POSSIBLE IMPROVEMENTS TO THE CARNEGIE INSTITUTE OF TECHNOLOGY SYNCHRO-CYCLOTRON

M. H. Foss
Nuclear Research Center
Carnegie Institute of Technology
Saxonburg, Pennsylvania

Abstract

Our synchro-cyclotron is a 450 MeV machine. We want to increase the beam energy and intensity. The magnetic field will be increased by 6 almost radial high field regions. The steel will come within 2 inches of the midplane. A coil close to the gap will reduce the leakage flux. The acceleration electrodes will be confined to two opposite valleys and will be frequency modulated. The machine improvement is still in the talking stage. I wish to describe two computer programs which may help us to a more definite proposal.

Ion Orbits in a Magnetic Field

This calculation gives the orbit of an ion in a magnetic field. Magnetic fields tend to be "smooth". This property is exploited by predicting the field along the orbit from previously observed values. The actual variables predicted are the radius of the orbit, (r), and the rate of rotation of the radius about the velocity, (db_2/dt_2). These are written as power series in the variable:

$$dt_2=ds/r$$
$$r=R_0+R_1t_2+R_2t_2^2+\ldots.$$
$$db_2/dt_2= B_1 +2B_2t_2+3B_3t_2^2+\ldots.$$

The angular coordinates of the radius and velocity may be given by a matrix,(M). The first column of the matrix gives the direction cosines of the radius. The second column is the direction cosines of the velocity. In terms of the angles shown in figure 2

$$\cos(r,x)=\cos(b)\ \cos(t)$$
$$\cos(r,y)=\cos(b)\ \sin(t)$$
$$\cos(r,z)=-\sin(b)$$

$$\cos(v,x)=-\cos(a)\ \sin(t) + \cos(t)\ \sin(a)\ \sin(b)$$
$$\cos(v,y)= \cos(a)\ \cos(t) + \sin(t)\ \sin(a)\ \sin(b)$$
$$\cos(v,z)= \cos(b)\ \sin(a)$$

The position of the particle may be represented by a column vector,(P).

These matrices may be written for a coordinate system which has the (x)axis parallel to the radius and the (y)axis parallel to the velocity (see figure 1):

$$dM_2=\begin{pmatrix} 1 & -dt_2 & db_2 \\ dt_2 & 1 & 0 \\ -db_2 & 0 & 1 \end{pmatrix} \quad \text{and} \quad dP_2=\begin{pmatrix} 0 \\ r\ dt_2 \\ 0 \end{pmatrix}$$

These are the equations of motion. They may be written for the coordinate system in which a step is calculated:

$$M_1+dM_1=M_1\ dM_2 \quad \text{and} \quad dP_1=M_1\ dP_2$$

These 12 differential equations may be reduced to

$$dx_1=r(\sin(a_1)\sin(b_1)\cos(t_1)-\cos(a_1)\sin(t_1))\ dt_2$$
$$dy_1=r(\sin(a_1)\sin(b_1)\sin(t_1)+\cos(a_1)\cos(t_1))\ dt_2$$
$$dz_1=r\ \sin(a_1)\cos(b_1)\ dt_2$$
$$db_1=\cos(a_1)\ db_2 - \sin(a_1)\ dt_2$$
$$dt_1=\sec(b_1)\ (\cos(a_1)\ dt_2 + \sin(a_1)\ db_2)$$
$$da_1=\sin(b_1)\ dt_1$$

These equations may be integrated as power series in (t_2) by using the series given above for (r) and (db /dt). The boundary conditions are

$t_2 = -u$ at the beginning of a step and
$M_1 = 1$ and
$P_1 = 0$ at the end of a step.

In table 1, positions are calculated to 3rd order, angles to 4th.

The matrices at the end of a step in the stationary coordinate system,(M_{new}), and (P_{new}) are

$$M_{new}\ M_1(-u) = M_{old}$$
$$P_{new} = P_{old} - M_{new}\ P_1(-u)$$

The magnetic field at (P_{new}) may now be determined. If the radius and angle at this position do not agree with the projected values then a smaller step size must be used.

One method of doing this calculation is given in an ALGOL like language in table 1. Periods are used in identifiers to make them more readable. (q1, q2, q3) are direction cosines of the radius in the stationary coordinate system. (q4, q5, q6) are those of the velocity. The variables (t7, t8, etc.) are used for temporary storage. The procedures (increase.step.size.if.desired) and (to.cut.error.cut.step.go.back) are just put in to indicate that these operations are part of a complete program. The coefficients (R_0,R_1,R_2,etc.) have been eliminated by using the 0th, 1st, 2nd, etc. differences in the observed radii, (r0d, r1d,r2d, etc.). The coefficients (B_1,etc.) have been eliminated with the differences (b1d,b2d,b3d,etc.). (dt) is the change in longitude in the stationary system. (a.sum) is the sum of (a) in the stationary system and (a1).

The real procedure (fast.sqr) rapidly extracts the roots of numbers equal to or somewhat less than one. The representation will depend on the hardware. For example:

```
real procedure fast.sqr(x); value x; real x;
begin real half.x, half.root;
check.range.of(x);
fast.sqr:=half.x/((half.root:=B((half.x:=
    .5*x)+A)+C)+half.root)+half.root; end;
```

Here (A,B, and C) are constants and (check.range.of) should not be a separate procedure. In order for this procedure to be effective in calculating (cos.b) from (q3), (b) must be a small angle. This restriction can be met by making (z)axis parallel to the transport axis in a transport system or parallel to the midplane field in a cyclotron.

In the procedure outlined in table 1, orthogonality of the matrix referred to the stationary system is lost due to round off errors. This may not be important. If it is it may be periodically corrected or a somewhat slower matrix multiplication can be used which maintains orthogonality.

TABLE 1

```
procedure step; begin  real  t7,
     t8,t9,t11,t12,t13,t14,t15,t16,t17,t18,t19,
     t20,t21,t22,t23,t24,t25,t26;
old.x:=x; old.y:=y; old.z:=z;
old.q3:=q3; old.cos.b:=cos.b;
old.sin.a:=sin.a; old.cos.a:=cos.a;
old.sin.t:=sin.t; old.cos.t:=cos.t;
increase.step.size.if.desired;
b1d:=b1d+(b2d:=b2d+(b3d:=b3d+(b.error:=b4d)));
r0d:=r; r1d:=r1d+(r2d:=r2d+r3d);
z1:=(x1:=(k1d3*r1d+r0d)*c5)*b1d*k1d3;
y1:=((k5d6*r2d+r1d)*.5+r0d)*c2;
cos.t1:=fast.sqr(1-sin.t1*(sin.t1:=
     (t7:=(2.125*b2d+b1d)*b1d)*c1+c2));
t8:=(cos.a1:=(1-sin.a1*(sin.a1:=
     (((k1d3*t7*b1d+b3d)*k15d14+b2d)*
     k7d6+b1d)*c3))*old.cos.a;
t9:=(cos.b1:=fast.sqr(1-m.sin.b1*(m.sin.b1:=
     (b1:=(b2d*c6+b1d)*c4+b3d+b.error)*
     b1*b1*k1d6-b1)))*old.q3;
t13:=(t12:=(t11:=(cos.a.sum:=-old.sin.a*
     sin.a1+t8)*old.cos.b)*m.sin.b1
     +t9)*cos.t1;
r.cos.b:=1/(cos.b:=fast.sqr(1-q3*(q3:=
     (t14:=(sin.a.sum:=old.sin.a
     *cos.a1+old.cos.a*sin.a1)*
     old.cos.b)*sin.t1+t13)));
t15:=(sin.dt:=(cos.a.sum*sin.t1-cos.t1
     *sin.a.sum*m.sin.b1)*r.cos.b)*
     old.sin.t;
t16:=-old.q3*m.sin.b1;
t17:=(q1:=(cos.dt:=fast.sqr(1-sin.dt*
     sin.dt))*old.cos.t-t15)*cos.b)*x1;
t18:=(q2:=(sin.t:=old.cos.t*sin.dt+
     old.sin.t*cos.dt)*cos.b)*x1;
t20:=(t19:=(cos.a:=(t11*cos.b1+t16)*
     r.cos.b)*cos.t)*q3;
t22:=(t21:=(sin.a:=(q6:=t14*cos.t1-
     t12*sin.t1)*r.cos.b)*cos.t)*q3;
t24:=(q5:=(t23:=-sin.a*sin.t)*q3+t19)*y1;
y:=(q8:=(t25:=-cos.a*cos.t)*q3-t21)*
     z1+t24+t18+old.y;
t26:=(q7:=-t23-t20)*z1;
x:=(q4:=t25-t22)*y1+t26+t17+old.x;
z:=q3*x1+q6*y1+q9*z1+old.z;
find.hx.hy.hz.at.x.y.z;
abs.r.error:=abs(r3d:=(r:=h.rho/
     (hz1:=hx*q7+hy*q8xhz*q9))
     -r0d-r1d-r2d);
error.per.radian:=(error.per.step:=
     abs(b4d:=(hx*q1xhy*q2+hz*q3)/
     hz1)*r)*c7;
to.cut.error.cut.step.go.back;
end;
/////////////////////////////////////
The c's vary with step size,(u).
c1:=u/3;
c2:=u-u*u*u/6;
c3:=-u/2xu*u*u/24;
c4:=1-u*u/6;
c5:=u*u/2;
c6:=(1-5*u*u/24)/(1-u*u/6);
c7:=1/u;
k1d3:=1/3;
k1d6:=1/6;
k5d6:=5/6
k7d6:=7/6;
k15d14:=15/14;
```

Field Program

We have a program for calculating the magnetic potential on the points of a mesh. It can handle air, steel and current. The geometry can be 2 or 3 dimensional and cartesian or cylindrical. The boundary conditions can yield a repeating structure like a spiral machine (or a full 360 degree structure) or a reflected structure such as the other half of a radial ridge. A midplane or other structure with constant potential can be described. The scale factors on the axes need not be the same and the number of mesh points on each axis can be varied to suit the problem. Finally the interface between air and steel need not be 1/2 way between mesh points.

The first procedure modifies the program so that it will solve the desired problem. Since there are 18 parameters this is broken down into three procedures. The first gives an alternate place to get data for each of the 6 directions. This is to be used at the surface where data from one or more axes is not available. The second procedure lists the number of mesh points in each direction and the distance between points on each axis. The number of mesh points in the (z)direction is odd. The number of points in the (y) direction is odd if the mesh is more than one point wide in the (x)direction. This is done to simplify calculating points in the desired order. The third procedure in this group also has 6 parameters. The first, if it is greater than one, is the radius of the inner sheet of mesh points. If it is one an axis is provided. If it is zero, flux perpendicular to the axis will be neglected when permiability on the axis is being calculated. If this parameter is negative cartesian geometry is assumed. The second and third are the air and steel over-relaxation constants. The fourth parameter is the number of iterations between permiability calculations. The fifth is saturation flux density. And the last is the total number of amper turns in the coil.

The calculation of a point is based on the divergence of flux being zero. This calculation is done by one of many routines. The program is directed to the best routine by a few of the least significant bits in the 32 bit fixed point potential. There are complete routines for air and steel. These are relatively slow. Fast routines with fixed addressing are also available. Special routines for the axis and for constant potential (with either constant or variable permeability) are included. The criterion for including a routine was that the "average" running time should be reduced by 0.1 machine cycle per instruction.

For each point in the mesh a location is set aside for the reciprocal of the permeability. If an air point has one length to a steel interface not equal to 1/2, or one current cut, then this information replaces the permeability data. If extra information is needed at a steel point or more than one word at an air point then the permeability data is replaced by the address of a list. This list then contains the necessary data.

The resulting program has about 2000 instructions. The average run time is several miliseconds per point (50 microsecond multiply time). Experience with the program is limited.

The mesh can be divided into two sets of points. Members of one set have only members of the other set as nearest neighbors and vice versa. This program is arranged so that all members of one of these sets are calculated and then all members of the other. It has not been shown that this improves the convergence. However it does simplify analysis of convergence. One can write a difference equation which involves only members of one set. This equation can be separated into a function of iteration number and a function of position. This has been used to estimate the over-relaxation constant for all air problems but has not been set up for problems involving steel.

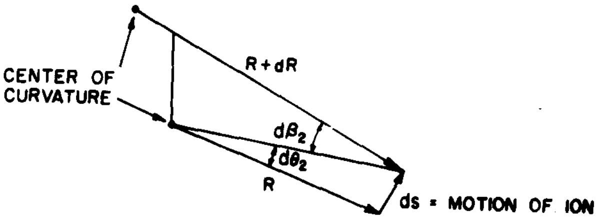

Figure 1.

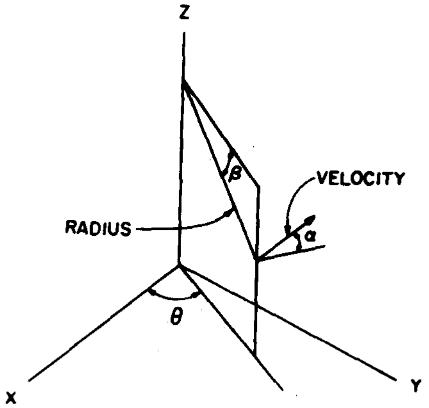

Figure 2.

DISCUSSION

TELEGDI: My remarks are made at this time because this is the last paper, as far as I can see, on improvement programs for existing synchrocyclotrons. A very small point has been bothering us ever since the Virginia Conference on Cyclotrons; the information that people seem to have and give you about the existing beams is very uncertain, sometimes conflicting. What really matters to the users of these machines is not the beam, but the secondary beam, or the extracted beam, not the circulating beam, which you would like to know. So we have finally, with some delay, taken the step which we promised to take over two years ago. At the Chicago cyclotron, we have prepared six carbon targets which have inner heaters. One can use them to produce mesons or whatever one wishes; they are normally used as the standard target.. One can put a battery on this heater, and calibrate the watts. Everybody can agree as to what beams he can get at a given level of wattage. We have six such targets which have been made for distribution among people who have synchrocyclotrons. You can talk to us at the lunch interval, if you wish.

FOSS: There will still be rhubarbs about internal beam because the ν_x and ν_z are different, and so on, but it will help.