NISTIR 7402
June, 2007

From B-Spline Mesh Generation to Effective Visualizations for the NIST Digital Library of Mathematical Functions

Bonita Saunders and Qiming Wang

Abstract. We discuss the use of tensor product B-splines to generate grids, or meshes, to facilitate the plotting of high level mathematical functions for the National Institute of Standards and Technology (NIST) Digital Library of Mathematical Functions Project. The plot data is placed inside a web based format such as VRML (Virtual Reality Modeling Language) to create interactive visualizations that allow users to carefully examine complicated function features such as zeros, branch cuts, poles and other singularities. We discuss the grid generation technique and look at its effectiveness in creating accurate visualizations.

§1. Introduction

The National Institute of Standards and Technology (NIST), a U.S. government agency under the Department of Commerce, is in the midst of developing the NIST Digital Library of Mathematical Functions (DLMF), a web-based collection of high level, or special, mathematical functions that will replace the widely used, but outdated, National Bureau of Standards Handbook of Mathematical Functions published in 1964 [1]. The website will include formulas, methods of computation, references, and links to software for over forty functions. It will feature interactive navigation, a mathematical equation search, 2D graphics, and dynamic interactive 3D visualizations.

This paper discusses our use of a mesh generation technique to facilitate the plotting of functions for the 3D visualizations. Often, the complicated nature of a special function is clearly indicated by its domain, which may be irregular, discontinuous, or multiply connected. By modifying an algebraic tensor product spline mesh generation algorithm that we

Conference Title
Editors pp. 1–6.
Copyright ©2005 by Nashboro Press, Brentwood, TN.
ISBN 0-0-9728482-x-x
All rights of reproduction in any form reserved.

originally designed for problems in aerodynamics and solidification theory, we were able to create boundary/contour fitted computational grids that allowed us to capture key function features such as zeros, poles, branch cuts and other singularities.

Although commercial packages often have many built-in special functions, their 3D plots are usually over a rectangular Cartesian mesh, leading to poor and misleading graphs. Also, when function values lie outside the range of interest, many packages have trouble properly clipping the function surface to produce an accurate graphical representation. Furthermore, even when the plot looks satisfactory inside a package, it may be completely unacceptable when the data is transformed to a format more suitable for display on the web.

We examine these problems and others that arise when trying to display dynamic 3D graphs of special functions on the web. We show how our mesh generation algorithm eliminates or lessens the severity of many problems, talk about progress in making the method more adaptive and discuss other possibilities for improvements.

§2. Constructing 3D Graphics for a Digital Library

The NIST DLMF will consist of approximately forty chapters authored by special function experts throughout the U. S. and abroad. The number and location of graphics for each chapter is determined by consultation with the authors and DLMF editors. The accuracy of the plotting data is being verified by plotting the function with at least two different methods, using commercial packages, publicly available codes, or the author's personal codes. A key concern, however, is whether or not the displayed plot accurately represents the graph of the function.

Commercial packages typically render 2D plots very well. Discontinuities are handled automatically or fairly easily with special options. If a user wants to restrict the vertical range of the function, the package properly cuts, or clips, the curve so that only points within the desired range appear. This is often not the case in 3D. Figure 1 illustrates some of the problems we have encountered when trying to use commercial packages. The first figure shows a plot of $1/|\Gamma(z)|$, where $z = x + iy$, rendered using a popular commercial package. The user has requested that the function only be plotted for values less than or equal to 6. The package produces a flat area, or shelf, where values greater than 6 are set to 6. Although the shelf-like area, which has nothing to do with the function, may not concern many users, it might be confusing to students and others unfamiliar with the function. The user can request that the shelf not be shown, but this produces a saw-tooth area which can be even more misleading. Although seasoned users can use alternative commands to successfully clip the function properly, we found that this was still not sufficient for our

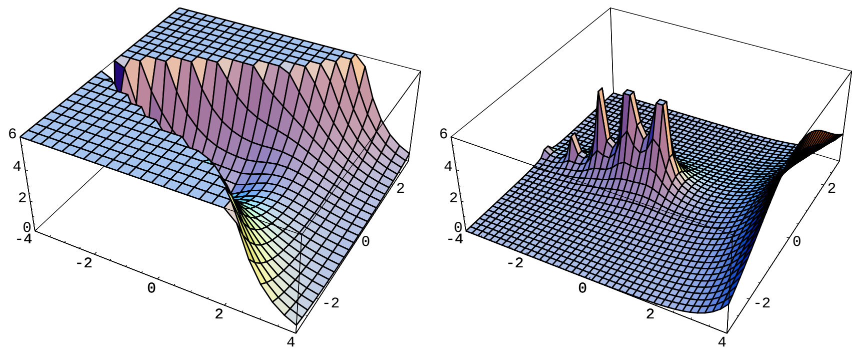

Fig. 1. Potential problems such as bad clipping and poor resolution of poles.

requirements. The problem is that the computation of the function is still over a rectangular Cartesian mesh. The figure looks fine when viewed inside the package, but when we put the data into our web format, the rectangular mesh cells produce a very irregular color map. For our web visualizations, we use the Virtual Reality Modeling Language (VRML), a standard 3D file format for creating interactive web-based visualizations [6]. The second figure, rendered using the same package, shows the modulus of the complex gamma function, $|\Gamma(z)|$, $z = x + iy$. Both figures can be improved by using a much larger number of mesh points, but large data files hamper the interactive features of the web-based VRML visualizations. The rendering problems can be eliminated or decreased in severity by computing the function over a specially designed boundary/contour fitted mesh. The next section discusses the tensor product B-spline mapping we have used to produce such meshes.

§3. Grid Generation Mapping

For our application, the difficulty of the mesh generation problem depends on the complexity of the computational domain for the special function. The domains range from simple rectangles to complicated multiply connected domains with branch cuts. Our primary mesh generation technique is based on an algorithm developed by one of the authors for meshes to be used in solving partial differential equations (pdes) related to aerodynamics and solidification theory [4], [5]. The algorithm uses an algebraic grid generation system defined by a mapping $\mathbf{T}$ from the unit square I_2 to a physical domain of arbitrary shape. Specifically, we let

$$\mathbf{T}(\xi,\eta) = \begin{pmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{ij} B_{ij}(\xi,\eta) \\ \sum_{i=1}^{m}\sum_{j=1}^{n} \beta_{ij} B_{ij}(\xi,\eta) \end{pmatrix}, \tag{1}$$

where $0 \le \xi, \eta \le 1$ and each B_{ij} is the tensor product of cubic B-splines. Hence, $B_{ij}(\xi, \eta) = B_i(\xi)B_j(\eta)$ where B_i and B_j are elements of cubic B-spline sequences associated with finite nondecreasing knot sequences, say, $\{s_i\}_1^{m+4}$ and $\{t_j\}_1^{n+4}$, respectively. For simple boundaries the coefficients α_{ij} and B_{ij} can be chosen to produce a very good approximation to a common algebraic generation technique, transfinite blending function interpolation. In short, this means the coefficients are obtained by evaluating $\mathbf{T}$ at average knot values as discussed in [2]. This shape preserving approximation reproduces straight lines and preserves convexity [2] . For more complicated or highly nonconvex boundaries we can turn the technique into a variational method with the coefficients chosen to minimize the functional

$$F = \int_{I_2} \left(w_1 \left\{ \left(\frac{\partial J}{\partial \xi}\right)^2 + \left(\frac{\partial J}{\partial \eta}\right)^2 \right\} + w_2 \left\{ \frac{\partial \mathbf{T}}{\partial \xi} \cdot \frac{\partial \mathbf{T}}{\partial \eta} \right\}^2 + w_3 J u \right) dA \quad (2)$$

where $\mathbf{T}$ denotes the grid generation mapping, J is the Jacobian of the mapping, w_1, w_2 and w_3 are weight constants, and u represents external criteria for adapting the grid. Like the variational method of Brackbill and Saltzman [3], the integral controls mesh smoothness, orthogonality, and depending on the definition of u, the adaptive concentration of the grid lines. Hence, when solving pdes, u might be the gradient of the evolving solution or an approximation of truncation error. Ideally, for our problem, we want u to be based on curvature and gradient information related to the function surface. To avoid solving the Euler equations for the variational problem, this functional is approximated in the computer code by the sum

$$\begin{aligned} G \quad = \quad & \sum_{i,j} w_1 \left[\left(\frac{J_{i+1,j} - J_{ij}}{\triangle \xi}\right)^2 + \left(\frac{J_{i,j+1} - J_{ij}}{\triangle \eta}\right)^2 \right] \triangle\xi \triangle\eta \\ & + \sum_{i,j} w_2 Dot_{ij}^2 \triangle\xi \triangle\eta \\ & + \sum_{i,j} w_3 J_{ij} u_{ij} \triangle\xi \triangle\eta, \end{aligned} \quad (3)$$

where J_{ij} is the Jacobian value, u_{ij} is the value of u, and Dot_{ij} is the dot product of $\partial\mathbf{T}/\partial\xi$ and $\partial\mathbf{T}/\partial\eta$ at mesh point (ξ_i, η_j) on the unit square. When $w_3 = 0$, G is actually a fourth degree polynomial in each spline coefficient so the minimum can be found by using a cyclic coordinate descent technique which sequentially finds the minimum with respect to each coefficient. This technique allows the minimization routine to take advantage of the small support of B-splines when evaluating the sums that comprise G.

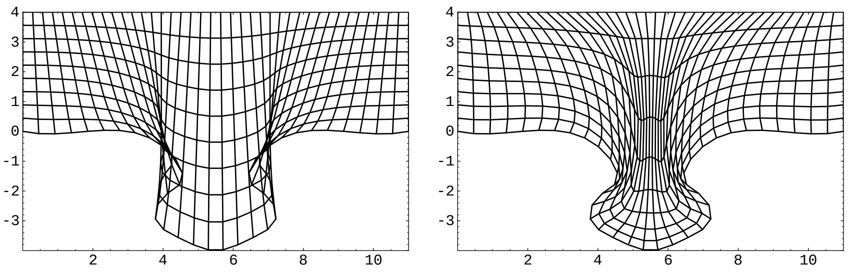

Fig. 2. Initial and optimized puzzle grids.

An application of the grid generation algorithm is shown in Figure 2 for a puzzle shaped domain. The initial grid, constructed using linear Lagrange polynomials for the blending functions, is shown on the left. Note that the grid lines overlap the nonconvex boundary. The grid on the right shows the mesh obtained after the spline coefficients are modified to minimize G. The overlapping grid lines have been pulled into the interior.

§4. Results

We have used boundary/contour fitted grid generation to produce computational grids for over one hundred 3D visualizations for the NIST DLMF. The plots in Figure 3 show the contour boundary and mesh for the function $1/|\Gamma(z)|$. To obtain the boundary we found the contour curves for $1/|\Gamma(z)| = 6$. We then connected the curves to parts of a rectangle to form a closed boundary and generated the boundary/contour fitted grid shown on the right. The picture on the left in Figure 4 shows the surface obtained after computing $1/|\Gamma(z)|$ over a purely rectangular grid with values outside the desired range set to 6. The picture on the right shows the result when we compute over our boundary/contour fitted grid. In both pictures we show a snapshot of the surface rendering obtained after the data is translated into VRML format for viewing on the web. The boundary/contour fitted grid properly clips the function and improves the smoothness of the color map.

Figure 5 shows a grid and surface for the modulus of the complex gamma function $|\Gamma(z)|$. The top and bottom halves were generated separately, with an exponential function used to concentrate the grid points near $y = 0$. The surface shown on the right illustrates how precisely the contour curves satisfying $|\Gamma(z)| = 6$ were computed. Once the coefficients are defined, we have the option of creating a coarser, or finer, grid simply be evaluating fewer, or more, points on the unit square. Also, notice that the grid spacing does not appear to be smooth in some areas. To guaran-

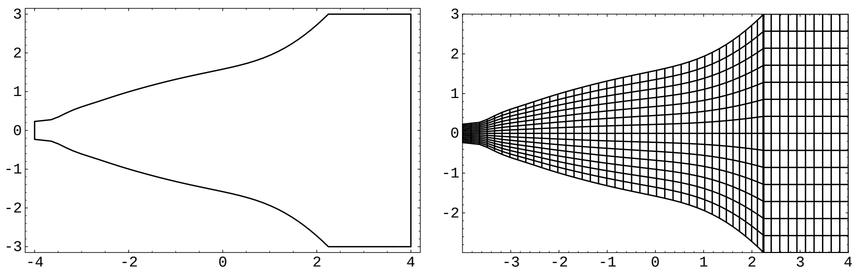

Fig. 3. Contour and grid for $1/|\Gamma(z)|$ where $z = x + iy$.

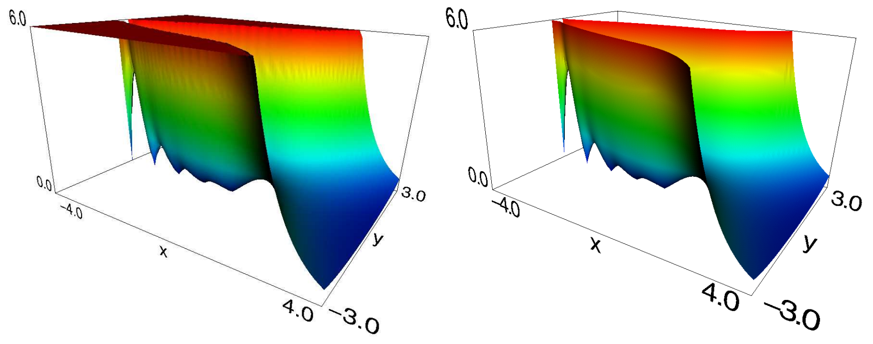

Fig. 4. Virtual Reality Modeling Language (VRML) Views of $1/|\Gamma(z)|$.

tee that key boundary or contour points, such as those at zeros or corners, are maintained regardless of grid size, we identify "fixed points," that is, boundary points that must always be kept. Grid lines are always drawn through these points. In most cases, the resulting discontinuities in cell spacing do not appear to affect the quality of the 3D visualizations.

All the visualizations in the NIST DLMF represent either real-valued or complex-valued functions of the form, $w = f(x, y)$. For complex-valued functions, the user has the option of using a height based color mapping where height $= |w|$, or a mapping based on the phase, or argument, or w. Figure 6 shows part of the computational grid and a plot of the Riemann zeta function $|\zeta(z)|$ with contour curves defined by $|\zeta(z)| = 3$ and a phase based color mapping. In Figure 7 the plot has been scaled down in the vertical direction to produce a phase density plot.

§5. Conclusions

We have successfully used boundary/contour fitted grid generation to create over one hundred visualizations of complex mathematical functions for the NIST Digital Library of Mathematical Functions Project. The grid

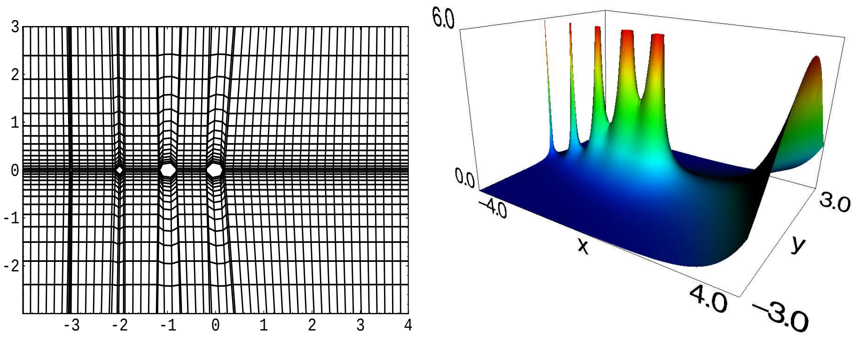

Fig. 5. Grid and Plot of Complex Gamma Function $|\Gamma(z)|$.

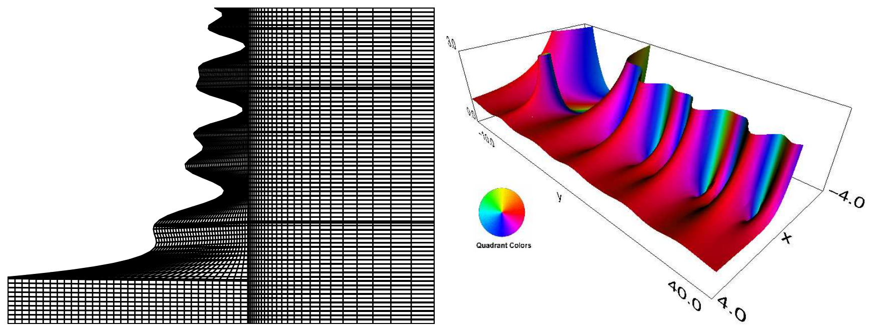

Fig. 6. Grid and Plot of Riemann Zeta Function.

generation technique has helped us address many of the problems such as poor clipping, inaccurate plots, and poor color mapping that can come with using standard commercial packages for 3D graphics. We continue to improve the technique and have made significant progress toward implementing the adaptive movement of grid lines based on external information such as function curvature and gradient data.

In Figure 8 we attract grid points in an equally spaced square grid to a circle. Currently, we are working on adaptive movement based on function curvature, but several problems must be addressed. Functional minimization routines, originally designed only for $w_3 = 0$, are being updated to accommodate a likely nonlinear dependence of u on the spline coefficients. Also, the specialized nature of many of the functions means that accessing and linking the codes needed to compute gradient and curvature data may not be a trivial task.

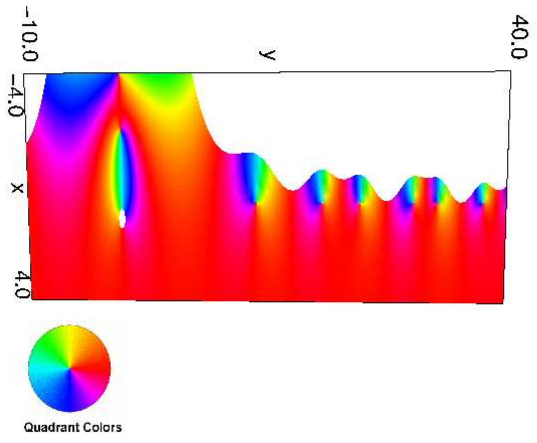

Fig. 7. Phase Density Plot of Riemann Zeta Function.

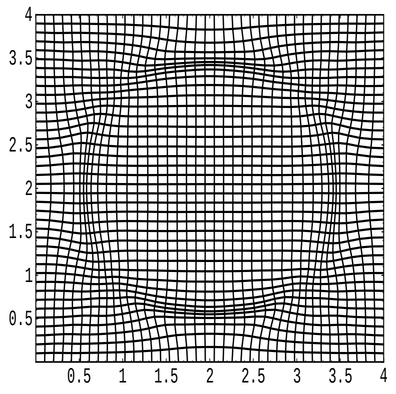

Fig. 8. Grid adapted to circular shape.

§6. Disclaimer

All references to commercial products are provided only for clarification of the results presented. Their identitication does not imply recommendation or endorsement by NIST.

References

1. Abramowitz, M. and I. A. Stegun (eds.), *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, National

Bureau of Standards Applied Mathematics Series **55**, U.S. Government Printing Office, Washington, D.C., 1964.

2. de Boor, C., *A Practical Guide to Splines*, Revised Edition, Springer-Verlag, New York, 2001.
3. Brackbill, J. U., and J. S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. **46** (1982), 342–368.
4. Saunders, B. V., A boundary conforming grid generation system for interface tracking, J. Computers Math Applic. **29** (1995), 1–17.
5. Saunders, B. V., and Q. Wang, From 2d to 3d: numerical grid generation and the visualization of complex surfaces, in *Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations*, B. K. Soni, et. al. (eds.), Whistler, British Columbia, Canada, September 25-28, 2000.
6. VRML. The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997.

Bonita V. Saunders, Qiming Wang
National Institute of Standards and Technology
Gaithersburg, Maryland, USA
`bonita.saunders@nist.gov,qiming.wang@nist.gov`