

# 1 Introduction

The purpose of this document is to describe important functionality of the SymPy Symbolic Vector module. This includes vector definitions, operations, and identities that are implemented in the module. Additionally the document describes how important SymPy functions, such as `expand` and `vector_simplify`, interact with various vector expressions.

The SymPy Symbolic Vector module implements the mainstream Gibbs/Heaviside vector algebra and calculus. The Symbolic Vector module focuses on manipulating vector algebra and differential vector expressions. The module focuses on these operations because of the multitude of identities and definitions in this domain. SymPy is well suited to using these identities to grind through complex vector expressions.

At this time, support for Clifford vector algebra (or other vector algebras) is not included. Additionally, extensions into tensors (such as taking the gradient of a vector function) is not supported.

Vector integral theorems, such as Greens, Stokes, etc., all require considerable mathematical machinery, for instance, defining a closed surface's relationship to a volume. These theorems are very useful to *conceptually* simplify a vector expression. However, the theorems are not well suited to *mechanistic* simplification procedures, which is what computer algebra system excel at. Therefore, no vector integral theorems are implemented as simplification methods in the Symbolic Vector module.

In this document:

- $\mathbf{F}$  and  $\mathbf{G}$  are vector functions.
- $\phi$  and  $\psi$  are scalar functions.

## 2 One $\nabla$ and Two Operands

Of the 12 possible combinations of  $\nabla$  acting on products, 6 are valid mathematical statements within Gibbs/Heaviside vector algebra ([1]).

For the 6 valid expressions, an identity is provided below for how this expression expands. For each of these valid expressions `expand` should change

the left hand side (LHS) of the expression into the right hand side (RHS). Similarly, `vector_simplify` should change the RHS into the LHS.

## 2.1 Gradient of two Scalar functions

$$\nabla(\phi\psi) = \phi\nabla\psi + \psi\nabla\phi$$

## 2.2 Gradient of a dot product

$$\nabla(\mathbf{F} \cdot \mathbf{G}) = \mathbf{F} \times (\nabla \times \mathbf{G}) + \mathbf{G} \times (\nabla \times \mathbf{F}) + (\mathbf{F} \cdot \nabla)\mathbf{G} + (\mathbf{G} \cdot \nabla)\mathbf{F}$$

## 2.3 Divergence of the product between a scalar and vector function

$$\nabla \cdot (\phi\mathbf{F}) = \phi(\nabla \cdot \mathbf{F}) + \mathbf{F} \cdot \nabla\phi$$

## 2.4 Divergence of vector functions cross product

$$\nabla \cdot (\mathbf{F} \times \mathbf{G}) = \mathbf{G} \cdot (\nabla \times \mathbf{F}) - \mathbf{F} \cdot (\nabla \times \mathbf{G})$$

## 2.5 Curl of the product between a scalar and vector function

$$\nabla \times (\phi\mathbf{F}) = \phi(\nabla \times \mathbf{F}) - \mathbf{F} \times \nabla\phi$$

## 2.6 Curl of vector functions cross product

$$\nabla \times (\mathbf{F} \times \mathbf{G}) = \mathbf{F}(\nabla \cdot \mathbf{G}) - \mathbf{G}(\nabla \cdot \mathbf{F}) + (\mathbf{G} \cdot \nabla)\mathbf{F} - (\mathbf{F} \cdot \nabla)\mathbf{G}$$

# 3 Two $\nabla$ and One Operand

Of the possible 18 ways to arrange two  $\nabla$ , three operations of multiply, dot, and cross, with one operand, only 5 are valid ([1]). Of these 5 expressions, two are always zero and the other three are related by the curl of a curl identity. Note that  $\nabla \cdot \nabla = \nabla^2$ , which is termed the Laplacian.

For these three identities, a discussion of how the SymPy Vector module should operate is provided below.

### 3.1 Curl of a Gradient is 0:

$$\nabla \times (\nabla \phi) = 0$$

If a curl of a gradient is contained in a SymPy vector expression, then `vector_simplify` should reduce this to 0.

### 3.2 Divergence of a Curl is 0

$$\nabla \cdot (\nabla \times \mathbf{F}) = 0$$

If a divergence of a curl is contained in a SymPy vector expression, then `vector_simplify` should reduce this to 0.

### 3.3 Curl of a Curl

$$\nabla \times (\nabla \times \mathbf{F}) = \nabla(\nabla \cdot \mathbf{F}) - \nabla^2 \mathbf{F}$$

Which can be rearranged as:

$$\nabla(\nabla \cdot \mathbf{F}) = \nabla \times (\nabla \times \mathbf{F}) + \nabla^2 \mathbf{F}$$

Or as:

$$\nabla^2 \mathbf{F} = \nabla(\nabla \cdot \mathbf{F}) - \nabla \times (\nabla \times \mathbf{F})$$

Given any of the 3 LHS above, `expand` should return the associated RHS. Likewise, given the RHS of the expression, `vector_simplify` should return the associated LHS.

## References

- [1] Robert J. Lopez, “Advanced Engineering Mathematics”, Addison- Wesley, 2001.