1 ・ ・ 23, 1

ソ連科学誌・クヴァントから

«Квант» для младших школьников

やさしい物理学 ⑮

小島　英夫（静岡大学）訳

ディスプレイ上の気体

A. N. ビレンキン
G. L. ユトキン

(1989, No. 6, pp. 61～62)

この小論では，論文"ビリヤード球の気体"（本誌，1992年11月号）を補足して，数個のビリヤード球の運動をシミュレートするプログラムの作成法を示した．読者が自分で具体的に適当な言語でプログラムが書けるように説明した．

"気体"プログラム（こう命名する）は，７個の段階に分けられる；(1)変数の定義とデータの設定，(2)入力，(3)画面の準備，(4)最初の衝突の計算，(5)新しい座標の計算，(6)画面上へのデータの出力，(7)モデルの制御．

変数の定義とデータの設定　系を記述するには，次のような変数が必要である：N ― 球の数；a ― 球の半径；T ― 時間のカウンター（最初の時刻を $T=0$ とする）；$T0$ ― 時間の刻み；$A1$ と $A2$ ― 画面の限界．これらの変数の値は，実験ごとに設定しなくてよいように，しかし変更できるように（プログラムを編集する際に）プログラム中に与えておくのがよい．

全体の球の状態は，1個の球に対して4個の変数で記述できる：球の中心の座標 x と y，およびそれに対応する速度 v_x と v_y である．これらすべての変数は，それに対する計算のフォーマットが同じになるように，1個の2次元配列 $R(4, N)$ と書くのがよい．球の古い位置のためには2次元配列 $RS(2, N)$ が必要である．

入力　この段階では，球の中心の座標とそれに対応する速度との初期値を設定する．ここで，過程の"刻み" $T0$ も与えられる．

画面の準備　点の数の多いグラフィック画面は，一定の領域（例えば，$1 \leqq x \leqq 256$，$1 \leqq y \leqq 192$）に含まれる，整数の座標 (x, y) を持つ"点"からなる座標面 xy でできている．画面の境界は変数 $A1$ と $A2$ で与えられるが，球の描写がその境界をはみ出さないだけでなく，玉突台の境界の描写にも適すように配慮しなければならない．したがって，この段階では，画面を点の数の多いグラフィック画面にし，玉突台の境界を描き，球の中心の位置の限界を計算する（$G1=A1-2a-1$，$G2=A2-2a-1$ の型の）ことが必要である．

最初の衝突の計算　この段階では，最初の衝突の時刻 TC を計算し，衝突の型を変数 K の値として，また衝突する球の番号を $N1$, $N2$ の値として記録しなければならない．そのため，この段階の計算の結果は，次のようになる：最初に衝突した球を $N1$ とし――水平な壁との衝突を $K=1$，垂直な壁との衝突を $K=2$，他の球（$N2$ とする）との衝突を $K=3$ とする．

実際の計算では，最初に大きな TC および $N1=0$, $N2=0$, $K=0$ を与えておく．それから順次（球の番号に従って），各球が時刻 TC までに画面の境界に達しないことを確かめる（$x+v_x \times (TC-T) < G1$ の型の条件）．もし到達したなら，その壁と衝突する時刻を決定し，それに応じて TC, $N1$ および K を変更する（このような手続きにより，球が左の壁に垂直に衝突する時間を求めようとするときに生ずる，コンピュータにゼロで割り算させる危険を避けることができる）．つぎに，第2のサイクル（$N3$ について1から $N-1$ まで，$N4$ について $N3+1$ から N まで）において，TC までの時間に球 $N3$ と $N4$ とが衝突しないことを確かめなければならず，もし衝突する

ならば，それに応じて TC，$N1$，$N2$，およびKを変更する．

2個の球，たとえば1と2の衝突の計算は，時間 t について2次の，次の方程式を取扱うことになる：

$$(x_1+v_{1x}t-x_2-v_{2x}t)^2+(y_1+v_{1y}t-y_2-v_{2y}t)^2 =4a^2.$$

この式は次のことを意味する：時間 t の間に2個の球の中心間の距離が $2a$ になる．もし，この式の判別式Dが正ならば，2個の球は衝突する（$D=0$ のときは接触する）；小さい根は衝突の時刻を与え，大きな根は，2個の球が“相互浸透”した後に“接触する”時刻を与える．負の根は，過去に衝突したことを示す．

新しい座標の計算　プログラムの内容は，“時間を ΔT だけずらす”ことである．もし $TC>T+T0$ ならば，衝突は予見されず，球の古い座標を記憶するために $T1=T0$ と置かねばならない（画面上で古い図を消去するために）．それから $T1$ 時間後の新しい座標を計算する．

もし，$TC<T+T0$ であれば，$T1=TC-T$ と置かねばならない．前と同じことを繰り返して，新しい $N1$ 番目の球の($K\neq3$ のとき)，あるいは $N1$ 番目と $N2$ 番目の球の（$K=3$ のとき）新しい速度を改めて計算することになる．球の壁との衝突は，前論文の問題3にあるように，直ちに回答が得られる：球1と2が衝突するとき，新しい速度 v^n は，次の式で決められる：

$$\begin{aligned}
VI&=\{(x_2-x_1)(v_{2x}-v_{1x})\\
&\quad+(y_2-y_1)(v_{2y}-v_{1y})\}/4a^2,\\
v_{1x}{}^n&=v_{1x}+VI(x_2-x_1),\\
v_{2x}{}^n&=v_{2x}-VI(x_2-x_1),\\
v_{1y}{}^n&=v_{1y}+VI(y_2-y_1),\\
v_{2y}{}^n&=v_{2y}-VI(y_2-y_1).
\end{aligned}$$

画面上へのデータの出力　次にやることは，われわれが何を知りたいかにかかっている．もし球の運動ならば，球の古い画像を消して（そのためには，古い座標を記憶しておく）画面を背景の色にし，球を新しい位置に基本色で描く．しかし球の数が多いときには，それを描くのに長い時間を要するので，古い画像を消さないで，古い位置と新しい位置における球の中心を結ぶ線分だけを描くことにしてもよい――すると画面には軌跡ができる．画面をキャラクター画面にしておいて，そこに図形は描かないことにしてもよい．その代り，系のいくつかの特性量（例えば平均速度）を計算し，それを画面の一定の位置に出力するようにする．

モデルの制御　ここではまず，T に $T1$ を加えなければならない．そして最後に次の命令を出す：$TC>T$ のとき――新しい座標を計算する段階へ，$TC\leqq T$ のとき――最初の衝突を計算する段階へ．これによってプログラムは過程をくり返し，球の運動を自動的に画面上に描いて見せる．しかも，球が衝突したとき画面は，次の新しい最初の衝突を計算するまでの時間“凍結”される．

これらの演算の間，どのキーにも触らずに，コンピュータを見ているだけでよい．キーを押さなければ，計算は継続され，押せば何らかの操作をさせられる．

例えばここで，次のように入力するようにすることができる：“S”――停止；“R”――スタート；“B”すべての速度を逆にし，基本色の色調を変えて計算過程を実行する；“>”――$T0$ を2倍にする；“<”―― $T0$ を半分にする（もし，そのような演算が TC を変化させるなら，その次の命令は最初の衝突の計算に移るようにしなければならない）；“0”―― 画面の準備の段階へ移るコマンド（画面の更新）；“E”―― 作業の終了とモデルの作業方式を制御する他のコマンド．衝突の瞬間に音を出すようにすると，モデルは活きいきしたものになる．そのようなことは，一般にあなたとあなたのコンピュータの能力に全面的に依存している．

（Basic 数学，’92，11月号）の練習問題の答．

1. $\{(v_{1x}-v_{2x})^2+(v_{1y}-v_{2y})^2\}^{-1}\{(v_{2x}-v_{1x})(x_1-x_2)+(v_{2y}-v_{1y})(y_1-y_2)-[4a^2\{(v_{2x}-v_{1x})^2+(v_{2y}-v_{1y})^2\}-\{(v_{2x}-v_{1x})(y_1-y_2)-(v_{2y}-v_{1y})(x_1-x_2)\}^2]^{1/2}\}$
2. $\vec{v}_1{}'=-\vec{v}_2$, $\vec{v}_2{}'=-\vec{v}_1$
3. $\vec{r}_2-\vec{r}_1=2a\vec{n}$ とおき，$\vec{n}$ の向きに x 軸をとると，$\vec{v}_1{}'=(\vec{v}_2\cdot\vec{n},|\vec{v}_1\times\vec{n}|)$, $\vec{v}_2{}'=(\vec{v}_1\cdot\vec{n},|\vec{v}_2\times\vec{n}|)$

（訳　こじま　ひでお）