DATE DUE

SUPERCOMPUTER SIMULATION OF CRACKS AND FRACTURES
BY QUASIMOLECULAR DYNAMICS

DONALD GREENSPAN

TR# 260

UNIVERSITY OF TEXAS
AT ARLINGTON
LIBRARY
JUL -9 1990
TEXAS
DOCUMENT

Supercomputer Simulation of Cracks and Fractures by Quasimolecular Dynamics*

By

Donald Greenspan

TR# 260

Abstract

The gross physical behavior of solids and liquids is the result of atomic or molecular reactions to external forces. Using molecular dynamics, we can study such reactions in the small, on the molecular level. In quasimolecular dynamics, molecules are aggregated into larger units, called quasimolecules, in order to simulate behavior in the large. Mass and energy conservation considerations are fundamental in this approach. In particular, we will simulate the generation of cracks in a slotted plate which, for illustrative purposes, is chosen to be of pure copper. CRAY X-MP/24 supercomputer examples are described and discussed.

*Computations performed at the University of Texas Center for High Performance Computing.

1. Introduction. The method of molecular dynamics applies to the response of N atoms or molecules of a solid or fluid to external forces [1]–[4], [6], [10], [11]. The primary objective of many molecular dynamics studies is the deduction of fundamental thermodynamic properties. However, the question as to whether or not simulations in the small ($N < 10^4$) are valid representations of dynamical behavior in the large ($N > 10^{15}$) is not as yet answered comprehensively [5].

Our aim in this paper is to apply a generalization of molecular dynamics, called quasimolecular dynamics [7], [9], for the simulation in the large of crack generation in a stressed copper plate [2], [3], [12], [14]. The basic idea is to group molecules into aggregates, called quasimolecules, and apply dynamically a Lennard–Jones type potential which is readily computable and which conserves both the mass and the total energy of the original molecular system, but normalized in a precise fashion prescribed by the method of molecular aggregation.

2. Mathematical, Physical, and Modelling Preliminaries. In any solid, the forces of molecular interaction have components of both attraction and repulsion. Classically, these forces have magnitude F given by

$$(1) \quad F = -\frac{G}{r^p} + \frac{H}{r^q}, \quad G > 0, H > 0, q > p \geq 7,$$

where r is the radius from molecule P to a neighboring molecule.

To simulate the reaction of a solid to external forces, we will proceed as follows. First, we group the large number of molecules which are physically present into a relatively small number of larger units called quasimolecules, or particles. Next, we define the motion of each particle, denoted by P_i , of the resulting system of, say, N particles, by the coupled system of nonlinear, second order, ordinary differential equations

$$(2) \quad \vec{F}_i = m_i \vec{r}_i, \quad i = 1, 2, \dots, N,$$

in which m_i is the mass of P_i . Finally, the resulting differential system (2) is solved numerically from given initial data as follows [7]. Consider N particles P_i , $i = 1, 2, \dots, N$. For $\Delta t > 0$, let $t_k = k\Delta t$, $k = 0, 1, 2, \dots$. For each i , let m_i denote the mass of P_i and let P_i at t_k be located at $\vec{r}_{i,k} = (x_{i,k}, y_{i,k})$, have velocity $\vec{v}_{i,k} = (v_{i,k,x}, v_{i,k,y})$ and have acceleration $\vec{a}_{i,k} = (a_{i,k,x}, a_{i,k,y})$. Let position, velocity, and acceleration be related by the formulas

$$(3) \quad \vec{v}_{i,1/2} = \vec{v}_{i,0} + \frac{1}{2}(\Delta t)\vec{a}_{i,0}$$

$$(4) \quad \vec{v}_{i,k+1/2} = \vec{v}_{i,k-1/2} + (\Delta t)\vec{a}_{i,k}, \quad k = 1, 2, \dots$$

$$(5) \quad \vec{r}_{i,k+1} = \vec{r}_{i,k} + (\Delta t)\vec{v}_{i,k+1/2}, \quad k = 0, 1, 2, \dots$$

At t_k , let the force acting on P_i be $\vec{F}_{i,k} = (F_{i,k,x}, F_{i,k,y})$. We relate force and acceleration by the dynamical difference equation

$$(6) \quad \vec{F}_{i,k} = m_i \vec{a}_{i,k}.$$

The motion of each P_i will be determined explicitly and recursively by equations (3)–(6) from given initial data once the force $\vec{F}_{i,k}$ is prescribed, and this is done as follows. If P_j is a neighbor of P_i , 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 due to P_j at time t_k is defined by

$$(7) \quad \vec{F}_{ij,k} = \left[-\frac{G}{(r_{ij,k})^p} + \frac{H}{(r_{ij,k})^q} \right] \frac{\vec{r}_{ji,k}}{r_{ij,k}}.$$

The force $\vec{F}_{i,k}$ on P_i at t_k is defined by

$$(8) \quad \vec{F}_{i,k} = \sum_{\substack{j=1 \\ j \neq i}} \vec{F}_{ij,k},$$

where the summation is taken over all neighbors of P_i .

For the convenience of the reader, a typical CRAY FORTRAN program is available in Greenspan [8].

3. Parameter Selection for a Rectangular Copper Plate. Let us now consider the development of cracks in a slotted copper plate. For this simulation, the choices of N , p , and q will be dominated by computational considerations, while those of m , G , H will be dominated by physical considerations.

For computational convenience and simplicity, set $p = 3$, $q = 5$. A relatively large time step can then be employed in implementing (3)–(5). In addition, the computer program will require no square root routine. For budgetary reasons, we are limited to $N \sim 1250$. By the introduction of physical symmetry, we can increase this to $N \sim 2500$ and hence proceed as follows. Let the points P_i , with respective coordinates (x_i, y_i) , $i = 1, 2, \dots, 2713$ be defined by

$$x(1) = -3.9, y(1) = -5.71576764$$

$$x(41) = -4.0, y(41) = -5.54256256$$

$$x(i+1) = x(i) + 0.2, y(i+1) = y(1), i = 1, 2, \dots, 39$$

$$x(i+1) = x(i) + 0.2, y(i+1) = y(41), i = 41, 42, \dots, 80$$

$$x(i) = x(i-81), y(i) = y(i-81) + 2(0.17320508), i = 82, 83, \dots, 2713.$$

The resulting arrangement is shown in Figure 1. The (x_i, y_i) are vertices of a regular triangular mosaic in which the distance from any P_i to an immediate neighbor is 0.2 cm. The P_i are assumed to represent quasimolecules of an 8 cm by 11.43 cm rectangular copper plate.

As shown in Figure 1, the particle numbering begins on the bottom row, increases left to right, and the numbering then proceeds to the next higher row. For precision, we state now that the neighbors of any particle P_i are those particles in Figure 1 which are 0.2 units from P_i . Thus, for

$$(8) \quad \vec{F}_{i,k} = \sum_{\substack{j=1 \\ j \neq i}} \vec{F}_{ij,k},$$

where the summation is taken over all neighbors of P_i .

For the convenience of the reader, a typical CRAY FORTRAN program is available in Greenspan [8].

3. Parameter Selection for a Rectangular Copper Plate. Let us now consider the development of cracks in a slotted copper plate. For this simulation, the choices of N , p , and q will be dominated by computational considerations, while those of m , G , H will be dominated by physical considerations.

For computational convenience and simplicity, set $p = 3$, $q = 5$. A relatively large time step can then be employed in implementing (3)–(5). In addition, the computer program will require no square root routine. For budgetary reasons, we are limited to $N \sim 1250$. By the introduction of physical symmetry, we can increase this to $N \sim 2500$ and hence proceed as follows. Let the points P_i , with respective coordinates (x_i, y_i) , $i = 1, 2, \dots, 2713$ be defined by

$$x(1) = -3.9, y(1) = -5.71576764$$

$$x(41) = -4.0, y(41) = -5.54256256$$

$$x(i+1) = x(i) + 0.2, y(i+1) = y(1), i = 1, 2, \dots, 39$$

$$x(i+1) = x(i) + 0.2, y(i+1) = y(41), i = 41, 42, \dots, 80$$

$$x(i) = x(i-81), y(i) = y(i-81) + 2(0.17320508), i = 82, 83, \dots, 2713.$$

The resulting arrangement is shown in Figure 1. The (x_i, y_i) are vertices of a regular triangular mosaic in which the distance from any P_i to an immediate neighbor is 0.2 cm. The P_i are assumed to represent quasimolecules of an 8 cm by 11.43 cm rectangular copper plate.

As shown in Figure 1, the particle numbering begins on the bottom row, increases left to right, and the numbering then proceeds to the next higher row. For precision, we state now that the neighbors of any particle P_i are those particles in Figure 1 which are 0.2 units from P_i . Thus, for

example, each interior particle has exactly six neighbors, while the number of neighbors of a particle on the outer boundary or on the slot may be 3,4, or 5, depending on the particle's particular boundary or slot position. The neighbors of P_i , so defined, will remain the neighbors of P_i throughout the simulation.

To determine m , G and H , the following results will be of value. A convenient least square fit of the Morse potential for copper [3] is

$$(9) \quad \phi(r) = - \frac{(1.398068)10^{-10}}{r^6} + \frac{(1.55104)10^{-8}}{r^{12}} \text{ erg},$$

in which r is measured in angstroms. From (9), it follows that the magnitude F of the force $\vec{F}$, in dynes, between two atoms r angstroms apart is

$$(10) \quad F(r) = - \frac{8.388408}{r^7} 10^{-10} + \frac{1.866125}{r^{13}} 10^{-7}.$$

The maximum of ϕ results when $F(r) = 0$, which implies $r = 2.46 \text{ \AA}$ and $\phi(2.46) = - (3.15045)10^{-13} \text{ erg}$.

Now, suppose that the rectangular plate were to be filled with copper atoms by using, again, a regular triangular mosaic, but in which the distance between two immediate neighbors is $2.46 \text{ \AA}$. Then the number of atoms N^* in the plate is, approximately,

$$(11) \quad N^* = \frac{8(10^8)}{2.46} \cdot \frac{(11.43)10^8}{2.13} = (1.745)10^{17}.$$

Since the mass of a copper atom is $(1.0542)10^{-22} \text{ gr}$, the total mass M of the atoms is $M = (1.840)10^{-5} \text{ gr}$. Distributing this mass over the 2713 quasimolecules yields a quasimolecular mass m given by

Figure 1. The initial configuration.

Fig. 2 A quasimolecule and its six nearest neighbors.

$$(12) \quad m = (6.782)10^{-9} \text{ gr.}$$

To approximate G and H, consider first the total energy E^* of the configuration of copper atoms described above. Recall that the energy determined by one molecule and an immediate neighbor is

$$(13) \quad \phi(2.46) = -(3.15045)10^{-13} \text{ erg.}$$

Since the maximum potential results from two such adjacent atoms, we can estimate E^* as follows. Suppose one traverses the rows of atoms from left to right, starting at the top, and determines the potential energies from immediate neighbors only. When one comes to an atom P_0 , as shown in Figure 2, P_0 will have exactly six neighbors, which are shown as $P_1, P_2, P_3, P_4, P_5, P_6$ in Figure 2. However, the potential energies of the pairs P_0P_2, P_0P_3, P_0P_4 will already have been counted, due to the method of traversing the rows. Thus, at P_0 , only three contributions are made to the energy calculation, and these are from the pairs P_0P_1, P_0P_5, P_0P_6 . And, since the interior points far outnumber the boundary points, the total energy E^* is, approximately,

$$(14) \quad E^* = 3(1.745)10^{17} (-3.15045)10^{-13} = -(1.6493)10^5 \text{ erg.}$$

Now, let us assume that the force $\vec{F}$, in dynes, between two quasimolecules has magnitude F given by

$$(15) \quad F(R) = -\frac{G}{R^3} + \frac{H}{R^5},$$

where R is measured in cm. Hence,

$$\phi(R) = -\frac{G}{2R^2} + \frac{H}{4R^4} \text{ erg,}$$

and we assume first that ϕ is maximal for $R = 0.2$, so that $F(0.2) = 0$, or,

$$(16) \quad -\frac{G}{(.2)^3} + \frac{H}{(.2)^5} = 0.$$

Approximating the total energy E for the quasimolecular system is the fashion used to obtain E^* yields

$$(17) \quad E = 3(2713) \left(-\frac{G}{2(.2)^2} + \frac{H}{4(.2)^4} \right) \text{ erg.}$$

Conserving total energy by equating E and E^* yields

$$3(2713) \left(-\frac{G}{2(.2)^2} + \frac{H}{4(.2)^4} \right) = -(1.6493)10^5,$$

or, equivalently,

$$(18) \quad -\frac{G}{2(.2)^2} + \frac{H}{4(.2)^4} = -20.264.$$

The solution of (16) and (18) is $G = 3.24224$, $H = 0.12969$, so that (15) takes the form

$$(19) \quad F(R) = -\frac{3.24224}{R^3} + \frac{0.12969}{R^5}.$$

4. Dynamical Equations. Dynamically, the force between two quasimolecules must be local in the presence of gravity. Thus, we must introduce a normalizing factor α such that at the distance $2(.2) = 0.4$ cm, the force between two quasimolecules is small relative to gravity. This is essential because we have assumed that forces act only between neighbors, and these are initially 0.2 cm apart. If we then define "small relative to gravity" to mean 0.1% of the effect of gravity, then we must have

$$\alpha \left| -\frac{3.24224}{(0.4)^3} + \frac{0.12969}{(0.4)^5} \right| < (0.1\%) (980 \text{ m}),$$

which implies $\alpha < (1.749)10^{-10}$. Our choice is $\alpha = (1.25)10^{-10}$. Thus, the dynamical equation for the motion of each P_i is taken to be

$$(20) \quad m \frac{d^2 \vec{R}_i}{dt^2} = (1.25)10^{-10} \sum \left[\left(-\frac{3.24224}{(R_{ij})^3} + \frac{0.12969}{(R_{ij})^5} \right) \frac{\vec{R}_{ji}}{R_{ij}} \right]$$

in which $\vec{R}_{ji}$ is the vector from P_j to P_i and the summation is taken over the neighbors of P_i . From (11), then,

$$(21) \quad \frac{d^2 \vec{R}_i}{dt^2} = \sum \left[\left(-\frac{0.0597582}{(R_{ij})^3} + \frac{0.00239033}{(R_{ij})^5} \right) \frac{\vec{R}_{ji}}{R_{ij}} \right].$$

For computational simplicity, then, we make the transformations

$$(22) \quad R^* = 4R, \quad T^2 = 10t^2$$

which yield finally

$$(23) \quad \frac{d^2 \vec{R}_i^*}{dT^2} = \sum \left[\left(-\frac{1.52981}{(R_{ij}^*)^3} + \frac{0.97908}{(R_{ij}^*)^5} \right) \frac{\vec{R}_{ji}^*}{R_{ij}^*} \right].$$

The distance D , at which a quasimolecular bond breaks, is chosen, as in Ashurst and Hoover [2], to be that distance R^* at which $\frac{dF}{dR^*}$ first becomes negative, so that F becomes a decreasing function of R^* . From (23), it follows that $D = 1.033$.

5. Examples. Let us begin by considering a slotted plate. To do this, we simply delete the fifteen particles with subscripts $i = 1070 + 41k$, $k = 0, 1, 2, \dots, 14$. The resulting configuration is shown in Figure 3. To each particle P_i in Figure 3, we then assign an initial velocity of zero, so that all initial data are available. For the numerical solution of system (22) by (3)–(5), we set $\Delta T = 0.0001$. Tensile

forces are now applied in the following way. At each time step, the particles P_1-P_{40} are relocated so that their X coordinates remain the same, their Y coordinates are decreased by 0.00002, and their velocities are continued to be zero. The particles $P_{2674}-P_{2713}$ are treated similarly, with the exception that their Y coordinates are increased by 0.00002. In this fashion the top and bottom rows are stretched. Each remaining quasimolecule reacts to this stretching process by interacting with its neighbors in accordance with (22). If and when a particle and one of its neighbors move a distance apart greater than D , then the force between these particles is set to zero and their bond is considered to be broken, thereby inducing a crack between the two.

Figures 4-9 show the developing force field throughout the bottom half of the plate at the times $T = 2.0, 4.0, 6.0, 8.0, 10.0$, and 12.0 , after 20000, 40000, 60000, 80000, 100000, and 120000 steps, respectively. The total computing time was 25 minutes. By the symmetry of the physics, it suffices, of course, to show the reaction only in the lower half of the plate. In these figures, the force field is represented by vectors emanating from the centers of the quasimolecules. Figures 4-7 reveal the stress effect being transmitted to the interior of the plate. Though this transmission occurs in a relatively short time, the fact that it is not instantaneous is significant, as will be indicated later. Figures 8 and 9 reveal large effects on both the right and left sides of the slot, which yield a widening of the slot. (This effect will, perhaps, be more readily apparent in figures to be described later.) Figure 10, at $T = 12.8$ shows clearly from the force field that the first crack has occurred at the lower left of the slot, and hence, by symmetry, at the upper right.

In the remaining figures, for clarity we will not show the force field. The gross motion of the entire plate is then shown at $T = 13.8, 15.8, 18.8$ in Figures 11-13, respectively. The widening of the slot, a bending and distortion of various rows, and the patterns of fracture are all apparent in these figures. Figures 14(a)-(d), again for the lower half of the plate only, show the crack development at the times $T = 13.8, 14.8, 15.8$, and 16.8 . The multiplicity of the left central, upper cracks in Figure 14(d) is associated with the row dislocations shown in Figure 15.

Figure 3. The slotted plate.

Figure 4. 2×2.0 .

Figure 5. $t = 4.0$.

Figure 6. $t = 6.0$.

Figure 7. $t = 8.0$.

Figure 8. $T = 10.0$.

Figure 9. $T = 12.0$.

Figure 10. $T = 12.8$.

Figure 11. $T=13.8$

Figure 12. $T=15.8$

Figure 13. $T=18.8$

(a) $T=13.8$

(b) $T=14.8$

(c) $T=15.8$

(d) $T=16.8$

Figure 14. Crack development.

Figure 15. Row dislocation at $T=16.8$.

Figure 16. $T = 20.0$.

Figure 17. $T = 30.0$.

A large variety of other examples were run and some of the results are now described briefly. Variation of the above example by moving the bottom and top rows 0.0001 units per time step did not allow sufficient time for the force effect to be transmitted to the interior of the plate, and both the bottom two and top two rows were pulled off by the time $T = 1.5$, thereby simulating an impulsive force effect. When the top and bottom rows were moved only 0.000001 units per time step, no cracks developed through 300000 time steps. To study the effects of shear, a full plate, that is, a plate without a slot, was studied. Moving the bottom row downward 0.00002 units per time step and to the right 0.000005 units per time step, and simultaneously moving the top row symmetrically upward and to the left, resulted in the fractures shown in Figures 16 and 17 at the respective time $T = 20.0$ and $T = 30.0$. Note, however, that in the banded region $-1.0 \leq y \leq 1.0$ of Figure 17, the results shown are not entirely reliable, since the magnitudes of the forces in this neighborhood of the X axis have become so large that roundoff errors have yielded a loss of perfect symmetry. Reliability in this relatively vacuous area could have been improved by decreasing the time step, but funding had been exhausted.

5. Remarks. Our results indicate clearly that copper is not brittle, but is inherently ductile [3]. Moreover, whenever molecular force potentials are available, the methodology developed should be directly applicable in both two and three dimensions to other materials. When such potentials are not available, one can still approximate G and H using stress-strain measurements, as shown by Reeves and Greenspan [13].

Of course, refinement of this first paper is in order. For example, parameter estimates should be sharpened, the definition of a neighboring particle should be extended, and longer time studies with smaller external forces should be carried out.

References

1. B. J. Alder and T. Wainwright, Phys. Rev. A, 1 (1970) 18.
2. W. T. Ashurst and W. G. Hoover, Phys. Rev. B, 14 (1976) 1465.
3. B. de Celis, A. S. Argon and S. Yip, J. App. Phys., 54 (1983) 4864.
4. R. M. J. Cotterill, W. D. Kristensen, J. W. Martin, L. B. Pederson, and E. J. Jensen, Comp. Phys. Comm., 5 (1973) 28.
5. G. J. Dienes and A. Paskin, in Atomistics of Fracture, R. M. Latanision and J. R. Pickens, Eds., Plenum, N. Y. (1983) 671.
6. D. J. Evans and W. G. Hoover, Ann. Rev. Fluid Mech., 18 (1986) 243.
7. D. Greenspan, Arithmetic Applied Mathematics, Pergamon, Oxford (1980).
8. D. Greenspan, "Supercomputer simulation of cracks and fractures by quasimolecular dynamics," TR# 260, Dept. Math., Univ. Texas at Arlington, 1989.
9. D. Greenspan, "Quasimolecular simulation of pendent water drops," TR# 259, Dept. Math., Univ. Texas at Arlington, 1989.
10. T. Halicoiglu and D. M. Cooper, Materials Sci. and Eng., 62 (1984) 121.
11. W. G. Hoover, Physics Today, January (1984) 44.
12. Y. Murakami, in Analytical and Experimental Fracture Mechanics, G. C. Sih and M. Mirable, Eds., Sijthoff and Noordhoff, Alphen aan den Rijn (1981) 871.
13. W. R. Reeves and D. Greenspan, Appl. Math. Modelling, 6 (1982) 185.
14. G. C. Sih (Editor), Dynamics of Crack Propagation, Noordhoff, Leiden (1973).

APPENDIX

FORTRAN PROGRAM: COPPEX.FOR

c This is the 2-d fracture program for a copper plate
 c which is approximately 8 cm by 11.4 cm. The given plate
 c has 2713 quasimolecules. The force formula has magnitude
 c $-1.5260/r^{**3} + 0.97665/r^{**5}$. The particles which form
 c the crack are given 0 mass so that they can be included
 c in the calculations. These particles form a 60 degree
 c angle with the horizontal.

```
      dimension xo(2713),yo(2713),vxo(2713),vxo(2713),
      ix(2713,2),y(2713,2),vx(2713,2),vy(2713,2),
      lacx(2713),acy(2713),pmass(2713)
      OPEN (UNIT=21)
      OPEN (UNIT=31)
      OPEN (UNIT=41)
      OPEN (UNIT=51)
```

c Unit 21 is the input file
 c Unit 31 is the fracture file-gives k,i,r**2
 c Unit 41 is the kprint file-gives k,x,y,acx,acy
 c Unit 51 is the output file

```
10      format (4f16.10)
265     format (i12,4f20.5)
81      format (f10.6,i12,i8,f16.10,i5)
      kprint=20000
      kk=2000
      k=1
      do 4 i=1,2713
      x(i,1)=0.
      y(i,1)=0.
      vx(i,1)=0.
      vy(i,1)=0.
      x(i,2)=0.
      y(i,2)=0.
      vx(i,2)=0.
      vy(i,2)=0.
      acx(i)=0.
      acy(i)=0.
      pmass(i)=1.0
4      continue
```

```

c now fix the initial crack.
c This case is the short slot.
  pmass(1070)=0.0
  do 5 i=1,14
    pmass(1070+41*i)=0.0
5    continue
  read (21,10) (xo(i),yo(i),vxo(i),vyo(i),i=1,2713)
  do 30 i=1,2713
    x(i,1)=xo(i)
    y(i,1)=yo(i)
    vx(i,1)=vxo(i)
    vy(i,1)=vyo(i)
30    continue
  go to 71
65    do 70 i=1,2713
    x(i,1)=x(i,2)
    y(i,1)=y(i,2)
    vx(i,1)=vx(i,2)
    vy(i,1)=vy(i,2)
70    continue
    do 701 i=1,2713
      acx(i)=0.
      acy(i)=0.
701    continue
71    do 77 i=1,1377
      if (i.le.40) go to 7001
      if (i.eq.41) go to 7005
      if (i.lt.81) go to 7003
      if (i.eq.81) go to 7006
      if (i.eq.82) go to 7002
      if (i.lt.121) go to 7003
      if (i.eq.121) go to 7004
      if (mod(i-41,81).eq.0.0) go to 7005
      if (mod(i-82,81).eq.0.0) go to 7002
      if (mod(i-40,81).eq.0.0) go to 7004
      if (mod(i-81,81).eq.0.0) go to 7006
      go to 7003
7001   acx(i)=0.
       acy(i)=0.
       go to 77
7002   r11=(x(i,1)-x(i-41,1))*2+(y(i,1)-y(i-41,1))*2
       r22=(x(i,1)-x(i-40,1))*2+(y(i,1)-y(i-40,1))*2
       r33=(x(i,1)-x(i+1,1))*2+(y(i,1)-y(i+1,1))*2
       r44=(x(i,1)-x(i+40,1))*2+(y(i,1)-y(i+40,1))*2
       r55=(x(i,1)-x(i+41,1))*2+(y(i,1)-y(i+41,1))*2
       f1=pmass(i)*pmass(i-41)*(-1.5260
1+0.97665/(r11))/(r11**2)
       if (r11.lt.1.0667) go to 5000
       f1=0.0
       if (mod(k,kk).st.0) go to 5000
       write (31,81) 1.0667,k,i,r11,1
5000   f2=pmass(i)*pmass(i-40)*(-1.5260
1+0.97665/(r22))/(r22**2)
       if (r22.lt.1.0667) go to 5001
       f2=0.0
       if (mod(k,kk).st.0) go to 5001
       write (31,81) 1.0667,k,i,r22,2

```

```

5001  f3=pmass(i)*pmass(i+1)*(-1.5260
      1+0.97665/(r33))/(r33**2)
      if (r33.lt.1.0667) go to 5002
      f3=0.0
      if (mod(k,kk).st.0) go to 5002
      write (31,81) 1.0667,k,i,r33,3
5002  f4=pmass(i)*pmass(i+40)*(-1.5260
      1+0.97665/(r44))/(r44**2)
      if (r44.lt.1.0667) go to 5003
      f4=0.0
      if (mod(k,kk).st.0) go to 5003
      write (31,81) 1.0667,k,i,r44,4
5003  f5=pmass(i)*pmass(i+41)*(-1.5260
      1+0.97665/(r55))/(r55**2)
      if (r55.lt.1.0667) go to 5004
      f5=0.0
      if (mod(k,kk).st.0) go to 5004
      write (31,81) 1.0667,k,i,r55,5
5004  fx1=f1*(x(i,1)-x(i-41,1))
      fy1=f1*(y(i,1)-y(i-41,1))
      fx2=f2*(x(i,1)-x(i-40,1))
      fy2=f2*(y(i,1)-y(i-40,1))
      fx3=f3*(x(i,1)-x(i+1,1))
      fy3=f3*(y(i,1)-y(i+1,1))
      fx4=f4*(x(i,1)-x(i+40,1))
      fy4=f4*(y(i,1)-y(i+40,1))
      fx5=f5*(x(i,1)-x(i+41,1))
      fy5=f5*(y(i,1)-y(i+41,1))
      acx(i)=fx1+fx2+fx3+fx4+fx5
      acy(i)=fy1+fy2+fy3+fy4+fy5
      go to 77
7003  r11=(x(i,1)-x(i-41,1))**2+(y(i,1)-y(i-41,1))**2
      r22=(x(i,1)-x(i-40,1))**2+(y(i,1)-y(i-40,1))**2
      r33=(x(i,1)-x(i-1,1))**2+(y(i,1)-y(i-1,1))**2
      r44=(x(i,1)-x(i+1,1))**2+(y(i,1)-y(i+1,1))**2
      r55=(x(i,1)-x(i+40,1))**2+(y(i,1)-y(i+40,1))**2
      r66=(x(i,1)-x(i+41,1))**2+(y(i,1)-y(i+41,1))**2
      f1=pmass(i)*pmass(i-41)*(-1.5260
      1+0.97665/(r11))/(r11**2)
      if (r11.lt.1.0667) go to 5010
      f1=0.0
      if (mod(k,kk).st.0) go to 5010
      write (31,81) 1.0667,k,i,r11,1
5010  f2=pmass(i)*pmass(i-40)*(-1.5260
      1+0.97665/(r22))/(r22**2)
      if (r22.lt.1.0667) go to 5011
      f2=0.0
      if (mod(k,kk).st.0) go to 5011
      write (31,81) 1.0667,k,i,r22,2
5011  f3=pmass(i)*pmass(i-1)*(-1.5260
      1+0.97665/(r33))/(r33**2)
      if (r33.lt.1.0667) go to 5012
      f3=0.0
      if (mod(k,kk).st.0) go to 5012
      write (31,81) 1.0667,k,i,r33,3

```

```

5012   f4=pmass(i)*pmass(i+1)*(-1.5260
      1+0.97665/(r44))/(r44**2)
      if (r44.lt.1.0667) go to 5013
      f4=0.0
      if (mod(k,kk).st.0) go to 5013
      write (31,81) 1.0667,k,i,r44,4
5013   f5=pmass(i)*pmass(i+40)*(-1.5260
      1+0.97665/(r55))/(r55**2)
      if (r55.lt.1.0667) go to 5014
      f5=0.0
      if (mod(k,kk).st.0) go to 5014
      write (31,81) 1.0667,k,i,r55,5
5014   f6=pmass(i)*pmass(i+41)*(-1.5260
      1+0.97665/(r66))/(r66**2)
      if (r66.lt.1.0667) go to 5015
      f6=0.0
      if (mod(k,kk).st.0) go to 5015
      write (31,81) 1.0667,k,i,r66,6
5015   fx1=f1*(x(i,1)-x(i-41,1))
      fy1=f1*(y(i,1)-y(i-41,1))
      fx2=f2*(x(i,1)-x(i-40,1))
      fy2=f2*(y(i,1)-y(i-40,1))
      fx3=f3*(x(i,1)-x(i-1,1))
      fy3=f3*(y(i,1)-y(i-1,1))
      fx4=f4*(x(i,1)-x(i+1,1))
      fy4=f4*(y(i,1)-y(i+1,1))
      fx5=f5*(x(i,1)-x(i+40,1))
      fy5=f5*(y(i,1)-y(i+40,1))
      fx6=f6*(x(i,1)-x(i+41,1))
      fy6=f6*(y(i,1)-y(i+41,1))
      acx(i)=fx1+fx2+fx3+fx4+fx5+fx6
      acy(i)=fy1+fy2+fy3+fy4+fy5+fy6
      go to 77
7004   r11=(x(i,1)-x(i-41,1))**2+(y(i,1)-y(i-41,1))**2
      r22=(x(i,1)-x(i-40,1))**2+(y(i,1)-y(i-40,1))**2
      r33=(x(i,1)-x(i-1,1))**2+(y(i,1)-y(i-1,1))**2
      r44=(x(i,1)-x(i+40,1))**2+(y(i,1)-y(i+40,1))**2
      r55=(x(i,1)-x(i+41,1))**2+(y(i,1)-y(i+41,1))**2
      f1=pmass(i)*pmass(i-41)*(-1.5260
      1+0.97665/(r11))/(r11**2)
      if (r11.lt.1.0667) go to 5020
      f1=0.0
      if (mod(k,kk).st.0) go to 5020
      write (31,81) 1.0667,k,i,r11,1
5020   f2=pmass(i)*pmass(i-40)*(-1.5260
      1+0.97665/(r22))/(r22**2)
      if (r22.lt.1.0667) go to 5021
      f2=0.0
      if (mod(k,kk).st.0) go to 5021
      write (31,81) 1.0667,k,i,r22,2
5021   f3=pmass(i)*pmass(i-1)*(-1.5260
      1+0.97665/(r33))/(r33**2)
      if (r33.lt.1.0667) go to 5022
      f3=0.0
      if (mod(k,kk).st.0) go to 5022
      write (31,81) 1.0667,k,i,r33,3

```

```

5022  f4=pmass(i)*pmass(i+40)*(-1.5260
      1+0.97665/(r44))/(r44**2)
      if (r44.lt.1.0667) go to 5023
      f4=0.0
      if (mod(k,kk).st.0) go to 5023
      write (31,81) 1.0667,k,i,r44,4
5023  f5=pmass(i)*pmass(i+41)*(-1.5260
      1+0.97665/(r55))/(r55**2)
      if (r55.lt.1.0667) go to 5024
      f5=0.0
      if (mod(k,kk).st.0) go to 5024
      write (31,81) 1.0667,k,i,r55,5
5024  fx1=f1*(x(i,1)-x(i-41,1))
      fy1=f1*(y(i,1)-y(i-41,1))
      fx2=f2*(x(i,1)-x(i-40,1))
      fy2=f2*(y(i,1)-y(i-40,1))
      fx3=f3*(x(i,1)-x(i-1,1))
      fy3=f3*(y(i,1)-y(i-1,1))
      fx4=f4*(x(i,1)-x(i+40,1))
      fy4=f4*(y(i,1)-y(i+40,1))
      fx5=f5*(x(i,1)-x(i+41,1))
      fy5=f5*(y(i,1)-y(i+41,1))
      acx(i)=fx1+fx2+fx3+fx4+fx5
      acy(i)=fy1+fy2+fy3+fy4+fy5
      go to 77
7005  r11=(x(i,1)-x(i-40,1))**2+(y(i,1)-y(i-40,1))**2
      r22=(x(i,1)-x(i+1,1))**2+(y(i,1)-y(i+1,1))**2
      r33=(x(i,1)-x(i+41,1))**2+(y(i,1)-y(i+41,1))**2
      f1=pmass(i)*pmass(i-40)*(-1.5260
      1+0.97665/(r11))/(r11**2)
      if (r11.lt.1.0667) go to 5030
      f1=0.0
      if (mod(k,kk).st.0) go to 5030
      write (31,81) 1.0667,k,i,r11,1
5030  f2=pmass(i)*pmass(i+1)*(-1.5260
      1+0.97665/(r22))/(r22**2)
      if (r22.lt.1.0667) go to 5031
      f2=0.0
      if (mod(k,kk).st.0) go to 5031
      write (31,81) 1.0667,k,i,r22,2
5031  f3=pmass(i)*pmass(i+41)*(-1.5260
      1+0.97665/(r33))/(r33**2)
      if (r33.lt.1.0667) go to 5032
      f3=0.0
      if (mod(k,kk).st.0) go to 5032
      write (31,81) 1.0667,k,i,r33,3
5032  fx1=f1*(x(i,1)-x(i-40,1))
      fy1=f1*(y(i,1)-y(i-40,1))
      fx2=f2*(x(i,1)-x(i+1,1))
      fy2=f2*(y(i,1)-y(i+1,1))
      fx3=f3*(x(i,1)-x(i+41,1))
      fy3=f3*(y(i,1)-y(i+41,1))
      acx(i)=fx1+fx2+fx3
      acy(i)=fy1+fy2+fy3
      go to 77

```



```

7006   r11=(x(i,1)-x(i-41,1))*2+(y(i,1)-y(i-41,1))*2
      r22=(x(i,1)-x(i-1,1))*2+(y(i,1)-y(i-1,1))*2
      r33=(x(i,1)-x(i+40,1))*2+(y(i,1)-y(i+40,1))*2
      f1=pmass(i)*pmass(i-41)*(-1.5260
1+0.97665/(r11))/(r11**2)
      if (r11.lt.1.0667) go to 5040
      f1=0.0
      if (mod(k,kk).st.0) go to 5040
      write (31,81) 1.0667,k,i,r11,1
5040   f2=pmass(i)*pmass(i-1)*(-1.5260
1+0.97665/(r22))/(r22**2)
      if (r22.lt.1.0667) go to 5041
      f2=0.0
      if (mod(k,kk).st.0) go to 5041
      write (31,81) 1.0667,k,i,r22,2
5041   f3=pmass(i)*pmass(i+40)*(-1.5260
1+0.97665/(r33))/(r33**2)
      if (r33.lt.1.0667) go to 5042
      f3=0.0
      if (mod(k,kk).st.0) go to 5042
      write (31,81) 1.0667,k,i,r33,3
5042   fx1=f1*(x(i,1)-x(i-41,1))
      fy1=f1*(y(i,1)-y(i-41,1))
      fx2=f2*(x(i,1)-x(i-1,1))
      fy2=f2*(y(i,1)-y(i-1,1))
      fx3=f3*(x(i,1)-x(i+40,1))
      fy3=f3*(y(i,1)-y(i+40,1))
      acx(i)=fx1+fx2+fx3
      acy(i)=fy1+fy2+fy3
      go to 77
77      continue
      do 7019 i=1378,2713
      acx(i)=-acx(2714-i)
      acy(i)=-acy(2714-i)
7019      continue
      do 7123 i=1,2713
      if (i.le.40) go to 7985
      if (i.le.1377) go to 7986
      if (i.st.1377) go to 7988
7986      vx(i,2)=vx(i,1)+0.0001*acx(i)
      vy(i,2)=vy(i,1)+0.0001*acy(i)
      x(i,2)=x(i,1)+0.0001*vx(i,2)
      y(i,2)=y(i,1)+0.0001*vy(i,2)
      go to 7123
7985      x(i,2)=x(i,1)
      y(i,2)=y(i,1)-0.00002
      vx(i,2)=0.0
      vy(i,2)=0.0
      go to 7123
7988      x(i,2)=-x(2714-i,2)
      y(i,2)=-y(2714-i,2)
      vx(i,2)=-vx(2714-i,2)
      vy(i,2)=-vy(2714-i,2)
7123      continue

```

```
8043      k=k+1
      if (mod(k,kprint).st.0) go to 82
      do 810 i=1,1377
      write (41,265) k,x(i,2),y(i,2),acx(i),acy(i)
810        continue
82        if (k.lt.20001) go to 65
102        write (51,10) (x(i,2),y(i,2),vx(i,2),vy(i,2),
1i=1,2713)
      stop
      end
```