

pst-marble v 1.4

A PSTricks package to draw marble-like patterns

Aubrey Jaffer
with the help of
Jürgen Gilg
Manuel Luque

February 26, 2019

Contents

1	History and Introduction	4
2	Techniques	6
2.1	Drop paint	6
2.2	Random drops	7
2.3	Concentric rings	8
3	The command <code>\psMarble</code>	9
4	Rendering	11
4.1	oversample	11
4.2	overscan	12
5	Colors	13
6	Basic actions	14
6.1	drop	14
6.2	line-drops	15
6.3	serpentine-drops	16
6.4	coil-drops	18
6.5	Gaussian-drops	19
6.6	uniform-drops	20
6.7	concentric-rings	21
6.8	rake	24
6.9	stylus	29
6.10	stir	31
6.11	vortex	35
6.12	wiggle	38
6.13	shift	43
7	Spray actions	44
7.1	Gaussian-spray	44
7.2	uniform-spray	45

8 Combined actions – Gallery	46
9 Acknowledgments	60

This program can 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.

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 end-papers 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}(-8,-6)(6,6)
\psMarble[background={1 1 1}],
  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 24 colors 36 concentric-rings
    180 [ 20 50 25 tines ] 40 200 31 rake
    0 350 shift
    0 270 0 -120 wiggle
    180 [ 3 600 -150 tines ] 40 200 31 rake
    0 270 0 240 wiggle
    180 [ 3 600 150 tines ] 40 200 31 rake
    0 270 0 -120 wiggle
  }
](12,12)
\psframe(-8,-6)(6,6)
\rput{90}(-7,0){\parbox{10cm}{\centering\bf\Large Marbling effects by Aubrey Jaffer\ and PSTricks}}
\end{pspicture}

```

1 History and Introduction

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:

<https://arxiv.org/abs/1810.04646>

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) having 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 y -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 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.

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 13 types of actions defined and ready to use:

```
drop
line-drops
serpentine-drops
coil-drops
Gaussian-drops
uniform-drops
concentric-rings
rake
stylus
stir
vortex
wiggle
shift
```

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

```
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-rings”.

Aubrey Jaffer coded all the possible simulations with the expected deformations (rake, stylus, 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.

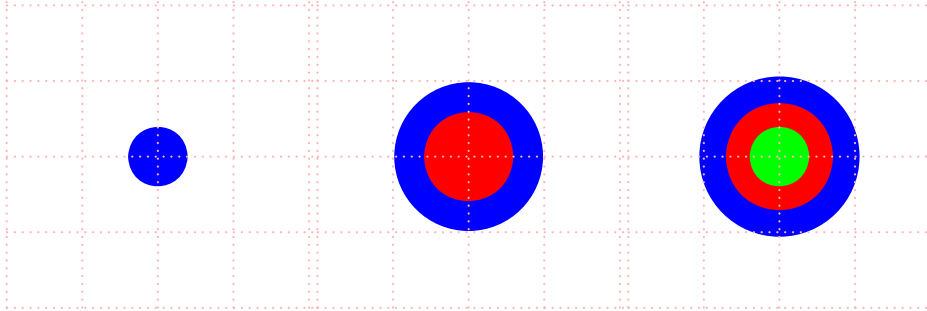
When a new drop is placed, the points in that drop retain their color, the outer points are calculated before being assigned their initial color. This possibility is operational by taking a negative value for the viscosity.

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 or more drops of different colors, this gives:



The command to drop a drop is written as follows:

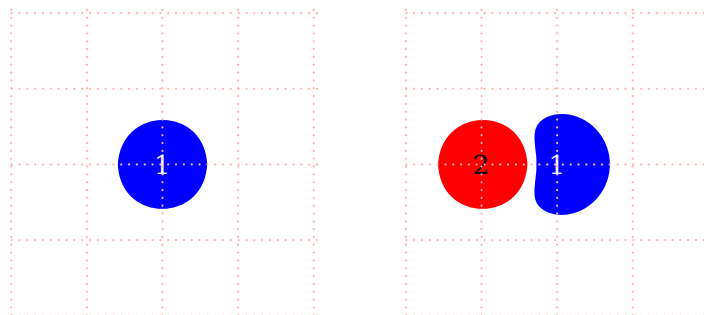
```
0 0 100 [0 0 1] drop
```

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:

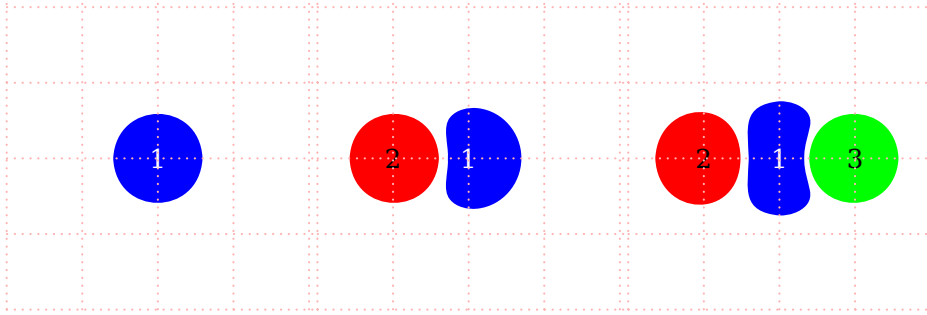
$$\vec{P}' = \vec{C} + (\vec{P} - \vec{C}) \sqrt{1 + \frac{r^2}{\|\vec{P} - \vec{C}\|^2}}$$

$P(x, y)$ 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

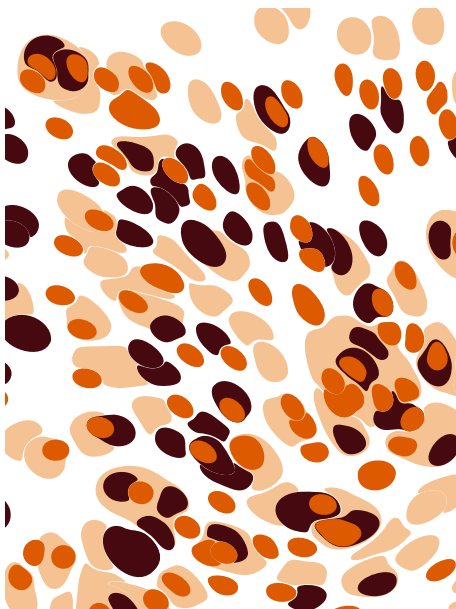
One of the techniques is to project with a brush drops of color on the surface of the liquid in several stages by changing color. The position of the drops is therefore random. Each drop influences its neighbors and assuming that initially the drops would form a round spot on the surface, they will deform 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={
    0 0 1000 1000 0 [0.960 0.764 0.576] 125 30 uniform-drops
    0 0 1000 1000 0 [0.270 0.035 0.058] 100 25 uniform-drops
    0 0 1000 1000 0 [0.866 0.353 0.000] 150 20 uniform-drops
    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] (width,height)
```

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

If none of the optional arguments $(width,height)$ or $(x-,y-)(x+,y+)$ are taken, the default value $(10,10)$ respectively $(-5,-5)(5,5)$ is used. If the `\begin{pspicture}` arguments do not match the optional arguments $(width,height)$ or $(x-,y-)(x+,y+)$ the image will be cropped or padded.

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

Name	Default	Meaning
<code>actions</code>	<code>0 0 35 colors 35 concentric-rings</code>	The type of marbling action
<code>spractions</code>	<code>{}</code>	Specifies the sequence of spray commands to perform. Spray commands are performed after marbling.
<code>background</code>	<code>[1 1 1]</code>	Background color to be used with <code>rgb</code> or <code>RGB</code> or hexadecimal notation
<code>seed</code>	Mathematical Marbling	Random seed to obtain the same arrangement of random drops within <code>Gaussian-drops</code> and <code>uniform-drops</code>
<code>oversample</code>	<code>0</code>	This is a rendering option: <code>oversample=0</code> makes the image pixel free; <code>oversample>0</code> : the smaller the positive value, the larger the pixels.
<code>overscan</code>	<code>1</code>	When the <code>overscan</code> value is greater than 1, proportionally more image (outside of the specified area) is shown, and the specified area is outlined with a dashed rectangular border.
<code>bckg</code>	<code>true</code>	Boolean: to turn on/off the background color
<code>colors</code>	<code>[0.275 0.569 0.796]</code> <code>[0.965 0.882 0.302]</code> <code>[0.176 0.353 0.129]</code> <code>[0.635 0.008 0.094]</code> <code>[0.078 0.165 0.518]</code> <code>[0.824 0.592 0.031]</code> <code>[0.059 0.522 0.392]</code> <code>[0.816 0.333 0.475]</code> <code>[0.365 0.153 0.435]</code> <code>[0.624 0.588 0.439]</code>	Colors of the marbling can be set within the <code>rgb-color</code> system or as hexadecimal color constants. Shown are <code>rgb</code> constants between 0 and 1.
<code>drawcontours</code>	<code>false</code>	Boolean: if set to <code>true</code> , it only draws the contours
<code>viscosity</code>	<code>1000</code>	Global primitive: viscosity of the system

Notes:

- If `oversample>0`, the image will be 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.
- Colors can be setup within the `rgb-color-system`: `colors={ [0.1 0.4 0.9] [1 0 1] ... }` or `colors={ [255 0 0] [123 245 129] ... }`. As well can be entered hexadecimal color constants which are set up within parentheses like: `colors={ (e7cc9b) (c28847) (80410b) ... }` or with capital letters like: `colors={ (E7CC9B) (C28847) (80410B) ... }`
- For the background color curly braces are needed: `background={ [0.2 0.5 0.7] }` or `background={ [2 255 2] }`.
- Following are introduced some basic actions, like `drop`, `line-drops`, `serpentine-drops`, `coil-drops`, `Gaussian-drops`, `uniform-drops`, `concentric-rings`, `rake`, `stylus`, `stir`, `vortex`, `wiggle` and `shift`.

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

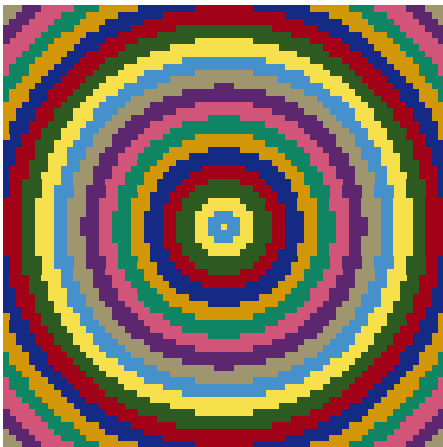
As designs get more complicated, hundreds of drops and styluses, reverse-rendering is the only practical way to render them. As the number of strokes increases, the number of points in the contours needs to increase as well. As the number of drops increases, the time to compute each pixel becomes less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

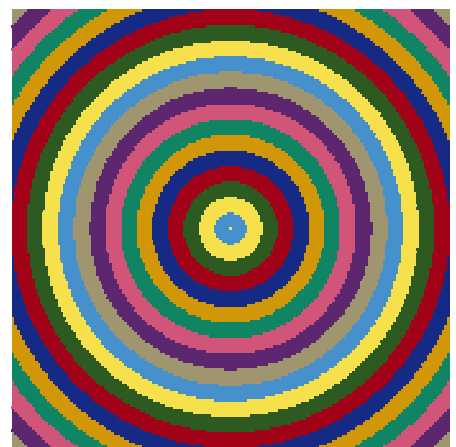
4.1 oversample

- `oversample=0` is contour rendering (pixel-free).
- `oversample>0` is raster rendering.
- `oversample=0.5` is raster rendering at half resolution. It renders blocky images relatively quickly.
- `oversample=1` is raster rendering; the same as negative `viscosity=` from v1.2.
- `oversample>1` will take longer to render, the image produced by ghostscript will be no better than `oversample=1`.

Note: The smaller the `oversample=` value, the more blocky the image gets. Typical values might be: 0, 0.5, 1.



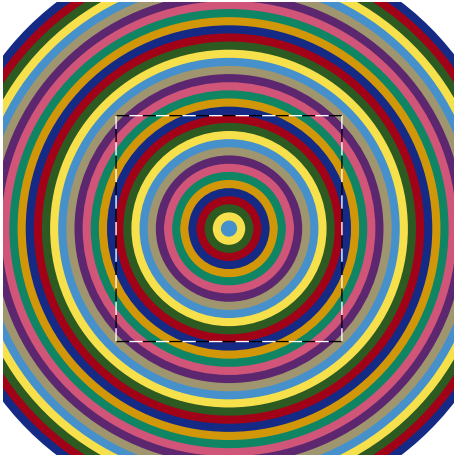
```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[oversample=0.4](6,6)
\end{pspicture}
```



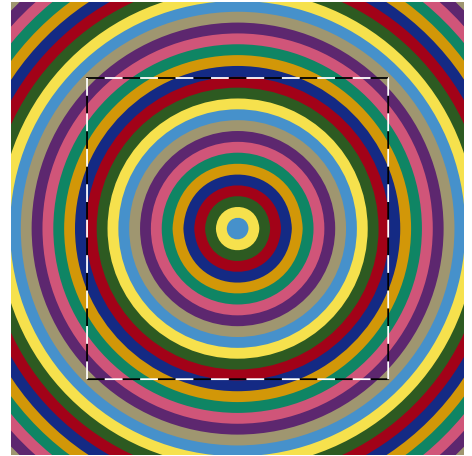
```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[oversample=1](6,6)
\end{pspicture}
```

4.2 overscan

When the overscan value is greater than 1, proportionally more image (outside of the specified area) is shown, and the specified area is outlined with a dashed rectangular border. This is a utility for developing marblings, new for version 1.4.



```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[overscan=2](6,6)
\end{pspicture}
```



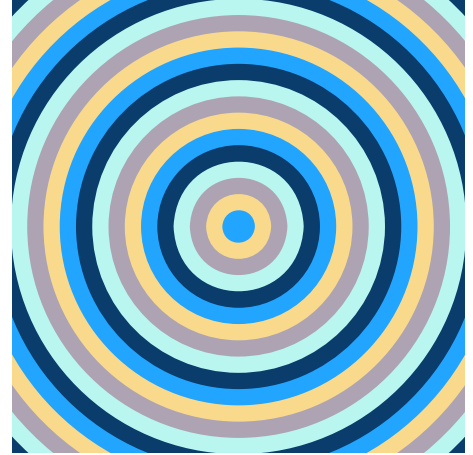
```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[overscan=1.5](6,6)
\end{pspicture}
```

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-rings` as follows:

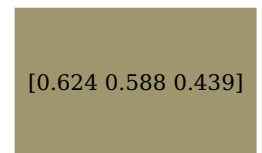
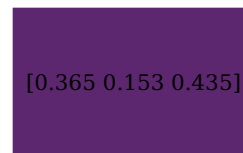
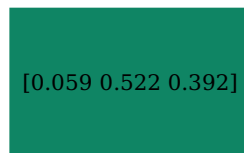
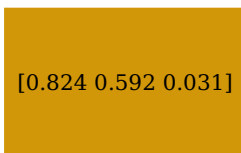
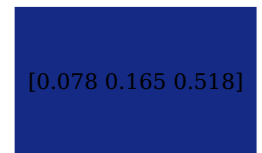
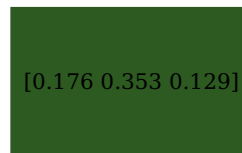
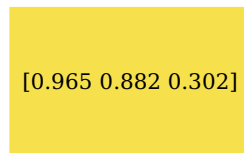
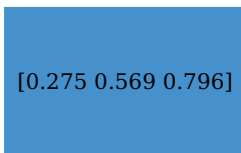


```
\begin{pspicture}(-3,-3)(3,3)
\psMarble(6,6)
\end{pspicture}
```



```
\begin{pspicture}(-3,-3)(3,3)
\psMarble[colors={
[0.134 0.647 1.000]
[0.977 0.855 0.549]
[0.684 0.638 0.702]
[0.730 0.965 0.942]
[0.040 0.236 0.424]
}](6,6)
\end{pspicture}
```

Hint: 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:



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

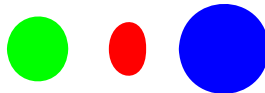
`drop` defines a single drop set on the surface of a liquid.

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

```
cx, cy  Center coordinates
r       Radius of the 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`.

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 10 default colors. Colors can be used like this:

```
0 0 50 colors 1 get drop
-200 0 70 colors 2 get drop
200 0 100 colors 3 get drop
```

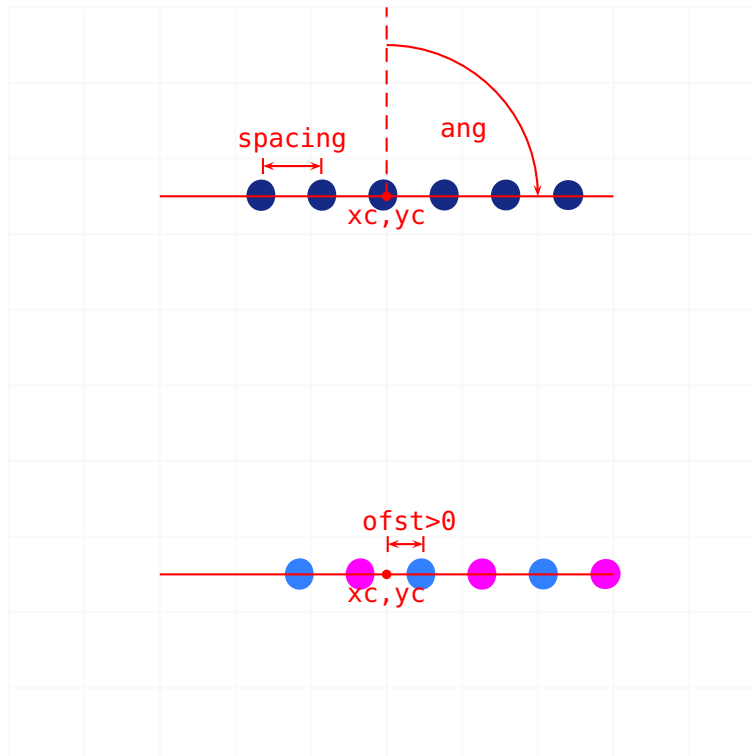
6.2 line-drops

`xc yc ang [r] [rgb] drad line-drops`

Drops color [rgb] or color series of radius drad in a line centered at `xc`, `yc` and `ang` degrees from vertical (clockwise). One drop is placed at each `r` distance from `xc`, `yc`. For [r] we can use

`[cnt spacing ofst tines]`

Returns `cnt` numbers spacing apart with middle element equal to `ofst`. Used for `rake` and `line-drops` command.



```
\begin{pspicture*}(-5,-5)(5,5)
\psgrid[subgriddiv=1,gridcolor=lightgray!10]
\psMarble[viscosity=1000,bckg=false,
actions={
0 250 90 [ 6 80 0 tines ] colors 4 get 20 line-drops
0 -250 90 [ 6 80 50 tines ] [[0.2 0.5 1][1 0 1]] 20 line-drops
}](10,10)
\rput(0,2.5){
\psdot[linecolor=red](0,0)
\uput[-90](0,0){\textcolor{red}{\texttt{xc,yc}}}
\psline[linecolor=red,linestyle=dashed](0,0)(0,2.5)
\psline[linecolor=red](-3,0)(3,0)
\psarcn[linecolor=red]{->}(0,0){2}{90}{0}
\uput[1cm][45](0,0){\textcolor{red}{\texttt{ang}}}
\psline[linecolor=red]{<->}(-1.65,0.4)(-0.85,0.4)
\uput[90](-1.25,0.4){\textcolor{red}{\texttt{spacing}}}
}
\rput(0,-2.5){
\psdot[linecolor=red](0,0)
\uput[-90](0,0){\textcolor{red}{\texttt{xc,yc}}}
\psline[linecolor=red](-3,0)(3,0)
\psline[linecolor=red]{|<->}|(0,0.4)(0.5,0.4)
\uput[90](0.3,0.4){\textcolor{red}{\texttt{ofst>0}}}
}
\end{pspicture*}
```

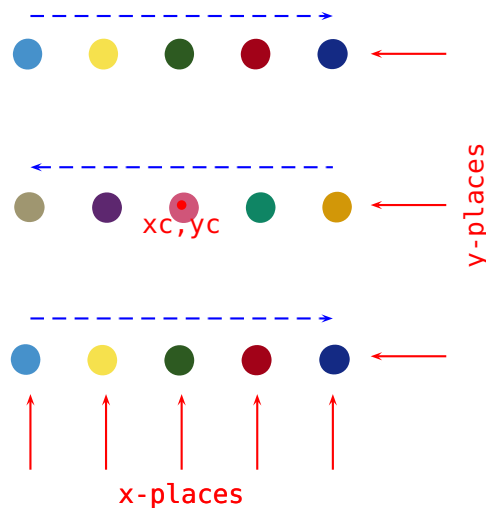
6.3 serpentine-drops

serpentine-drops deposits a series of drops on a user-specified “grid” in a serpentine sequence.

```
xc yc [ x-places ] [ y-places ] ang rgb drad serpentine-drops
```

xc, yc Coordinates of the center
 [x-places] x-coordinates for the x times y number of drops
 [y-places] y-coordinates for the x times y number of drops
 ang Rotation angle from vertical (clockwise)
 rgb Color of the drops or color series (array)
 drad Radius of the drops

Places drops of colors [rgb] of radius drad in a serpentine pattern (starting lower left to right; right to left; left to right ...) at coordinates [x-places] x [y-places] relative to location xc, yc and rotated by ang degrees clockwise from vertical. The sequences [x-places] and [y-places] determine the order in which drops are placed. The resulting grid will not be square because each drop is moved by subsequent drops.

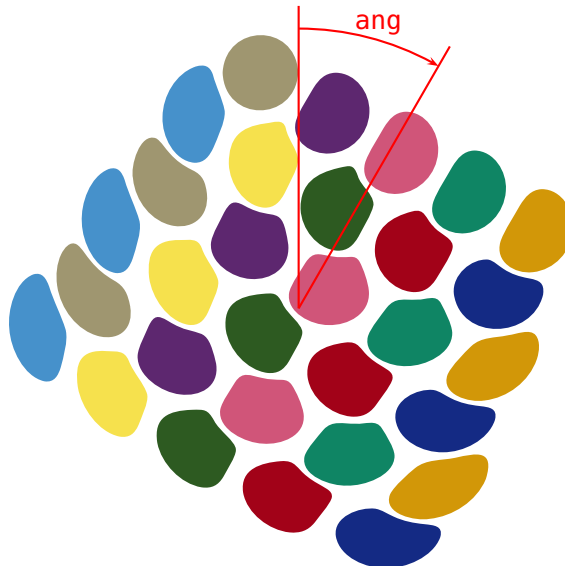


```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
actions={
0 0 [-200 -100 0 100 200][-200 0 200 ] 0 colors 20 serpentine-drops
}
](10,10)
\multido{\iA=-2+1}{5}{
\psline[linecolor=red]{->}(\iA,-3.5)(\iA,-2.5)
}
\uput[-90](0,-3.5){\color{red}\texttt{x-places}}
\multido{\iA=-2+2}{3}{
\psline[linecolor=red]{->}(3.5,\iA)(2.5,\iA)
}
\rput[90](3.9,0){\color{red}\texttt{y-places}}
\uput[-90](0,-3.5){\color{red}\texttt{x-places}}
\psdot[linecolor=red](0,0)
\uput[-90](0,0){\color{red}\texttt{xc,yc}}
\psline[linecolor=blue,linestyle=dashed]{->}(-2,-1.5)(2,-1.5)
\psline[linecolor=blue,linestyle=dashed]{<-}(-2,0.5)(2,0.5)
\psline[linecolor=blue,linestyle=dashed]{->}(-2,2.5)(2,2.5)
\end{pspicture}
```



```
[ cnt spacing ofst tines ]
```

Returns cnt numbers spacing apart with middle element equal to ofst.
Used as well for the rake and line-drops command.



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
actions={
0 0 [5 100 0 tines][6 75 20 tines] 30 colors 50 serpentine-drops
}
](10,10)
\psline[linecolor=red](0,0)(0,4)
\psline[linecolor=red](0,0)(4;60)
\psarcn[linecolor=red]{->}(0,0){3.7}{90}{60}
\uput{3.7}[75](0,0){\color{red}\texttt{ang}}
\end{pspicture}
```

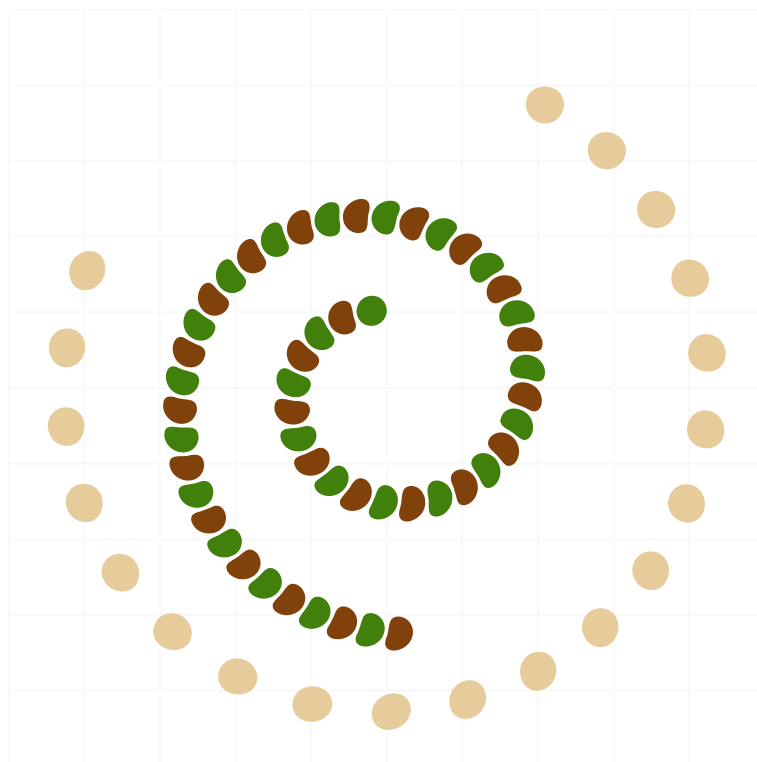
6.4 coil-drops

coil-drops defines a series of drops along a circle or spiral.

```
xc yc r ang-strt arcinc rinc [rgb] cnt drad coil-drops
```

xc, yc	Coordinates of the center
r	Radius of the circle where the drops will lay on
ang-strt	Start angle from vertical (clockwise)
arcinc	Arc-length between the drops
rinc	Increment of r: if taken 0 it gives a circle, if taken >0 it spirals outwards, if taken <0 it spirals inwards.
rgb	Color of the drops or color series (array)
cnt	Number of drops
drad	Radius of the drops

Drops cnt paint drops with radius drad in arc around xc,yc at radius r starting at ang-strt and spaced by arcinc distance. r is incremented (or decremented if rinc is negative) by rinc after each drop.



```
\begin{pspicture*}(-5,-5)(5,5)
\psgrid[subgriddiv=1,gridcolor=lightgray!10]
\psMarble[bckg=false,viscosity=1000,
actions={
0 0 400 30 95 0 [ 231 204 155 ] 20 25 coil-drops
0 0 300 180 35 -4 [[ 128 65 11 ] [ 65 128 11 ]] 50 20 coil-drops
}](10,10)
\end{pspicture*}
```

6.5 Gaussian-drops

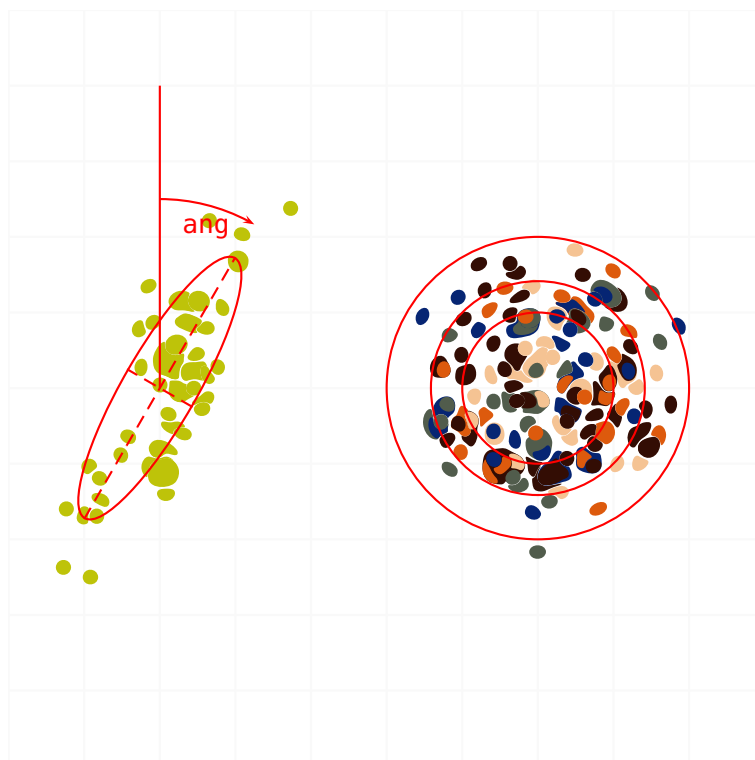
Gaussian-drops defines a randomly calculated series of drops mostly within a circle/ellipse.

`xc yc r ang eccentricity [rgb] cnt drad Gaussian-drops`

Drops `cnt` paint drops with radius `drad` in normal (Gaussian) distribution centered at `xc`, `yc` with radius `r`, `ang` degrees from vertical (clockwise) and length to width ratio `eccentricity` (1 is circular).

`[rgb]` can be one color or a color series.

63% of drops are centered within radius `r`, 87% of drops are centered within radius $r \cdot \sqrt{2}$, 98% of drops are centered within radius $r \cdot 2$. The `eccentricity` stretches and shrinks the target from a circle into an ellipse. If `eccentricity` > 1, it stretches the circle in `y`-direction and shrinks it in `x`-direction. If $0 < \text{eccentricity} < 1$, it stretches the circle in `x`-direction and shrinks it in `y`-direction. `eccentricity` is the ratio of the major axis to the minor axis. And the `ang` should be the major axis counter-clockwise from vertical.



```
\begin{pspicture*(-5,-5)(5,5)
\psgrid[subgriddiv=1,gridcolor=lightgray!10]
\psMarble[bckg=false,viscosity=1000,
colors={
[0.960 0.764 0.576][0.316 0.362 0.298]
[0.200 0.050 0.015][0.023 0.145 0.451]
[0.866 0.353 0.050][0.200 0.050 0.015]
},
actions={
200 0 100 0 1 colors 150 10 Gaussian-drops
-300 0 100 30 4 [190 195 9] 55 10 Gaussian-drops
}](10,10)
\pscircle[linecolor=red](2,0){!1}\pscircle[linecolor=red](2,0){!1 2 sqrt mul}
\pscircle[linecolor=red](2,0){!1 2 mul}
\rput{60}(-3,0){
\psellipse[linecolor=red](0,0){!1 2 mul 1 2 div}
\psline[linestyle=dashed,linecolor=red]{!1 2 mul neg 0}{!1 2 mul 0}
\psline[linestyle=dashed,linecolor=red]{!0 1 2 div neg}{!0 1 2 div}
}
\psline[linecolor=red](-3,0)(-3,4)
\psarcn[linecolor=red]{->}(-3,0){2.5}{90}{60}
\uput{2cm}[75](-3,0){\textcolor{red}{\texttt{ang}}}
\end{pspicture}
```

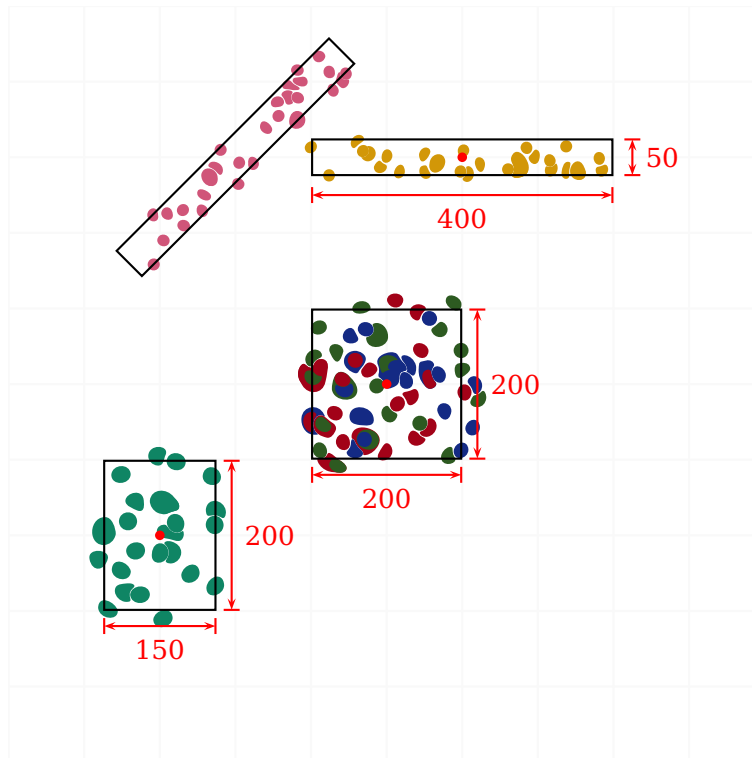
6.6 uniform-drops

uniform-drops defines a randomly calculated series of drops within a rectangular box.

```
xc yc xsid ysid angle [ rgb ] cnt drad uniform-drops
```

Drops cnt paint drops with radius drad in a uniform distribution in a xsid by ysid box centered at xc, yc and rotated by angle from vertical (clockwise).

[rgb] can be one color or a color series.



```
\begin{pspicture*}(-5,-5)(5,5)
\psgrid[subgriddiv=1,gridcolor=lightgray!10]
\psMarble[viscosity=1000,bckg=false,
  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 0 200 200 0 [[0.176 0.353 0.129][0.635 0.008 0.094][0.078 0.165 0.518]] 65 10 uniform-drops
-300 -200 150 200 0 colors 4 get 25 12 uniform-drops
100 300 400 50 0 colors 3 get 30 8 uniform-drops
-200 300 50 400 45 colors 5 get 30 8 uniform-drops
}](10,10)
\rput(0,0){\psframe(-1,-1)(1,1)}
\psline[linecolor=red]{|<->|}(-1,-1.2)(1,-1.2)\uput[-90](0,-1.2){\textcolor{red}{200}}
\psline[linecolor=red]{|<->|}(1.2,-1)(1.2,1)\uput[0](1.2,0){\textcolor{red}{200}}
\psdot[linecolor=red](0,0)
\rput(-3,-2){\psframe(-0.75,-1)(0.75,1)}
\psline[linecolor=red]{|<->|}(-3.75,-3.2)(-2.25,-3.2)\uput[-90](-3,-3.2){\textcolor{red}{150}}
\psline[linecolor=red]{|<->|}(-2.05,-3)(-2.05,-1)\uput[0](-2.05,-2){\textcolor{red}{200}}
\psdot[linecolor=red](-3,-2)
\rput(1,3){\psframe(-2,-0.25)(2,0.25)}
\psline[linecolor=red]{|<->|}(-1,2.5)(3,2.5)\uput[-90](1,2.5){\textcolor{red}{400}}
\psline[linecolor=red]{|<->|}(3.25,2.75)(3.25,3.25)\uput[0](3.25,3){\textcolor{red}{50}}
\psdot[linecolor=red](1,3)
\rput{45}(-2,3){\psframe(-2,-0.25)(2,0.25)}
\end{pspicture*}
```

6.7 concentric-rings

With `concentric-rings`, we set the number of different colored concentric rings (`count`) (at center `cx, cy`) with a thickness of `thick`.

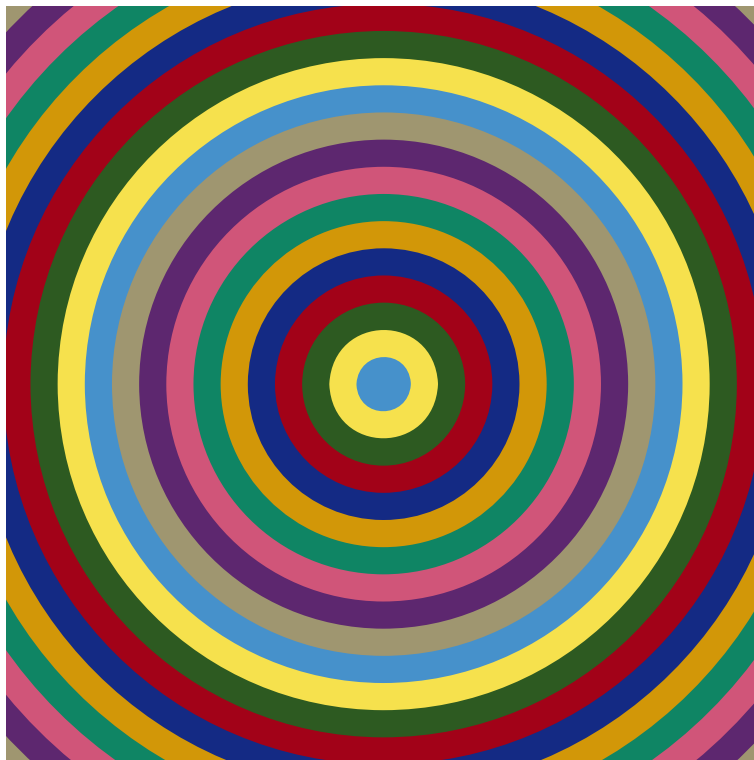
To code it within the L^AT_EX file we use the following syntax:

```
xc yc thick [ color ] count concentric-rings
```

<code>cx, cy</code>	Center coordinates
<code>thick</code>	Thickness of the rings
<code>count</code>	Number of rings
<code>color</code>	Array of colors: <code>[[rgb][rgb]...[rgb]]</code>

Example 1:

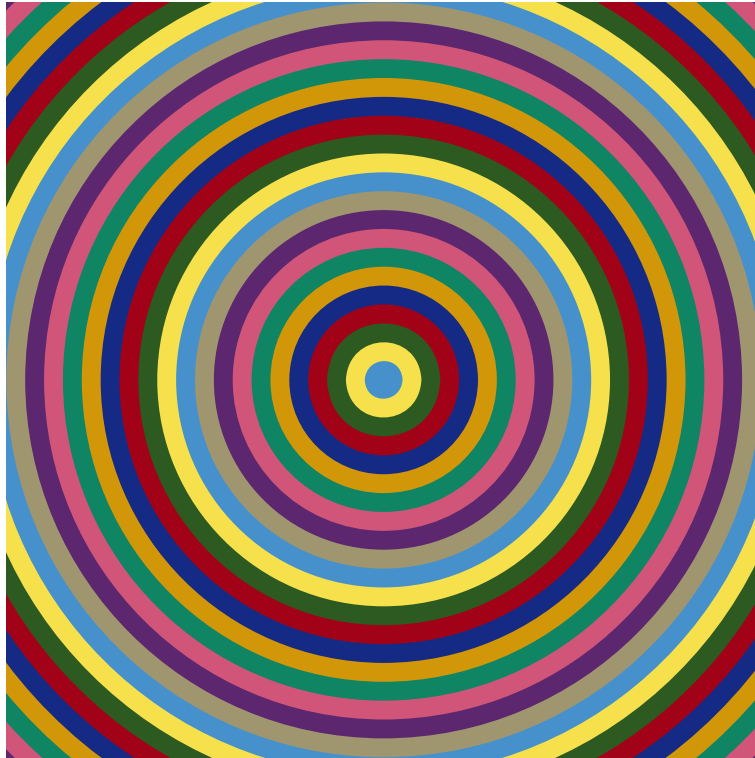
`concentric-rings` is the default action of the `actions={...}` key, meaning if **no** action is chosen, `concentric-rings` with its default thickness and its default color list is in effect.



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

Example 2:

If we want to change thick and count, we do the following:



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  actions={
% cx cy thick color count
  0 0 25 colors 40 concentric-rings
  }](10,10)
\end{pspicture}
```

Note: If one increases thick to large values > 100 the area of the bands will increase by square, so not all bands 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: $35 < \text{thick} < 50$

6.8 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.

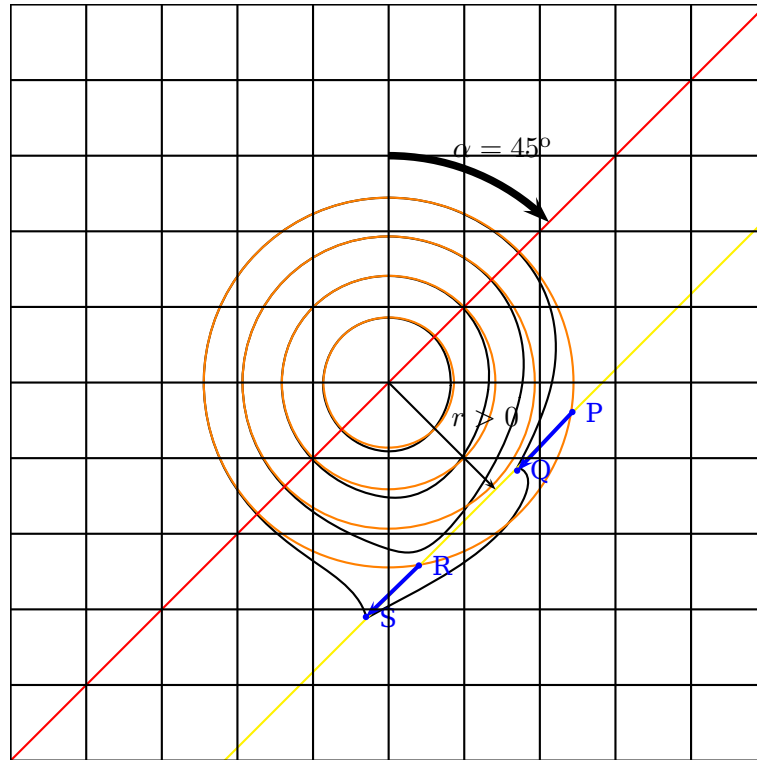
For the following examples viscosity=1000 is set. This is a typical value (default value).

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 $|\overrightarrow{PQ}|$:
 $tU = |\overrightarrow{PQ}| = 1$ 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 $|\overrightarrow{RS}|$:
 $tU = |\overrightarrow{RS}| = 1$ 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.

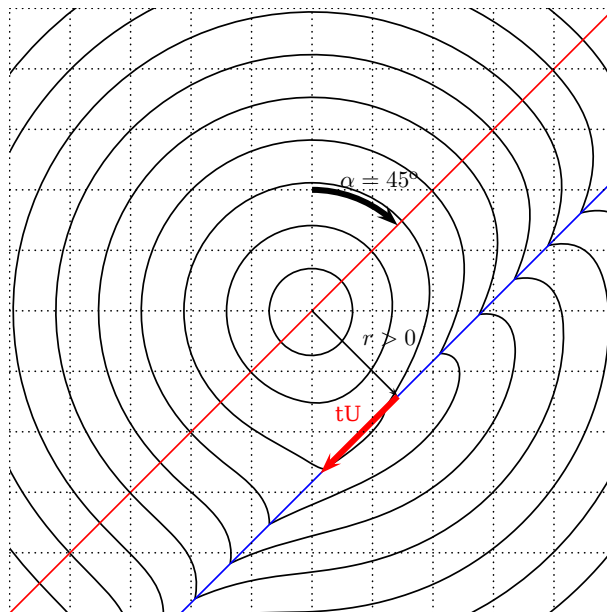
The distance [$r=200$] (in mm) of one tooth from the center of the rake on the right side referred to the stylus track direction, if r is taken positive; to the left side to the stylus track direction, if r is taken negative.

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,bckg=false,drawcontours,
colors={[0 0 0]},
actions={
0 0 50 2 sqrt mul colors 9 concentric-rings
% 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){$\alpha=45^{\mathrm{o}}$}
\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)(1.8;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.

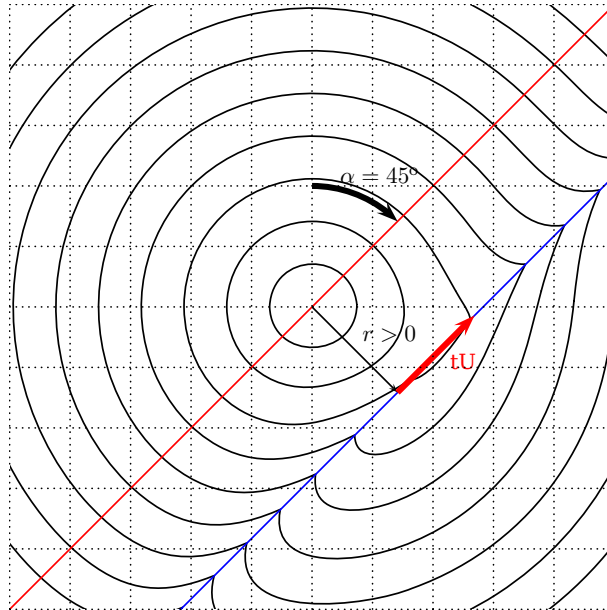
The distance [$r=200$] of one tooth from the center of the rake on the right side referred to the stylus track direction, if r is taken positive; to the left side to the stylus track direction, if r is taken negative.

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,bckg=false,drawcontours,
colors={[0 0 0]},
actions={
0 0 50 2 sqrt mul colors 9 concentric-rings
% 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)(1.8;45)}
\rput(2.5,-0.9){\red tU}
\psgrid[subgriddiv=1,griddots=10,gridlabels=0pt]
\end{pspicture*}

```

Example 3: 0 [11 100 0 tines] 50 100 30 rake

[cnt spacing ofst tines]

cnt Number of teeth
spacing Displacement between the teeth
ofst Offset of the middle tooth
 to the left (negative),
 to the right (positive)

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

The distance [r] is a list of 11 teeth: [11 100 0 tines] meaning the distances of the teeth are: -500, -400, -300, -200, -100, 0, 100, 200, 300, 400, 500. Starting at 0 to the left and right in steps of 100 up to 11 teeth with offset 0 (the center). (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. If the offset is set to a negative value, the center tooth moves to the left if taken positive it moves the center tooth to the right.

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[
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 35 colors 32 concentric-rings
% rake with 11 teeth aligned 1 cm (= 100 pts) with
0 [11 100 0 tines] 50 100 30 rake
}](8,10)%
\psMarble[
linewidth=0.05,
colors={[1 1 1]},
bckg=false,
drawcontours,
actions={
0 0 35 colors 32 concentric-rings
%% rake with 11 teeth aligned 1 cm (= 100 pts)
0 [11 100 0 tines] 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`.

6.9 stylus

Parameters for stylus: bx, by, ex, ey, V, D

bx by ex ey V D stylus

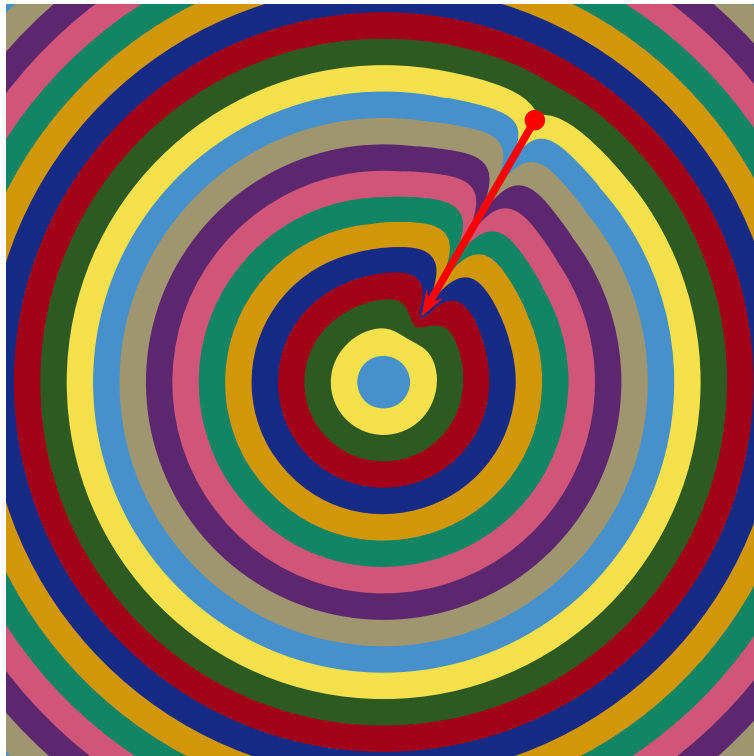
bx, by Beginning of stroke

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 colors 32 concentric-rings
% 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 stylus
}](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={
  0 0 35 colors 32 concentric-rings
% 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 stylus
}](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}

```

6.10 stir

Consider a single stylus (a cylinder of diameter D) that is placed on the y -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 clockwise².

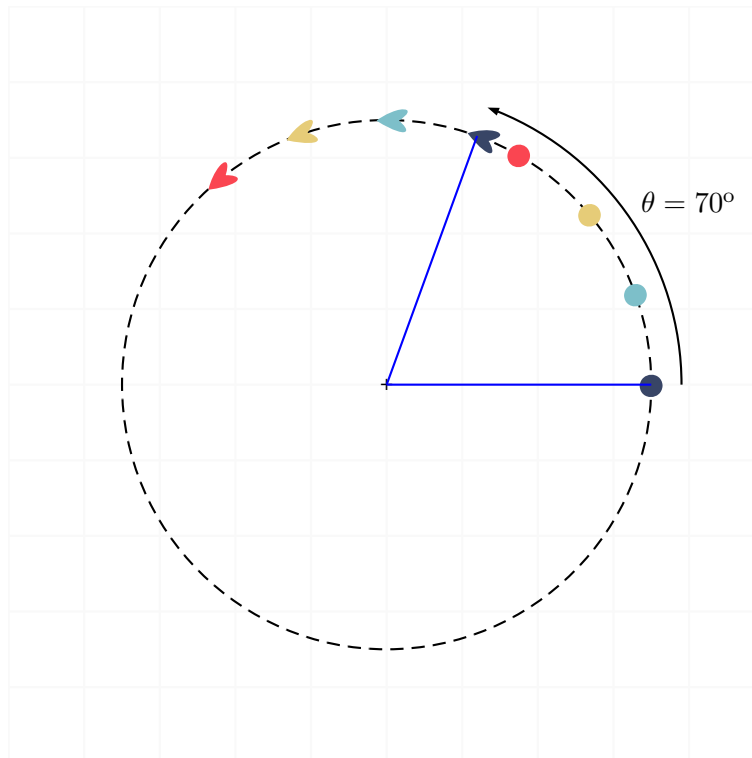
```
cx cy [ r ] w th D stir
```

cx, cy	Center coordinates in mm.
[r]	List of radii in mm.
w	Angular velocity in degrees/s.
th	Angle displacement at tines in degrees (clockwise).
D	Tine diameter in mm.

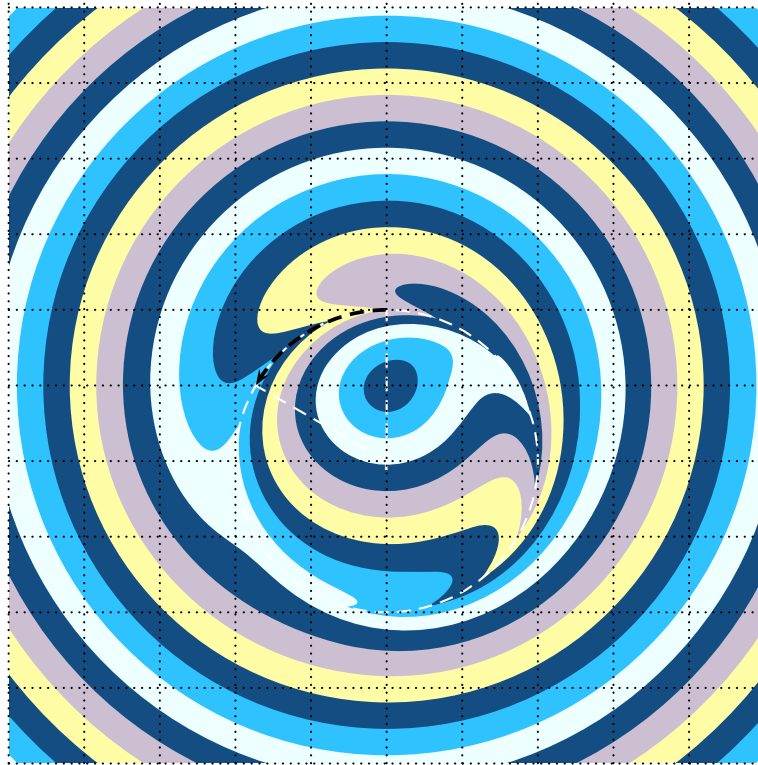
Explanations for the key th:

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

All points on the circle are rotated by $\text{th}=70$. 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:

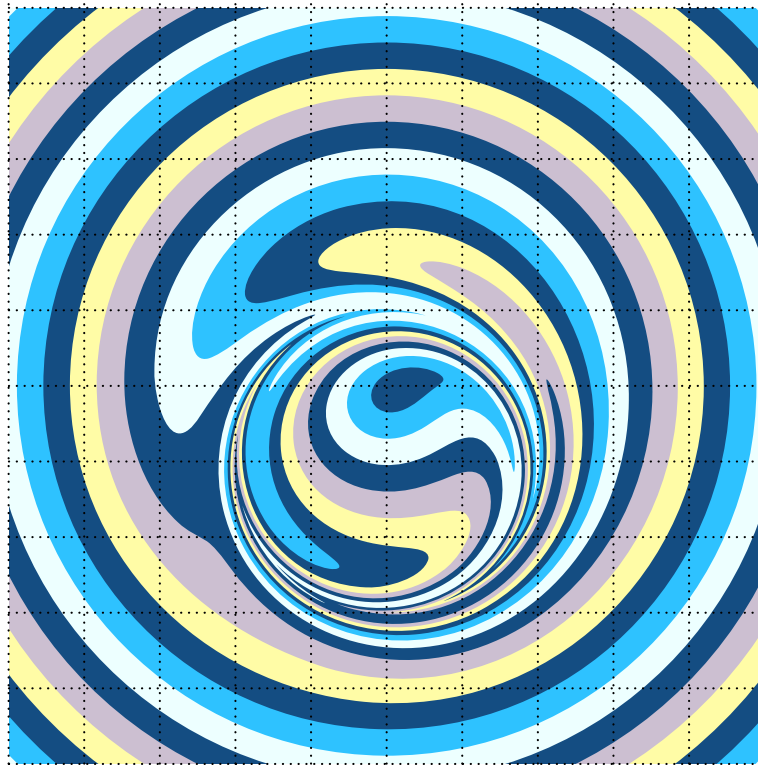
```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  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={
    0 0 35 colors 32 concentric-rings
    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:



```

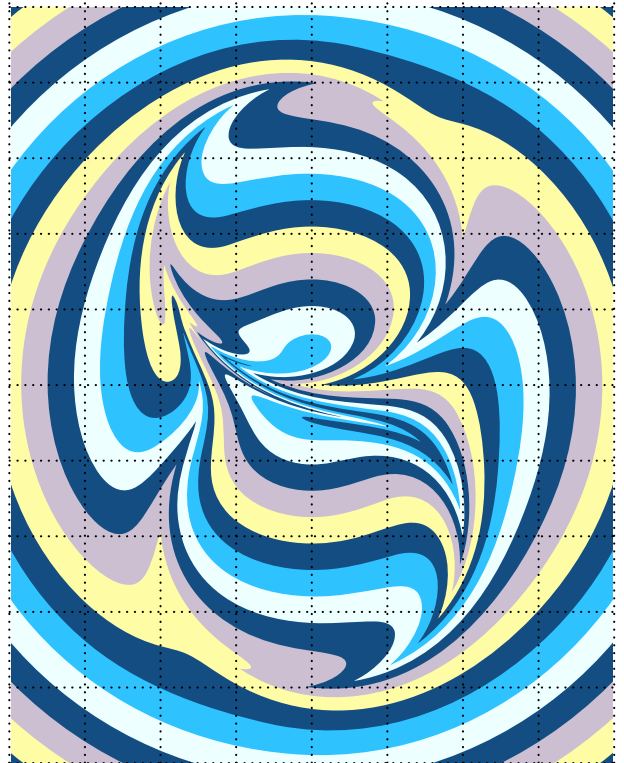
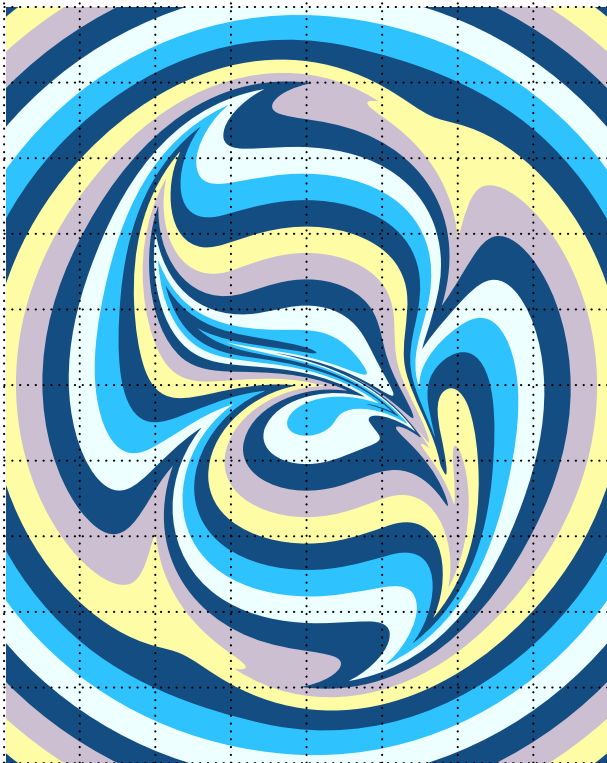
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  colors={
    [0.08 0.3 0.51]
    [0.18 0.76 1]
    [0.93 1 1]
    [0.08 0.3 0.51]
    [0.8 0.75 0.82]
    [1 0.99 0.65]
  },
  actions={
    0 0 35 colors 32 concentric-rings
    0 -100 [200] 10 60 30 stir
    0 -100 [200] 10 60 30 stir % repeated action
    0 -100 [200] 10 60 30 stir % repeated action
    0 -100 [200] 10 60 30 stir % repeated action
  }](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[
  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={
    0 0 35 colors 32 concentric-rings
    0 200 [200] 10 60 30 stir
    0 -200 [200] -10 60 30 stir
  }](8,10)
\psgrid[subgriddiv=1,griddots=10,gridlabels=0pt]
\end{pspicture}
```

```
\begin{pspicture}(-4,-5)(4,5)
\psMarble[
  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={
    0 0 35 colors 32 concentric-rings
    0 -200 [200] -10 60 30 stir
    0 200 [200] 10 60 30 stir
  }](8,10)
\psgrid[subgriddiv=1,griddots=10,gridlabels=0pt]
\end{pspicture}
```

6.11 vortex

cx cy circ t vortex

cx, cy Center coordinates in mm.

circ Circulation (in mm^2/s) is a simple scale factor.

Typical value: $30\text{e}3 \text{ mm}^2/\text{s}$.

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:

<https://arxiv.org/abs/1810.04646>

The documentation illustrates the used parameters:

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

After a long enough time, the whole surface returns to its initial state. This can be easily proofed within an animation.

Here the animation code for the animate package by Alexander Grahn:

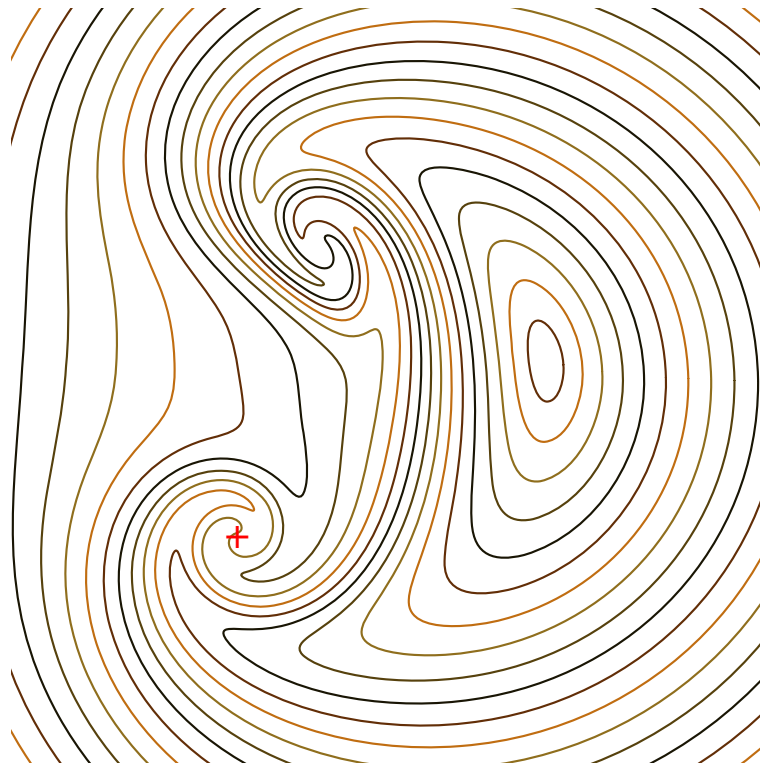
```
\begin{document}
\begin{animateinline}[%
controls,loop,
begin={\begin{pspicture}(-5,-5)(5,5)},
end={\end{pspicture}}
]{5}% 5 image/s
\multiframe{20}{rA=-3+0.65}{
\psMarble[
colors={
(622e07)
(c06d11)
(8f6e1d)
(56410d)
(191504)},
actions={
0 0 40 colors 30 concentric-rings
90 [3 400 24 tines] 40 200 31 rake
-90 [3 400 24 tines] 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]],
  colors={
    (622e07)
    (c06d11)
    (8f6e1d)
    (56410d)
    (191504)
  },
  actions={
    0 0 35 colors 33 concentric-rings
    90 100 shift
    0 0 -32e3 10 vortex
  }](10,10)
\end{pspicture}
```

Example 2:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,drawcontours,
linewidth=0.1,
bckg=false,
colors={
(622e07)
(c06d11)
(8f6e1d)
(56410d)
(191504)
},
actions={
0 0 35 colors 33 concentric-rings
90 100 shift
-200 200 -32e3 10 vortex
-200 -200 32e3 10 vortex
}](10,10)
\psdot[dotstyle=+,dotsize=0.25,linecolor=red](-2,-2)
\end{pspicture}

```

6.12 wiggle

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

wiggle affects the whole tank. In order to trace a wiggly rake in part of the tank, wiggle, then rake in part of the tank, then unwiggle (wiggle with negative depth).

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

demonstrates this.

angle period ofst depth wiggle

Applies sinusoidal wiggle: $y = \text{depth} * \sin(360 * x / \text{period} + \text{ofst})$.

angle	Wiggle will be perpendicular to angle from y-axis up.
period	Period of the sinusoidal wiggle (in degrees)
depth	Amplitude of the sinusoidal wiggle
ofst	Displacement of the sinusoidal wiggle (phase)

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 (depth) and angular velocity (period) $\omega = 5$: $f(x, y) = 50 \sin(5y)$, we will have:

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

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 (period): $\omega = 5 \times 360/1000 = 1.8$ and as we want it in degrees, we multiply by 360: $1.8 \times 360 = 648$

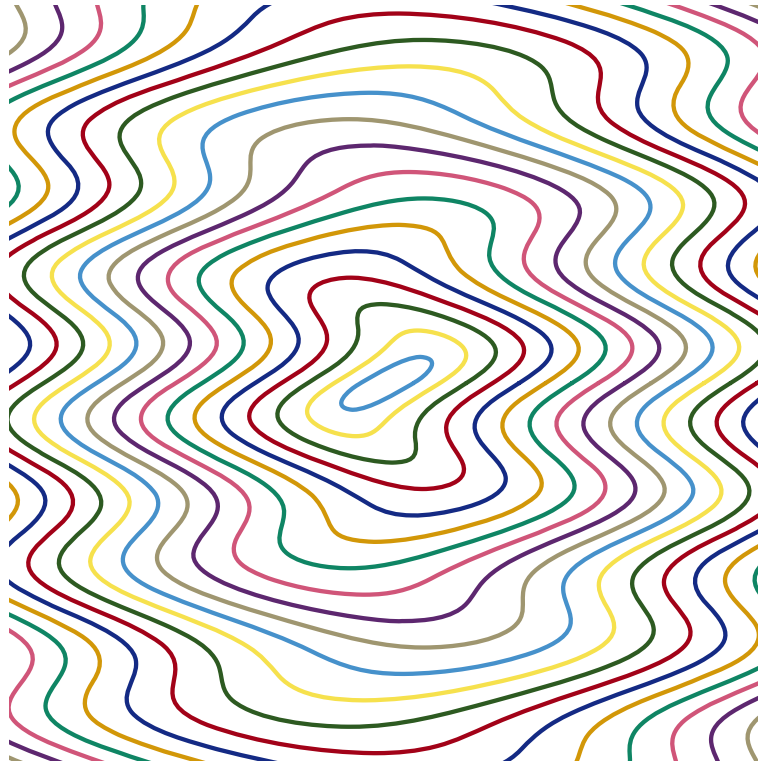


```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[drawcontours,bckg=false,
linewidth=0.2,
actions={
0 0 35 colors 33 concentric-rings
-90 648 0 50 wiggle
}](10,10)
\end{pspicture}
```

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 $f(x, y) = 50 \sin(5x)$, thus:

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



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[drawcontours,bckg=false,
linewidth=0.2,
actions={
0 0 35 colors 33 concentric-rings
0 648 0 50 wiggle
}](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={
    0 0 35 colors 33 concentric-rings
    45 648 0 50 wiggle
  }](10,10)
\end{pspicture}
```

Example 4:

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



```
\begin{pspicture}(-4,-4)(4,4)
\psMarble[
actions={
0 0 50 colors 25 concentric-rings
0 1080 0 30 wiggle
0 1080 0 -30 wiggle % reverse action
}](8,8)
\end{pspicture}
```

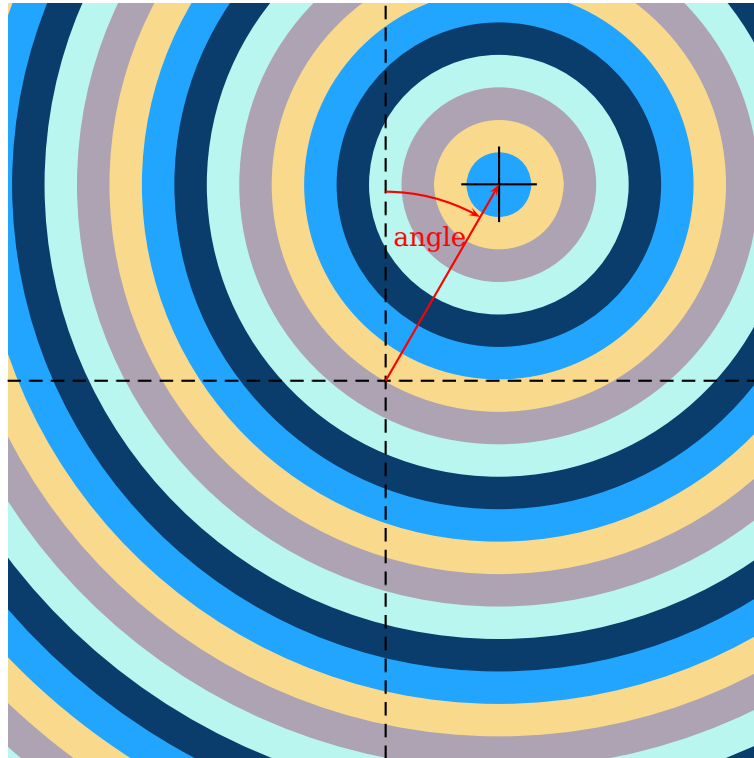
6.13 shift

angle rad shift

angle Angle from vertical (clockwise)

rad Length of displacement vector

The choice of units was made, so that 100 pts correspond to 1 cm within the image. rad is the length shifted.



The displacement vector is given by its angle α and its length rad in pts.

```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=50,
colors={
[0.134 0.647 1.000]
[0.977 0.855 0.549]
[0.684 0.638 0.702]
[0.730 0.965 0.942]
[0.040 0.236 0.424]
},
actions={
0 0 43 colors 32 concentric-rings
30 300 shift})(10,10)
\pstVerb{
/scaleFactor 10 1000 div def
/xS1 30 sin 300 mul scaleFactor mul def
/yS1 30 cos 300 mul 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[linestyle=dashed]{->}(0,0)(3;60)
\psarcn[linestyle=dashed]{->}(0,0){2.5}{90}{60}
\uput{1.7}[75](0,0){\color{red}angle}
\end{pspicture}
```

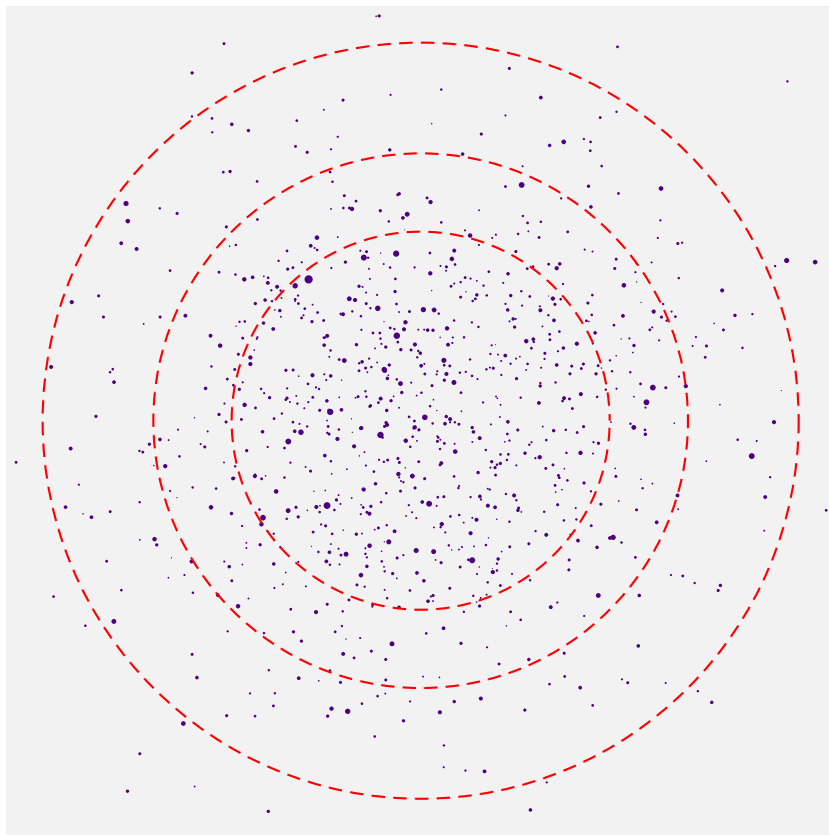
7 Spray actions

Spray actions are intended for drops small enough that they don't noticeably move paint boundaries. The radii of spray droplets are the cube roots of log-normal distributed values with mean R_d . Spray commands are performed after marbling!

7.1 Gaussian-spray

`xc yc r ang eccentricity [rgb] n Rd Gaussian-spray`

Places n drops of colors `[rgb]` randomly in a circular or elliptical disk centered at xc , yc having mean radius R_d , `ang` degrees clockwise from vertical, and length-to-width ratio `eccentricity`. For a circular disk, 63% of drops are within radius r , 87% of drops are within $r\sqrt{2}$, and 98% of drops are within radius $2r$.

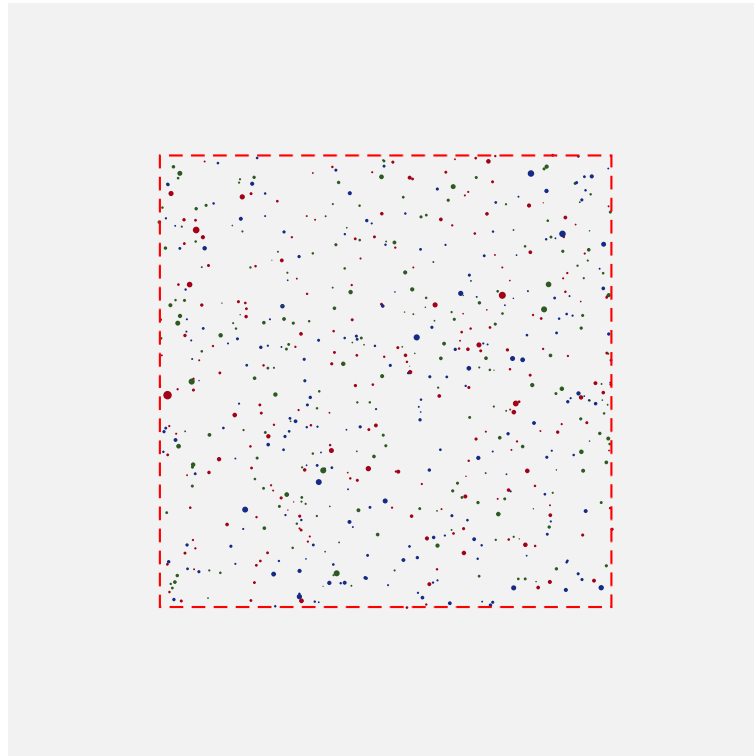


```
\begin{pspicture}(-5.5,-5.5)(5.5,5.5)
\psMarble[
  colors={[0.95 0.95 0.95]},
  spractions={
    0 0 250 -45 1 [0.3 0 0.5] 1000 3 Gaussian-spray
  }](11,11)
\pscircle[linecolor=red,linestyle=dashed](0,0){2.5}
\pscircle[linecolor=red,linestyle=dashed](0,0){!2.5 2 mul}
\pscircle[linecolor=red,linestyle=dashed](0,0){!2.5 2 sqrt mul}
\end{pspicture}
```

7.2 uniform-spray

`xc yc xsid ysid angle [rgb] n Rd uniform-spray`

Places `n` drops of colors `[rgb]` randomly in a `xsid` by `ysid` rectangle centered at location `xc`, `yc` and rotated by `ang` degrees clockwise from vertical.



```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
  colors={{[0.95 0.95 0.95]}},
  spractions={
    0 0 600 600 0 [[0.176 0.353 0.129][0.635 0.008 0.094][0.078 0.165 0.518]] 650 4 uniform-spray
  }](10,10)
\psframe[linecolor=red,linestyle=dashed](-3,-3)(3,3)
\end{pspicture}
```

8 Combined actions – Gallery

Note that `pst-marble` ships with an “examples” folder. Therein some example files contain some advanced PostScript techniques (for the interested PostScript user).

Example 1:



```
\begin{pspicture*}(-4,-1)(4,12)
\psMarble[viscosity=1000,
colors={
(36462a)
(4f6335)
(5d723c)
(78965b)
(a6a780)
},
actions={
0 0 45 colors 26 concentric-rings
-30 150 shift
-100 20 140
{/idx exch def
-270 idx sub -30 idx 2 mul add [-270 idx 3 mul sub] 10 90 50 stir
} for
0 720 0 10 wiggle
}](8,24)
\end{pspicture*}
```

Example 2:

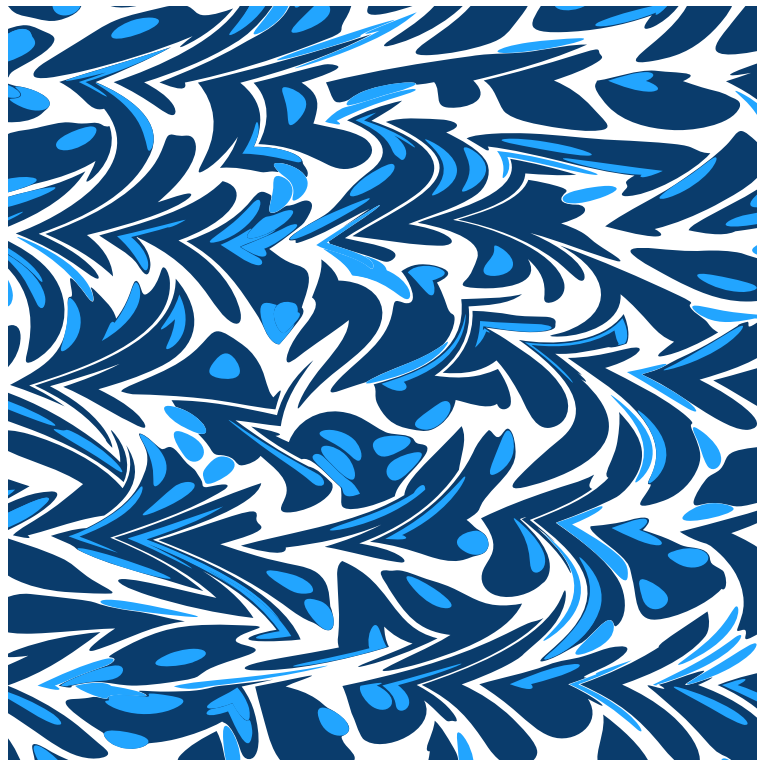
```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[viscosity=1000,
background=[[0.9 0.9 0.9]],
actions={
-400 200 400
{/idx exch def
0 idx 90 [ 5 200 0 tines ] [0.22 0.27 0.40] 80 line-drops
} for
-400 200 400
{/idx exch def
0 idx 90 [ 5 200 10 tines ] [0.49 0.75 0.79] 60 line-drops
} for
-400 200 400
{/idx exch def
0 idx 90 [ 5 200 -5 tines ] [0.90 0.80 0.47] 30 line-drops
} for
-400 200 400
{/idx exch def
0 idx 90 [ 5 200 -5 tines ] [0.98 0.27 0.32] 60 line-drops
} for
180 [11 100 0 tines] 50 100 30 rake
}](10,10)
\end{pspicture}

```

Example 3:

```
\begin{pspicture}(-5,-5)(5,5)
\psMarble[
actions={
0 0 40 colors 26 concentric-rings
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
0 0 1500 1500 0 [0.134 0.647 1] 250 18 uniform-drops
45 [6 200 0 tines] 40 200 31 rake
100 0 [-350] 30 30 15 stir
-90 [6 200 0 tines] 40 200 31 rake
0 0 [-150] 60 30 15 stir
}](10,10)
\end{pspicture}

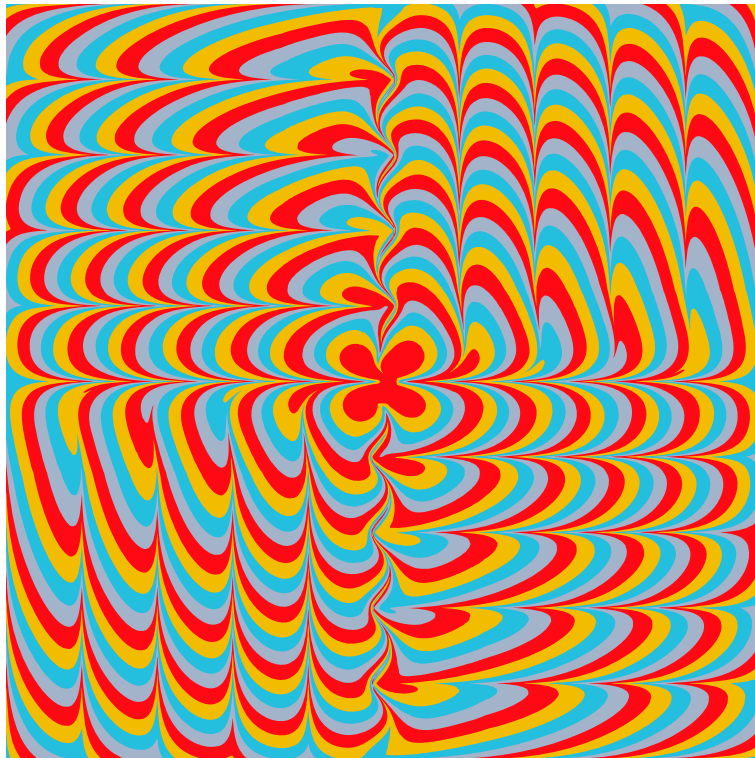
```

Example 5:

```

\newsstyle{YellowGlass}{linecolor=gray,linewidth=0.1}
\newsstyle{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]],
colors={
[0.27 0.01 0.02]
[0.78 0.02 0.10]
[0.77 0.92 0.47]
[0.11 0.18 0.07]
[0.96 0.85 0.10]
},
actions={
0 0 25 colors 15 concentric-rings
0 0 100 [0.78 0.02 0.10] drop
0 0 50 [0.77 0.92 0.47] drop
0 0 20 [0.11 0.18 0.00] 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 stylus
} for
}}%
\psMarble(10,10)
\PstLens[LensMagnification=2,LensRotation=50,LensSize=2,LensShadow=false,%
LensStyleGlass=YellowGlass](1,-1){
\psMarble(10,10)}
\end{pspicture}

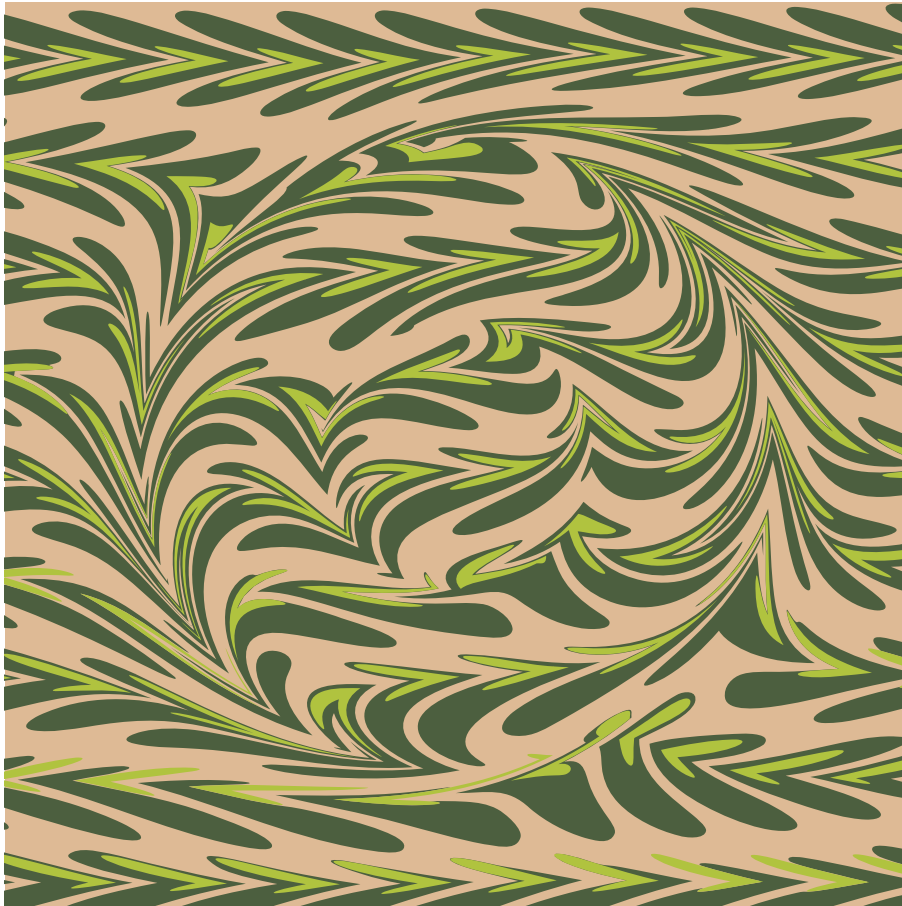
```

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 [0.64 0.70 0.79] drop
    0 0 rad [0.14 0.75 0.87] drop
    0 0 rad [0.95 0.74 0.00] drop
    0 0 rad [1.00 0.04 0.08] drop
  } for
-500 100 0 {/xpos exch def
  xpos -500 xpos 0 50 20 stylus
} for
0 100 500 {/xpos exch def
  xpos 500 xpos 0 50 20 stylus
} for
-500 100 0 {/ypos exch def
  500 ypos 0 ypos 50 20 stylus
} for
0 100 500 {/ypos exch def
  -500 ypos 0 ypos 50 20 stylus
} for
}}(10,10)
\end{pspicture}

```

Example 7:

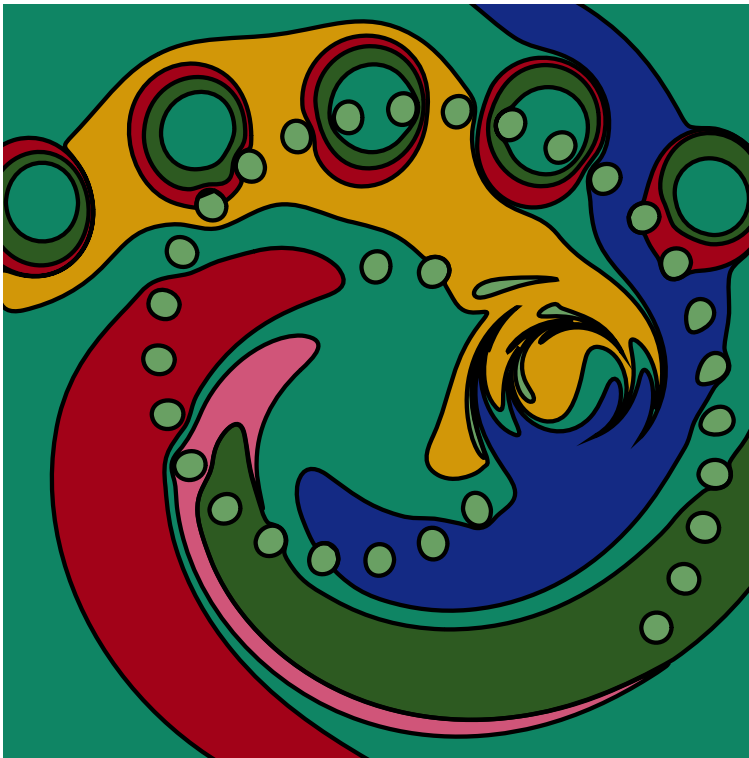
```

\begin{pspicture}(-6,-6)(6,6)
  \psMarble[
    background={
      [ 222 186 149 ]
    },
    viscosity=1000,
    actions={
      0 0 [ 12 100 -100 tines ] [ 12 98 80 tines ] 0 [ 76 95 63 ] 45 serpentine-drops
      0 0 [ 12 115 -100 tines ] [ 12 109.5 3 tines ] 0 [ 176 195 63 ] 20 serpentine-drops
      90 [ 9 228 18 tines ] 40 200 31 rake
      -90 [ 9 228 -132 tines ] 40 200 31 rake
      0 0 [ -350 ] 30 30 15 stir
      0 0 [ -150 ] 60 30 15 stir
    }
  ](12,12)
\end{pspicture}

```

Example 8:

```
\begin{pspicture}(-7.5,-8)(7.5,8)
\psMarble[
actions={
0 0 40 colors 26 concentric-rings
90 50 shift
0 0 [75 150 225 400 375 450] 10 180 30 stir
}](15,16)
\end{pspicture}
```

Example 9:

```

\begin{pspicture}(-5,-5)(5,5)
\psMarble[
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]
},
viscosity=1000,
actions={
0 0 600 colors 4 get drop
-200 -90 100 colors 5 get drop
-200 -200 200 colors 0 get drop
-200 200 200 colors 1 get drop
200 -200 200 colors 2 get drop
200 200 200 colors 3 get drop
0 0 32e3 750 vortex
0 -500 850 -30 212 -5 colors 1 get 5 50 coil-drops
0 -500 840 -30 215 5 colors 0 get 5 50 coil-drops
0 -500 850 -30 220 0 colors 4 get 5 50 coil-drops
0 0 150 0 70 10 [105 160 99] 35 20 coil-drops
250 0 [ 120 80 ] 20 100 20 stir
}](10,10)
\psMarble[
colors={
[0 0 0]
[0 0 0]
[0 0 0]
[0 0 0]
[0 0 0]
[0 0 0]
},
viscosity=1000,drawcontours,linewidth=0.2,linecolor=black,bckg=false,
actions={
0 0 600 colors 4 get drop
-200 -90 100 colors 5 get drop
-200 -200 200 colors 0 get drop
-200 200 200 colors 1 get drop
200 -200 200 colors 2 get drop
200 200 200 colors 3 get drop
0 0 32e3 750 vortex
0 -500 850 -30 212 -5 colors 1 get 5 50 coil-drops
0 -500 840 -30 215 5 colors 0 get 5 50 coil-drops
0 -500 850 -30 220 0 colors 4 get 5 50 coil-drops
0 0 150 0 70 10 [0 0 0] 35 20 coil-drops
250 0 [ 120 80 ] 20 100 20 stir
}](10,10)
\end{pspicture}

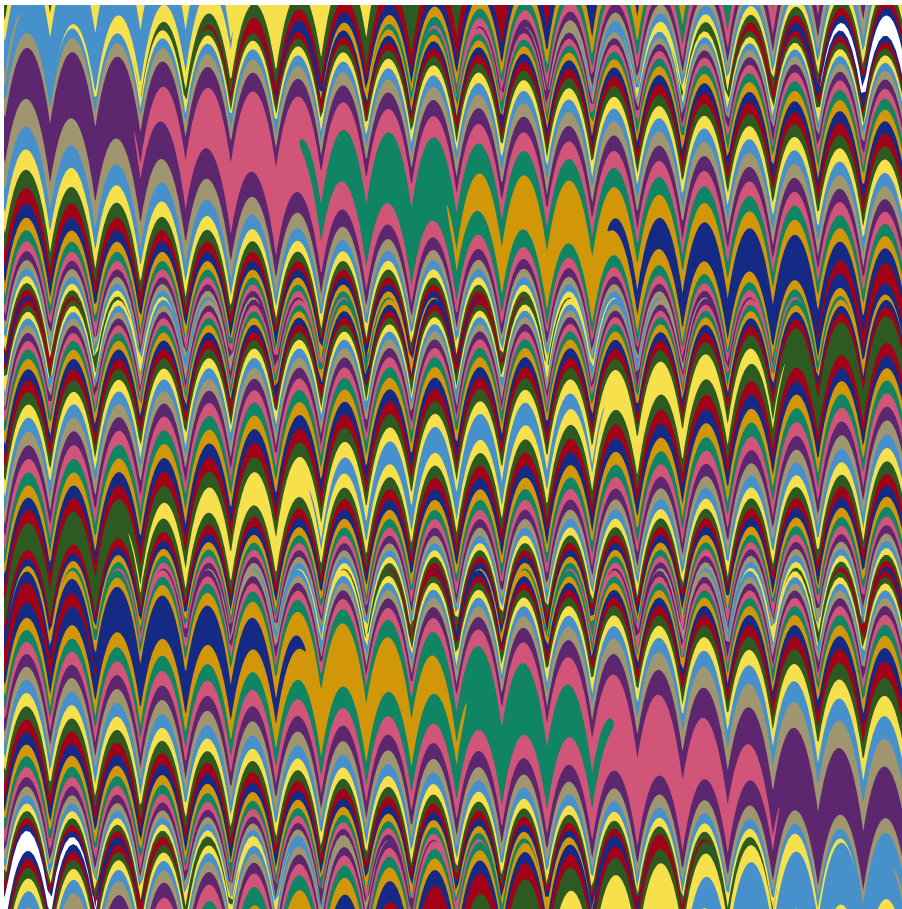
```

Example 10:

```

\begin{pspicture}(-6,-6)(6,6)
\psMarble[
  background={
    [0.1 0 0.1]
  },
  colors={(e7cc9b)(c28847)(80410b)},
  viscosity=1000,
  actions={
    %% coffee mug rim and stirred foam
    0 0 150 [.8 .8 .7] drop
    0 0 150 [.8 .9 .8] drop
    0 0 150 [.9 .9 .8] drop
    0 0 500 colors 2 get drop
    0 0 283 0 1 colors 1 get 30 30 Gaussian-drops
    0 0 -50e3 100 vortex
    %% tulip
    0 -250 30 colors 1 get drop
    0 -250 50 colors 0 get drop
    0 -200 20 colors 1 get drop
    0 -200 30 colors 2 get drop
    0 -150 30 colors 1 get drop
    0 -150 50 colors 0 get drop
    0 -100 20 colors 1 get drop
    0 -100 30 colors 2 get drop
    0 -50 30 colors 1 get drop
    0 -50 50 colors 0 get drop
    0 75 0 -450 20 30 stylus
    %% wreath
    0 10 390 90 95 0 colors 1 get 14 25 coil-drops
    0 0 400 90 100 0 colors 0 get 14 40 coil-drops
    0 0 [-400] 1 20 31 stir
    %% short vine
    -100 250 75 [ 6 80 0 tines ] colors 1 get 20 line-drops
    -100 250 75 [ 6 80 0 tines ] colors 0 get 30 line-drops
    -300 200 300 350 20 30 stylus
  },
  spractions={
    0 0 300 -45 1 [0.1 0.1 0.1] 1000 3 Gaussian-spray
  }
](12,12)
\end{pspicture}

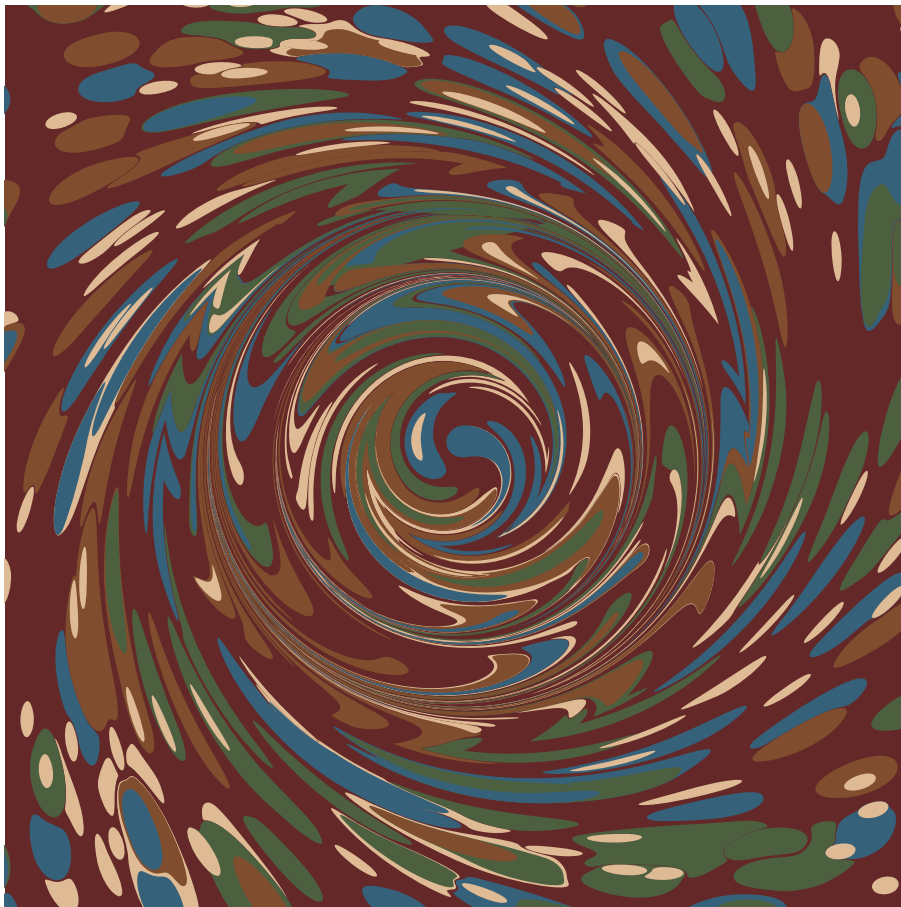
```

Example 11

```

\begin{pspicture}(-6,-6)(6,6)
  \psMarble[
    colors={
      [0.275 0.569 0.796][0.965 0.882 0.302]
      [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]
      [0.365 0.153 0.435][0.624 0.588 0.439]
    },
    actions={
      0 0 48 colors 25 concentric-rings
      90 [-150 450] 100 750 31 rake
      -90 [-150 450] 100 750 31 rake
      180 [ 25 50 0 tines ] 30 200 31 rake
      0 230 shift
    }
  ](12,12)
\end{pspicture}

```


Example 12

```

\begin{pspicture}(-6,-6)(6,6)
  \psMarble[
    background={
      [ 100 40 40 ]
    },
    colors={
      [ 76 95 63 ]
      [ 53 97 122 ]
      [ 128 78 46 ]
    },
    actions={
      0 0 1000 1000 0 [ 222 186 149 ] 85 1.72 10 mul uniform-drops
      0 0 1000 1000 0 colors 250 1.72 16 mul uniform-drops
      0 0 1000 1000 0 [ 222 186 149 ] 100 1.72 7 mul uniform-drops
      0 0 [ 100 ] 40 300 31 stir
      0 0 [ 200 275 ] 20 120 10 stir
      0 0 [ 325 ] 20 90 31 stir
    }
  ](12,12)
\end{pspicture}

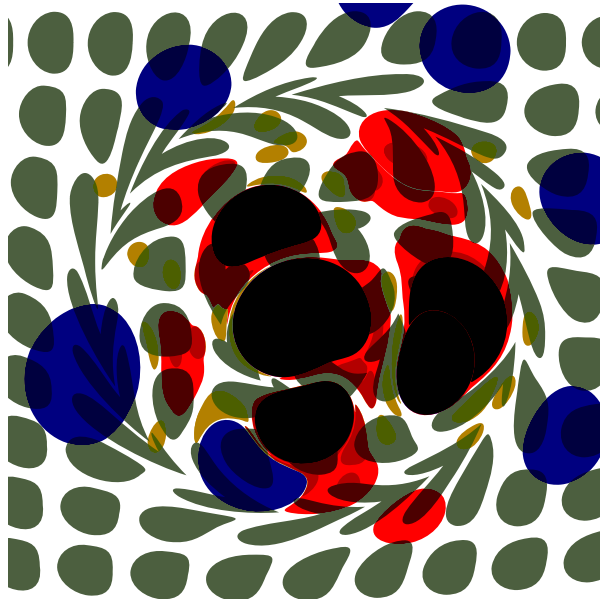
```

Example 13: Blendmodes

In case one want to overlap various marblings one can use the following blendmodes (basic option in PSTricks):
 /Normal: blendmode=0, /Compatible: blendmode=1, /Screen: blendmode=2, /Multiply: blendmode=3,
 /HardLight: blendmode=4, /Darken: blendmode=5, /Lighten: blendmode=6, /Difference: blendmode=7,
 /ColorDodge: blendmode=8, /ColorBurn: blendmode=9, /SoftLight: blendmode=10, /Hue: blendmode=11,
 /Saturation: blendmode=12, /Luminosity: blendmode=13, /Overlay: blendmode=14,
 /Exclusion: blendmode=15, /Color: blendmode=16.

Just set

```
\psMarble[blendmode=5,shapealpha=1, ...]
```



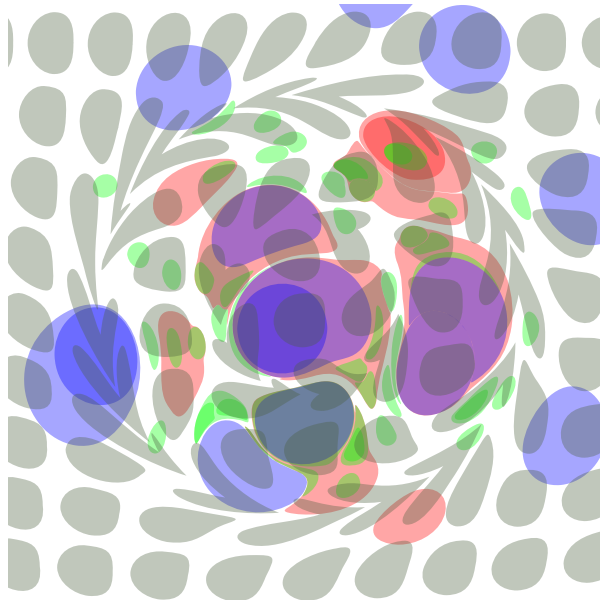
```
\begin{pspicture}(-4,-4)(4,4)
\psMarble[blendmode=5,shapealpha=1,viscosity=1000,
actions={
0 0 200 0 1 [1 0 0] 10 50 Gaussian-drops
0 0 200 0 1 [0.7 0.5 0] 50 20 Gaussian-drops
0 0 300 0 1 [0 0 0.5] 15 75 Gaussian-drops
}](8,8)
\psMarble[blendmode=5,shapealpha=1,viscosity=1000,bckg=false,
actions={
-300 92 500
{
0 exch 90 [ 12 100 -100 tines ] [ 76 95 63 ] 45 line-drops
} for
90 [11 200 0 tines] 40 200 31 rake
-90 [11 200 0 tines] 40 200 31 rake
0 0 [-350] 30 30 15 stir
0 0 [-150] 60 30 15 stir
}](8,8)
\end{pspicture}
```

Example 14: Transparency

In case one want to overlap various marblings one can also use transparency, which is a basic option in PSTricks `opacity=`. Just set

```
\psMarble[opacity=0.45, ...]
```

The values need to be from 0 to 1.



```
\begin{pspicture}(-4,-4)(4,4)
\psMarble[opacity=0.35,viscosity=1000,
actions={
0 0 200 0 1 [1 0 0] 10 50 Gaussian-drops
0 0 200 0 1 [0 1 0] 50 20 Gaussian-drops
0 0 300 0 1 [0 0 1] 15 75 Gaussian-drops
}](8,8)
\psMarble[opacity=0.35,viscosity=1000,bckg=false,
actions={
-300 92 500
{
0 exch 90 [ 12 100 -100 tines ] [ 76 95 63 ] 45 line-drops
} for
90 [11 200 0 tines] 40 200 31 rake
-90 [11 200 0 tines] 40 200 31 rake
0 0 [-350] 30 30 15 stir
0 0 [-150] 60 30 15 stir
}](8,8)
\end{pspicture}
```

9 Acknowledgments

Many thanks to D. P. Story who coded some additions to the `pst-marble.pro` file so it might be used for Adobe Distiller users.

The file size for the documentation could so be reduced tremendously.

Also many thanks to A. Grahn who sent a patch to use transparency and blendmode effects with the usual `PSTricks` options.