Full Example Model (on a 2x2 linear system)

```
model Eig2File_1A_V1
parameter Real A[:, :]=[0, 1; 1.35, 0.7];
parameter Integer dim[:]=size(A);
parameter Integer dimmy1=dim[1];
parameter Integer dimmy2=dim[2];
parameter Real my_range=20;
parameter Real x1_0=0;
parameter Real x2_0=0;
parameter String dirname = "my_new_directory";
parameter String mkd = "mkdir ";
parameter String fnam =  "my_new_file.txt";
parameter String fS1 = "/";
String dircmdstr;
String wdir;
String f_path;
String myFrmt = "G";
//String ini_path;
String arow1;
String arow2;
String brow1;
String brow2;
//Boolean dummy;
Real my_x[2, 1];
Real eigval[2, 2];
Real eigvec[2, 2];
Real a11;
Real a12;
Real a21;
Real a22;
Real b11;
Real b12;
Real b21;
Real b22;
initial equation
my_x[1, 1] = x1_0;
my_x[2, 1] = x2_0;

equation
(eigval,eigvec) = Modelica.Math.Matrices.eigenValues(A);
der(my_x) = A*my_x;
a11=eigvec[1,1];
a12=eigvec[1,2];
a21=eigvec[2,1];
a22=eigvec[2,2];
b11=eigval[1,1];
b12=eigval[1,2];
b21=eigval[2,1];
b22=eigval[2,2];

 when terminal() then
 arow1=  String(a11)+ " " +String(a12);
 arow2=  String(a21)+ " " +String(a22);
 brow1=  String(b11)+" "+String(b12);
 brow2=  String(b21)+" "+String(b22);
 end when;
```

Try altering these values to create different types of system, such as one with complex eigenvalues.

Important initial conditions.

Here, strings used to send results to the operating system are assigned.

Change the file and folder names if needed. The folder will need to be created in the model or already exist!

These strings are there to create lines of text written to file, see upcoming post on file management!

Using different initial conditions will put the trajectory on a different path. Any linear algebra textbook will show a number of phase-portraits each with a number of trajectories plotted.

Have a look at:

http://mathlets.org/mathlets/linear-phase-portraits-matrix-entry/

Terminal condition prevents instructions being re-executed at every timestep.

algorithm
 	
when terminal() then
 	
wdir := Modelica.Utilities.System.getWorkDirectory();
dircmdstr := mkd + dirname;
system(dircmdstr);
 	
f_path :=wdir + fSl + dirname;
Modelica.Utilities.System.setWorkDirectory(f_path);
 	
Modelica.Utilities.Streams.print(arow1,fnam);
Modelica.Utilities.Streams.print(arow2,fnam);
Modelica.Utilities.Streams.print(brow1,fnam);
Modelica.Utilities.Streams.print(brow2,fnam);
end when;
 	
end Eig2File_1A_V1;
	Algorithm section with procedural code, instead of universally applicable relationships.

-0.756887 -0.538817 0.653546 -0.842423 -0.863466 0 1.56347 0	The output will be a .txt file with these contents. The upper block of four numbers are the two eigenvectors (in columns). The lower block of four values are the real (left) and imaginary (right) parts of the eigenvalues. Spaces have been placed to assist reading the values. You can also add more spaces in the programme.

Figure 5. Code to obtain eigenvectors and write to file