Drawing Piecewise Smooth Curves

Ross Ihaka

The Big Picture

- Given an arbitrary function f and an interval $[a,b]$, I want to have an automatic procedure that will draw the graph of the function over the interval.
- Using the procedure should be as easy as this.

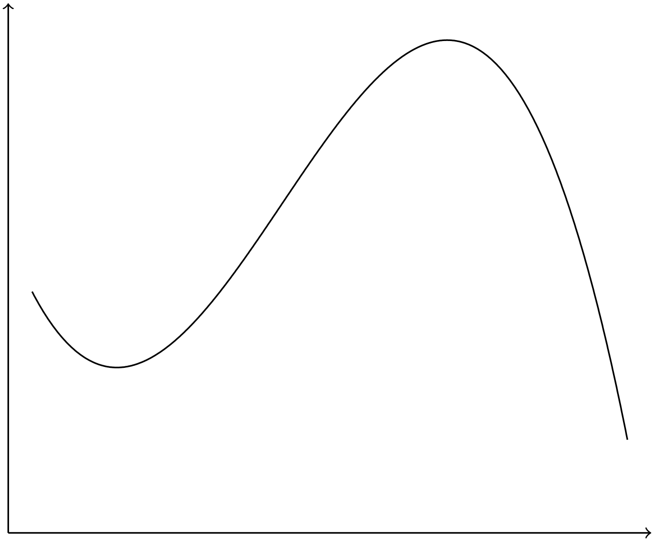
```
plot(f, a, b)
```

- At this level of generality the problem is "hard" but, for more restricted classes of function, progress can be made.
- This talk will mostly consider smooth (i.e. differentiable) functions but will also mention piecewise-smooth functions.

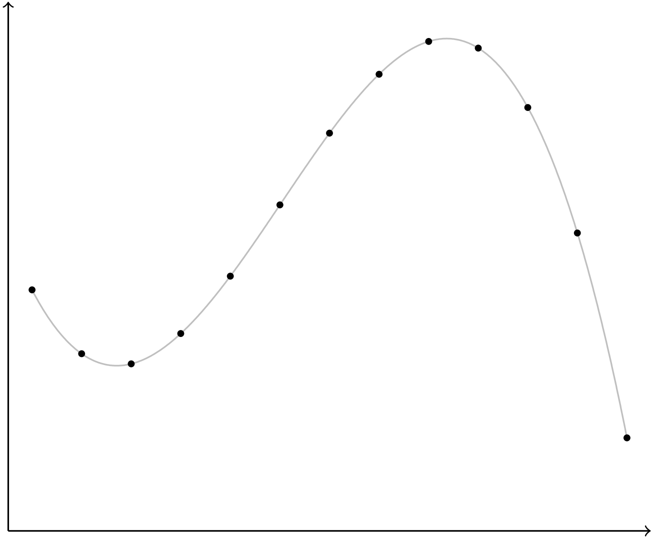
Curves and Polylines

- In computer graphics, a *polyline* is a connected set of line segments.
- A standard method for drawing parametric curves in a vector drawing system uses a polyline approximation based on (many) equally-spaced parameter values.
- Drawing the graph of a function $f(x)$ over an interval $[a,b]$ is an important special case.
- This is the method used in the R function `curve`, which uses 101 equally-spaced points by default.

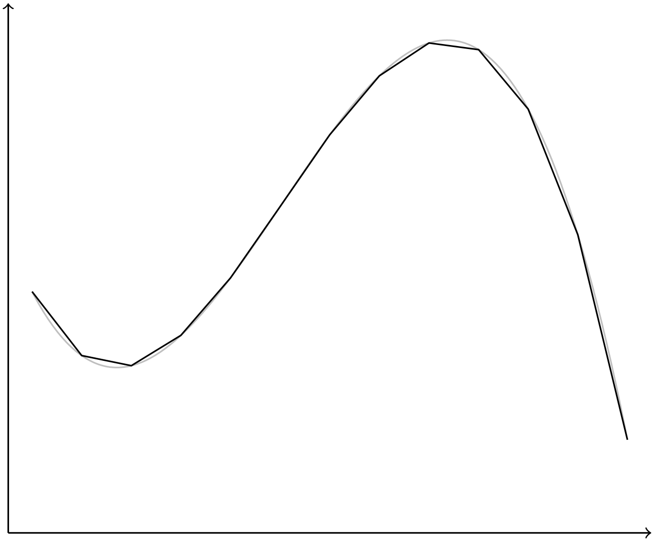
A Smooth Function



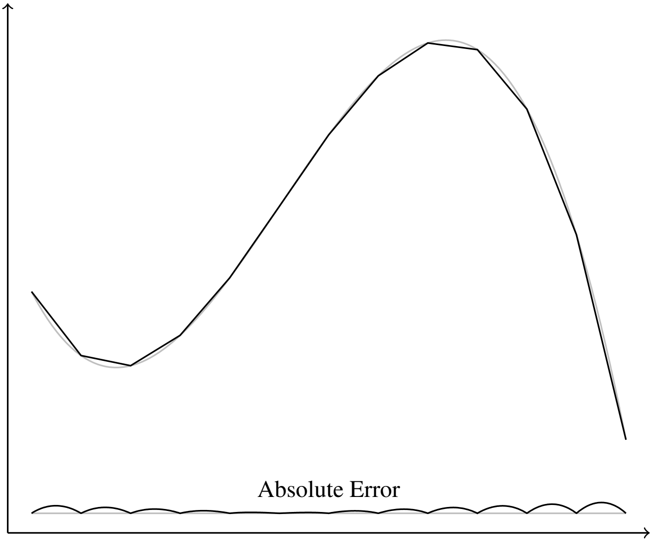
A Smooth Function



A Smooth Function



A Smooth Function

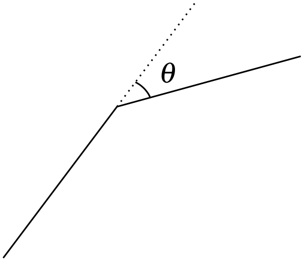


What Makes a Polyline Approximation Good?

- Even when the accuracy of the polyline approximation is good, it may not look “right” because it is clearly not a smooth curve.
- This is because the visual system detects the “corners” at the joins of the segments making up the polyline.
- This is very much like the edge-detection process built in to the visual system.
- The real issue here is one of *visual perception* rather than mathematical accuracy.

Visual Detection of Polyline Joins

- As with many perceptual phenomena, there appears to be a threshold effect for detection of a join between polylines.
- When the change direction, θ, of the polylines is "small" (e.g. $< 3°$), no join is detected and the transition from one segment to the next is seen as "smooth."



Drawing Visually Smooth Polylines

- To draw a polyline approximation to a curve that is visually smooth:
  - Choose points along the curve so that the change of angle from segment to segment is less than the "visually smooth" threshold.
  - Draw the polyline that joins the points.

Mathematical “Formulation”

- Given a smooth function $f : [a, b] \to \mathbb{R}$, let $\phi(x)$ be the angle that the tangent to $f(x)$ at x makes with the x-axis.
- We want to choose $a = x_0 < x_1 < x_2 < \cdots < x_n = b$ so that the change of angle from x_i to x_{i+1} is small.
- I.e., for some suitably small δ the x_i should satisfy

  $$\left|\phi(x_{i+1}) - \phi(x_i)\right| < \delta$$

  for $i = 1, \ldots, n$.
- How can the x_i values be chosen to make this happen?

A Solution

The function

$$h(x) = \int_a^x \left|\phi'(t)\right| dt.$$

is a monotonically non-decreasing function of x and so has a well-defined inverse. Further, if $u < v$,

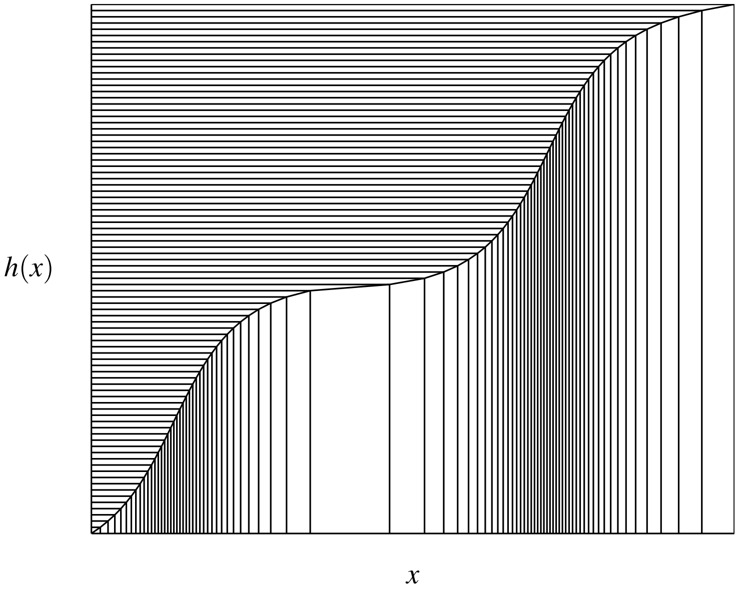
$$|\phi(v) - \phi(u)| = \left| \int_u^v \phi'(t)\, dt \right| \leq \int_u^v \left|\phi'(t)\right| dt = h(v) - h(u)$$

Choose points $h(a) = y_0 < y_1 < y_2 < \cdots < y_n = h(b)$ such that $y_{i+1} - y_i < \delta$ for all i. Set $x_i = h^{-1}(y_i)$ then the points

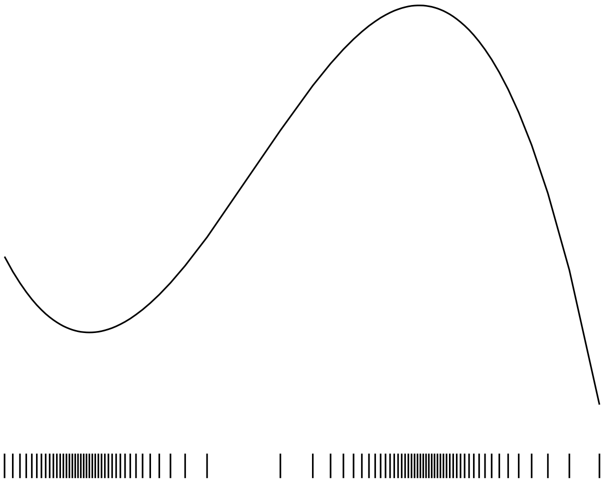
$$\big(x_0, f(x_0)\big), \big(x_1, f(x_2)\big), \ldots, \big(x_n, f(x_n)\big)$$

define the approximating polyline.

Choice of Plotting Ordinates



A Smooth Polyline Approximation (86 Points)



Computer Implementation

- To implement the method in a computer the continuous problem is approximated by its discrete version using a very finely spaced grid of points splitting up the interval from a to b.
  - Derivatives are replaced by differences.
  - Indefinite integrals are replaced by cumulative sums.
- Linear interpolation can be used to compute $h(x)$ and $h^{-1}(u)$.

An R Implementation

```
smoothx =
    function(f, a, b, delta = 2.5, n = 1001) {
        dr = delta * pi / 180
        x = seq(a, b, length = n)
        y = f(x)
        <Compute scalings ax and ay>
        phi = atan2(ay * diff(y), ax * diff(x))
        d.phi = c(0, diff(phi), 0)
        u = cumsum(abs(d.phi) + 1e-3/n)
        if (u[n] - u[1] < dr)
            x[c(1, length(x))]
        else
            approx(u, x,
                   n = ceiling((u[n] - u[1])/dr) + 1)$y
    }
```

Piecewise Continuous Functions

- There are many (useful/interesting) functions that not continuous on their domain or which lie partly outside the plotting window of interest.
- Extending the drawing method outlined previously requires decomposing the domain of a function into intervals where the smooth polyline approximation can be used.
- This means dealing with:
  - Regions where the function is out of the viewing window (clipping).
  - Points where the function is not defined.
  - Simple function discontinuities.
  - Asymptotes.

Houston, We Have a Problem

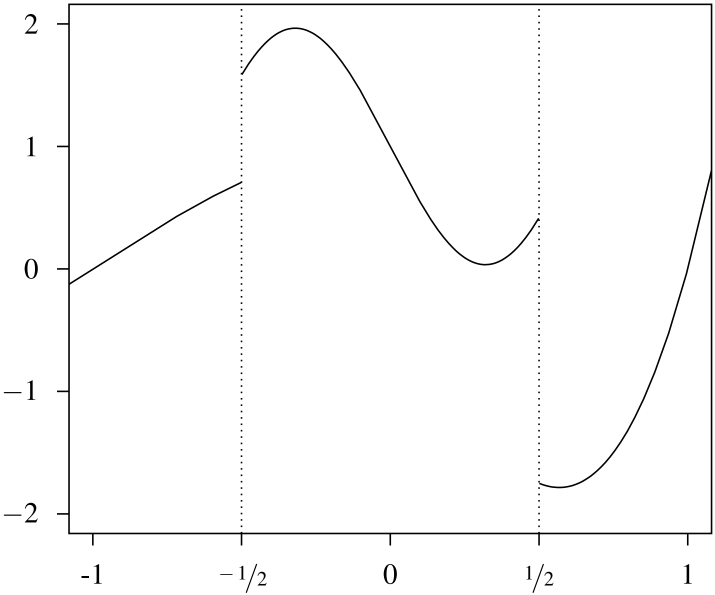
Patent No.: US 6,704,013 B2
Michael E. Hosea, 2004 (Texas Instruments)
FUNCTION GRAPHING METHOD WITH DETECTION OF DISCONTINUITIES

Patent No.: US 7,595,801 B1
Barry M. Cherkas, 2009
COMPLETE FUNCTION GRAPHING SYSTEM AND METHOD

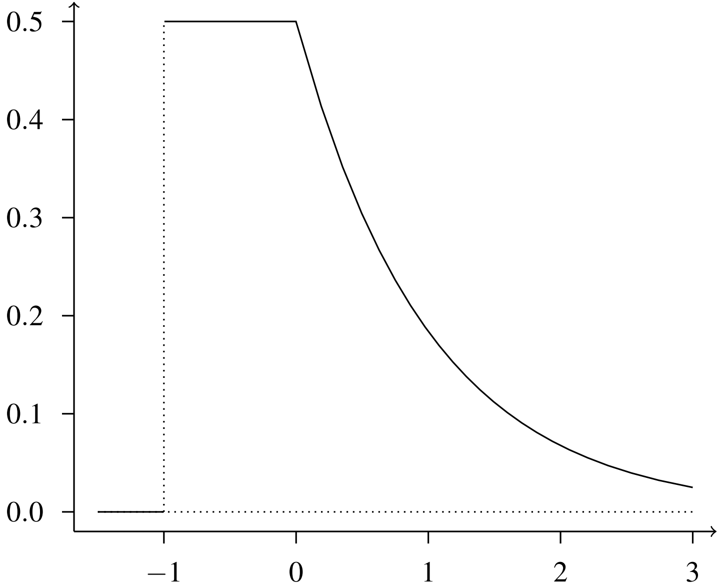
Patent No.: US 7,920,142 B2
Luke Kelly and Jinsong Yu, 2009 (Microsoft)
IDENTIFYING ASYMPTOTES IN APPROXIMATED CURVES AND SURFACES.

These (software) patents make work in this area problematic.

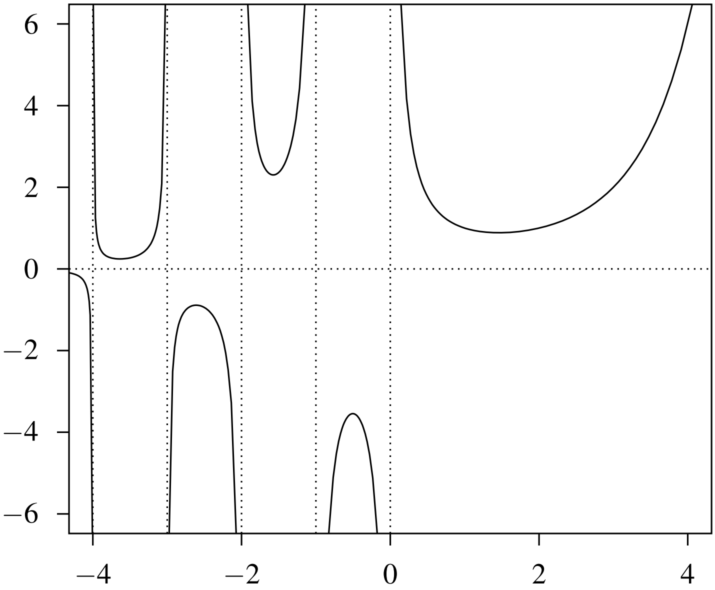
Example: A Discontinuous Function



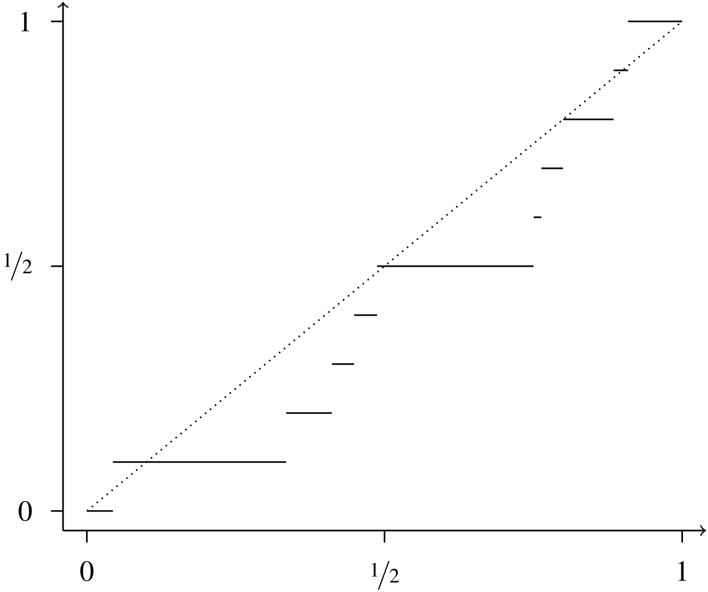
Example: A Density From Ash's *Basic Probability Theory*



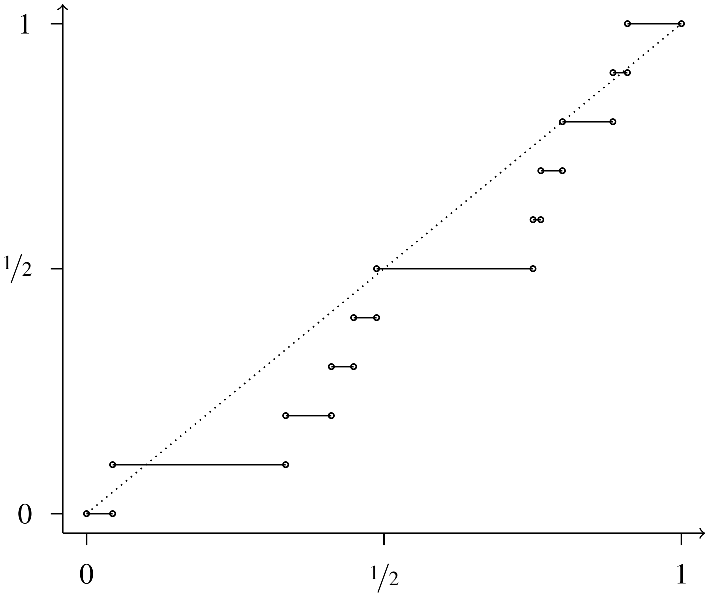
Example: The Gamma Function (A Torture Test)



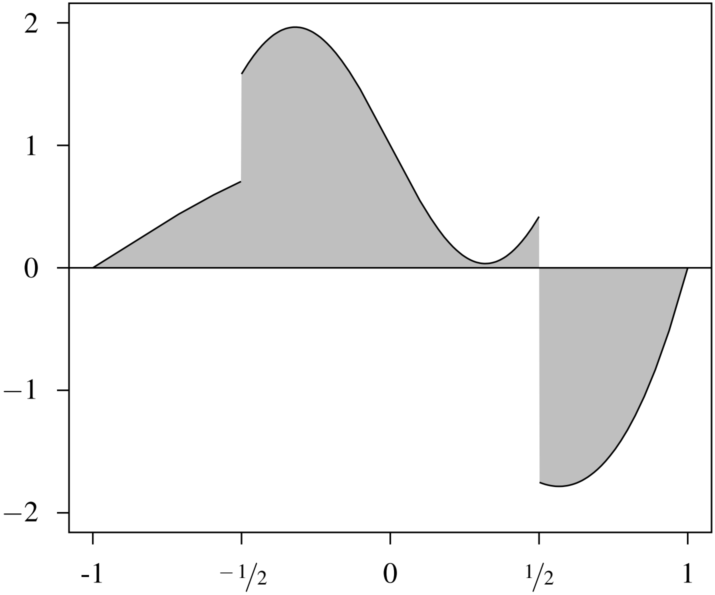
Example: An Empirical CDF



Example: An Empirical CDF



Curves and Area Fills



Summary

- A novel method of curve drawing based on visual smoothness has been presented.
- The method can be incorporated into a general system for automatically drawing piecewise smooth functions.
- United States software patents make the delivery of this technology problematic. (Oppose the TPP!)