DISCRETE MODELING OF MINIMAL SURFACES

by

C. Coppin¹ and D. Greenspan

Technical Report No. 186
Department of Mathematics
University of Texas at Arlington
May, 1982

¹Expresses appreciation for support from the University of Texas at Arlington and the O'Hara Science Institute at the University of Dallas.

ABSTRACT

A new computer approach to the modeling of minimal surfaces is described and implemented. Quasi-molecular particles and forces are used to simulate actual molecular structure and forces. Detailed formulas and computer techniques are given. As an example, the approximation of the unique minimal surface through a skew quadrilateral is constructed and displayed.

1. Introduction

Soap films and minimal surfaces have long been of interest both mathematically and physically ([1], [4], [9]). The Belgian physicist J. A. F. Plateau (1801-1883) determined experimentally a number of geometric properties of soap films. His experiments involved dipping a closed, thin wire into a soap solution. When the wire was removed, the wire was spanned by a soap film or a minimal surface. Plateau concluded through his many experiments that every closed wire contour always bounds a minimal surface. Mathematically, this conclusion and questions relating to uniqueness and regularity have been and continue to be studied widely. The basic theory was related to the existence of a minimal surface and was developed in 1930-31 by J. Douglas and T. Rado'. It is known that, in general, uniqueness cannot be established because a closed wire contour may bound different soap film surfaces at different times.

The mathematical problem of determining a minimal surface, even when it is unique, is, in general, intractable. There are no analytical methods available for this purpose. It is natural, then, with the advent of modern digital computers, that numerical methods have emerged in recent years. Greenspan's method [6] and the method of Hinata, Shimasaki and Kiyono [8] are one's which minimize an appropriate functional numerically. The method of Concus [2] approximates the solution of an associated nonlinear boundary value problem.

In this paper we will develop an entirely new approach to the approximation of minimal surfaces. It will be founded on the construction of a quasi-molecular model [7] which approximates the molecular structure of a soap film.

2. Classical Molecular Mechanics.

For purposes of intuition, it will be important to review, first, how molecules interact.

Within any solid, liquid or gas, molecules interact only locally, that is, only with their nearest neighbors. This interaction is of the following general nature [5]. If two molecules are pushed together they repel each other, if pulled apart they attract each other, and mutual repulsion is of a greater order of magnitude than is mutual attraction. Mathematically, this behavior is often formulated as follows. The magnitude F of the force $\vec{F}$ between two molecules which are locally r units apart is of the form

$$F = - \frac{G}{r^p} + \frac{H}{r^q}, \quad (2.1)$$

where, typically,

$$G > 0, H > 0, q > p > 7. \quad (2.2)$$

The major problem in any simulation of a physical body is that there are too many component molecules to incorporate into the model. The classical mathematical approach is to replace the large, but finite, number of molecules by an

infinite set of points. In so doing, the rich physics of molecular interaction is lost because every point always has an infinite number of neighbors which are arbitrarily close. A viable computer alternative is to replace the large number of molecules by a much smaller number of particles and then readjust the parameters in (2.1) to compensate [7]. It is this latter approach which we will follow.

3. The Computer Algorithm.

The general idea outlined in Section 2 will be implemented in the following constructive fashion.

Consider N particles P_i , $i = 1, 2, \dots, N$, each of unit mass. For $\Delta t > 0$, let $t_k = k\Delta t$, $k = 0, 1, 2, \dots$. For each of $i = 1, 2, \dots, N$, let P_i at t_k be located at $\vec{r}_{i,k}$, have velocity $\vec{v}_{i,k}$, and have acceleration $\vec{a}_{i,k}$. Let position, velocity, and acceleration be related by the recursion formulas

$$\vec{v}_{i,1/2} = \vec{v}_{i,0} + \frac{1}{2}(\Delta t)\vec{a}_{i,0}, \quad (\text{starter formula}) \quad (3.1)$$

$$\vec{v}_{i,k+1/2} = \vec{v}_{i,k-1/2} + (\Delta t)\vec{a}_{i,k}, \quad k = 1, 2, 3, \dots \quad (3.2)$$

$$\vec{r}_{i,k+1} = \vec{r}_{i,k} + (\Delta t)\vec{v}_{i,k+1/2}, \quad k = 0, 1, 2, \dots \quad (3.3)$$

At t_k , let the force acting on P_i be $\vec{F}_{i,k}$. We relate force and acceleration by the dynamical equation

$$\vec{F}_{i,k} = m_i \vec{a}_{i,k},$$

which, by the assumption of unit mass, simplifies to

$$\vec{F}_{1,k} = \vec{a}_{1,k} . \quad (3.4)$$

As soon as the precise structure of $\vec{F}_{1,k}$ is given, then the motion of each P_1 will be determined explicitly and recursively by (3.1)-(3.4) from given initial data. The force $\vec{F}_{1,k}$ is described now as follows. Let $\vec{r}_{ij,k}$ be the vector from P_i to P_j at time t_k , so that $r_{ij,k} = |\vec{r}_{i,k} - \vec{r}_{j,k}|$ is the distance between the two particles. Then the force $\vec{F}_{ij,k}$ on P_i exerted by P_j at time t_k is assumed to be

$$\vec{F}_{ij,k} = \left(-\frac{G}{(r_{ij,k})^p} + \frac{H}{(r_{ij,k})^q} \right) \frac{\vec{r}_{ji,k}}{r_{ij,k}} , \quad (3.5)$$

which is consistent with (2.1). The total force $F_{1,k}$ on P_1 due to all other particles different from P_1 is defined by

$$\vec{F}_{1,k} = \sum_{\substack{j=1 \\ j \neq 1}}^N \left\{ \left(-\frac{G}{(r_{1j,k})^p} + \frac{H}{(r_{1j,k})^q} \right) \frac{\vec{r}_{j1,k}}{r_{1j,k}} \right\} . \quad (3.6)$$

Note finally that the introduction of an additional parameter D is essential to assure that particle interactions are local. We will require that whenever $r_{ij,k} > D$, then (3.5) must be replaced with

$$\vec{F}_{ij,k} = \vec{0} \quad (r_{ij,k} > D). \quad (3.7)$$

For the convenience of the reader, a typical FORTRAN program of the algorithm is given in the Appendix of Coppin and Greenspan [3].

4. A Prototype Minimal Surface Problem.

When a given closed wire contour is a skew quadrilateral, the Plateau problem has a unique minimal surface solution [9, p. 31]. Since a skew quadrilateral is one of the simplest contour shapes one can consider, we will consider now the Plateau problem for it. In particular, we will consider developing a model for a soap film that would span a skew quadrilateral which has one diagonal in the XY plane and whose two triangular sections which have this diagonal in common are each bent 5° with respect to the XY plane. (See Figure 4.1).

In order to know how to begin to approximate the minimal surface by a quasi-molecular set of particles, consider first a planar quadrilateral and the problem of

Figure 4.1

determining initial positions for the particles we will use to model the quadrilateral and the soap film. We desire a stable initial configuration of the particles. A good candidate for such a configuration would be a surface decomposed into equilateral triangular building blocks in which the particles are at the vertices of the triangles (See Figure 4.2).

Figure 4.2

The boundary particles will represent the soap film that adheres to the wire contour, and the interior particles will represent the soap film itself. Notice that there would be a difference which diagonal is used for the bend in converting the given plane quadrilateral into a skew quadrilateral. For, let the distance from any particle to its closest neighbors be unity, and suppose also that

the local interaction distance D satisfies $D < 1.3$. Then if the bend were at the diagonal AB, there would be no attraction or repulsion across the diagonal; however, if the bend were at CD there would be attraction or repulsion across diagonal CD. In the former choice, that is, of diagonal AB, the particles in either half of the quadrilateral would remain planar for all time. Obviously, this does not fit observation and we must make the bend at CD.

With the above ideas in mind, consider now 81 particles to model the soap film, with 32 boundary particles representing the molecules that adhere to the wire contour. (See Figure 4.3).

Figure 4.3

The equilateral building blocks have legs of unit length. Of course, the 32 boundary particles will remain fixed and the 49 interior ones will be allowed to move. All initial velocities will be taken to be zero. The initial positions of the particles will be the positions obtained if one bends the quadrilateral from particle 66 to particle 81 and elevates each half 5° , and these are given in TABLE I.

TABLE I

INITIAL POSITIONS

1	-2.98858410	0.00000000	0.26146723
2	-1.99238940	0.00000000	0.17431148
3	-0.99619469	0.00000000	0.08715574
4	0.00000000	0.00000000	0.00000000
5	0.99619469	0.00000000	0.08715574
6	1.99238940	0.00000000	0.17431148
7	2.98858410	0.00000000	0.26146723
8	-2.49048670	0.86602540	0.21788935
9	-1.49429200	0.86602540	0.13073361
10	-0.49809735	0.86602540	0.04357787
11	0.49809735	0.86602540	0.04357787
12	1.49429200	0.86602540	0.13073361
13	2.49048670	0.86602540	0.21788935
14	-1.99238940	1.73205080	0.17431148
15	-0.99619469	1.73205080	0.08715574
16	0.00000000	1.73205080	0.00000000
17	0.99619469	1.73205080	0.08715574
18	1.99238940	1.73205080	0.17431148
19	-1.49429200	2.59807620	0.13073361
20	-0.49809735	2.59807620	0.04357787
21	0.49809735	2.59807620	0.04357787
22	1.49429200	2.59807620	0.13073361
23	-0.99619469	3.46410160	0.08715574
24	0.00000000	3.46410160	0.00000000
25	0.99619469	3.46410160	0.08715574
26	-0.49809735	4.33012700	0.04357787
27	0.49809735	4.33012700	0.04357787
28	0.00000000	5.19615240	0.00000000
29	-2.49048670	-0.86602540	0.21788935
30	-1.49429200	-0.86602540	0.13073361
31	-0.49809735	-0.86602540	0.04357787
32	0.49809735	-0.86602540	0.04357787
33	1.49429200	-0.86602540	0.13073361
34	2.49048670	-0.86602540	0.21788935
35	-1.99238940	-1.73205080	0.17431148
36	-0.99619469	-1.73205080	0.08715574
37	0.00000000	-1.73205080	0.00000000
38	0.99619469	-1.73205080	0.08715574
39	1.99238940	-1.73205080	0.17431148
40	-1.49429200	-2.59807620	0.13073361
41	-0.49809735	-2.59807620	0.04357787
42	0.49809735	-2.59807620	0.04357787
43	1.49429200	-2.59807620	0.13073361
44	-0.99619469	-3.46410160	0.08715574
45	0.00000000	-3.46410160	0.00000000
46	0.99619469	-3.46410160	0.08715574
47	-0.49809735	-4.33012700	0.04357787
48	0.49809735	-4.33012700	0.04357787
49	0.00000000	-5.19615240	0.00000000

TABLE I (continued)

50	-3.98477880	0.00000000	0.34862297
51	3.98477880	0.00000000	0.34862297
52	-3.48668140	0.86602540	0.30504509
53	3.48668140	0.86602540	0.30504509
54	-2.98858410	1.73205080	0.26146723
55	2.98858410	1.73205080	0.26146723
56	-2.49048670	2.59807620	0.21788935
57	2.49048670	2.59807620	0.21788935
58	-1.99238940	3.46410160	0.17431148
59	1.99238940	3.46410160	0.17431148
60	-1.49429200	4.33012700	0.13073361
61	1.49429200	4.33012700	0.13073361
62	-0.99619469	5.19615240	0.08715574
63	0.99619469	5.19615240	0.08715574
64	-0.49809735	6.06217780	0.04357787
65	0.49809735	6.06217780	0.04357787
66	0.00000000	6.92820320	0.00000000
67	-3.48668140	-0.86602540	0.30504509
68	3.48668140	-0.86602540	0.30504509
69	-2.98858410	-1.73205080	0.26146723
70	2.98858410	-1.73205080	0.26146723
71	-2.49048670	-2.59807620	0.21788935
72	2.49048670	-2.59807620	0.21788935
73	-1.99238940	-3.46410160	0.17431148
74	1.99238940	-3.46410160	0.17431148
75	-1.49429200	-4.33012700	0.13073361
76	1.49429200	-4.33012700	0.13073361
77	-0.99619469	-5.19615240	0.08715574
78	0.99619469	-5.19615240	0.08715574
79	-0.49809735	-6.06217780	0.04357787
80	0.49809735	-6.06217780	0.04357787
81	0.00000000	-6.92820320	0.00000000

Recursive formulas (3.1) - (3.7) were applied next in the following way to produce the desired saddle shaped-minimal surface. We first fixed the parameters $G = 1$, $H = 0.5$, $p = 2$, $q = 4$, $\Delta t = 0.001$, $D = 1.3$. Initial calculation revealed great volatility due to the very large local forces which resulted from the bending process. The solution to this problem, at first, was to set any component velocity of a particle to zero if that velocity exceeded a preset given upper bound. We were then able to obtain cycling of coordinates and velocities over a time period of approximately ten seconds (10,000 iterations). However, the changes in the positions of the particles and their velocities were still too great. So, a bootstrap procedure was used as follows. At the end of each run of 10,000 iterations the average coordinate position of each interior particle was used to generate a new set of initial positions for the next run. Thereafter, the averages after each run began to converge and the component velocities of the particles appeared to approach zero. In TABLE 2 are listed the results of the fifth computer run. However, the surface was at this time so stable, that the actual final positions of the fourth run, given in TABLE 3, not their averages, were used as initial data for the run. TABLE 4 gives the average position during the fifth run. Notice in TABLES 2-4 that each coordinate in any position agrees remarkably well with the corresponding entry in both of the other tables. Also, the velocities are now

entirely negligible, as is shown in TABLE 5. The total movement of any coordinate over 10,000 time steps did not exceed .00034823 and no component velocity had to be set to zero during the iteration.

Figure 4.4 shows a plot of the final positions of the particles. The shape is exactly what one would expect for a real soap film.

TABLE 2

FINAL POSITIONS FOR THE FIFTH RUN

1	-2.98780870	0.00000003	0.29012458
2	-1.99152460	-0.00000036	0.25475762
3	-0.99565876	0.00000020	0.23524624
4	0.00000111	0.00000021	0.22894563
5	0.99565672	-0.00000156	0.23524625
6	1.99152290	0.00000268	0.25475766
7	2.98780840	-0.00000069	0.29012458
8	-2.48970900	0.86577069	0.26366488
9	-1.49358690	0.86566716	0.23804362
10	-0.49782404	0.86562841	0.22575726
11	0.49782451	0.86562729	0.22575726
12	1.49358080	0.86566581	0.23804373
13	2.48971400	0.86577071	0.26366473
14	-1.99176540	1.73165010	0.23264228
15	-0.99574584	1.73145720	0.21589245
16	-0.00000088	1.73140120	0.21044558
17	0.99574867	1.73145790	0.21589246
18	1.99176600	1.73165080	0.23264224
19	-1.49388710	2.59763270	0.19889109
20	-0.49791986	2.59741340	0.18952451
21	0.49791949	2.59741270	0.18952445
22	1.49389750	2.59763540	0.19889119
23	-0.99598234	3.46368050	0.16294201
24	-0.00000438	3.46360110	0.15920019
25	0.99598891	3.46368070	0.16294198
26	-0.49805120	4.32984690	0.12503719
27	0.49805055	4.32985320	0.12503696
28	0.00000202	5.19609970	0.08526757
29	-2.48970890	-0.86577094	0.26366486
30	-1.49358500	-0.86566764	0.23804363
31	-0.49782513	-0.86562837	0.22575730
32	0.49782469	-0.86562888	0.22575722
33	1.49358500	-0.86566691	0.23804355
34	2.48970870	-0.86577060	0.26366492
35	-1.99176550	-1.73164990	0.23264228
36	-0.99574663	-1.73145760	0.21589249
37	-0.00000053	-1.73140400	0.21044557
38	0.99574636	-1.73145790	0.21589263
39	1.99176610	-1.73165030	0.23264223
40	-1.49388990	-2.59763290	0.19889111
41	-0.49791985	-2.59741650	0.18952453
42	0.49791989	-2.59741550	0.18952448
43	1.49389210	-2.59763430	0.19889111
44	-0.99598485	-3.46368260	0.16294201
45	0.00000018	-3.46360480	0.15920015
46	0.99598561	-3.46368240	0.16294195
47	-0.49805047	-4.32985490	0.12503688
48	0.49805089	-4.32985530	0.12503684
49	0.00000002	-5.19609950	0.08526757

TABLE 3

INITIAL POSITIONS FOR THE FIFTH RUN

1	-2.98785400	0.00000000	0.29021650
2	-1.99156240	0.00000000	0.25483306
3	-0.99568637	0.00000001	0.23541792
4	-0.00000001	0.00000001	0.22912845
5	0.99568637	0.00000002	0.23541792
6	1.99156240	0.00000000	0.25483316
7	2.98785400	0.00000001	0.29021650
8	-2.48975910	0.86579983	0.26376109
9	-1.49361650	0.86568716	0.23809350
10	-0.49783854	0.86564452	0.22599674
11	0.49783855	0.86564451	0.22599674
12	1.49361690	0.86568716	0.23809350
13	2.48975890	0.86579983	0.26376109
14	-1.99182310	1.73167030	0.23274407
15	-0.99576951	1.73147450	0.21601891
16	-0.00000001	1.73141990	0.21052842
17	0.99576934	1.73147450	0.21601891
18	1.99182310	1.73167030	0.23274405
19	-1.49391650	2.59765610	0.19897847
20	-0.49791826	2.59745660	0.18961494
21	0.49791827	2.59745660	0.18961494
22	1.49391650	2.59765610	0.19897847
23	-0.99600807	3.46373330	0.16297813
24	0.00000000	3.46355670	0.15920862
25	0.99600816	3.46373330	0.16297813
26	-0.49803277	4.32983400	0.12506531
27	0.49803277	4.32983400	0.12506531
28	0.00000000	5.19613480	0.08528388
29	-2.48975910	-0.86579983	0.26376109
30	-1.49361650	-0.86568716	0.23809350
31	-0.49783854	-0.86564452	0.22599674
32	0.49783856	-0.86564451	0.22599674
33	1.49361650	-0.86568716	0.23809350
34	2.48975910	-0.86579983	0.26376109
35	-1.99182310	-1.73167030	0.23274407
36	-0.99576958	-1.73147450	0.21601891
37	0.00000000	-1.73141990	0.21052842
38	0.99576938	-1.73147450	0.21601891
39	1.99182320	-1.73167030	0.23274407
40	-1.49391650	-2.59765610	0.19897847
41	-0.49791827	-2.59745660	0.18961494
42	0.49791829	-2.59745660	0.18961494
43	1.49391650	-2.59765610	0.19897847
44	-0.99600807	-3.46373330	0.16297813
45	0.00000001	-3.46355670	0.15920862
46	0.99600811	-3.46373330	0.16297813
47	-0.49803277	-4.32983400	0.12506531
48	0.49803277	-4.32983400	0.12506531
49	0.00000001	-5.19613480	0.08528390

TABLE 4

AVERAGE POSITIONS FOR THE FIFTH RUN

1	-2.98780580	0.00000000	0.29008270
2	-1.99148800	-0.00000003	0.25464207
3	-0.99563269	-0.00000003	0.23495869
4	0.00000006	-0.00000001	0.22860078
5	0.99563264	-0.00000011	0.23495870
6	1.99148800	0.00000000	0.25464207
7	2.98780580	-0.00000007	0.29008270
8	-2.48974470	0.86578315	0.26359765
9	-1.49356330	0.86567212	0.23781000
10	-0.49781411	0.86561688	0.22543953
11	0.49781415	0.86561688	0.22543953
12	1.49356330	0.86567212	0.23781000
13	2.48974470	0.86578315	0.26359766
14	-1.99176720	1.73167560	0.23253905
15	-0.99574075	1.73145300	0.21567352
16	0.00000008	1.73142880	0.21014447
17	0.99574076	1.73145300	0.21567350
18	1.99176730	1.73167560	0.23253907
19	-1.49390670	2.59765620	0.19877377
20	-0.49791129	2.59743260	0.18931825
21	0.49791132	2.59743260	0.18931831
22	1.49390670	2.59765620	0.19877377
23	-0.99598652	3.46368350	0.16288059
24	-0.00000006	3.46360240	0.15908265
25	0.99598662	3.46368350	0.16288059
26	-0.49802930	4.32989810	0.12502981
27	0.49802842	4.32989810	0.12502969
28	-0.00000012	5.19615600	0.08525457
29	-2.48974470	-0.86578315	0.26359765
30	-1.49356330	-0.86567212	0.23780999
31	-0.49781410	-0.86561666	0.22543953
32	0.49781414	-0.86561688	0.22543953
33	1.49356330	-0.86567212	0.23781000
34	2.48974470	-0.86578315	0.26359767
35	-1.99176720	-1.73167560	0.23253906
36	-0.99574075	-1.73145300	0.21567352
37	-0.00000003	-1.73142880	0.21014447
38	0.99574073	-1.73145300	0.21567350
39	1.99176680	-1.73167560	0.23253907
40	-1.49390670	-2.59765620	0.19877377
41	-0.49791130	-2.59743260	0.18931825
42	0.49791129	-2.59743260	0.18931825
43	1.49390670	-2.59765620	0.19877377
44	-0.99598652	-3.46368350	0.16288059
45	0.00000001	-3.46360240	0.15908265
46	0.99598648	-3.46368350	0.16288059
47	-0.49802905	-4.32989810	0.12502973
48	0.49802903	-4.32989810	0.12502974
49	0.00000002	-5.19615600	0.08525457

TABLE 5

FINAL VELOCITIES FOR THE FIFTH RUN

1	0.00000537	-0.00000007	0.00009513
2	-0.00002296	-0.00000001	0.00006609
3	-0.00003175	0.00000021	0.00010247
4	-0.00000073	-0.00000077	0.00054269
5	0.00003249	0.00000007	0.00010249
6	0.00002108	0.00000024	0.00006606
7	-0.00000299	0.00000096	0.00009514
8	-0.00001222	-0.00000163	0.00004678
9	-0.00000719	-0.00000872	0.00008887
10	-0.00000933	0.00000517	0.00014376
11	0.00001065	0.00000771	0.00014383
12	0.00000468	-0.00000749	0.00008873
13	0.00001322	-0.00000234	0.00004674
14	-0.00001189	0.00000288	0.00006839
15	-0.00000933	-0.00000350	0.00013421
16	-0.00000050	0.00000871	0.00022406
17	0.00000853	-0.00000385	0.00013431
18	0.00001186	0.00000112	0.00006829
19	0.00000584	-0.00000084	0.00009156
20	-0.00000874	-0.00000403	0.00006273
21	0.00000773	0.00000628	0.00006265
22	-0.00000654	-0.00001189	0.00009181
23	-0.00000832	0.00003259	0.00007528
24	-0.00000085	0.00000535	0.00004247
25	0.00001015	0.00003638	0.00007512
26	-0.00000945	-0.00001178	0.00001413
27	0.00001201	-0.00001523	0.00001441
28	0.00000110	-0.00001403	0.00003781
29	-0.00001256	0.00000150	0.00004678
30	-0.00000659	0.00000932	0.00008889
31	-0.00001066	-0.00000542	0.00014377
32	0.00001006	-0.00000835	0.00014374
33	0.00001190	0.00000925	0.00008903
34	0.00001099	0.00000158	0.00004670
35	-0.00001284	-0.00000381	0.00006838
36	-0.00001030	0.00000229	0.00013421
37	0.00000079	-0.00000760	0.00022413
38	0.00000977	0.00000470	0.00013407
39	0.00001293	-0.00000560	0.00006837
40	0.00000614	-0.00000130	0.00009161
41	-0.00000825	0.00000786	0.00006275
42	0.00000630	0.00000300	0.00006266
43	-0.00000635	0.00000556	0.00009179
44	-0.00000860	-0.00003554	0.00007507
45	-0.00000015	-0.00000808	0.00004248
46	0.00000936	-0.00003535	0.00007513
47	-0.00000974	0.00002100	0.00001438
48	0.00000998	0.00002400	0.00001431
49	0.00000006	0.00000977	0.00003769

Figure 4.4

References

1. F. J. Almgren, Jr., and J. E. Taylor, "The geometry of soap films and soap bubbles", *Scientific American*, 235 (1), 1976, pp. 82-93.
2. P. Concus, "Numerical solution of the minimal surface equation", *Math. Comp.*, 21, 1967, pp. 340-350.
3. C. Coppin and D. Greenspan, "Discrete modeling of minimal surfaces", TR 186, Dept. Math., Univ. Texas at Arlington, 1982.
4. R. Courant, Dirichlet's Principle, Conformal Mapping, and Minimal Surfaces, Interscience, N. Y., 1950.
5. R. P. Feynman, R. B. Leighton, and M. Sands, The Feynman Lectures on Physics, Addison-Wesley, Reading, Mass., 1963.
6. D. Greenspan, "On approximating extremals of functionals", Part II, *Int. Jour. Eng. Sci.*, 5, 1967, pp. 571-588.
7. D. Greenspan, Arithmetic Applied Mathematics, Pergamon Press, Oxford, 1980.
8. M. Hinata, M. Shimasaki, and T. Kiyono, "Numerical solution of Plateau's problem by a finite element method", *Math. Comp.*, 28, 1974, pp. 45-60.
9. T. Rado', On the Plateau Problem, Chelsea, N. Y., 1951.

APPENDIX

```

DIMENSION M(100),P(100,3),P1(100,3),VO(100,3),V1(100,3)
DIMENSION A(100,3),AVP(100,3)
DIMENSION R(100,100)
INTEGER C,F,C2
OPEN(UNIT=20,DEVICE='DSK',ACCESS='SEQIN',FILE='DATIN.DAT')
OPEN(UNIT=21,DEVICE='DSK',ACCESS='SEQOUT',FILE='DATOU.DAT')
READ(20,1196) L
DO 50 I=1,100
DO 50 J=1,3
READ(20,75) I,J,P(I,J),AVP(I,J),VO(I,J)
50 CONTINUE
75 FORMAT(1X,2I3,3F15.8)
G=1
H=.5
B=(3.14159265/180)*5
N=49
SQRN=N**2
D=.001
S=2
T=4
L=0
DO 450 I=1,(SQRN+2)**2
DO 450 J=1,3
P(I,J)=AVP(I,J)
AVP(I,J)=0
VO(I,J)=0
450 CONTINUE
WRITE(5,606)
606 FORMAT(1X,'INITIAL POSITIONS')
DO 630 I=1,(SQRN+2)**2
WRITE(5,616) I,P(I,1),P(I,2),P(I,3)
616 FORMAT(I4,3F15.8)
M(I)=1
630 CONTINUE
CALL NAME(R,P,VI,VO,A,M,G,H,S,T,N,SQRN,D)
DO 740 I=1,N
DO 740 J=1,3
V1(I,J)=VO(I,J)+(D/2)*A(I,J)
740 CONTINUE
GOTO 780
770 CALL NAME(R,P,V1,VO,A,M,G,H,S,T,N,SQRN,D)
780 DO 830 I=1,N
DO 830 J=1,3
P1(I,J)=P(I,J)+D*V1(I,J)
P(I,J)=P1(I,J)
TOTDIS+TOTDIS+ABS(D*V1(I,J))
VO(I,J)=V1(I,J)
830 CONTINUE
L=L+1
DO 890 I=1,(SQRN+2)**2
DO 890 J=1,3

```

```

      AVP(I,J)=(AVP(I,J)*(L-1)+P(I,J))/L
890  CONTINUE
      DO 960 I=1,N
      DO 960 J=1,3
      IF(ABS(VO(I,J)).LT..03) GOTO 960
      VELHIG=VELHIG+1
      VO(I,J)=0
960  CONTINUE
      IF(L.EQ.10000) GOTO 1080
      GOTO 770
1080 WRITE(5,1081)
1081 FORMAT(1X,'AVERAGE POSITIONS')
      DO 1110 I=1,N
      WRITE(5,1101) I,AVP(I,1),AVP(I,2),AVP(I,3)
1101 FORMAT(I4,3F15.8)
1110 CONTINUE
      WRITE(5,1121)
1121 FORMAT(1X,'FINAL POSITIONS')
      DO 1150 I=1,N
      WRITE(5,1141) I,P(I,1),P(I,2),P(I,3)
1141 FORMAT(I4,3F15.8)
1150 CONTINUE
      WRITE(5,1161)
1161 FORMAT(1X,'VELOCITIES')
      DO 1190 I=1,N
      WRITE(5,1181) I,VO(I,1),VO(I,2),VO(I,3)
1181 FORMAT(I4,3F15.8)
1190 CONTINUE
      AVDIS=TOTDIS/(3*N)
      WRITE(5,1194) AVDIS
1194 FORMAT(1X,'AVERAGE COORDINATE MOVEMENT IS',F15.8)
      WRITE(5,1196) L
1196 FORMAT(1X,'TIME STEP L=',I8)
      WRITE(5,1198) VELHIG
1198 FORMAT(1X,'VELOCITY HIGH,NUMBER=',I8)
      WRITE(21,1196) L
      DO 1240 I=1,100
      DO 1240 J=1,3
      WRITE(21,75) I,J,P(I,J),AVP(I,J),VC(I,J)
1240 CONTINUE
      CLOSE(UNIT=20,DEVICE='DSK',ACCESS='SEQIN',FILE='DATIN.DAT')
      CLOSE(UNIT=21,DEVICE='DSK',ACCESS='SEQOUT',FILE='DATOU.DAT')
      STOP
      END
      SUBROUTINE NAME(R,P,V1,VO,A,M,G,H,S,T,N,SQRN,D)
      DIMENSION R(100,100),P(100,3),V1(100,3),VO(100,3)
      DIMENSION A(100,3),M(100)
      DO 1360 I=1,N
      DO 1360 J=I+1,(SQRN+2)**2

      XCORD=P(I,1)-P(J,1)
      YCORD=P(I,2)-P(J,2)
      ZCORD=P(I,3)-P(J,3)
      R(I,J)=SQRT(XCORD**2+YCORD**2+ZCORD**2)

```

```
      R(J,I)=R(I,J)
1360  CONTINUE
      DO 1510 I=1,N
      DO 1510 J=1,3
      A(I,J)=0
      DO 1490 K=1,(SQRN+2)**2
      IF(K.EQ.I) GOTO 1490
      IF(R(I,K).GT.1.3) GOTO 1490
      FACT=-G/(R(I,K)**S)+H/(R(I,K)**T)
      A(I,J)=A(I,J)+M(K)*FACT*(P(I,J)-P(K,J))/R(I,K)
1490  CONTINUE
      V1(I,J)=VO(I,J)+D*A(I,J)
1510  CONTINUE
      RETURN
      END
```