One thus solves

$$g_{22}\,\mathbf{x}_{\xi\xi} - 2g_{12}\,\mathbf{x}_{\xi\eta} + g_{11}\,\mathbf{x}_{\eta\eta} = 0 \tag{29}$$

where $\mathbf{x} \in R^3$.

To correctly apply the formula of the previous section one must ensure that the adjacent nodes are consistently ordered (either clockwise or counterclockwise) with respect to the owning surface. A special routine within CUBIT was written to do this. To make this approach compatible with CUBIT's paving/cleanup algorithm [9], two-valent nodes were specially handled by doing Laplacian smoothing.

Figure 1 shows the results of Winslow smoothing on an annular surface in which the upper left-hand side has been meshed with a structured mesh, and the lower right-hand side meshed using paving. The large cell aspect-ratios on the smoothed structured mesh may strike some readers as not particularly good. Recall, however, that mesh quality is physics-dependent, as well as geometry-dependent, i.e. the application determines whether or not the mesh is

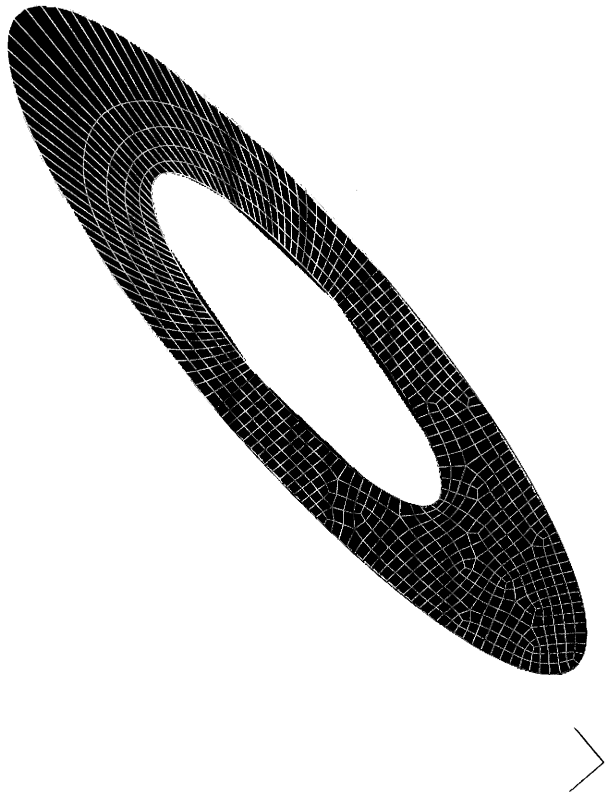

Fig. 1. Winslow smoothing on structured and unstructured mesh.

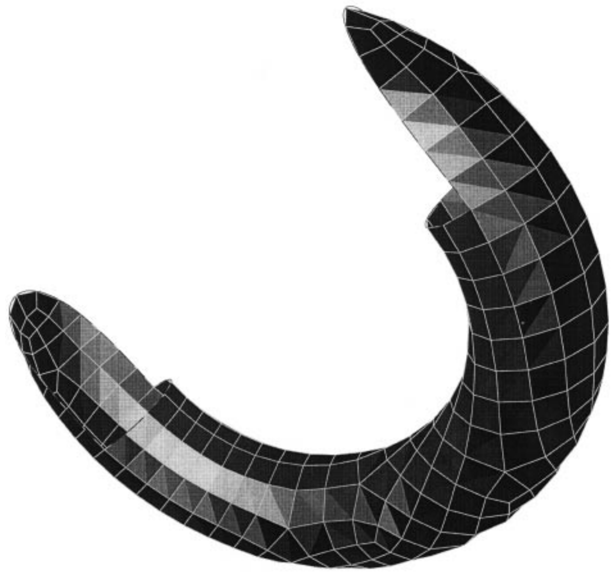

Fig. 2. Winslow smoothing on unstructured mesh on a torus.

acceptable. If small aspect-ratios are wanted, then the Winslow-smoothed paved mesh on the lower right of Fig. 1 is more appropriate. Readers familiar with Winslow smoothing on structured meshes will consider the result in the upper left to be a good mesh compared to that produced by Laplacian smoothing – the latter is well-known to produce badly folded meshes if a structured mesh is used on a sufficiently stretched version of the geometry [10]. A paved mesh on a toroidal surface was Winslow-smoothed in Fig. 2. These examples alone do not demonstrate the robustness of Winslow smoothing against folding, but a year's experience with this smoother in CUBIT has demonstrated that the property exhibited on structured meshes has been sucessfully carried over to unstructured quadrilateral meshes. If mesh density is too coarse, truncation error can lead to Winslow producing folded meshes despite the existence of a guarantee for the continuum global mapping [10]. The author has observed this on very coarse paved meshes and also at the center of CUBIT's Circle primitive. Winslow smoothing on structured meshes is derived by solving an elliptic boundary value problem. As a result, the Winslow guarantee does not extend to the mesh at the boundary and folded cells on the boundary are not precluded. In the present implementation, boundary nodes are not smoothed.

The author has implemented Winslow for unstructured triangular meshes, using a modified version of the method in Section 2. Formulas (12)–(16) in that section will apply provided any node in the triangular mesh has a valence of 5 or greater (this includes Winslow's six-valent case of a regular triangular mesh). If a node in the triangular mesh is 3- or 4-valent, valence can be increased by using nodes opposite the edges of the triangles that are opposite the smooth node. The only case for which this cannot be done is if the node is near the boundary; then the opposite nodes may not exist. For certain degenerate cases, such as that of a 3-valent node adjacent to the corner of the domain, one may not be able to sufficently increase the valence by grabbing adjacent nodes. Laplacian smoothing can be applied to those particular nodes. As an example, Figs 3 and 4 compare the results of Laplacian and Winslow smoothing on a non-uniform, unstructured triangular mesh. The most obvious feature in the Laplacian-smoothed mesh is the stretching of the elements away from the curved boundary. This stretching does not occur in the Winslow-smoothed mesh. Indeed, the maximum aspect ratio over all triangles in the Laplacian-smoothed mesh is 2.0, compared to 1.6 for the Winslow-smoothed mesh. The maximum angles are 129° and 118° for Laplace and Winslow, respectively, while the minimum angles are 17.6° and 24.4°, respectively.

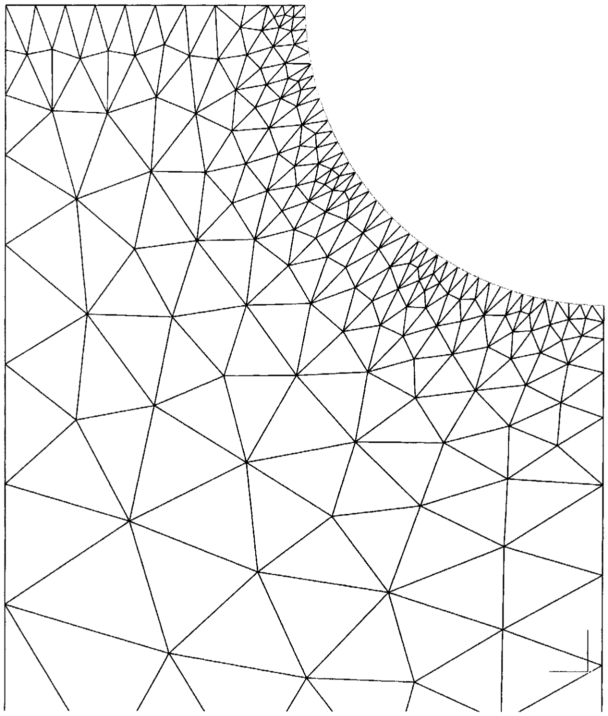

Fig. 3. Laplace smoothing on non-uniform triangle mesh.

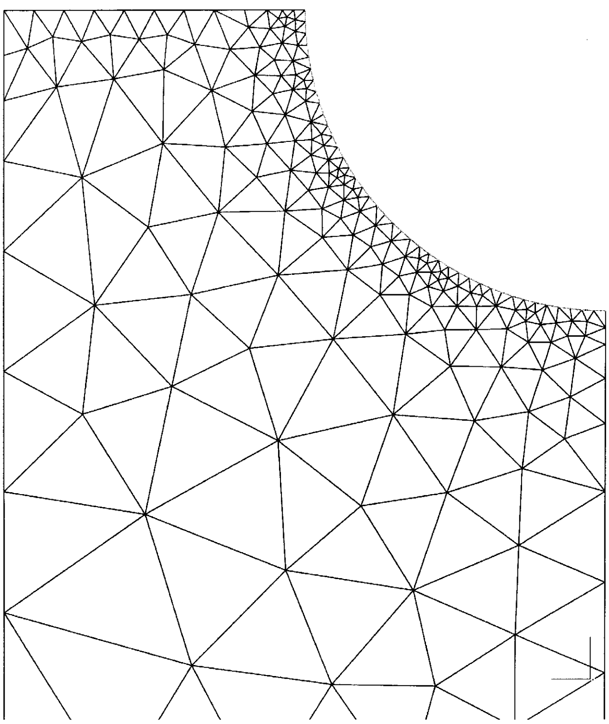

Fig. 4. Winslow smoothing on non-uniform triangle mesh.

4. Summary and Conclusions

Winslow smoothing from structured meshing has been extended to two-dimensional unstructured meshes. Experience with this extension of Winslow within the CUBIT code strongly suggests that resistance to mesh folding has been carried over to the unstructured case. Furthermore, although the computations outlined here are more expensive than those needed for Laplacian smoothing, we have found that the overall CPU time involved with Winslow is slightly less than with Laplacian smoothing. This has been observed in smoothing of structured meshes as well and suggests that the rate of convergence of the iterative smoothing method is larger for Winslow. It is not clear that one can extend the finite difference approach outlined in this paper to three-dimensions. An extension to three-dimensional unstructured meshes using FEM has already been done in [4]. Unfortunately, experience suggests that resistance to mesh folding is not as strong in three-dimensional Winslow smoothing. The author has successfully extended two-dimensional Winslow smoothing to a weighted smoothing scheme for anisotropic smoothing [11].

Acknowledgements

The author wishes to thank Stanly Steinberg, Tim Tautges, David White and the rest of the CUBIT team.

References

1. Winslow, A. (1967) Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh. J Comp Phys, 2, 149–172
2. Thompson, J., Thames, F., Mastin, C. (1974) Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. J Comp Phys, 15, 299–319
3. Knupp, P., Steinberg, S. (1993) The Fundamentals of Grid Generation, Boca Raton FL, CRC Press
4. Tipton, R. (1990) Grid Optimization by Equipotential Relaxation, unpublished manuscript, Lawrence Livermore National Laboratory
5. Allievi A., Calisal, S. (1992) Application of Bubnov-Galerkin Formulation to Orthogonal Grid Generation, J Comp Phys, 98, 163–173
6. Hagmeier, R., (1994) Personal communication, Swansea, UK
7. Khamayseh A., Kuprat, A. (1996) Anisotropic Smoothing and Solution Adaption for Unstructured Grids. Int J Num Meth Eng, 39, 3163–3174
8. Blacker, T. et al. (1994) CUBIT Mesh Generation

Environment, Volume 1: User's Manual. SAND94–1100, Sandia National Laboratories, Albuquerque, NM
9. Blacker, T., Stephenson, M. (1991) Paving: A New Approach to Automated Quadrilateral Mesh Generation. Int J Numer Methods Eng, 32, 811–847
10. Knupp, P., Luczak, R. (1995) Truncation Error in Grid Generation: A Case Study. Num Methods for PDEs, 11, 561–571
11. Knupp, P. (1999) Applications of Mesh Smoothing: Copy, Morph, and Sweep on Unstructured Quadrilateral Meshes. Vol. 45, Issue 1, 37–45. IJNME.

Appendix: Expansions of Formula (10)

$K = 1,\ M \geq 3$:

$$\sum_{m=0}^{M-1} \cos\ \theta_m = 0 \tag{30}$$

$$\sum_{m=0}^{M-1} \sin\ \theta_m = 0 \tag{31}$$

$k = 2,\ M \geq 3$:

$$\sum_{m=0}^{M-1} \cos^2\ \theta_m = \frac{M}{2} \tag{32}$$

$$\sum_{m=0}^{M-1} \cos\ \theta_m \sin\ \theta_m = 0 \tag{33}$$

$$\sum_{m=0}^{M-1} \sin^2\ \theta_m = \frac{M}{2} \tag{34}$$

$k = 3,\ m = 3$:

$$\sum_{m=0}^{M-1} \cos^3\ \theta_m = \frac{M}{4} \tag{35}$$

$$\sum_{m=0}^{M-1} \cos^2\ \theta_m \sin\ \theta_m = 0 \tag{36}$$

$$\sum_{m=0}^{M-1} \cos\ \theta_m \sin^2\ \theta_m = -\frac{M}{4} \tag{37}$$

$$\sum_{m=0}^{M-1} \sin^3\ \theta_m = 0 \tag{38}$$

If $k = 3$ and $M \geq 4$, then the four sums (35)–(38) are all zero.

$k = 4,\ M \geq 3,\ M \neq 4$:

$$\sum_{m=0}^{M=1} \cos^4\ \theta_m = \frac{3M}{8} \tag{39}$$

$$\sum_{m=0}^{M=1} \cos^3\ \theta_m \sin\ \theta_m = 0 \tag{40}$$

$$\sum_{m=0}^{M=1} \cos^2\ \theta_m \sin^2\ \theta_m = \frac{M}{8} \tag{41}$$

$$\sum_{m=0}^{M=1} \cos\ \theta_m \sin^3\ \theta_m = 0 \tag{42}$$

$$\sum_{m=0}^{M=1} \sin^4\ \theta_m = \frac{3M}{8} \tag{43}$$

$k = 4,\ m = 4$:

$$\sum_{m=0}^{M=1} \cos^4\ \theta_m = \frac{M}{2} \tag{44}$$

$$\sum_{m=0}^{M=1} \cos^3\ \theta_m \sin\ \theta_m = 0 \tag{45}$$

$$\sum_{m=0}^{M=1} \cos^2\ \theta_m \sin^2\ \theta_m = 0 \tag{46}$$

$$\sum_{m=0}^{M=1} \cos\ \theta_m \sin^3\ \theta_m = 0 \tag{47}$$

$$\sum_{m=0}^{M=1} \sin^4\ \theta_m = \frac{M}{2} \tag{48}$$