Pic2Mag Ace

Ace Version 1.31 by Michael Snyder 09/25/2016

You draw virtual magnets and Pic2Mag draws the fields! (tm)

Usage : pic2mag myfile.png or pic2mag myfile.png z w v

Pic2mag and your 640x640 pixel png file should be in the same folder. Start a command prompt, change directory to the folder with your files, and enter 'pic2mag myfile.png'.

Where z is the distance in millimeters away from the magnetic surface in a range between -1000 and 1000 millimeters; and the default distance is 3 mm.

Where w is the thickness of the magnetic surface in mm between -1000 and 1000 millimeters, and the default thickness is 1 mm.

Where v is the letter 'v' which instructs the program to export the vector tables in text file format.

example - 'pic2mag testgrid01.png' writes the png files for 3 mm above the surface.
example - 'pic2mag testgrid01.png 3' writes the png files for 3 mm above surface.
example - 'pic2mag testgrid01.png 3 1 v' writes the png files and the vector data files in text format for 3 mm above the surface with the defined volume as 1 mm thick.

Note - Using the vector data export option, requires you to enter the z layer distance and thickness value options because the program inputs are sequence dependant.

Pic2Mag uses colors in a graphics file to represent magnetic materials with different magnetic moments.(tm) The program looks for certain RGB values in the png file and when it finds the exact match the program plots the pixel as a permanent magnet with a defined magnetic moment angle. Different colors have different magnetic moment angles. More pixels take more time to process.

You can edit your png file in any graphics editor. I did the program testing with mspaint. Set your picture size to 640x640 and edit the colors to match the listed RGB colors. See the included mspaintcolors.png as an example of setting up the colors in mspaint. Pic2Mag expects colored objects on a white background.

Renaming the Pic2Mag Executable File

The Pic2Mag executable is only about 400 kilobytes and the test image png files are only a few hundred kilobytes which makes Pic2Mag program ideal to email to customers, but sadly most email systems freak out over an executable file.

To get around this limitation, the Pic2Mag executable file in the zip file is renamed to 'pic2mag.rename' and it needs to be renamed using the 'rename pic2mag.rename pic2mag.exe' command, or right clicking in windows and picking the rename option.

The author suggests right clicking and running an antivirus scan on the executable file after renaming it. While not an ideal solution, renaming the stored file is a handy way to get the program into the hands of the users.

'pic2mag myfile.png' produces five output files

The first output file is called contours.png and is a isopotential plot of the vector field energies. This is similar to a topographical map but using a repeating sequence of red, green and blue lines.

The second file is called streamlines.png and it is the streamline plot of the vector fields. The streamlines are plotted as compass needles with yellow and purple markings. The compass needles map the direction of the magnetic fields.

It should be noted that the compass needle algorithm has a limitation when the vector data has the following condition $\left(z^2\hat{k}\right) \gg \left(x^2\hat{i} + y^2\hat{j}\right)$ When the data z vector components are much greater than the combined x and y components then the magnetic streamline plots can converge into a single point as seen in the 'colorwhl01___Z5003_streamlines.png' file.

The reason this happens is that compass needle algorithm does not have access to the z component field data of the adjoining layers and can only base the plotting on the x and y vector components, and thus the plots may at first seem inconsistent with Maxwell's law that a magnetic field has a zero divergence.

The irony is that it is exactly the magnetic zero divergence law causing the 2d algorithm plotting limitations. The reason that the lines seem to converge in some of the plots is that there is a high z component of the layer being plotted that is only balanced out, in the other z layers of the stack. When in doubt, load and examine the vector data in matlab. I believe that you will find that the vector data is highly accurate and has zero divergence for a large number of z layers.

The easiest way to explain this plotting inconsistency to your students is to point out that real compass needles would do exactly the same behavior; as plotted by the Pic2Mag's compass needle algorithm. A physical compass needle which is restricted to an xy plane would point in the same directions as seen in the plots.

The last three files called i_vector, j_vector, and k_vector are the vector field data of permanent magnets. The calculation of the vectors are accurate and use an physical constant of 1. The vector field data is processed internally in non-scaled units and a person would need to scale the set of vector field data in order to fit the calculated vector fields to the real world with a few measurements.

You can load these vector data files in other programs. For example in matlab

```
load myfile_i_vector.txt
load myfile_j_vector.txt
load myfile_k_vector.txt
```

Then use the matlab gui to change the 2d array names to i_vector,j_vector,k_vector and enter:

```
test=sqrt(i_vector.*i_vector+j_vector.*j_vector+k_vector.*k_vector);
contour(log(test),100);
```

Pic2Mag Ace Cache Files

The Ace version of the program works at 640x640 resolution and requires the use of z layer solution cache files based on the z layer distance of the observed plane from the magnetic surface. In other words, each time you observe a different z layer, the program computes new internal data structures and saves them to disk for later reuse. This saves significant time the next time you process that same color and z layer.

Cache File Size Warning!

If you use all the colors on a single z layer, the complete set of cache files will weigh in at 340 megabytes per layer. A 64 gigabyte SSD hard drive has room for about 150 full sets of z layer cache files when using all the colors in the Clock Color Spin Vectors(tm) table.

When in doubt, you can delete the cache files in the Pic2Mag program directory to make more room with 'del *.dat' command; or use windows to choose and delete the cache files. The Pic2Mag program only creates cache files as they are needed and will produce smaller numbers of cache files, if smaller numbers of colors are used within the png files. In other words if you only use a few colors then you can have thicker magnetic layers with many z layers.

Processing Magnetic Surface Thickness Parameters

At first glance the idea of program that is designed to process magnetic surfaces, to have a input parameter called thickness might seem like a oxymoron. How can we talk about a magnetic pixel and a magnetic volume at the same time?

We do this by using the following set of definitions:

First we define the volume represented by a magnetic pixel as 1 mm x 1 mm x 1 mm of permanent magnet material with the origin z plane of our system bisecting the cube's volume along the z axis.

In other words a magnetic pixel in the center of the image, is at the x,y,z origin and represents a volume of material defined on the x axis of -0.5 mm to 0.5 mm and the y axis of -0.5 mm to 0.5 mm and the z axis from -0.5 mm to 0.5 mm

Given the benefit that the Pic2Mag program only works with integer coordinates, we can then collapse our 1mm x 1 mm x 1mm volume of a permanent magnet material into a 2d plane at z=0 that still effects the vector tables of all the z layer planes above and below it in mostly the same way as the magnetic volume.

In other words, if we only care about the plane's effects on other planes then it makes little difference if we collapse a unit volume into a unit area, or extrude a unit area into a unit volume. Conceptually the effects on other layers are nearly the same as long as we use integers for our distance measurements.

The Pic2Mag program fixes the magnetic surface at the z value of zero and allows the user to observe the effects of the magnetic surface at all the possible z layers. The notation of 'S0' stands for the surface z layer at a zero value. For observation we will use the notation of 'C' for calculation. So the default processing operation of Pic2Mag program is to calculate the function V[S0,C3] which reads as producing the vector field at 3mm above the magnetic surface.

Now the user should convince themselves that if we fixed that magnetic surface at z=100 layer and the observation layer at z=103, expressed as V[S100,C103] then this produces the same set of vector data as V[S0,C3]. The fact that V[S0,C3] = V[S100,C103] is obvious because we know that the magnetic vector fields 3 mm above the magnetic surface must be the same values; no matter where we place the Cartesian origin.

Therefore the vector fields that the program produces are only a function of the relative z distances between magnetic plane and observed plane. Imagine that we have three magnet surface layers in the same system instead of one surface.

For example we have magnetic surface at z=0 and a copy of the same surface at z=-1 and a copy of the same surface at z=-2 and we want to know their effects at the standard z=3 observation layer. Writing it in my Pic2Mag notation and remembering that vector field functions are accumulative:

V[S0,C3]+V[S(-1),C3]+V[S(-2),C3]

But we know that the Pic2Mag vector fields are only functions of relative z distances meaning that I can rewrite the function sequence as:

V[S0,C3]+V[S0,C4]+V[S0,C5]

The reader should convince themselves that these two vector field sequences produce the same results. This concept is similar to how 3d printers print by moving the print bed along the z axis instead of moving the print head.

When you use the thickness option in the Pic2Mag program, the program changes the vector field operations to keep the active magnetic layer at the z=0 but produces a valid vector field for the specified stack of z layer surfaces.

'pic2mag testgrid01.png 3 2' with a positive thickness parameter.

V[S0,C3]+V[S(-1),C3]=V[S0,C3]+V[S0,C4]

Produces the combined vector fields for two identical magnetic surfaces one located at z=0, and one at z=-1 and then the field observed at z=3 .

'pic2mag testgrid01.png 3 -2' with a negative thickness parameter.

V[S0,C3]+V[S1,C3]=V[S0,C3]+V[S0,C2]

Produces the combined vector fields for two identical magnetic surfaces with one located at z=0, and one at z=1 and then the field observed at z=3 .

Parallel Processing with Pic2Mag Ace

The program is setup to be self contained and has about a 250 megabyte RAM memory footprint while processing. Other than writing a cache file when the file is not found and reading cache files; the program has no temporary file operations meaning that you can run multiple copies of Pic2mag in the same program directory and using the same cache files.

In other words the program is highly parallel in the sense that an eight core processor can run eight copies of the program at the same time by using different command prompt instances. The author processes animations frame by frame.

The Pic2Mag Ace version accepts 640 pixel by 640 pixel png files where each pixel is a 1mm cube of permanent magnet and then generates the vector field 3mm above the plane. It is free for everyone and everybody to use.

For example in a school lab where each student creates his or her own png file and then plots out the vector fields of their own magnetic system. The Pic2Mag command line executable is a 32 bit PC program and should run on most windows systems from Windows NT to Windows 10. It would not be difficult to setup batch files to run pic2mag from a windows icon on the desktop.

The Pic2Mag Pro version accepts 1280 by 1280 pixel png files and can be ordered for $29.95 by emailing msnyder@pic2mag.com.

The Pic2Mag Cuda version is coming in the spring of 2017 and produces 3d cubes of vector field data with arbitrary shaped magnets defined by STL files and user defined magnetic moment angles in 3d space.

Clock Color Spin Vectors(tm) on a White Background (255 255 255)

XY Plane Color RGB values

12:00 oclock light yellow	**(255 255 128)**
1:30 oclock orange	**(255 128 000)**
3:00 oclock aqua	**(000 255 255)**
4:30 oclock purple	**(128 000 255)**
6:00 oclock dark green	**(000 128 064)**
7:30 oclock brown	**(128 064 000)**
9:00 oclock green	**(000 255 000)**
10:30 oclock lavender	**(255 128 255)**
blue z axis into page south	**(000 000 255)**
red z axis out of page north	**(255 000 000)**
pink monopole source	**(255 000 128)**
dark purple monopole sink	**(128 000 128)**

pink and dark purple can be used for electric field modeling

The Pic2Mag Ace 640x640 version may be used by everyone and shared with everyone, including professors, students, and school computer labs, but may not be resold.