Physica Scripta. Vol. T16, 133–142, 1987.

Alpha Particle Effects as a Test Domain for PAP, a *P*lasma *A*pprentice *P*rogram

Harry E. Mynick

Princeton Plasma Physics Laboratory, Princeton, NJ 08544, USA

Received June 25, 1986; accepted June 30, 1986

Abstract

A new type of computational tool under development, employing techniques of symbolic computation and artificial intelligence to automate as far as possible the research activities of a human plasma theorist, is described. Its present and potential uses are illustrated using the area of the theory of alpha particle effects in fusion plasmas as a sample domain.

1. Introduction

Alpha particles may be expected to affect, and be affected by, fusion plasmas according to a coupled system of mechanisms involving the heating, equilibrium, transport, and stability of the plasma. While much of the specifics of this interaction remains unexplored, the process of exploration will largely involve bringing to bear theory and concepts developed in other plasma contexts.

This combination of an unexplored problem domain, being approached with a collection of related perspectives and results from a more thoroughly studied area, makes the alpha–plasma interaction an appropriate test domain for PAP, an "apprentice" program for plasma theory [1]. Presently at a fairly early stage of implementation, the general goal of the PAP project is to employ techniques of symbolic computation and artificial intelligence to do as much as possible of the work of a human plasma theorist. The term "apprentice" is intended to connote, first of all, that the performance level of the system is not competetive with that of a human theorist (at least for the forseeable future), and also that the system is intended to interact with the human user, informing him of its status and asking him for further guidance when it runs into difficulty, and to be able to learn new facts and methods. This paper will describe the structure of PAP, its present capabilities and some of those which seem well within reach, and illustrate how such a facility could be used by the plasma community in furthering research in a new problem domain such as alpha–plasma phenomena.

The notion of automating the mental activities of a human theorist generates (quite understandably) a degree of skepticism in the minds of human theorists, who know how simple-minded computers are. Two related questions may be asked, characterizing the nature of this skepticism: (a) Given present-day AI capabilities, can such a facility be constructed which is actually *useful*? (b) Can such a facility actually generate any theoretical results which are *new*, without a simple regurgitation or trivial rearrangement of all the necessary information given to it by a human theorist, who has effectively already found the result beforehand? In the remainder of this paper, support will be given for the claim that the answers to these questions are "yes" and "probably," respectively.

Clearly, if the answer to question (b) were "yes," the answer to (a) would be "yes" as well. Additionally, however, it is important to remember that much of the mental effort of a human theorist is devoted to tasks which, though absolutely essential to making theoretical progress, are rather mundane in nature, having a structure which is fairly clear, and therefore amenable to machine imitation. Thus, even were the apprentice unable to do any serious physics calculations itself, it could still be of considerable use performing these "lower-level" functions (examples of which will be given shortly).

In support of an affirmative answer to question (b), later in this paper we will describe in some detail a transport calculation already carried out by PAP in an almost entirely automated fashion, in which some nontrivial (though previously-known) transport results are found.

2. PAP Structure

A schematic depiction of the structure of PAP is shown in Fig. 1. Adopting as a top goal the automation of nontrivial plasma calculations [achieved by the "high-level calculator/problem-solver" (CALC) in Fig. 1], the lower-level facilities shown there were automatically required. That is, each of these facilities is called on by the high-level control structure in carrying out a calculation, as will be seen. However, as indicated by the arrows leading from the "human user" box in Fig. 1, each of these facilities may be accessed directly by the user, and these can be very useful in their own right.

PAP rests on two cornerstones (shown in Fig. 1), on MACSYMA, and on the PAP "knowledge base" (KB). In addition to giving PAP a serious algebraic capability (an obvious necessity for doing physics), MACSYMA in addition contains a full programming language. The MACYSMA language has a structure quite similar to that of LISP (an AI standard), in terms of which it is written, and it is easy to call on routines written in one language from the other. For these reasons, PAP is written mostly in MACSYMA, with a few routines written in LISP.

The other PAP cornerstone is its KB. This contains PAP's "understanding" of plasma physics. This includes not only the important physical quantities, equations, and results of plasma physics, but also information specifying the relations among them, and how they are used. In conjunction with the "low-level calculator/evaluator" (NUMVAL) indicated in Fig. 1 (to be described shortly), a mature version of the KB would serve a powerful pedagogic and archival function – a

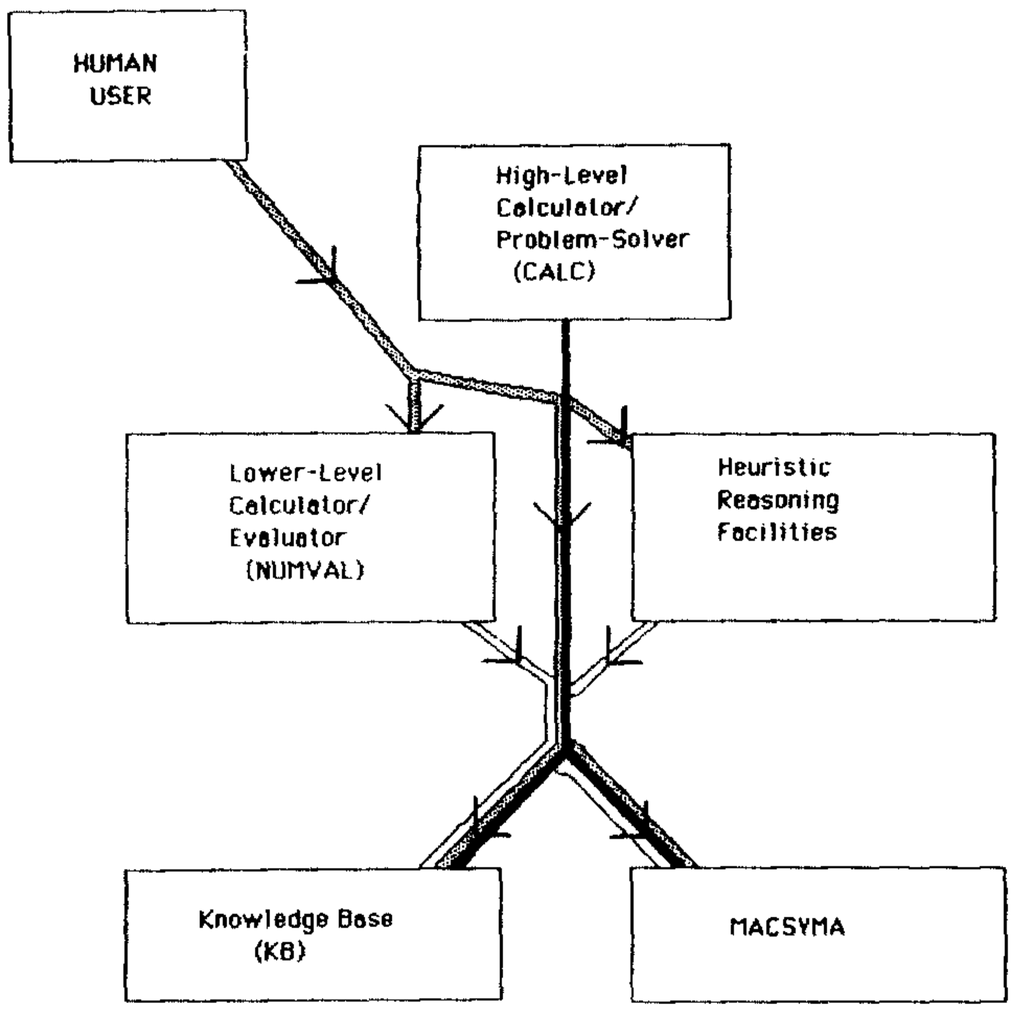

Fig. 1. Schematic Representation of the structure of PAP.

single, unified, easily updated, quickly-accessed body of plasma knowledge, which plasma physicists presently must synthesize from many sources.

At the intermediate level of Fig. 1, calling on the KB and MACSYMA foundations, are NUMVAL, and a set of routines enabling PAP to do some of the kinds of heuristic, qualitative reasoning which is important to the way in which people understand and communicate physical ideas, and attempt to solve physics problems. The action of these facilities will be demonstrated, first through some sample interactions of the human user with PAP, and later in the description of the transport calculation, where CALC calls on them automatically.

2.1. *Use of the KB and NUMVAL*

In addition to direct interrogation of the information in the KB, a simple but quite useful PAP facility drawing heavily on the KB is the function NUMVAL. In getting a feel for the size or importance of physical effects, a theorist is often called upon to draw out a set of relevant formulae, and plug in typical values for the variables occuring there, or sometimes to see the scaling of these expressions with important parameters. For example, suppose one wants to know how large the alpha banana width to system size is, or what a tolerable ripple amplitude is in some contemplated machine design. If the expressions to be evaluated are short, and the number of times they are to be evaluated few, one typically pulls out his plasma formulary and a calculator, and spends time guiding the calculator through a procedure which is tedious, time consuming and error prone. If the calculation is more involved (the above-mentioned example of tolerable ripple levels already qualifies), one often writes a modest computer program, whose structure is clear, but which takes time to assemble. NUMVAL uses the KB as an on-line formulary, and given an expression to evaluate, effectively writes the program the theorist normally writes by hand, accessing the appropriate formulae in the KB, and performing the evaluation. The following briefly illustrates some uses of the KB and NUMVAL:

(C4) get(cit, 'params);

(D4) [RR = 1.225d+0, A = 0.454d+0, KPELIP = 1.8d+0, BB = 10.4d+0, QA = 2.5d+0, DENS = 6.5B20, TTEVI = 7000, TTEVE = 7000, N = 20, DLA = 1.5D−02]

(C5) boltzeqa;

(D5) $$\frac{dF}{dTH}\,THDT + \frac{dF}{dR}\,RDT = \frac{(\frac{d}{dY}(AA\,\frac{dF}{dY}))\,NUR}{AAP}$$

(C6) why('boltzeqa);

(D6) (BOUNCEAVG(BOLTZEQ), NEGLECT(%, [1]))

(C7) englishdef(dd);

(D7) [diffusion coeff]

(C8) instances(dd)

(D8) [DIM = 1d, DDCL, DIM = 2d, DD2D, DIM = 3d, DD3D]

(C9) instances(ddbd);

(D9) [NUREGIME = [NUN = 0], DDSTOCH, NUREGIME = [NUN < FIDT N], DDBD1, NUREGIME = [FIDT N < NUN, NUN < WB], DDBDM1, NUREGIME = [WB < NUN], DDRP]

(C10) englishdef(ddrp);

(D10) [diffusion coeff, DIM = 3d, MECH = banana drift, NUREGIME = [WB < NUN]]

(C11) get(ddrp, 'givenby);
Evaluation took 0.00 seconds (0.00 elapsed).

(D11) $$\frac{DDBP\ DL^2\ N\ Q\ SGRP}{EPT^2\ SGBP}$$

(C13) numval([ddstoch,ddbd1,ddbn],[per(cit,species = alf), q = 2,r = a/2]);
Evaluation took 50.06 seconds (56.03 elapsed).

(D13) $$[1.6635B7\ DL^2\ SGSTOCH,\ \frac{2.2709B13\ DL^2\ SGBD1}{FVO^2\ TTEV^{1.5d+0}},\ \frac{4.117B7}{TTEV^{1.5d+0}}]$$

(C15) numval(d13,[ttev = egyevalf,fv0 = 1, dl = .01]);
Evaluation took 2.31 seconds (2.53 elapsed).

(D15) [1.6635B3 SGSTOCH, 3.4681B−1 SGBD1, 6.2875B−3]

(C16) tauconf(dd,a) := a^2/dd;

(D16) $$TAUCONF(DD, A) := \frac{A^2}{DD}$$

(C21) opmap('tauconf(d15,.454),1,1);
Evaluation took 0.43 seconds (0.43 elapsed).

(D21) $$[\frac{1.2391B-4}{SGSTOCH},\ \frac{5.9432B-1}{SGBD1},\ 3.2782B1]$$

(C22) numval([nu,nuslw],[per(cit,species = alf),ttev = egyevalf]);
Evaluation took 20.97 seconds (21.78 elapsed.

(D22) [1.6875B−1, 3.2606B1]

The numbers Ci, Di appearing on the left are the usual MACSYMA line numbering. The MACSYMA command GET(v, p) retrieves property p of variable v. Some of the types of information in the KB are shown by use of this command. Thus, in C4, the values of typical parameters for a given device of interest (in this case, for CIT) are recalled. In C5–D6, a different type of information is requested. In C5, D5, a particular variant ("boltzeqa") of the Boltzmann equation, the one used in the transport calculation discussed later, is requested and displayed. In C6 the KB is asked for a description of the origin of this equation. The response in D6 is a sequence of MACSYMA instructions whose which represents a derivation of bolteqa from the full Boltzmann equation. It may be read in English as "First, bounce average the Boltzmann equation. In the result of this, neglect the first term. The result is boltzeqa." In a fashion illustrated by this simple example, a novice in some area of plasma theory could, by stepping through the "derivation networks" which tie together the important equations of that area in a mature version of the KB, teach himself the structure of the theory of that area.

In C7–D7 and C10–D10, the user requests a definition of two symbols whose meaning he does not know, using the command ENGLISHDEF. The symbol "dd" (double letters are conventionally used when the corresponding mathematical symbol is upper case) stands for the general term "diffusion coefficient." Having found this general term, in C8, D8, the user asks what special instances of this term are known to PAP. Within the KB, dd represents the top node of an "inheritance network", a conceptual hierarchy, where the lower-level nodes represent more concrete versions of the higher-level ones. This is illustrated in Fig. 2 for the particular

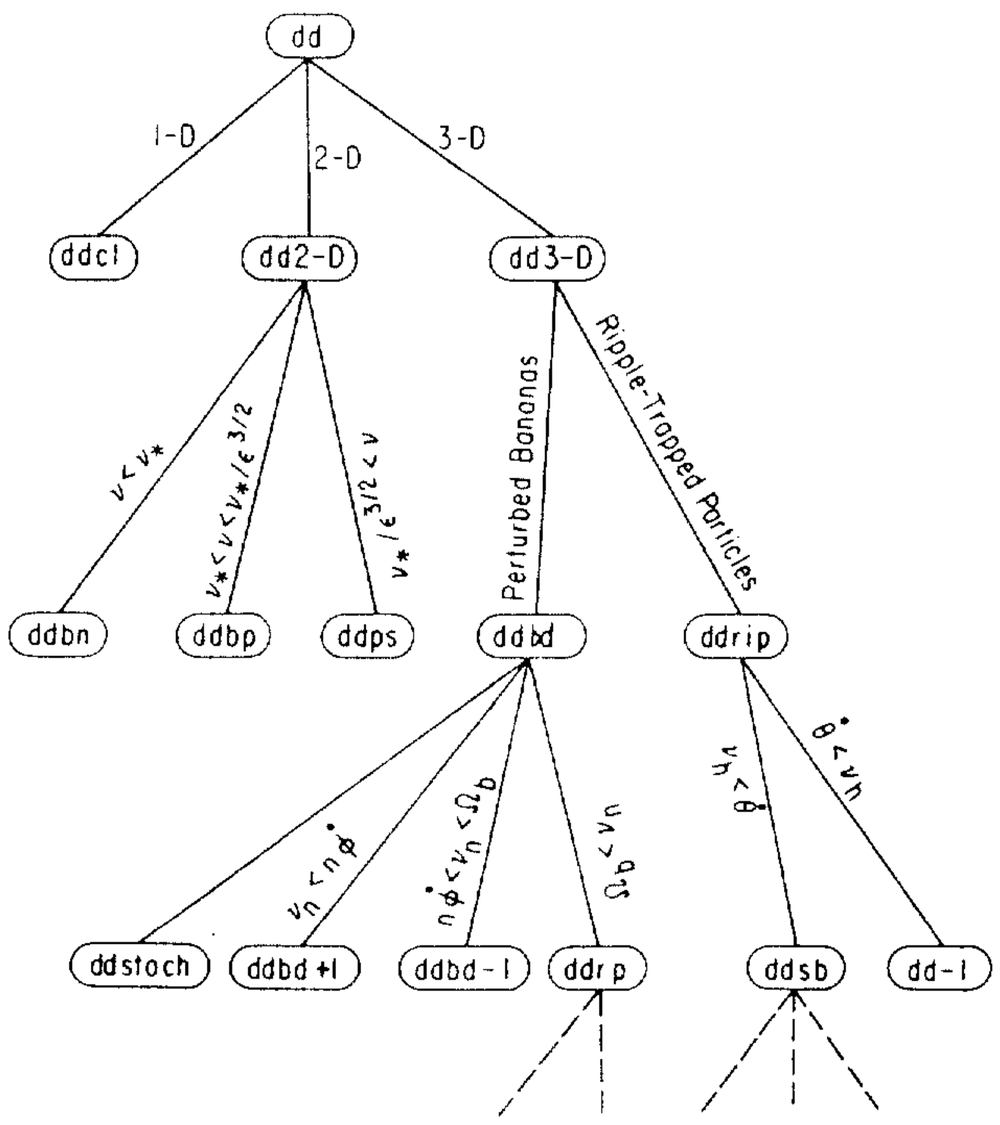

Fig. 2. Depiction of an "inheritance network" for plasma transport results. Deeper nodes are more specific instances of the higher nodes. The labels on the connecting arcs describe the nature of the particular specialization.

concept of diffusion coefficients. Such networks are very useful organizing structures in the KB. The labels on the arcs of the network in Fig. 2 are the "qualifiers," indicating the specializing assumption(s) made to move from the more general to the more specific concept. The form of the response D8, 9 to the INSTANCES requests is [qualifier1,symbol1, qualifier2,symbol2, ...], giving all the special cases symbol(i) of the "parent" concept, preceded by thge qualifier(i) connecting the parent to the symbol. Thus, it gives a "horizontal slice" of the tree shown in Fig. 2. The response D10 to the ENGLISHDEF request gives a "vertical slice", listing the path of qualifiers connecting the specified variable (in this case, the ripple-plateau coefficient D_{rp}) to the top-level concept. Paraphrasing D10, "D_{rp} is that diffusion coefficient arising from the banana-drift branch of the several 3-D transport mechanisms which exist, in the collisionality regime $\omega_b < \nu_n$." In C11, the user asks for the algebraic form of D_{rp}. Interrogating the KB in this fashion, a theorist can aquaint himself with the known transport mechanisms, what the assumptions are for their validity, and what their algebraic form is.

In C13, the user requests a numerical evaluation of three diffusion coefficients he considers of possible relevance to alpha transport (those of the stochastic-regime D_{stoch} [2], the "precession-dominant" banana-drift regime D_{bd1} [3], and the axisymmetric banana regime D_{bn} [4]), using the function call NUMVAL (expr,parmvals). The first argument (expr) is the object whose numerical evaluation is sought, and may be a variable, any algebraic expression, or a list of such expressions. The second argument (parmvals) is a list specifying the values of parameters which are to be assumed in the evaluation. Thus, one may paraphrase C13 as "numerically evaluate the list [D_{stoch},D_{bd1},D_{bn}], assuming $q = 2, r = a/2$, and otherwise for CIT parameters, for alphas." The Latin work "per" is used for the moment because "for" is a special word to the MACSYMA reader. The items inside the "per"-construct in C13 are used to set the "computational context" within which the evaluation is carried out [1].

The evaluation process NUMVAL uses may be clarified by the following trace of NUMVAL applied to the electron cyclotron frequency wce in TFTR:

(C4) numval(wce,[per(tftr)]);
1 Enter NUMVAL1 [[WCE, [ ]]]
2 Enter NUMVAL1 [[$\frac{\text{BB EC}}{\text{MM}}$, [SPECIES = ELEC]]]
3 Enter NUMVAL1 [[BB, [SPECIES = ELEC]]]
4 Enter NUMVAL1 [[5, [SPECIES = ELEC]]]
4 Exit NUMVAL1 [5, [ ]]
[Heap grown by CONS: 747392 → 772992, free = 681471]
3 Exit NUMVAL1 [5, [ ]]
3 Enter NUMVAL1 [[EC, [SPECIES = ELEC]]]
4 Enter NUMVAL1 [[1.6B-19, [ ]]]
4 Exit NUMVAL1 [1.6B-19, [ ]]
3 Exit NUMVAL1 [1.6B-19, [ ]]
3 Enter NUMVAL1 [[MM, [SPECIES = ELEC]]]
4 Enter NUMVAL1 [[9.11B-31, [ ]]]
4 Exit NUMVAL1 [9.11B-31, [ ]]
3 Exit NUMVAL1 [9.11B-31, [ ]]
2 Exit NUMVAL1 [8.7815B11, [SPECIES = ELEC]]
1 Exit NUMVAL1 [8.7815B11, [SPECIES = ELEC]]
(D4) 8.7815B11

As shown here, the evaluation process is basically handled recursively by function calls NUMVAL1([expr,cmpcntxt]), where expr has the same meaning as for NUMVAL, and cmpcntxt is the computational context. At the level-1 call, to NUMVAL1([wce, . .]), PAP looks to see if it has an approximate formula or value, first in parmvals, and if not, in the KB. In the KB, it finds nothing under the "givenby" property of wce, and so examines wce's inheritance-network connections to see if wce is a special case of some symbol for which a formula is given. There, it finds the symbol wc, given-by formula eB/M, and so, at level 2, embarks on evaluating this, with cmpcntxt expanded to remember that the evaluation is to be for electrons. To evaluate this compound expression, NUMVAL first recursively evaluates each of the symbols in the expression, and then substitutes the resultant values into the expression. During the evaluation of each symbol, an attempt is made to specialize the symbol as far as cmpcntxt indicates (cf., e.g., the evaluations of ec and mm).

When none of the specified information allows PAP to numerically evaluate a symbol, it returns the symbol itself. This provides a convenient way to determine the parametric scaling of some expression with important parameters. This illustrated in D13, where, for example, the magnetic ripple-strength δ (dl) has been left unspecified, and one sees the quadratic scaling of the two ripple-transport coefficients with δ. If desired, as in C15, D15, these parameters may then be given values, and the expression further evaluated, with only a small additional cost in time. In C16, a simple MACSYMA function is defined for the purpose of converting the values of the diffusion coefficients in D15 into confinement times, and in C21, this function is mapped onto the three D's in that list. Because from D21 $D_{\text{stoch}} \gg D_{\text{bdl}}$ for alphas in CIT, it is D_{stoch} of the banana-drift "branch" of transport coefficients which will apply (this may also be checked, using other NUMVAL calls, on the qualifiers giving the collisionality boundaries previously seen using INSTANCES and ENGLISHDEF), and one sees that the assumed value $\delta = 0.01$ yields an alpha confinement time which is much too small. A firmer meaning is given to the term "too small" in C22, D22, where the slowing-down frequency (nuslw $= 1/\tau_{\text{slw}}$) is evaluated. (The 90-degree scattering frequency nu ($= \nu$) is also evaluated, just for comparison.) Using the numbers in D22 and D21, and imposing the usual condition that $\tau_{\text{conf}} > \tau_{\text{slw}}$ for acceptable alpha-confinement, one is lead to an estimate of a tolerable ripple size $\delta < 0.06\%$ halfway out in CIT.

Thus, using less than two minutes of computer time, having had to issue only a few one-line commands in the sequence C13–D22, one has been able to answer a question of some interest to alpha transport. The result is guaranteed to be algebraically and numerically correct, the only possibilities for error being an incorrect formula being entered by the person/people who wrote the KB, or a misinterpretation of the meaning of the numbers being generated on the part of the user. It is easy to ask PAP to "show its work" after a call to NUMVAL. The time and error-saving utility of the facility would be still greater were one interested in several different sets of CIT parameters, or several different machines.

3. Qualitative and heuristic reasoning

When analyzing and talking about physics problems, physicists frequently engage in various types of "heuristic" or "qualitative" reasoning. These heuristic arguments can communicate in a simple manner what the essential features are entering into a given physical phenomenon or mathematical result, providing guidance in the more rigorous and detailed (and often more obscure) formal derivations which ultimately provide the solid basis of a physical theory. Giving PAP the ability to engage in these kinds of heuristic reasoning is important, in order for it to be able to derive the same benefits for problem solving as its human practitioners do, and to be able to "understand" in an efficient manner arguments communicated to it by a human user. In this section we discuss three facilities developed to give PAP this kind of capability. These will also be helpful in understanding the subsequent discussion of the automated transport calculation.

3.1. *HEURMAP*

Often, in a preliminary analysis of the sorts of solutions one may expect from a given differential equation, one maps the equation into a simpler one by, for example, replacing a complicated function by a simpler one which reflects the rough behavior of the function, or by replacing the derivatives appearing in the original equation by characteristic scale lengths over which the functions on which the derivatives act vary. HEURMAP (for "HEURistic MAPping") is a function which gives PAP this kind of ability. Here, we illustrate HEURMAP by some user-directed examples:

(C3) ballooneq;

(D3) $$\frac{\mathrm{d}}{\mathrm{dS}}\left(\frac{K^2\,\frac{\mathrm{dU}}{\mathrm{dS}}}{\mathrm{BB}}\right) + \frac{2\,K^2\,KP\,\frac{\mathrm{dP}}{\mathrm{dR}}\,\mathrm{V}}{\mathrm{BB}^3} = 0$$

(C4) [any,*s*,1/(*q***rr*),any,*r*,1/*a*];

(D4) $$[\mathrm{ANY},\ S,\ \frac{1}{Q\ RR},\ \mathrm{ANY},\ R,\ \frac{1}{A}]$$

(C5) heurmap(d3,d4);

(D5) $$\frac{6\ \mathrm{K}^2\ \mathrm{V}}{\mathrm{BB}\ \mathrm{Q}^2\ \mathrm{RR}^2} + \frac{2K^2\ \mathrm{KP\ P\ V}}{\mathrm{A\ BB}^3} = 0$$

(C6) subst(p=beta*bb^2,d5);

(D6) $$\frac{6\ \mathrm{K}^2\ \mathrm{V}}{\mathrm{BB}\ \mathrm{Q}^2\ \mathrm{RR}^2} + \frac{2\ \mathrm{BETA}\ \mathrm{K}^2\ \mathrm{KP\ V}}{\mathrm{A\ BB}} = 0$$

(C7) solve(d6,beta);

(D7) $$[\mathrm{BETA} = \frac{3\ \mathrm{A}}{\mathrm{KP}\ \mathrm{Q}^2\ \mathrm{RR}^2}]$$

(C8) diffeq1;

(D8) $$\frac{\mathrm{dN}}{\mathrm{dT}} = \frac{\mathrm{d}}{\mathrm{dR}}\left(\mathrm{DD}\,\frac{\mathrm{dN}}{\mathrm{dR}}\right)$$

(C9) [any,t,1/tau,any,r,1/a];

(D9) $$[\mathrm{ANY},\ \mathrm{T},\ \frac{1}{\mathrm{TAU}},\ \mathrm{ANY},\ \mathrm{R},\ \frac{1}{\mathrm{A}}]$$

(C10) heurmap(d8,d9);

(D10) $$\frac{\mathrm{N}}{\mathrm{TAU}} = \frac{1\ \mathrm{DD\ N}}{\mathrm{A}^2}$$

(C11) solve(d10,tau);

(D11) $$[\mathrm{TAU} = \frac{\mathrm{A}^2}{3\ \mathrm{DD}}]$$

In C3,D3 above, we introduce a slightly simplified version of the ballooning equation [5]. The first term represents field line bending, and the second represents the pressure(p)–curvature(kp) destabilizing term. The perpendicular displacement is denoted by v. In D4, the "replacement list" rplst is introduced, which determines the mapping to be made from the derivatives present. rplst runs in segments of three elements [. . . , vs,x,k_x, . .], where vs denotes which variables are affected, x denotes the variable of differentiation, and k_x gives the inverse scale length into which the x-derivative maps. vs may be either a single variable, a list of variables, or (as in D4) the wildcard variable "any". Thus, the first triplet in D4 means $\partial(\text{any})/\partial s \rightarrow (\text{any})/qR$, i.e., the usual mapping one makes in ballooning mode theory that the scale for variation along a field line is the connection length qR. By "any" here is meant any variable not known to be independent of s from other information in the KB. Additionally, the KB can provide the information for these heuristic mappings, overridden by the explicit input rplst of D4.

In C5 HEURMAP is applied to the ballooning equation, resulting in D5. The numerical factors are an indication of the number of separate terms in the differand having nonzero derivative. Replacing p in favour of beta (β) in D6, in D7 we solve for β, arriving at (up to numerical factors) the expression for the critical β for ballooning stability.

A similar heuristic analysis is made for the diffusion equation given in D8, using the rplst of D9. Applying HEURMAP in C10 and solving for tau, in D11 one finds the usual expression for the characteristic diffusion time.

3.2. *The KITO-calculus*

Much of the structure of a mathematical problem of procedure may be outlined without having to do any of the actual algebra. For example, when setting up a problem, it is common to check that the number of governing equations is equal to the number of independent variables. Another algebra-free rule of this type is exemplified by the logical link ". . . this gives us the action J as a function of p_x, p_y, and energy E, which we then invert to give the Hamiltonian $E(J, p_x, p_y)$". The downward-recursing stage of the evaluation procedure described above for NUMVAL is a simple case of an algebra-free information manipulation of this type. During that stage, what is done is to gather the relevant equations, linked by which equation is used to determine which variable into a "solution network", or "solution net", which then serves as a solution plan used during the upward-recursing stage, in which the actual algebra is done. The ability to perform this kind of abstract information bookkeeping and manipulation is a facility now partially implemented in PAP, referred to as the "KITO" (for "Known-In-Terms-Of")-calculus. The solution networks constructed by the KITO-calculus function KITOGEN are an important part of giving PAP a serious capacity for problem-solving [6, 7]. For most plasma problems, the simple recursive evaluation procedure used by NUMVAL is inadequate. As a simple example, consider the simple harmonic oscillator (SHO) equations

$$\partial x/\partial t = p/m, \quad \partial p/\partial t = -k(x - x_0). \tag{1}$$

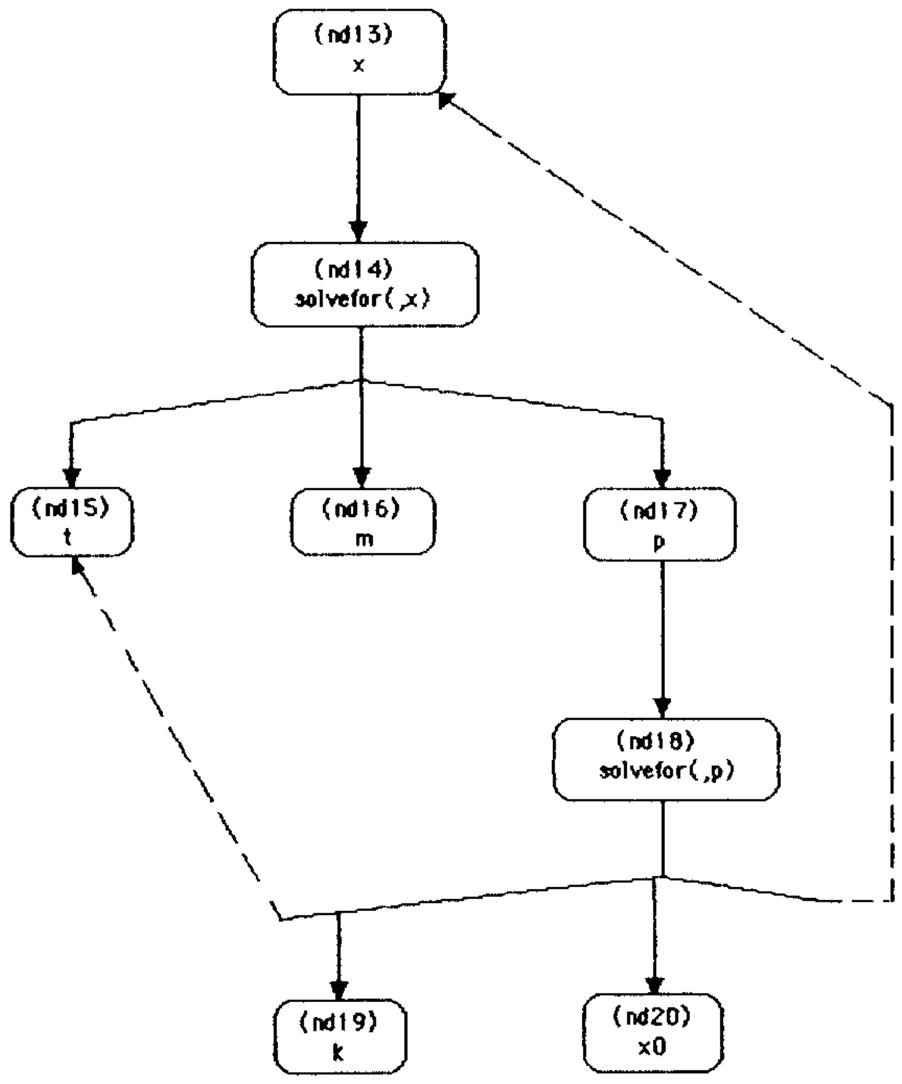

Fig. 3. Structure of a simple "solution network" for the simple-harmonic oscillator system of eqs. (1). The main goal [at "root" node (nd13)], the calculation of x, appears at the top of the network, and subgoals arising from this main task appear below. The upward directed, dashed lines lead to subgoals already established at the time of the creation of the parent goal from which the lines emanate.

The solution network corresponding to this system is shown in Fig. 3. NUMVAL applied to this system would go into an infinite loop, because the "root node" x of the solution net is an "ancestor" (and "descendant") of itself, i.e., the network is not a tree.

Solution nets provide a more flexible, human approach to problem solving than the simple recursive approach of NUMVAL. The plans they can represent contain the tree-like networks of strict recursion as a special case, and can be readily modified when it is discovered that an initial plan is unworkable or inferior to an altered approach. Moreover, because they provide a record of the solution strategy being pursued and the status of the solution at any given point in its execution, they give PAP the potential to "show its work", an ability which will be essential for effective interaction with the user.

Shortly we will illustrate the operation of KITOGEN, but first it is convenient to describe a second facility for qualitative reasoning presently under development, the "Qualitative SIMulation" routine QLSIM..

3.3. *QLSIM*

Consider the following simple system of equations, modeling the ignition and thermal runaway or quenching of an alpha-heated plasma:

$$\partial T/\partial t = P_\alpha - \chi T, \tag{2a}$$

$$P_\alpha = \sigma T, \tag{2b}$$

$$\chi = \chi_0 v, \quad v = v_0/T. \tag{2c}$$

(The constants σ, χ_0, and v_0 are all assumed positive.) Equation (2a) states that the plasma temperature T becomes hotter the larger the power P_α from the alphas, and colder due to losses to the wall, the greater the thermal conductivity χ, and the greater T. Equation (2b) is a model of the scaling of

the fusion cross section with T. Equations (2c) model the scaling of χ with collisionality ν (valid, e.g., in the banana regime), and the inverse dependence of ν on T. A qualitative description of the behavior of the system modeled by these equations, might be rendered as "Assume P_α is great enough that initially T is increasing. Then as T increases, ν, and so χ, both decrease, thus reducing thermal losses, while P increases. Both these effects act to further increase T, so T will increase indefinitely (until the validity of the model is violated)". A similar description may be given for quenching behavior, by assuming $\partial T(t = 0)/\partial t < 0$ instead. Though it is easy enough to explicitly solve this system for $T(t)$, the description just given is actually a formal *proof* of the runaway behavior, independent of the precise details which make the model solvable, but depending only on some simple qualitative features of the model, which would apply to a less tractable model with the same qualitative properties. QLSIM is a function which attempts this kind of qualitative analysis on systems of rate equations, i.e. systems of the form $\partial z^i/\partial t = v^i(z)$, where $z = (z^1, z^2, \ldots)$ is the set of dynamical variables z^i, and the v^i are functions of z. Thus, it is applicable to dynamics problems, such as system (1), to power-balance models like system (2), and to models describing a very broad range of other phenomena. It is an extension of previous AI work [8, 9] in "qualitative process theory" to qualitatively reasoning about real systems of equations, instead of about qualitatively-describing systems.

The following shows a sample PAP protocol (slightly edited for brevity) illustrating the application of KITOGEN and QLSIM to systems (1) and (2) above:

(C3) get(*x*,'givenby);

(D3) $$\frac{dX}{dT} = \frac{P}{M}$$

(C4) get(*p*,'givenby);

(D4) $$\frac{dP}{dT} = -K(X - XO)$$

(C5) facts(context);

(D5) [M > O, K > O, SGO > CHIO > O, NUO > O]

(C6) get(tt,'givenby);

(D6) $$\frac{dTT}{dT} = PP - CHI\ TT$$

(C9) kitogen(tt);

(D9) [DONE, ndkt =, 12]

(C10) wrglb( ); ndkt 12

$$eqlst = [PP = SGO\ TT,\ NU = \frac{NUO}{TT},$$

$$CHI = CHIO\ NU,\ \frac{dTT}{dT} = PP - CHI\ TT]$$

vvcci s [SGO, CHIO, NUO] [T] [TT] [CHI, NU, PP]

(D10) [ ]

(C11) wrtree('nd,1,inf);
ND1
TT
[DESCDT, [ND11, ND9], EXNOTES, [DESCDT, DESCDT], CHLDN, [ND2]]
ND2

$$SOLVEFOR(\frac{dTT}{dT} = PP - CHI\ TT, TT)$$

(C12) genvis( );

(D12) [SGO TT − CHIO NUO]

(C14) qlsim(vvcc2,d12,[pos]);

$$1\text{-cycle,zasmpls,sgnasmp} = [TT - \frac{CHIO\ NUO}{SGO}]\ [POS]$$

Evaluation took 6.28 seconds (32.26 elapsed).

(D14) [DONE, qlndkt =, 1]

(C15) wrtree('qlnd,1,inf);
QLND1

$$[[POS], [TT - \frac{CHIO\ NUO}{SGO}]]$$

[CHLDN, [QLND1], PARNT, [QLND1]]

(D15) DONE

(C20) kitogen(x);
Evaluation took 6.94 seconds (25.85 elapsed).

(D20) [DONE, ndkt=, 20]

(C21) wrglb( );
ndkt 20

$$eqlst = [\frac{dP}{dT} = -K(X - XO), \frac{dX}{dT} = \frac{P}{M}]$$

vvcci s [M, K, XO] [T] [X, P] [ ]

(D21) [ ]

(C22) wrtree('nd,13,inf);
ND13
X
[DESCDT, [ND18], EXNOTES, [DESCDT], CHLDN, [ND14]]
.
.
.

(C23) genvis( );

(D23) $$[\frac{P}{M}, -K(X - XO)]$$

(C24) qlsim(vvcc2,d23,[pos,pos]);
Evaluation too 18.97 seconds (38.09 elapsed).

(D24) [DONE, qlndkt=, 5]

(C25) wrtree('qlnd,2,inf);
QLND2
[[POS, POS], [P, XO − X]]
[PARNT, [QLND5], CHLDN, [QLND3]]
QLND3
[[POS, NEG], [P, X − XO]]
[CHLDN, [QLND4], PARNT, [QLND2]]
QLND4
[[NEG, NEG], [−P, X − XO]]
[CHLDN, [QLND5], PARNT, [QLND3]
QLND5

```
[[NEG, POS], [-P, XO - X]]
[CHLDN, [QLND2], PARNT, [QLND4]]

(D25)     DONE
```

This protocol essentially contains two samples of the consecutive execution of the PAP functions KITOGEN, GENVIS, and QLSIM, one sample for system (2), and the second for system (1), interspersed with the display of diagnostic information, for the purposes of demonstration. Statements C3–D6 elicit some of the governing equations from the KB. C5,D5 lists the assumptions MACSYMA makes about the signs of the parameters in the systems (1, 2).

First, KITOGEN is called on the variable (tt) of interest for system (2) in C9, and on x for system (1) in C20. From this, solution nets are generated, by accessing the KB in a manner similar to that already described for the downward-recursing portion of NUMVAL. In D9, the number of nodes (ndkt) in the resultant net is given. Two diagnostic functions (WRGLB and WRTREE) are then called to display the work KITOGEN has done. WRGLB displays ndkt, then a list (eqlst) of the equations which the construction of the solution net has shown are needed for the problem, and finally a partition of the variables involved into four "variables classes" VC_i ($i = 0$–3). This classification, derived from the structure of the set of equations found to belong in eqlst, gives information on the role each variable plays in the problem. VC_0 is the set of constant "parameters" in the problem, VC_1 is the set of "independent variables" (t alone, in the case of rate equations), VC_2 is the set of "dependent", or "dynamical" variables z^i in the problem [tt alone, in the case C10–D10 of system (2), and (x, p) in C21–D21 for system (1)]. VC_3 is the set of symbols which should be regarded as functions of the z^i.

WRTREE then writes out the contents (cf. C11, C22) of the tree or network specified by its arguments. In the case of C22, it is asked to write out the contents of nodes nd13, nd14, . . , nd20 which, graphically represented, are just the network of Fig. 3. Only the first note (nd 13, for x) is shown here, for brevity. One sees the "self-descendant" property (descdt) previously described present in this node's "execution-notes" (exnotes).

We return from the diagnostic messages. Before the execution of QLSIM, the list (vlst) of v^is corresponding to the list (zlst) of z^is is generated by the call to GENVIS at C12,C23, using the information just described coming from KITOGEN. Things are then ready for the function call QLSIM(zlst,vlst,sgnvlst) at C14,C24. Here, sgnvlst is a list giving the initial sign to be assumed for the corresponding v^is. Thus, sgnvlst = [pos] in C14 corresponds to the assumption made in the runaway argument above that initially $\partial T/\mathrm{d}t > 0$. Using this information, QLSIM determines the algebraic conditions needed to satisfy the assumptions in sgnvlst, and then qualitatively advances the system, using the logic illustrated in the English description above, developing a state diagram describing the qualitative evolution. Each state is characterized by the updated version of sgnvlst, and the set of algebraic assumptions made to imply that sgnvlst. At "1-cycle.zasmpls, . . ." following C14, QLSIM notes that the original state maps into itself, i.e., that the original growth assumed for T implies indefinitely-continued growth, so the state diagram for this qualitative simulation has only a single state, the node qlnd1, displayed by WRTREE in C15. The first element ([pos]) there is the sign of tt for that state, and the second is the algebraic expression which must be positive in order for that sign to hold. Thus, QLSIM has found the critical value in T (tt) needed for ignition/runaway for this model.

The state diagram for system (1), displayed at C25, is somewhat more interesting. There are four states, corresponding to the sequential advance of the SHO-system through the four quadrants in (x, p) space on one cycle of its familiar elliptical orbit.

It is expected that facilities such as these for qualitative analysis will considerably expand both the range of problems PAP can successfully analyse, as well as the efficiency and range of PAP's ability to communicate with the human user in a convenient manner. For example, it should be feasible to permit the user to qualitatively communicate to PAP a description of some physical process via an argument like the one given for the thermal runaway process, and then to have PAP try to fill in the details of the argument, and to check if the argument "makes sense", using the facilities KITOGEN and QLSIM. The transport calculation described in the following section shows some other ways in which the routines just described are useful.

4. An automated transport calculation

We have seen above how the PAP–KB can be used to apply known results to new sets of parameters. For example, using NUMVAL, we evaluated the banana-drift diffusion coefficient D_{bdl} for alpha parameters. However (as may also be verified using NUMVAL on the validity requirements in the KB), the standard result evaluated there does not apply to alphas, because its derivation assumes that the grad-B contribution Ω_B to the precessional drift is negligible compared with the $E \times B$ contribution Ω_E, while actually $\Omega_E \ll \Omega_B$ holds for alphas. The high-level calculator (CALC) whose operation is illustrated in this section has the potential for generating results which are "new" in the more profound sense of having a different analytic form, and, to some extent, of requiring a different solution approach, from other results. At the present early stage of implementation and experimentation, PAP has recovered some known transport results (including the standard D_{bdl} result discussed here), but no new ones. However, we see no intrinsic obstacle to its being able to generate something new (such as a form for D_{bdl} valid for alphas). A fuller description of this calculation is given in Ref. [1].

CALC uses an evaluation procedure similar to NUMVAL, except that it uses the more flexible and refined solution net approach provided by KITOGEN in the "downward", information-gathering portion of its evaluation. The solution net corresponding to the calculation is shown in Fig. 4. The PAP/user interaction is given in the protocol of Appendix A.

PAP is instructed to calculate the radial flux, "gamr" (nd1 in Fig. 4) by the function call CALC(gamr). Under the "givenby" property of gamr, CALC finds the algebraic expression (nd2 in Fig. 4)

$$\mathrm{gamr} = \mathrm{int}(\mathrm{aap}*\mathrm{rdt}*\mathrm{f},[\mathrm{th},\mathrm{y}],[0,\mathrm{indef}],[2*\%\mathrm{pi},\mathrm{indef}]), \qquad (3a)$$

or in non-computerized algebraic notation

$$\Gamma_r = \int_0^{2\pi} \mathrm{d}\theta \int \mathrm{d}v\, A' \dot{r} f. \qquad (3b)$$

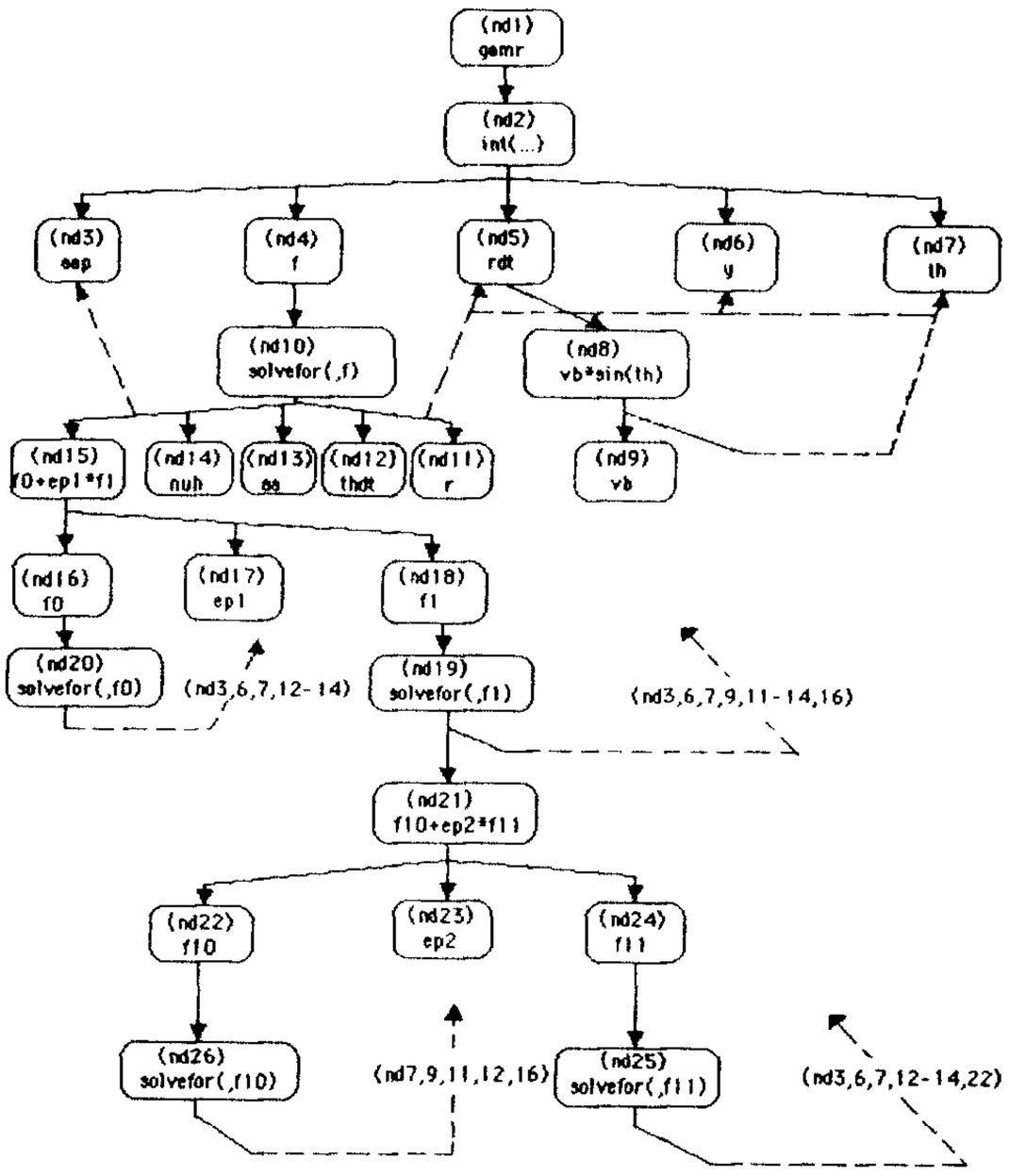

Fig. 4. Solution network generated by PAP in doing the transport calculation discussed in Section 4.

This is a representative form for the radial flux. Here, θ is the poloidal azimuth, y is a pitch-angle variable, $\dot{r}$ is the particle radial drift velocity, A is a function of y, giving the proper phase-space Jacobian, with A' its derivative, and f is the particle distribution function.

The solution-net development proceeds downward from node nd1 as previously described, until nodes such as nd10, which has as its value SOLVEFOR(boltzeqa, f), which signifies a further task which must be performed. This value is just the function call which is made in order to accomplish the task. The entrance into the SOLVEFOR attempt is announced following C4 in Appendix A, first for f, and later on for the terms f0, f1, f10, and f11 which PAP creates in attempting a perturbative solution for f.

The function call SOLVEFOR (eq, v) attempts to solve equation eq for variable v. For a (currently rather limited) range of algebraic equations, integral equations, o.d.e.s and p.d.e.s, it will successfully solve eq and return the result. The messages "homog-f0-type-de" and "slv1tm" appearing in App. A following the announcements "solvefor (f0, f10, f11)" are instances of such outright solution responses.

In addition to recognizing those equations it can solve outright (for which it returns an answer) and those it cannot solve, for some equations, SOLVEFOR recognizes a situation in which a perturbative solution might prove successful. In Appendix A this occurs following the announcements "solvefor ff1." We focus on the protocol following "solvefor f" for definiteness.

Performing its analysis of the type of equation given it, SOLVEFOR concludes it is one of the d.e.s it cannot currently solve outright, and prints "other de type" to denote this. With "consider ordering on . . . ," it then attempts to solve boltzeqa by perturbative means. In this effort (occurring within the "ordering-suggesting" function ORDRSUG), PAP makes use of the HEURMAP and NUMVAL routines described above, as well as requesting information from the user. These requests from ORDRSUG are the only portion of the calculation presently requiring user decisions.

First, ORDRSUG applies HEURMAP to each term in boltzeqa, drawing on the KB for the appropriate scale lengths with which to fill out rptlst. The result is given in the list following "term scalings". Next, at "num ratios", NUMVAL is applied to the ratio of each element in that list to the first element, again using the KB to provide those values it already knows. At this point, the user is told which values are as yet unknown, and is given the option of further specifying those values, or proceeding to the decision, based on this list of the size of the terms in the equation, of how to order them.

At "give eplst1," the user specifies the ordering of the terms in boltzeqa. Each element of boltzeqa is multiplied by the corresponding element of eplst1. The parameter ep1 is then treated as small, so the eplst1 shown corresponds to the usual assumption that the radial extent of the collisionless orbits is small compared with the minor radius of the confinement device. From the mechanized portions of ORDRSUG just described, or some extension of them, it should not be difficult to have PAP generate eplst1, or at least suggest one, but at present this has not been done.

Once given eplst1, the rest of the ordering process is automatic. f is expressed as an expansion to some specified order in ep1, this is inserted into boltzeqa, and the appropriate hierarchy ("eqhier=" below) of kinetic equations to solve is generated (in the function GENEQHIER, whose name is announced in Appendix A). The value returned by SOLVEFOR here is:

$$[F = EP1\ F1 + F0,\ INWHICH,\ [XPNPARM = EP1],$$

$$EQHIER = [\frac{dFO}{dTH}\ THDT - \frac{(\frac{d}{dY}(AA\ \frac{dFO}{Dy}))\ NUH}{AAP} = 0,$$

$$\frac{dF1}{dTH}\ THDT + \frac{dF0}{dR}\ RDT - \frac{(\frac{d}{dY}(AA\ \frac{dF1}{dY}))\ NUH}{AAP} = 0]]]$$

This may be read "The result is $f = f_0 + \varepsilon_1 f_1$(+higher order terms, if needed), IN WHICH the expansion parameter is ε_1, and the newly-introduced variables f_0, f_1 are found by solving the corresponding equations given in eqhier." When PAP receives a response of this form from SOLVEFOR, it creates new subtasks (nodes nd16,nd18 in Fig. 4) of the same form as the earlier problem (node nd4) of computing f.

Note that this represents an incremental increase of the KB. Instead of the new knowledge being "taught" to PAP by the user, the possibility for which was indicated in Section 3, here it is PAP itself which has generated the new knowledge increment. The kinetic equations generated by ORDRSUG are associated with the variables f_0 and f_1 created by PAP in the same manner as boltzeqa is associated with the member f of the permanent KB. This uniformity in the representation of newly-developed and permanent information will permit PAP to be able to operate on them in a unified fashion, and will facilitate the incorporation of new results into the KB, i.e., "learning" [10–12]. This uniformity is part of the general design philosophy for PAP, and provides constraints on the representation of both information in the KB, as well as output from PAP's routines.

Continuing with the calculation, at nd20 PAP applies

SOLVEFOR to f_0, and finds that this is an equation it can solve outright. For f_1 (nd19), it instead finds that it must try to induce a second ordering, invoking ORDRSUG as before. This time the ordering is in ep2 (ε_2), which determines the collisionality regime one is interested in. Here, an analysis of the type induced by ORDRSUG is inconclusive on whether the collisional (3rd) term in boltzeqa should be ordered large or small compared with the precessional (1st) term. This is to be expected, because realistic devices can operate in differing collisionality regimes. Indeed, the transport literature can with some validity be viewed as a search through the space of possible relative orderings of the terms in the full kinetic equation consistent with reasonable parameter values, in which conceptually the same calculation is done for each ordering. Here, PAP could be instructed to investigate a particular ordering, or all reasonable ones. The term "reasonable" is not a clearly defined one, of course. The assumption $\Omega_E \ll \Omega_B$ was not "reasonable" for the transport of thermal particles (the context in which this theory was developed), but becomes so for alphas. In Appendix A, the choice made is to order the collision frequency small. The calculation is thus of the same class of calculations as the tokamak banana regime [4], the precession-dominant banana-drift regime [3], and the superbanana regime in rippled tokamaks [13] and stellarators [14, 15], the hallmark of which is a diffusion coefficient linear in the effective collision frequency, here called nuh (ν_h). The assumption made by PAP in the present calculation, that thdt ($\dot{\theta}$) is independent of θ and y, makes it most closely analogous to the banana-drift calculation of Ref. [3], and that result may be recovered from the result generated by PAP here, with the appropriate expression substituted for vb. For all these calculations, however, most of the solution tree of Fig. 2 is valid, giving a more concrete formal meaning to the general statement that all the calculations are "conceptually similar". By using such solution trees stored in its KB to direct its approach to problems not previously attempted, it is hoped that PAP will be able to apply analogical reasoning [16–18] to apply solution approaches known to be effective for previously-treated problems to new problems of similar character.

In D4 in Appendix A, the answer for gamr is first given formally, leaving the function INT unevaluated. INT is evaluated, where possible, by the application to the formal result of INTEV, in C5,D5.

A human theorist would usually collect the two terms appearing in the integrand in D5 into one. Because of the operation of the MACSYMA function DIFF and INTEGRATE which it uses, the function INTEV leaves the result in the less than optimal form shown. Improvements in making decisions of this sort in doing formal algebraic manipulations would be useful for MACSYMA and MACSYMA-like facilities.

When the collisional term is ordered large instead of small at nd19, i.e., if eplst1 = [1,ep2,1,1] instead of the value [1,1,ep2,1] is taken in Appendix A, PAP undertakes a calculation of the "$1/\nu$" variety, also familiar from rippled tokamaks [3, 19] and stellarators [20]. PAP recovers the stellarator result, except that it as yet has no ability to reason about the appropriate boundary conditions in y, thus leaving the y-integration indefinite.

5. Conclusion

The possibilities for a facility like PAP for enhancing the learning environment and research productivity for plasma physicists seem considerable. In response to the two questions put in the Introduction, even at its present early stage of implementation, the lower-level facilities have already begun to be useful. With regard to developing "new" results, as I have attempted to illustrate by the automated transport calculation just described, most of what goes into new physics calculations is not new. What is new is often largely the new pattern in which known techniques and results are combined. Thus, the prospects for making PAP useful even at this more ambitious level also seem promising.

Acknowledgements

This work supported by United States Department of Energy Contract No. DE-AC02-CH0-3073.

Appendix A

(C4)calc(gamr);

solvefor F
other de type

consider ordering on $\frac{dF}{dR}$ SIN(TH) VB + $\frac{dF}{dTH}$ THDT

$$-\frac{(\frac{d}{dY}(AA\ \frac{dF}{dY}))\ NUH}{AAP} = 0$$

term scalings: [$\frac{F\ SIN(TH)\ VB}{A}$, F KPTH THDT,

$\frac{2\ AA\ F\ KPY^2\ NUH}{AAP}$, 0]

num ratios [1, $\frac{A\ THDT}{SIN(TH)\ VB}$,

$$\frac{8.4994B{-}13\ A\ AA\ DENSI}{AAP\ DL\ SQRT(MASS)\ SIN(TH)\ TT^{1.5d+0}VB},$$

0.0B0]
unevaled parms: [A, TH, THDT, VB, AA, AAP, DENSI, DL, MASS, TT]
give values, or continue → usr ordr
continue;
give eplst1
[ep1,1,1,1];
geneqheir

solvefor F0
homog-f0-type de

solvefor F1
other de type

consider ordering on $\frac{dF0}{dR}$ SIN(TH) VB + $\frac{dF1}{dTH}$ THDT −

$$\frac{(\frac{d}{dY}(AA\ \frac{dF1}{dY}))\ NUH}{AAP} = 0$$

term scalings: [$\frac{F0\ SIN(TH)\ VB}{A}$, F1 KPTH THDT,

$\frac{2\ AA\ F1\ KPY^2\ NUH}{AAP}$, 0]

num ratios [1, $\frac{\text{A F1 THDT}}{\text{F0 SIN(TH) VB}}$,

$\frac{\text{8.4994B–13 A AA DENSI F1}}{\text{AAP DL F0 SQRT(MASS) SIN(TH) TT}^{\text{1.5d+0}}\text{ VB}}$, 0.0B0]

unevaled parms: [A, F0, F1, TH, THDT, VB, AA, AAP, DENSI, DL, MASS, TT]

give values, or continue → usr ordr
continue;

give eplst1
[1,1,ep2,1];
geneqheir

solvefor F10
slv1tm

solvefor F11
slv1tm

(D4) INT(AAP SIN(TH) VB (EP1 (EP2

INT

$$\frac{\left(\text{NUH}\left(\frac{d}{dY}\left(\text{AA}\left(\frac{d}{dY}\left(\text{INT}\left(-\frac{\frac{dF0}{dR}\,\text{SIN(TH) VB}}{\text{THDT}}, \text{TH}\right)\right)\right)\right)\right)\right)}{\text{AAP THDT}},$$

TH)

$$+\ \text{INT}\left(\frac{\frac{dF0}{dR}\,\text{SIN(TH) VB}}{\text{THDT}}, \text{TH}\right))\ +\ \text{F0, [TH, Y],}$$

[0, INDEF], [2%PI, INDEF])

(C5) intev(%);

(D5) INT(%PI AA EP1 EP2 $\frac{dF0}{dR}$ NUH VB $\frac{d^2VB}{dY^2}$

$+$ %PI $\frac{dAA}{dY}$ EP1 EP2 $\frac{dF0}{dR}$ NUH VB $\frac{dVB}{dY}$, Y)/THDT^2

References

1. Mynick, H. E., Introducing PAP: A Plasma Apprentice Program, Princeton University Report PPPL-2323 (1986) (submitted for publication).
2. Goldston, R. J., White, R. B. and Boozer, A. H., Phys. Rev. Lett. **47**, 647 (1981).
3. Linsker, R. and Boozer, A. H., Phys. Fluids **25**, 143 (1982).
4. Hinton, F. L. and Hazeltine, R. D. Reviews of Modern Physics **48**, 239 (1976).
5. Berk, H., Rosenbluth, M. N. and Shohet, J. L., Phys Fluids **26**, 2616 (1983).
6. Fikes, R. E. and Nilsson, N. J., Artificial Intelligence **2**, No. 3/4 (1971).
7. Sacerdoti, E. D., Artificial Intelligence **5**, 115 (1974).
8. de Kleer, J. and Brown, J. S., A Qualitative Physics Based on Confluences, Xerox PARC Report (12/83).
9. Forbus, K. D., MIT-AI Laboratory Report AI-TR-789 (1984).
10. Mitchell, T. M., Rutgers University Report LCSR-TR-45 (1983).
11. Lenat, D. B., Artificial Intelligence **21**, 61 (1983).
12. Langley, P., Simon, H. A. and Bradshaw, G. L., Machine Learning, Tioga, Palo Alto, CA (1983).
13. Yushmanov, P. N., Nucl. Fusion **22**, 315 (1982).
14. Galeev, A. A. and Sagdeev, R. Z., Reviews of Plasma Physics (Edited by M. A. Leontovich), Vol. 7, p. 257, Consultants Bureau, New York (1979).
15. Mynick, H. E., Phys. Fluids **26**, 2609 (1983).
16. Winston, P. H., Artificial Intelligence **19**, 321 (1982).
17. Burstein, M. H., AAAI-83, Proceedings of National Conference on Artificial Intelligence, p. 45, Washington, DC (1983).
18. Rissland, E. L., Valcarce, E. M. and Ashley, K. D., AAAI-84 Proceedings of National Conference on Artificial Intelligence, Austin, TX (1984).
19. Connor, J. W. and Hastie, R. J., Nucl. Fusion **13**, 221 (1973).
20. Connor, J. W. and Hastie, R. J., Phys. Fluids **17**, 114 (1974).