

PSTricks

pst-marble v 1.0

A PSTricks package to draw marble-like patterns

October 19, 2018

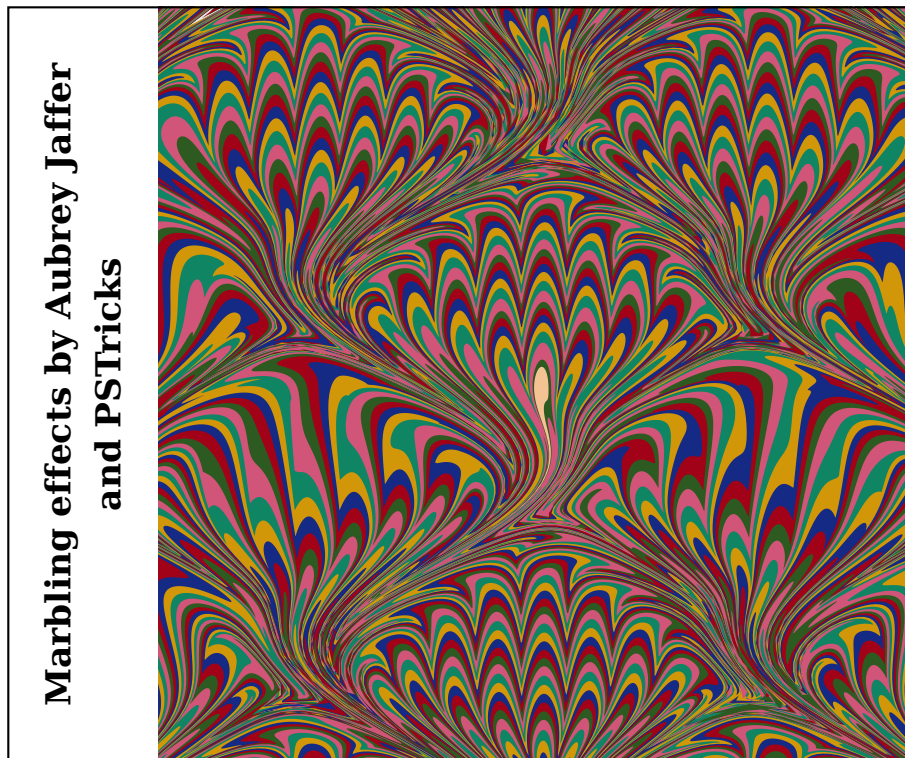
Package author(s):
Aubrey Jaffer
with the help of
Jürgen Gilg
Manuel Luque

Contents

1	Introduction	4
2	Techniques	7
2.1	Drop paint	7
2.2	Random drops	8
2.3	Concentric rings	9
3	The command \psMarble	10
4	Rendering	11
5	Colors	12
6	Basic actions	13
6.1	/drop	13
6.2	random-drops	14
6.3	random-drops-colors	16
6.4	concentric-bands	17
6.5	/rake	21
6.6	/stroke	26
6.7	/stir	28
6.8	/vortex	32
6.9	/wiggle	35
6.10	/offset	40
7	Combined actions – Gallery	41
8	List of all optional arguments for pst-marble	51

Marbling originated in Asia as a decorative art more than 800 years ago and spread to Europe in the 1500s where it was used for endpapers and book covers.

The mathematical fascination with paint marbling is that while rakings across the tank stretch and deform the paint boundaries, they do not break or change the topology of the surface. With mechanical guides, a raking can be undone by reversing the motion of the rake to its original position. Raking is thus a physical manifestation of a homeomorphism, a continuous function between topological spaces (in this case between a topological space and itself) that has a continuous inverse function.



```
\begin{pspicture}(-7,-5)(5,5)
\psMarble[viscosity=1000,background=0.9 0.9 0.9,
  colors={{[0.176 0.353 0.129][0.635 0.008 0.094][0.078 0.165 0.518]
    [0.824 0.592 0.031][0.059 0.522 0.392][0.816 0.333 0.475]}},
  actions={%
    0 0 40 6 concentric-bands
    [ 0 0 20 background color1 /drop]
    [ 0 240 /offset]
    [ 180 [ -460 60 460 {} for ] 40 200 31 /rake]
    [ 90 {0.75 mul sin -120 mul} /wobble]
    [ 180 [-750 -150 450] 40 200 31 /rake ]
    [ 90 {0.75 mul sin 240 mul} /wobble ]
    [ 180 [-450 150 750] 40 200 31 /rake ]
    [ 90 {0.75 mul sin -120 mul} /wobble ]
  }](10,10)
\psframe(-7,-5)(5,5)
\put{90}(-6,0){\parbox{10cm}{\centering\bf\Large Marbling effects by Aubrey Jaffer\ and PSTricks}}
\end{pspicture}
```

This program can be redistributed and/or modified under the terms of the LaTeX Project Public License Distributed from CTAN archives in directory macros/latex/base/lppl.txt; either version 1.3c of the License, or (at your option) any later version.

1 Introduction

Aubrey Jaffer finds a similarity between whirlwinds in the great spot of jupiter and those that appear in some marbled papers.

<http://voluntocracy.blogspot.com/2018/08/>

You can see a swirl on a marbled paper at Wikipedia:

https://fr.wikipedia.org/wiki/Papier_marbr%C3%A9#/media/File:PaperMarbling003France1880Detail.jpg

It is true that in both cases, although at very different scales, the laws of fluid mechanics apply. Aubrey Jaffer's article on the physical and mathematical interpretation of the formation of various types of marbling:

<https://arxiv.org/abs/1702.02106>

Aubrey Jaffer has improved the model shown in the previous version of pst-marble. Now it is closer to reality and more consistent in the choice of units. This version allows to perform more accurate simulations, however with some new parameters, which will be explained.

But then everything will depend on your patience, your talent so that we can exclaim looking at one of your achievements:

"Beautiful, it's a big piece of art that you have done!"

Many articles deal with marbled paper techniques which are used to adorn bindings and book covers.

Here a link to an article devoted to it by the famous *Encyclopaedia of Diderot and D'Alembert*.

https://fr.wikisource.org/wiki/L%E2%80%99Encyclop%C3%A9die/1re_%C3%A9dition/MARBREUR_DE_PAPIER

Aubrey Jaffer and some computer scientists working with him or on their own, tried to understand and model marblings that appear when the artist uses a stylus which he moves the tip on a surface of liquid. As a result in its wake, the drops it encounters get deformed and will also influence the shape of their neighbors according to the properties of the medium (viscosity), the speed of the movement of the stylus, and the nature of its trajectory: line segment, line crossing the whole tank, bow on a circle, ripples, swirls, etc.

The artist can also use a comb (rake) whose spacing between teeth can be adjusted to make more complex figures. These studies follow the laws of fluid mechanics to model and thus be able to create realistic simulations of marbling.

On Aubrey Jaffer's website, we'll find many links concerning the theoretical studies.

<http://people.csail.mit.edu/jaffer/Marbling/>

Compared to the previous version, Aubrey Jaffer has reviewed some parameters:

/vortex now models a Lamb-Oseen vortex. We'll refer to the article he wrote to study the theory:

<http://people.csail.mit.edu/jaffer/Marbling/vortex.pdf>

The documentation illustrates the parameters that are now used:

Center coordinates in mm, circulation in mm^2/s and time in s.

The primitive `/line` has now become `/rake` and allows to represent the obtained image when the artist equips himself with a comb (rake) comprising a certain number of identical teeth of a given diameter. He places the comb perpendicularly to a direction fixed by the angle made with the Oy -axis (the angle is positive clockwise) and moves it with a speed of (V) along the indicated direction or contrary to it, depending on the sign of the parameter `tU`. The positions of the teeth are fixed by the distances (in mm) indicated within the list [between the brackets]—the comb/rake can also comprise only one tooth.

By default, the tank's dimensions are $1\text{ m} \times 1\text{ m}$. The scaling factor of the image is 0.1. All lengths are in mm, velocities (in mm/s), angles (in degrees), angular velocity (in $\text{degrees}/\text{s}$), and viscosity and circulation (in mm^2/s).

For a convex stylus (or tine), D (in mm) is the ratio of its submerged volume to its wetted surface area. For a long cylinder it is its diameter.

Aubrey Jaffer retains 1 global parameter: the dynamic viscosity, see in particular the document “Oseen Flow in Paint Marbling”:

<https://arxiv.org/abs/1702.02106>

There are 10 types of actions defined and ready to use:

```
/drop
random-drops
random-drops-colors
concentric-bands
/rake
/stroke
/stir
/vortex
/wiggle
/offset
```

They make it possible to create a very large variety of marblings with combinations of the various actions.

Initially there are drops of colors that the artist spreads with a brush on the surface (a bit of a hazard, even if they are located in a given region) and whose size depends on the brush. He performs the operation several times with other colors and also brushes of different sizes. These single drops, circular in shape, are placed with the following command

```
cx cy r [bgc] [rgb]
[ 0 0 100 [] [0 0 1] /drop]
```

Note, that the coordinates (cx , cy) of the center of the drop and its radius r are in points, the colors need to be setup in the `rgb-color-system`: (values between 0 and 1). Details are given in the following sections. So this is the first phase: arrange the drops on the surface in several stages with different radii and colors. To facilitate the experimentation of different types of actions, Aubrey Jaffer imagined an initial background obtained by dropping (one after the other) drops of different colors (we can also differentiate their radii) at the same point, they all have the same center, we then obtain an initial background consisting of concentric rings, named “concentric-bands”.

Aubrey Jaffer coded all the possible simulations with the expected deformations (rake, stroke, stir, wiggle, vortex) in pure PostScript and his new code, perfectly structured, and whose use is very

simple, would be enough to itself, if it weren't necessary for each test, to add lines, delete others, save them within the original PostScript file . . .

Therefore, Manuel Luque and Jürgen Gilg have decided to adapt that into PSTricks (with the agreement of Aubrey Jaffer). A `\psMarble` command to switch easily between the different types of actions and add a global viscosity parameter to the PostScript code. There are two ways to calculate and represent the drops.

- We are interested only in their contour whose transformation is calculated after each addition of a new drop and whose interior is colored with its color (each drop retains its color);
- in the second case we consider the surface as a grid of points (square pixels of side 1 pt) and each drop is represented by the points situated between its edges.

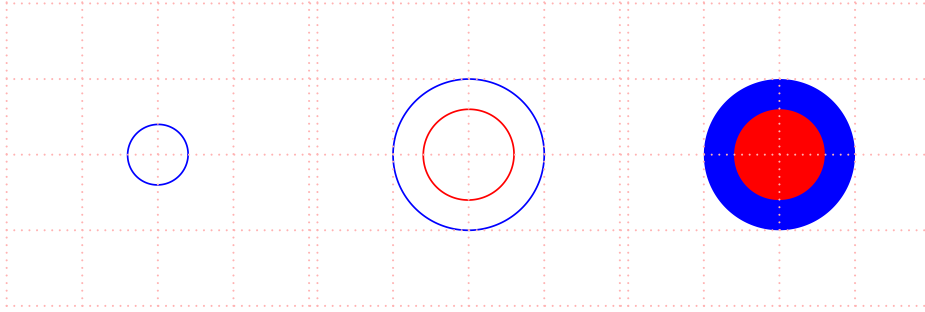
When a new drop is placed, the points in that drop retain their color, the outer points are calculated before being assigned their initial color. This possibility is operational by taking a negative value for the viscosity.

The documentation contains, of course, some more other information than within this short introduction and is likely to be reworked and completed as well as the code.

2 Techniques

2.1 Drop paint

The first drop of paint placed within water forms a circle with the area a . If a second drop with the area b is placed within the center of the first drop, the total area increases from a to $a + b$. For the first drop, points very close to the center will change from an infinitely small radius to a radius $\sqrt{b/\pi}$; and the points on the border of the circle will change from $\sqrt{a/\pi}$ to $\sqrt{(a+b)/\pi}$. If we take 2 drops of different colors, this gives:



The command to drop a drop is written as follows:

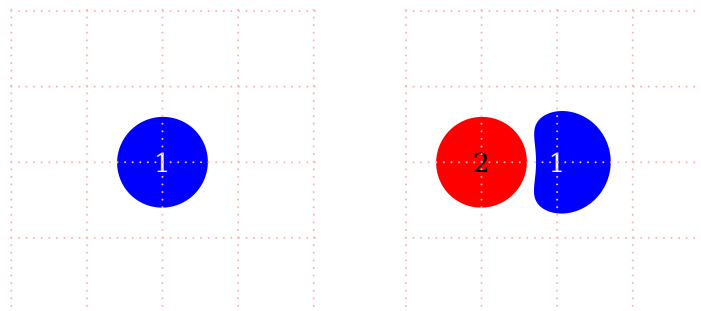
```
cx cy r      r g b
[ 0 0 100 [] [0 0 1] /drop]
```

Note that the coordinates of the center of the drop and its radius are in points¹ and its color is in the system rgb: (values between 0 and 1).

When we place the second drop of radius r at the point $C(cx, cy)$, Aubrey Jaffer considers that this one remains round, intact, but that the first then undergoes the influence of the second and deforms according to the law:

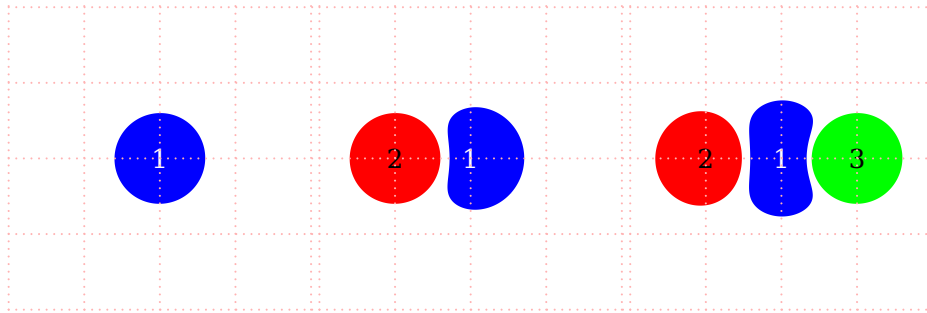
$$\vec{P}' = \vec{C} + \left(\vec{P} - \vec{C} \right) \sqrt{1 + \frac{r^2}{\|\vec{P} - \vec{C}\|^2}}$$

$P(x, y)$ is a point of the first drop and $P'(x', y')$ the transformed point.



If a third drop is placed, the two previous drops will then be influenced by the third, which remains intact.

¹ There is a scaling. Example: if the largest dimension of the page is 4, 100 pts will be represented 0.4 cm



All drops are influenced by the last drop deposited.

2.2 Random drops

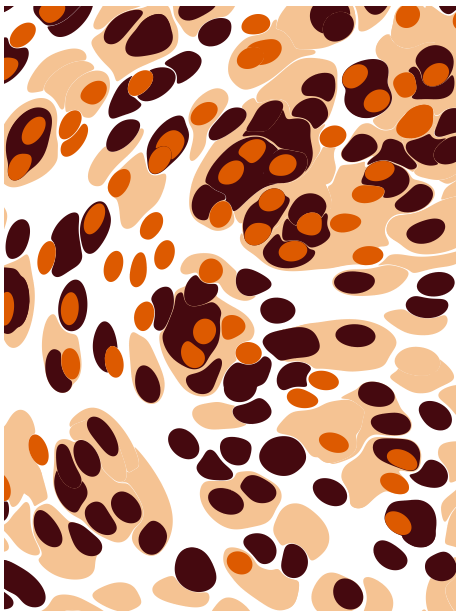
One of the techniques is to project with a brush drops of color on the surface of the liquid in several stages by changing color. The position of the drops is therefore random. Each drop influences its neighbors and assuming that initially the drops would form a round spot on the surface, they will deform depending on the size and proximity of neighbors. The modeling of this phenomenon has been studied in the document “*Mathematical Marbling*” by Shufang Lu, Aubrey Jaffer, Xiaogang Jin, Hanli Zhao and Xiaoyang Mao.

<http://people.csail.mit.edu/jaffer/Marbling/Mathematics>

<https://www.computer.org/csdl/mags/cg/2012/06/mcg2012060026-abs.html>

Then, with a fine stick, a comb one tries to draw the marbling.

The following example illustrates that technique. Three steps with drops of different size and color on which 2 swirls are applied.



```
\begin{pspicture}(-3,-4)(3,4)
\psMarble[%
actions={
250 {
[random:uniform 1000 mul 500 sub % -500 <x<500
random:uniform 1000 mul 500 sub % -500 <y<500
30 background [0.960 0.764 0.576] /drop]
} repeat
200 {
[random:uniform 1000 mul 500 sub % -500 <x<500
random:uniform 1000 mul 500 sub % -500 <y<500
25 background [0.270 0.035 0.058] /drop]
} repeat
100 {
[random:uniform 1000 mul 500 sub % -500 <x<500
random:uniform 1000 mul 500 sub % -500 <y<500
20 background [0.866 0.353 0.000 ] /drop]
} repeat
[ 300 200 -32e2 750 /vortex ]
[ 0 -300 32e2 750 /vortex ]}](6,8)
\end{pspicture}
```

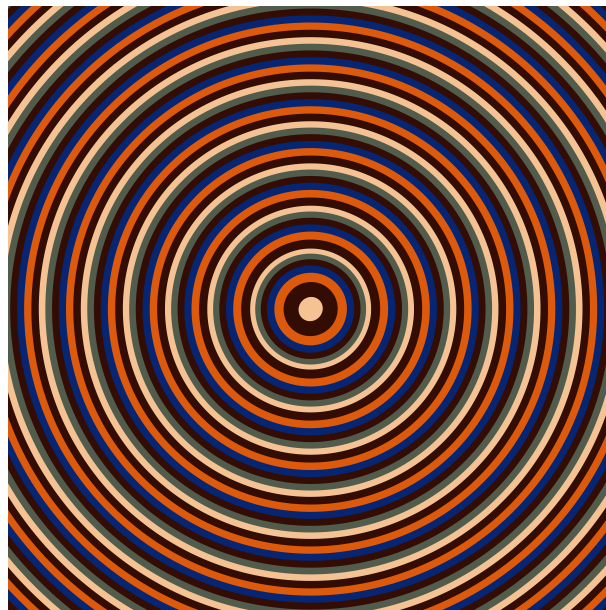
2.3 Concentric rings

Aubrey Jaffer describes the idea of “concentric rings” in:

<http://people.csail.mit.edu/jaffer/Marbling/Mathematics>

“At the start of the real marbling process, paints are dropped from one or more locations to form expanding disks on a substrate. The mathematics is described in [Dropping Paint](#). For now, we just want a paint pattern which shows subsequent displacements. In my first renderings, 5 virtual paints are dropped from the center to form 25 concentric rings of equal radial width.

The boundaries between virtual paint rings will be traversed using the Minsky circle algorithm; although walking the circles using coordinates generated by sin and cos would work as well. The angular step size is made inversely proportional to the ring radius, making the distance between successive points uniform.”



```
\begin{pspicture}(-4,-4)(4,4)
\psMarble(8,8)
\end{pspicture}
```

3 The command \psMarble

`\psMarble [Options] (x,y)`

The command `\psMarble` contains the options `actions=`, `background=`, `bckg=true/false`, `viscosity=`, `drawcontours=true/false` and `colors=`.

The optional argument `(x,y)` gives the x and y dimension of the image—the default is `(10,10)`.

Name	Default	Meaning
actions	0 0 40 colors length concentric-bands	The type of marbling action
background	0 0 0	Background color
bckg	true	Boolean: to turn on/off the background color
colors	[0.960 0.764 0.576]	Colors of the marbling to be set within the rgb-color-system
	[0.316 0.362 0.298]	
	[0.200 0.050 0.015]	
	[0.023 0.145 0.451]	
	[0.866 0.353 0.050]	
drawcontours	false	Boolean: if set to true, it only draws the contours
viscosity	1000	Global primitive: viscosity of the system—if taken negative pixelated image

Notes:

- If `viscosity<0`, the image will be pixelated, if set to positive values it will be non-pixelated.
- The boolean option `drawcontours` is by default set to `false`. If set to `true`, only the contours are drawn within the image.
- Sometimes it is quite helpful to be able to turn off the background color. This can be handled with the boolean key `bckg`, which if set to `false` turns off the background color.

- All colors need to be setup within the rgb-color-system.

If one has a color from another color system, for example:

`0.5 1 1 sethsbcolor` it is sufficient to just add `currentrgbcolor` at its end, like

`0.5 1 1 sethsbcolor currentrgbcolor` to get the equivalent within the rgb-color-system.

- Following are introduced some basic actions, like `/drop`, `random-drops`, `random-drops-colors`, `concentric-bands`, `/rake`, `/stroke`, `/stir`, `/vortex`, `/wiggle` and `/offset`.

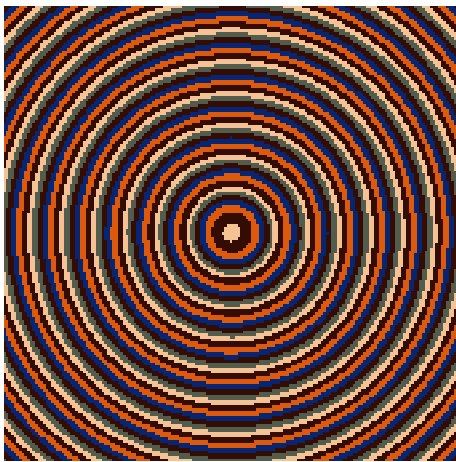
Within the basic actions `/stir` and `/vortex`, there is defined each with a radius `r` parameter. If `r>0` is set, the deformation is counterclockwise, if set to positive values, the deformation is clockwise.

4 Rendering

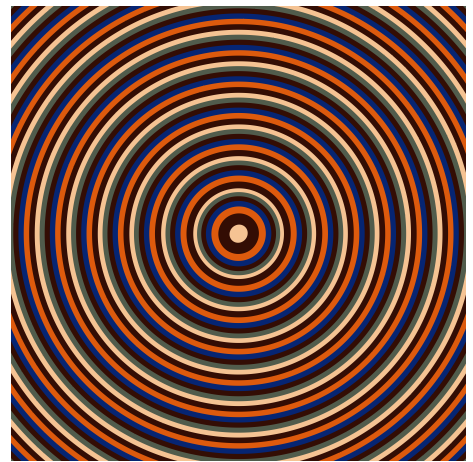
As designs get more complicated, hundreds of drops and strokes, reverse-rendering is the only practical way to render them. As the number of strokes increases, the number of points in the contours needs to increase as well. As the number of drops increases, the time to compute each pixel becomes less than the time to compute each contour-point on the drops.

The reason that we don't always reverse-render is because its resolution is limited to the raster; forward-rendering designs remain crisp at any magnification.

- To use forward-rendering (pixelfree) we choose the option `viscosity>0` with a positive value.
- To use reverse-rendering (pixeled) we choose the option `viscosity<0` with a negative value. When a new drop is placed, the points in that drop retain their color, the outer points are calculated before being assigned their initial color. This possibility is operational by taking for viscosity (characteristic constant) a negative value.



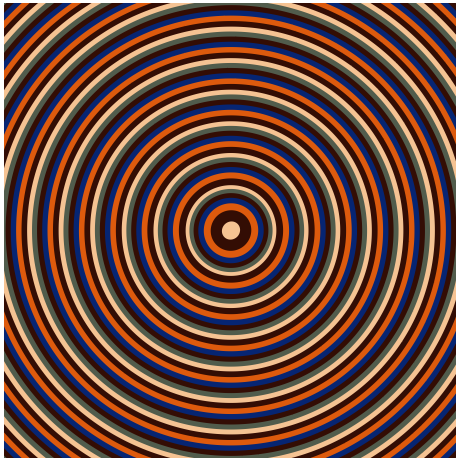
```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[viscosity=-1000](6,6)
\end{pspicture}
```



```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[viscosity=1000](6,6)
\end{pspicture}
```

5 Colors

All colors are setup within the rgb-color-system. Besides the preset colors= which are initially setup within the pst-marble.pro, we can change them within the concentric circles basic figure concentric-bands as follows:

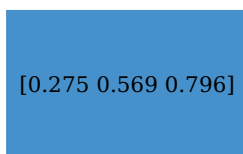


```
\begin{pspicture}(-3,-3)(3,3)
\psMarble(6,6)
\end{pspicture}
```

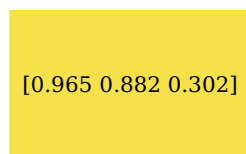


```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[colors={
[0.134 0.647 1.0 ]
[0.977 0.855 0.549]
[0.684 0.638 0.702]
[0.73 0.965 0.942 ]
[0.04 0.236 0.424 ]
}](6,6)
\end{pspicture}
```

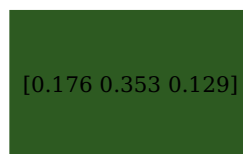
Note: As experience tells, not all colors will print as well as shown within the PDF file, so one has to print the image to see if the colors are OK for a paper. Here a list of colors that print well:



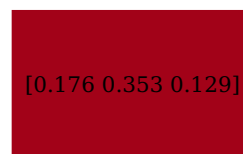
[0.275 0.569 0.796]



[0.965 0.882 0.302]



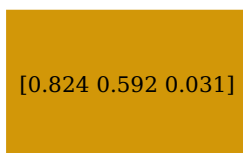
[0.176 0.353 0.129]



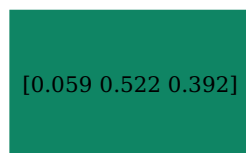
[0.176 0.353 0.129]



[0.078 0.165 0.518]



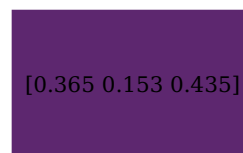
[0.824 0.592 0.031]



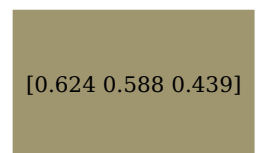
[0.059 0.522 0.392]



[0.816 0.333 0.475]



[0.365 0.153 0.435]



[0.624 0.588 0.439]

6 Basic actions

Some of the deformation actions= which are initially setup within the `pst-marble.pro` can be manually changed by its parameters:

6.1 /drop

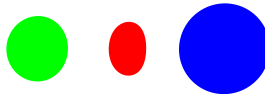
`/drop` defines a single drop set on the surface of a liquid.

```
[ cx cy r [ bgc ] [ rgb ] /drop ]
```

`cx, cy` Center coordinates
`r` Radius of the paint drop
`[bgc]` Background color of paint drop
`[rgb]` Color of paint drop

This initially is a circle with its center at (cx, cy) and a radius r . The paint color is defined by the `rgb-color-system`. The empty slot `[]` is the background drop color.

In order to interpolate the color in reverse-rendering, the adjacent color must be known.



```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[background=1 1 1, %white
actions={%
[ 0 0 50 [] [1 0 0] /drop]
[ -200 0 70 [] [0 1 0] /drop]
[ 200 0 100 [] [0 0 1] /drop]
}](6,6)
\end{pspicture}
```

Note: The paint drop top most on the stack is left undeformed (intact), whereas all the others are influenced by each other, according to the system constant. There are as well 6 colors predefined which can be used like that:

```
[ 0 0 50 [] color1 /drop]
[ -200 0 70 [] color2 /drop]
[ 200 0 100 [] color3 /drop]
```

6.2 random-drops

With random-drops, we randomly place drops onto a liquid surface within an area of (1 m × 1 m) with also randomly varying radii.

The varying radii of the drops are calculated with: $r = (\text{size (in mm)}) \cdot e^x$, with: $e = 2.71828182$ and $0 < x < 1$.

So the radii will vary between $r_{\min} = \text{size}$ and $r_{\max} = 2.71828182 \cdot \text{size}$

```
[ size [ color ] count random-drops ]
```

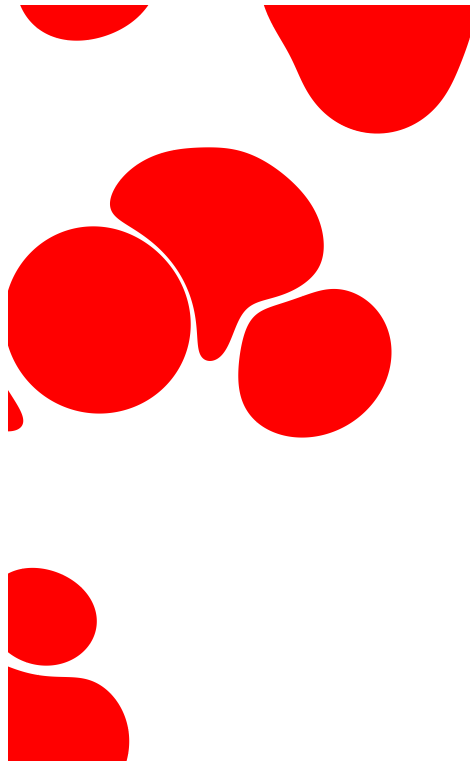
count Number of the drops

size Size of the drops

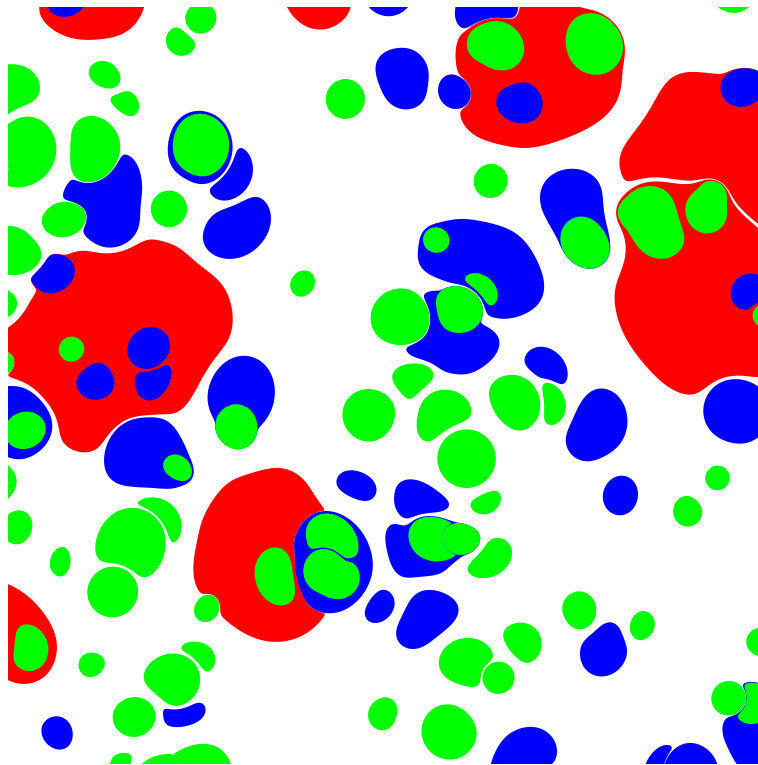
color Color of the drops

Note: If we choose a large number of drops, the compilation time increases.

Example 1:



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
  actions={
% size color count
  50 [1 0 0 ] 10 random-drops
  }](10,10)
\end{pspicture}
```

Example 2:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
  actions={
% size color count
  50 [1 0 0 ] 10 random-drops
% size color count
  20 [0 0 1 ] 50 random-drops
% size color count
  15 [0 1 0 ] 100 random-drops
  }](10,10)
\end{pspicture}

```

6.3 random-drops-colors

With random-drops-colors, we randomly place drops onto a liquid surface within an area of (1 m \times 1 m) with also randomly varying radii and randomly chosen colors.

The varying radii of the drops are calculated with: $r = (\text{size (in mm)}) \cdot e^x$, with: $e = 2.71828182$ and $0 < x < 1$.

So the radii will vary between $r_{\min} = \text{size}$ and $r_{\max} = 2.71828182 \cdot \text{size}$

The varying colors are calculated with:

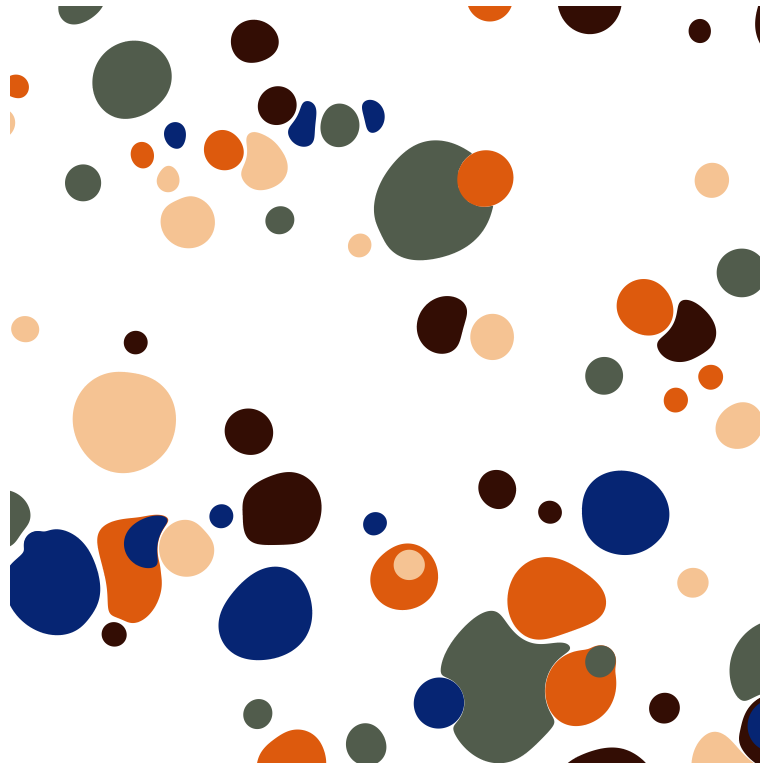
```
[background colors random:uniform colors length mul floor cvi get /drop]
```

```
[ size count random-drops-colors ]
```

count Number of the drops

size Size of the drops

Example:



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
  actions={
% size count
    40 10 random-drops-colors
    25 20 random-drops-colors
    15 50 random-drops-colors
  }](10,10)
\end{pspicture}
```

6.4 concentric-bands

With concentric-bands, we set the number of different colored concentric bands (number of bands) (at center cx, cy) with a radius increment.

The original PostScript code we find within `pst-marble.pro` as:

```
/concentric-bands{
% cx cy:           Center coordinates
% radius increment: Multiplication coefficient between the sets of bands
% number of bands:  Number of drops within one set = number of elements
%                  within the [array of colors] list
%                  [array of colors]: this is the list of colors
%                  within the colors={...} key
    /nbands exch def
    /rinc exch def
    /yc exch def
    /xc exch def
11 -1 1
{
    /rad exch sqrt rinc mul def
    /cnt 0 def
nbands {
    [ xc yc rad background colors cnt colors length mod get /drop]
    /cnt cnt 1 add def
    } repeat
} for
[ xc yc rinc 2 div background colors 0 get /drop]
} def
```

Explanation:

We have 11 sets of drops, within every set, the drops have the same radii and their radii will decrease with every step. The last set is given by the argument `rinc` and all the other radii are a function of this final radius.

Within the first set of drops (with same radii), the number of drops is `nbands`—every drop of it has a color taken from the `colors` array and its radius values $\sqrt{11} \cdot \text{rinc}$.

The sets go from 11 to 1 with a step of 1, meaning that the second set has a radius of $\sqrt{10} \cdot \text{rinc}$ etc.

The last set has a radius of $\sqrt{1} \cdot \text{rinc} = \text{rinc}$.

A last single drop is then added with the radius of $r = \text{rinc} / 2$.

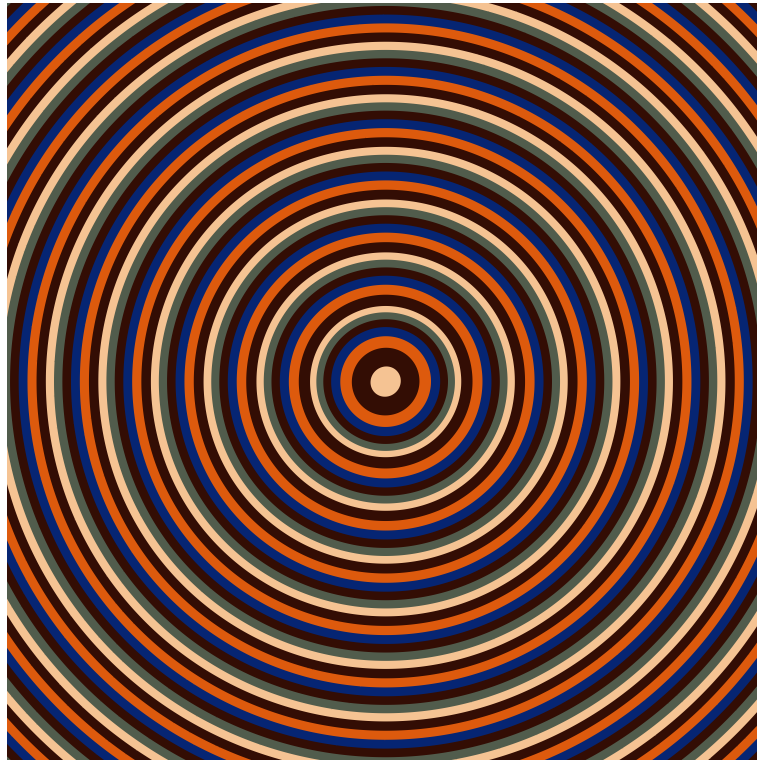
To code it within the \LaTeX file we use the following syntax:

```
[ cx cy RadInc NbrBands concentric-bands ]
```

<code>cx, cy</code>	Center coordinates
<code>RadInc</code>	Multiplication coefficient between the bands
<code>NbrBands</code>	Number of the elements within the [array of colors] list
	[array of colors]: this is the list of colors within the colors= key

Example 1:

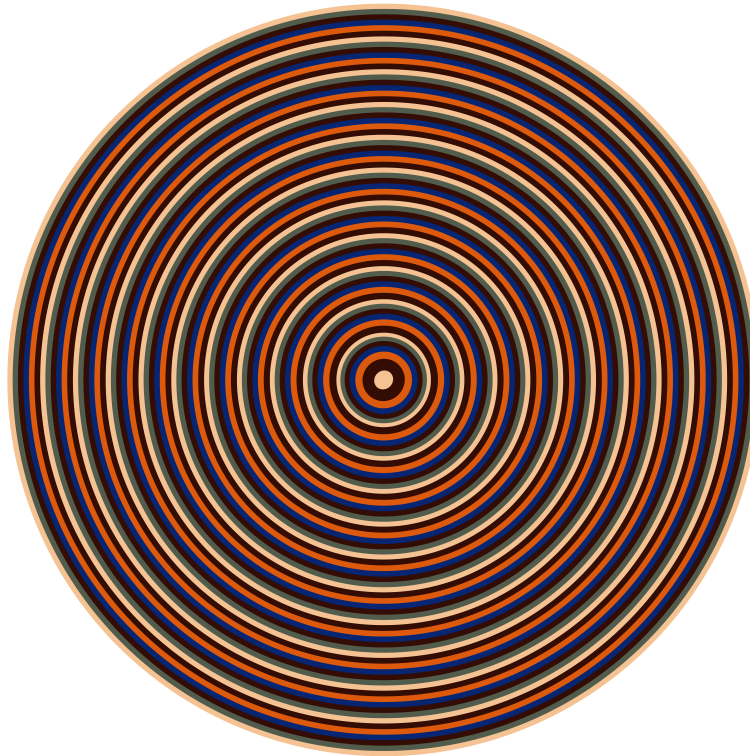
concentric-bands is the default action of the `actions={...}` key, meaning if **no** action is chosen, concentric-bands with its default radius increment and its default color list is in effect.



```
\begin{pspicture}(-5,-5)(5,5)
  \psMarble(10,10)
\end{pspicture}
```

Example 2:

If we want to change RadInc and NbrBands, we do the following:



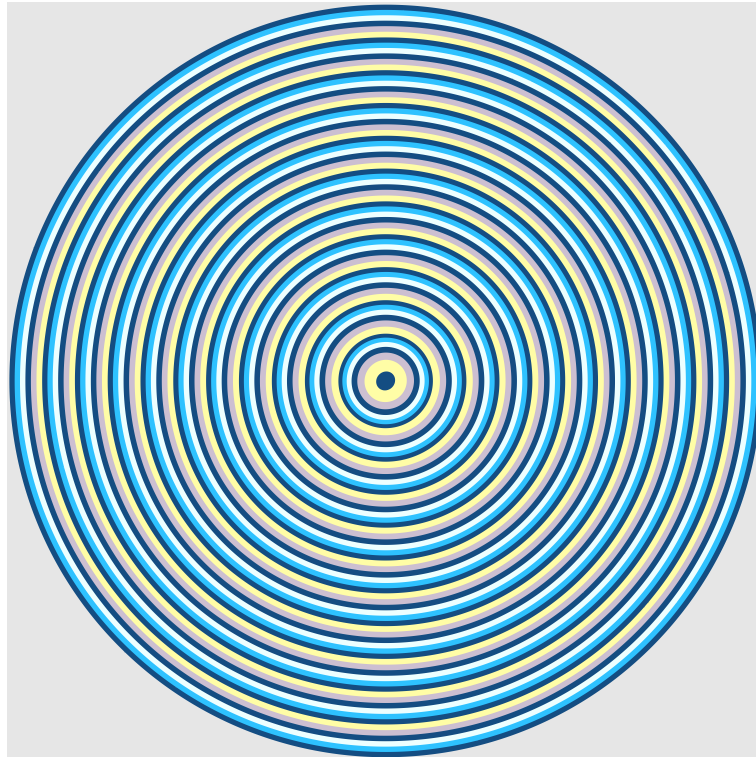
```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  actions={
% cx cy RadInc NbrBands
  0 0 25 6      concentric-bands
  }](10,10)
\end{pspicture}
```

Note: If one increases RadInc to large values > 100 the area of the annuli increases by square, so not all annuli will be shown within the image, however they are calculated, which leads to longer compilation times and increases the final file size.

Typical values are: $35 < \text{RadInc} < 50$

Example 3:

If we like to change the colors as well, we do this with the `colors={...}` key, as follows:



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  background=0.9 0.9 0.9,
  colors={
    [0.08 0.3 0.51 ]
    [0.18 0.76 1]
    [0.93 1 1]
    [0.08 0.3 0.51 ]
    [0.8 0.75 0.82]
    [1 0.99 0.65]
  },
  actions={
%   cx cy RadInc NbrBands
    0 0 25 6 concentric-bands
  }](10,10)
\end{pspicture}
```

Note: If the parameter `NbrBands` is i. e. set to 6, this means that there are six different colors within the bands in order (mod 6), so 6 colors should be setup within the `colors={...}` key. If no colors are individually chosen by the `colors={...}` key, the 6 predefined colors are used by default.

6.5 /rake

This is to represent the image obtained when the artist is equipped with a comb (rake) containing a number of identical teeth of a given diameter. He places the comb perpendicularly to the direction fixed by the angle made with the axis Oy (the angle is counted: if taken positive values—clockwise, if taken negative values—counterclockwise) and moves it with a speed of V in the indicated direction or contrary to it, following the sign of the parameter tU . The positions of the teeth are set up by the distances (in mm) indicated [between brackets], the comb can also have only one tooth.

By default, the tank's dimensions are $1\text{ m} \times 1\text{ m}$. The scaling factor of the image is 0.1. All lengths are in mm, velocities (in mm/s), angles (in degrees), angular velocity (in degrees/s), and viscosity and circulation (in mm^2/s).

For a convex stylus or tine, D (in mm) is the ratio of its submerged volume to its wetted surface area. For a long cylinder it is its diameter.

[angle [r] V tU D /rake]

angle Angle from y-axis in degrees; 0 is up.
 - If angle positive: direction is north-east (>90 south-east).
 - If angle negative: direction is north-west (<-90 south-west).
 [r] List of distances to the teeth of the rake from its center.
 - If r positive: distance to tooth, right to the indicated direction.
 - If r positive: distance to tooth, left to the indicated direction.
 V Stylus velocity in mm/s
 tU Distance between the original points and the deformed points
 along the stylus track.
 - If tU positive: deformation in the indicated direction.
 - If tU negative: deformation contrary to the indicated direction.
 D Stylus diameter in mm. Make larger to affect paint farther away.

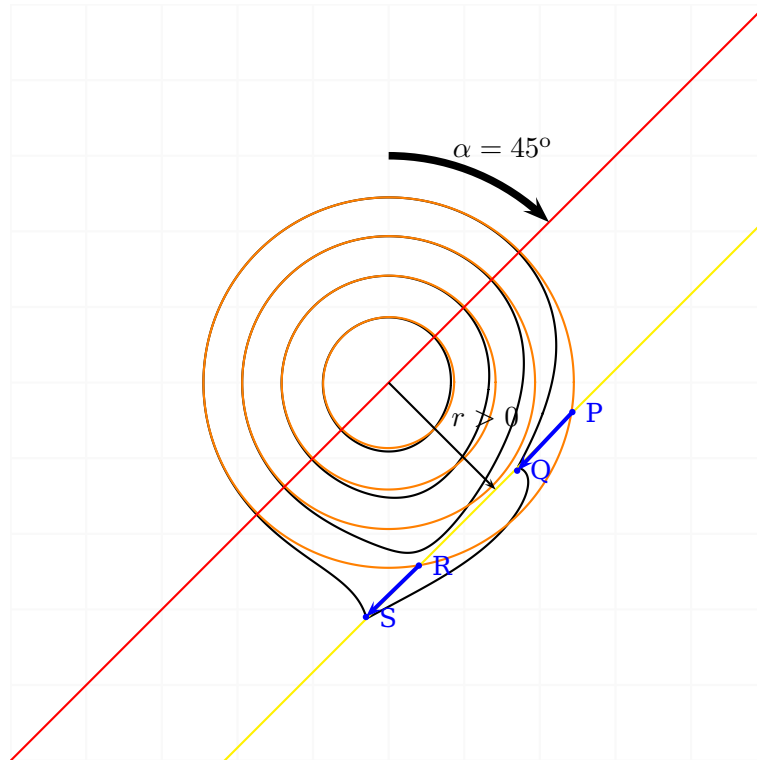
For the following examples viscosity=1000 is set. This is a typical value (default value).

Explanations for the key tU:

Setting: [45 [200] 20 -100 50 /rake]

The orange circles are the ones without deformation. The black ones are the ones after deformation.

- Intersect orange outer circle with the yellow line = P and R
- Intersect the black outer circle with the yellow line = Q and S



- The distance between P and Q is $|\overrightarrow{PQ}|$:
 $tU = |\overrightarrow{PQ}| = 1 \text{ cm}$ with respect to the scaling factor 0.1 for the image, this gives $tU=100$, as it should.
- The distance between R and S is $|\overrightarrow{RS}|$:
 $tU = |\overrightarrow{RS}| = 1 \text{ cm}$ with respect to the scaling factor 0.1 for the image, this gives $tU=100$, as it should.

Note: Within the given example $tU=-100$ was chosen *negative*. This indicates that the deformation is made contrary to the stylus track (set with `angle=45` (at a distance [`r=200`] from the red line) and drawn in yellow, so points to north-east, thus the deformation points move necessarily to south-west.

Example 1: [45 [200] 20 -100 50 /rake]

The angle is $\alpha=45$, means the direction of the stylus track north-east. (If the angle would be chosen to $\alpha=-45$, the stylus track would move north-west.

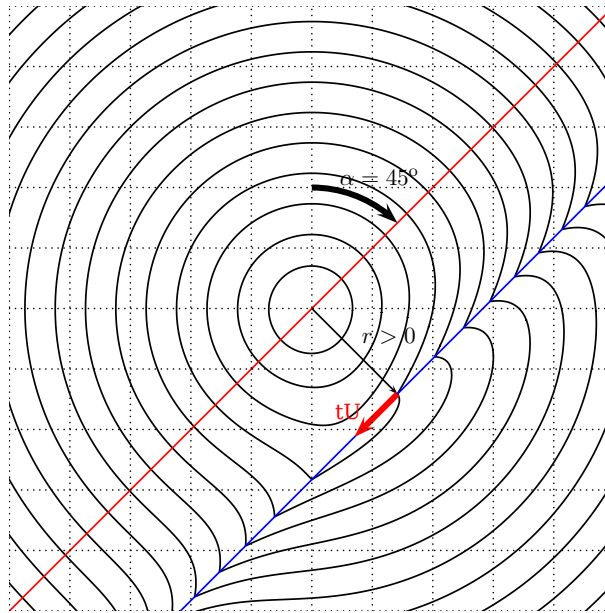
The distance [r=200] (in mm) of one tooth from the center of the rake on the right side referred to the stylus track direction, if r is taken positive; to the left side to the stylus track direction, if r is taken negative.

Note: The scaling factor of the image is 0.1. Thus $200 \text{ mm} \times 0.1 = 2 \text{ cm}$ within the image.

The stylus velocity is given with $V=20$ (in mm/s).

The distance $tU=-100$ between the original points and the deformed points along the stylus track is set to negative (the deformation is made contrary to the to the direction of the stylus track). If taken a positive value for tU , the deformation is made in the direction of the stylus track.

The stylus parameter D (given in mm) is the ratio of its submerged volume to its wetted surface area. The bigger this value, the wider the area of points that are affected by the deformation.



```
\begin{pspicture*}(-5,-5)(5,5)
\psMarble[viscosity=1000,linewidth=0.1,drawcontours,
actions={30 -2 2
{
/rad exch sqrt 50 mul def
[ 0 0 rad background [0 0 0] /drop]
} for
% angle r    V    tU    D
[ 45 [ 200 ] 20 -100 50 /rake]
}](10,10)%
\psline[linecolor=red](-5,-5)(5,5)
\rput(!2 sqrt 2 sqrt neg){\psline[linecolor=blue](-5,-5)(5,5)}
\psarcn[linewidth=0.1]{->}(0,0){2}{90}{45}
\uput{2.1}[67.5](0,0){$\alpha=45^{\circ}$}
\psline{->}(0,0)(!2 sqrt 2 sqrt neg)
\uput[45](0.707,-0.707){$r>0$}
\rput(!2 sqrt 2 sqrt neg){\psline[linecolor=red,linewidth=0.1]{->}(0,0)(1;225)}
\rput(0.6,-1.7){\red tU}
\psgrid[subgriddiv=1,griddots=10,gridlabels=0pt]
\end{pspicture*}
```

Example 2: [45 [200] 20 100 50 /rake]

The angle is $\alpha=45^\circ$, means the direction of the stylus track is north-east.

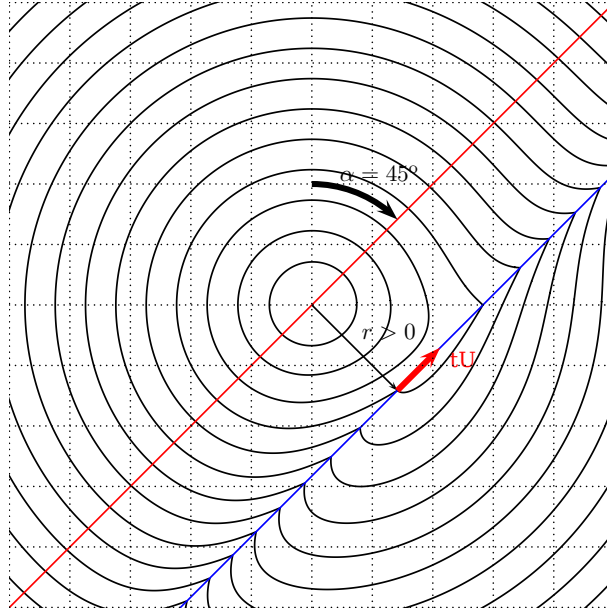
The distance [r=200] of one tooth from the center of the rake on the right side referred to the stylus track direction, if r is taken positive; to the left side to the stylus track direction, if r is taken negative.

Note: The scaling factor of the image is 0.1. Thus $200 \text{ mm} \times 0.1 = 2 \text{ cm}$ within the image.

The stylus velocity is given with $V=20$ (in mm/s).

The distance $tU=100$ between the original points and the deformed points along the stylus track is set to positive (the deformation is made to the direction of the stylus track).

The stylus parameter D (given in mm) is set to 50 mm.



```
\begin{pspicture*}(-5,-5)(5,5)
\psMarble[viscosity=1000,linewidth=0.1,drawcontours,
actions={30 -2 2
{
/rad exch sqrt 50 mul def
[ 0 0 rad background [0 0 0] /drop]
} for
% angle r      V tU    D
[ 45 [ 200 ] 20 100 50 /rake]
}](10,10)%
\psline[linecolor=red](-5,-5)(5,5)
\rput(!2 sqrt 2 sqrt neg){\psline[linecolor=blue](-5,-5)(5,5)}
\psarcn[linewidth=0.1]{->}(0,0){2}{90}{45}
\uput{2.1}[67.5](0,0){$\alpha=45^\circ$}
\psline{->}(0,0)(!2 sqrt 2 sqrt neg)
\uput[45](0.707,-0.707){$r>0$}
\rput(!2 sqrt 2 sqrt neg){\psline[linecolor=red,linewidth=0.1]{->}(0,0)(1;45)}
\rput(2.5,-0.9){\red tU}
\psgrid[subgriddiv=1,griddots=10,gridlabels=0pt]
\end{pspicture*}
```

Example 3: [0 [-500 100 500 {} for] 50 100 30 /rake]

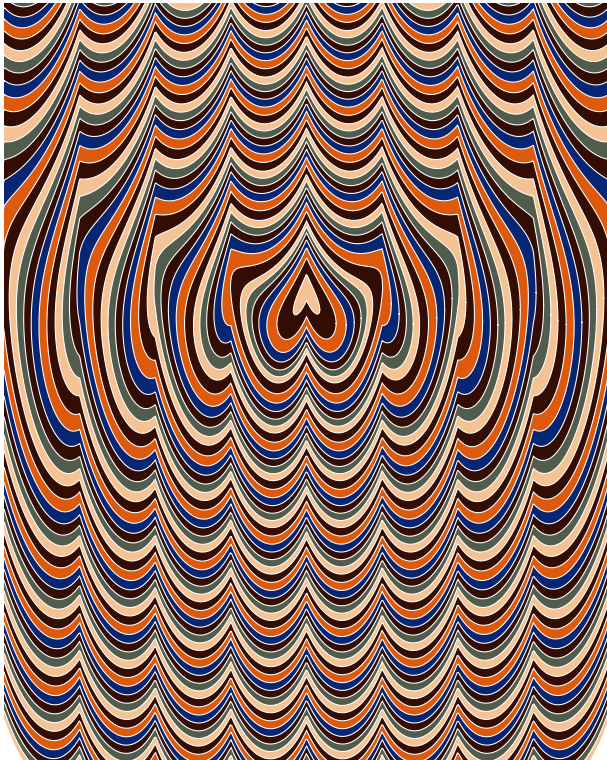
The angle is $\text{angle}=0$, means the direction of the stylus track is north.

The distance [r] is a list of 11 teeth: [-500 100 500 {} for] meaning the distances of the teeth are: -500, -400, -300, -200, -100, 0, 100, 200, 300, 400, 500. Starting at -500 in steps of 100 up to 500. (5 to the left of the indicated direction (the ones with negative values) and 5 to the right of the indicated direction (the ones with the positive values) and one tooth within the center.

The stylus velocity is given with $V=50$ (in mm/s).

The distance $tU=100$ between the original points and the deformed points along the stylus track is set to positive (the deformation is made to the direction of the stylus track).

The stylus parameter D (given in mm) is set to 30 mm.



```
\begin{pspicture}(-4,-5)(4,5)
\psset{viscosity=1000}%
\psMarble[
actions={%
% cx cy radinc count
0 0 35 6 concentric-bands
% angle r V tU D
% rake with 11 teeth aligned 1 cm (= 100 pts)
[ 0 [-500 100 500 {} for] 50 100 30 /rake]
}](8,10)%
\psMarble[
linewidth=0.05,
colors=[[1 1 1]],
bckg=false,
drawcontours,
actions={%
% cx cy radinc count
0 0 35 6 concentric-bands
% angle r V tU D
% rake with 11 teeth aligned 1 cm (= 100 pts)
[ 0 [-500 100 500 {} for] 50 100 30 /rake]
}](8,10)%
\end{pspicture}
```

Note: Within this example two `\psMarble` commands are used! The second command of `\psMarble` is used to highlight the contours of the deformations in white color. Therefore it is needed to suppress the background color for this second command—which can be done with `bckg=false`.

6.6 /stroke

The documentation is on the web page:

<http://people.csail.mit.edu/jaffer/Marbling/Mathematics>

Parameters for /stroke-deformation: bx, by, ex, ey, V, D

[bx by ex ey V D /stroke]

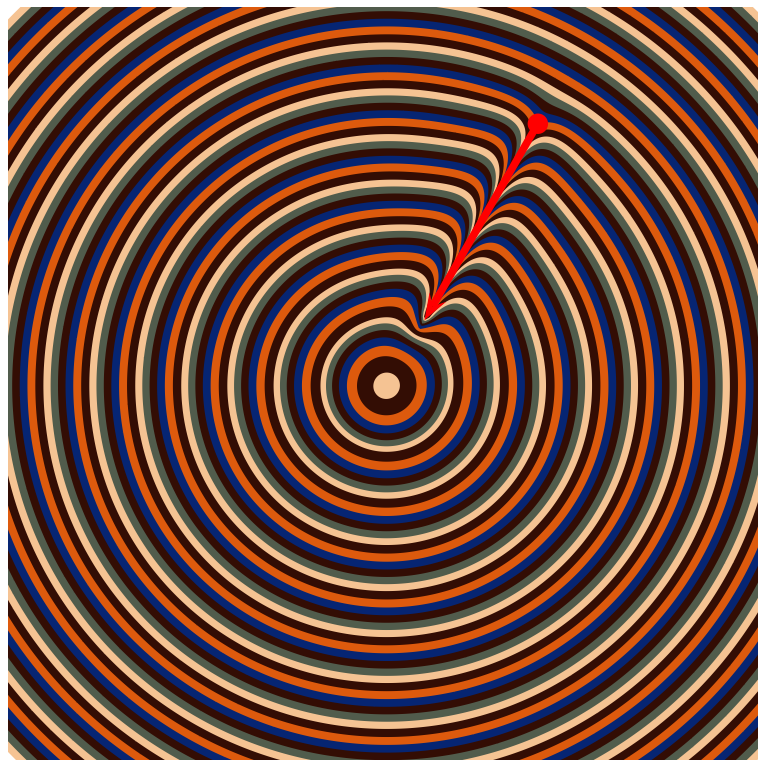
bx, by Beginning of stroke

ex, ey End of stroke

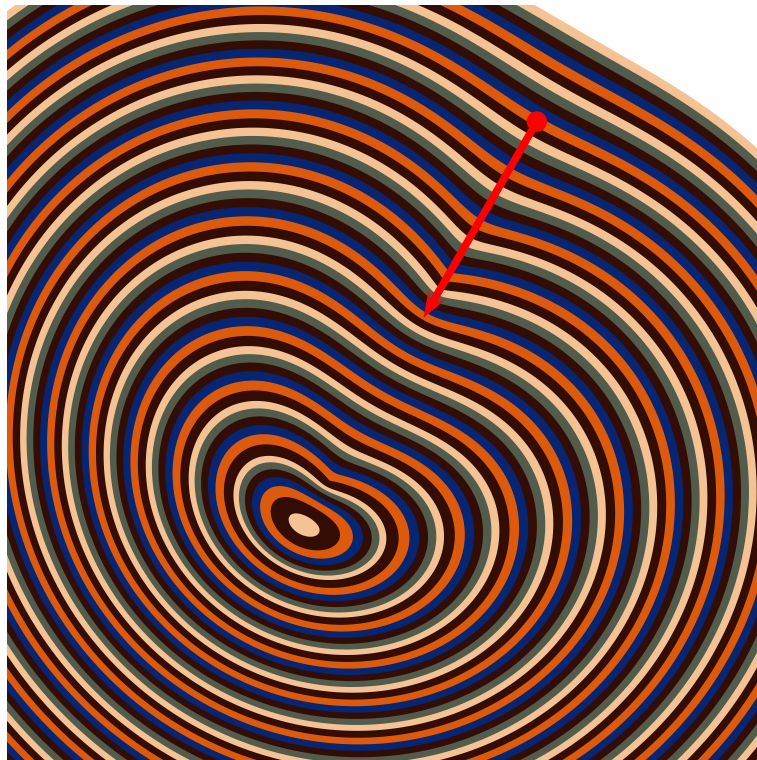
V Stylus velocity in mm/s

D Stylus diameter in mm. Make larger to affect paint farther away.

Example 1:



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=250,
actions={%
0 0 35 6 concentric-bands
% X1 begin Y1 begin X2 end Y2 end V D
[ 30 sin 400 mul 30 cos 400 mul 30 sin 100 mul 30 cos 100 mul 50 10 /stroke ]
}](10,10)
\psset{linecolor=red,linewidth=0.1}
\pstVerb{/scaleFactor 10 1000 div def
/X1 30 sin 400 mul scaleFactor mul def /Y1 30 cos 400 mul scaleFactor mul def
/X2 30 sin 100 mul scaleFactor mul def /Y2 30 cos 100 mul scaleFactor mul def
}%
\psdot(!X1 Y1)\psline{->}(!X1 Y1)(!X2 Y2)
\end{pspicture}
```

Example 2:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=250,
actions={%
% cx cy radinc count
  0 0 35 6 concentric-bands
% X1 begin Y1 begin X2 end Y2 end V D
[ 30 sin 400 mul 30 cos 400 mul 30 sin 100 mul 30 cos 100 mul 200 30 /stroke ]
}](10,10)
\psset{linecolor=red,linewidth=0.1}
\pstVerb{/scaleFactor 10 1000 div def
/X1 30 sin 400 mul scaleFactor mul def
/Y1 30 cos 400 mul scaleFactor mul def
/X2 30 sin 100 mul scaleFactor mul def
/Y2 30 cos 100 mul scaleFactor mul def
}%
\psdot(!X1 Y1)(!X2 Y2)
\psline{->}(!X1 Y1)(!X2 Y2)
\end{pspicture}

```

6.7 /stir

Consider a single stylus (a cylinder of diameter D) that is placed on the Oy -axis at a distance r from the chosen center. In a circular motion, the artist rotates the stylus by an angle θ . The angular velocity will influence the shape of the deformation. The direction of rotation is fixed by the sign of θ . If $\theta > 0$ the artist rotates counterclockwise².

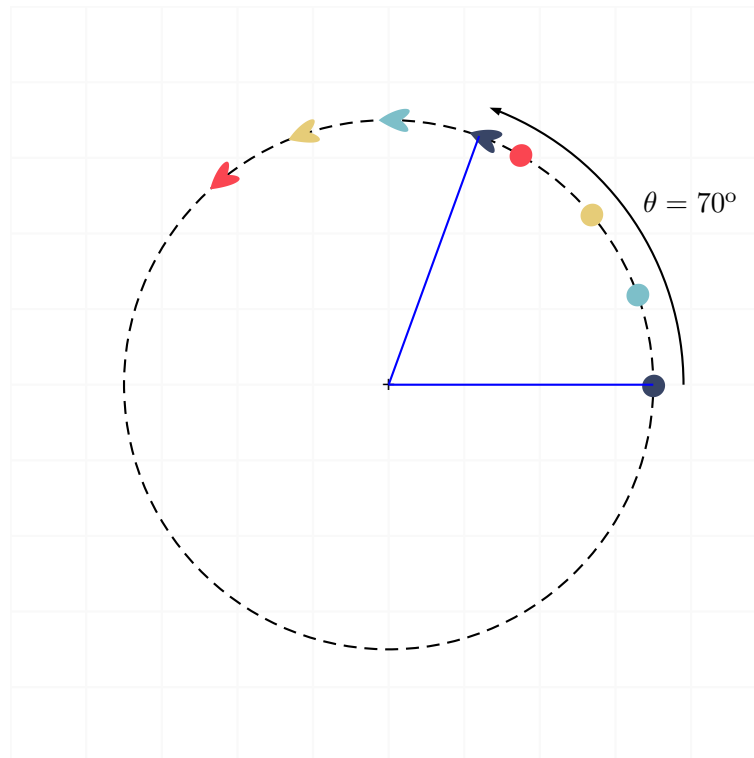
```
[ cx cy [ r ] w th D /stir ]
```

cx, cy Center coordinates in mm.
 $[r]$ List of radii in mm.
 w Angular velocity in degrees/s.
 $th=theta$ Angle displacement at times in degrees.
 D Tine diameter in mm.

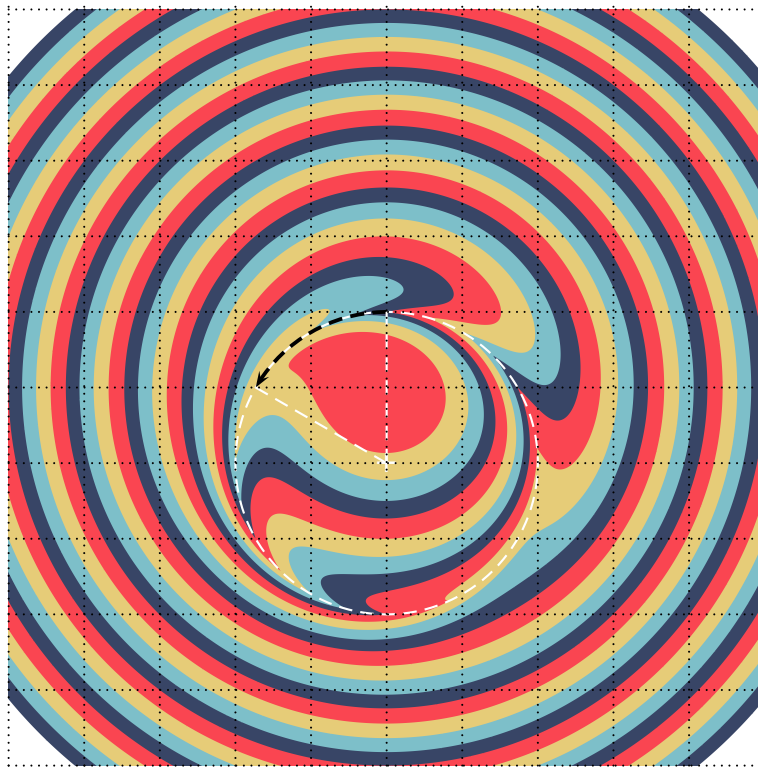
Explanations for the key th:

Setting: `[0 0 [350] 10 70 10 /stir]`

All points on the circle are rotated by $th=70$. There is **no** partial stir operation.



² The sign of r can also indicate the direction of rotation. If $r < 0$ the direction of the rotation fixed by θ inverts.

Example 1:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  actions={
    9 -1 3
    {
      /rad exch sqrt 50 mul def
      [ 0 0 rad background [0.22 0.27 0.4] /drop]
      [ 0 0 rad background [0.49 0.75 0.79] /drop]
      [ 0 0 rad background [0.9 0.8 0.47] /drop]
      [ 0 0 rad background [0.98 0.27 0.317]/drop]
    } for
    [ 0 -100 [ 200 ] 10 60 30 /stir ]
  }](10,10)
\psdot[dotstyle=+,linecolor=white,linewidth=2pt](0,-1)
\pscircle[linestyle=dashed,linecolor=white](0,-1){2}
\psarc[linewidth=0.05,linestyle=dashed]{->}(0,-1){2}{90}{150}
\psline[linecolor=white,linestyle=dashed](0,-1)(0,1)
\psline[linecolor=white,linestyle=dashed](0,-1)(-1.732,0)
\psgrid[subgriddiv=1,griddots=10,gridlabels=0pt]
\end{pspicture}

```

Example 2:

If the artist repeats the same gesture several times, a whirlwind effect is created:



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  actions={
    9 -1 3
  }
/rad exch sqrt 50 mul def
[ 0 0 rad background [0.22 0.27 0.4] /drop]
[ 0 0 rad background [0.49 0.75 0.79] /drop]
[ 0 0 rad background [0.9 0.8 0.47] /drop]
[ 0 0 rad background [0.98 0.27 0.317]/drop]
} for
5{[ 0 -100 [ 200 ] 10 60 30 /stir ]} repeat
  }(10,10)
\psgrid[subgriddiv=1,griddots=10,gridlabels=0pt]
\end{pspicture}
```

Example 3:

The artist turns from two different centers, changing the direction of rotation.

Note: Doing multiple deformations, the order of them is of importance! See the following examples placed next to each other where only the order of deformations is changed.



```
\begin{pspicture}(-4,-5)(4,5)
\psMarble[
actions={
9 -1 3
{
/rad exch sqrt 50 mul def
[ 0 0 rad background [0.22 0.27 0.4] /drop]
[ 0 0 rad background [0.49 0.75 0.79] /drop]
[ 0 0 rad background [0.9 0.8 0.47] /drop]
[ 0 0 rad background [0.98 0.27 0.317]/drop]
} for
[ 0 200 [ 200 ] 10 60 30 /stir ]
[ 0 -200 [ 200 ] -10 60 30 /stir ]
}](8,10)
\end{pspicture}
```



```
\begin{pspicture}(-4,-5)(4,5)
\psMarble[
actions={
9 -1 3
{
/rad exch sqrt 50 mul def
[ 0 0 rad background [0.22 0.27 0.4] /drop]
[ 0 0 rad background [0.49 0.75 0.79] /drop]
[ 0 0 rad background [0.9 0.8 0.47] /drop]
[ 0 0 rad background [0.98 0.27 0.317]/drop]
} for
[ 0 -200 [ 200 ] -10 60 30 /stir ]
[ 0 200 [ 200 ] 10 60 30 /stir ]
}](8,10)
\end{pspicture}
```

6.8 /vortex

```
[ cx cy circ t /vortex ]
```

cx, cy Center coordinates in mm.

circ Circulation (in mm²/s) is a simple scale factor.

Typical value: 30e3 mm²/s.

t Time after circulation impulse at center. As t gets very large, the whole surface returns to its original pattern, possibly with rigid rotation. Typical value 10 s.

/vortex is modeled by a Lamb-Oseen vortex. We refer to the article written by Aubrey Jaffer:

<http://people.csail.mit.edu/jaffer/Marbling/vortex.pdf>

The documentation illustrates the used parameters:

center coordinates in mm, circulation mm²/s and the time s.

After a long enough time, the whole surface returns to its initial state. This can be proofed within an animation to be compiled by the following verbatim code:

```
\documentclass{article}
\usepackage{pst-marble,animate}
\begin{document}
\begin{animateinline}[%
    controls,palindrome,
    begin={\begin{pspicture}(-5,-5)(5,5)},
    end={\end{pspicture}}
]{5}% 5 image/s
\multiframe{20}{rA=-3+0.65}{%
\psMarble[
    actions={%
        0 0 40 6 concentric-bands
        [ 0 0 20 background color1 /drop]
        [ 90 [ -400 400 600 {} for ] 40 200 31 /rake]
        [-90 [ -600 400 600 {} for ] 40 200 31 /rake]
        [0 0 -25200 5 \rA\space exp /vortex]
    }](10,10)
}
\end{animateinline}
\end{document}
```

Animated gifs can be seen at:

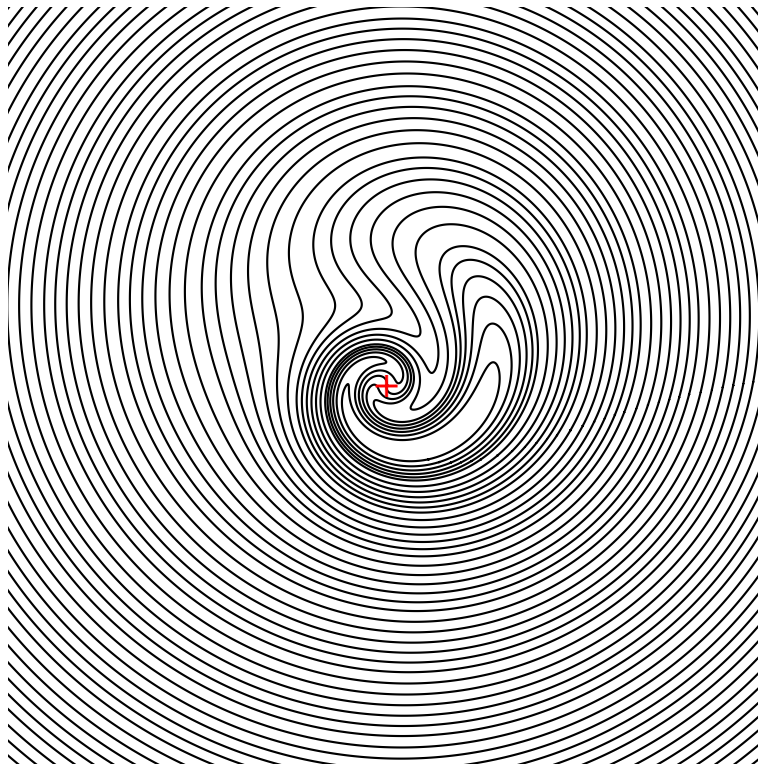
<http://people.csail.mit.edu/jaffer/Marbling/>

Example 1:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
          background=1 1 1,
          actions= {
            9 -1 1
            {
              /rad exch sqrt 50 mul def
              [ 0 0   rad background [0.08 0.3 0.51 ] /drop]
              [ 0 0   rad background [0.18 0.76 1] /drop]
              [ 0 0   rad background [0.93 1 1] /drop]
              [ 0 0   rad background [0.08 0.3 0.51 ] /drop]
              [ 0 0   rad background [0.8 0.75 0.82] /drop]
              [ 0 0   rad background [1 0.99 0.65] /drop]
            } for
            [0 100 /offset]
            [ 0 0 -32e3 10 /vortex ]
          }](10,10)
\end{pspicture}

```

Example 2:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,drawcontours,
linewidth=0.1,
background=1 1 1,
actions= {
9 -1 1
{
/rad exch sqrt 50 mul def
[ 0 0 rad background [0 0 0 ] /drop]
[ 0 0 rad background [0 0 0 ] /drop]
[ 0 0 rad background [0 0 0 ] /drop]
[ 0 0 rad background [0 0 0 ] /drop]
[ 0 0 rad background [0 0 0 ] /drop]
[ 0 0 rad background [0 0 0 ] /drop]
[ 0 0 rad background [0 0 0 ] /drop]
} for
[0 100 /offset]
[ 0 0 -32e3 10 /vortex ]
}](10,10)
\psdot[dotstyle=+,dotsize=0.25,linecolor=red](0,0)
\end{pspicture}

```

6.9 /wobble

This instruction simulates the action of an artist who with the tip of the stylus draws undulations on the surface of the liquid.

```
[ angle {func} /wobble ]
```

angle Wobble will be perpendicular to angle from y-axis up.

{func} Function that "wobbles" the original image.

The direction is defined by the angle (we call it $\alpha = \text{angle}$) with respect to the y -axis upwards; a positive value of α points clockwise. (dx, dy) represents the unit vector in the desired deformation direction, $(u_x = \cos \alpha, u_y = \sin \alpha)$.

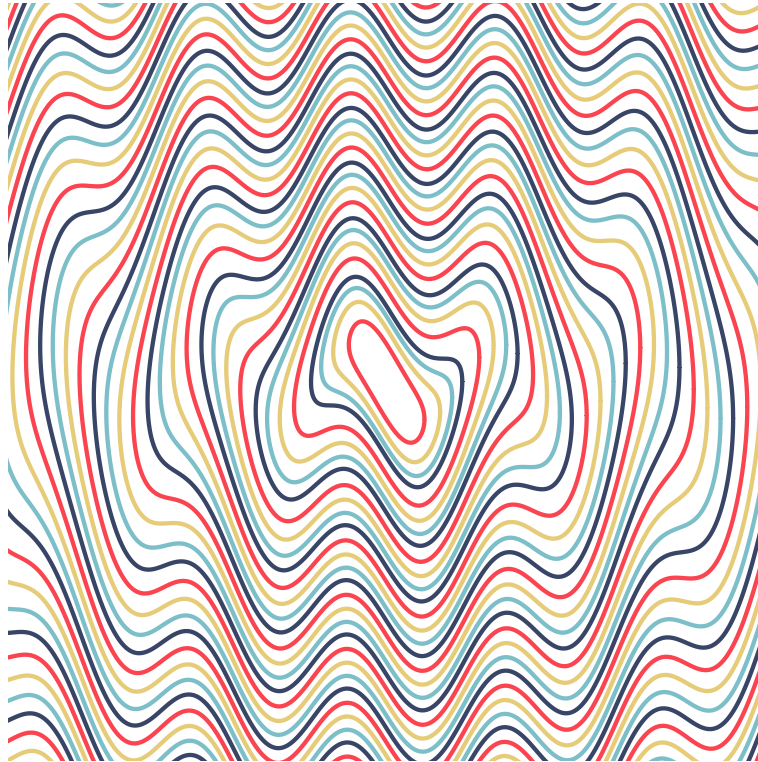
$$a = f(yu_x - xu_y) \implies x' = x + au_x; y' = y + au_y$$

Example 1:

If one wishes to obtain a sinusoidal undulation parallel to the axis Oy , we set $\alpha = 0$. In this case ($u_x = 1, u_y = 0$), and a function i. e., a sinusoidal function with amplitude 50 and angular velocity $\omega = 5$: $f(x, y) = 50 \sin(5y)$ that is coded: {5 mul sin 50 mul}, we will have:

$$x' = x + 50 \sin(5y); y' = y.$$

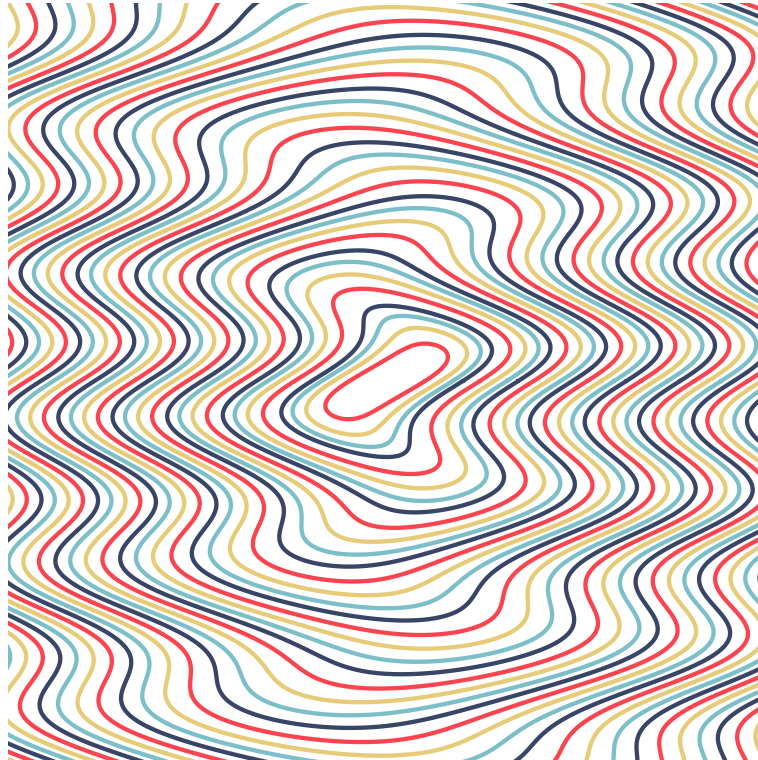
It is recalled that the coordinates are in mm. If on the interval $-500 < x < 500$ we want 5 *periods*, we will take as angular velocity: $\omega = 5 \times 360/1000 = 1.8$



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[drawcontours,linewidth=0.2,
actions={
10 -1 1
{
/rad exch sqrt 50 mul def
[ 0 0 rad background [0.22 0.27 0.4] /drop]
[ 0 0 rad background [0.49 0.75 0.79] /drop]
[ 0 0 rad background [0.9 0.8 0.47] /drop]
[ 0 0 rad background [0.98 0.27 0.317]/drop]
} for
[ 0 {1.8 mul sin 50 mul} /wobble]
}](10,10)
\end{pspicture}
```


Example 2:

For a sinusoidal undulation parallel to the axis Ox , we set $\alpha = 90$. In that case ($u_x = 0, u_y = 1$), the function becomes $f(x, y) = 50 \sin(5x)$ and is coded like that: {5 mul sin 50 mul}, thus:
 $x' = x; y' = y + 50 \sin(5x)$.



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[drawcontours,linewidth=0.2,
  actions={
    10 -1 1
    {
/rad exch sqrt 50 mul def
    [ 0 0 rad background [0.22 0.27 0.4] /drop]
    [ 0 0 rad background [0.49 0.75 0.79] /drop]
    [ 0 0 rad background [0.9 0.8 0.47] /drop]
    [ 0 0 rad background [0.98 0.27 0.317]/drop]
    } for
    [ 90 {1.8 mul sin 50 mul} /wobble]
  }](10,10)
\end{pspicture}
```

Example 3:

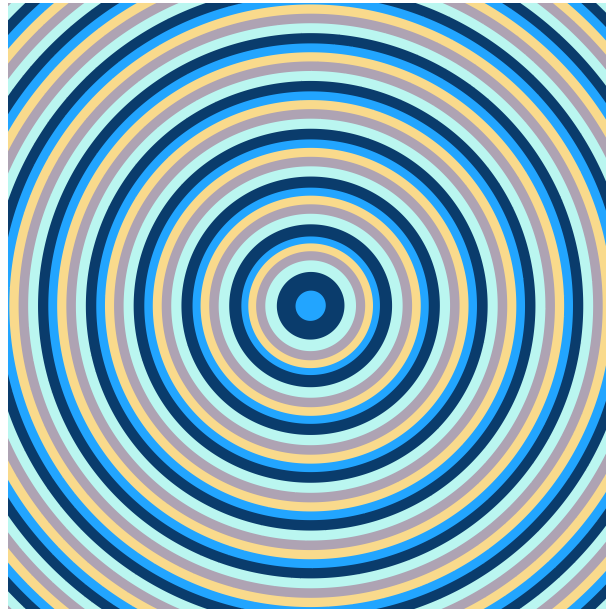
For a sinusoidal undulation in direction of the line $y = x$, we set $\alpha = 45^\circ$:



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  actions={
    10 -1 1
    {
/rad exch sqrt 50 mul def
      [ 0 0 rad background [0.22 0.27 0.4] /drop]
      [ 0 0 rad background [0.49 0.75 0.79] /drop]
      [ 0 0 rad background [0.9 0.8 0.47] /drop]
      [ 0 0 rad background [0.98 0.27 0.317]/drop]
    } for
    [ 45 {2.5 mul sin 50 mul} /wobble]
%    [ -45 {2.5 mul sin 50 mul} /wobble]
  }](10,10)
\end{pspicture}
```

Example 4:

Note: These transformations can be combined and also be reversed, example:



```
\begin{pspicture}(-4,-4)(4,4)
\psMarble[
  colors={
    [0.134 0.647 1.0  ]
    [0.977 0.855 0.549]
    [0.684 0.638 0.702]
    [0.73 0.965 0.942  ]
    [0.04 0.236 0.424  ]
  },
  actions={%
    0 0 50 5 concentric-bands
    [0 {3 mul sin 30 mul } /wobble ]
    [0 {3 mul sin -30 mul } /wobble ] % reverse action
  }](8,8)
\end{pspicture}
```

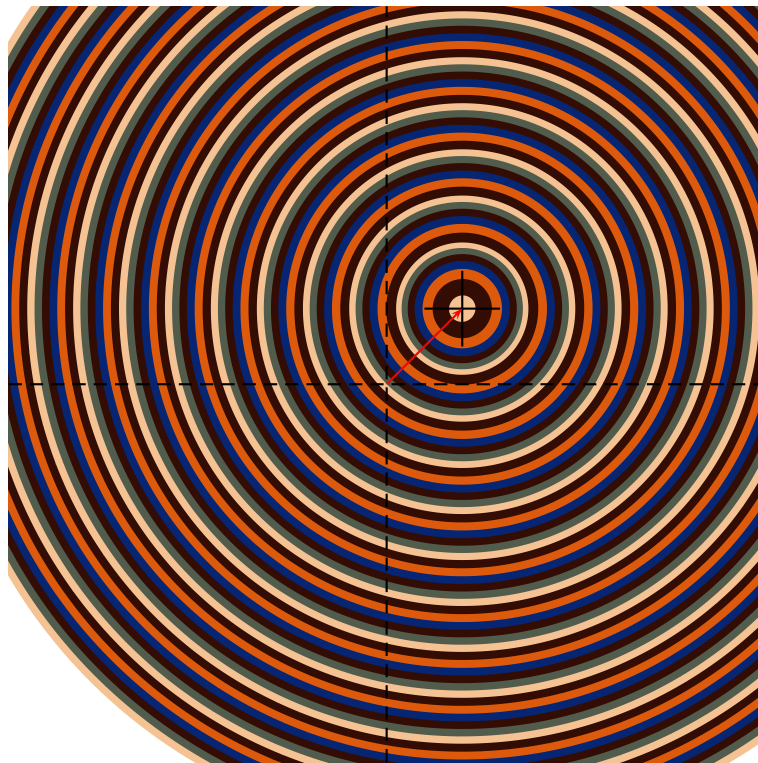
6.10 /offset

[dx dy /offset]

dx Displacement of the image in x direction

dy Displacement of the image in y direction

The choice of units was made, so that 100 pts correspond to 1 cm within the image. dx, dy is the translation vector with which the image is shifted.



The displacement vector is given by dx and dy in pts.

```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=50,
  actions={%
    0 0 35 6 concentric-bands
%    dx dy
    [100 100 /offset]]]
\pstVerb{/scaleFactor 10 1000 div def
/xS1 100 scaleFactor mul def
/yS1 100 scaleFactor mul def
}%
\rput(!xS1 yS1){\psline(0.5,0)(-0.5,0)\psline(0,0.5)(0,-0.5)}
\psline[linestyle=dashed](-5,0)(5,0)
\psline[linestyle=dashed](0,-5)(0,5)
\psline[linecolor=red]{->}(0,0)(1,1)
\end{pspicture}
```

7 Combined actions – Gallery

Example 1:



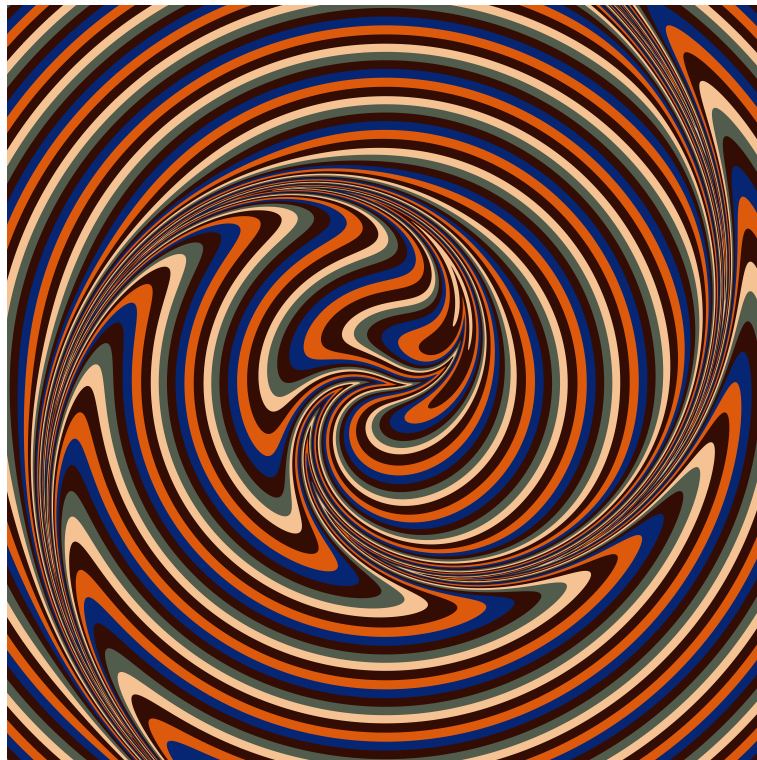
```
\begin{pspicture*}(-4,-1)(4,12)
\psMarble[viscosity=1000, background=0.9 0.9 0.9,
  actions={%
    0 0 45 6  concentric-bands
    [200 -100 /offset]
    -100 20 140
    { /idx exch def
      [-270 idx sub -30 idx 2 mul add [ -270 idx 3 mul sub ] 10 -90 50 /stir ]
      } for
      [90 {2 mul sin 10 mul} /wobble ]
      }](12,24)
\end{pspicture*}
```

Example 2:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
  background=0.9 0.9 0.9,
  actions={
    -400 200 400 {/cy exch def
    -400 200 400 {/cx exch def
    [ cx cy 75 background [0.22 0.27 0.4] /drop]
      } for
    } for
    -400 200 400 {/cy exch def
    -400 200 400 {/cx exch def
    [ cx cy 75 background [0.49 0.75 0.79] /drop]
      } for
    } for
    -400 200 400 {/cy exch def
    -400 200 400 {/cx exch def
    [ cx cy 75 background [0.9 0.8 0.47] /drop]
      } for
    } for
    -400 200 400 {/cy exch def
    -400 200 400 {/cx exch def
    [ cx cy 50 background [0.98 0.27 0.317] /drop]
      } for
    } for
  [ 180 [-500 100 500 {} for] 50 100 30 /rake]
  ](10,10)
\end{pspicture}

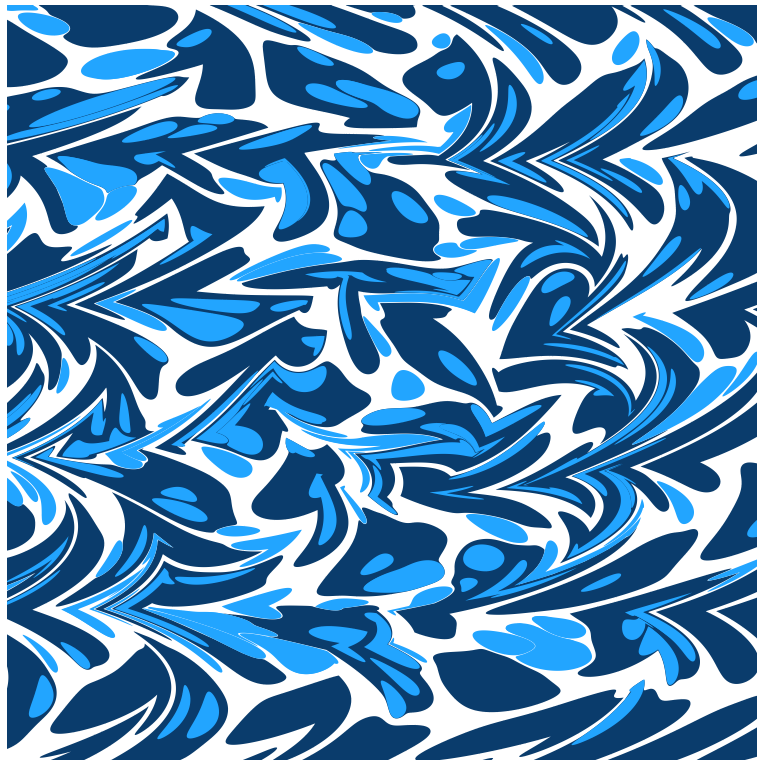
```

Example 3:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  actions= {
    0 0 40 colors length concentric-bands
    [ 0 [ 0 ] 40 200 31 /rake ]
    [ 0 0 -32e3 750 /vortex ]
  }](10,10)
\end{pspicture}

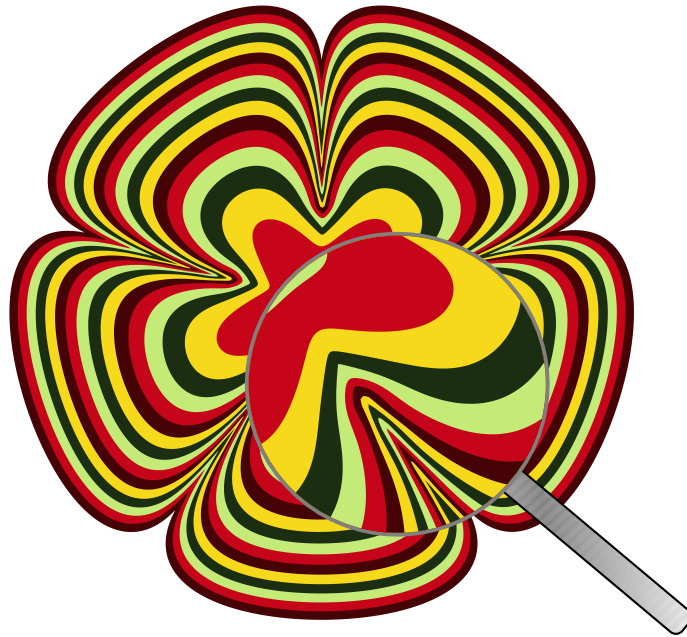
```


Example 4:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=500,
  actions= {
    -500 100 500
  {
/idy exch def
-500 100 500
{
    /idx exch def
    [ idx idy 55 [] [0.040 0.236 0.424] /drop ]
} for
  } for
    250 {
[ rand 1000 mod 500 sub % -800<x<800
rand 1000 mod 500 sub % -400<y<400
rand 15 mod 10 add % 10<r<20
background [0.134 0.647 1 ] /drop]
  } repeat
[ 45 [ -400 200 600 {} for ] 40 200 31 /rake]
[ 100 0 [ -350 ] 30 30 15 /stir ]
[-90 [ -500 200 600 {} for ] 40 200 31 /rake]
[ 0 0 [ -150 ] 60 30 15 /stir ]
  ](10,10)
\end{pspicture}

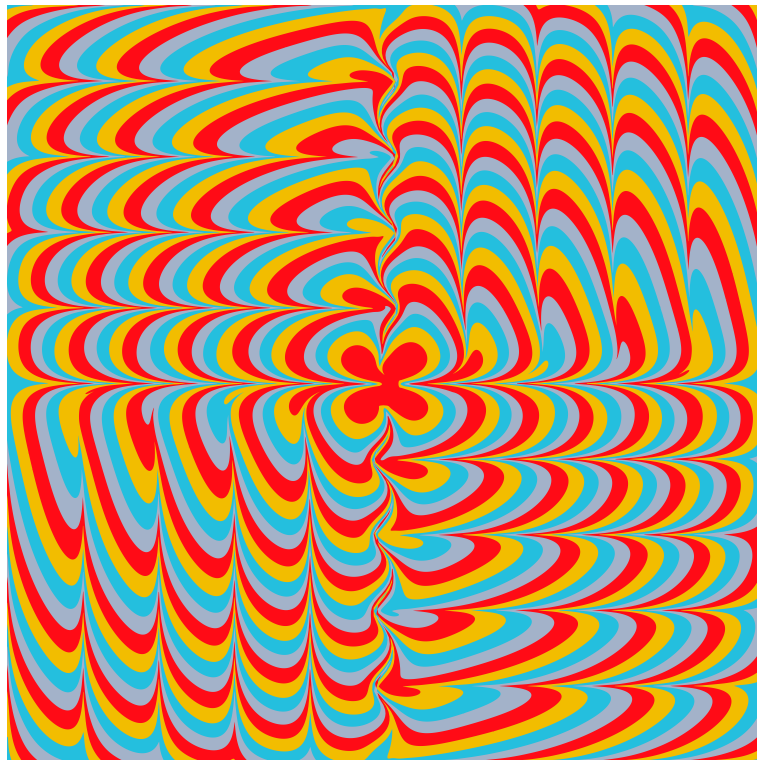
```


Example 5:

```

\newpsstyle{YellowGlass}{linecolor=gray,linewidth=0.1}
\newpsstyle{LensStyleHandle}{%
fillstyle=gradient,framearc=0.6,linewidth=0.5\pslinewidth,
gradmidpoint=0.5,gradangle=45,gradbegin=white,gradend=gray}
\begin{pspicture}(-5,-5)(5,5)
\psset{viscosity=500,background=1 1 1,
actions={
3
{
/rad 100 def
[ 0 0 rad background [0.27 0.01 0.02] /drop]
[ 0 0 rad background [0.78 0.02 0.1] /drop]
[ 0 0 rad background [0.77 0.92 0.47] /drop]
[ 0 0 rad background [0.11 0.18 0.07] /drop]
[ 0 0 rad background [0.96 0.85 0.1] /drop]
} repeat
[ 0 0 100 background [0.78 0.02 0.1] /drop]
[ 0 0 50 background [0.77 0.92 0.47] /drop]
[ 0 0 20 background [0.11 0.18 0.0] /drop]
0 72 359
{
/a exch 2 mul def
[ a sin 400 mul a cos 400 mul a sin 100 mul a cos 100 mul 10 50 /stroke]
} for
}}%
\psMarble(10,10)
\PstLens[LensMagnification=2,LensRotation=50,LensSize=2,LensShadow=false,LensStyleGlass=YellowGlass]
\psMarble(10,10)}
\end{pspicture}

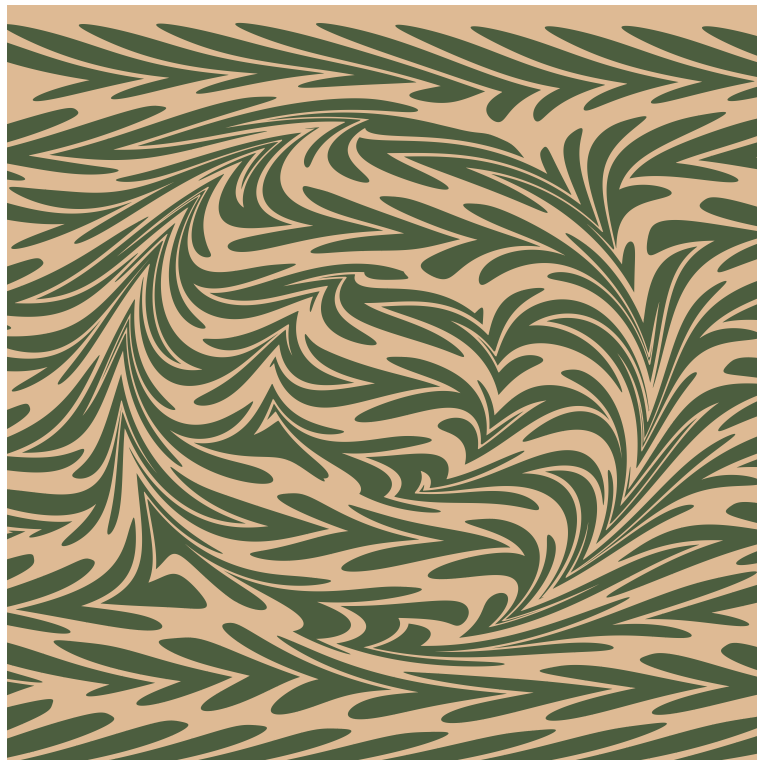
```

Example 6:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
  background=0.64 0.70 0.79,
  actions={%
11 -1 1
{
/rad exch sqrt 50 mul def
[ 0 0 rad background [0.64 0.70 0.79] /drop]
[ 0 0 rad background [0.14 0.75 0.87] /drop]
[ 0 0 rad background [0.95 0.74 0] /drop]
[ 0 0 rad background [1 0.043 0.086] /drop]
} for
-500 100 0 {/xpos exch def
[xpos -500 xpos 0 50 20 /stroke]
} for
0 100 500 {/xpos exch def
[xpos 500 xpos 0 50 20 /stroke]
} for
-500 100 0 {/ypos exch def
[500 ypos 0 ypos 50 20 /stroke]
} for
0 100 500 {/ypos exch def
[-500 ypos 0 ypos 50 20 /stroke]
} for
}}(10,10)
\end{pspicture}

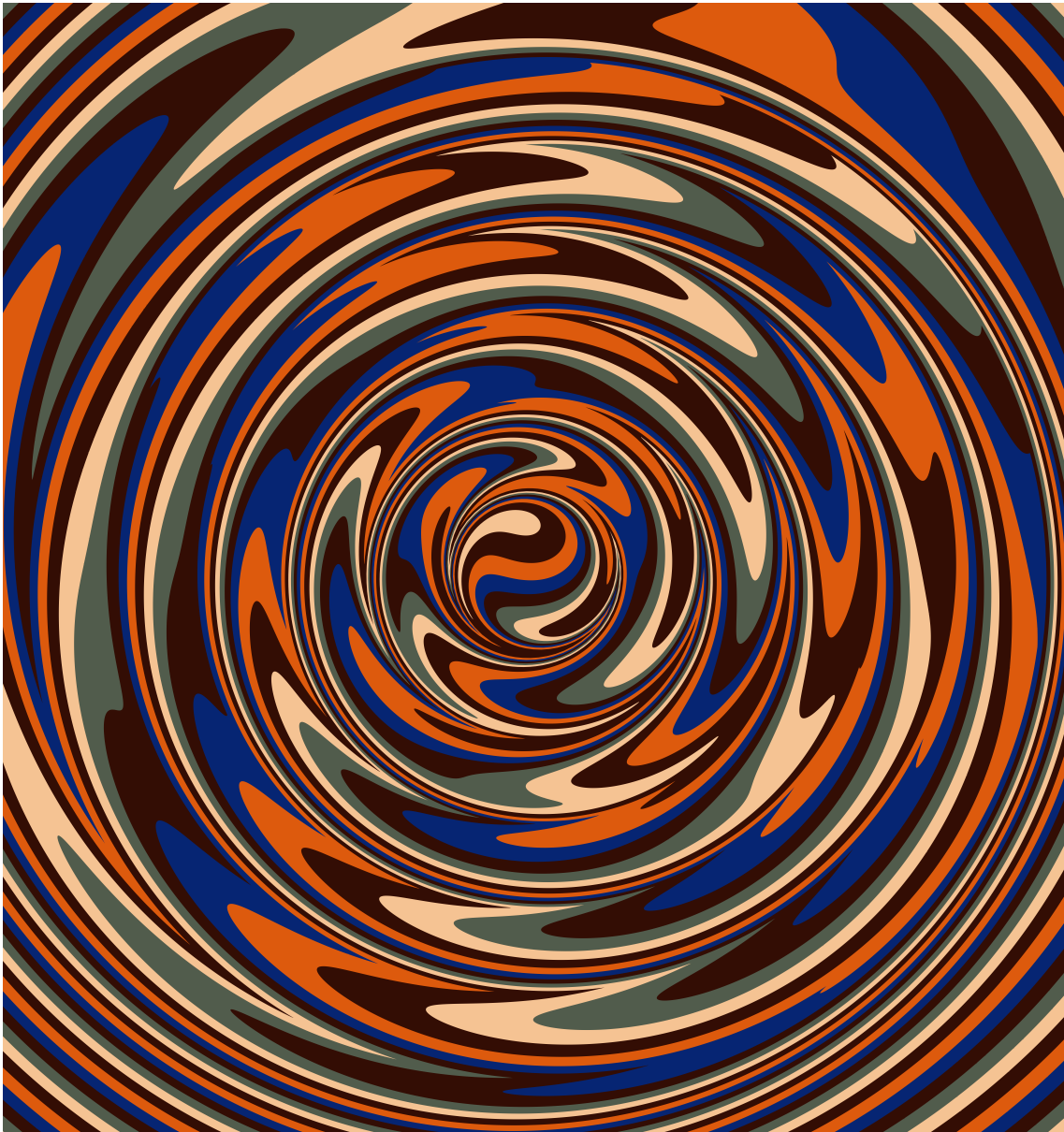
```

Example 7:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
          background=0.87 0.73 0.58,
          actions= {
            -500 84 500
            {
              /idy exch def
              -500 100 600
              {
                /idx exch def
                [ idx idy 45 background [0.298 0.370 0.247] /drop ]
              } for
            } for
            [ 90 [ -400 200 600 {} for ] 40 200 31 /rake]
            [-90 [ -500 200 600 {} for ] 40 200 31 /rake]
            [ 0 0 [ -350 ] 30 30 15 /stir ]
            [ 0 0 [ -150 ] 60 30 15 /stir ]
          }](10,10)
\end{pspicture}

```

Example 8:

```

\begin{pspicture}(-7.5,-8)(7.5,8)
\pstVerb{/ripple 75 def
/riplim ripple 6 mul def}%
\psMarble[
  actions={
    0 0 40 colors length concentric-bands
    [ 0 50 /offset]
    [ 0 0 [ ripple ripple riplim {} for ] 10 180 30 /stir ]
  }](15,16)
\end{pspicture}

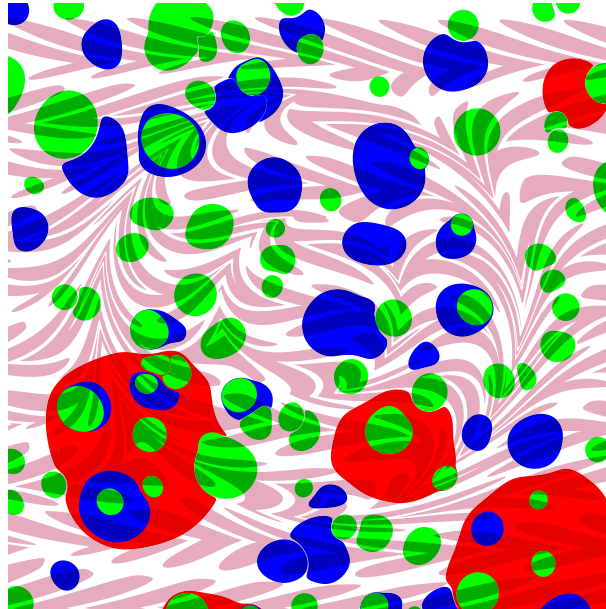
```

Example 9: Blendmodes

In case one want to overlap various marblings one can use the following blendmodes:

/Lighten, /Darken, /Normal, /Multiply, /Screen, /Overlay, /ColorDodge, /ColorBurn, /HardLight, /SoftLight, /Difference, /Exclusion, /Saturation, /Color, /Luminosity.

The blendmode is setup right after actions={ like: /Darken .setblendmode or some other of the blendmodes.

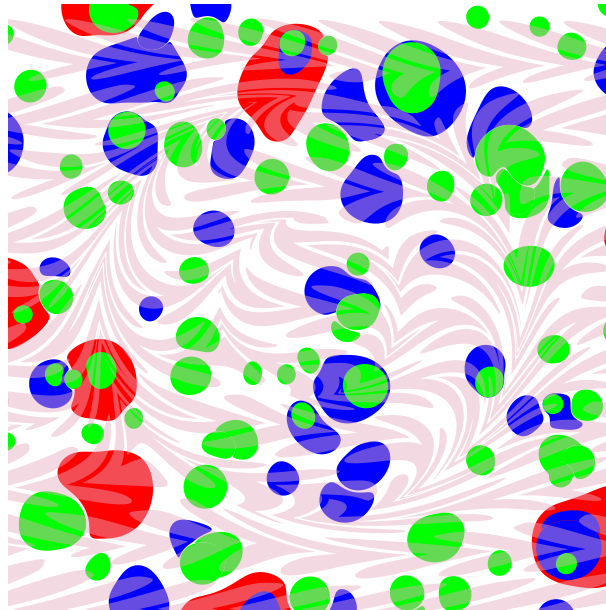


```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
  actions={
    50 [1 0 0 ] 10 random-drops
    20 [0 0 1 ] 50 random-drops
    15 [0 1 0 ] 100 random-drops
  }](10,10)
\psMarble[viscosity=1000,bckg=false,
  actions={
    /Darken .setblendmode
    -500 84 500
    {
/idy exch def
-500 100 600
{
    /idx exch def
    [ idx idy 45 background [0.898 0.670 0.747] /drop ]
} for
    } for
    [ 90 [ -400 200 600 {} for ] 40 200 31 /rake]
    [-90 [ -500 200 600 {} for ] 40 200 31 /rake]
    [ 0 0 [ -350 ] 30 30 15 /stir ]
    [ 0 0 [ -150 ] 60 30 15 /stir ]
    }](10,10)
\end{pspicture}
```

Example 10: Transparency

In case one want to overlap various marblings one can also use transparency:

The transparency is setup right after actions={ like: 0.45 .setopacityalpha or some other value between 0 and 1.



```
\begin{pspicture}(-4,-4)(4,4)
\psMarble[viscosity=1000,
  actions={
    50 [1 0 0 ] 10 random-drops
    20 [0 0 1 ] 50 random-drops
    15 [0 1 0 ] 100 random-drops
  }](8,8)
\psMarble[viscosity=1000,bckg=false,
  actions={
    0.45 .setopacityalpha
    -500 84 500
  }
/idy exch def
-500 100 600
{
  /idx exch def
  [ idx idy 45 background [0.898 0.670 0.747] /drop ]
} for
  } for
  [ 90 [ -400 200 600 {} for ] 40 200 31 /rake]
  [-90 [ -500 200 600 {} for ] 40 200 31 /rake]
  [ 0 0 [ -350 ] 30 30 15 /stir ]
  [ 0 0 [ -150 ] 60 30 15 /stir ]
  }](8,8)
\end{pspicture}
```

8 List of all optional arguments for *pst-marble*

Key	Type	Default
background	ordinary	[none]
viscosity	ordinary	[none]
actions	ordinary	[none]
colors	ordinary	[none]
drawcontours	boolean	true
bckg	boolean	true