ODE SOFTWARE WORKSHOP

Argonne National Laboratory
June 3 - 7, 1974

List of Participants

Leonard Brown
c/o William Gear
University of Illinois
Department of Computer Science
Urbana, Illinois 61801

George Byrne
Department of Mathematics
804 Schenley Hall
University of Pittsburgh
Pittsburgh, Pennsylvania 15260

Wayne Enright cancelled
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada

Fred N. Fritsch
Lawrence Livermore Laboratory
University of California
P.O. Box 808, L-310
Livermore, California 94550

C. William Gear
University of Illinois
Department of Computer Science
Urbana, Illinois 61801

Marilyn Gordon
Applied Mathematics Division, 2642
Sandia Laboratories
Albuquerque, New Mexico 87115

Alan Hindmarsh
Lawrence Livermore Laboratory
P.O. Box 808, L-310
Livermore, California 94550

Thomas E. Hull
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada

Fred Krogh
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 91103

[illegible] Bengt Lindberg
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada

Lawrence F. Shampine
Applied Mathematics Division 5121
Sandia Laboratories
Albuquerque, New Mexico 87115

Joan Walsh
Department of Mathematics
The University,
Manchester
M13 9PL
ENGLAND

David Sayers
NAG Central Office
Oxford University Computing
Laboratory
13 Banbury Road
Oxford OX2 6NN
ENGLAND

ODE Software Workshop

Travel and Lodging Information

Lodging

A limited number of rooms in the Argonne Guest Facility have been reserved for participants in the workshop. It will be necessary for some participants to stay in a motel about ten minutes driving time from the Laboratory. Anyone who prefers the motel may stay there. (We will arrange for at least one person at the motel to have a rental car.) On the other hand, we can guarantee a place in the Guest Facility for any pair willing to share a room since the available rooms can be used as singles or doubles. You can indicate your preference on the attached sheet.

Transportation

Parking at the Laboratory is no problem if you drive your own car. If you arrive at O'Hare we will arrange to have you picked up by Travelers Limousine Service. The driver will meet you at the marble drinking fountain in the baggage claim area of your airline. The published fare to Argonne (about $13.00) is reimbursable.

We will make special arrangements for some individuals to obtain rental cars for transportation between the motel and the Laboratory.

PLEASE RETURN THE ATTACHED SHEET
AS SOON AS YOUR PLANS ARE KNOWN

U of C-AUA-USAEC

ARGONNE NATIONAL LABORATORY

July 2, 1974

TO: Distribution

FROM: Wayne Cowell Wayne Cowell

SUBJECT: Comments on Argonne's Role in ODE Software

Attached is a set of notes from ODE Software Workshop I. I want to thank George Byrne for assembling and editing these notes.

We at Argonne wish to express our sincere thanks to the ODE Software Workshop participants for their time and energy during an intensive week. Time will tell whether it was a week that will strongly influence the future course of ODE software. I believe that it will, although it is too early to say what the exact nature of the impact will be.

It has not been altogether easy for the Argonne participants to formulate a response to the questions raised by the workshop. We are eager to find an approach which is realistic in terms of available resources and special interests. At the same time we recognize that an imaginative program may serve to justify requests for additional resources. The proposed activities over the next year represent a careful, perhaps conservative, but certainly deliberate approach to the creation of a systematized collection of ODE software. The operational model I proposed on Thursday morning (page 22 of the attached notes) may be taken as a long-term goal whose feasibility is to be explored during the coming year.

George Byrne and Jerry Kaganove (possibly with the assistance of a postdoc, if we can find the right one) will attempt to bring several items of software (the choice is not yet final) to operational status at Argonne. They will test and compare these codes and so develop information useful in determining the nature and contents of a systematized collection. They will periodically report to and seek the advice of the workshop participants as well as other experts who may be consulted as the work progresses. The first major occasion for feedback will be at ODE Software Workshop II to be held sometime between October and December of this year. Argonne will organize this meeting but cannot promise full financial support for the participants. In preparing for Workshop II we may request discussion papers on specific topics by specific people.

We should be careful not to tell people that Argonne is committed to the preparation of a systematized collection of ODE software. Argonne, however, is committed to pursuing the above studies and to coordinating the exchange of information so that, in about one year, a firm decision can be made whether to create ODEPACK. An affirmative response would surely imply a major effort involving the workshop participants in significant ways. Your suggestions for postdoctoral candidates and visiting faculty who are interested in ODE software will be welcome.

The openness and spirit of compromise exhibited during the workshop were extremely gratifying. Probably the honest differences of viewpoint will never be completely resolved but, to use Tom Hull's imagery, we at Argonne may be able to act as marriage brokers or, at least, to encourage the formation of convenient alliances.

WRC:dp

Distribution: Garrett Birkhoff, Harvard Univ.
James Boyle, AMD/ANL
Leonard Brown, Univ. of Illinois
George Byrne, AMD/ANL
James Cody, AMD/ANL
Robert Dickinson, AMD/ANL
Jack Dongarra, AMD/ANL
Wayne Enright, Univ. of Toronto
Brian Ford, Oxford Univ.
Fred Fritsch, Lawrence Livermore Lab.
Burt Garbow, AMD/ANL
William Gear, Univ. of Illinois
Marilyn Gordon, Sandia Lab.
Ken Hillstrom, AMD/ANL
Alan Hindmarsh, Lawrence Livermore Lab.
Thomas Hull, Univ. of Toronto
Jerry Kaganove, AMD/ANL
Peter Kemp, Cambridge Univ.
Fred Krogh, Jet Propulsion Lab.
Bengt Lindberg, Univ. of Toronto
James Lyness, AMD/ANL
James Pool, AMD/ANL
Richard Royston, AMD/ANL
David Sayers, Oxford Univ.
Lawrence Shampine, Sandia Lab.
Brian Smith, AMD/ANL
Joan Walsh, Manchester Univ.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN URBANA, ILLINOIS 61801, U.S.A.

July 11, 1974

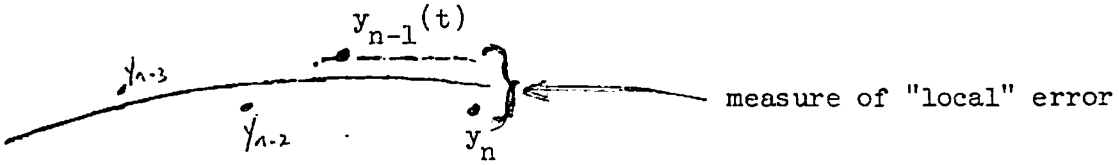
Professor T. E. Hull
Department of Computer Science
University of Toronto
Toronto, Ontario
Canada

Dear Tom:

Bengt and I discussed testing at ICASE two weeks ago, and a couple of thoughts occurred. In case he hasn't had a chance to pass them on to you, I am putting them in writing!

Your "reliability" test checks the local error defined in a sometimes unfortunate way - the difference between the next computed point, $y_{\underset{\sim}{n}}$, and the solution passing through the last computed point, $y_{n-1}(t)$. This is $\doteq L_h(y_{n-1}(t))$ for a linear multistep method when all previous values are on $y_{n-1}(t)$, where the normalization is $\alpha_0 = -1$. However, the amount of local error transmitted into the global error is $L_h(y_{n-1}(t))/\Sigma\beta_i$, and it is this quantity which a program should be controlling. (For Adams methods, $\Sigma\beta_i = 1$ and there is no difference.) Therefore, I feel that your reliability test should check something else.

The second point concerns the fact that previous values are not on $y_{n-1}(t)$. The worst example of this is a weakly stable method, say the formula $y_n = y_{n-2} + 2hy'_{n-1}$. In this, a component of the error oscillates:



The measured "local error is about twice the global error in the worst case. For example, if $y' = 2t$, $y_0 = 0$, $y_1 = 0$ (error h^2) then the computed solution is $y_{2n} = t_{2n}^2$, $y_{2n+1} = t_{2n+1}^2 + h^2$, and, although there is zero local error, this measure gives h^2.

I still feel we have a way to go to decide on appropriate test criteria.

Sincerely,

Bill

Bill Gear

BG:p

cc: W. Enright
A. Hindmarsh
F. Krogh
B. Lindberg
L. Shampine

So let's just agree to treat tolerances as "knobs" to be turned by users, with no guarantees attached.

ARGONNE
NATIONAL
LABORATORY

DATE: October 6, 1974

TO: Participants in the Argonne ODE Software Workshop and
Other Interested Parties

FROM: George Byrne, Argonne National Laboratory GB

SUBJ: ODE Odds and Ends

This is an informal report to indicate some topics which may be of interest to participants in the ODE Workshop, held at Argonne National Laboratory, June 3-6, 1974.

By now, all of the participants should have received a copy of the notes of that meeting. Although these notes are not intended for general circulation, copies can still be obtained upon request. Several of you have commented on the notes either directly or through Wayne Cowell. One point for clarification occurs on the bottom of page 22. By "willing to participate", I simply meant willing to furnish codes and to give occasional advice.

I have not gotten very far with testing or codes selection. I have been very involved in bringing up EPISODE in single and double precision. The code was originally prepared for the CDC 6600 and 7600 and the CHAT Compiler at Lawrence Livermore Laboratory. This compiler is for LRLTRAN, an extension of FORTRAN. Our goals are a code that can be carried around, i.e. literally portable. Consequently, we are embedding a single precision code within a double precision code in such a way that the conversion can be carried out quickly and easily by a variety of text edit ors or a preprocessor. For example, there are no explicit floating point or double precision constants on the right hand side of a FORTRAN statement in the code. The whole code (the double precision version with the embedded single precision version) will be a running code. We have also put in several changes as a direct consequence of the ODE Workshop, such as the root mean square norm and so on.

There have been many false starts, too. The code embedding I first put in is not compatible with CHAT (printing out asterisks is a no-no at Lawrence Livermore Laboratory), nor with practices of the Algorithms section of Comm. ACM.

Alan Hindmarsh will be visiting Argonne on October 30-31, and Novermber 1. He is working on a new version of GEAR, as well as EPISODE. He also turned out an interesting set of lecture notes, Numerical Solution of Ordinary Differential Equations: Lecture Notes, UCID-16558. These notes include errata for Bill Gear's "Numerical Initial Value Problems in Ordinary Differential Equations".

Tom Hull and his colleagues are in the process of putting out another report, which concerns testing of stiff ODE solvers.

Speaking of stiff ODE's have you seen "Stiff Differential Systems" edited by Ralph Willoughby and published by Plenum Press? It appears interesting and perhaps even provocative. Alan, Bill Gear, and Tom are authors or co-authors of three of the papers.

Larry Shampine telephoned the other day to chat about some of his numerical testing of non-stiff ODE's. Some of his tentative results were mildly surprising. Larry's invited lecture at the SIAM Fall meeting will present these results, as will a forthcoming Sandia Laboratories Technical Memorandum. I agree with Larry's remark that the work of Bill Gear and Fred Krogh have held up well.

Ralph Klopfenstein of RCA made a few comments about the ODE Workshop notes and sent along some material on his stiff ODE code. He has versions for different problem types (implicit & explicit derivatives) tridiagonal & full Jacobian).

I'm beginning to see the value of preprocessors, now that I'm almost through the modification of EPISODE. The EISPACK crew here likes to use PL/I for preprocessing.

I am now convinced that there is no such thing as "standard" FORTRAN. It also seems to take two orders of magnitude more time to clean up the little details in a portable code than I had thought it would. I am still learning more about ANL's computer system daily.

Dick Varga was in for the annual meeting of the Review Committee of the Applied Mathematics Division. He mentioned that he is working on a third order rational approximation for the problem $y'(t) = \lambda y(t)$, $y(0) = y_0$. The idea is to develop an A-stable formula for ODE's and to apply it to PDE's. It wasn't clear to me whether the integration should be thought of as a Crank-Nicholson scheme or as a Cody-Meinardus-Varga type approximation. He mentioned that he did not think the method of lines, in conjunction with a big variable order-variable step size ODE package, was appropriate for many diffusion-convection problems in the oil industry.

It appears that the panel discussion for the 1975 AIChE (American Institute of Chemical Engineers) meeting is on. That should be an excellentopportunity to meet users head on.

There is an AICA (Association Internationale pour be Calcul Analogique) Symposium on Computer Methods for Partial Differential Equations, to be held January 15-17, 1975 at Lehigh University, Bethlehem, PA 18015. Papers presented will appear in a proceedings and papers will be accepted until the time of the meeting. Several of the members of the organization are developers of PDE software. Some use the method of lines with Runge-Kutta-Merson methods of various orders and at least one uses Bill Gear's Code. For further information about the meeting write

Professor W. E. Schiesser
Computing Center
Lehigh University
Bethlehem, PA 18015

His telephone number is 215-691-7000, extension 229. (On FTS, first call the FTS Allentown operator 215-434-0181.)

It now appears that the October ODE Workshop should be held in January or February 1975.

I hope to see many of you at the JPL Preprocessor Workshop in Pasadena or the ACM Annual Meeting in San Diego.

GB:jp

U of C-AUA-USAEC

ARGONNE NATIONAL LABORATORY

November 25, 1974

Dr. Alan Hindmarsh
Lawrence Livermore Laboratory
P.O. Box 808, L-310
Livermore, CA 94550

Dear Alan:

At the ODE Software Workshop in June, there was general agreement that we had made a good start toward a consensus in several key areas. But we all recognized that a great deal of spadework must be done before we could attempt to organize a collaborative effort leading to a systematized collection of ODE software. We feel that we have made significant progress in understanding the problem of portability but, as George Byrne indicated in a recent memo, the problems are harder than we realized and we have not gone as far as we had hoped with testing or with the specifications of an ODE package/test package interface.

We intend to hold an ODE Software Workshop II but there are at least two reasons why we now feel we should postpone it until mid-May, 1975. The first is the fact that we need more time for development and testing. The second reason is that a statement by Argonne about a possible role as coordinator of a collaborative effort will be more meaningful when the dust settles from the AEC→ERDA transformation. We are confident that the mathematical software effort will be invariant under (or expanded by) this transformation but we want the crystal ball to clear somewhat before making a commitment.

I will write to you in January suggesting topics for position papers and will recommend to each participant the preparation of a statement for circulation and discussion. We at Argonne will address the question of organization as well as technical matters.

I hope you approve of this approach. Recent conversations with participants show that some are eager to meet again while others regard a meeting in say, January, as premature. Since we will make every effort to ensure that Workshop II is of benefit to you and hence to your institution, we do not expect to offer reimbursement of travel expenses.

Yours very truly,

Wayne

Wayne R. Cowell
Applied Mathematics Division

WRC:mbg

WRITE IT - DON'T SAY IT

TO Fred

RE: ODE Workshop II

I got a call from George. He says the Workshop will now be planned for mid-May 1975. Reasons: ① reluctance of participants and ② uncertainty of ANL role + funds. The latter is a much more convincing reason than the former, to me.

FROM Alan

DATE 11/22/74

a C Hindmarsh

LAWRENCE LIVERMORE LABORATORY

May 16, 1974

MEMORANDUM

TO: Participants in ODE Software Workshop at A. N. L.

FROM: A. C. Hindmarsh

SUBJECT: Thoughts on Workshop Topics

The attached is a list of my initial thoughts on many of the topics to be discussed at the Argonne ODE Workshop of May 29 - 31. I want to emphasize that these are initial and highly negotiable thoughts only, based on my own experiences in the Numerical Mathematics Group at LLL.

I should say, also, that the topics covered by this memo are those of a list included in Wayne Cowell's April 2 letter, and certainly do not include all topics we will need to cover.

As the time of the workshop is now rather close, I am sending this memo only to participants in the U. S., with extra copies going to Wayne Cowell for distribution to those missed by this mailing.

See you soon.

a C Hindmarsh

Alan C. Hindmarsh, L-310
Numerical Mathematics Group

ACH:aas

THOUGHTS ON THE ODE SOFTWARE WORKSHOP

A. C. Hindmarsh

May 1974

[illegible] of Terminology package [illegible]

1. Problem domain

To a large extent, I feel the domain of problems for which the package is intended should be determined by the capabilities of algorithms and (especially) software which exists now. What exists now roughly reflects the needs of the users of ODE software.

(a) For the first edition, I feel only the initial value problem should be covered, as this includes the bulk of existing codes and of users' problems, and simplifies the task of standardization greatly.

(b) Both stiff and nonstiff problems should be allowed for, by the package as a whole.

(c) First-order systems of ODE's only should be considered at first. This simplifies the standardization problem, and the relative advantage of methods for higher-order equations (vs. 1st-order systems) has not been demonstrated, to my knowledge.

(d) If the problem is written $\dot{y} = f(y,t)$, $y(t_0) = y_0$, both expensive and inexpensive f should be allowed for. That is, there should be codes which go to considerable length to minimize the number of evaluations of f (or of $\partial f/\partial y$ if needed). And there might (should?) also be codes that have minimal overhead costs, at the expense of function evaluations, to be used when f is inexpensive.

(e) If only first-order systems are to be allowed, then the use of higher order derivatives should be discouraged. If methods requiring $\ddot{y}$ are included, they should allow for $\ddot{y} = f_y f + f_t$ to be supplied by the user (for the case when this is easy to code), *and also* for internal generation of $\ddot{y}$ (for the case when it is not).

(f) *Specialized versions* should be allowed if the expected usage of such is sizable, if the version exists now (excluding reformatting), and if the additional special software introduced (e. g. sparse or band matrix solvers) is made to meet the coding standards. From the point of view of user need, I feel that a stiff system code that uses the Jacobian in banded form should be included as a minimum in this respect. Possibly a differential-algebraic system solver should also.

2. Code sources

(a) For the initial value problem for first-order systems, there are many ODE solvers in use, of which many should be considered as *possible candidates* for the proposed package. The subject of ODE solution is sufficiently rich and complex that no small number of solvers (like 5) can be found which adequately covers the field. Even for narrow subclasses of problems, there is frequently no clearly *best* choice of solver, and often not even a best choice of *method*. Where there is a best method, there may be a multiplicity of *algorithms* which implement the method, and a multiplicity of *codes* which implement each algorithm. Of all the codes based on a given method, there is rarely one with a clear superiority over the others, all things considered. In view of this, I am inclined to accept as candidates for this package a large number of solvers which together entail considerable overlap in domain of applicability. This acceptance is conditional on the documentation of the virtues of the codes, both in an absolute sense, and relative to other codes.

Listed below are ODE solvers that have been used by me or others at LLL, with sufficient success to be considered as possible candidates.

(1) DIFSUB. Written by Gear, has 3 method options (2 for stiff problems), widely used.

(2) GEAR. Written by Hindmarsh, based on Gear's DIFSUB, has 8 method options (3 for stiff problems), widely used at and outside LLL.

(3) DE/STEP. Written by Shampine and Gordon, uses Adams methods, for non-stiff problems.

(4) EPISODE. Written by Byrne and Hindmarsh, same method options as GEAR, not yet released for general use.

(5) RKF. Fehlberg's Runge-Kutta method, in several variant forms.

(6) GEARB. Version of GEAR for banded Jacobian.

(7) GEARBI. Version of GEAR for block-iterative solution of Jacobian.

(b) At present I know of no unimplemented algorithms for which there is great need. However, there are variants of existing codes that should be, and can easily be, constructed for possible inclusion. For example, I intend to write a differential-algebraic system solver which is a variant of GEAR. Also, of course, minor changes, in the nature of "fine tuning," should be done to any candidate code.

3. Programming standards

(a) To the greatest extent consistent with what is reasonable and practical, the programming standards imposed on EISPACK and FUNPACK should be followed. At least there should be some uniform standard for card format, variable names, commenting, etc., and the use of an appropriate subset of Fortran, which the codes included should be made to follow. Wherever this entails a significant loss of efficiency, exceptions should be made. For many of the routines, it will be necessary to include different versions for different machines, as is done in FUNPACK. But for most of the routines, I would think that a single version could contain a few comment cards giving the changes necessary for other versions, as is done in EISPACK.

(b) It will be impossible to make the calling sequences identical for all ODE solvers in the package. Solvers intended for different problem types will necessarily have different calling sequences. Moreover, the various solvers will have various substructures of auxiliary routines, which cannot be made standard throught the package. In spite of these differences, I propose that each solver conform to the following loose structure rules:

(1) Variables which have the same roles in more than one solver shall be given the same names. Specifically, I suggest the use of

T for the independent variable,

TØ for the initial value of T,

TOUT for the next output value of T,

Y for the dependent variables,

YØ for the value of Y at TØ.

N for the number of equations in the ODE system,

EPS for the error tolerance parameter,

H for the step size in T.

A set of names such as this should be strictly adhered to in calling sequences seen by the user, but perhaps less strictly elsewhere.

(2) The interface between the user and the solver shall be by way of a "driver" subroutine whose calling sequence shall include TØ, YØ, N, and TOUT, plus other parameters that pertain to error control and the specification of the particular method. It should also contain an input integer flag which indicates whether the call is the first one for the problem or not, and an output integer flag (perhaps the same variable) which indicates the success or failure of the solver. This routine should perform the integration of the ODE from TØ to TOUT and return the values of the vector Y at TOUT, unless a failure of some kind occurred sooner. It shall make repeated calls to lower level routines in order to achieve this. I feel that the name of the function f should not be included in the calling sequence, as this is virtually always unnecessary and costly from the user's point of view. (It also causes a portability problem.)

(3) Wherever a lower level routine is called for to perform a calculation common to more than one solver, the same routine should be used throughout the package. Thus, for example, the same general linear system solver should be used by all ODE solvers requiring one.

(c) The use of Common should be allowed between the highest level routine (the driver) and the others, but not between the user and the driver. My experience is that the use of Common can reduce run times significantly (up to 20%) by reducing indirect addressing. However, the user-driver interface will not normally be crossed frequently enough to justify Common there, considering the added burden this would place on the user.

(d) The use of modular structure should be highly encouraged. Among the items that should be put into separate modules are the following:

(1) Solution of linear equation systems,

(2) Input/output,

(3) Interpolation of output values,

(4) Setting arrays of coefficients (because this is so machine-dependent).

Entry points should be avoided for the sake of portability.

4. Testing

(a) There are numerous sources of problems in the literature which can serve as test problems. I have a number of problems to add to these. Test problems should be such that an analytic (or otherwise exact) solution can be readily computed for test purposes. Even with this restriction, it is not hard to find problems which, together, cover a broad range of size, smoothness, stiffness, "realisticness", and accuracy needs. As all of these aspects are important, a good test problem collection for the ODE package will attempt to cover a 5-dimensional (at least) problem space. It may well consist of ≥ 100 problems. It should be an easy matter to systematize this collection such that, for each problem, routines for the evaluation of f(y, t) (as called by the solver) and of the true y(t) are constructed with standardized calling sequences. [illegible]

Regarding the methodology of testing, the published work of Hull et al., Krogh, and others should be used as guidelines. I have only a few comments to add:

(1) It will probably be impossible, and I think unwise, to settle on the use of error per step or error per unit step -- i.e. one or the other throughout the package. Unless and until a convincing clear-cut resolution of this controversy is achieved, ODE solvers should allow for either choice, at the user's option. Moreover, in the testing of any solver on any problem

for which there is any doubt as to the better choice, both versions should be run, and the better results selected to represent the performance of that solver on that problem.

(2) The calculation of a "true" solution by another numerical method should be avoided, as this opens a certain credibility gap. There are enough problems with closed-form exact solutions to make such a calculation unnecessary.

(3) For testing purposes it may be necessary to make programming changes in the solvers -- e.g. the addition of statistics collection, or the inclusion of f in calling sequences. The distinction between what is most convenient for testing and what is most convenient for the intended users must be kept in mind.

(4) Opportunity for final fine tuning of solvers on test set on all machines,

(b) Independent funded test centers should be established for the sizable job of constructing of a test problem set, setting up testing rules and software, and performing the tests on the tentative candidate solvers. Final results must of course then be documented and published. The centers must collectively represent the various machines for which the package is intended. The EISPACK experience should be helpful here.

Performance profiles published [illegible] [illegible] info [illegible] [illegible] [illegible].

5. Funding

(a) For the first edition of the package, I would hope that funding for code development will not be needed. Once a set of standards and a list of tentative candidate solvers is established, I would expect that the originators of those solvers will be willing to reformat and restructure them as necessary to meet the standards. However, there needs to be an independent group which will check the solvers for adherence to the standards, and make judgements as to exceptions. This could wait until after some of the testing, and after the final selection of solvers has been made.

(b) The need for funding of testing will be a major one. It could easily involve four man-years, perhaps more, and considerable machine time. (Remember, some of the test problems will be large and complex, in order to represent the realities of expected usage.)

(c) Site selection for test centers could be done much as was done for EISPACK, but with the increased manpower need kept in mind. A bare minimum of five sites would be needed, I would think, in view of the range of hardware in use. Twice that many would be desirable, so as to get independent verification of results.

6. Name

ODEPACK?

ODE Software Workshop

Argonne National Laboratory

June 3-7, 1974
Bldg. 221, Room A-216
Conf Rm: C-201

Monday, June 3 - 9:00 AM

Introductory Remarks - Cowell

Agenda Discussion

Topic I: The capabilities of an ODE software package
Statements by Gear, Hindmarsh, Hull, Krogh, Shampine, Walsh
Discussion - Cowell, Chairperson

Lunch

Agenda Review

Continuation of Topic I as required

Topic II: The measurement and control of error
Discussion - Lyness, Chairperson

Tuesday, June 4 - 9:00 AM

Agenda Review

Topic III: Software to provide the capabilities in Topic I
Statements by Walsh, Shampine, Krogh, Hull, Hindmarsh, Gear
Discussion - Pool, Chairperson

Lunch

Agenda Review

Continuation of Topic III as required

Topic IV: Coding conventions, standards, and structure
Discussion - Cody, Chairperson

Picnic in Argonne Park

Wednesday, June 5 - 9:00 AM

Agenda Review

Topic V: Testing ODE software
Statements by Krogh, Shampine, Walsh, Gear, Hindmarsh, Hull
Discussion - Smith, Chairperson

Lunch

Agenda Review

Continuation of Topic V as required

Topic VI: Maintenance of ODE software
Discussion - Boyle, Chairperson

Dinner at Dakory - J Boyle - 7:0;

Thursday, June 6 - 9:00 AM

Agenda Review

Topic VII: Collaboration to produce a systematized collection
Statements by Cowell, Pool
Discussion - Ford, Chairperson

Lunch

Agenda Review

Continuation of Topic VII as required

Further topics and continuations as determined in agenda reviews

Cocktails and dinner at ~~Argonne Staff Club~~ [illegible]

Friday, June 7 - 9:00 AM

Further topics and continuations as determined in agenda reviews