A Path Planning Strategy for Physically Based Animation

Jesse Reiter

March 20, 1995

1 Introduction

Consider the problem of animating the motion of a bird in flight. Aside from the engineering problem of determining and actuating the mechanics of locomotion, we must be able to find a path that will lead the bird from one point in space to another. A satisfactory flight plan must avoid a set of obstacles, both local (*e.g.*, trees) and global (*e.g.*, the ground), maneuvering the bird successfully about them. Once a plan is obtained, it becomes a matter of solving a series of *forward* and *inverse kinematics* problems to achieve the desired motion. A related but simpler problem is that of a cyclist trying to get from one point to another. Like the bird, he must maneuver his way successfully between obstacles to avoid collision, yet his movements are strictly planar. The discussion in this paper shall be restricted to the simpler problem.

Supplementing the various geographical constraints, a number of restrictions will be imposed on the actual path. First, a realistic demand for animation would be to prescribe velocity vectors at the endpoints of the path, providing the speed and direction of motion at these points. Second, in order to maintain the cyclist's balance and sustain the minimal "lift" required to keep a bird in air, the velocity at every point on the path must be confined to lie within certain bounds. Third, it would prove beneficial to further restrict the curvature of the path, keeping it synchronized with the velocity at every point along the way, in order to maintain a reasonable level of strain. For example, sharp turns at high speed aught to be avoided for they give rise to a severe centripetal force that may cause the cyclist to loose traction, skid, or "jack-knife", or might cause a bird to stall. Finally, in general Euclidean space if there exists a path connecting two points which satisfies all the above constraints, then an infinite number of such paths exist. The best one, then, is to be selected.

The solution to the problem finds a path consisting of a piecewise collection of curves, that manages to avoid all obstacles while doing its best to satisfy the constraints. The method followed in searching for this path can be divided into two parts: collision resolution and optimization. A rough pseudocode algorithm is the following:

```
while (progress is being made)
   resolve collisions in the path
   optimize
```

The two steps are mutually exclusive. The resolution process is solely concerned with the removal of obstacles and obeys none of the other constraints. Conversely, the optimization has no knowledge of the geography and may produce a collision course. Thus, a two pronged approach is taken. But while the first step results in an absolute solution in the sense that no collisions are tolerated, the second does not. Rather, it utilizes search techniques to find a curve that only *best* satisfies the constraints, since a curve that successfully satisfies them all may not exist. In light of this strategy, most of the paper is dedicated toward describing the constraint optimization method. The collision resolution problem is discussed briefly and a solution is proposed.

2 Obtaining a Curve

Finding a curve $\mathbf{P}(t)$ parameterized over the interval $[0,1] \in \mathbf{R}$ that satisfies all four prescribed end-point conditions is a well known problem[1]. If we let the initial and final positions and the initial and final velocities be $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{T}_0$, and $\mathbf{T}_1$ respectively, we get the following set of initial conditions:

$$\mathbf{P}(0) = \mathbf{P}_0 \qquad \mathbf{P}(1) = \mathbf{P}_1 \qquad \dot{\mathbf{P}}(0) = \mathbf{T}_0 \qquad \dot{\mathbf{P}}(1) = \mathbf{T}_1, \tag{1}$$

where the dot in $\dot{\mathbf{P}}$ signifies partial differentiation w.r.t. t. If $\mathbf{P}$ is a cubic polynomial, say

$$\mathbf{P}^3(t) = \mathbf{b}t^3 + \mathbf{c}t^2 + \mathbf{d}t + \mathbf{e} \tag{2}$$

with the superscript on $\mathbf{P}$ denoting the degree, then $\dot{\mathbf{P}}^3(t) = 3\mathbf{b}t^2 + 2\mathbf{c}t + \mathbf{d}$ and some simple algebra yields

$$\begin{aligned}
\mathbf{b} &= 2(\mathbf{P}_0 - \mathbf{P}_1) + \mathbf{T}_0 + \mathbf{T}_1 \\
\mathbf{c} &= 3(\mathbf{P}_1 - \mathbf{P}_0) - 2\mathbf{T}_0 - \mathbf{T}_1 \\
\mathbf{d} &= \mathbf{T}_0 \\
\mathbf{e} &= \mathbf{P}_0.
\end{aligned}$$

The can be represented succinctly in matrix form as

$$\mathbf{P}^3(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{T}_0 \\ \mathbf{T}_1 \end{pmatrix},$$

where the rightmost product is to be understood as componentwise multiplication. $\mathbf{P}^3(t)$ is called the *cubic Hermite curve.*

The cubic curve, however, is completely determined by the four initial conditions, and is unable to accommodate any further constraints. Additional flexibility may be gained by increasing the degree of the interpolating polynomial, to get the *quartic Hermite curve*

$$\mathbf{P}^4(t) = \mathbf{A}t^4 + \mathbf{B}t^3 + \mathbf{C}t^2 + \mathbf{D}t + \mathbf{E}. \tag{3}$$

[1] Traditionally the interval $[0,1]$ or $[0,L]$ (for some $L \in \mathbf{R}$) is used, but any closed interval $[x,y] \in \mathbf{R}$ will do equally well.

Using the initial conditions (1) in (3), some slightly tedious algebraic manipulations and rearrangements will give us the coefficients of the polynomial:

$$\begin{aligned}
\mathbf{B} &= 2(\mathbf{P}_0 - \mathbf{P}_1) + \mathbf{T}_0 + \mathbf{T}_1 - 2\mathbf{A} \\
\mathbf{C} &= 3(\mathbf{P}_1 - \mathbf{P}_0) - 2\mathbf{T}_0 - \mathbf{T}_1 + \mathbf{A} \\
\mathbf{D} &= \mathbf{T}_0 \\
\mathbf{E} &= \mathbf{P}_0.
\end{aligned}$$

Determining the value of the leading coefficient $\mathbf{A}$ uniquely will require one additional initial condition, but that will once again fix the curve.

We have thus obtained the family of quartic parametric curves given by

$$\mathbf{P}^4(t) = \mathbf{A}t^4 + [2(\mathbf{P}_0 - \mathbf{P}_1) + \mathbf{T}_0 + \mathbf{T}_1 - 2\mathbf{A}]t^3 + [3(\mathbf{P}_1 - \mathbf{P}_0) - 2\mathbf{T}_0 - \mathbf{T}_1 + \mathbf{A}]t^2 + \mathbf{T}_0 t + \mathbf{P}_0. \quad (4)$$

Rearranging (4) to isolate $\mathbf{A}$, gains us greater insight into the behavior of these polynomials:

$$\begin{aligned}
\mathbf{P}^4(t) &= t^2(t-1)^2\mathbf{A} + [2(\mathbf{P}_0 - \mathbf{P}_1) + \mathbf{T}_0 + \mathbf{T}_1]t^3 + [3(\mathbf{P}_1 - \mathbf{P}_0) - 2\mathbf{T}_0 - \mathbf{T}_1]t^2 \\
&\quad + \mathbf{T}_0 t + \mathbf{P}_0 \\
&= t^2(t-1)^2\mathbf{A} + \mathbf{b}t^3 + \mathbf{c}t^2 + \mathbf{d}t + \mathbf{e} \\
&= q(t)\mathbf{A} + \mathbf{P}^3(t), \qquad (5)
\end{aligned}$$

where $q(t) = t^2(t-1)^2$ is another quartic polynomial in t with roots at 0 and 1 (see figure 1). It is immediately apparent, then, that the polynomial $\mathbf{P}^4$ is just a modified version of the cubic one $\mathbf{P}^3$, and that both agree at the endpoints of the parameter domain[2]. We call the parameter $\mathbf{A}$ the *modifier*, and the polynomial $q(t)$ the *modification factor*. At each point t on the path, $q(t)$ is a scalar determining the magnitude of modification in the direction of $\mathbf{A}$ (see, *e.g.,* figures 5 and 7). We can, thus, satisfy our additional constraints by perturbing $\mathbf{P}^3$ at every point in a given direction, by a factor of q. In each specific case, the modifier $\mathbf{A}$ must be determined explicitly to obtain the desired effects.

Having found a family of curves that satisfy the requirements, each member must be evaluated in order to select the one best suited for the given constraints. The evaluation will be carried out using an energy functional that involves three criteria – arclength, speed, and curvature.

arclength The arclength of a parametric curve $\mathbf{P}(t)$ over the interval $[a, b]$ is given analytically in [2] and [4] by

$$A(\mathbf{P}; [a, b]) = \int_a^b \|\dot{\mathbf{P}}(t)\| dt$$

where $\dot{\mathbf{P}}$ denotes the componentwise derivative of $\mathbf{P}$ w.r.t. t. Solving this integral can be somewhat gruesome, as in general the integral has no closed form. An approximation to the solution could be found using numerical methods, however we prefer a different

[2]In fact, $\mathbf{P}^3$ can be identified as a special case of $\mathbf{P}^4$ with the parameter value $\mathbf{A}$ set to 0.

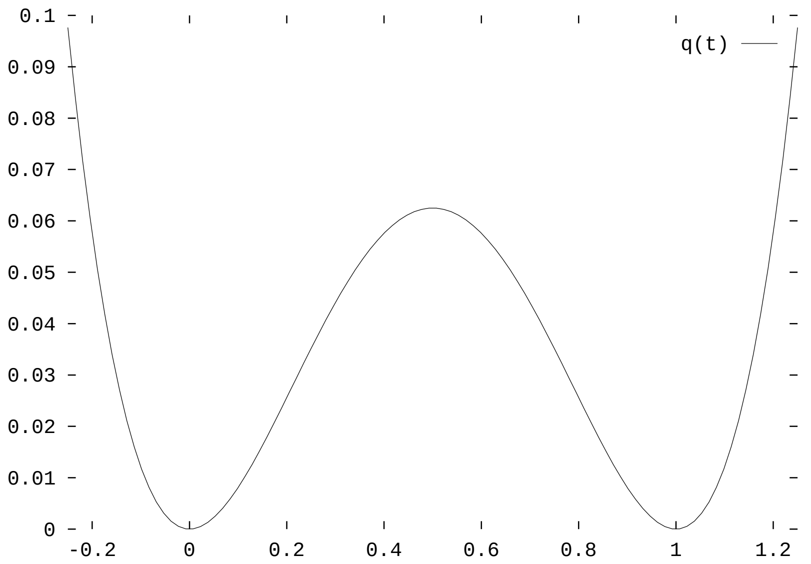


Figure 1: The modification factor $q(t) = t^2(t-1)^2$

means. We approximate the arclength over a uniform partition of the domain interval using chord length:

$$A(\mathbf{P}; [a,b]) \approx \sum_i \frac{\|\mathbf{P}(a + i\Delta t) - \mathbf{P}(a + (i-1)\Delta t)\|}{\Delta t}, \tag{6}$$

where $\Delta t = (b-a)/n$, and $i = 1, \ldots, n$ for some positive integer n. This sum is in fact just a Riemann sum, and approaches the integral by definition as $\Delta t \to 0$.

We can also define a continuous arclength functional $f_P(t) = A(\mathbf{P}; [a,t])$ giving the arclength of $\mathbf{P}$ between a and any point $t \in [a,b]$ in the domain. The chordlength approximation (6) will be used to evaluate f_P.

speed The *velocity* of the moving object is given by the time derivative of its trajectory $\mathbf{P}$. It has a magnitude – which may be positive or negative – and a direction. The *speed*, conversely, is a non-negative scalar defined as the unsigned length of the velocity vector. It can be estimated using the chordlength approximation (6), as

$$s(t_i) = \frac{f_P(t_i) - f_P(t_{i-1})}{\Delta t}, \tag{7}$$

where $t_i = a + i\Delta t$. Note that the speed is always non-negative since f_P is strictly increasing. When the parameter domain is $[0,1]$, Δt reduces to $1/n$ and (7) becomes

$$s(t_i) = n \times [f_P(t_i) - f_P(t_{i-1})].$$

curvature The curvature of a trajectory at a given point is defined as the rate of change of the unit tangent vector at that point. It is given by the second arclength derivative of the curve (see [1]). As such, an analytic evaluation of the curvature necessitates the reparameterization of the $\mathbf{P}$, which leads to an unseemly mess. Reverting to the arclength approximation of the speed, an estimate of the curvature is obtained:

$$\kappa(t_i) = \frac{s(t_i) - s(t_{i-1})}{\Delta t}.$$

Once again, this simplifies to $\kappa(t_i) = n \times [s(t_i) - s(t_{i-1})]$ over the unit domain.

An interesting point regarding the curvature is noted by Lee in [3]. Taking the time derivative of $\mathbf{P}$ gives $d\mathbf{P}/dt = (d\mathbf{P}/du)(du/dt)$, where du is the arclength component of $\mathbf{P}$. Differentiating again, yields

$$\begin{aligned} \frac{d^2\mathbf{P}}{dt^2} &= \frac{d^2u}{dt^2}\frac{d\mathbf{P}}{du} + \left(\frac{du}{dt}\right)^2 \frac{d^2\mathbf{P}}{du^2} \\ &= \frac{d^2u}{dt^2}\mathbf{T} + \left(\frac{du}{dt}\right)^2 \kappa\mathbf{N}, \end{aligned}$$

where $\mathbf{T}$, and $\mathbf{N}$ denote the unit tangent vector, and unit normal vector of $\mathbf{P}$ at t respectively. To ensure a smooth "ride" along the trajectory we would require the tangential component of the acceleration to be negligible. This reduces the magnitude of the second time derivative to be roughly $\kappa(du/dt)^2 = \kappa s(t)^2$,which is the centripetal force acting on the object. We would, of course, like to keep this minimized.

We are now finally ready to evaluate the *goodness* of a given curve. Writing s_i and κ_i for $s(t_i)$ and $\kappa(t_i)$ respectively, two error functionals, E_{speed} and $E_{curvature}$, are defined on the curve $\mathbf{P}^4$ for a given (fixed) parameter $\mathbf{A}$:

$$E_{speed}(\mathbf{P}^4; \mathbf{A}) = \sum_i [\max(s_i - s_{max}, 0) + \min(s_i - s_{min}, 0)]^2 \tag{8}$$

and

$$E_{curvature}(\mathbf{P}^4; \mathbf{A}) = \sum_i [\max(\kappa_i - \kappa_{max}, 0) + \min(\kappa_i - \kappa_{min}, 0)]^2, \tag{9}$$

where $s_{max} = \max_i\{s_i\}$, $s_{min} = \min_i\{s_i\}$, and κ_{max} and κ_{min} are similarly defined. These functionals measure how far the given curve protrudes from the confines of the speed and curvature bounds. Aided by these, the total energy functional can be defined as

$$E(\mathbf{P}^4; \mathbf{A}) = \sqrt{E_{speed} + E_{curvature} + [f_P(b)]^2}. \tag{10}$$

The arclength term is included in (10) because we would nominally like to find the shortest curve satisfying all the constraints.

2.1 Results

The evaluator (10) permits selection of the best curve in the following fashion. First, the parameter space is randomly searched for the best modifier. The number of random selections made is user defined, and in our simulation was taken to be 1000. Once the best modifier is found, it is used as the basis of a constant step-size, gradient-descent search. The search terminates at a minimal point in the search space. An interesting experiment would be to determine whether this final point is actually the *global* minimum or just *locally* minimal. In our experience, the best modifier found using this search technique offered no significant improvement over ones found by skipping the initial random search and basing the gradient descent at the null modifier, $\mathbf{A} = \mathbf{0}$ (*i.e.,* when $\mathbf{P}^4 = \mathbf{P}^3$). This lends credence to the claim that the minimum is, in fact, global.

In the graphs below, the term *clamped* is used to describe a tight bound on the speed or curvature (or both) for the given set of initial conditions. The energy functional graphed is a slight modification of (10) that does not include the arclength term:

$$\hat{E}(\mathbf{P}^4; \mathbf{A}) = \sqrt{E_{speed} + E_{curvature}}$$

Figure 3 shows a curve whose initial and final velocities are equal, but go against the general direction of motion. The cubic curve is straight line, as the path begins with the initial velocity, stops, backs up on itself, overshoots its destination, stops again, backs up and finally arrives at its destination at the required velocity. The speed and curvature on the quartic curve, conversely, are clamped to disallow stopping, which leads to the much smoother trajectory. Figure 4 graphs the speed, curvature, and energy consumption by arclength of the quartic curve.

Figures 5 and 7 show curves whose initial and final velocities are identical, called *s-curves* because of their shape. The initial conditions of both curves are identical; the only difference being the bounds on speed and curvature. Figure 5 shows a trajectory with tight bounds on both speed and curvature. The bounds on curvature were relaxed in figure 7. Compare the similarity of the quartic curve to the cubic one when the constraints are unclamped. Figures 6 and 8 graph the speed, curvature, and energy functionals by arclength.

Finally, figures 9 and 11 show curves whose initial and final velocities are equal in magnitude but are opposite in orientation. Notice the difference between the clamped 9 and the unclmaped 11. Figures 10 and 12 give their energy plots.

3 Collision Resolution

Although collision resolution has not been implemented, a naive solution would be the following. Surround each obstacle, O, with a k-dimensional open ball $B^k(\mathbf{c}, r)$, centered at $\mathbf{c}$ and having radius r. When two balls interpenetrate, say $O_i \cap O_j \neq \phi$ (for $i \neq j$), define O_{ij} to be the smallest ball containing B_i and B_j. This can be done simply by setting the common radius to be the average of the diameters ($r_{ij} = (2r_i + 2r_j)/2 = r_i + r_j$), and setting the common center, $\mathbf{c}_{ij}$, appropriately on the line segment connecting the two centers. Interpenetration of a greater number of obstacles can easily be generalized from this. It may be assumed, then, that the set of geographical constraints is given by a collection of non-intersecting balls. We will use O_i and B_i interchangeably.

Suppose, now, that the quartic Hermite curve segment $\mathbf{P}$ connects points $\mathbf{u}$ and $\mathbf{v}$. To resolve collisions, we test for intersection of the curve with every obstacle O_i. If $\mathbf{P} \cap O_i = \phi$

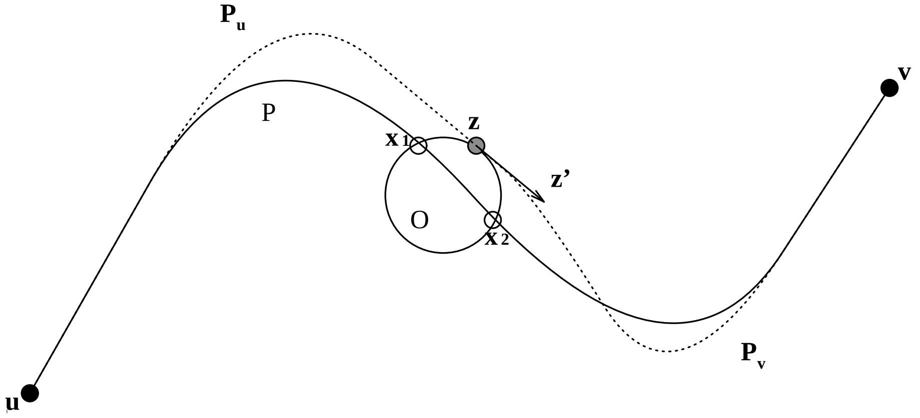


Figure 2: Collision resolution

then the intersection is empty, and we go on to the next obstacle. If not, then $\mathbf{P} \cap O_i = \{\mathbf{x}_1, \mathbf{x}_2\}$. Note that $\mathbf{x}_1$ and $\mathbf{x}_2$ are distinct, since we don't consider "brushing by" an obstacle as a collision. In this case, the curve segment is split into two parts $\mathbf{P}_u$, and $\mathbf{P}_v$. Let $\mathbf{z}$ be the midpoint of the geodesic path on the boundary of B_i between $\mathbf{x}_1$ and $\mathbf{x}_2$ (*i.e.*, the midpoint of the shortest arc on the surface of B_i connecting the two points). Let $\mathbf{z}'$ be a vector tangent to B_i at $\mathbf{z}$ whose magnitude is r_i. Several such vectors exist[3], so let $\mathbf{z}'$ be the one whose direction is "closest" to that of $\mathbf{x}_2 - \mathbf{x}_1$. Set $\mathbf{P}_u$ to be the quartic Hermite family (now A is a free parameter) connecting $\mathbf{u}$ and $\mathbf{z}$ whose final velocity is $\mathbf{z}'$, and set $\mathbf{P}_v$ to be the family connecting $\mathbf{z}$ and $\mathbf{v}$ whose initial velocity is $\mathbf{z}'$ (see figure 2). The curve consisting of these two segments now avoids obstacle O_i. Both pieces can now be processed independently to test for collisions with the remaining obstacles.

At the end of this resolution process, a path $\hat{\mathbf{P}}$ consisting of a piecewise collection of quartic Hermite curves is obtained. It avoids all the obstacles, and may now be passed back to the optimization process.

4 Conclusions and Future Work

The two pronged approach outlined above is a greedy algorithm. Problems are solved locally as they are encountered, in the hope that this will lead to a satisfactory global solution. This approach, however, requires refinement for the solution found is not guaranteed to be optimal. It is not difficult to cook up a scenario in which the solution to the global problem is fairly simple, yet the collection of local ones produces a complex path. Nevertheless, this method is not unlike a real-life situation in which a cyclist charts his way through unknown territory. If he is not previously afforded with a map or *bird's eye* view, he will wind his way around obstacles in the order by which they are encountered, and may completely miss a simple, roundabout path if one exists.

A number of issues remain for future work. As mentioned above, the minimum found

[3]Actually, there are either two or infinitely many of them, depending on whether the problem is planar (*i.e.*, in $\mathbf{R}^2$) or spatial (in R^3).

by the gradient descent method is not known to be global. If indeed it is not, the search method must be modified to accommodate local ascents in the hope that they will lead to lower minima, in a fashion similar to simulated annealing. Another major issue requiring investigation is that of collision resolution. Admittedly, the selection of a tangent vector on the geodesic path, that is equal in length to the obstacle's radius, was completely arbitrary. No evidence has yet been shown to support or to contrast to that choice.

References

[1] Farin, Gerald, *Curves and Surfaces for CAGD*, 3rd ed., Academic Press, London, 1993.

[2] Fiume, Eugene. "Isometric Piecewise Polynomial Curves", to appear *Computer Graphics Forum*. Eurographics, 1994.

[3] Lee, E T Y. "Choosing Nodes in Parametric Curve Interpolation", *Computer Aided Design*, 21(6), 1989.

[4] Watt, Alan and Watt, Mark. *Advanced Animation and Rendering Techniques: Theory and Practice*, ACM Press, 1992.

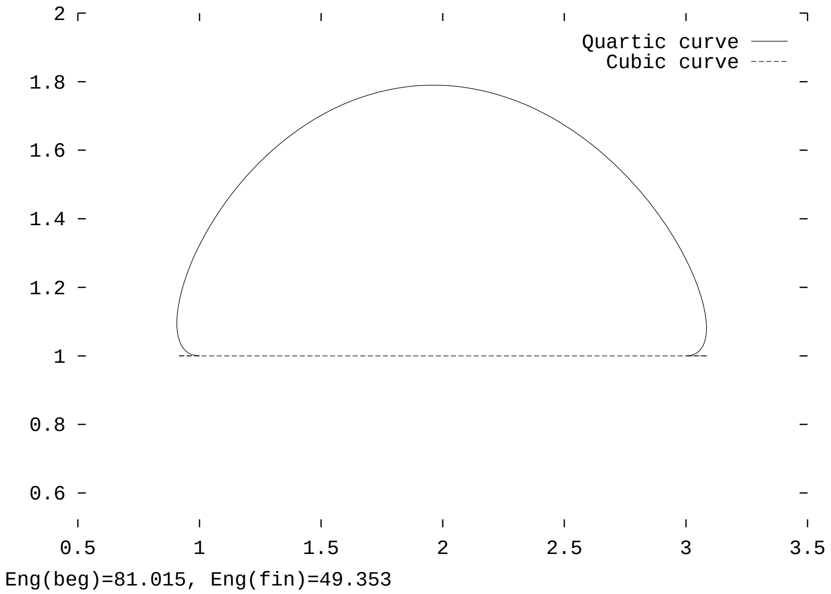


Figure 3: A curve that backs up on itself

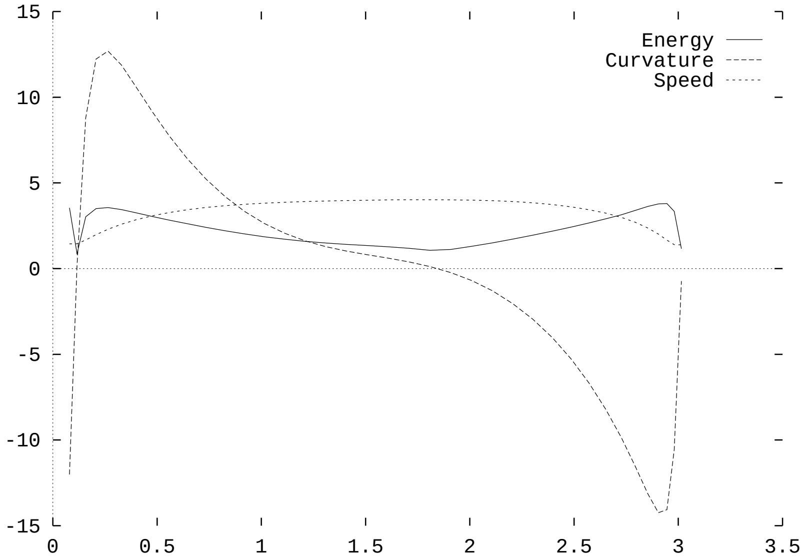
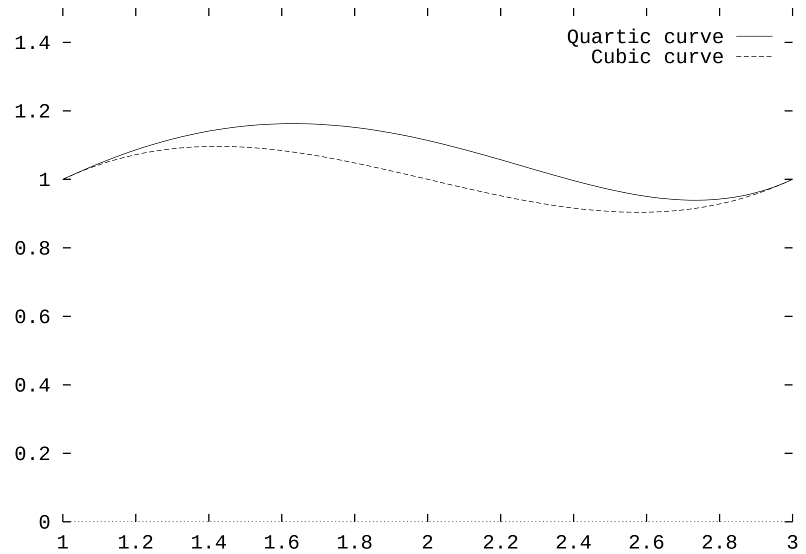


Figure 4: Speed, curvature, and energy consumption of a backed up curve



Eng(beg)=6.591, Eng(fin)=6.188

Figure 5: A clamped s-curve

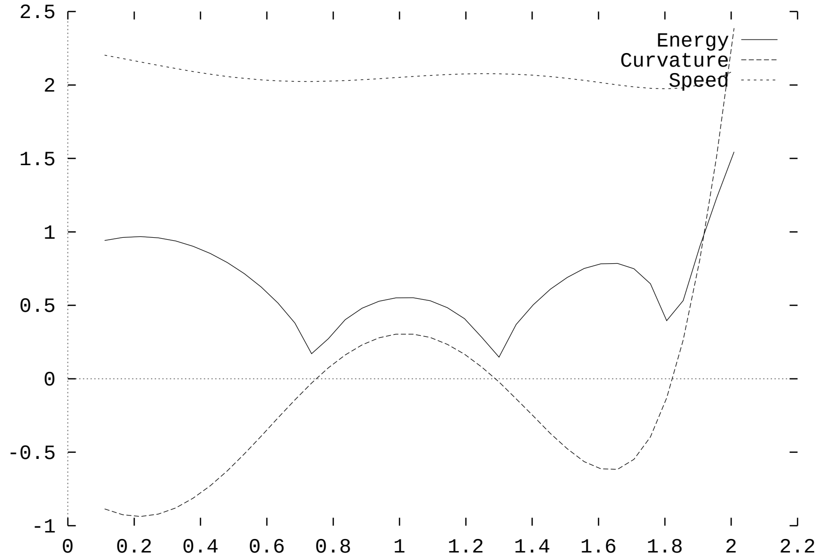


Figure 6: Speed, curvature, and energy consumption of a clamped s-curve

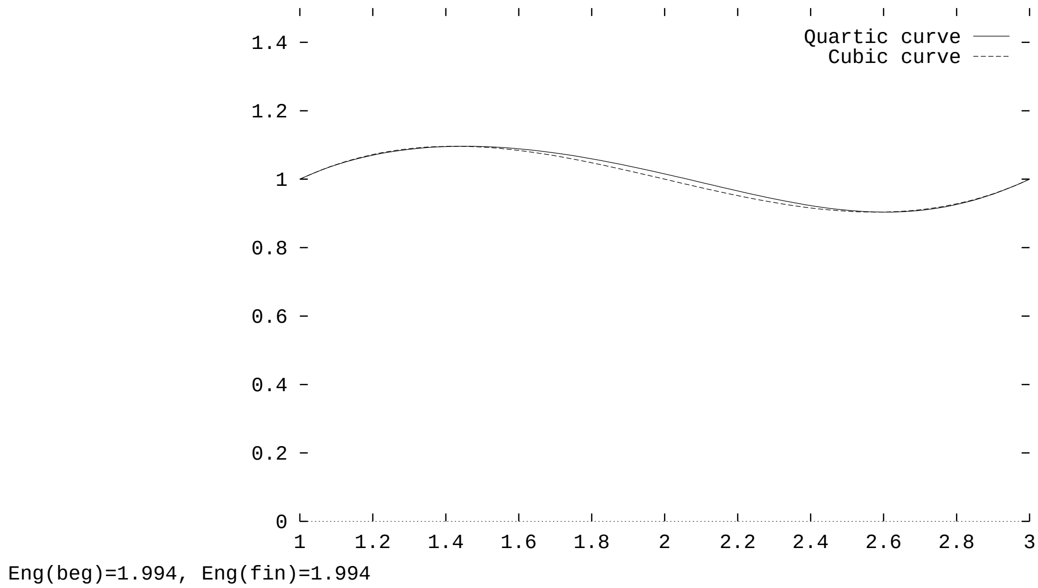


Figure 7: An s-curve with clamped speed, and unclamped curvature

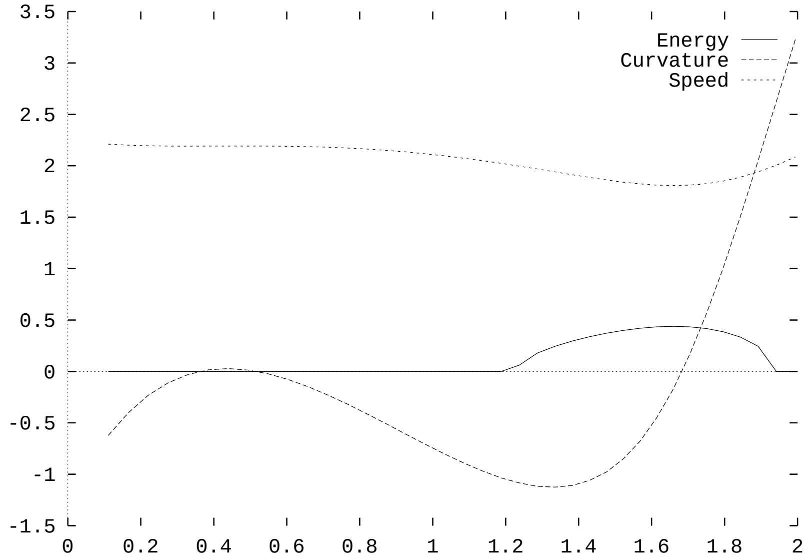


Figure 8: Speed, curvature, and energy consumption of an unclamped s-curve

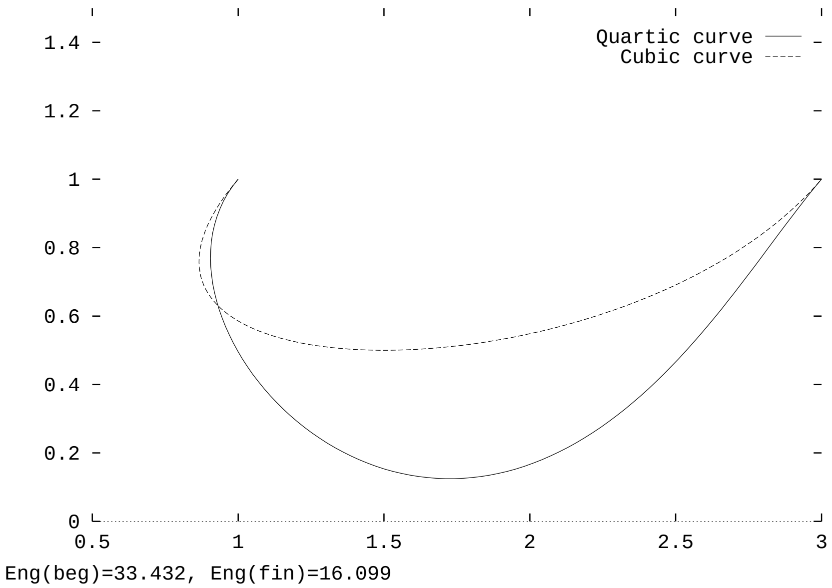


Figure 9: A curve with clamped speed and curvature

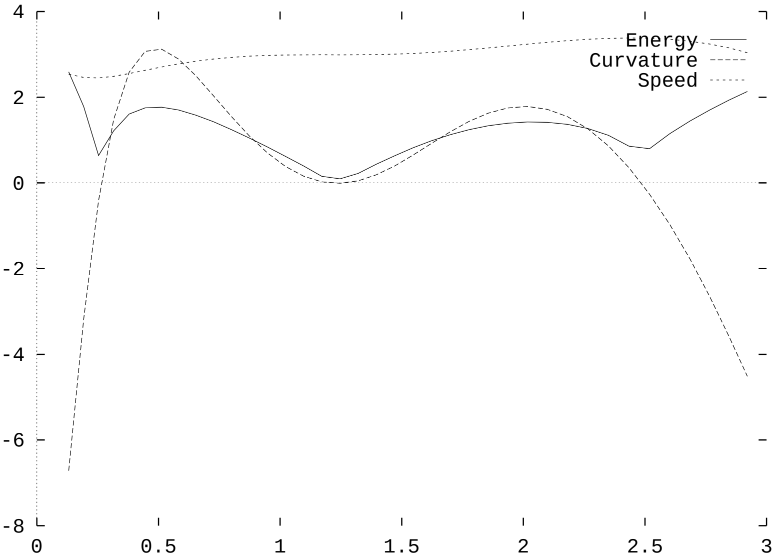


Figure 10: Speed, curvature, and energy consumption of a standard clamped curve

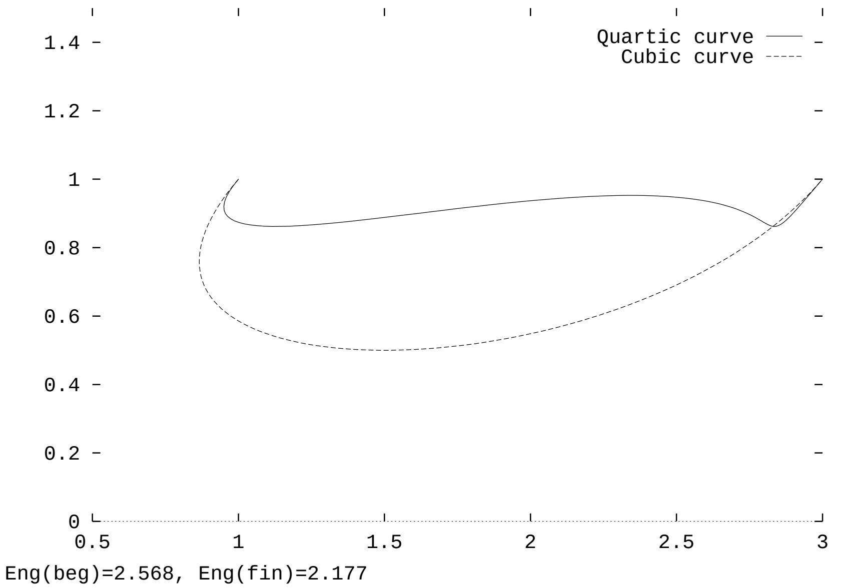


Figure 11: A curve with unclamped speed and curvature

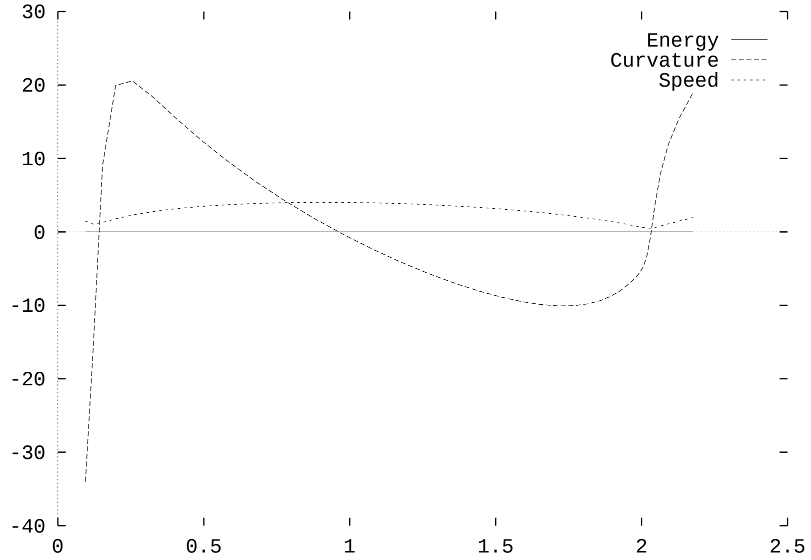


Figure 12: Speed, curvature, and energy consumption of a standard unclamped curve