Vector calculus MA2VC/MA3VC — `VCplotter.m` readme

The present document briefly explains how to use the Matlab/Octave[1] function `VCplotter.m` to visualise scalar and vector fields, curves, domains and surfaces, in various coordinate systems. This might be helpful to get a better geometric understanding of these objects and to comprehend how formulas and shapes are related to each other.

First, save the file `VCplotter.m` on your computer (keeping the file name `VCplotter.m`), open Matlab or Octave, and move to the directory where you saved the file using the command `cd`, for instance: `cd E:\Octave\OctaveProjects`

Now you can proceed in two ways.

—Use 1—

Simply type `VCplotter` in the Matlab/Octave main window, you will see something like

```
What do you want to plot?  Choose your option:
  1 - Curve
  2 - 2D scalar field in Cartesian coordinates
  3 - 2D scalar field in polar coordinates
  4 - 2D vector fields in Cartesian coordinates
  5 - 2D vector fields in polar coordinates
  6 - Mapping of 2D domains
  7 - Domains in cylindrical coordinates
  8 - Domains in spherical coordinates
  9 - Parametric surface (on rectangular region)
--->
```

Type the choice corresponding to the desired kind of plot. For instance, if you want to draw a two-dimensional scalar field in polar coordinates, just type `3` and press enter. You will be asked to type the field you want to plot, as a Matlab function:
`Type field f(r,theta), e.g.:  @(r,theta) cos(5*theta)*r^2/(1+r^2)`
Now type, for instance following the suggestion, `@(r,theta) cos(5*theta)*r^2/(1+r^2)`. Do not forget the `@(r,theta)` to tell Matlab that your input is a function in the variables r and θ. Even if you want to enter a constant function, type for instance `@(r,theta) 1` and not simply `1`. (Important: always respect the order of the variables, if you use `@(theta,r)` instead of `@(r,theta)` you will get some other field!) Finally you will be asked to type the domain radius, which has to be a positive number (e.g. type `2`). After you have inserted this, you will see both the level line plot and the graph surface of the field you have selected. You can rotate three-dimensional images with the mouse (in most cases).

All the other options in the initial menu work similarly, some of them require more inputs (two or three components for vector fields and curves, four domain extrema for rectangular domains in Cartesian coordinates).

—Use 2—

Alternatively, you can directly pass the same values asked by the interface (as described above) as inputs of the `VCplotter` function, separated by commas. E.g. typing
`VCplotter(3, @(r,theta) cos(5*theta)*r^2/(1+r^2), 2);`
will immediately generate the same figure as the procedure above. This way of using the function is more convenient when you want to visualise in sequence several small modifications of the same object (e.g. the same field on different domains). Some examples (covering all the available options) are:

```
VCplotter(1,1,@(t) (1+t)*cos(t),@(t) (1+t)*sin(t),0,2*pi);     % planar Cartesian curve
VCplotter(1,2,@(t) t^2,@(t) sin(t),0,20);                      % planar polar curve
VCplotter(1,3,@(t) t,@(t) t*sin(1/t),@(t) cos(10*t)*t^2,0,1);  % 3D cartesian curve
VCplotter(1,4,@(t)t,@(t)t,@(t)t,0,100);                        % 3D cylindrical curve
VCplotter(1,5,@(t)1,@(t)t,@(t)t,0,2*pi);                       % 3D spherical curve
VCplotter(2,@(x,y) x^2+cos(5*y),-1,1,0,3);                     % 2D scalar Cartesian field
VCplotter(3,@(r,theta) cos(5*theta)*r^2/(1+r^2),2);            % 2D scalar polar field
VCplotter(4,@(x,y) -x-y,@(x,y) x-y,-1,1,-1,1);                 % 2D vector Cartesian field
VCplotter(5,@(r,theta) 0,@(r,theta) cos(3*theta)*(r^2-r),1);   % 2D vector polar field
VCplotter(6,@(x,y) x-y,@(x,y) y+sin(x),-pi,pi,-2,2);           % 2D domain mapping
VCplotter(7,@(theta,z) (1-z^2)*(2+sin(6*theta)),-1,1);         % cylindrical domain
VCplotter(8,@(phi,theta) 2+sin(3*phi)+cos(5*theta));           % spherical domain
VCplotter(9,@(u,w)(3+cos(w))*cos(u),@(u,w)(3+cos(w))*sin(u),@(u,w)sin(w),-pi,pi,-pi,pi);  % parametric surface
```

If you copy and paste these commands directly in Matlab/Octave you immediately see some examples (right-click or centre-click with the mouse to paste text in Octave). You can see the same list of examples also by typing `help VCplotter`.

[1] If you are not familiar with it, **GNU Octave** is a free software for numerical computations. It is extremely similar to Matlab, most of the commands are identical (and indeed the single file `VCplotter.m` works in both environments). You don't even need to install it: to use it on Windows, go to
`http://sourceforge.net/projects/octave/files/Octave%20Windows%20binaries/`
Click on the link `Octave 3.6.4 for Windows MinGW installer`, download the file `Octave3.6.4_gcc4.6.2_20130408.7z`, extract the directory tree from the zip file, click on the link `Octave3.6.4_gcc4.6.2` (in your folder) and you are ready to go! (More information in the online readme file.) If you use Linux, Octave is probably already installed in your system, type `octave` in the terminal to run it.