# MATLAB code for EM integral solver

## Classes

### Vector

Class of complex vector.

Properties:

* Value: 3 complex doubles of the vector
* Type: ‘cart’ for Cartesian, ‘cyl’ for Cylindrical, ‘pol’ for Polar

Methods:

* Constructor: Creates Vector instance from 3 complex values and optional type string. If no type is given, defaults to Carthesian. Also has copy-constructor behavior.
* Ordinary arithmetic operations on vectors, abs, real, imag and conj.
* Convert\_to(‘new\_type\_here’): Converts to different type
* Mtimes: scalar multiplication, overloads ‘\*’
* And: Cross multiplication, overloads ‘&’

### Coordinate

Inherits from Vector class, the only difference – allows only real values.

### Function3

Function of 3 variables.

Properties:

* F: function handle

Methods:

* Constructor: Creates Function3 instance from anonymous function handle. Function handle must begin with @(c1,c2,c3). Also has copy-constructor behavior.
* Ordinary arithmetic operations on functions, abs, real, imag, exp and conj.
* Subsref: Overloads ‘()’ to allow for standard f(x) behavior.
* D1(dc1),D2(dc2),D3(dc3): Numeric differentiation by first, second or third variable. Dc1,dc2,dc3 are differentiation epsilons.

### Scalar\_field

Class of scalar field.

Properties:

* Gen\_func: Function3
* Type: type of vector, for which gen\_func is written

Methods:

* Constructor: Makes Scalar\_field instance from one Function3 and one type inputs. Also supports copy-constructor.
* Subsref: Overloaded ‘()’, to allow for the following behavior:

Scalar\_field\_value = Scalar\_field(‘Coordinate instance here’).

* Some arithmetic operations on scalar fields.
* Grad(dc1, dc2, dc3): Generates new Vector\_field instance, which is numerical gradient of the given Scalar\_field. Dc1, dc2, dc3 are differentiation epsilons.

### Vector\_field

Class of vector field.

Properties:

* Gen\_func1, gen\_func2, gen\_func3: Function3 entities generating vector field values.
* Type: type of vector, for which gen\_funcs are written.

Methods:

* Constructor: Makes Vector\_field instance from 3 Function3 inputs and type input. Also supports copy-constructor.
* Subsref: Overloads ‘()’, to allow for functionality identical to that of Scalar\_field.
* Some arithmetic operations on vector fields.
* Div(dc1, dc2, dc3): Generates new Scalar\_field instance, which is numerical divirgence of the given Vector\_field. Dc1, dc2, dc3 are differentiation epsilons.
* Curl(dc1, dc2, dc3): Generates new Vector\_field instance, which is numerical curl of the given Vector\_field. Dc1, dc2, dc3 are differentiation epsilons.

### G

Class of general Green function.

Properties:

* Mu: permeability of space
* K: wave vector of needed frequency
* String: string, which contains describes this Green function instance.

Methods:

* Constructor: Makes new G instance from value of k and optional value of mu.
* Subsref: Overloads ‘()’ operator. Two different behaviors:
  + G(‘Coordinate1, Coordinate2’) produces value of Green function between those two coordinates.
  + G(‘’source’, Coordinate1’) produces Scalar\_field instace, which represents value of Green function with a source at Coordinate1.

## Aux functions

A lot of auxiliary functions for proper functioning of class library. Generally, may be called also from main workspace, but I don’t quite see, why one would want to.