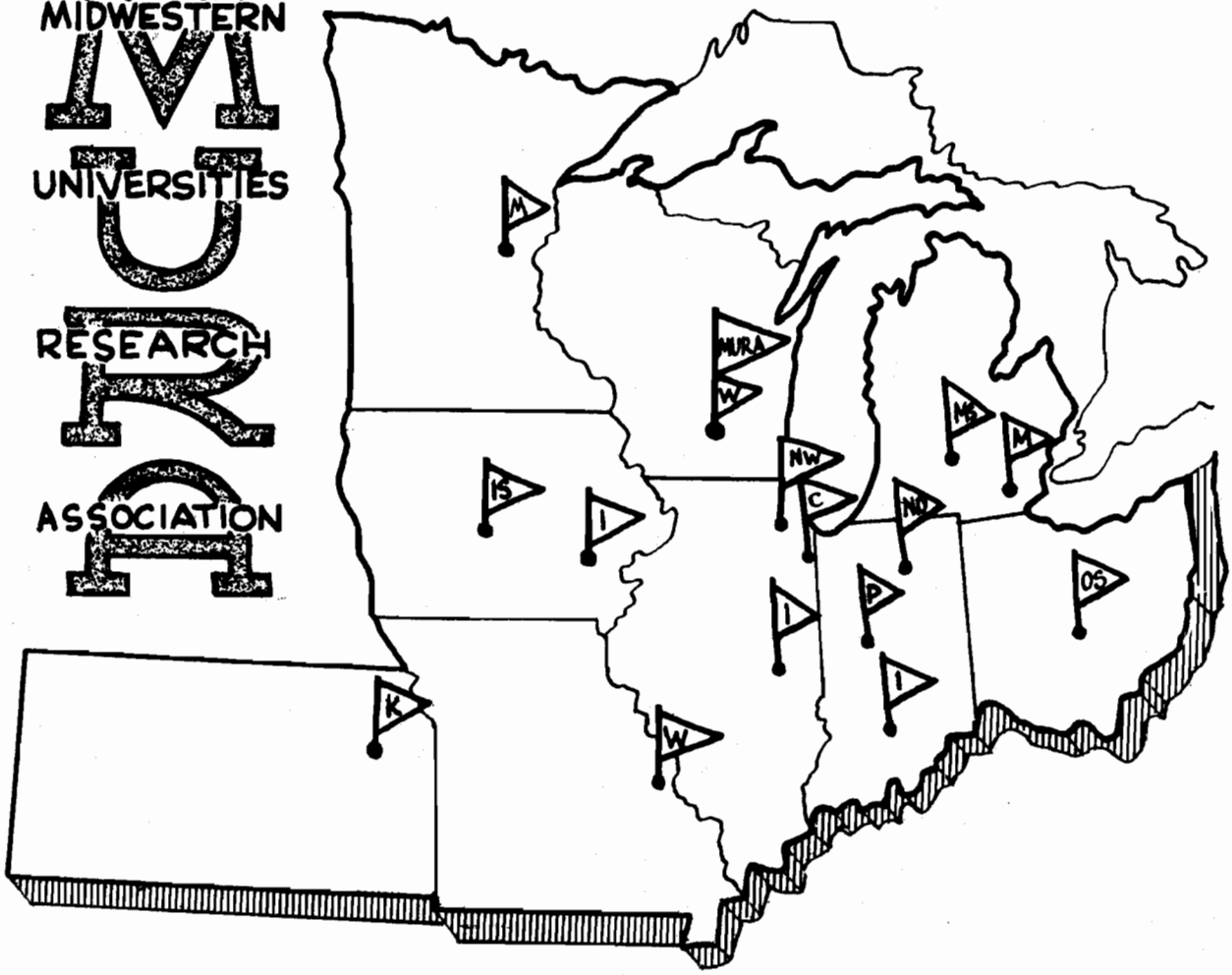
Fermilab Library
0 1160 0038742 7

442

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION



PROPOSED DIGITAL COMPUTER PROGRAM TO STUDY AN N-BODY PROBLEM WITH ONE DEGREE OF FREEDOM

A. M. Sessler

REPORT **NUMBER** 442

MURA-442
Internal

MIDWESTERN UNIVERSITIES RESEARCH ASSOCIATION*

2203 University Avenue, Madison, Wisconsin

PROPOSED DIGITAL COMPUTER PROGRAM TO STUDY AN N-BODY PROBLEM WITH ONE DEGREE OF FREEDOM**

A. M. Sessler
The Ohio State University, Columbus, Ohio

January 1, 1959

ABSTRACT

A digital computer program is suggested for studying a many-body problem with one degree of freedom and long-range interactions between particles. The program studies longitudinal space charge effects in a beam of particles subject to one external rf cavity.

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

**Research supported in part by the National Science Foundation.

I. INTRODUCTION

It has been pointed out[1] that for particles in an accelerator it is often an excellent approximation to assume that Liouville's Theorem is rigorously correct in μ-space. The proposal made here is based on this assumption.

Actually we at MURA are interested in general in three-dimensional problems. These are probably beyond the techniques of present day digital computers. However, two-dimensional problems may just be possible by digital computation and theoretical analysis may be able to reduce the three-dimensional problems to two-dimensional problems either exactly[2] or to good approximation. There are a number of one-dimensional problems of interest to the MURA group,[3] besides which the programming of a one-dimensional problem will aid greatly in gaining experience which may make the two-dimensional problems feasible. It thus seems advisable to seriously propose programming a one-dimensional problem.

II. DESCRIPTION OF THE PROGRAM

In our problem φ and W are canonical coordinates and momenta. The equation for the distribution function $\psi(\varphi, W, t)$ simply says that

---

[1] MURA-433.

[2] Such as in the case where energy is a good quantum number.

[3] In the interest of obtaining an operative program as quickly as possible, the present suggestion is only applicable to longitudinal space charge effects. Applications of the method to other problems such as transverse space charge effects will perhaps be a subject for future study.

density, as we move with the particles in φ-W space, is a constant. Hence

$$\frac{d\psi(\varphi, W, t)}{dt} = 0, \tag{1}$$

or

$$\frac{\partial\psi(\varphi, W, t)}{\partial t} = -\dot{\varphi}\,\frac{\partial\psi(\varphi, W, t)}{\partial\varphi} - \dot{W}\,\frac{\partial\psi(\varphi, W, t)}{\partial W}\,. \tag{2}$$

The quantities $\dot{\varphi}$ and $\dot{W}$ are given by the equations of motion for a particle, namely

$$\begin{aligned} \dot{\varphi} &= hF(W) \\ \dot{W} &= eV\sin\varphi - \dot{w}_s + 2\pi eh\,\frac{\partial v}{\partial\varphi}, \end{aligned} \tag{3}$$

where h, V, and $\dot{w}_s$ can be constants, F (W) is a given cubic polynomial in W, while $v(\varphi)$ is a function to be specified below. The program should also be planned so that at least V and $\dot{w}_s$ and perhaps the coefficients in F(W) may be made some simple functions of time such as in the TTT program. It should be noted that φ is a cyclic variable so that $\varphi + 2\pi = \varphi$. The function v is given by:

$$v(\varphi, t) = \int K(\varphi, \varphi') \int \psi(\varphi', W, t)\, dW\, d\varphi', \tag{4}$$

where $K(\varphi, \varphi')$ will be specified below. Observe that the kernel $K(\varphi, \varphi')$ is a property of the accelerator geometry and actually is only a function of $(\varphi - \varphi')$.

One should imagine that $\psi(\varphi, W, t)$ is given for $t = 0$ (by inserting a mesh of values), and we wish to determine $\psi(\varphi, W, t)$ for all subsequent t. Presumably $\psi(\varphi, W, t)$ will be stored in the fast memory on a two-dimensional grid of perhaps $n_\theta \times n_W$ points. From this grid of $\psi(\varphi, W)$ at the time t_n, we may obtain $\psi(\varphi, W)$ at the time $t_n + \Delta t$ by any standard method. For concrete example let us assume (to be simple minded) that the algorithm used is:

$$\psi(\varphi, W, t_n + \Delta t) = \left[-\dot{\varphi} \frac{\partial \psi}{\partial \varphi}(\varphi, W, t_n) - \dot{W} \frac{\partial \psi(\varphi, W, t_n)}{\partial W} \right] \Delta t + \psi(\varphi, W, t_n) \tag{5}$$

Thus one could proceed by first evaluating and storing $\dot{W}(\varphi)$ at the n_φ values of φ. Then one works through the mesh (taking the values of φ in order and for each value of φ doing all the values of W) evaluating by some simple differentiation formula $\frac{\partial \psi}{\partial \varphi}$ and $\frac{\partial \psi}{\partial W}$ and consequently by Eq. (5) the new value of $\psi(\varphi, W, t_n + \Delta t)$. After doing three columns or so (depending on the complexity of the interpolation formula for $\frac{\partial \psi}{\partial \varphi}$) one could erase the first column and enter the new values of $\psi(\varphi, W, t_n + \Delta t)$. In fact one might be tempted to calculate $\frac{\partial \psi}{\partial \varphi}$ and $\frac{\partial \psi}{\partial W}$ using some new $(t_n + \Delta t)$ values and some old values (t_n), thus allowing a simple evaluation of $\psi(t_n + \Delta t)$ at each point followed by an erasure of $\psi(t_n)$ and insertion of the new value.

The evaluation of $\dot{W}$ is not as complicated as might appear from a cursory examination of Eqs. (3) and (4). As one proceeds through an interation,

one simply sums $\psi(\varphi, W, t_n)$ over the n_W values of W associated with each value of φ, and stores the n_φ values of $\int \psi(\varphi, W, t_n)\, dW$. (It might prove more convenient to evaluate this quantity after completion of an iteration.) The quantity $K(\varphi - \varphi')$ should be inserted as a table, so that the evaluation of $v(\varphi, t)$ proceeds by approximately n_φ multiplications plus additions for each of n_φ values of φ. Then $\dot{W}$ is evaluated and stored for use in the iteration procedure.

The program should be capable of printing out after a specifiable number of iterations (different for each of the following):

1. The number of particles obtaining a value of $W_{min} < W < W_{max}$, where W_{min} and W_{max} may be specified as functions of time.
2. $\int \psi(\varphi, W, t)\, dW$.
3. $\int (W - W_o)^m \psi(\varphi, W, t)\, dW$, $m = 1, 2, \ldots$; W_o may be specified as a function of time.
4. $\int \left[\int W \psi(\varphi, W, t)\, dW \right] \begin{Bmatrix} \sin n\varphi \\ \cos n\varphi \end{Bmatrix} d\varphi$, $n = 1, 2, \ldots$
5. $\psi(\varphi, W, t)$ for all mesh points.

There should be convenient ways of entering $\psi(\varphi, W, t = 0)$ (such as by giving a boundary curve inside which ψ is the same on all points). The particularly simple (and rather useful) case in which $K(\varphi, \varphi')$ is simply zero for $\varphi' \neq \varphi$ might be programmed specially so as to obtain high speed, (i.e. don't multiply lots of zeros and then add them too).

The magnitudes and ranges of the various quantities involved are given here:

1. φ ranges continuously from 0 to 2π. The problem is cyclic so that $2\pi+\varphi=\varphi$.

2. The initial value of W is arbitrary. If energies are measured in units of 10^5 Mev, and time in seconds, then $eV \approx 10^{-5}$ and in a run of 1000 revolutions $\Delta W \approx (10^{-5})\,(10^{-6})\,(10^3) \approx 10^2$.

3. As has been observed, eV will be between about 10^4 ev and 10^7 ev.

4. $\dot{w}_s$ will be between zero and 10^7 ev.

5. h is an integer between 1 and 50.

6. ψ has an arbitrary normalization, but it is cyclic in φ so that $\psi(2\pi) = \psi(0)$. It probably will prove convenient to normalize ψ so that $\int \psi\, dW\, d\varphi = N$ where N is of the order of 10^{15} or less.

7. F (W) is positive or negative and ranges up to about 10^9 sec^{-1}.

III. DEDICATION

This proposal is dedicated to L. Jackson Laslett in the hope that he will become interested enough to convert it from its present form into a form useful to programmers.