

# Magnetic Curves: Curvature-Controlled Aesthetic Curves Using Magnetic Fields

Ling Xu and David Mould

School of Computer Science, Carleton University, Canada

---

## Abstract

We describe “magnetic curves”, a particle-tracing method that creates curves with constantly changing curvature. It is well known that charged particles in a constant magnetic field trace out circular or helical trajectories. Motivated by John Ruskin’s advice to use variation in curvature to achieve aesthetic curves, we propose to continuously change the charge on a simulated particle so that it can trace out a complex curve with continuously varying curvature. We show some examples of abstract figures created by this method and also show how some stylized representational forms, including fire, hair, and trees, can be drawn with magnetic curves.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation

---

## 1. Introduction

Curves are a critical element in art and design. Abstract forms made from curves have appeared as ornamentation on virtually every type of object made by mankind, and abstract art using curves has existed since the first abstract art was created.

In *The Elements of Drawing* [Rus71], Ruskin advocates using curved forms in artistic compositions, and advises a constantly changing curvature: “Graceful curvature is distinguished... by its variation, that is to say, its never remaining equal in degree at different parts of its course... a steady change through the whole line, from less to more curvature, or from more to less, so that *no* part of the line is a segment of the circle.” Here, we suggest an algorithm for creating curves with continuously varying curvature.

In a magnetic field  $\vec{B}$ , a particle with charge  $q$  traveling with velocity  $\vec{v}$  experiences a force  $\vec{F} = q\vec{v} \times \vec{B}$ . For  $q$  and  $B$  constant, and no other forces, the particle will exhibit circular (helical) motion. If, however, we permit charge and/or field strength to vary, the particle will trace out a path whose curvature varies.

In this paper we suggest exploiting simulations of charged particles in a magnetic field to obtain aesthetic curves. Families of curves can be obtained by slight modifications of initial  $q$ , or of parameters of an equation governing  $q(t)$ . Fur-

ther, appealing branching structures can be produced in a simple way, by splitting a particle into two particles that are identical except for having charges of opposite sign. The two particles will evolve differently from that point.

The body of this paper describes the particulars of our magnetic curve formulation and gives some suggestions for variations and algorithmic art. We demonstrate the value of magnetic curves through many examples of two-dimensional artworks, including abstract forms, ornamentation, hair, fire, and trees. In closing, we suggest future abstractions and stylizations that could be produced using magnetic curves.

## 2. Previous Work

Computer-generated abstract forms have had a long history in computer graphics, going back almost to the very beginnings of the field [Die86]. In some works, curves serve merely as rendering primitives to convey the contents of the rendered space; Strothotte and Schlechtweg [SS02] give many examples in their survey of non-photorealistic rendering. In others, algorithmically computed curves are themselves the art; an early example is provided by Mezei [Mez69], while a more recent example is found in the work on computer-generated Celtic knots pioneered by Kaplan and Cohen [KC03]. More closely re-

lated to our work, Wong, Zongker, and Salesin created computer-generated patterns with floral motifs [WZS98]. The computer-generated ornament of Wong et al. used a grammar with serial rewriting and prevented the elements from overlapping by detecting and forbidding actions that would create an overlap. We use a similar approach to prevent overlap when creating our space-filling magnetic curves.

Our core algorithm depends on simulating the movement of particles that experience state-dependent forces. Particle tracing has a long history of use in modeling plants, from the earliest days of particle systems [RB85] to more recent results [NFD07]. We are not aware of work using particle tracing using magnetism for aesthetic purposes, although it is worth mentioning that Gruber and Glaeser [GG07] had aesthetic motives when they proposed creating minimal surfaces inspired by magnetism. Like the cited work on particle-traced trees and grass, and unlike the minimal surfaces, we seek to create curves: one-dimensional structures embedded in a higher dimensional space.

We can also use the particle tracing to outline figures, in the spirit of Curtis's loose and sketchy filter [Cur98], which uses particle tracing to portray boundaries. Vector graphics has been used since the earliest days of computer graphics, and is in widespread use today for 2D rendering; improving the capabilities of vector graphics is still an active area of research, as evidenced by the recent development of "diffusion curves" by Orzan et al. [OBW\*08]. Diffusion curves are intended for data representation and rendering, while our magnetic curves are intended for modeling. We should note that diffusion curves and magnetic curves are compatible: it is quite possible to extend the magnetic curves so as to add a blur value to the traced particle, and then render a raster output using a Poisson solver, rather than using the vector curves as output directly, as we do in most of the examples we show in this paper.

Cubic splines [FvDFH97] are likely the most common way of representing curves in modern computer graphics. We are not suggesting magnetic curves as an alternative to splines. Rather, we are suggesting magnetic curves as a mechanism for determining the paths, which can then be approximated by points, line segments, or higher-order polynomial segments.

### 3. Algorithm

A particle with charge  $q$  moving in a magnetic field  $\vec{B}$  experiences the Lorentz Force,  $\vec{F} = q\vec{v} \times \vec{B}$ , which causes a circular motion of the particle [LM75]. In our method, we restrict the particle to move in the  $xy$  plane, and the magnetic field must be in the  $z$  direction. Future states of the particle are obtained using numerical integration (in our implementation, we used forward Euler integration). For a constant magnetic field  $\vec{B}$  and a constant  $q$ , the particle traces out a circle in the  $xy$  plane.

When we change the particle charge  $q$  and the magnetic field  $B$ , the motion track of the particle is no longer a circle; instead, the particle can trace out some interesting curves. In the following, we will describe approaches to varying  $q$ , varying  $B$ , and to creating elaborate structures of branching and space-filling curves.

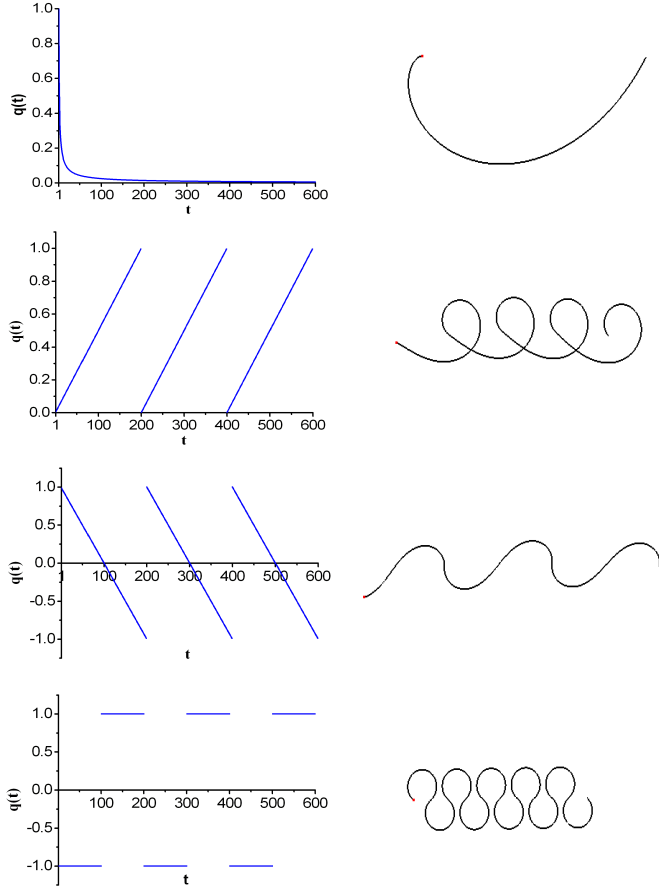
#### 3.1. Varying curvature

A constant acceleration perpendicular to the direction of motion will cause circular motion with radius  $r$ , where  $r = v^2/a$ . Combined with  $a = F/m = qvB/m$ , we can see that for fixed velocity and magnetic field, higher values of  $q$  give smaller value of radius  $r$ . If we increase  $q$ , the radius of the curve will decrease accordingly. Similarly, with a decreasing magnitude of  $q$ , the resulting curve will have an increased radius. If  $q$  changes continuously, the curvature of the path will also vary continuously. Note also that if we instantaneously change the sign of  $q$  while preserving its magnitude, the direction of acceleration will be instantaneously reversed, leading to a curve that winds in the opposite direction.

Figure 1 shows the relationship between  $q(t)$  and the resulting curves, where  $q$  is the charge and  $t$  is time. The red point is the initial position. In the first row, the left image shows  $q$  decreasing with time according to the function  $q(t) = t^{-\alpha}$ . Here we use  $\alpha = 0.7$ . The right image shows the resulting curve growing with an increasing radius. In the second row,  $q(t)$  is a triangular wave with amplitude 1 and mean 0.5. Again, the right image is the resulting curve; the point where the charge drops to 0 is visible as the location on the curve where it abruptly straightens (no charge means the particle will travel in a straight line). In the third row,  $q(t)$  is an inverted triangular wave with amplitude 2 and mean 0. It can be seen that reversing the charge does not create as obvious a discontinuity in the graph as simply setting the charge to an arbitrary different value. In the fourth row,  $q(t)$  is a square wave with 0 mean. Because the charge is constant in each half-period, the curve follows a piecewise circular route; whenever the sign of  $q$  flips, the circle is reversed. It holds less visual interest than the other graphs because of the large sections of constant curvature.

#### 3.2. Branching curves

Instead of tracking a single particle to get a single curve, we can use multiple particles to get complicated patterns. We suggest releasing new particles at intervals along the initial particle's path. We refer to the initial particle as the "parent particle" of the new particles. Each new particle will be tracked and produce a new curve. The process can be made recursive: we can spawn new particles from the child particle's path if we want. To create an effect reminiscent of vegetation, with curves growing to either side of the current branch, we set the sign of the child particles' charge randomly. Figure 2 shows an example of the process. In the left



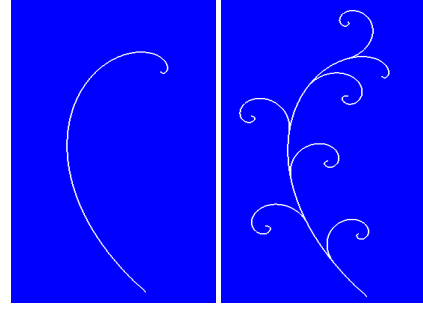
**Figure 1:** Different  $q(t)$  (left) and the resulting curves (right).

image, we grow a single curve. In the right image, we grow six new curves from six positions at equal intervals along the initial curve.

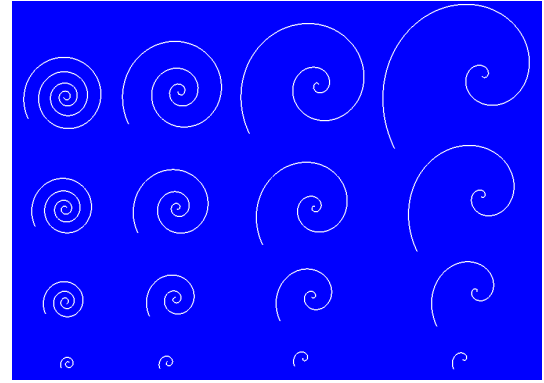
### 3.3. Space-filling curves

Here we demonstrate how branching magnetic curves can be used to fill space. We use the equation  $q(t) = (T - t)^{-\alpha}$ , where  $t$  is the elapsed time for a given curve and  $\alpha$  is a parameter that governs the rate of change of curvature for a curve;  $T$  dictates the length of a curve, where a given curve ranges from  $t = 0$  to  $t = T - 1$ . Figure 3 shows different curves with different values for  $T$  and  $\alpha$ . From top to bottom,  $T=650, 450, 250, 50$ ; from left to right,  $\alpha=0.8, 0.75, 0.7, 0.65$ . Observe that in each column from the curves become shorter as  $T$  decreases; in each row the spiral is looser as  $\alpha$  decreases.

We use curves grown with four pairs of  $(T, \alpha)$ :  $(T_1, \alpha_1)$ ,  $(T_2, \alpha_2)$ ,  $(T_3, \alpha_3)$ ,  $(T_4, \alpha_4)$ , where  $T_1 > T_2 > T_3 > T_4$  and



**Figure 2:** Left: a single curve; right: six new curves growing from the left initial curve.



**Figure 3:** Different curves with different  $T$  and  $\alpha$ .

$\alpha_1 > \alpha_2 > \alpha_3 > \alpha_4$ , to fill a space of size  $x_{\max}, y_{\max}$ . The process is described as follows.

We first release a particle at a position  $(x_0, y_0)$ . Here, we chose a position located at center bottom. Initially, the curve has parameters  $T_1$  and  $\alpha_1$ . We terminate the progress of a particle if either of the following are met: i) the particle travels beyond the boundary, that is  $x > x_{\max}$  or  $x < 0$  or  $y > y_{\max}$  or  $y < 0$ ; ii) the curve crosses previously drawn curves.

To detect whether a curve crosses other curves we use the following method. We divide the whole space into cells. Each cell stores an initial ID code to indicate that it is unoccupied. If a particle moves into one cell and either the cell is unoccupied or the cell is marked by its parent particle, the particle can survive; otherwise, we judge the curve crosses another curve. A formerly unoccupied cell is marked with the ID code of the entering particle. Crossing is prohibited; if a crossing event is detected, we remove the markers for the current particle and restart it at its initial position with the next smaller  $(T, \alpha)$  pair. If a crossing is detected when the smallest available  $T, \alpha$  is used, we abandon that position and move to the next position.

As we observed from Figure 3, the curve with largest value of  $T$  and  $\alpha$  occupies the largest space. If the curve with

$T_1$  and  $\alpha_1$  fails, we restart the particle at its initial position with  $T_2$  and  $\alpha_2$ . If  $T_2$  does not fit, we use  $T_3$  and  $\alpha_3$  and then  $T_4$  and  $\alpha_4$ . The intent behind doing so is to fill the space with the largest possible curves, while still being able to use the small curves to fill the remaining small spaces.

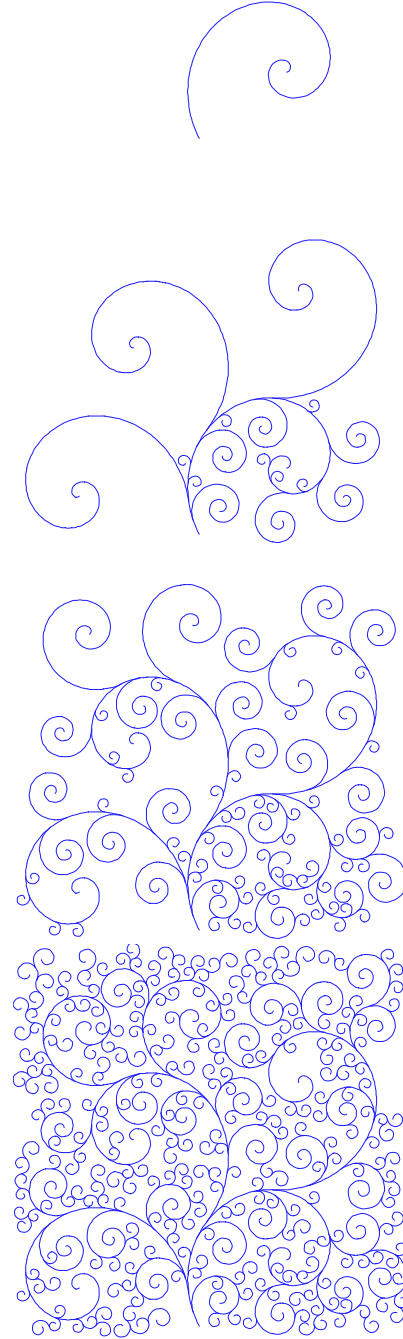
If we release new particles successively, we can obtain a space filled with different shapes of curves. The process is shown in Figure 4. The first image shows the initial curve. The second image shows different sizes of curves growing from the initial curve. Some curves are relegated to small size to fit in small spaces. The third image shows more curves growing from new curves in second image. The last image shows the final outcome, in which the curves fill the whole space. Note that while the process was demonstrated with a particular set of  $q(t)$  equations, it is a general process that (as implemented) can use any set of equations or distributions of parameters.

We also have the option of varying the magnetic field instead of using a constant magnetic field. Next, we show the effect of using Perlin noise to vary  $B$  spatially; in particular we take  $B_z$  from the noise, leaving the other two components at zero. We first build a lattice of size  $m \times m$ . For each node in the lattice we obtain a magnetic field value:  $B_z = 0.5 * (1 + \text{noise}(x/50, y/50))$ . We then choose a position in the lattice to release a particle and create a space-filling branching curve as described previously. At each time step, we update the velocity of our particle according to the magnetic field at its current position. The irregular values of  $B$  give irregular curves. In Figure 5 we visualize the magnetic field and the resulting curves. From this figure, we can observe that curves are much more irregular than the curves in Figure 4, while still possessing the same level of continuity of curvature, leading to a more visually appealing structure.

#### 4. Results

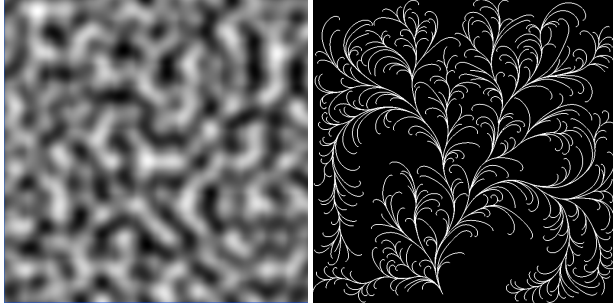
Magnetic curves are well suited to creating stylized depictions of certain classes of objects, including trees, hair, water, and fire. In addition, they can be used to create decoration and appealing abstract forms. In this section, we will show some examples of both representational and abstract structures created with magnetic curves.

Trees have been long studied in computer graphics and stylized depictions in the art world abound. Our branching structures can resemble stylized trees, and we will demonstrate this with a comparison to a historical artwork: Figure 6 shows the “Tree of Life”, created by Gustav Klimt. In this painting, the tree is composed of curvilinear branches. Our tree in a similar style is shown in Figure 7. In our image, new curve branches grow from one main branch and the space is filled with different sizes of curves. To achieve the result, we release a particle to grow the main curve. Then we release new particles on the main curve and repeat the process until



**Figure 4:** Iterations in creating the space-filling curve.





**Figure 5:** Left: the magnetic field with value of Perlin noise; right: the resulting irregular curves.

the space (top 3/4 of the image) is filled with different sizes of curves, using the method described in section 3.3 to avoid the curves crossing over each other. We are not intending to exactly reproduce Klimt’s tree, but to demonstrate that this kind of stylization can be captured by the magnetic curves. The spirals used in the “Tree of Life”, and present in some of Klimt’s other work, are easy to create with magnetic curves.



**Figure 6:** The painting “Tree of life” by Gustav Klimt.

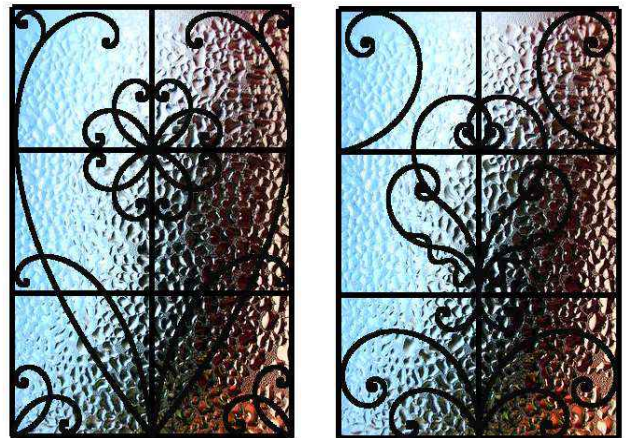
Architectural ornamentation on doors, windows, and arches is common; some examples appear in Figure 8. Here we use our method to generate two window frames, shown in Figure 9. We release particles at manually chosen initial positions and modify the initial velocity by hand to make the curves fit in the window. We use the method described in section 3.1 to get different shapes of curves. In our result, all patterns are composed of different curves with a background taken from a real glass image. Since we wanted a symmetrical pattern, we only built the right side of the frame and mirrored it. The resulting ornamentation is akin to existing designs. We use different functions of  $q(t)$  to get different shapes based on the method described in section 3.1. Most curves in the patterns use the equation  $q(t) = (T - t)^{-\alpha}$ . The difficult part is the design of pattern and finding proper  $T$  and  $\alpha$ . We tentatively select  $T$  and  $\alpha$  to make the curve fit in the window area. If the curve with current  $T$  and  $\alpha$  is too long and out of the window boundary, we decrease  $T$  or change  $\alpha$ .



**Figure 7:** Our stylized tree created with magnetic curves.



**Figure 8:** Architectural ornamentation. The window image comes from Richard Marcin and John Wahlert’s “The city of Kosice”. The door image comes from Cambridge 2000 Gallery.



**Figure 9:** Two window frames adorned with curves.

Figure 10 shows three different hair styles created by our method, overlaid on a hand-drawn image (also shown). To create each of the images, we grow the curves by releasing particles from points on a hand-drawn “hairline” curve. To obtain the upper right image, we set  $q(t) = (T - t)^{-\alpha}$  with large  $T$  and small  $\alpha$  values; we used different  $T$  and  $\alpha$  values for different strands, and reversed the sign of  $q$  partway along the curve so that the hair would look wavy. In the lower left image, we set  $q(t)$  to the sine function. In the lower right image, we set the curves growing with different time length  $T$  and different  $\alpha$ , so the hair is composed of curves with different lengths and curvatures. A closer look at yet another hairstyle can be seen in Figure 11; this was also produced using  $q(t) = (T - t)^{-\alpha}$ .



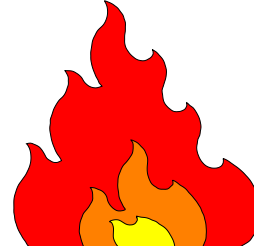
**Figure 10:** Three hair styles by our method.

We use our method to create a fire image with a curvilinear boundary as shown in Figure 12. We release an initial particle with a charge profile of  $q(t) = (T - t)^{-\alpha}$  and reverse its charge partway along its trajectory (at  $t = T/2$ ), causing the visible point of inflection. Subsequently, we terminate the particle and spawn a new one when a condition is met; in the depicted fire, the condition was that the  $x$  component of velocity was 0. The new particle is given a random velocity with a small uniform angular distribution about the positive  $x$  direction. When a sufficient number of repetitions have been performed (say 3 or 4) we start moving down the left side. The overall result is a boundary composed of piecewise continuous curves; the positions where the velocity changes happen are the tips of the flames. The resulting flame image is a conventional cartoon flame. A similar process could be used to produce conventional cartoon water waves.

Our remaining results comprise additional abstract curves. Figure 13 shows two abstract forms created by setting  $B$  according to Perlin noise, as described in section 3.3.



**Figure 11:** Another magnetic hairstyle.



**Figure 12:** Cartoon flames created with magnetic curves.

Each point on the curve is colored by taking one RGB color channel from the local noise value and setting the other two to zero. These are space-filling curves with irregular spatial variation in their curvature, yielding a subjectively pleasing structure.

Figure 14 shows curves grown in a magnetic field by setting  $B$  according to Perlin noise. To create this figure, we tracked a single particle with a fixed charge as it wandered within the image area. Whenever it reached the image boundary, we reversed its velocity, ensuring that it remains within the image area. Every time it reentered the image area, we assigned it a different random color, determined by assigning each RGB channel an independently chosen value from a uniform random distribution. From the figure we can see the different colors of curves which represent different periods of time spent continuously within the image area. While we did not find this image as straightforwardly attractive as the space-filling curve, we confess to a certain fas-



**Figure 13:** Two abstract curvilinear forms.



**Figure 14:** Meanderings of a charge in a varying magnetic field.

cination with the unpredictable meanderings of the particle and the resulting tangle of curves.

## 5. Discussion

The main strength of the method lies in its ability to produce constantly and smoothly varying curvature within the well-understood framework of particle systems. The ability to create branches by negating  $q$  is also an asset.

The limitation of the method is the difficulty of controlling the exact course of the curve. This is a limitation held in common with other particle tracing methods.

We have not yet much explored the possibility of allowing  $q$  to vary according to other elements of the particle's state vector. For example, we could have  $q$  depend on position, or on the direction of the particle's travel. We demonstrated some position-dependent effects by varying  $B$  spatially; more investigation is warranted, considering the success of the results obtained so far.

Image	# curves	processing time
Space-filling example	312	1.5s
Stylized Tree	72	0.7s
Perlin curves	272	1.0s
Hair	376	4.3s
Window	7	0.05s

**Table 1:** Timing results for some of the magnetic curves.

Table 1 summarizes timing results, given with respect to a 1.8GHz P4 with 512 MB RAM. The process is not particularly demanding, and computer times are only a few seconds per image. Note that the hair timing includes the time needed to load files (the face image and an image containing the hairline to spawn hair particles from). In terms of human time, the process can be time-consuming; we expect that most users lack intuition for controlling curves by modifying charges and magnetic fields. It took on the order of two minutes to find the appropriate parameters for each curve on the window.

We have showed one method for creating branching curves, but we have not much considered the aesthetics of curve thickness as it relates to branching. Consider Figure 15, which reproduces an image created by Alfons Mucha [Hof84] near the beginning of the 20th century. This image served as inspiration for our own hair results, but also points towards using curve thickness to add richness to magnetic curves.

An obvious limitation of our present implementation is its restriction to 2D. The idea of magnetic curves extends straightforwardly to three dimensions, and we hope both to create curves in three dimensions and to create 2D curves on the surfaces of complex structures. Magnetic fields could vary spatially or could be textures on the surfaces of objects.





**Figure 15:** Art by Alfons Mucha: inspiration for magnetic curves.

## 6. Conclusions

In this paper, we proposed simulating magnetic forces to create curves with continuously varying curvature. The resulting curves can be used to create appealing abstract forms. They are also a potential tool for creating stylized depictions of classes of objects and phenomena that are well described by curves; such phenomena include trees, hair, and conventionally-drawn cartoon flames.

Other potential uses of the method abound. Water waves are conventionally stylized with a repeating structure similar to the cartoon flames we showed. We showed only one stylized tree (after Klimt) but other vegetation, including trees but also vines and leaves, can be depicted also. (Indeed, a variety of leaf outlines can be created by adapting our approach for cartoon flames.) We showed a few hairstyles; we envision possibilities for also drawing stylized fur, feathers, cloth, and smoke. Finally, we showed a few possibilities for purely abstract forms, but experimentation will reveal a much wider variety of such forms; our hope is that the constantly changing curvature, governed by formulas for  $q(t, x, y, z)$  and  $\vec{B}(t, x, y, z)$ , will give the resulting curves high aesthetic value.

## References

- [Cur98] CURTIS C.: Loose and sketchy animation. In *SIGGRAPH '98 Technical Sketch* (1998), ACM Press, p. 317.
- [Die86] DIETRICH F.: Visual intelligence: The first

- decade of computer art (1965-1975). *Leonardo* 19, 2 (1986), 159–169.
- [FvDFH97] FOLEY J. D., VAN DAM A., FEINER S. K., HUGHES J. F.: *Computer Graphics - Principles and Practice*. Addison-Wesley, 1997.
- [GG07] GRUBER F., GLAESER G.: Magnetism and minimal surfaces - a different tool for surface design. In *Computational Aesthetics* (2007), Eurographics, pp. 81–88.
- [Hof84] HOFSTATTER H. H.: *Art Nouveau: Prints, Illustrations And Posters*. Greenwich House, 1984.
- [KC03] KAPLAN M., COHEN E.: Computer generated Celtic design. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 9–19.
- [LM75] LOBKOWICZ F., MELISSINOS A. C.: *Physics for Scientists and Engineers*. W.B.Saunders, 1975.
- [Mez69] MEZEI L.: SPARTA: A procedure oriented programming language for the manipulation of arbitrary line drawings. In *Proceedings of IFIP 69* (1969), pp. 567–604.
- [NFD07] NEUBERT B., FRANKEN T., DEUSSEN O.: Approximate image-based tree-modeling using particle flows. *ACM Transactions on Graphics* 26, 3 (2007), FIX ME.
- [OBW\*08] ORZAN A., BOUSSEAU A., WINNEMÖLLER H., BARLA P., THOLLOT J., SALESIN D.: Diffusion curves: a vector representation for smooth-shaded images. In *Proceedings of SIGGRAPH 2008* (New York, NY, USA, 2008), ACM, pp. 1–8.
- [RB85] REEVES W. T., BLAU R.: Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Proceedings of SIGGRAPH '85* 19, 3 (1985), 313–322.
- [Rus71] RUSKIN J.: *The Elements of Drawing*. Dover Publications, 1971.
- [SS02] STROTHOTTE T., SCHLECHTWEG S.: *Non-Photorealistic Computer Graphics*. Morgan Kaufmann, 2002.
- [WZS98] WONG M. T., ZONGKER D. E., SALESIN D. H.: Computer-generated floral ornament. In *Proceedings of SIGGRAPH '98* (1998), ACM Press, pp. 423–434.