A USER-FRIENDLY INTERFACE AND
EXTENSIONS TO THE POISCR PROGRAM PACKAGE

by

W. Hofer

Abstract

The basic idea of this user-friendly interface is that in a dialogue the user is asked to answer questions relevant to his problem, without being bothered with tedious details concerning curious job control cards. Thus the job creation is not only shortened and simplified, the risk of errors in the job cards is reduced to a minimum.

The extensions to the POISCR program package include two programs for data preparation which avoid the necessity of assigning logical mesh coordinates to the physical coordinates describing your problem geometry, one program to print a concise and practical map of potential and field values in a regular grid and one program to estimate eddy-current effects under certain conditions.

Geneva, Switzerland
August 1986
CONTENTS

1. INTRODUCTION
   1.1 The POISCR program package 1
   1.2 Wylbur exec files 1
   1.3 Additional programs 2

2. A USER-FRIENDLY INTERFACE
   2.1 Basic idea 2
   2.2 Defining the problem 3
   2.3 Solving the problem 3

3. GENERATING THE TRIANGLE MESH
   3.1 Introduction 4
   3.2 Describing the problem geometry 4
   3.3 Logical mesh design 5
   3.4 Variable mesh size 5

4. ESTIMATING EDDY CURRENTS 6

5. FURTHER INFORMATION 6

6. EXAMPLES 7
1. INTRODUCTION

1.1 The POISCR program package

The POISCR program package consists of a set of programs designed for the solution of Poisson's or Laplace's equation in two-dimensional regions and is available in the CERN program library (for IBM, operating system MVS); a detailed description is given by Chr. Iselin in the long write-up T604.

The individual programs are:

LATTCR a generator for the physical mesh (discretization of the given problem), requiring input data for the logical and physical coordinates describing the problem.

POISCR solves Poisson's or Laplace's equation for the potential, using the mesh generated by LATTCR.

FORCCR computes the forces on selected regions.

TRIPCR for plotting the triangle mesh and the potential distribution.

In version 3 of the POISCR program package (24 August 1983) the AUTOMESH program (automatic logical mesh design) has not been included. Nevertheless, a revision of the old AUTOMESH program was done by Chr. Iselin (3 February 1983). Essential parts of this revision have now been rewritten and extended (AUTMESH9) to allow for a variable mesh size, to improve the path finding algorithm and to support the user-friendly interface.

1.2 Wybur exec files

In order to use any program of the POISCR program package a job file containing control cards and input data has to be created; but writing such a job file is a tedious and error-prone task. Therefore a user-friendly interface consisting of Wybur exec files as well as some additional programs have been written. The Wybur exec files are:

AUT9 In a dialogue the user is asked to specify the geometry and other data relevant to the problem. All job control cards and all data cards necessary to run the programs PREAUT9 (data pre-processor) and AUTMESH9 (logical mesh design) are created automatically by the exec file.
POS  In a dialogue the user is asked to specify which programs of the POISCR program package are to be used and how to present the results.

POSED  In a dialogue the user is asked to specify the details necessary for running POISED (eddy-current estimations).

1.3 Additional programs

Some additional programs have been written to facilitate the use of the POISCR program package:

PREAUT9  a pre-processor to check and prepare the input data for AUTMESH9.

AUTMESH9  to design automatically a logical mesh accordingly to the given problem geometry, i.e. to assign logical mesh coordinates to the physical coordinates as it is needed in the input data for LATTCR. (AUTMESH9 is a largely revised version of AUTOMESH).

FEDIT  for printing a concise and handy map of potential values and its derivatives in a regular grid (an alternative to the somehow clumsy tables offered by POISCR).

POISED  to estimate eddy-current effects under certain conditions.

2. A USER-FRIENDLY INTERFACE

2.1 Basic idea

The basic idea of this user-friendly interface is that in a dialogue the user is asked to answer questions relevant to his problem, without being bothered with tedious details concerning curious job control cards or redundant input. Thus the job creation is not only shortened and simplified, the risk of errors in the job cards is reduced to a minimum.
This dialogue is split up into two parts, one (AUT9) to define the problem physically (specifying the geometry, the boundary conditions, the material properties, the sources), the other (POS) to launch the programs for solving the problem and presenting the results in the desired form. In addition, there is a third dialogue (POSED) to deal with eddy-current estimations.

As the POISCR program package is offered on the IBM under the operating system MVS, Wyibur exec files have been used to realize the dialogue.

2.2 Defining the problem

In a dialogue with the Wyibur exec file AUT9 the user is asked to specify the geometry and other data relevant to the problem. An unambiguous problem code ("PCODE", a name of 1 to 5 characters) has to be assigned to each individual problem to identify the corresponding data files, as well as a title line (up to 80 characters) which will be used to identify any output of any program.

To describe the problem its geometry is divided up into several regions to which different material properties and sources are assigned. Boundary conditions are represented by line regions. The geometry of each region is given by its boundary line, which has to be specified by a combination of straight lines and arcs of a circle or a hyperbola. The exec file AUT9 will ask for all the necessary details.

After this dialogue the exec file AUT9 will create a job file (file name PCODE.AUT) containing all the job control cards and all data cards necessary to run the programs for data preparation (PREAMUT9 and AUTMESH9). Executing this job will create the problem definition file PCODE.DEF that contains all input data for LATTICR and additional data for POISCR, i.e. all data necessary to define the problem for the POISCR program package.

2.3 Solving the problem

In a dialogue the Wyibur exec file POS will ask the user to specify some details concerning the programs to launch for solving the problem and for presenting the results in the desired form. These details and the data stored in the problem definition file PCODE.DEF will be used to create all the job and data cards necessary to run the POISCR program package as well as the programs to produce the graphic output.
3. GENERATING THE TRIANGLE MESH

3.1 Introduction

The program LATTUM is the generator for the physical triangular mesh (which represents the discretization of the problem), requiring input data for the logical and physical coordinates describing the problem split up in regions of different properties.

The AUTMESH9 program takes over the job of assigning logical mesh coordinates which outline the different regions (i.e. designing the logical mesh), and its data pre-processor PREAMUT9 further simplifies data preparation.

3.2 Describing the problem geometry

To describe the problem physically, its geometry, boundary conditions, material properties and sources are specified in a dialogue with the exec file AUT9. The problem has to be divided up into several regions, each region being specified by its boundary line, material properties and sources. To define the problem's boundary conditions special line regions are used.

For AUTMESH9 as well as for PREAMUT9 each region's boundary line is given as a sequence of line elements (straight lines, arcs of a circle or a hyperbola), being defined by a list of points to be connected by the respective line element.

Each point is given by its coordinates in a local cartesian or polar coordinate system together with the global coordinates of the respective origin. For an arc (specified by two points) the center is defined by the origin of the second point's local coordinate system. The exec file AUT9 will ask for all the necessary details and will indicate the respective default values.

To enable AUTMESH9 to find distinct logical boundary points even for a delicate geometry, it is absolutely necessary to indicate all subdivisions of a line if any part of it will be used for another region (otherwise zero triangle areas might result). To avoid this cumbersome data preparation; however, an input data pre-processor (PREAMUT9) has been written to check the input data and to insert automatically any necessary subdivision.
3.3 Logical mesh design

Once the geometry is defined, each region's boundary line has to be approximated by a polygon, the sides of which are sides of mesh triangles. All this work will be done by AUTMESH9. At first a regular triangle mesh (logical as well as corresponding physical coordinates) is set up accordingly to the required mesh size. Then the path finding algorithm allocates an appropriate sequence of logical mesh points to represent each region's boundary line by adjusting the corresponding physical mesh coordinates. Storing the previously defined lines enables the program to identify new lines, thus guarding against the risk of any mesh point being used a second time with different coordinates.

Last of all, the designed logical mesh representation of the problem will be given in the format required as LATTCR input data, that is a list of the logical as well as the corresponding physical coordinates to represent each region's boundary line as a polygon of mesh triangle sides.

3.4 Variable mesh size

Sometimes it is convenient to have a denser mesh inside critical regions or to have a larger outer region for establishing reasonable boundary conditions without exceeding a given total number of mesh triangles (e.g. for the program's dimension limit). The AUTMESH9 program allows one to construct such a variable mesh size.

The basic idea is to rearrange an initially regular triangle mesh before starting with the path finding algorithm, by applying a geometric transformation to the physical coordinates assigned to the logical mesh points. After the transformation the area is divided up into several subsets, each showing a constant mesh triangle size which may be different from the initial one or from that of any other subset. As it is only a rearrangement, the total number of mesh points remains unchanged.

Due to the algorithm inside LATTCR which tries to construct a physical triangle mesh inside any region which is as regular as possible, one should split up the problem in such a way that no region or what is left after successive region overwriting encloses subsets of different (or at least not comparable) mesh triangle sizes; otherwise negative or zero triangle areas might result due to an inopportune distortion of the mesh. Further information on details for using the variable mesh size is available (cf. Chapter 5).
4. ESTIMATING EDDY CURRENTS

Under certain conditions the program POISED (a modified version of POISCR) can be used to estimate eddy-current effects (T. Tortschanoff, private communication). The eddy-current density created in a conductor by slow variation of a two dimensional field represented by its vector potential \((0, 0, A_z)\) is given by the product of the electric conductivity and the time derivative of \(A_z\). The eddy current having the same direction as the vector potential and a homogeneous conductivity will be assumed as well as only slow variations of flux and a constant current rise rate. Thus POISED can estimate the eddy-current effects by approximating the time derivative by the difference quotient and by redefining the source term for each mesh point (triangle) accordingly before solving Poisson's equation.

To estimate eddy-current effects, the problem without eddy currents has first to be solved, using the AUT9 and POS exec files, and the POISCR output file (dump file) catalogued. Then by use of the dialogue with the exec file POSED one enters the eddy-current specific data (electric conductivity and current rise time) for the respective regions and the desired form for representing the output data, and the job file will be set up automatically.

5. FURTHER INFORMATION

The exec files AUT9, POS and POSED are members of the Wylbur library $IZ.HFR.ACL which may be called by, for example,

EXE FRO $IZ.HFR.ACL(AUT9).

Before starting with the dialogue, however, the user should be familiar with the CERN-POISSON program package user guide (long write-up T604 by Chr. Iselin).

Further information on details of the additional programs and examples of the use of the exec files are given in the following internal notes by the author (LEP-MA/WH/wh):

- AUT and POS exec files and modified AUTOMESH (28 January 1985)
- Variable mesh size with AUTOMESH (31 October 1985)
- PREAUT9 and AUTMESH9 (30 July 1986)

The author would like to thank Chr. Iselin for fruitful discussions about AUTOMESH and the POISCR program package.
6. EXAMPLES

Two different geometries are used to illustrate the logical mesh design as it is done by AUTMESH9 and the resulting physical mesh generated by LATTCR.

Figure 1 shows a part of the physical mesh for a C-shaped dipole with vacuum chamber, generated by LATTCR as the result of a constant mesh size design by AUTMESH9. Just a few region boundaries consisting of straight lines and arcs are sufficient to specify the complex geometry of the vacuum chamber and to make AUTMESH9 resolve the respective region's outline into a polygon composed of mesh triangle sides.

Figures 2 and 3 illustrate some possibilities offered by a variable mesh size used for a superconducting dipole. The different allocation of the same total number of mesh triangles to the respective regions is due to a different transformation table resulting in a different logical mesh design by AUTMESH9 for the same problem geometry.

In Figure 4a flux lines are shown for the C-shaped dipole, whereas those due to eddy-current effects in the vacuum chamber are given in Figure 4b.

Figure 1: C-shaped dipole with vacuum chamber: a part of the physical mesh generated by LATTCR, the result of a constant mesh size design by AUTMESH9.
Figure 2: Variable mesh size used for a superconducting dipole
a) The logical mesh design done by AUTMESH9
b) The resulting physical mesh generated by LATTCR
Figure 3: Variable mesh size used for a superconducting dipole
a) The logical mesh design done by AUTMESH9
b) The resulting physical mesh generated by LATTCR
Figure 4: C-shaped dipole with vacuum chamber
   a) Flux lines (no eddy currents; POISCR)
   b) Flux lines due to eddy currents (POISED)