Symbolic Vector Analysis in Plasma Physics

H. Qin, W. M. Tang, and G. Rewoldt

Princeton Plasma Physics Laboratory, Princeton University, Princeton, NJ, 08543-451

Many problems in plasma physics involve substantial amounts of analytical vector calculation. The complexity usually originates from both the vector operations themselves and the choice of underlying coordinate system. A computer algebra package for symbolic vector analysis in general coordinate systems, GeneralVectorAnalysis (GVA), is developed using *Mathematica*. The modern viewpoint for 3D vector calculus, differential forms on 3-manifolds, is adopted to unify and systematize the vector calculus operations in general coordinate systems. This package will benefit physicists and applied mathematicians in their research where complicated vector analysis is required. It will not only save a huge amount of human brain-power and dramatically improve accuracy, but this package will also be an intelligent tool to assist researchers in finding the right approaches to their problems. Several applications of this symbolic vector analysis package to plasma physics are also given.

Key words: symbolic computation, vector analysis, plasma physics, *Mathematica*

1 Introduction

The analytical calculations using vector calculus that appear in plasma physics, fluid dynamics, and other fields sometimes can become extremely complex. Even though there are no difficulties in principle in performing these complicated calculations by human brain-power, practically it is often too involved to get the right result in an affordable amount of time.

The complexity usually originates from both the vector operations themselves and the choice of underlying coordinate system. An example would be the linearized ideal MHD (magnetohydrodynamics) motion equation for a magnetized plasma[1]:

$$-\omega^2 \rho \xi = \mathbf{F}(\xi), \tag{1}$$

where the linear operator $\mathbf{F}$ acting on a displacement vector ξ is given by:

$$\mathbf{F}(\xi) = \nabla(\gamma p_0 \nabla \cdot \xi + \xi \cdot \nabla p_0) + \frac{1}{4\pi}(\nabla \times \mathbf{Q}) \times \mathbf{B}_0 + \frac{1}{4\pi}(\nabla \times \mathbf{B}_0) \times \mathbf{Q}, \tag{2}$$

and

$$\mathbf{Q} = \nabla \times (\xi \times \mathbf{B}_0). \tag{3}$$

$\mathbf{B}_0$ here is a given magnetic field, and p is a scalar pressure. They both are functions of spatial position. The spectrum of $\mathbf{F}$ determines a wide range of physical phenomena, from the waves propagating in interplanetary space to the instabilities of fusion devices. In the simplest situation where $\mathbf{B}_0$ is uniform, the computation required to obtain the spectrum of operator $\mathbf{F}$ is already considerable. For a general $\mathbf{B}_0$ field, the number of terms involved in the process of obtaining the expression for $\mathbf{F}$ can be of the order of 500. In a realistic inhomogeneous $\mathbf{B}_0$ field such as that in a fusion device like a tokamak, the task of finding the spectrum is too herculean to be analytically doable when the underlying coordinate system is chosen to be either flux coordinates or Shafranov coordinates. This much needed calculation has never been done analytically, not because it is analytically impossible, but rather because it is too algebraically involved. Even for the simpler case where the magnetic flux surfaces are assumed to be circular and concentric, the only available result is the instability criterion analysis using the energy principle by Bussac *et al*[4] in 1975. More complicated vector calculus computations can be found in plasma kinetic theory.

However, in most cases the required analytic calculation can be finished in a finite number of steps through a prescribed process. The total number of terms and basic computations needed could be as large as 10,000, yet it is still finite. It is this very fact that suggests the possibility of performing these complicated symbolic calculations automatically on computers.

The ideal of performing non-numerical scientific computations using computers is not new at all. However only in recent years has this field gained attention in the scientific community. A wide variety of applications have been found in biology, chemistry, and physics. Important applications in physics include tensor calculations in general relativity and the evaluation of Feyman diagrams. In plasma physics, computer algebra made its debut in the analytical formalism of the PEST code[5]. Compared with scientific numerical computation, scientific symbolic computation is still in a preliminary phase. More and more new applications are expected as advanced system become available and efficient algorithms are discovered.

To implement automatic symbolic vector analysis in general coordinate systems, we need a simple and systematic mathematical framework. The modern viewpoint of 3D vector calculus, differential forms on 3-manifolds, is utilized for this purpose. On the other hand, a well-developed high level programming language with a symbolic computation capability is also necessary. To the end, we chose *Mathematica* by the Wolfram Research Inc.[2]

We have successfully developed the GeneralVectorAnalysis (GVA) *Mathematica* package, and used it in the analytical derivation of plasma gyro-kinetic-MHD theory. Compared with the `Calculus'VectorAnalysis'` package provided by Wolfram Research as a standard add on package, our package GVA is more advanced and efficient. The standard `Calculus'VectorAnalysis'` package works only on 14 standard right-handed orthogonal coordinate systems, while our package GVA works on any mathematically well defined coordinate system, including all the standard right-handed orthogonal 3D coordinate systems as well as on non-standard, non-orthogonal coordinate systems such as the magnetic flux coordinate system used extensively in fusion calculations. Moreover, users can define their own coordinate systems to work on. This unique ability will be appreciated by researchers in different fields where non-standard coordinate systems are needed. Since mathematically a modern viewpoint is used, the algorithm inside GVA is also more general and efficient. Other features like the asymptotic analysis capability also make this package more applicable to realistic physical and engineering problems.

The main difference between our package GVA and the `Calculus'VectorAnalysis'` package by Wolfram Research is the methodology. Our methodology is to look at the problem from a higher viewpoint; thus a systematic treatment is possible. This philosophy can be found in the core of *Mathematica*[2], for example, in the function for indefinite integration: `Integrate[ ]`, and the function for algebraic factoring: `Factor[ ]`. But such a methodology is not adopted in the add-on package `Calculus'VectorAnalysis'`. Modern concepts and systematic treatment inside the GVA package provide us with more freedom and new utility, by which research time can be saved, accuracy can be improved, analytic derivations otherwise impossible can be carried out, and thus researchers are able to concentrate on the meaningful physical results of their problems.

In Section 2, the basic mathematical formulas are summarized in the framework of differential forms on a 3-manifold. The advantages of the modern viewpoint will also be discussed. As an example, we show that the existence of the well-known Clebsch magnetic coordinates is a direct conclusion of the Darboux theorem in the theory of differential forms. The realization of the basic algorithm using *Mathematica* is discussed in Section 3. In Section 4 several applications of this GVA package to plasma physics are given.

2 The mathematics of vector calculus in general coordinate systems and its application to magnetic coordinates

In this section we briefly summarize the theory of vector calculus using the language of differential forms on 3-manifold[6,7]. Basics vector calculus formulas are derived in this framework. As an example of the many applications to physics, we show the connection between the Clebsch magnetic coordinates[8] and the Darboux theorem[9].

A vector field space is isomorphic with a 1-form space (i.e. its dual space, or T_0^1 tensor space) after introducing the Riemann metric tensor. In a 3-manifold, the Hodge star operator maps a 1-form space one-one onto a 2-form space, and a 0-form space one-one onto a 3-form space. If a vector field is viewed as a 1-form, and a scalar field as a 0-form, then all the elementary vector calculus operations can be expressed in terms of the exterior product and the exterior derivative of differential forms.

Actually all physical vector fields can more naturally be treated as 1-forms. Let's demonstrate this idea by considering the magnetic field $\mathbf{B}$, for example. All the information we can get about the $\mathbf{B}$ field is from first-hand experimental measurements. What we can directly measure is the strength of $\mathbf{B}$ in any given direction. Usually the strengths along different directions are different. In another words, what we can really measure as the $\mathbf{B}$ field is nothing but a function of the spatial direction. We can confirm that this function is linear from the measurement data. Therefore, $\mathbf{B}$ is indeed a 1-form. Only after the introduction of a metric (the Riemann metric tensor), can this 1-form be identified with a vector field[3].

In R^3, a well-defined coordinate system is given by three independent scalar functions, $x^1(\mathbf{r})$, $x^2(\mathbf{r})$, and $x^3(\mathbf{r})$ with $\nabla x^1 \cdot (\nabla x^2 \times \nabla x^3)$ non-vanishing everywhere. A coordinate system is uniquely determined by its Riemann metric matrix g_{ij}, or the inverse of the Riemann metric matrix g^{ij}, defined by:

$$g_{ij} \equiv \mathbf{e}_i \cdot \mathbf{e}_j, \tag{4}$$

$$g^{ij} = \mathbf{e}^i \cdot \mathbf{e}^j = \nabla x^i \cdot \nabla x^j, \tag{5}$$

where the $\mathbf{e}_i$ and $\mathbf{e}^i$ are the basis and the dual basis respectively.

A vector field is specified by $(a_1(x), a_2(x), a_3(x))$. In the form of a 1-form:

$$A = a_1 dx^1 + a_2 dx^2 + a_3 dx^3 \tag{6}$$

The Hodge star operator maps it into a 2-form,

$$*A = J(a^3 dx^1 dx^2 + a^1 dx^2 dx^3 + a^2 dx^3 dx^1), \tag{7}$$

where $J = \sqrt{det|g_{ij}|}$, $a^i = g^{ij} a_j$, $i, j = 1, 2, 3$. The image of $*A$ under the Hodge star operator is A itself, *i.e.*,

$$* * A = A. \tag{8}$$

A scalar field f is viewed as a 0-form, on which the Hodge star operator is:

$$*f = J f dx^1 dx^2 dx^3. \tag{9}$$

Also, we have

$$* * f = f. \tag{10}$$

The 1-form of a vector field is its covariant representation; on the other hand the 2-form of the same vector field is its contravariant representation. Similarly, the 0-form of a scalar field is it covariant representation while the 3-form of it is its contravariant representation.

The correspondence between vector calculus operations and differential forms operations can be summarized as following:

$$\begin{aligned}
A \times B \Longleftrightarrow & \quad *(AB) & \Longrightarrow (A \times B)^k = (\epsilon_{kij} A_i B_j)/J, && (11)\\
A \cdot B \Longleftrightarrow & *(A(*B)) & \Longrightarrow (A \cdot B) = A_i B^j, && (12)\\
\nabla f \Longleftrightarrow & \quad *df & \Longrightarrow (\nabla f)_i = \partial f/\partial x^i, && (13)\\
\nabla \cdot A \Longleftrightarrow & \quad *d(*A) & \Longrightarrow \nabla \cdot A = (\partial J A^i/\partial x^i)/J, && (14)\\
\nabla \times A \Longleftrightarrow & \quad *dA & \Longrightarrow (\nabla \times A)^k = (\epsilon_{kij} \partial A^j/\partial x^i)/J && (15)
\end{aligned}$$

The third column in the above equations constitutes the basic formulas which will appear in the GVA package. The derivations of these equations are trivial from the corresponding differential forms representations in the second column. The advantage of the differential forms formalism for 3D vector calculus is obvious by comparing the derivations of the basic formulas given here and the derivations which do not using this modern technique[10].

The differential form formalism of 3D vector calculus also makes it possible to perform coordinate independent vector analysis using the computer, because the basic vector calculus identities can be unified and systemized in the language of differential forms. As we know:

$$\nabla(fg) = f\nabla g + g\nabla f, \tag{16}$$
$$\nabla \cdot (f\mathbf{A}) = f\nabla \cdot \mathbf{A} + \mathbf{A} \cdot \nabla f, \tag{17}$$
$$\nabla \times (f\mathbf{A}) = f\nabla \times \mathbf{A} + \nabla f \times A, \tag{18}$$

and

$$\nabla \cdot (\mathbf{A} \times \mathbf{B}) = \mathbf{B} \cdot \nabla \times \mathbf{A} - \mathbf{A} \cdot \nabla \times \mathbf{B}, \tag{19}$$

are nothing but the chain rule for differential forms:

$$d(\omega\theta) = (d\omega)\theta + (-1)^{deg\ \omega}\omega d\theta; \tag{20}$$

The formulas $\nabla \cdot \nabla \times \mathbf{A} = 0$ and $\nabla \times (\nabla f) = 0$ are nothing but $dd\omega = 0$.

There is an interesting application of the differential form theory in the construction of magnetic coordinate systems. We prove the existence of the Clebsch magnetic coordinates as following: because $*B$ is closed (i.e. $d(*B) = 0$), in a starshaped region there exists an A such that $*B = dA$ (Poincare lemma). According to the Darboux theorem, there exist three independent scalar functions α, β, and γ such that $A = \alpha d\beta + \gamma$, and therefore $*B = d\alpha d\beta$; α and β are thus the Clebsch coordinates for the magnetic field. Similar arguments can be applied to other magnetic coordinate systems including flux coordinates, Boozer-Grad coordinates, and Hadam coordinates[11,10].

3 The realization of automatic symbolic analysis in general coordinate systems in the context of *Mathematica*

In the context of *Mathematica*, the realization of symbolic vector analysis in general coordinate systems is a *Mathematica* add-on package, GeneralVectorAnalysis(GVA), which provides users with about 20 functions to perform vector operations.

We have for the first time built the 2D conventional vector analysis notation into our system. For examples, instead of calling the function `Curl[A]`, we can use `∇ × A`; to call the function `CovariantDerivative[A, B]`, we can just use `(A · ∇)B`. When the `CoordinateSystem` is set to `''None''`, GVA will perform coordinate independent vector analysis. `VectorExpand[ ]` will expand a vector expression into the canonical form, using all the known vector identities (see Figure 1). A coordinate system is defined by the inverse Riemann metric matrix. Frequently used coordinate systems such as the Cartesian coordinate system, the cylindrical coordinate

■ Automatic symbolic vector analysis (demo 1)

Load in the GenerelVectorAnalysis package

```
In[1]:= << ~/mathematica/GeneralVectorAnalysis.m
```

Conventional vector analysis notation is built in.

```
In[2]:= FullForm/@{ a·(b + c), (a + d)×(b×c)}
Out[2]= {DotProduct[a, Plus[b, c]], CrossProduct[Plus[a, d], CrossProduct[b, c]]}

In[3]:= FullForm/@{∇×(a + a×(b×c)), ∇·(a + b×d)}
Out[3]= {Curl[Plus[a, CrossProduct[a, CrossProduct[b, c]]]],
          Div[Plus[a, CrossProduct[b, d]]]}

In[4]:= FullForm/@{∇² (f g), Â, |B |, 𝒥[r]}
Out[4]= {Laplacian[Times[f, g]], UnitVector[A], AbsoluteValue[B], Jacobian[r]}
```

Vector calculations independent of coordinate system.

```
In[5]:= DeclareVector[A, B0, B, C, D, E, F, G, H, J, ξ, Q];
        DeclareScalar[a, b, c, d, e, f, g, h]
In[6]:= {A×A, A·(A×C), ∇×(∇f), ∇·(∇×A), VectorExpand[A×(B×C)]}
Out[6]= {0, 0, 0, 0, −C A·B + B A·C}

In[7]:= VectorExpand/@{A×(B×C) + B×(C×A) + C×(A×B), (A×B)×(C×D),
            ∇·(f (A×B + ∇×C))}
Out[7]= {0, −D A·(B×C) + C A·(B×D),
          −f A·(∇×B) + f B·(∇×A) + (A×B)·(∇f) + (∇×C)·(∇f)}
```

An example from ideal megnetohydrodynamics

```
In[8]:= ∇·(B0) = 0;  Q = ∇×(ξ×B0);  J = ∇×Q
Out[8]= ∇×(∇×(ξ×(B0)))

In[9]:= VectorExpand[J]
Out[9]= −(∇(∇·ξ))×(B0) − ∇×((ξ·∇) (B0)) + ∇×(((B0)·∇) ξ) − ∇×(B0) ∇·ξ
```

Fig. 1. *Mathematica* note book GVA.demo1.nb.

system, and the spherical coordinate system are built into GVA. Users can easily add their own coordinate system by adding in the corresponding inverse Riemann metric matrix. A coordinate system has to be chosen before any vector operations can be correctly performed. To choose a coordinate system, use the function `SetCoordinateSystem[ ]`. For example,

```
SetCoordinateSystem[''Cartesian''][x,y,z]
```

chooses the Cartesian coordinates, and uses x, y, and z as the three coordinates.

Asymptotic analysis is supported in GVA. The Riemann metric matrix can be a series ex-

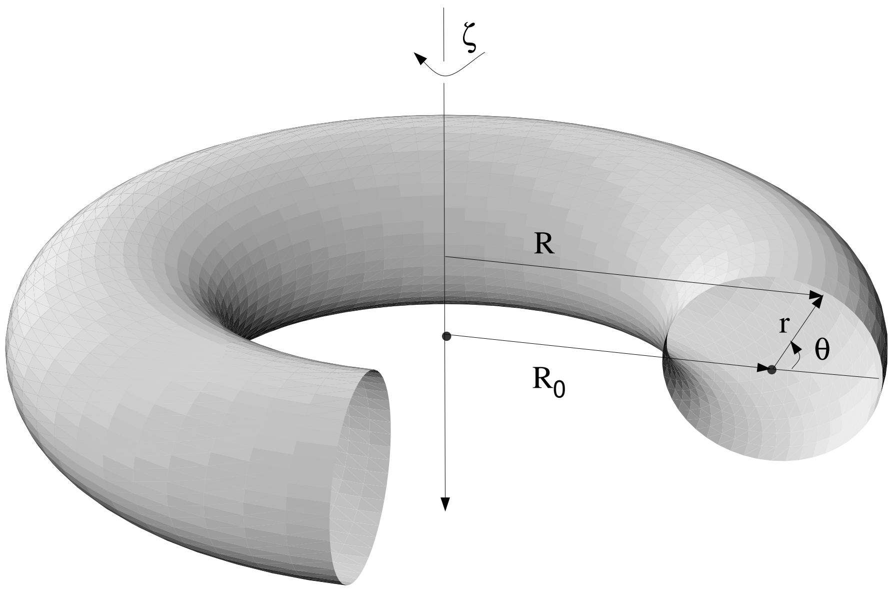

Fig. 2. Circular concentric tokamak coordinate system.

pansion to any order. All the vector operations will consequently be carried out to the same order. This is an extremely powerful feature of this package. In practice, the coordinate systems could be complicated, and asymptotic treatment is often required. This is what happens in the coordinate system for fusion devices like tokamaks. For the purpose of applications to fusion plasma physics, the "straight" tokamak coordinate system and several other tokamak coordinate systems, are built in as well. We can use

`SetCoordinateSystem[''TKCircular'',`R_0, ϵ`][r,`θ, ζ`]`

to choose the conventional coordinate system for a large aspect-ratio tokamak with circular, concentric flux surfaces as shown in Figure 2. The parameter ϵ will be used as the small asymptotic expansion parameter. All vector operations can be carried out to a chosen order of ϵ. In the GVA package, a vector object has the form `Vector`$[\{\mathtt{a}_1, \mathtt{a}_2, \mathtt{a}_3\}, \{\mathtt{a}^1, \mathtt{a}^2, \mathtt{a}^3\}]$. $\{\mathtt{a}_1, \mathtt{a}_2, \mathtt{a}_3\}$ are the covariant components, and $\{\mathtt{a}^1, \mathtt{a}^2, \mathtt{a}^3\}$ are the contravariant ones. Since a vector is not a simple scalar function, it must be created by the `DefineVector[ ]` function, or as a result of vector operations. The following two functions first set up the cylindrical coordinate system and create a vector `A` using its covariant components:

SetCoordinateSystem[''Cylindrical''][r,θ,z],

$\mathtt{A}[\mathtt{r}, \theta, \mathtt{z}] = \mathtt{DefineVector}[1, \mathtt{A}_\mathtt{r}[\mathtt{r}, \theta, \mathtt{z}], \mathtt{A}_\theta[\mathtt{r}, \theta, \mathtt{z}], \mathtt{A}_\mathtt{z}[\mathtt{r}, \theta, \mathtt{z}]]$.

To create a vector using its contravariant components, we use the constant 2 as the first argument of DefineVector[ ] instead:

$\mathtt{B}[\mathtt{r}, \theta, \mathtt{z}] = \mathtt{DefineVector}[2, \mathtt{B}_\mathtt{r}[\mathtt{r}, \theta, \mathtt{z}], \mathtt{B}_\theta[\mathtt{r}, \theta, \mathtt{z}], \mathtt{B}_\mathtt{z}[\mathtt{r}, \theta, \mathtt{z}]]$.

The vector operation functions of GVA are easy to use. To calculate the cross product of A and B and name the result as C, for instance, we execute:

$\mathtt{C}[\mathtt{r}, \theta, \mathtt{z}] = \mathtt{A}[\mathtt{r}, \theta, \mathtt{z}] \times \mathtt{B}[\mathtt{r}, \theta, \mathtt{z}]$.

The command

$\mathtt{D}[\mathtt{r}, \theta, \mathtt{z}] = \mathtt{A}[\mathtt{r}, \theta, \mathtt{z}] \times \nabla \times \mathtt{B}[\mathtt{r}, \theta, \mathtt{z}]$

gives the result of the cross product between A and $\nabla \times$ B as D.

The elementary functions provided by GVA are listed below:

- SetCoordinateSystem. SetCoordinateSystem[''Coordsys'',p1,p2][c1,c2,c3] will set up a Coordsys coordinate with coordinates c1,c2,c3 and parameters p1,p2.
- Metric. Metric[1][c1,c2,c3] is the metric matrix $\mathtt{g}_{\mathtt{ij}}$, Metric[2][c1,c2,c3] is the metric matrix $\mathtt{g}^{\mathtt{ij}}$.
- Jacobian. Jacobian[c1,c2,c3] gives the Jacobian.
- Christoffel. Christoffel[1][c1,c2,c3] is the Christoffel symbol of the 1st kind, Christoffel[2][c1,c2,c3] is the Christoffel symbol of the 2nd kind.
- DefineVector. DefineVector[1,A1,A2,A3] defines a vector with A1, A2, A3 as its covariant components, DefineVector[2,A1,A2,A3] defines a vector with A1, A2, A3 as its contravariant components.
- DotProduct. DotProduct[A,B] or $\mathtt{A} \cdot \mathtt{B}$ gives the dot product.
- CrossProduct. CrossProduct[A,B] or $\mathtt{A} \times \mathtt{B}$ gives the cross product.
- Grad. Grad[f] or ∇f gives the gradient of the scalar f.
- Div. Div[A] or $\nabla \cdot \mathtt{A}$ gives the divergence of the vector A.
- Curl. Curl[A] or $\nabla \times \mathtt{A}$ gives the curl of the vector A.
- CovariantDerivative. CovariantDerivative[A,x] or $(\mathtt{A} \cdot \nabla)\mathtt{x}$ gives the directional derivative of x in the A direction. x is either a scalar or a vector.
- Laplacian. Laplacian[f] or $\nabla^2 f$ gives the Laplacian of the scalar f.
- AbsoluteValue. AbsoluteValue[A] or $|A\,|$ gives the absolute value of the vector A.

- `UnitVector`. `UnitVector[A]` or $\hat{A}$ gives the normalized vector in the direction of `A`.
- `Parallel`. `Parallel[A, B]` gives the projection of `A` in the direction of `B`. It returns a scalar.
- `Perp`. `Perp[A,B]` gives the perpendicular components of `A` with respect to `B`. It returns a vector.
- `VectorExpand`. `VectorExpand[A]` expands the vector expression `A` into canonical form.

4 Applications to plasma physics

A lot of problems in theoretical plasma physics require substantial amount of vector analysis, as in the example discussed in Section 1. In many situations, the real physics is camouflaged under the drudgery work of algebra. What makes things worse is that quite often we have to make some assumptions to simplify the algebra, with the risk of losing some important physical effects. We employ three examples here to demonstrate how the GVA package can be used to facilitate tedious algebraic derivations such that we can focus directly on the underlying physics.

4.1 Application I: ideal MHD waves in a homogeneous magnetized plasma

As the simplest application of the GVA package, we examine the ideal MHD waves in a homogeneous magnetized plasma. With the help of GVA, the things we need to do are simply to set up a coordinate system and field variables, and to define the linear force operator, as is done in the *Mathematica* note book file wave.nb. The rest of the file is pretty straightforward (See Figure 3). It is easy to see that there are three eigenmodes; the first one is the shear Alfven wave, the second one is the slow magnetosonic wave, and the last one is the fast magnetosonic wave.

4.2 Application II: particle drift motions in tokamaks

In a large aspect ratio tokamak, the inverse aspect ratio ϵ is usually used as the small parameter for asymptotic approximation. The capability of asymptotic analysis is built into the GVA package. As a demonstration, we calculate the particle drift velocity in tokamaks to the order $O(\epsilon^2)$. Again, the things we need to do are simple: loading in the GVA package, setting up the coordinate system and **B** field, and defining the drift velocity. This is all done

■ Ideal MHD Waves in a Magnetized Plasma

Load in the GeneralVectorAnalysis package, set up coordinate system and field variables

In[1]:= << ~/mathematica/GeneralVectorAnalysis.m
SetCoordinateSystem["Cartesian"][x, y, z];

In[3]:= B_0 = DefineVector[1, B_X, B_Y, B_Z];

In[4]:= ξ[x, y, z] = Exp[I (K_X x + K_Y y + K_Z z)] DefineVector[1, ξ_X, ξ_Y, ξ_Z];

Define the linear force operator:

In[5]:= $F[\xi\_] := \Big(Q = \nabla\times(\xi\times B_0);$

$$\frac{(\nabla\times Q)\times(B_0)}{4\pi} + \frac{(\nabla\times(B_0))\times Q}{4\pi} + \nabla(\xi\cdot\nabla p + \gamma\, p\,\nabla\cdot\xi)\Big)$$

Create the matrix for the force operator, solve for the eigenvalues :

In[6]:= force = Expand$\left[-\frac{F[\xi[x, y, z]]}{\rho\, \mathrm{Exp}[I\,(K_X\, x + K_Y\, y + K_Z\, z)]}\right]$;

In[7]:= matrix =
Table[Coefficient[force[[1, i]], Switch[j, 1, ξ_X, 2, ξ_Y, 3, ξ_Z]], {i, 3}, {j, 3}];

In[8]:= eigenfreq = FullSimplify[Expand[Eigenvalues[matrix]]]

Out[8]=
$$\Big\{\frac{(B_X K_X + B_Y K_Y + B_Z K_Z)^2}{4\pi\rho},$$
$$\frac{1}{8\pi\rho^2}\Big(\rho\,(4\,p\,\pi\,\gamma + B_X^2 + B_Y^2 + B_Z^2)\,(K_X^2 + K_Y^2 + K_Z^2) - \sqrt{\Big(\rho^2\,(K_X^2 + K_Y^2 + K_Z^2)\Big(-16\,p\,\pi\,\gamma\,(B_X K_X + B_Y K_Y + B_Z K_Z)^2 + (4\,p\,\pi\,\gamma + B_X^2 + B_Y^2 + B_Z^2)^2\,(K_X^2 + K_Y^2 + K_Z^2)\Big)\Big)}\Big),$$
$$\frac{1}{8\pi\rho^2}\Big(\rho\,(4\,p\,\pi\,\gamma + B_X^2 + B_Y^2 + B_Z^2)\,(K_X^2 + K_Y^2 + K_Z^2) + \sqrt{\Big(\rho^2\,(K_X^2 + K_Y^2 + K_Z^2)\Big(-16\,p\,\pi\,\gamma\,(B_X K_X + B_Y K_Y + B_Z K_Z)^2 + (4\,p\,\pi\,\gamma + B_X^2 + B_Y^2 + B_Z^2)^2\,(K_X^2 + K_Y^2 + K_Z^2)\Big)\Big)}\Big)\Big\}$$

Fig. 3. *Mathematica* note book waves.nb

in the *Mathematica* note book file drift.nb (See Figure 4). The physics is clear from the output. The drift in the radial direction is caused by the toroidicity, and is of order $O(\epsilon)$; the $O(\epsilon)$ poloidal drift is a result of the toroidicity also, and the $O(\epsilon^2)$ poloidal drift is the result of the weak poloidal field. The drift in the toroidal direction is of order $O(\epsilon^2)$ and is caused by the toroidicity; this is the higher-order toroidal precession.

■ Drift Motion in a Tokamak

Load in the GeneralVectorAnalysis package

In[1]:= << ~/mathematica/GeneralVectorAnalysis.m

Set up the coordinate system for a large aspect ratio tokamak, calculate every operation to the order $O(\epsilon^2)$.

In[2]:= SetCoordinateSystem["TKCircular2", R_0, ϵ][r, θ, φ];

Set up the model equilibrium

In[3]:= Needs["Utilities‘Notation‘"]
Symbolize[$T_{\parallel}$] ; Symbolize[$T_{\perp}$] ; Symbolize[B_0];

In[5]:= $B = \mathrm{DefineVector}\left[1, 0, \mathrm{Series}\left[\dfrac{\epsilon\, r^2\ B_0}{R_0\ q[r]\left(1+\frac{\epsilon\, r}{R_0}\,\mathrm{Cos}[\theta]\right)}, \{\epsilon, 0, 2\}\right], B_0\ R_0\right];$

Define the guiding center drift velocity:

In[6]:= $b = \mathrm{Simplify}\left[\hat{B}\right];$

In[7]:= $Vg = \dfrac{c\ T_{\perp}}{e\,(|B|)^2}\ \ b\times\nabla\,(|B|);\ Vc = \dfrac{c\,2\ T_{\parallel}}{e\,(|B|)}\ b\times((b\cdot\nabla)\,b);$
Vd = FullSimplify[Vg + Vc];

To the order $O(\epsilon^2)$, the covariant form of drift velocity is :

In[8]:= Vd⟦1⟧

$$\text{Out[8]=}\ \Big\{-\frac{c\,(2\,T_{\parallel}+T_{\perp})\,\mathrm{Sin}[\theta]\,\epsilon}{B_0\ e\,R_0}+O[\epsilon]^3,\ -\frac{c\,r\,(2\,T_{\parallel}+T_{\perp})\,\mathrm{Cos}[\theta]\,\epsilon}{B_0\ e\,R_0}+\frac{c\,r^2\ ((-2\,T_{\parallel}+T_{\perp})\,q[r]-r\,T_{\perp}\ q'[r])\,\epsilon^2}{B_0\ e\,q[r]^3\ R_0^2}+O[\epsilon]^3,\ \frac{c\,r\,(2\,T_{\parallel}+T_{\perp})\,\mathrm{Cos}[\theta]\,\epsilon^2}{B_0\ e\,q[r]\,R_0}+O[\epsilon]^3\Big\}$$

Fig. 4. *Mathematica* note book drift.nb

4.3 Application III: analytic derivation of gyrokinetic-MHD formalism

The direct motivation for developing this GVA package is to automate the sophisticated vector analysis appearing in the full gyrokinetic-MHD theory[12]. We have derived the gyrokinetic moment equation with all the important physical factors as:

$$-\frac{\partial}{\partial t}\left[\frac{c^2}{4\pi}\nabla\cdot\left(\frac{1}{V_A^2}\nabla_{\perp}\phi\right)\right]+\frac{c}{4\pi}(\mathbf{B}_0\cdot\nabla)\frac{(\nabla\times\nabla\times\mathbf{A})\cdot\ \mathbf{B}_0}{B_0^2}+(\nabla A_{\|}\times\mathbf{b}_0)\cdot\nabla\frac{j_{0\|}}{B_0} \tag{21}$$

$$= -\sum_j \int (e\mathbf{v}_d \cdot \nabla f)_j \; d^3v + \sum_j \frac{1}{4\pi} \frac{c^2}{v_A^2} (\frac{3v_t^2}{4\Omega^2})_j \nabla_\perp^4 \frac{\partial \phi}{\partial t} \tag{22}$$

$$+\mathbf{B} \cdot \nabla [\frac{1}{B} \sum_j (\frac{e^2 n}{mc} \frac{v_t^2}{2\Omega^2})_j \nabla_\perp^2 A_\|] + c\mathbf{b} \times \sum_j (\frac{en_0 v_t^2}{2B_0\Omega^2})_j \cdot \nabla \nabla_\perp^2 \phi, \tag{23}$$

and the perturbed distribution functions f_j for species j will be solved for as functions of ϕ, $\psi_\|$, and the velocity space independent variables, E and μ, from the gyro-kinetic equation[12], where ϕ is the perturbed electrostatic potential, $\psi_\|$ is defined by the perturbed vector potential $A_\|$ as $A_\| = ck_\|\psi_\|/\omega$, E is the energy, and μ is the magnetic moment. The special case of this equation when all the kinetic effects are neglected can also be derived from the ideal MHD equations. Many physics features are captured inside this equation. The background inhomogeneity responsible for the TAE (Toroidal Alfven Eigenmode) modes, the kink instabilities, and the density gradient instabilities are completely localized in the left hand side, while the perturbed pressure effects, the Landau damping effects, and the FLR (Finite Larmor Radius) effects appear on the right hand side of the equation. This equation will be a key to investigate systematically some bewildering questions in today's fusion plasma physics, such as the interaction between the hot kinetic particles produced in an ignited tokamak and the long wavelength electromagnetic waves studied before using the ideal MHD theory. As one can imagine, the left hand side of this equation as a scalar function of $\psi_\|$ and ϕ is extremely complicated when applied to tokamak geometries. To study the TAE modes kinetically, we have to evaluate this equation to order $O(\epsilon)$. For internal kink modes, it has to be calculated to order $O(\epsilon^2)$. Even if assuming circular, concentric surfaces, the total number of terms involved to order $O(\epsilon^2)$ is about 1,000, which is obviously problematic if calculating by hand. With the assistance of the GVA package, we can obtain the needed results very quickly with 100% accuracy. The simplest model equilibrium assumes circular concentric flux surfaces, and uses the coordinates (r, θ, ζ) shown in Figure 2.

The magnetic field is given by

$$\mathbf{B} = \frac{B_0}{h(\theta)} (\mathbf{e}_\zeta + \frac{r}{q(r)R_0} \mathbf{e}_\theta), \tag{24}$$

where

$$h(\theta) \equiv 1 + \epsilon \cos\theta, \qquad \epsilon \equiv \frac{r}{R_0} \tag{25}$$

Assuming the general 2D expansion

$$(\psi_\|, \phi) = \sum_m (\psi_{\|m}(r), \phi_m(r)) e^{in\zeta - im\theta - i\omega t}, \tag{26}$$

we work out the left hand side of the GKM equation in this coordinate system to the order $O(\epsilon^2)$. There are more than 100 terms.

$$-\frac{\partial}{\partial t}[\frac{c^2}{4\pi}\nabla\cdot(\frac{1}{V_A^2}\nabla_\perp\phi)] + \frac{c}{4\pi}(\mathbf{B}_0\cdot\nabla)\frac{(\nabla\times\nabla\times\mathbf{A})\cdot\ \mathbf{B}_0}{B_0^2} + (\nabla\times\mathbf{A})_\perp\cdot\nabla\frac{j_{0||}}{B_0} \tag{27}$$

$$= \sum_m e^{in\zeta - im\theta - i\omega t}(\frac{TM2}{R_0^2} + \frac{TM3}{R_0^3} + \frac{TM4}{R_0^4}), \tag{28}$$

where O_2, O_3, O_4, *etc.* are differential operators acting upon every pair of $\psi_{||m}(r)$ and $\phi_m(r)$. O_2, O_3 and O_4 are $O(\epsilon^0)$, $O(\epsilon^1)$ and $O(\epsilon^2)$, respectively. The expressions for these operators are printed out below:

$$O_2 = (\frac{-2\,m\,n}{q(r)} + \frac{m^2}{q(r)^2} + n^2)\,\psi_{||}''(r) - \frac{4\,\omega^2 R_0^2\,\pi\,\rho(r)}{{B_0}^2}\phi''(r) + (\frac{2\,m\,n\,q'(r)}{q(r)^2} - \frac{2\,m^2\,q'(r)}{q(r)^3}$$
$$-\frac{2\,m\,n}{r\,q(r)} + \frac{m^2}{r\,q(r)^2} + \frac{n^2}{r})\,\psi_{||}'(r) + (\frac{-4\,\omega^2 R_0^2\,\pi\,\rho'(r)}{{B_0}^2} - \frac{4\,\omega^2 R_0^2\,\pi\,\rho(r)}{{B_0}^2\,r})\,\phi'(r)$$
$$+(\frac{-2\,m\,n\,q'(r)}{r\,q(r)^2} + \frac{2\,m^2\,q'(r)}{r\,q(r)^3} + \frac{2\,m^3\,n}{r^2\,q(r)} - \frac{m^4}{r^2\,q(r)^2} - \frac{m^2\,n^2}{r^2})\,\psi_{||}(r) + \frac{4\,m^2\,\omega^2 R_0^2\,\pi\,\rho(r)}{{B_0}^2\,r^2}\phi(r);$$

$$O_3 = (\frac{-i\,m\,r\,\sin(\theta)}{q(r)^2} + \frac{2\,m\,n\,r\,\cos(\theta)}{q(r)} - 2\,n^2\,r\,\cos(\theta))\,\psi_{||}''(r) - \frac{8\,\omega^2 R_0^2\,\pi\,r\,\cos(\theta)\,\rho(r)}{{B_0}^2}\phi''(r)$$
$$+(\frac{2\,i\,m\,r\,\sin(\theta)\,q'(r)}{q(r)^3} - \frac{2\,m\,n\,r\,\cos(\theta)\,q'(r)}{q(r)^2} - \frac{2\,i\,n\,\sin(\theta)}{q(r)} - \frac{i\,m\,\sin(\theta)}{q(r)^2} + \frac{2\,m\,n\,\cos(\theta)}{q(r)}$$
$$+\frac{m^2\,\cos(\theta)}{q(r)^2} - 3\,n^2\,\cos(\theta))\,\psi_{||}'(r) + (\frac{-8\,\omega^2 R_0^2\,\pi\,r\,\cos(\theta)\,\rho'(r)}{{B_0}^2} - \frac{20\,\omega^2 R_0^2\,\pi\,\cos(\theta)\,\rho(r)}{{B_0}^2})\,\phi'(r)$$
$$+(\frac{-2\,i\,m\,\sin(\theta)\,q'(r)}{q(r)^3} + \frac{4\,m\,n\,\cos(\theta)\,q'(r)}{q(r)^2} - \frac{2\,m^2\,\cos(\theta)\,q'(r)}{q(r)^3} + \frac{2\,i\,m^3\,\sin(\theta)}{r\,q(r)^2}$$
$$-\frac{i\,m\,n^2\,\sin(\theta)}{r}\frac{2\,m^3\,n\,\cos(\theta)}{r\,q(r)} - \frac{2\,m\,n\,\cos(\theta)}{r\,q(r)} + \frac{2\,m^2\,n^2\,\cos(\theta)}{r})\,\psi_{||}(r)$$
$$+(\frac{-12\,i\,m\,\omega^2 R_0^2\,\pi\,\rho(r)\,\sin(\theta)}{{B_0}^2\,r} + \frac{8\,m^2\,\omega^2 R_0^2\,\pi\,\cos(\theta)\,\rho(r)}{{B_0}^2\,r})\phi(r);$$

$$O_4 = (\frac{\frac{i}{2}\,m\,r^2\,\sin(2\theta)}{q(r)^2} - \frac{m\,n\,r^2\,\cos(2\theta)}{q(r)} - \frac{m\,n\,r^2}{q(r)} - \frac{n^2\,r^2}{q(r)^2} + \frac{2\,m\,n\,r^2}{q(r)^3} - \frac{m^2\,r^2}{q(r)^4}$$
$$+\frac{3\,n^2\,r^2\,\cos(2\theta)}{2} + \frac{3\,n^2\,r^2}{2})\,\psi_{||}''(r) + (\frac{4\,\omega^2 R_0^2\,\pi\,r^2\,\rho(r)}{{B_0}^2\,q(r)^2} - \frac{2\,\omega^2 R_0^2\,\pi\,r^2\,\cos(2\theta)\,\rho(r)}{{B_0}^2}$$

$$
\begin{aligned}
&-\frac{2\,\omega^2 R_0^2\,\pi\, r^2\,\rho(r)}{{B_0}^2})\,\phi''(r)+(\frac{-i\,m\,r^2\,\sin(2\theta)\,q'(r)}{{q(r)}^3}+\frac{m\,n\,r^2\,\cos(2\theta)\,q'(r)}{{q(r)}^2}+\frac{m\,n\,r^2\,q'(r)}{{q(r)}^2}\\
&+\frac{2\,n^2\,r^2\,q'(r)}{{q(r)}^3}-\frac{6\,m\,n\,r^2\,q'(r)}{{q(r)}^4}+\frac{4\,m^2\,r^2\,q'(r)}{{q(r)}^5}+\frac{3\,i\,n\,r\,\sin(2\theta)}{q(r)}-\frac{m\,n\,r\,\cos(2\theta)}{q(r)}-\frac{m\,n\,r}{q(r)}\\
&-\frac{m^2\,r\,\cos(2\theta)}{2\,{q(r)}^2}-\frac{3\,n^2\,r}{{q(r)}^2}-\frac{m^2\,r}{2\,{q(r)}^2}+\frac{6\,m\,n\,r}{{q(r)}^3}-\frac{3\,m^2\,r}{{q(r)}^4}+3\,n^2\,r\,\cos(2\theta)+3\,n^2\,r)\,\psi_{||}'(r)\\
&+(\frac{4\,\omega^2 R_0^2\,\pi\,r^2\,\rho'(r)}{{B_0}^2\,{q(r)}^2}-\frac{2\,\omega^2 R_0^2\,\pi\,r^2\,\cos(2\theta)\,\rho'(r)}{{B_0}^2}-\frac{2\,\omega^2 R_0^2\,\pi\,r^2\,\rho'(r)}{{B_0}^2}-\frac{8\,\omega^2 R_0^2\,\pi\,r^2\,\rho(r)\,q'(r)}{{B_0}^2\,{q(r)}^3}\\
&+\frac{12\,\omega^2 R_0^2\,\pi\,r\,\rho(r)}{{B_0}^2\,{q(r)}^2}-\frac{8\,\omega^2 R_0^2\,\pi\,r\,\cos(2\theta)\,\rho(r)}{{B_0}^2}-\frac{8\,\omega^2 R_0^2\,\pi\,r\,\rho(r)}{{B_0}^2})\,\phi'(r)+(-\frac{n^2\,r^2\,{q'(r)}^2}{{q(r)}^4}\\
&+\frac{2\,m\,n\,r^2\,{q'(r)}^2}{{q(r)}^5}-\frac{m^2\,r^2\,{q'(r)}^2}{{q(r)}^6}+\frac{2\,i\,m\,r\,\sin(2\theta)\,q'(r)}{{q(r)}^3}-\frac{3\,m\,n\,r\,\cos(2\theta)\,q'(r)}{{q(r)}^2}\\
&-\frac{3\,m\,n\,r\,q'(r)}{{q(r)}^2}+\frac{m^2\,r\,\cos(2\theta)\,q'(r)}{{q(r)}^3}+\frac{m^2\,r\,q'(r)}{{q(r)}^3}+\frac{6\,m\,n\,r\,q'(r)}{{q(r)}^4}-\frac{6\,m^2\,r\,q'(r)}{{q(r)}^5}\\
&-\frac{i\,m^3\,\sin(2\theta)}{{q(r)}^2}+\frac{3\,i}{2}\,m\,n^2\,\sin(2\theta)+\frac{m^3\,n\,\cos(2\theta)}{q(r)}+\frac{3\,m\,n\,\cos(2\theta)}{q(r)}-\frac{2\,m\,n^3}{q(r)}\\
&+\frac{m^3\,n}{q(r)}+\frac{3\,m\,n}{q(r)}-\frac{m^2\,\cos(2\theta)}{2\,{q(r)}^2}+\frac{6\,m^2\,n^2}{{q(r)}^2}-\frac{3\,m^2}{2\,{q(r)}^2}-\frac{6\,m^3\,n}{{q(r)}^3}-\frac{4\,m\,n}{{q(r)}^3}+\frac{2\,m^4}{{q(r)}^4}+\frac{4\,m^2}{{q(r)}^4}\\
&-\frac{3\,m^2\,n^2\,\cos(2\theta)}{2}-\frac{3\,m^2\,n^2}{2})\,\psi_{||}(r)+(\frac{-6\,i\,m\,\omega^2 R_0^2\,\pi\,\rho(r)\,\sin(2\theta)}{{B_0}^2}+\frac{8\,m\,n\,\omega^2 R_0^2\,\pi\,\rho(r)}{{B_0}^2\,q(r)}\\
&-\frac{8\,m^2\,\omega^2 R_0^2\,\pi\,\rho(r)}{{B_0}^2\,{q(r)}^2}+\frac{2\,m^2\,\omega^2 R_0^2\,\pi\,\cos(2\theta)\,\rho(r)}{{B_0}^2}+\frac{2\,m^2\,\omega^2 R_0^2\,\pi\,\rho(r)}{{B_0}^2})\phi(r).
\end{aligned}
$$

Some expressions similar to the right hand side of Equation (21) have been derived before to study TAE modes [13–15] . Usually, only a few terms of the toroidal correction are kept. As we know from the expression for O_3 above, there are 23 terms for the first order toroidal correction. Being aware of this problem, Berk *et al* [15] argued that keeping only the first order correction for the second derivative terms would be sufficient to obtain the main features of the TAE modes. However, the instability criterion will be substantially affected by other first order corrections. As for the internal kink modes, every term up to order $O(\epsilon^2)$ is important.

5 Conclusions and future work

GVA has been tested and has proved to be reliable and efficient in doing automatic symbolic calculation in general coordinate systems for those problems where the real physics contained in the basic starting equations is obscured by the complexity of the detailed algebra. Its usefulness in carrying out asymptotic analysis in practical problems is greatly appreciated. In the field of fusion plasma physics, all the important physics phenomena happen in the order $O(\epsilon)$ and $O(\epsilon^2)$, or even $O(\epsilon^3)$, where many more applications of the GVA are expected. New functions such as factoring will be added in so that the capability of doing coordinate independent vector calculus derivations will be enhanced. In the framework of the differential forms formalism, we believe that there are no difficulties in principle in implementing these functionals. Not only will this save a huge amount of the researcher's time and improve the accuracy dramatically, but also this GVA package will be an intelligent tool in finding the right approaches for theoretical problems. The GVA package and other related documents are available at `http://w3.pppl.gov/~hongqin/`.

Acknowledgement

This work was supported by U.S. Department of Energy Contract No. DE-AC02-76-CH0-3073. Hong Qin was partially supported by the *Mathematica* Visiting Scholar Grant, Wolfram Research Inc., USA.

References

[1] J. P. Freidberg, *Ideal Magnetohydrodynamics* (Plenum, New York, 1987), pp. 132-145.

[2] S. Wolfram, *The* Mathematica *Book*, 3rd edition (Wolfram Media, 1996).

[3] R. M. Wald, *General Relativity* (University of Chicago Press, 1984), pp. 19.

[4] M. N. Bussac, R. Pellat, D. Edery, and J. L. Soule, Phys. Rev. Letter **35**, 1638 (1975).

[5] R. Grimm, J. M. Greene, and L. Johnson, Methods in Computational Phys. **16**, 253 (1976).

[6] R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*, 2nd Edition (Springer-Verlag, 1988), pp. 415-446.

[7] C. Von Westenholz, *Differential Forms in Mathematical Physics* (North-Holland, 1981), pp. 141-207.

[8] R. M. Kulsrud, *Basic Plasma Physics* (North-Holland, 1983), pp. 1.

[9] S. Sternberg, Lectures on Differential Geometry (Prentice-Hall, Englewood Cliffs, 1964), pp 137-141.

[10] W. D. D'haeseleer, W. N. G. Hitchon, J. D. Callen, and J. L. Shohet, *Flux Coordinates and Magnetic Field Structure* (Springer-Verlag, 1991) pp. 116-155.

[11] A. H. Boozer, Phys. Fluid B **26**, 1288 (1983).

[12] H. Qin, W. M. Tang, and G. Rewoldt, "Gyrokinetic Theory for Arbitrary Wavelength Electromagnetic Modes in Tokamaks", submitted to Phys. Plasmas.

[13] M. N. Rosenbluth and P. H. Rutherford, Phys. Rev. Letter **34**, 1428 (1975).

[14] G. Y. Fu and J. W. Van Dam, Phys. Fluids B **1**, 1949 (1989).

[15] H. L. Berk, J. W. Van Dam, Z. Guo, and D. M. Lindberg, Phys. Fluids B **1**, 1806 (1992).