Acknowledgements

The 3D-XplorMath Consortium

An early version of this program goes back to 1987. It included a basic set of 3D graphics routines plus some routines for viewing and rotating three dimensional objects. A little later Hermann Karcher visited me from Bonn and we worked hard together on the program, making many improvements. In particular, Hermann showed me what a wonderful tool the Painter's Algorithm is—I know of no better method for getting intimately familiar with the anatomy of a complex surface in space than seeing it built up in stages from several different viewpoints.

In August of 1995 Hermann visited me again, and for two weeks we worked very hard to program the Weierstrass-Enneper technique for defining minimal surfaces into 3D-XplorMath. (I travelled to Bonn for a week the next December to return the visit and finish the job.) Karcher is one of a small group of experts in this field, and I feel very fortunate and grateful to have had this opportunity to learn his remarkably elegant and efficient numerical methods for carrying out the integrations and analytic continuations required to implement the Weierstrass-Enneper method. The many famous and beautiful minimal surfaces from the last century (those of Scherk, Riemann, Enneper, Schwarz) as well as the new ones discovered in the past decade by Chen-Gackstatter, Costa, Hoffman, Karcher, Meeks, Rosenberg, Wei and several others, are best exhibited by this

method. In fact, for most of these, the Weierstrass integrals cannot be found in closed form, so that numerical quadrature is essential. The Costa-Hoffman-Meeks surface was added and the Chen-Gackstatter surface improved when Hermann and I met at the 1998 ICM in Berlin.

It is not widely appreciated that a major problem in constructing "good" graphical representations for these surfaces—i.e., ones that suggest the correct behaviour of the surface outside of the picture, or in more mathematical terms a parametrization adapted to the ends of the minimal surface—is choosing in each case an appropriate grid along which to evaluate the Weierstrass integral. This involves starting with an equi-spaced rectangular grid, deforming the spacing so that more lines will be displayed "where interesting things are going on", and then mapping this grid conformally (often with an algebraic function that is computed by analytic continuation).

This finally produces the grid along which the complex integrals must be evaluated to map the grid as parameter lines onto the minimal surface in three-space. But even then one is not finished! Because of various singularities and branch points (caused by the fact that one is really integrating on a topologically non-trivial Riemann surface rather than on a simply-connected domain) one cannot display the entire surface this way, and one must first use this technique to create a "tile" or "fundamental domain" of the surface, and then create the entire surface by applying a sequence of reflections and rotations to this tile.

Carrying out each of the above steps requires an intimate and detailed knowledge of the conformal properties and symmetries

of these remarkable geometric objects, and I particularly appreciate Hermann's willingness to contribute freely from his vast store of knowledge on these matters, much of which he discovered himself.

Hermann's wife Traudel is a physicist, and suggested the addition of the magnetic field examples for the second order, three-dimensional ODE category. The simulated motion of a charged particle in a dipole field (e.g., the motion of an electron in the Van Allen belt) is perhaps my favorite visualization of the 3D-XplorMath repertory.

After Hermann's initial visit, I thought a lot about how to improve the program, but did not work on it for many years. Then, in mid-1993, David Eck showed me his remarkable Macintosh program shell, xWindows. It made programming in Think Object Pascal so easy and enjoyable that I was quickly able to implement all of the features I had been thinking about for so long. The "User Defined..." feature of the program is based on a powerful and speedy expression parser and evaluator, also developed by Eck. Indeed, when I needed a complex expression evaluator for the Conformal Map category, David took time out from his own work, to modify his real evaluator so it would handle complex expressions instead. I think it is fair to say that this program simply could not have been written without David's software contributions. (I am happy to report that, as of May 2003, David rejoined the 3DXM Consortium and is leading the effort to create a platform independent Java version of 3D-XplorMath called 3D-XplorMath-J.)

At about the same time, Alfred Gray was kind enough to send me a pre-publication version of his excellent book, "Mod-

ern Differential Geometry of Curves and Surfaces" (CRC Press 1993). Having available his Mathematica code for so many beautiful curves and surfaces greatly simplified the chore of choosing and programming interesting examples for this program.

The highly efficient code for performing fast Fourier transforms was written by Bob Palais. It plays a crucial role in many parts of the program.

The interface between 3D-XplorMath and the ray tracing program POVRay was the idea of Christophe Favergeon Borgialli, and he implemented the conversion program and auxiliary file needed to carry out his idea. My thanks to Christophe.

The parameterization of the pseudospherical surface corresponding to the "Breather" solution of the Sine-Gordon equation is due to Jochen Denzler.

Février 2001, j'ai passé le mois en Provence avec mon ami Patrick Iglesias. Il m'a montré comment rendre 3D-XplorMath scriptable—et a écrit aussi la plupart du code pour le faire.

Starting in 2006, Gale Paeper made very major modifications of the Pascal code to render it compatible with the GPC Pascal compiler, in order to make a Universal Binary version of 3D-XplorMath that would run native on both PowerPC and Intel Macs. Unfortunately, when Gael was nearly done illness prevented him from finishing, and it was later Adriaan van Os who started over and converted our code to work with the FreePascal compiler. After that he also helped us with finding bugs and with useful advice. Many thanks to him.