

1 Chebyshev methods for non-periodic functions

We learned that smooth periodic functions and their derivatives can be well approximated by trigonometric interpolants:

$$f(x) \approx p(x) = \sum_{k=-(n-1)/2}^{(n-1)/2} \tilde{c}_k e^{ikx}$$

and there are fast algorithms for mapping function values to coefficients and vice versa and for differentiation. We saw how Fourier approximation methods could be used to compute an approximate solution to a PDE.

Here we'll learn that smooth non-periodic functions and their derivatives can be accurately and stably approximated by expansions in orthogonal polynomials (OPs),

$$f(x) \approx \sum_{k=0}^n c_k p_k(x).$$

and there are fast algorithms for computing with OPs. We'll also learn how OPs can be used to approximate solutions to PDEs.

1.1 The Runge phenomenon and its resolution via OPs

Suppose we want to approximate the function $f(x) = \frac{1}{1+25x^2}$ (known as the Runge function) on the interval $x \in [-1, 1]$. We know that a trigonometric interpolant will not converge very fast to f . Instead, we approximate f by a polynomial interpolant that interpolates f at $n+1$ equally spaced points on $[-1, 1]$. One method to find the interpolating polynomial is to use the formula for the Lagrange interpolating polynomial. Instead, here we use the following "quick and dirty" method, based on inverting the Vandermonde matrix.

Definition (Vandermonde) The *Vandermonde matrix* associated with $n+1$ distinct points $x_0, \dots, x_n \in \mathbb{R}$ is the matrix

$$V := \begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix}$$

We recall the definition of a polynomial interpolant / interpolating polynomial / interpolatory polynomial:

Definition (polynomial interpolant) Given $n+1$ distinct points $x_0, \dots, x_n \in \mathbb{R}$ and $n+1$ samples $f_0, \dots, f_n \in \mathbb{R}$, a degree n polynomial interpolant $p(x)$ satisfies

$$p(x_j) = f_j, \quad j = 0, \dots, n$$

Proposition (interpolatory polynomial uniqueness) The interpolatory polynomial is unique, and the Vandermonde matrix is invertible.

Proof Suppose p and \tilde{p} are both interpolatory polynomials. Then $p(x) - \tilde{p}(x)$ vanishes at $n+1$ distinct points x_j . By the fundamental theorem of algebra it must be zero, i.e., $p = \tilde{p}$.

For the second part, if $V\mathbf{c} = \mathbf{0}$ for $\mathbf{c} \in \mathbb{R}^{n+1}$ then for $q(x) = c_0 + c_1x + \cdots + c_nx^n$ we have

$$q(x_j) = \mathbf{e}_j^\top V \mathbf{c} = 0, \quad j = 1, \dots, n+1,$$

where $\mathbf{e}_j \in \mathbb{R}^{n+1}$ are the canonical basis unit vectors,

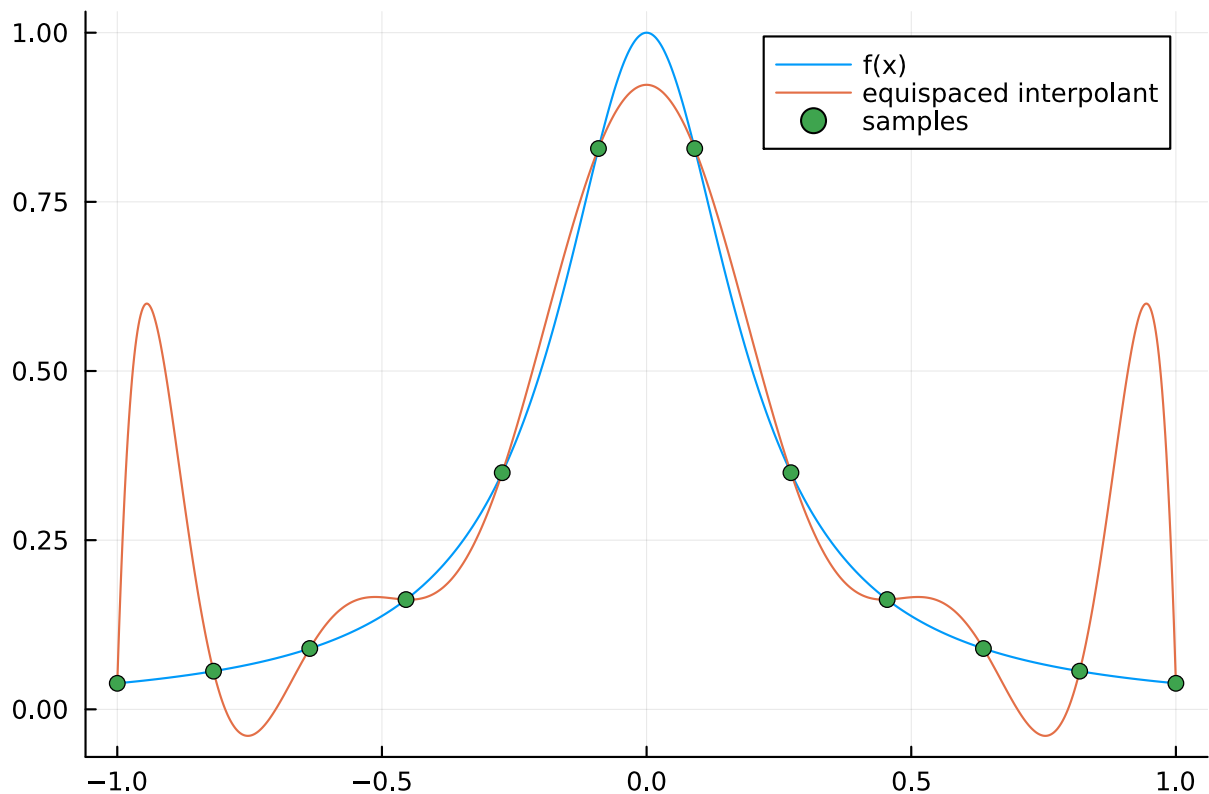
$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots$$

hence q vanishes at $n+1$ distinct points and is therefore 0, i.e., $\mathbf{c} = \mathbf{0}$. ■

using Plots, LinearAlgebra, ApproxFun, FFTW

```
f = x -> 1/(25x^2 + 1)

# Construct an interpolant through n + 1 equally spaced points on [-1, 1]
n = 11
x = range(-1,1;length=n+1) # n+1 equispaced nodes
xx = range(-1,1;length=1001) # plotting grid
V = x .^ (0:n)' # Vandermonde matrix
c = V \ f.(x) # coefficients of interpolatory polynomial
p = x -> dot(c, x .^ (0:n))
plot(xx,f.(xx);label="f(x)")
plot!(xx,p.(xx);label="equispaced interpolant")
scatter!(x,p.(x);label="samples")
```



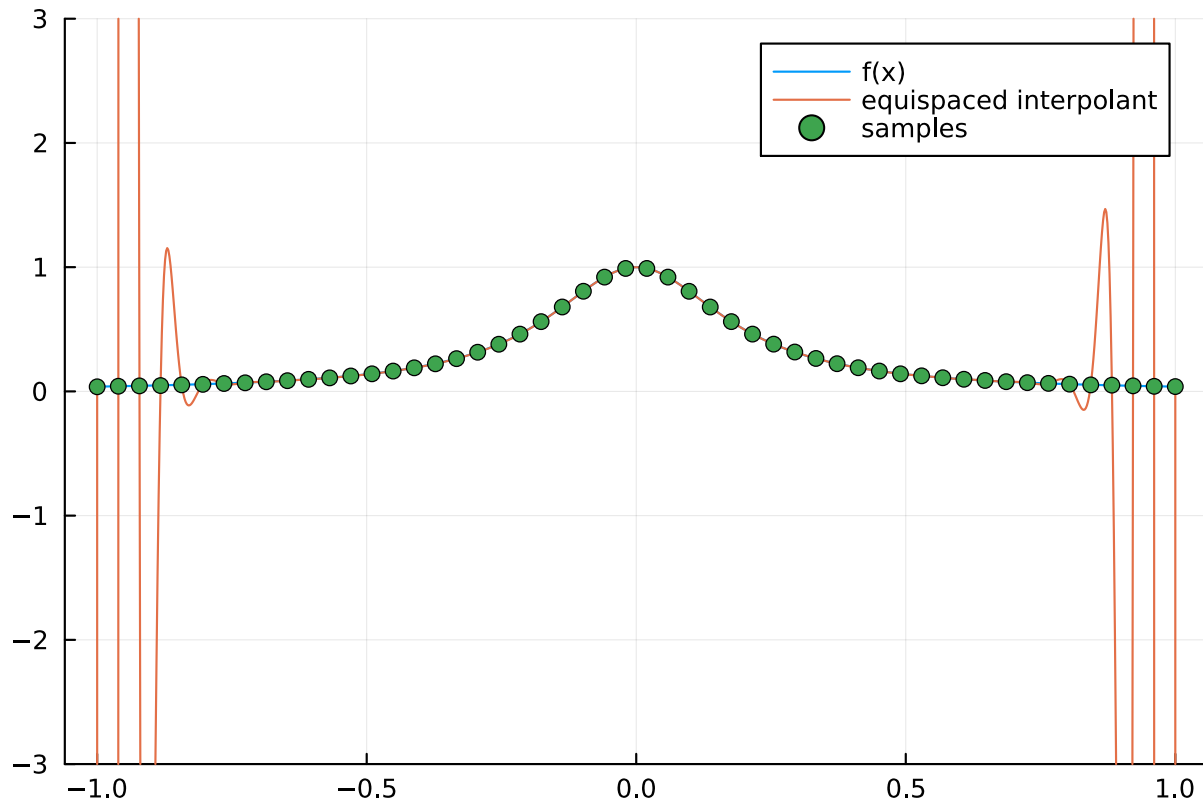
The oscillations at the ends of the interval become larger as we increase n :

$n = 51$

```

x = range(-1,1;length=n+1) # n+1 equispaced nodes
V = x .^ (0:n)' # Vandermonde matrix
c = V \ f.(x) # coefficients of interpolatory polynomial
p = x -> dot(c, x .^ (0:n))
plot(xx,f.(xx);label="f(x)",ylims=(-3,3))
plot!(xx,p.(xx);label="equispaced interpolant")
scatter!(x,p.(x);label="samples")

```



These oscillations of the interpolant through equally spaced nodes is known as the *Runge phenomenon* (not to be confused with the Gibbs phenomenon). Equally spaced interpolation nodes is clearly not a good choice for high-degree polynomial interpolation. Maybe the oscillations at the endpoints can be suppressed if we choose nodes that are clustered at the ends of the interval?

```

# Construct an interpolant using ApproxFun
n = 11
S = Chebyshev()
p = Fun(f,S,n+1) # an interpolant through the unevenly spaced nodes x
x = points(S,n+1) # n+1 unevenly spaced points
plot(xx,f.(xx);label="f(x)")
plot!(p;label="interpolant")
scatter!(xn,p.(xn);label="samples")

```

Error: UndefVarError: xn not defined

In the above figure, we interpolated $f(x)$ at the roots of an orthogonal polynomial (OP), which we'll learn about later. The interpolants at the roots of the OP converge exponentially fast to f on $[-1, 1]$ as n increases.

```

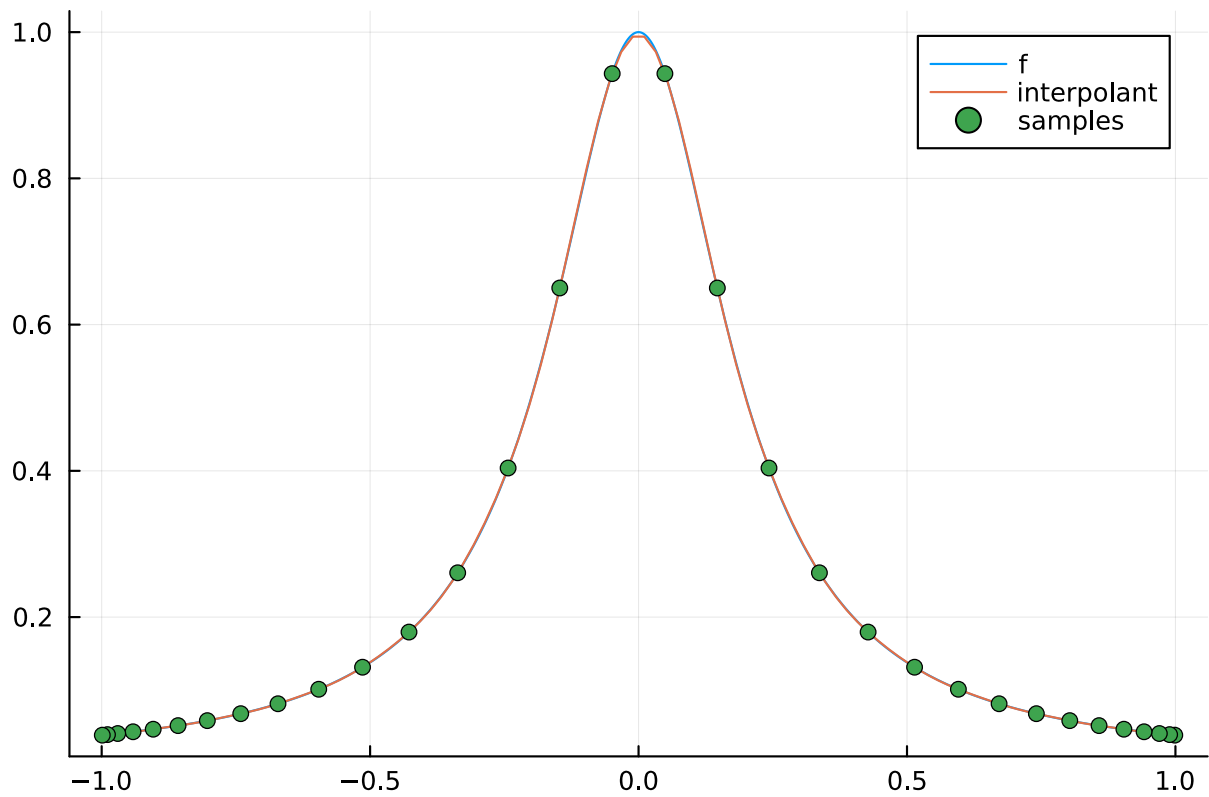
n = 31
S = Chebyshev()
p = Fun(f,S,n+1)

```

```

x = points(S,n+1)
plot(xx,f.(xx);label="f")
plot!(p;label="interpolant")
scatter!(x,p.(x);label="samples")

```



Remark (advanced) A deep and quantitative understanding of the reasons why equispaced interpolation failed for the above function and why interpolants through clustered points converged fast is a beautiful and advanced topic that requires tools from complex analysis and potential theory (See *Approximation Theory and Approximation Practice* by L.N. Trefethen).