

SYMBOLIC COMPUTATION OF NONLINEAR WAVE INTERACTIONS ON MACSYMA*

A. BERS, J.L. KULP and C.F.F. KARNEY

*Research Laboratory of Electronics and Laboratory for Computer Science,
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

In this paper we describe the use of a large symbolic computation system – MACSYMA – in determining approximate analytic expressions for the nonlinear coupling of waves in an anisotropic plasma. MACSYMA was used to implement the solutions of a fluid plasma model nonlinear partial differential equations by perturbation expansions and subsequent iterative analytic computations. By interacting with the details of the symbolic computation, the physical processes responsible for particular nonlinear wave interactions could be uncovered and appropriate approximations introduced so as to simplify the final analytic result. Details of the MACSYMA system and its use are discussed and illustrated.

1. Introduction

Analytical and numerical methods are used complementarily for the understanding of physical phenomena and the solution of technical problems. The use of computers in numerical simulations of physical problems has a long history of development and success, and is indeed intimately tied to the history of development of large computers. On the other hand, the use of computers in carrying out algebraic and other mathematical operations in symbolic form is of much more recent origin, and primarily connected with developments in programming languages and symbolic computation algorithms.

The importance of the use of computers for both numerical and analytical work is best illustrated in the context of solving and understanding many of the complex plasma dynamics problems. Thus, the use of computers in numerically simulating charged-particle dynamics has increased our understanding of certain types of nonlinear plasma phenomena. However, such numerical particle simulations become impractical for studying nonlinear phenomena involving several disparate time and space scales, and evolving in three dimensions. In addition, simulations give a particular picture of the physical phenomena but not an analytic description of them. Hence, scaling laws and interrelations to other phenomena are difficult to extract from such simulations. By contrast, many nonlinear plasma problems can be usefully studied by analytic perturbation techniques. Here, the handling of three dimensions and disparate time and space scales presents no problem. However, the analytical calculations that need to be performed are often very tedious, involving long algebraic computations with tensor quantities and integro-differential operators. In addition, the aim of such calculations is often to obtain approximate analytic expressions that describe the phenomena of interest and their underlying physics. For such calculations the use of a computer to carry out the tedious analytic operations is clearly of great importance. It is the purpose of this paper to show that the progress of recent years in development of the symbolic computation system MACSYMA has made it possible for us to carry out such perturbation analyses for complex nonlinear wave interactions in a plasma. It will be obvious, hopefully, that such symbolic computation techniques can be readily adapted to other nonlinear plasma problems, e.g. wave-particle interactions and turbulence, and MHD equilibrium and stability problems in complex magnetic fields and geometric configurations.

* This work was supported in part by the Energy Research and Development Administration Grant Number E(11-1)-3070 and in part by the National Science Foundation Grant Number ENG75-06242.

In section 2 we present a short review of symbolic computation systems and their use in plasma physics. Section 3 outlines the general formulation of nonlinear interactions of coherent waves in a plasma. Section 4 illustrates the symbolic computation on MACSYMA for a particular nonlinear interaction of three waves. This shows how we determined the dominant physical processes responsible for the interaction, and through this, how we obtained a simple, approximate analytic description of the interaction. Section 5 describes some of the other current uses of MACSYMA and discusses the future potential of symbolic computation in plasma physics. The MACSYMA system and examples from its use and described in the Appendix.

2. Symbolic computation systems

A symbolic or algebraic manipulation system is a computer program which recognizes and accepts mathematical formulae in symbolic form, executes certain mathematical operations on these formulae, and returns a symbolic expression of the results of such operations. Computer programs for carrying out algebraic tasks have been in development for about 15 years. The early (up to about 1970) work on symbolic manipulation systems and their application to physics and engineering problems has been reviewed by Barton and Fitch [1]. These symbolic manipulation systems have been most successfully used on problems where the procedure for deriving an analytic solution is very well defined and straightforward to code, and the symbolic computation is frequently used as an intermediate aid in setting up numerical calculations. Such systems are usually limited to specific representations for handling symbolic expressions. They can be very efficient in handling some specific symbolic computations but lack the breadth needed for carrying out a large variety of mathematical operations. More recent developments in symbolic manipulation systems can be found in the proceeding of various recent conferences on symbolic computation [2, 3]. The MACSYMA system at MIT is a large symbolic manipulation system with multiple representations and many built-in facilities. It is geared to interactive use, but batch operation is also available. Its wide range of interactive capabilities allows one to attack problems which have reasonably simple approximate solutions following from very long and complex intermediate results. In the Appendix we give an illustration of some of MACSYMA's facilities and several examples of its capabilities. A more detailed technical description of MACSYMA can be found in refs. [2] and [3]. All of the current facilities and capabilities on MACSYMA can be found in its most recent manual [4].

The use of symbolic computation in plasma physics problems dates back to the late 1960s. Some early examples were the use of FORMAC in generating higher order terms in the magnetic moment series for a charged particle (R. Lewis, unpublished), and the use of symbolic ALGOL for the expression and solution of partial differential equations in MHD, Roberts et al. [5] and Petravac et al. [6]. The REDUCE system has been used in calculations of integrals that arise in MHD stability analysis [7, 8]. Both linear and nonlinear dispersion relations have been derived using FORMAC [9]. FORMAC has also been used as an aid in setting up Tokamak fluid codes [10]. We at MIT, using MACSYMA, have been studying nonlinear wave-wave interactions [11–13]. In this work we have paid particular attention to developing facilities for investigating various limiting forms of large analytic solutions derived by the computer. This work is described in detail in the theses of Kulp [14] and Karney [15], and is briefly reported in Kulp et al. [16] and Kulp [19]. A detailed report on our work is in preparation [18].

3. Nonlinear wave coupling theory

Briefly, the formalism for describing second-order wave interactions in a homogeneous plasma proceeds as follows [19]. The first-order, linear relationship between the current (for a particular species of charged particle) and the electric field is

$$\vec{J}^{(1)} = \bar{\sigma}(\vec{k}, \omega) \vec{E}, \quad (1)$$

where the linear conductivity tensor $\bar{\bar{\sigma}}$ is determined from the linearized equations of the plasma model. This together with Maxwell's equations lead to the dispersion relation for small-amplitude waves

$$\det [\bar{\bar{D}}(\bar{k}, \omega)] = D(\bar{k}, \omega) = 0, \quad (2a)$$

where

$$\bar{\bar{D}}(\bar{k}, \omega) = \left(\frac{c}{\omega}\right)^2 \bar{k}\bar{k} + \left[1 - \left(\frac{ck}{\omega}\right)^2\right] \bar{I} + \frac{i\bar{\bar{\sigma}}(\bar{k}, \omega)}{\omega\epsilon_0}, \quad (2b)$$

giving $\omega(\bar{k})$. When the plasma model equations are solved to second-order in the electric field, one obtains the nonlinear conductivity tensor (third rank) relating the second-order current to the product of two electric field components in general at different wave vectors and frequencies,

$$\bar{J}^{(2)} = \bar{\bar{\bar{\sigma}}}_{a,b} \bar{E}_a \bar{E}_b. \quad (3)$$

Here the subscripts refer to functional dependence on (\bar{k}_a, ω_a) and (\bar{k}_b, ω_b) . The effect of this nonlinear current can be studied in a perturbation theory framework. Letting \bar{E}_a and \bar{E}_b be linear wave fields, we can regard eq. (3) as giving rise to a current that will perturb the linear wave field \bar{E}_n , where

$$\bar{k}_n = \bar{k}_a + \bar{k}_b, \quad \omega_n = \omega_a + \omega_b. \quad (4, 5)$$

When such a triplet of fields are weakly damped waves, with each (\bar{k}, ω) approximately satisfying eq. (2), the effect of the nonlinear current is termed as giving rise to a "resonant three-wave interaction". Combining eqs. (1) and (3) with Maxwell's equations, this three-wave coupling can be expressed in a set of equations that give the slow variation in space and time (compared to k^{-1} and ω^{-1} , respectively) of the complex amplitudes of the waves. Letting $a = 1, b = 2, n = 3$, we then find

$$\left[\frac{\partial}{\partial t} + \bar{v}_{g1} \cdot \bar{\nabla} + \gamma_1 \right] a_1 = p_1 K a_2 a_3, \quad (6)$$

$$\left[\frac{\partial}{\partial t} + \bar{v}_{g2} \cdot \bar{\nabla} + \gamma_2 \right] a_2 = -p_2 K^* a_1 a_3^*, \quad (7)$$

$$\left[\frac{\partial}{\partial t} + \bar{v}_{g3} \cdot \bar{\nabla} + \gamma_3 \right] a_3 = -p_3 K^* a_1 a_2^*, \quad (8)$$

where $a(\bar{r}, t)$ is the slowly varying complex amplitude of the mode, $\bar{v}_g = \partial\omega/\partial\bar{k}$ is its group velocity, γ is its weak temporal damping rate, $p = \pm 1$ is the sign of its time-averaged energy density $w = (\omega\epsilon_0/4)\bar{E}^* \cdot [(\partial/\partial\omega)\bar{\bar{D}}] \cdot \bar{E}$, and K is the nonlinear coupling constant for a conservative interaction, i.e. one for which energy and momentum are conserved. If $a(\bar{r}, t)$ is normalized so that $p|a|^2 = w/\omega$, the action density of the wave, then

$$K = M_0/4 |a_{10} a_{20} a_{30}|, \quad (9)$$

where

$$M = \frac{-\bar{E}_1^* \cdot \bar{J}_1^{(2)}}{\omega_1} = \frac{\bar{E}_2 \cdot \bar{J}_2^{(2)*}}{\omega_2} = \frac{\bar{E}_3 \cdot \bar{J}_3^{(2)*}}{\omega_3}, \quad (10)$$

and the subscript 0 indicates that the quantities are evaluated using the unperturbed linear fields. Thus the nonlinear, conservative coupling of three waves can be evaluated from a detailed study of eqs. (9) and (10). The physics of the nonlinear coupling is contained in M , where, for example, the coupling of waves 1 and 2 to wave 3 is given by $\bar{E}_3 \cdot \bar{J}_3^{(2)*}$, etc.

The nonlinear coupling constant K describes the dynamics of the three-wave interaction. For example, for three positive energy waves, if wave 1 can be considered as a constant amplitude pump, $a_1 = a_{10}$, and $\gamma_i = 0$, then small-

amplitude waves 2 and 3 may grow in time at a maximum rate given by $\gamma_0 = |K| |a_{10}|$, which is the maximum parametric excitation rate of waves 2 and 3. A complete solution of the nonlinear set of eqs. (6)–(8) with $\gamma_i = 0$ has also been found recently [20], but will not concern us here.

In the following section we shall describe the use of MACSYMA in deriving and understanding the nonlinear coupling coefficient M for a particular plasma model and a particular set of plasma waves in a magnetic field. Needless to say, this can be done here only in a sketchy manner. To understand the details behind the various commands used, the reader should not only consult the Appendix but the MACSYMA manual as well [4]. The next section should however suffice to show in broad terms our use of an advanced symbolic computation system.

4. Nonlinear coupling derived with MACSYMA

For the purpose of illustration, consider the equations for a multi-species warm-fluid model of a plasma:

$$(\partial/\partial t)\mathbf{n} + \bar{\mathbf{v}} \cdot (\mathbf{n}\bar{\mathbf{v}}) = 0, \quad (11)$$

$$\frac{\partial}{\partial t} \bar{\mathbf{v}} + (\bar{\mathbf{v}} \cdot \bar{\mathbf{v}})\bar{\mathbf{v}} = -v_{\text{th}}^2 \left(\frac{n_0}{\mathbf{n}} \right) \bar{\mathbf{v}} \left(\frac{\mathbf{n}}{n_0} \right)^\gamma + \frac{q}{m} \bar{\mathbf{E}} + \frac{q}{m} \bar{\mathbf{v}} \times \bar{\mathbf{B}}, \quad (12)$$

$$\bar{\mathbf{J}} = \sum_s q_s n_s \bar{\mathbf{v}}_s, \quad (13)$$

$$\bar{\mathbf{v}} \times \bar{\mathbf{E}} = -\frac{\partial}{\partial t} \bar{\mathbf{B}}, \quad \bar{\mathbf{v}} \times \frac{\bar{\mathbf{B}}}{\mu_0} = \bar{\mathbf{J}} + \frac{\partial}{\partial t} \epsilon_0 \bar{\mathbf{E}}, \quad (14, 15)$$

where $\bar{\mathbf{v}}$, \mathbf{n} , $\bar{\mathbf{J}}$, and $\bar{\mathbf{B}}$ are field variables depending on space and time $(\bar{\mathbf{r}}, t)$, $\bar{\mathbf{E}}(\bar{\mathbf{r}}, t)$ is assumed given, and q , m , n_0 , v_{th}^2 , and γ are constants. Note that these equations contain vector functions $(\bar{\mathbf{v}}, \bar{\mathbf{J}}, \bar{\mathbf{B}}, \bar{\mathbf{E}})$, vector operators (in three spatial dimensions), and five nonlinearities [e.g. $q\mathbf{n}(\bar{\mathbf{r}}, t)\bar{\mathbf{v}}(\bar{\mathbf{r}}, t)$]. To make these equations tractable, we expand them by expressing the dependent variables as power series in a small parameter, for instance, $\mathbf{n}(\bar{\mathbf{r}}, t) \rightarrow n_0 + \epsilon \mathbf{n}^{(1)}(\bar{\mathbf{r}}, t) + \epsilon^2 \mathbf{n}^{(2)}(\bar{\mathbf{r}}, t) + \dots$, and then by ordering the equations in ϵ . For example, eq. (12) of order ϵ^2 becomes

$$\frac{\partial}{\partial t} \bar{\mathbf{v}}^{(2)} + \gamma v_{\text{th}}^2 \bar{\mathbf{v}} \frac{\mathbf{n}^{(2)}}{n_0} + \frac{q}{m} \bar{\mathbf{B}}_0 \times \bar{\mathbf{v}}^{(2)} = -(\bar{\mathbf{v}}^{(1)} \cdot \bar{\mathbf{v}})\bar{\mathbf{v}}^{(1)} + \frac{q}{m} \bar{\mathbf{v}}^{(1)} \times \bar{\mathbf{B}} - \gamma(\gamma - 2)v_{\text{th}}^2 \frac{\mathbf{n}^{(1)}}{n_0} \bar{\mathbf{v}} \frac{\mathbf{n}^{(1)}}{n_0}, \quad (16)$$

where we have assumed $\bar{\mathbf{E}} \sim O(\epsilon)$ and $\bar{\mathbf{v}}^{(0)}(\bar{\mathbf{r}}, t) \equiv 0$.

Now to find the second order (ϵ^2) coupling between linear solutions (order ϵ), it is necessary to compute $\bar{\mathbf{J}}^{(2)}$ where, from eq. (13), $\bar{\mathbf{J}}^{(2)} = qn_0 \bar{\mathbf{v}}^{(2)} + q\mathbf{n}^{(1)} \bar{\mathbf{v}}^{(1)}$. This is accomplished by solving the first order equations for the variables, $\bar{\mathbf{v}}^{(1)}$, $\mathbf{n}^{(1)}$, and $\bar{\mathbf{B}}$, using an assumed form for $\bar{\mathbf{E}}(\bar{\mathbf{r}}, t)$ (i.e. in terms of $\bar{\mathbf{E}}$). Then the first order solutions are used as sources or driving terms to the second order equations. Eq. (16) has been structured to reflect this distinction, with all terms involving first order quantities appearing on the right side of the equation. All of these terms on the right side of eq. (16) contain products of first order variables, as expected from the ordering process. If the second order equations for $\bar{\mathbf{v}}^{(2)}$ and $\mathbf{n}^{(2)}$ are solved in terms of the first order variables, and if the first order variables are eliminated in terms of $\bar{\mathbf{E}}$, then the products of the first order terms (e.g. $\mathbf{n}^{(1)} \bar{\mathbf{v}}^{(1)}$) become an expression proportional to $\bar{\mathbf{E}}^2$. Thus we obtain $\bar{\mathbf{J}}^{(2)}$ expressed as in eq. (3). This solution process can be iterated to higher orders, so that, for example, the third order (ϵ^3) coupling can be found by calculating $\bar{\mathbf{J}}^{(3)}$ in terms of products like $\mathbf{n}^{(1)} \bar{\mathbf{v}}^{(2)}$, etc.

To find $\bar{\mathbf{J}}^{(1)}$ and $\bar{\mathbf{J}}^{(2)}$, we apply Fourier transforms in space and time to the set of ordered equations typified by eq. (16). Consistent with our goal of studying a specific collection of waves, we assume $\bar{\mathbf{E}}(\bar{\mathbf{r}}, t)$ is composed of a discrete set of plane waves

$$\bar{\mathbf{E}}(\bar{\mathbf{r}}, t) = \sum_n \bar{\mathbf{E}}_n(\bar{\mathbf{r}}, t) \exp(-i\omega_n t + i\bar{\mathbf{k}}_n \cdot \bar{\mathbf{r}}).$$

The transformed equations in the complex Fourier amplitudes, $\bar{\mathbf{v}}_n, \mathbf{n}_n, \bar{\mathbf{E}}_n$, etc. are solved algebraically for $\bar{\mathbf{J}}^{(1)}$ by using the equations derived from eqs. (11) and (14) to eliminate $\bar{\mathbf{B}}$ and $\mathbf{n}^{(1)}$ in the first order form of eq. (12). This relates $\bar{\mathbf{v}}_n^{(1)}$ to $\bar{\mathbf{E}}_n$. Now this equation can be used in the first order form of eq. (13), $\bar{\mathbf{J}}_n^{(1)} = qn_0 \bar{\mathbf{v}}_n^{(1)}$, to relate $\bar{\mathbf{J}}_n^{(1)}$ to $\bar{\mathbf{E}}_n$. The result is shown below in eq. (17). There, $\bar{\mathbf{B}}_0 \times$ is a matrix operator $\bar{\mathbf{k}}_n \bar{\mathbf{k}}_n$ is a matrix $(k_i k_j)$, and $\bar{\mathbf{I}}$ is the identity matrix. Eq. (17) can be viewed as a matrix operator applied to $\bar{\mathbf{E}}_n$, producing $\bar{\mathbf{J}}_n^{(1)}$. This operator can be extracted and defined as a linear conductivity, $\bar{\sigma}_n \equiv \bar{\sigma}(\bar{\mathbf{k}}_n, \omega_n)$:

$$\bar{\mathbf{J}}_n^{(1)} = \bar{\sigma}_n(\bar{\mathbf{k}}_n, \omega_n) \cdot \bar{\mathbf{E}}_n = qn_0 \left(\frac{q}{m} \bar{\mathbf{B}}_0 \times + -i\omega_n \bar{\mathbf{I}} + \frac{i\gamma v_{\text{th}}^2 \bar{\mathbf{k}}_n \bar{\mathbf{k}}_n}{\omega_n} \right)^{-1} \cdot \frac{q}{m} \bar{\mathbf{E}}_n. \quad (17)$$

To solve for $\bar{\mathbf{J}}^{(2)}$, we eliminate $\bar{\mathbf{B}}, \mathbf{n}^{(1)}$, and $\mathbf{n}^{(2)}$ from eq. (16), and express $\bar{\mathbf{v}}^{(1)}$ in terms of $\bar{\mathbf{E}}$. Note that the nonlinear products in $\bar{\mathbf{J}}^{(2)}$, such as $qn^{(1)}\bar{\mathbf{v}}^{(1)}$, will result in convolutions in frequency, when Fourier transformed. The complete expression for $\bar{\mathbf{J}}_n^{(2)}$ is given in eq. (18). A third rank tensor, $\bar{\sigma}_{n,a,b}$ operating on $\bar{\mathbf{E}}_a \bar{\mathbf{E}}_b$ can be extracted from the formula for $\bar{\mathbf{J}}_n^{(2)}$, similar to the way $\bar{\sigma}_n$ was extracted from $\bar{\mathbf{J}}_n^{(1)}$.

$$\begin{aligned} \bar{\mathbf{J}}_n^{(2)} = & \frac{1}{qn_0} (\bar{\mathbf{E}}_b \cdot \bar{\sigma}_a \cdot \bar{\mathbf{E}}_a) \left(\bar{\sigma}_n \cdot \frac{\bar{\mathbf{k}}_b}{\omega_b} \right) - \frac{1}{qn_0} \left(\frac{\bar{\mathbf{k}}_b}{\omega_b} \cdot \bar{\sigma}_a \cdot \bar{\mathbf{E}}_a \right) (\bar{\sigma}_n \cdot \bar{\mathbf{E}}_b) \quad (\text{Lorentz}) \\ & - i \frac{m}{q^3 n_0^2} (\bar{\mathbf{k}}_b \cdot \bar{\sigma}_a \cdot \bar{\mathbf{E}}_a) \bar{\sigma}_n \cdot (\bar{\sigma}_b \cdot \bar{\mathbf{E}}_b) \quad (\text{convective}) \\ & + \frac{1}{qn_0} \left(\frac{\bar{\mathbf{k}}_a}{\omega_a} \cdot \bar{\sigma}_a \cdot \bar{\mathbf{E}}_a \right) (\bar{\sigma}_b \cdot \bar{\mathbf{E}}_b) \quad (\text{current}) \\ & - i \frac{\gamma v_{\text{th}}^2}{q^3 n_0^2} \left(\frac{\bar{\mathbf{k}}_n}{\omega_n} \cdot \bar{\sigma}_b \cdot \bar{\mathbf{E}}_b \right) \left(\frac{\bar{\mathbf{k}}_a}{\omega_a} \cdot \bar{\sigma}_a \cdot \bar{\mathbf{E}}_a \right) (\bar{\sigma}_n \cdot \bar{\mathbf{E}}_n) \quad (\text{continuity}) \\ & - i \frac{\gamma(\gamma-2)v_{\text{th}}^2}{q^3 n_0^2} \left(\frac{\bar{\mathbf{k}}_b}{\omega_b} \cdot \bar{\sigma}_b \cdot \bar{\mathbf{E}}_b \right) \left(\frac{\bar{\mathbf{k}}_a}{\omega_a} \cdot \bar{\sigma}_a \cdot \bar{\mathbf{E}}_a \right) (\bar{\sigma}_n \cdot \bar{\mathbf{k}}_n) \quad (\text{pressure}) \\ & + (a \leftrightarrow b). \end{aligned} \quad (18)$$

If electrons and different species of ions are considered in the plasma model, eqs. (17) and (18) are extended by summing the conductivities over the various particle types. Also, all of the frequencies (ω_n) and wavenumbers ($\bar{\mathbf{k}}_n$) are related by the characteristic (dispersion) equation (2), $\mathbf{D}(\bar{\mathbf{k}}_n, \omega_n) = 0$. An important feature of eq. (18) is the labels along the right margin. These correspond to the origin of each nonlinear term in the original eqs. (11)–(15) as indicated in table 1. These labels will be referred to subsequently.

Note that the expression for $\bar{\mathbf{J}}_n^{(2)}$ involves about 46 matrix multiplications between $\bar{\sigma}, \bar{\mathbf{E}}$, and $\bar{\mathbf{k}}$, or 92 multiplications when both the ions and electrons of a plasma are considered. The elements of $\bar{\sigma}$ are quite complicated in general. While this computation can be done, with some effort, by hand, the third order (ϵ^3) case would require an enormous effort.

We want MACSYMA to do the calculation of the general form of $\bar{\mathbf{E}}_n \cdot \bar{\mathbf{J}}_n^{(2)}$ in such a way that at each step the equations would be in a structured form similar to that which we would obtain doing the calculation by hand. This facilitates the check of the final expression and the observation of the propagation of physically important terms through the calculation. The final result is restructured to be compact and to display physical insights such as symmetry.

The first requirement is to express the vector–matrix differential equations (11)–(15) in the MACSYMA system.

Table 1

Equation	Nonlinear term	Name
11	$\bar{\mathbf{V}} \cdot (n\bar{\mathbf{U}})$	continuity
12	$(\bar{\mathbf{U}} \cdot \bar{\nabla})\bar{\mathbf{U}}$	convective
12	$v_{\text{th}}^2 \left(\frac{n_0}{n}\right) \bar{\nabla} \left(\frac{n}{n_0}\right)^\gamma$	pressure
12	$\bar{\mathbf{U}} \times \bar{\mathbf{B}}$	Lorentz
13	$qn\bar{\mathbf{U}}$	flow-current

Vectors are represented implicitly as single symbols by declaring them to be nonscalar variables (i.e. have a NON-SCALAR property associated with them). They can also be represented explicitly as a 1×3 matrix. The implicit form is the most efficient to use until actual matrix operations are executed. Certain vector operators, which are not an inherent part of the MACSYMA system, are either represented using the noncommutative multiplication operator “ \cdot ”, as in $\text{GRAD} \cdot \mathbf{N}$, or as a new operator, by extending the syntax of MACSYMA, as with the vector cross product, “ $\cdot \times$ ” (e.g. $\bar{\mathbf{A}} \times \bar{\mathbf{B}} \rightarrow \mathbf{A} \cdot \times \cdot \mathbf{B}$). Eq. (12) is entered into MACSYMA by the sequence:

(C2) DDTASSDC:FALSE\$

(C3) DEPENDENCIES (V (r, t), B (r, t), n (r, t), E (r, t)) \$

(C4) DECLARE ([V, E, B], NONSCALAR, [g, q, m], CONSTANT) \$

(C5) INFIX (“ $\cdot \times$.”) \$

(C6) DIFF (V, t) +V.GRAD.V-TBOX (q/m)*E+TBOX (q/m)*(V.X.B)
+v [th] ↑2*(TBOX (n [0] /n) *(GRAD.TBOX (n/n [0]) ↑g, PARENS) ;

$$(D6) \quad \frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot (\text{GRAD} \cdot \mathbf{V}) - \left(\frac{\mathbf{0}}{n}\right) (\text{GRAD} \cdot \left(\frac{n}{n_0}\right)^g) v_{\text{th}}^2 + \frac{q}{m} \mathbf{V} \cdot \mathbf{X} \cdot \mathbf{B} + \frac{q}{m} \mathbf{E}$$

In performing the perturbation expansion, we follow the heuristic technique of substituting a series expansion for the dependent variables ($\bar{\mathbf{U}}, \mathbf{n}, \bar{\mathbf{E}}, \bar{\mathbf{B}}, \bar{\mathbf{J}}$) into the equations, distributing all products over sums, and extracting coefficients of the expansion parameter. This requires the following capabilities:

(1) effect the distribution of multilinear operators,

$$F(a_0 + \epsilon a_1 + \dots, b_0 + \epsilon b_1 + \dots) \rightarrow F(a_0, b_0) + F(a_0, \epsilon b_1) + F(a_1 \epsilon, b_0) + \dots ;$$

(2) Taylor expand nonlinear operators,

$$\bar{\nabla} \left(\frac{n}{n_0}\right)^\gamma \rightarrow \gamma \left(\frac{n}{n_0}\right)^{\gamma-1} \bar{\nabla} \left(\frac{n}{n_0}\right) \rightarrow \gamma \left[1 + (\gamma-2) \left(\frac{n-n_0}{n_0}\right) + (\gamma-2)(\gamma-3) \left(\frac{n-n_0}{n_0}\right)^2 + \dots \right] \bar{\nabla} \left(\frac{n}{n_0}\right) ;$$

and

(3) extract the expansion parameter from operators $[F(\epsilon^n a_n) \rightarrow \epsilon^n F(a_n)]$.

Step (1) can easily, though somewhat inefficiently, be achieved using the pattern matching facility in MACSYMA. The example given next demonstrates how an expansion capability for special operators can be entered into the MACSYMA system. Command C12 shows the rules, defined in C7 and C8, at work expanding D11. Refer to the MACSYMA Reference Manual [4] for explanations of the TELLSIMP rule definitions. These rules are automatically invoked by the simplifier in C12 when expansion is requested by EXPAND.

```
(C2) SUMP(X) := IS( NOT(ATOM(X)) AND PART(X,0) = "+" ) $
(C4) SUMPRED(X) := IS( EXPDP#0 AND SUMP(X) ) $
(C5) MATCHDECLARE( [SUM1,SUM2], SUMPRED) $
(C7) TELLSIMP( ATRUE .X. SUM2, MAP( LAMBDA( [Z], ATRUE .X. Z), SUM2) ) $
RULE PLACED ON .X.
(C8) TELLSIMP( SUM1 .X. BTRUE, MAP( LAMBDA( [Z], Z .X. BTRUE), SUM1) ) $
RULE PLACED ON .X.
(C11) (A[0]+E*A[1]+(E^2*A[2])) .X. (B[0]+E*B[1]+E^2*B[2]);
(D11)
      (A
       0
      + A
       1
      E + A
       2
      E2) .X. (B
       0
      + B
       1
      E + B
       2
      E2)
(C12) D11,EXPAND;
(D12) A
       0
      .X. B
       0
      + A
       0
      .X. (B
       1
      E) + A
       0
      .X. (B
       2
      E2) + (A
       1
      E) .X. B
       0
      + (A
       1
      E) .X. (B
       1
      E) + (A
       1
      E) .X. (B
       2
      E2) + (A
       2
      E2) .X. B
       0
      + (A
       2
      E2) .X. (B
       1
      E) + (A
       2
      E2) .X. (B
       2
      E2)
```

For expressions involving only scalar functions, the built-in Taylor series facility is adequate to perform step (2) (this facility is illustrated in the Appendix). In general, pattern rules must be defined for operators to accomplish step (3) but there do exist the built-in rules for the dot operator,

$$M_1 \cdot (s * M_2) \rightarrow s * (M_1 \cdot M_2) \text{ and } (M_1 * s) \cdot M_2 \rightarrow s * (M_1 \cdot M_2),$$

($M_1, M_2 =$ nonscalar expressions, $s =$ a scalar expression). These can be invoked efficiently in cases like $\text{GRAD} \cdot \epsilon \mathbf{n}_1 \rightarrow \epsilon \text{GRAD} \cdot \mathbf{n}_1$.

The heuristic scheme described above has been implemented in a function EXPEQS which is illustrated below.

```
(C26) EQSJ:EXPEQS(CURRENT, [J,N,V],0,3) $
EXPANSION COMPLETE
(E26)
      J0 = V0 N0 q
(E27)
      J1 = N0 q V1 + V0 q N1
(E28)
      J2 = N0 q V2 + q N1 V1 + V0 q N2
(E29)
      J3 = N0 q V3 + q N1 V2 + q N2 V1 + V0 q N3
```

The Fourier transform of the expanded equations is, again, carried out by heuristic matching rules. These rules include transforming operators such as GRAD, transforming all dependent variables, and transforming commutative and noncommutative products of dependent variables into convolutions.

The standard technique of “factoring” a vector operator and then inverting this operator can be achieved using the inverse of noncommutative multiplication (i.e. “ \uparrow ” is the exponential for the dot operator so that $A \cdot A \cdot A = A \uparrow \uparrow 3$ and $A \uparrow \uparrow (-1) \cdot A \rightarrow 1$). For example, consider solving eq. (12) to order ϵ . Below, command C13 multiplies each term on the left side of FORCEVE [eq. (12) transformed] by $V1_N^{-1}$ and thus “pulls-out” a factor of $V1_N$ ($V1_N$ is the MACSYMA representation of $\bar{v}_n^{(1)}$). The display of D13 is suppressed. Command C14 similarly “pulls-out” E_N from the right-hand side. The result of these operations is seen in D14. Command C15 solves for $V1_N$ by inverting the operator to the left of $V1_N$ in D14. The operator on the right side of D15 could now be extracted to define $\text{SIGMA}_N(\bar{v}_n)$.

(C13) SUBSPART (MULTTHRU (PIECE . V1 [n] ↑↑-1) . V1 [n] , FORCEVE , 1) \$

(C14) FORCEFAC:
SUBSPART ((TBOX (q/m) * EXPAND (MULTTHRU (PIECE . E [n] ↑↑-1) / TBOX (q/m))) . E [n] , %, 2);

$$(D14) \left(\frac{\%I (g v_{th}^2) (k_n \cdot k_n^T)}{\omega_n - v\theta \cdot k_n} - \%I \omega_n + (B\theta \cdot XPDT) \left(\frac{q}{m}\right) + \%I (v\theta \cdot k_n) \right) \cdot V1_n$$

$$= \left(\frac{q}{m}\right) \left(\left(\frac{v\theta \cdot (XPDT \cdot (k_n \cdot XPDT))}{\omega_n} + 1 \right) \cdot E_n \right)$$

(C15) V1SOL:EV (MAP (LAMBDA (X) , (LHSLINDP:PART (FORCEFAC, 1, 1)) ↑↑-1 . X) , FORCEFAC) ,
EVAL, DOTSCALRULES=FALSE);

(D15) $V1_n$

$$= \left(\frac{\%I (g v_{th}^2) (k_n \cdot k_n^T)}{\omega_n - v\theta \cdot k_n} - \%I \omega_n + (B\theta \cdot XPDT) \left(\frac{q}{m}\right) + \%I (v\theta \cdot k_n) \right)^{-1} \cdot \left(\frac{q}{m}\right) \left(\left(\frac{v\theta \cdot (XPDT \cdot (k_n \cdot XPDT))}{\omega_n} + 1 \right) \cdot E_n \right)$$

Note that in the above example $\bar{v}^{(0)}$ is nonzero. Similar operations are used to derive $\bar{J}_n^{(2)}$.

The next objective is to restructure the solved equations to show compactness, symmetry, and other useful characteristics. A technique we call “boxing” is used to label and isolate terms in an expression. “Boxes” are non-evaluating functions with the special display property of enclosing its argument in a box of characters. Also boxes have the effect of isolating expressions so that the simplifier can not combine them with other expressions. The display of these boxes may be turned on and off so as to avoid cluttering the display of expressions containing boxes except when specifically requested. A calculation can be effectively controlled by selectively removing boxes, allowing only particular subexpressions to combine in a given step. This makes a computation easier to comprehend when tracing the development of specific terms. As an example of restructuring using boxes, consider the current and convective terms of $\bar{E}_n \cdot \bar{J}_n^{(2)}$:

```
*FLOWCUR*****
*
*          *NB*****          *NA*****
*          *K . (SIGMA . E)*   *K . (SIGMA . E)*
*
*   *VA*****          *VB*****
*   *B . (SIGMA . E)*   *A . (SIGMA . E)*
*
(D5) * (E . (SIGMA . E)* + * (SIGMA . E)* ) (-----)
*   * N . A * * B * * A * * B * * 2 W N0 QI
*   * * * * * * * * * * * * * * * * * * * * * * *
*   * * * * * * * * * * * * * * * * * * * * * *
*****
```

```
*CONVEC*****
*
*   *VA*****          *VB*****
*   *K . (SIGMA . E)*   *K . (SIGMA . E)*
*
*   *VA*****          *VB*****
*   *A . (SIGMA . E)*   *B . (SIGMA . E)*
*
*   *VA*****          *VB*****
*   *A . (SIGMA . E)*   *B . (SIGMA . E)*
*
*   *VA*****          *VB*****
*   *A . (SIGMA . E)*   *B . (SIGMA . E)*
*
*****
```

$$\left(\frac{\%I}{2 \omega_n N_0 \epsilon \pi \omega_n^2 QI} \right)$$

The labels VA, VB indicate terms coming from $\bar{v}_a^{(1)}, \bar{v}_b^{(1)}$ and NA, NB denote those from $n_a^{(1)}, n_b^{(1)}$. Note that some factored terms are grouped in a non-displaying box on the right end of each of FLOWCUR and CONVEC.

The reduction of the general expression for $\bar{E}_n \cdot \bar{J}_n^{(2)}$ can be performed in several ways. First if we were just interested in the answer, and not its interpretation, we could approximate each term in $\bar{E}_n \cdot \bar{J}_n^{(2)}$ and then accomplish the matrix multiplications in one step. However, we are interested in determining which terms cancel and the display of the details of the calculation makes this possible. Global transformations of the expression for $\bar{E}_n \cdot \bar{J}_n^{(2)}$, such as substitutions or approximations, can be applied to the expression as a whole using “boxes” to inhibit the combination of the physically identified terms. Local transformations on subexpressions can be effected by interactive commands such as SUBSPART which extracts a subexpression, operates upon it, and inserts it back into the original term. Also specific simplifications, such as matrix operations, can be controlled locally or globally by the setting of special variables called “flags” or “switches”. An example of a form resulting from interactive manipulations, is a reduced expression for $\bar{E}_n \cdot \bar{J}_n^{(2)}$ in the limit where $\bar{\sigma}$ becomes diagonal ($\bar{\sigma}I$):

(C9) EJTOT1:REMBOX(EJTOT1);

$$(D9) \quad \frac{\text{SIGMA}_A \left| \begin{matrix} E \\ A \end{matrix} \right| \text{SIGMA}_B \left| \begin{matrix} E \\ B \end{matrix} \right| \left| \begin{matrix} E \\ N \end{matrix} \right| \left(\frac{\left| \begin{matrix} K \\ A \end{matrix} \right| \left| \begin{matrix} K \\ B \end{matrix} \right| \left| \begin{matrix} K \\ N \end{matrix} \right|}{\left| \begin{matrix} K \\ A \end{matrix} \right| \left| \begin{matrix} K \\ B \end{matrix} \right| \omega_N} + \frac{\left| \begin{matrix} K \\ B \end{matrix} \right| \left(\begin{matrix} K \\ N \end{matrix} \cdot \begin{matrix} K \\ A \end{matrix} \right)}{\left| \begin{matrix} K \\ A \end{matrix} \right| \omega_B \left| \begin{matrix} K \\ N \end{matrix} \right|} + \frac{\left| \begin{matrix} K \\ A \end{matrix} \right| \left(\begin{matrix} K \\ N \end{matrix} \cdot \begin{matrix} K \\ B \end{matrix} \right)}{\omega_A \left| \begin{matrix} K \\ B \end{matrix} \right| \left| \begin{matrix} K \\ N \end{matrix} \right|} \right)}{2 \omega_N \bar{\sigma} I}$$

The command REMBOX removes all boxes from the expression EJTOT1.

The second order coupling between some three waves is characterized by a coupling coefficient proportional to $\bar{E}_n \cdot \bar{J}_n^{(2)}$ [see eq. (10)]. The general expression for $\bar{E}_n \cdot \bar{J}_n^{(2)}$ can be multiplied out by MACSYMA, but this voluminous result is essentially worthless since it is so unwieldy that it is impossible to obtain any significant physical interpretation from it. It is reassuring, however, to know that this expression is correct and that we have not left anything out which was present in our original model. If we now reduce this expression by introducing assumptions particular to a given coupling problem, we are confident that the solution obtained is as accurate as our assumptions, a non-trivial accomplishment for nonlinear problems. In practice, however, there is a computational saving to be gained by introducing our assumptions, in terms of Taylor series expansions of the elements of $\bar{\sigma}$, before the matrix multiplications are carried out.

Before attempting to reduce the general result for specific cases, we must find appropriate small parameters for the particular problem. We have often found our intuition, based on the linear behavior of plasmas, to be wrong concerning nonlinear effects, so a numerical evaluation of the general form of $\bar{E}_n \cdot \bar{J}_n^{(2)}$ is carried out to obtain some initial insight. Since the result of our general derivation is already in the computer memory, the numerical reduction requires only that we assign values for parameters (such as $\bar{B}_0, n_0, \bar{v}_{th}^2, \gamma, \omega_n, \bar{k}_n$, etc.) and invoke a simplification of the expression.

By taking advantage of the capabilities of the MACSYMA system for controlling numerical evaluation and displaying intermediate results, we can learn a considerable amount about the interactions being studied. For instance, by keeping track of the contributions from each nonlinearity [as indicated in eq. (18)] using the “boxing” technique, we can determine the source of the major contribution in the result immediately. Also, examination of the display of $\bar{E}_n \cdot \bar{J}_n^{(2)}$ after numerical evaluation of the elements of $\bar{\sigma}, \bar{E}$, and \bar{k} , but *before* the matrix multiplications are executed, allows us to see which elements of the matrix $\bar{\sigma}$ are dominant on the vectors \bar{k} and \bar{E} , and hence on particular terms of $\bar{J}^{(2)}$. Furthermore, we can observe the occurrence of cancellations between large terms which can be troublesome, both computationally and physically.

Below in (D34), is an example of the results of this numerical evaluation; only one term of (18) is shown, namely the convective term. The symbol “CONVEC” indicates that this is the convective term, while the symbols “TERM1” and “TERM2” serve to distinguish the term explicitly given in (18) from the term generated by exchanging a and b ($a \leftrightarrow b$). %I is the SQRT(-1).

```
(D34) 1.6342976E-8 CONVEC

      [ 3.2068121E-2 ]
      [ 2.3563655 %I ]
      [ - 9.2678274 ]
(( [ - 1.2610264, 1, 7.8953511E-2] . [ 3.2068121E-2 ]
                                     [ 2.3563655 %I ]
                                     [ - 9.2678274 ]

([1, 1, 0.2]
 [ - 2.5642208E-2      2.4695094E-2 + 1.0000477 %I  4.9390187E-3 - 5.215057 %I ]
 . [ 2.4695094E-2 - 1.0000477 %I      - 2.5642208E-2      4.9390187E-3 + 5.215057 %I ]
   [ 4.9390187E-3 + 5.215057 %I  4.9390187E-3 - 5.215057 %I      - 45.414237 ]
   [ - ( - 1.6291615E-2 + 1.0203168 %I) ]
   [ - (1.2919328E-2 + 1.2866464 %I) ])) TERM1
   [ - ( - 6.3610957 + 3.7252903E-9 %I) ]

+ ([1, 1, 0.2]
 [ - 2.5642208E-2      2.4695094E-2 + 1.0000477 %I  4.9390187E-3 - 5.215057 %I ]
 . [ 2.4695094E-2 - 1.0000477 %I      - 2.5642208E-2      4.9390187E-3 + 5.215057 %I ]
   [ 4.9390187E-3 + 5.215057 %I  4.9390187E-3 - 5.215057 %I      - 45.414237 ]
   [ 3.2068121E-2 ]
   [ 2.3563655 %I ])) ([2.2610264, 0, 0.12104649] . [ - ( - 1.6291615E-2 + 1.0203168 %I) ]
   [ - 9.2678274 ] [ - (1.2919328E-2 + 1.2866464 %I) ]
   [ - ( - 6.3610957 + 3.7252903E-9 %I) ]

* TERM2)
```

Note that from the above display we can see that the $\bar{\sigma}_{3,3}$ element of the $\bar{\sigma}$ matrix is dominant and that the imaginary parts of the off-diagonal terms are important (thus causing anisotropic effects). Other information can be obtained such as which components of \bar{E} or \bar{k} strongly affect the result and thereby see how the direction of these vectors affect the current, $\bar{J}^{(2)}$.

A final numerical result for one species, with all nonlinear terms included, illustrates the convenience of following particular terms from the original equations. In (C35)–(D35), the matrix multiplications are carried out, but the different terms are not allowed to be combined together and the real and imaginary parts of each term are labeled separately. Note especially the cancellations between the CONTIN (continuity) and FLOWCUR (current) terms. In the second command, (C38), the labels are removed by giving them a value of 1, permitting the various terms to be added.

```
(C35) ANSE:EV(CCELECEVN, DDALL, EXPAND, FLDAT);
(D35) 4.6080382E-8 PRESSURE
- 2.8912808E-17 %I PRESSURE + 1.6764616E-6 CONTIN TERM1 - 3.0319546E-6 %I CONTIN TERM1
+ 7.2842154E-7 CONVEC TERM1 - 2.223024E-6 %I CONVEC TERM1 - 1.6074017E-6 FLOWCUR TERM1
+ 2.9070568E-6 %I FLOWCUR TERM1 - 1.1494268E-6 CONTIN TERM2 + 1.4869468E-6 %I CONTIN TERM2
+ 1.1101501E-6 CONVEC TERM2 - 3.1740891E-6 %I CONVEC TERM2 + 1.1020775E-6 FLOWCUR TERM2
- 1.4256937E-6 %I FLOWCUR TERM2

(C38) ANSET:SUBST([TERM1=1, TERM2=1, CONVEC=1, FLOWCUR=1, PRESSURE=1, CONTIN=1], ANSE);
(D38) 1.9063626E-6 - 5.4607579E-6 %I
```

The ability to visualize matrix computations with the elements displayed in this manner is particularly important for anisotropic plasma models where the three-dimensional and nonlinear nature of the plasma make a firm understanding of the physical effects very challenging. Thus, for a complex interaction, preliminary numerical calculations

which allow one to visualize the detailed evolution of the exact fluid result will indicate appropriate approximations that can be made for obtaining approximate analytic results.

Once we have settled on a small parameter, whose validity must be checked post facto, we expand the elements of \bar{k} , \bar{E} , $\bar{\sigma}$, etc. in terms of this parameter. An example of substituting powerseries expansions for the elements of $\bar{\sigma}$ is shown below. A typical $\bar{\sigma}_n$ whose elements were expanded in a parameter DEL by MACSYMA is shown below. The symbol "SIGMA" is the computer representation of $\bar{\sigma}$. In this case DEL represents $\sqrt{m_e/m_i}$, WN_N is ω_n normalized, and LM3 is an expression involving constants. The appearance of WN_A in this expression is due to the elimination of \bar{k}_n using the dispersion equation, $D(\bar{k}_n, \omega_n)$. Once all of the $\bar{\sigma}$ values have been expanded this way, they are substituted into eq. (18) and the matrix multiplications are executed.

$$\text{SIGMA}_N = \begin{bmatrix}
 \left[\frac{1}{WN_N} + \frac{LM3^2 WN_A^2 DEL^2}{WN_N^3} + \dots \right] & \left[\frac{LM3 WN_A \%I DEL}{WN_N^2} \right] & 0 \\
 \left[\frac{LM3 WN_A \%I DEL}{WN_N^2} \right] & \left[\frac{1}{WN_N} + \frac{LM3^2 WN_A^2 DEL^2}{WN_N^3} + \dots \right] & 0 \\
 0 & 0 & \frac{1}{WN_N}
 \end{bmatrix}$$

The expression resulting from matrix multiplications is truncated to the appropriate power in the expansion parameter, DEL. A sample result from one case shows a dependence on two parameters of the problem, $ALP = \sin^{-1}(k_{nx}/|k_n|)$ and $BET = \tan^{-1}(k_{ny}/k_{nz})$, and the constant multiplier which gives the physical dimensions. This (D54) is the complete expression for $\bar{E}_n \cdot \bar{J}_n^{(2)}$ reduced to an understandable size. The second expression below (D57) is the same result with the parameter DEL expressed in known quantities and restructured so that multiplicative terms are grouped to the right.

```
(C54) MIDN:MIFLOW+M1CDNV;
TIME= 175 MSEC.
```

$$(D54) \quad - \frac{(1 - 2 \sin^2(ALP)) LM1^2 DEL^2 + 10 \cos(ALP) \sin(ALP) \sin(BET) LM3 LM1^2 \%I DEL^3}{2}$$

```
(C56) (DEL:1/SQRT(MU), LM1:KNM[A]*SQRT(MU)/WN[A], LM2:KNZ[A]/KNM[A]*SQRT(MU),
LM3:SQRT(MU)/WN[A], LM4:WN[A])$
```

```
TIME= 147 MSEC.
```

```
(C57) MIDN:EV(TBDX(CONSTFACT)*TBDX(MIDN), EVAL);
TIME= 1623 MSEC.
```

$$(D57) \quad \left(- \frac{(1 - 2 \sin^2(ALP)) KNM_A^2 WN_A^2 + 10 \cos(ALP) \sin(ALP) \sin(BET) KNM_A^2 \%I}{2 WN_A^3} \right) * \left(\frac{E_A E_B E_N^* EP_D WN^2 Q}{KNM_A KNM_B W_{CI} MI_{KNM} C_N S} \right)$$

This small result (D57) is the desired approximate analytic form which we can interpret by tracing its origin in the calculation and thus arrive at a simple model exhibiting only the relevant effects.

The above example (D34)–(D57) pertains to the nonlinear coupling of a lower hybrid wave (pump), $\omega_a \approx \omega_{pi}$, $\cos^2\theta \ll m_e/m_i$, to two ion-acoustic waves $\omega_{b,n} = k_{b,n}c_s$ with $\omega_b \approx \omega_n \approx \frac{1}{2}\omega_a$ and $|\bar{k}_{b,n}| \gg |\bar{k}_a|$ at elevation and azimuthal angles α (ALP) and β (BET) with respect to the magnetic field $\bar{\mathbf{B}}_0$. The numerical calculations assumed a highly magnetized hydrogen plasma with $(\omega_{pi}/\Omega_i = 10$, a pump field with $k_a = \Omega_i/2c_s$ and $\cos\theta = 10^{-2}$, and $k_{b,n} \approx 10k_a$. From these calculations we discover that although the individual contributions to the coupling are largest from the electron continuity and flow current nonlinearities, these cancel (see D35 and D38) to such an extent that the overall nonlinear coupling is essentially dominated by the ion convective and flow current nonlinearities. The physics behind the strong cancellations between continuity and flow current nonlinearities can be traced to the fact that the electron motion in the ion-acoustic wave is adiabatic. An approximate analytic derivation of the nonlinear coupling of a lower hybrid wave with an ion-acoustic wave, producing another ion-acoustic wave, can therefore be obtained by retaining only the nonlinearity due to the ions. Furthermore, for both types of waves involved, the ions may be approximately described by a cold fluid plasma model in zero magnetic field. This leads to the first term of M in D57, which can be traced as due to the pump induced ion motion parallel to the pump electric field. The second term in D57 is due to the finite ellipticity of the induced ion orbits when the magnetic field is taken into account, and was obtained by the use of the Taylor package described above. Thus, un-normalizing D57 we find

$$M \approx M_1 \approx C \left(\cos 2\alpha + i5 \frac{\Omega_i}{\omega_i} \sin 2\alpha \sin \beta \right), \quad (19)$$

where

$$C = \frac{q\epsilon_0\omega_{pi}^2}{2m_i} \frac{k_a}{\omega_a^2\omega_b\omega_n} E_a E_b E_n^*. \quad (20)$$

From this and the equations for the time-averaged energy densities of the unperturbed waves we can determine the nonlinear coupling constant \mathcal{K} of eq. (9), and finally the maximum growth rate of the pump-driven parametric excitation

$$\frac{\gamma_0}{\omega_n} \approx \frac{1}{4} \frac{q|\bar{E}_a|}{m_i\omega_a} \frac{k_a}{\omega_a} \left| \cos 2\alpha + i \frac{\Omega_i}{\omega_a} 5 \sin 2\alpha \sin \beta \right|. \quad (21)$$

Note that $(q|\bar{E}_a|/m_i\omega_a)$ is the induced ion velocity along the pump field, and that the interaction is proportional to the pump wavenumber, both consistent with the above derived physical picture of the interaction. We have thus achieved an approximate analytic description of a three-dimensional nonlinear wave interaction, together with a physical understanding of the dominant nonlinear mechanisms that give rise to the interaction. Several nonlinear wave interactions, described by the warm-fluid plasma model, have been studied successfully using the computation scheme outlined above [15].

Some general observations which can be made about our experience using the MACSYMA system are that

- (1) the system is very flexible in terms of allowing us to perform the calculation close to the way with which we are familiar and to return results in a usefully structured form;
- (2) the user of the system requires considerable experience in order to efficiently accomplish complicated calculations; and
- (3) symbolic computation is very useful for avoiding error and for repeating several similar calculations. This permits the exploration of several alternative physical hypotheses where previously the overhead would have been prohibitive. Furthermore, the two-dimensional display of results and the tracing of terms using labeled boxes significantly aided in the physical interpretation of the calculation.

It should be emphasized that certain parts of the calculation were completely automatic in the sense that only one command was required to invoke them, while other parts required several commands and much interaction.

For instance, once the equations, variables, and operators are described to the system (comprising about ten commands), the equations can be expanded, ordered, and transformed using only two commands. If the resulting equations are scalar algebraic, they could be solved with one further command. In our case, about 20 commands were required to solve the equations to second order by interactively moving terms around. Finally, the approximate reduced forms can be generated with two or three commands, although we chose a longer route in order to see the details of the calculation.

5. Concluding remarks

The MACSYMA system was made available over the ARPA network, to users outside MIT, in May, 1972. It is currently running a DECSYSTEM 1080 (PDP-10) machine and serving a consortium of users, including the CTR-plasma physics community. We at MIT have been among its largest users. Other large users (i.e. several hours of daily interactive use over a period of at least six months) have dealt with different problems: calculations involving tensors in general relativity; carrying out the required integrations in the finite element technique for solving p.d.e.s; developing techniques for improving the convergence of series; and others. Currently in plasma physics, J. Rome of ORNL is using MACSYMA to set up the equations of particle orbits in a rippled field Tokamak, and in a study of the radial Fokker–Planck equation. At LASL, Ralph Lewis is using MACSYMA to evaluate matrix elements that arise in a finite ion-gyroradius description of high-beta plasma stability. We have also recently used MACSYMA to implement the calculations of group velocity ray trajectories for rf fields in a Tokamak. In addition to these large uses of MACSYMA, there has emerged within our plasma theory group a daily and almost continuous use of the system in aiding various small analytic calculations. Currently available on MACSYMA is also an extensive capability for direct two- and three-dimensional plotting of functions and data. The use of a symbolic manipulation system as a “mathematical assistant” is sure to evolve as systems such as MACSYMA extend their capabilities in applied mathematics. It should be emphasized, however, that the large size and generality of a system such as MACSYMA has its problems. Although the system is now in a fairly stable state, there are occasionally bugs due to the interactions between modules, particularly when new ones are added, but these are fixed in short order by people who maintain the system. Another problem with such a large system is that it takes some time to learn all of its features, and particularly how to use them effectively for complex problems. These difficulties notwithstanding, our experience shows that symbolic computation can have an important impact in aiding the understanding and solution of a large variety of problems in plasma physics, in particular, and in science and engineering in general.

6. Acknowledgement

We would like to thank J. Moses and members of his MACSYMA group for advise and help in all aspects of this work. The encouragement by L.D. Smullin was important in starting our plasma theory–MACSYMA interaction.

7. Appendix: the MACSYMA system

The MACSYMA system utilizes four major internal representations: general, rational, power series, and Poisson series. The general representation is for expressing most types of mathematical formulae including vector and matrix operations, integrals, derivatives, and noncommutative products. An extensive general simplifier allows MACSYMA to be mathematically knowledgeable about manipulating and simplifying general expressions. The rational representation is designed for greater efficiency and offers a canonical representation needed in many algorithms (e.g. GCD). Several important capabilities such as rational simplification and factoring are implemented in this representation. The power series representation is used mostly in obtaining Taylor or Laurent expansions of a function about a point. The implementation of such expansions retains truncation information and allows conta-

gious truncations when combined with other expressions. The Poisson series representation is used for manipulating sums of trigonometric functions with polynomial or power series as coefficients.

A schematic diagram of the software packages in the MACSYMA system is shown in fig. A1. There are essentially six interconnected subsystems: language and interactive facilities; general representation; rational function representation; integration; power series; and MACLISP. The MACSYMA system is thus a very large program currently requiring 175 K words of memory on a PDP-10. It is implemented in the MACLISP programming language [21] which is a much-improved version of McCarthy's original LISP 1.5. It is currently running at the MIT Laboratory for Computer Science on a PDP-10 which has 512 K 36-bit words of memory, and on the MULTICS system, using a Honeywell 6180.

It should be noted that MACSYMA is a programming language in itself, with a syntax similar to ALGOL. Functions and programs for executing symbolic computations can be written in this language. The user rarely, if ever, has to resort to programming in the underlying LISP language. The following examples from MACSYMA in fig. A2 will illustrate its interactive operation and some of its capabilities.

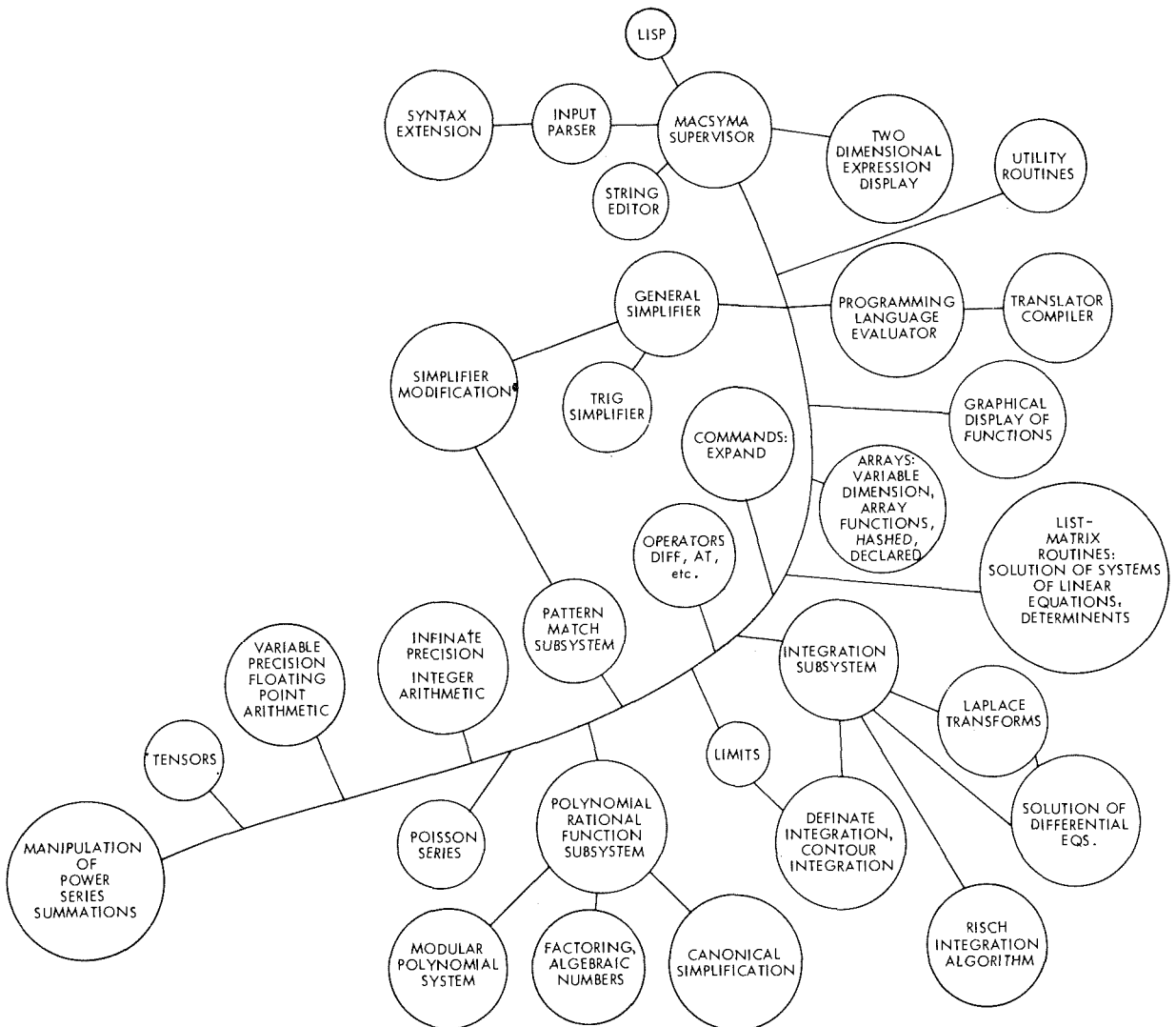


Fig. A1. The MACSYMA system software.

(C5) 1/(X**3+2);
TIME= 9 MSEC.

(D5)

$$\frac{1}{X^3 + 2}$$

(C6) INTEGRATE(%,X);

SIN FASL DSK MACSYM BEING LOADED
LOADING DONE

SCHATC FASL DSK MACSYM BEING LOADED
LOADING DONE
TOTALTIME= 1100 MSEC. GCTIME= 666 MSEC.

(D6)

$$-\frac{1/6 \operatorname{LOG}(X^2 - 2^{1/3} X + 2^{2/3})}{2^{2/3}} + \frac{\operatorname{ATAN}\left(\frac{2X - 2^{1/3}}{2^{1/3} \operatorname{SQRT}(3)}\right)}{2^{2/3} \operatorname{SQRT}(3)} + \frac{1/3 \operatorname{LOG}(X + 2^{1/3})}{2^{2/3}}$$

(C7) DIFF(%,X);
TIME= 397 MSEC.

(D7)

$$\frac{1/3}{1/3 (2X - 2^{1/3})^2} - \frac{1/6 (2X - 2^{1/3})}{2^{2/3} (X^2 - 2^{1/3} X + 2^{2/3})} + \frac{1/3}{2^{2/3} (X + 2^{1/3})} + 1$$

(C8) RATSIMP(%.);
TIME= 475 MSEC.

(D8)

$$\frac{1}{X^3 + 2}$$

Fig. A2(a). Example 1.

(C9) PROB:%E^X*SIN(X)^2;
TIME= 22 MSEC.

(D9)

$$X^2 \operatorname{E}^X \operatorname{SIN}(X)$$

(C16) INTEGRATE(PROB,X);

RISCH FASL DSK MACSYM BEING LOADED
LOADING DONE

TOTALTIME= 4263 MSEC. GCTIME= 983 MSEC.

(D16)

$$-1/10 X^2 (\operatorname{COS}(X) - \operatorname{SIN}(X)) - 2/5 X \operatorname{COS}(X) \operatorname{SIN}(X) + 1/2 X^2$$

(C17) TRIGREDUCE(%.);

TRIGRED FASL DSK MACSYM BEING LOADED
LOADING DONE

TOTALTIME= 914 MSEC. GCTIME= 351 MSEC.

(D17)

$$-1/5 X^2 \operatorname{SIN}(2X) - 1/10 X^2 \operatorname{COS}(2X) + 1/2 X^2$$

Fig. A2(b). Example 2.

(C19) %E^X*ERF(X);
TIME= 7 MSEC.

(D19)
$$e^X \operatorname{ERF}(X)$$

(C20) INTEGRATE(%,X);
TOTALTIME= 1576 MSEC. GCTIME= 355 MSEC.

(D20)
$$e^X \operatorname{ERF}(X) - e^{1/4} \operatorname{ERF}(X - 1/2)$$

(C21) DIFF(%,X);
TIME= 182 MSEC.

(D21)
$$e^X \operatorname{ERF}(X) + \frac{2 e^{X - X^2}}{\operatorname{SQRT}(\%PI)} - \frac{2 e^{1/4 - (X - 1/2)^2}}{\operatorname{SQRT}(\%PI)}$$

(C22) RADCAN(%);
TOTALTIME= 2114 MSEC. GCTIME= 356 MSEC.

(D22)
$$e^X \operatorname{ERF}(X)$$

Fig. A2(c). Example 3.

(C27) MATRIX((X^3,X^2,X,1),(Y^3,Y^2,Y,1),(Z^3,Z^2,Z,1),(W^3,W^2,W,1));
TIME= 31 MSEC.

(D27)
$$\begin{bmatrix} X^3 & X^2 & X & 1 \\ Y^3 & Y^2 & Y & 1 \\ Z^3 & Z^2 & Z & 1 \\ W^3 & W^2 & W & 1 \end{bmatrix}$$

(C28) DETERMINANT(%);
TIME= 302 MSEC.

(D28)
$$\begin{aligned} X^2 (-Y^2 (Z^3 - W^3) + W^2 Z^3 + Y^3 (Z^2 - W^2) - W^3 Z^2) - X^2 (-Y (Z^3 - W^3) + W Z^3 + Y^3 (Z - W) - W^3 Z) \\ - Y (W^2 Z^3 - W^3 Z^2) + Y^2 (W Z^3 - W^3 Z) + X^3 (-Y (Z^2 - W^2) + W Z^2 + Y^2 (Z - W) - W^2 Z) \\ - Y^3 (W Z^2 - W^2 Z) \end{aligned}$$

(C29) FACTOR(%);
TOTALTIME= 4917 MSEC. GCTIME= 1146 MSEC.

(D29)
$$- (W - X) (W - Y) (Y - X) (Z - W) (Z - X) (Z - Y)$$

Fig. A2(d). Example 4.

(C33) TAYLOR (COS (X) - SEC (X) , X, 0, 5);

HAYAT FASL_DSK MACSYM BEING LOADED
LOADING DDNE

EULBRN FASL_DSK MAXOUT BEING LOADED
LOADING DDNE

TOTALTIME= 59 MSEC. GCTIME= 778 MSEC.

(D33) /R/
$$- X^2 - 1/6 X^4 + \dots$$

(C34) TAYLOR ((COS (X) - SEC (X)) ^3, X, 0, 5);
TIME= 62 MSEC.

(D34) /R/
$$0$$

(C35) TAYLOR ((COS (X) - SEC (X)) ^-3, X, 0, 5);
TIME= 272 MSEC.

(D35) /R/
$$- \frac{1}{X^6} + \frac{1/2}{X^4} + \frac{11/120}{X^2} - 347/15120 - 6767/604800 X^2 - 15377/7983360 X^4 + \dots$$

(C36) TAYLOR (SQRT (1 - K^2 * SIN (X) ^2) , X, 0, 6);
TIME= 275 MSEC.

(D36) /R/
$$1 - 1/2 K^2 X^2 - 1/24 (3 K^4 - 4 K^2) X^4 - 1/720 (45 K^6 - 60 K^4 + 16 K^2) X^6 + \dots$$

(C37) D33 * D36;
TIME= 29 MSEC.

(D37) /R/
$$- X^2 + 1/6 (-1 + 3 K^2) X^4 + \dots$$

Fig. A2(e). Example 5.

Referring to fig. A2, one should note that each (C) line is a typed-in command by the user, and each (D) line is the output displayed by MACSYMA. The time it took MACSYMA to execute the command, i.e. compute the result, and some of the software packages that it had to load are also displayed. (Periodically MACSYMA takes time to clear memory space and shows this as GCTIME.) Thus in (C5) the expression $1/(x^3 + 2)$ is typed by the user; it takes 9 msec for MACSYMA to process it; it then displays it in (D5).

Line (C6) shows the command for integrating (D5) with respect to x ; the result is given in (D6). In (C7) MACSYMA is asked to differentiate (D6); it does it term by term, and displays it in (D7). Command (C8) asks for simplification of the rational functions in (D7); the result is given in (D8), which is the same expression we started with in (C5), as it should be.

In line (C9) we enter the function $e^x \sin^2 x$ and name it PROB so that it can be referred to later. The integrated result, (D16), is operated on, in (C17), by a command for simplifying trigonometric functions, giving (D17).

In lines (C19)–(D22) a sequence similar to the first two illustrates the handling of a special function, the error function. In this case yet another simplification package, RADCAN, is used in (C22) to obtain (D22).

In (C27) is shown the typing-in of a matrix. There is a command for taking the determinant of a symbolic matrix, (C28); the result can be factored as shown by (C29)–(D29).

The Taylor series of $(\cos x - \sec x)$ in x about $x = 0$ is shown in (C33)–(D33). When this function is cubed (C34), its Taylor series has no fifth order term (D34). If the same is asked for this function to the (-3) power (C35), MACSYMA returns the Laurant expansion (D35). Lines (C36) and (D36) show that the Taylor expansions can have symbolic polynomial coefficients. Lines (C37) and (D37) show the taking of the product of two Taylor series and illustrate how truncation is contagious.

The last example shows that special functions, like the plasma dispersion function, $Z(\xi)$, and their properties have been integrated into MACSYMA and can be manipulated. In (C38) the plasma wave dispersion relation containing $Z(\omega/kv_t)$ is typed in; its form for large arguments of $\xi = \omega/kv_t$ is obtained via (C39); and a numerical evaluation for $\xi = 0.1 + i0.5$ is obtained by (C40).

(C38) EPSILON: 1 - 'W[PE]^2 / (K^2) / (V[TH]^2) * ZETA (W/K/V[TH]);

(D38)
$$1 - \frac{W^2 \text{ZETA}\left(\frac{W}{K V_{TH}}\right)}{K^2 V_{TH}^2}$$

(C39) EPSILON, ZETALARGE: TRUE, MAXZ: 3, SIMPSUM: TRUE, EVAL;

(D39)
$$1 - \left(\frac{W^2}{PE} \left(\%I \text{ SQRT}(\%PI) \%E - \frac{4 K V_{TH}^2}{3 W} + \frac{2 K^3 V_{TH}^3}{W^3} - \frac{K^5 V_{TH}^5}{W^5} - \frac{K^7 V_{TH}^7}{2 W^7} \right) \right) / (K^2 V_{TH}^2)$$

(C40) EPSILON, W/K/V[TH] = 0.1 + %I * 0.5, NUMER;

(D40)
$$1 - \frac{(-0.090670332 + 1.09160396 \%I) W^2}{PE K^2 V_{TH}^2}$$

Fig. A2(f). Example 6.

References

- [1] D. Barton and J.P. Fitch, Rep. Progr. Phys. 35 (3) (1972) 235.
- [2] S.R. Petrick (Ed.), Proc. Second Symposium on Symbolic and Algebraic Manipulation (Association for Computing Machinery – SIGSAM, New York, 1971). Papers referring to the MACSYMA system begin on pages 59, 78, 282, 305, 311, 427 and 458.
- [3] Proceedings of EUROSAM '74 SIGSAM Bull. 8 (3) Aug. (1974) (issue November 31). MACSYMA is described in the paper by J. Moses on pp. 105–110.
- [4] MACSYMA Reference Manual, Project MAC, M.I.T., Nov. (1975).
- [5] K.V. Roberts and J.P. Boris, J. Comp. Phys. 8 (1971) 83.
- [6] M. Petravich, G. Kuo-Petravich, and K.V. Roberts, J. Comp. Phys. 10 (1072) 503.
- [7] G. Küppers, D. Pfirsch and W. Tasso, Proceedings of the Fourth International Conference on Plasma Physics and Controlled Nuclear Fusion Research at Madison, Vol. 2 (IAEA, Vienna, 1971) 529.
- [8] W. Kerner and J. Steuerwald, Comput. Phys. Commun. 9 (1975) 337.
- [9] B. Rosen, J. Comp. Phys. 15 (1974) 98 and 20 (1976) 22.
- [10] B. Rosen and M. Okabayashi, Nucl. Fusion 13 (1973) 3.
- [11] A. Bers, J.L. Kulp and D.C. Watson, Bull. Amer. Phys. Soc., Series II, 17 (11) (1972) 991.
- [12] A. Bers, J.L. Kulp and D.C. Watson, Quarterly Progress Report No. 108, Research Laboratory of Electronics, M.I.T. (1973) 167.
- [13] A. Bers, J.L. Kulp and D.C. Watson, Book of Abstracts – International Congress on Waves and Instabilities (Institute of Theoretical Physics, Innsbruck University, Innsbruck, Austria (1973) 13.
- [14] J.L. Kulp, S.M. Thesis, Department of Electrical Engineering, MIT (1973).
- [15] C.F.F. Karney, S.M. Thesis, Department of Electrical Engineering, MIT (1974).
- [16] J.L. Kulp, A. Bers and J. Moses, Proceedings of the Sixth Conference on Numerical Simulation of Plasmas (Lawrence Berkeley Laboratory, University of California at Berkeley, California (1973) p. 64.
- [17] J.L. Kulp, Proceedings of the ACM '74 Conference, Vol. II (Association for Computing Machinery, New York, 1974) p. 585.
- [18] J.L. Kulp, C.F.F. Karney and A. Bers, Research Laboratory of Electronics and Laboratory for Computer Science, MIT, Report in preparation (1976).
- [19] A. Bers, in: Plasma Physics – Les Houches 1972 (Gordon and Breach Publishers, New York–London–Paris, 1975) p. 183.
- [20] A. Bers, D.J. Kaup and A.H. Reiman, Phys. Rev. Lett. 37 (1976) 182; and Plasma Research Report 76/5, R.L.E., MIT Cambridge, Jan. (1976).
- [21] D.A. Moon, MACLISP Reference Manual, Project MAC, MIT (1974).