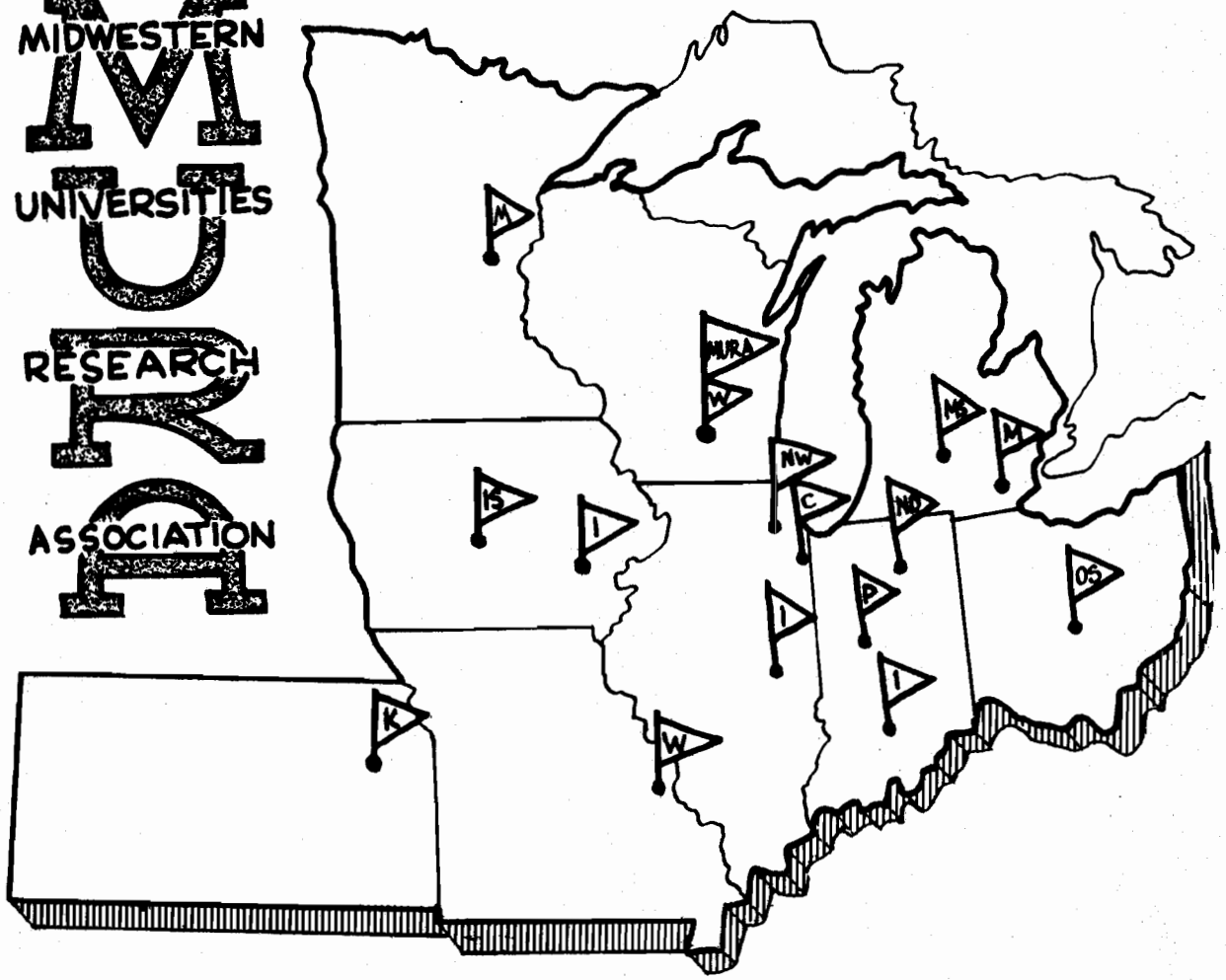
629

Fermilab Library
0 1160 0038895 3

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION




# THE PARMESH DYNAMICS PROGRAM

Jess Anderson and George Parzen

**REPORT** **NUMBER** 629

MURA-629
UC-28: Particle Accelerators and
High Voltage Machines
TID-4500 (16th Edition)

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION*

2203 University Avenue, Madison, Wisconsin

# THE PARMESH DYNAMICS PROGRAM

Jess Anderson and George Parzen**

September 22, 1961

## ABSTRACT

The PARMESH dynamics program is a Fortran Program which integrates the equations of motion of a particle in a magnetic field. The magnetic field is computed on a mesh and this mesh is written on tape. Various field inputs are available, only one of which is described in this report. The other field inputs and other features of the program will be presented in subsequent reports. The magnetic field is entered on the median plane. The field off the median plane is calculated as a power series in z by numerically differentiating the median plane field.

*AEC Research and Development Report. Research supported by the U. S. Atomic Energy Commission, Contract No. AT(11-1)-384.

**On leave from the University of Notre Dame, Notre Dame, Indiana.

This report describes the PARMESH program. Because of the complexity of the program, a series of reports is being planned. The first of these reports will present a general picture of the program and its use for one type of field input. Further details of the program and more complicated field inputs will be presented in subsequent reports.

## I. Problem Solved by PARMESH.

The PARMESH program integrates the differential equations of motion for a charged particle in a magnetic field. These equations may be written in the form

$$\frac{dx}{d\theta} = \frac{(A + x)\, p_x}{(1 - p_x^2 - p_y^2)^{1/2}} \tag{1.1a}$$

$$\frac{dp_x}{d\theta} = (1 - p_x^2 - p_y^2)^{1/2} - (A + x)\lambda \left[ H_y - p_y H_\theta/(1 - p_x^2 - p_y^2)^{1/2} \right] \tag{1.1b}$$

$$\frac{dy}{d\theta} = \frac{(A + x)\, p_y}{(1 - p_x^2 - p_y^2)^{1/2}} \tag{1.1c}$$

$$\frac{dp_y}{d\theta} = (A + x)\lambda \left[ H_x - p_x H_\theta/(1 - p_x^2 - p_y^2)^{1/2} \right]. \tag{1.1d}$$

In the above the radial coordinate is given by $r = r_o\,(A + x)$ and $z = r_o\, y$.

The field $H_y$ is entered on the median plane where $H_\theta = H_r = 0$. Off the median plane the fields $H_y$, $H_r$, $H_\theta$ are found using a power expansion in $y$ according to the following equations:

$$H_y = B_y - \frac{1}{2} \nabla^2 B_y \cdot y^2 + \cdots \tag{1.2a}$$

$$H_x = (\partial B_y/\partial x) \cdot y - \frac{1}{6} \frac{\partial}{\partial x} \nabla^2 B_y \cdot y^3 + \cdots \tag{1.2b}$$

$$H_\theta = \frac{1}{A + x} \frac{\partial B_y}{\partial \theta} \cdot y - \frac{1}{6} \frac{1}{A + x} \frac{\partial}{\partial \theta} \nabla^2 B_y y^3 + \cdots . \tag{1.2c}$$

$B_y = B_y (r, \theta)$ is the field $H_y$ for $y = 0$. $\nabla^2$ is the Laplacian operator $(d^2/dr^2 + (2/r)\, d/dr + (1/r)^2\, d^2/d\theta^2)/r_o^2$. The median plane field $B_y$ is numerically differentiated to obtain the coefficients in the above power series expansion in y.

One rather general way the field may be entered in PARMESH is according to the equation

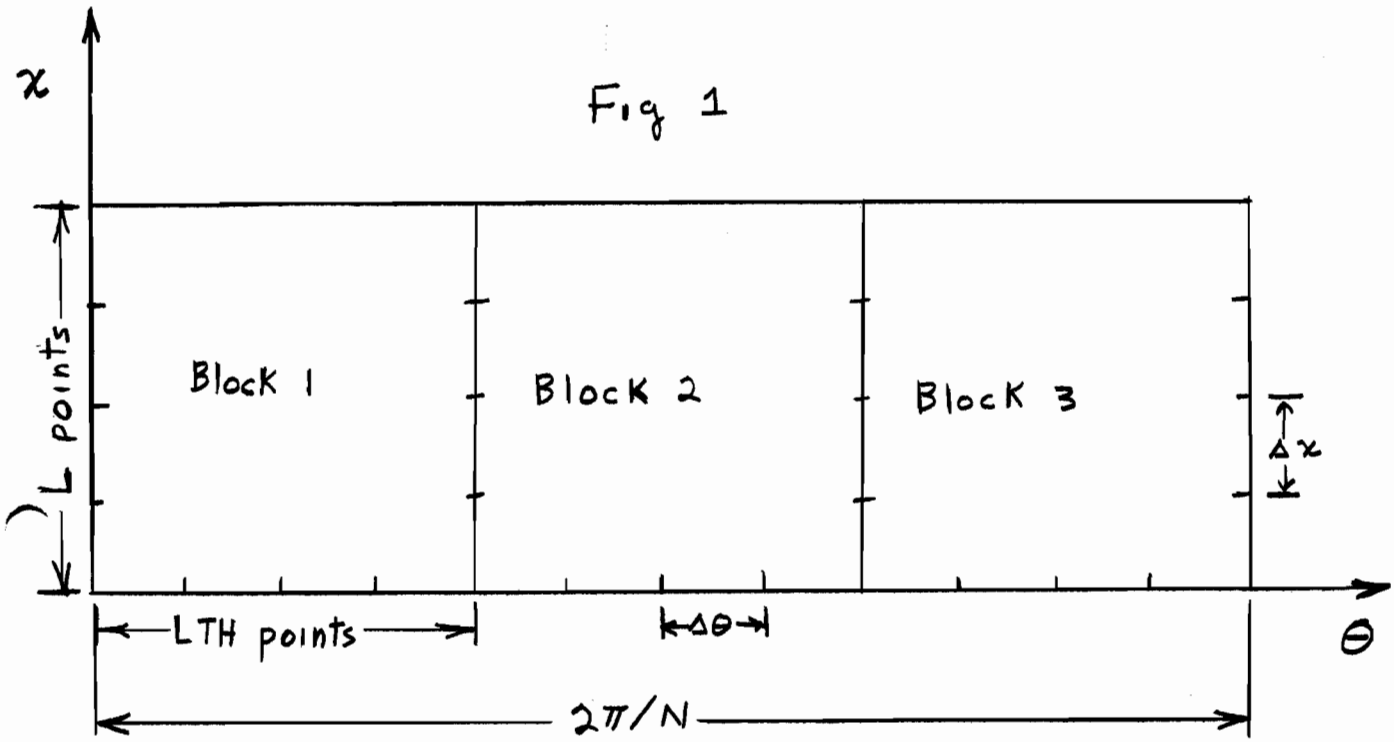
$$B_y = \Lambda (A + x)^k \sum_{n=o}^{NMX} C_n (r) \cos\left[n S \theta - \frac{n}{w} \ln (A + x) - \beta_n (r)\right] . \tag{1.3}$$

The harmonics $C_n (r)$ are given at intervals of $\Delta x$ starting with the radius $r_1 = r_o (A + x_1)$ .

## II. Mesh Construction.

The program takes the median plane field, $B_y$, given by Eq. (1.3) and constructs from it a mesh. The mesh interval in the r-direction is $\Delta x$, and the mesh starts at $x = x_1$. There are L mesh points in the r-direction, L being supplied by the user. In the $\theta$-direction the mesh interval is $\Delta\theta = \frac{1}{2} (2\pi/N)/NRK$, where $2\pi/N$ is the period of the field and NRK is the number of integration steps in a period using a fourth order Runge-Kutta integration procedure. NRK is provided by the user. In the $\theta$-direction the mesh starts at $(N\theta/2\pi)_o$ and has an extent of $2\pi/N$, the period of the field.

The mesh is broken up into blocks in the θ-direction. Each block contains LTH points. There are NRKB = (LTH - 1)/2 integration steps in each block, and there are NBL = NRK/NRKB blocks in total. The mesh is shown schematically in Fig. 1 for the case of three blocks.




This mesh is computed and written on tape called the PARMESH tape. In the dynamics part of the program the PARMESH tape is read into the memory a block at a time and the field on the median plane is found by interpolation between the mesh points.

## III. Data Input.

The data input cards for a PARMESH run is divided into three parts, each part being separated from the following cards by an END DATA card.

These three parts of the data input deck will be called Data #1, Data #2, and Data #3.

The data input cards in the part Data #1 informs the program whether the data to compute the median plane field will be read from cards or tape and which procedure will be used to compute the median plane field from this data. Data #1 contains the following parameters with the addresses shown.

DATA # 1.

| Parameter | Address |
|---|---|
| ID | 1 |
| FEN | 3600 |
| L | 28 |
| LTH | 16 |
| RDY1 | 29 |
| | END DATA |

The ID is the run identification number. FEN (originally called the field number) has the following possible values:

FEN = 2, field data to be read from cards.

FEN = 6, field data to be read from tape.

FEN = 5, no mesh is to be computed as the mesh was computed in some previous run and the PARMESH tape was saved.

L is the number of mesh points in the r-direction. $L_{max} = 35$.

LTH is the number of mesh points in the θ-direction in a block. $LTH_{max} = 65$.

RDY1 is the radius at which the mesh begins. It is given only when FEN = 6 and the Fourier coefficients of the field is read from tape. RDY1 tells the program at which record it should start reading the Fourier coefficient tape.

The data input cards in the part Data #2 give the data necessary to compute the field and some other geometrical properties of the mesh to be calculated.

DATA #2.

| Parameter | Address | |
|---|---|---|
| A | 21 | Mesh Geometry Data |
| $x_1$ | 26 | |
| $\Delta x$ | 27 | |
| N | 21 | |
| NRK | 15 | |
| $(N\theta/2\pi)_o$ | 6 | |
| $\Lambda$ | 20 | Field Data |
| NMX | 25 | |
| k | 22 | |
| 1/w | 23 | |
| S | 3601 | |

DATA # 2 (continued)

| Parameter | Address | |
|---|---|---|
| $C_o$ (1) to $C_o$ (34) | 30 $\longrightarrow$ 64 | Field Data |
| $C_1$ (1) to $C_1$ (34) | 65 $\longrightarrow$ 69 | |
| $\downarrow$ | | |
| $C_{50}$(1) to $C_{50}$(34) | 1780 $\longrightarrow$ 1814 | |
| $\beta_1$ (1) to $\beta_1$ (34) | 1815 $\longrightarrow$ 1849 | |
| $\downarrow$ | | |
| $\beta_{50}$(1) to $\beta_{50}$(34) | 3565 $\longrightarrow$ 3599 | |
| | END DATA | |

A describes how the radial variable x is related to $r_1$, $r = r_o (A + x)$.

$x_1$ is the beginning of the mesh in the r-direction.

$\Delta x$ is the mesh spacing in the r-direction.

N gives the period in the $\theta$-direction. The period is $2\pi/N$.

NRK is the number of integration steps in a period and gives the total number of mesh points in a period of $2\pi/N$ in the $\theta$-direction.

$(N\theta/2\pi)_o$ gives the beginning of the mesh in the $\theta$-direction in units of $2\pi/N$.

The rest of the parameters in Data #2 gives the data necessary to compute the field and they are defined by Eq. (1.3).

The data input cards in the part Data #3 give the data required by the dynamics such as the initial conditions, length of the run, etc.

DATA # 3

| Parameter | Address |
|---|---|
| $x$ | 2 |
| $p_x$ | 3 |
| $y$ | 4 |
| $p_y$ | 5 |
| $(N\theta/2\pi)_o$ | 6 |
| NP | 12 |
| NE | 13 |
| MEQ | 14 |
| X MAX | 7 |
| PX MAX | 8 |
| Y MAX | 9 |
| PY MAX | 10 |
| $\lambda$ | 20 |
| HD | 17 |

$x$ is the initial value of $x$.

$p_x$ is the initial value of $p_x$.

$y$ is the initial value of $y$.

$p_y$ is the initial value of $p_y$.

$(N\theta/2\pi)_o$ gives the initial $\theta$ of the run.

NP the coordinates of the particle are printed out every NP integration steps.

NE the run will continue for NE integration steps.

MEQ takes on the values 1 or 2.

MEQ = 1 means one dimensional motion and $y = p_y = 0$.

MEQ = 2 means two dimensional motion, $y \neq 0$. MEQ is internally set equal to 2.

X MAX, Y MAX are the largest values x and y may take on and are internally set at 0.5.

PX MAX, PY MAX are the largest values y and $p_y$ may take on and are internally set at 0.999.

$\lambda$ occurs in the equations of motion and is inversely proportional to the momentum, p, of the particle. The energy may be varied by varying $\lambda$.

HD tells the program up to which power of y the field is to be computed off the median plane. If HD = 0, only the first term in Eq. (1.2) is used. If HD = 1, the second term in Eq. (1.2) is also calculated.

## IV. Tapes Used.

The PARMESH program writes the field mesh it computes on tape; this tape is called the PARMESH tape and is on tape drive #2. This tape may be saved and used for future dynamics programs.

The PARMESH program itself is on tape and it is called the PARMESH master tape and is on tape drive #1.

The Fourier coefficients to compute the field may be supplied by cards or tape. If they are given on tape, then the Fourier coefficient tape called the JAN tape is on tape drive #3. The JAN tape has the following record structure:

Record 1. This record contains five identification numbers, TID, LR, R1, DR, NMX.

TID is the JAN tape ID number.

LR is the number of points in the r-direction at which the Fourier coefficients are given.

R1 is the first radius at which the Fourier coefficients are given.

DR is the radial distance between the points at which the Fourier coefficients are given.

NMX is the highest harmonic on the tape.

Record 2. The first LR words are the $C_o$ (r), and the second LR words are the $\beta_o$ (r).

Record 3. The first LR words are the $C_1$ (r), and the second LR words are the $\beta_1$ (r).

Record 4 up to the record NMX + 2 contain the rest of the harmonics in the same fashion.

All words on the tape are in floating point.