Chebyshev Picard Iteration

Fixed Point IVP Integration

A.C.

September 22, 2024

Outline

Starting Point

Consider:

$$\cos n\theta$$

How does this relate to $\cos\theta$?

$n + 1$

$$\cos 0 = 1$$
$$\cos\theta = \cos\theta$$
$$\cos 2\theta = 2\cos^2\theta - 1$$
$$\vdots$$
$$\cos(n+1)\theta = 2\cos\theta\cos n\theta - \cos(n-1)\theta$$

See notes for proof.

Recurrence

Let $T_n(\cos\theta) := \cos n\theta$ and $x := \cos\theta$ and observe the following recurrence

$$T_0 = 1$$
$$T_1 = x$$
$$T_{n+1} = 2xT_n - T_{n-1}$$

In closed form: $T_n(x) = \cos n \cos^{-1} x$

Polynomial

$$T_{n+1} = 2xT_n - T_{n-1}$$

$2x$ multiplier increases degree of each subsequent T_n by exactly one. Easy to see that $\{T_0, T_1, \ldots, T_n\}$ spans forms a basis set for $\mathbb{P}_n$.

More directly, we can find coefficients $c_0, c_1, \ldots, c_n$ for any polynomial p_n s.t.

$$p_n = c_0 T_0 + c_1 T_1 + \cdots + c_n T_n$$

Domain

The recurrence form of Chebyshev polynomials (of the first kind) applies across the whole real line.

The cosine form requires we restrict ourselves to the interval $[-1, 1]$.

We will take it as a given that we are in the range $[-1, 1]$ throughout.

Integral

$$\int T_k = \frac{1}{k+1} T_{k+1} - \frac{1}{k-1} T_{k-1} + C$$

For $k = 0, k = 1$, there are special cases:

$$\int T_0 = T_1$$

$$\int T_1 = \frac{1}{4}(T_0 + T_2)$$

See notes for proof.

Key Result

If we have a polynomial in Chebyshev-space, then we can integrate it easily without changing basis (e.g. to the "usual" basis $1, x, x^2, \ldots$, where integration is even simpler).

Lagrange Interpolation

Recall: a polynomial of degree n is uniquely determined by $n+1$ points $(x_0, y_0), \ldots, (x_n, y_n)$.

Langrange interpolation is a standard method for constructing such a polynomial:

$$p_n(x) = \sum_i y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

What if we would like the polynomial in terms of the Chebyshev basis?

System of Equations

Given points $(x_0, y_0), \ldots, (x_n, y_n)$, can write and solve a system of equations

$$\begin{aligned} c_0 T_0(x_0) + c_1 T_1(x_0) + \cdots + c_n T_n(x_0) &= y_0 \\ c_0 T_0(x_1) + c_1 T_1(x_1) + \cdots + c_n T_n(x_1) &= y_1 \\ &\vdots \\ c_0 T_0(x_n) + c_1 T_1(x_n) + \cdots + c_n T_n(x_n) &= y_n \end{aligned}$$

Fairly heavyweight computation, would prefer a cheaper solution.

Orthogonality

Define

$$\langle f, g \rangle = \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} fg \, dx$$

It can be shown that

$$\langle T_j, T_k \rangle = \begin{cases} 0, & \text{if } j \neq k \\ \pi, & \text{if } j = k = 0 \\ \frac{\pi}{2}, & \text{if } j = k > 0 \end{cases}$$

That is, Chebyshev polynomials form an *orthogonal* basis over $\mathbb{P}_n$. See notes for proof.

Inner Product Interpolation

Consider the equation

$$p_n = c_0 T_0 + c_1 T_1 + \cdots + c_n T_n$$

Take the inner product against T_k on each side to see

$$\langle p_n, T_k \rangle = c_k \langle T_k, T_k \rangle$$

If we can compute $\langle p_n, T_k \rangle$ efficiently, then we have a fast procedure for acquiring the polynomial in Chebyshev form.

Quadrature

It is far from obvious but Lobatto quadrature at Chebyshev nodes $x_j = \cos\frac{j\pi}{n}$ computes $\langle p_n, T_k \rangle$ exactly, giving us the formula

$$\langle p_n, T_k \rangle = \frac{\pi}{n} \sum_j{}'' \, p_n\left(\cos\frac{j\pi}{n}\right) T_k\left(\cos\frac{j\pi}{n}\right)$$

See notes for proof.

Key Result

If we choose to interpolate a function f at $n+1$ Chebyshev nodes, then the Chebyshev form of the resulting polynomial may be efficiently computed through simple sums.

Weierstrass Approximation Theorem

Given any smooth function f on a compact interval $[a, b]$ and an arbitrary error tolerance ϵ, there exists a polynomial p s.t.

$$||p - f||_\infty \leq \epsilon$$

That is, we may approximate any smooth function arbitrarily well on a finite interval with *some* polynomial.

It seems a fair guess that interpolating f at a large number of points should yield such a polynomial.

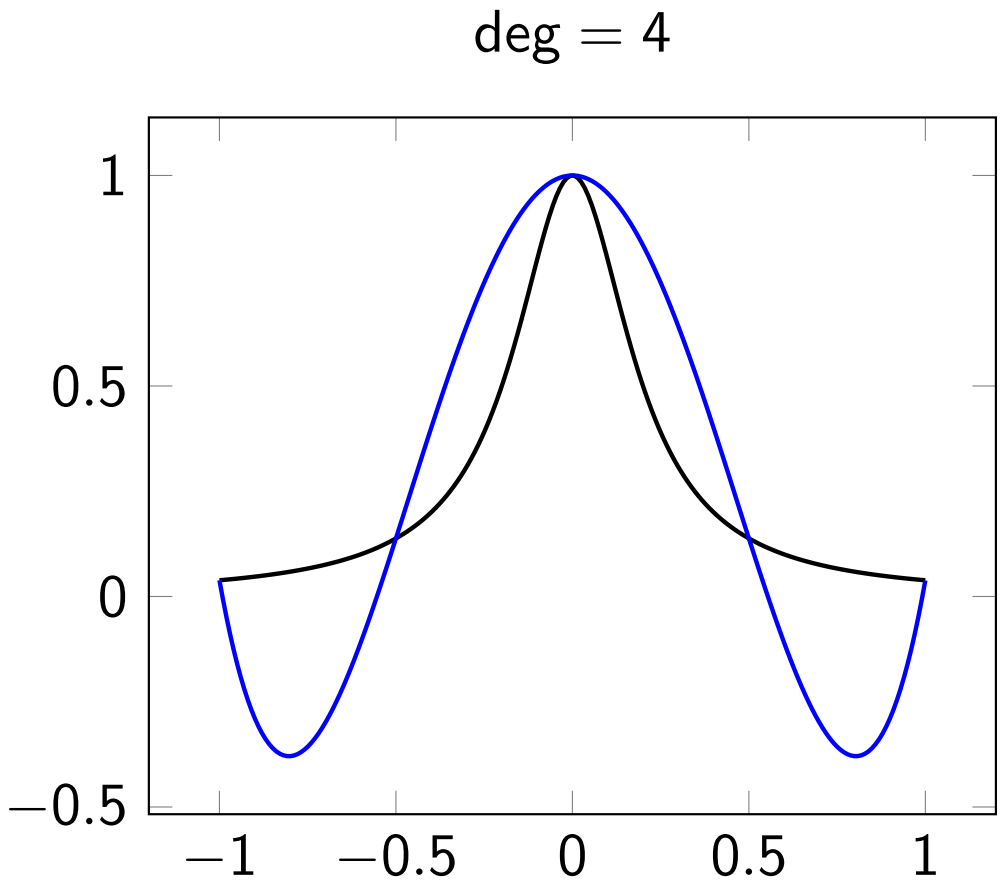
Runge's Phenomenon

Let us consider the behavior of polynomial interpolants on the following function

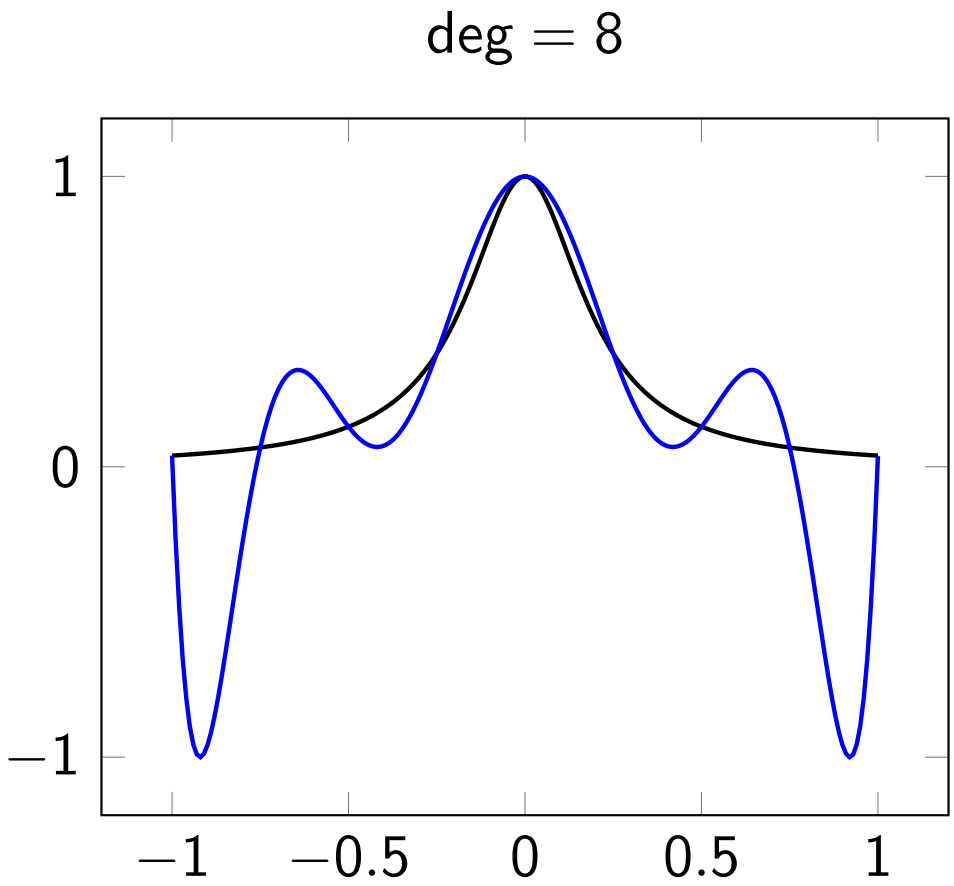
$$y = \frac{1}{1 + 25x^2}$$

We'll do the obvious thing, and select equally spaced points for evaluation.

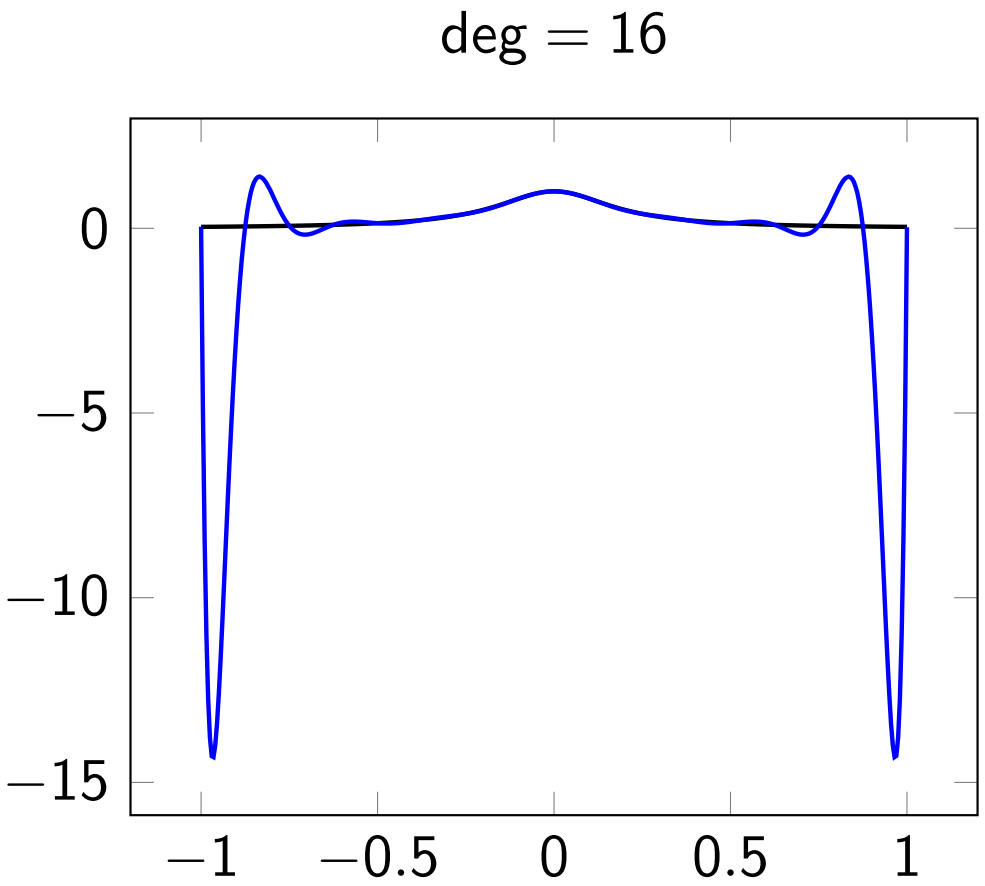
Runge's Phenomenon



Runge's Phenomenon



Runge's Phenomenon



Lagrange Error Formula

Let f be a smooth function on a compact interval, and p_n be the polynomial interpolating it at points $x_0, x_1, \ldots, x_n$. Then

$$f(x) = p_n(x) + \frac{f^{(n+1)}(\eta_x)}{(n+1)!} \prod (x - x_j)$$

See notes for proof.

Controlling Approximation Error

Our error term has two components – a derivative part, and a product part.

It is difficult to fathom a mechanism to control the former, but the latter should be directly impacted by our choice of interpolating points.

Chebyshev Node Error

Set $x_0, \ldots, x_n$ to the zeros of the n-th Chebyshev polynomial. We observe the following bounds on the "product" component of the Lagrange error, which no other choice of nodes can beat.

$$\max_{[-1,1]} \prod (x - x_j) = \frac{1}{2^n}$$

The extrema of the n-th Chebyshev polynomial are actually of more practical interest to us, but the result for them is the same, up to a constant factor. See notes for proof.

Key Result

Interpolating through Chebyshev nodes minimizes a key term underpinning L_∞ error.

We have not shown that Chebyshev nodes are always best (in fact, there exists no choice of nodes with is best for all polynomial-approximable functions), but it is very good, very often.

Outline

Fixed Point Iteration

Consider an initial value problem of the form

$$\dot{x} = f(x, t); x(t_0) = x_0$$

We will guess an initial solution X_0 s.t. $X_0(t_0) = x_0$, then iterate as

$$X_k(t) = x_0 + \int_{t_0}^{t} f(X_{k-1}, t)\, dt$$

Convergence Intuition

$$X_k(t) = x_0 + \int_{t_0}^{t} f(X_{k-1}, t)\, dt$$

- Obviously, if we find a solution, the FPI will stop moving
- If we are not yet converged, then we must be converged at least to some time $t_{ok} \geq t_0$
- On the next iteration, will be OK to $t_{ok} + \epsilon$
- Compare to Euler's method:
  - Euler computes further in time at each iteration, taking tiny steps.
  - X_k becomes more accurate for (at least slightly) longer at each iteration.

More Formally: Picard-Lindelöf

If f is bounded and Lipschitz over the period of integration, and our initial guess is as well, then there exists a solution to our IVP and our fixed point procedure converges to it.

See notes for proof.

More Complicated Problems

We have concerned ourselves with a procedure for solving first order IVPs.

We will not address it here, but a similar procedure can be applied to higher order problems by integrating multiple times at each step of the fixed point iteration.

It is similarly out of scope, but Picard Iteration has been adapted successfully to BVPs.

Outline

Core Insight

1. Picard Iteration is only usable as a practical tool if we can integrate our candidate solutions symbolically.
2. Chebyshev Polynomials are highly versatile approximators which are convenient to integrate symbolically.
   2.1 Only useful on $[-1, 1]$, but easily resolved by affine transformation of the time domain.

Procedure (IVP)

procedure ChebyshevPicardIterate(f, x_0, tol)
    $X_0 \leftarrow$ CreateAnsatz(x_0)
    for $i \in [1, \infty)$ **do**
        $P_i \leftarrow$ ChebyshevInterpolate($f(X_{i-1})$)
        $X_i \leftarrow$ Integrate(P_i)
        if $||X_i - X_{i-1}|| <$ tol **then**
            return

Refinement

Chebyshev-Picard Iteration is a fixed point method, which allows it to terminate quickly if a good initial guess can be provided to it.

This is of particular interest when dealing with perturbation-type problems, wherein a reasonably good initial estimate may be obtained analytically, but which requires additional refinement to be practically useful.

Parallelism

Most traditional methods for evolving differential equations struggle to make good use of parallel hardware. Step solvers, for instance, perform comparatively little computation.

Each iteration of Chebyshev-Picard depends on the last, but within an iteration there is a substantial volume of independent computation to be done, making it easier to leverage parallel hardware. In particular, f must be evaluated at each Chebyshev node each iteration, and each evaluation can be performed independently.

Mesh-Free Solution

The return value of Chebyshev-Picard is a surrogate function which approximates the IVP solution across the full interval of integration.

This is of limited practical importance – methods which return a mesh can generally have their solutions interpolated into the points between – but the way that it falls out of Chebyshev-Picard "for free" is undeniably elegant.

Further Reading

JO Cappellari, CE Vélez, and Arthur J Fuchs. *Mathematical theory of the Goddard trajectory determination system*. Vol. 71106. Goddard Space Flight Center, 1976.

Xiaoli Bai. "Modified Chebyshev-Picard iteration methods for solution of initial value and boundary value problems". PhD thesis. Texas A&M University, 2010.