

NUMERICAL METHODS-LECTURE VIII: INTERPOLATION

(See Judd Chapter 6)

Trevor Gallen

WARNING

- ▶ Over last few years I've noticed students don't use interpolation I teach here
- ▶ Easier to use built-in Matlab
- ▶ I'll run you through it because it is good to know the vulnerabilities and how we can avoid them
- ▶ But I won't dwell on equations

MOTIVATION

- ▶ Most solutions are functions
- ▶ Many functions are (potentially) high-dimensional
- ▶ Want a way to simplify
- ▶ A cloud of points and connecting the dots is one way
- ▶ How should we connect the dots (and choose where they are?)

THE MOST BASIC

- ▶ The basics:
 - ▶ Evenly-spaced grid, Connect the dots
 - ▶ Linearly or quadratically summarize the line

POLYNOMIAL BASIS

- ▶ We can summarize functions with different bases
- ▶ The most common is the polynomial basis:

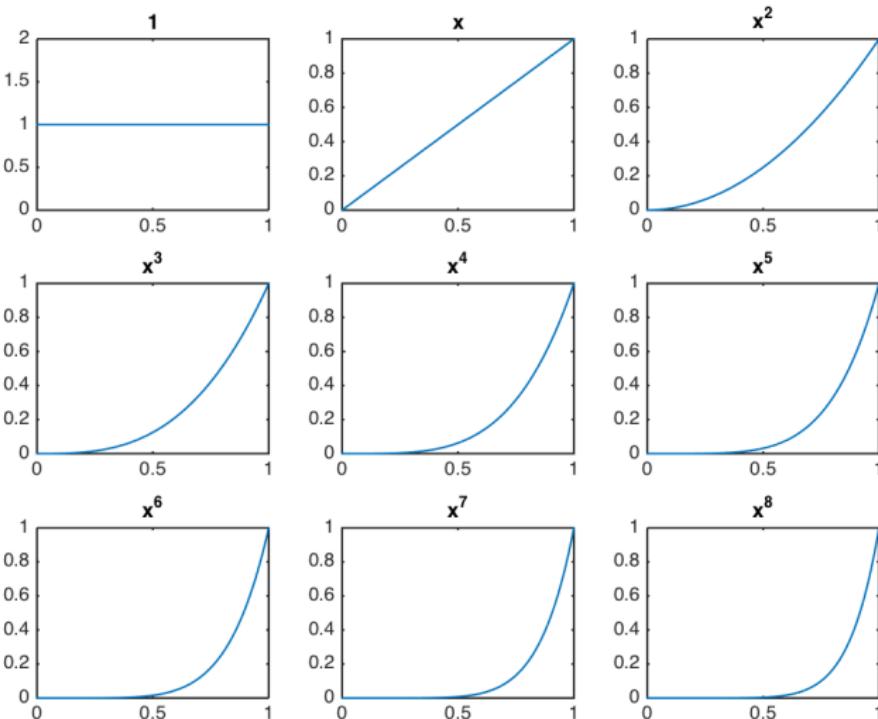
$$f(x) = [\ a_0 \ \ a_1 \ \ a_2 \ \ \cdots \ \ a_n \] \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^n \end{bmatrix}$$

- ▶ This is really just the taylor polynomial:

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

- ▶ But the polynomials are really colinear
- ▶ This is pretty inefficient coding

MONOMIAL BASIS



EXAMPLE: “REGRESSION”

- ▶ Have a cloud of n points (data is Y), location in grid summarized by X
- ▶ Regress cloud on polynomial basis of dimension i , $i < n$:

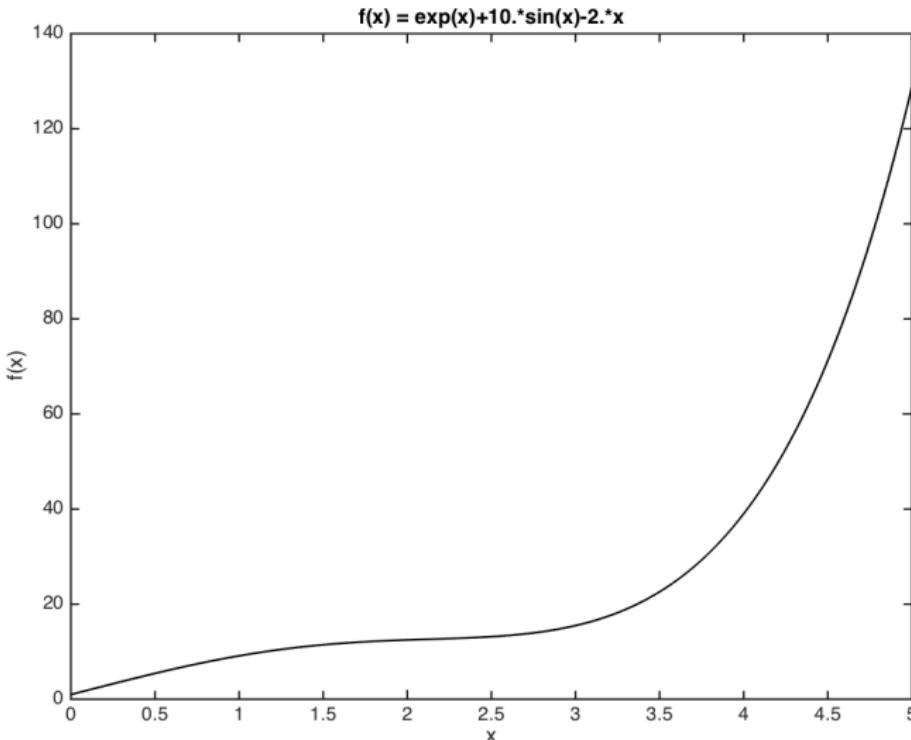
$$\hat{\beta} = (X' * X)^{-1} X' * Y$$

- ▶ Then your interpolation is always just:

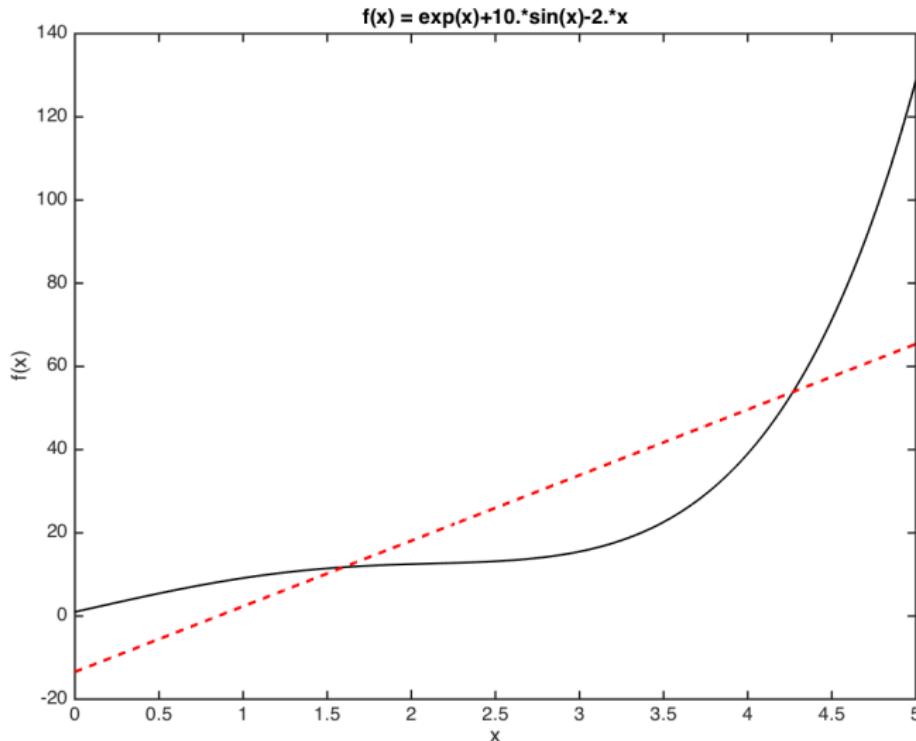
$$\hat{Y} = \hat{\beta}X$$

- ▶ This is okay for a lot of problems

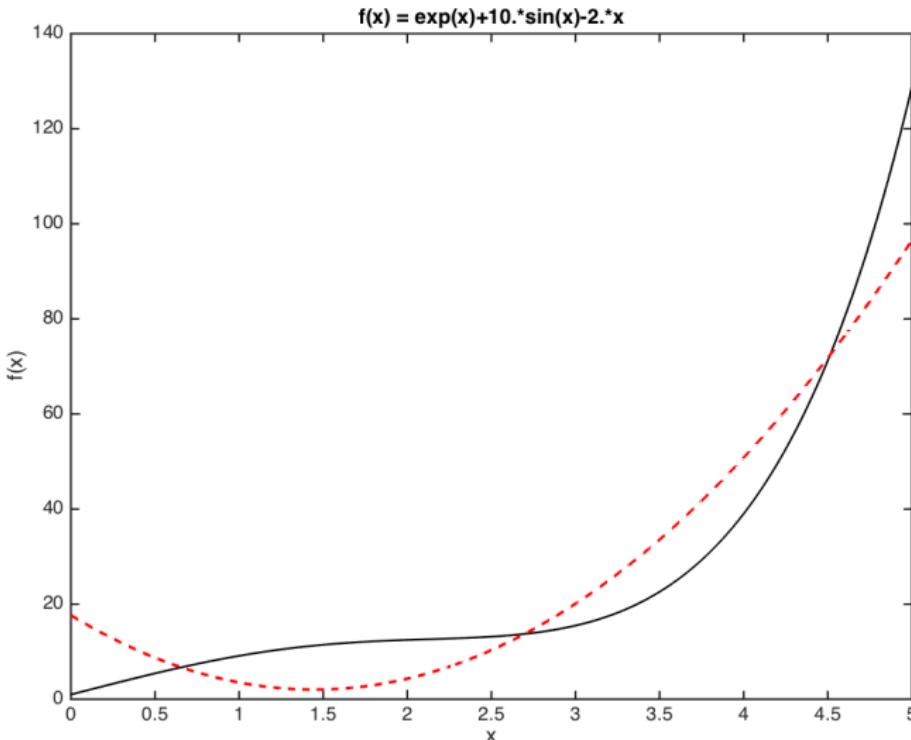
EXAMPLE: “REGRESSION”



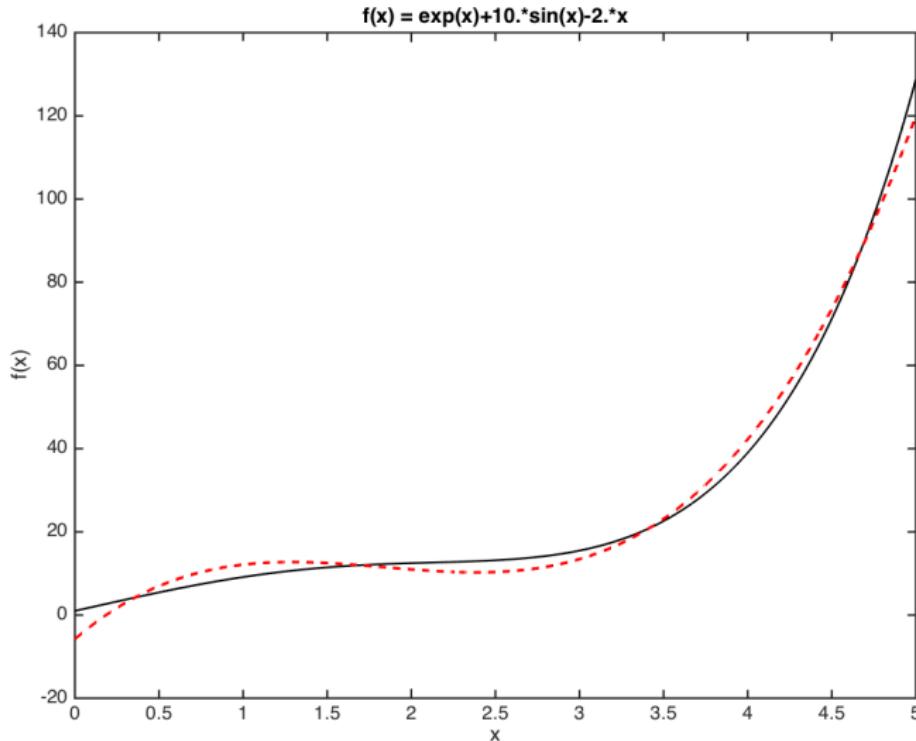
EXAMPLE: “REGRESSION”



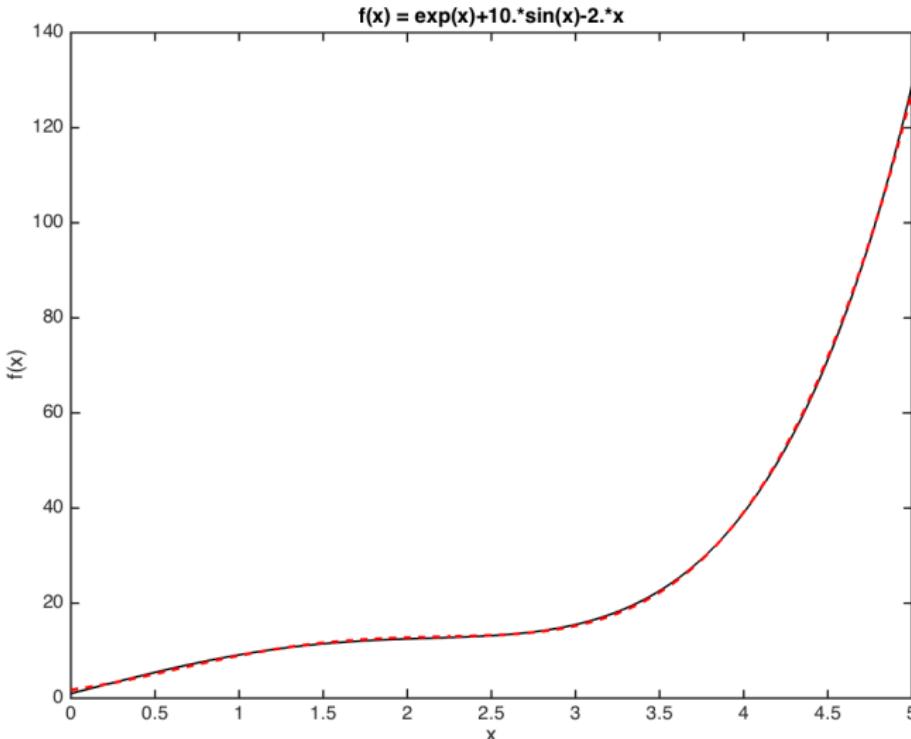
EXAMPLE: “REGRESSION”



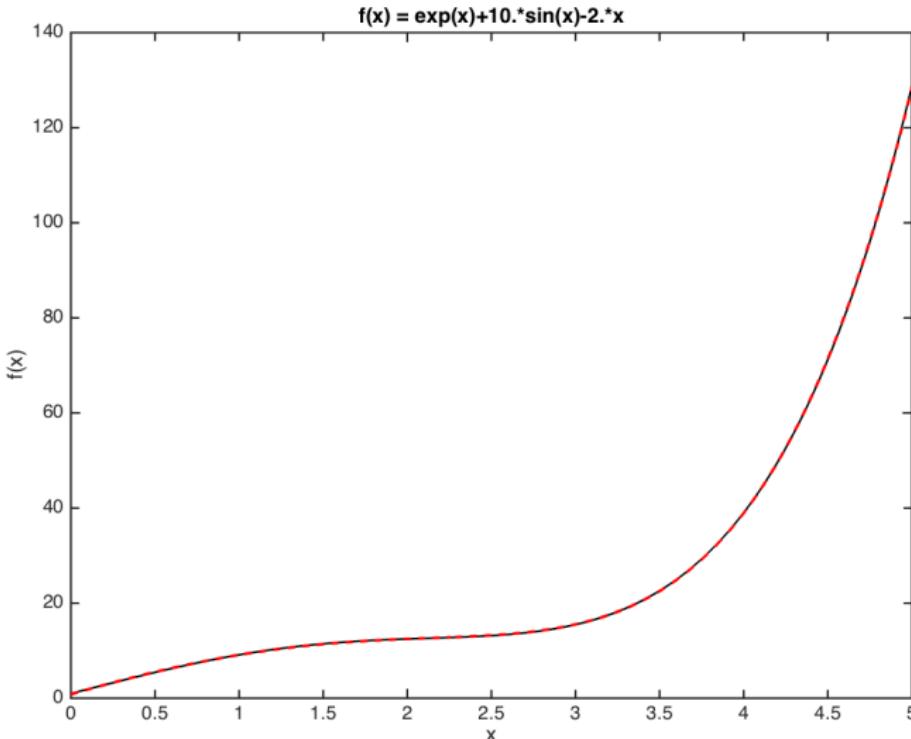
EXAMPLE: “REGRESSION”



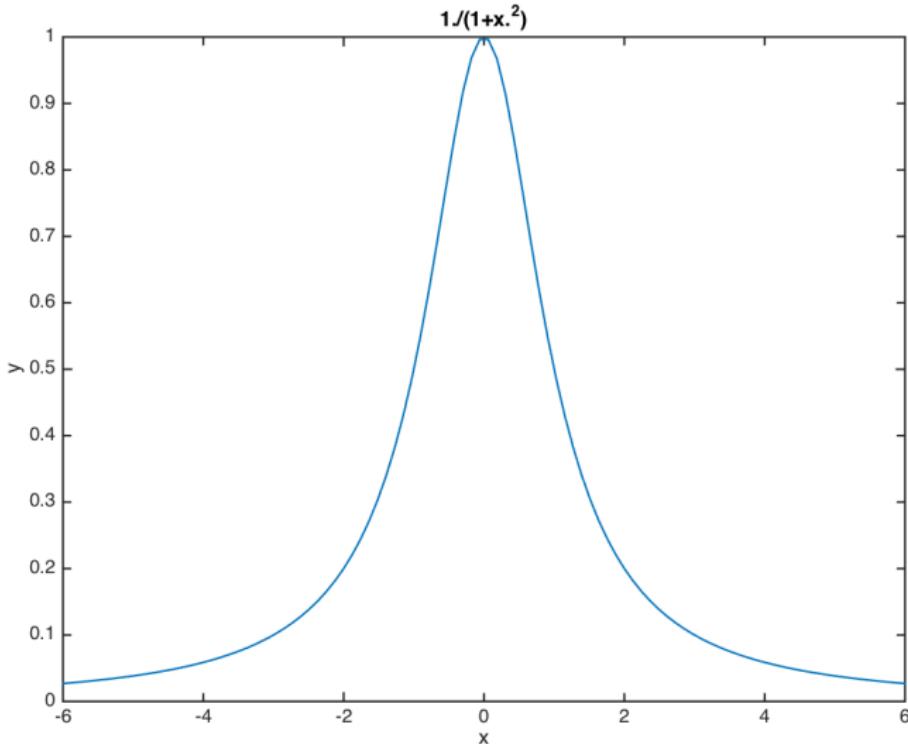
EXAMPLE: “REGRESSION”



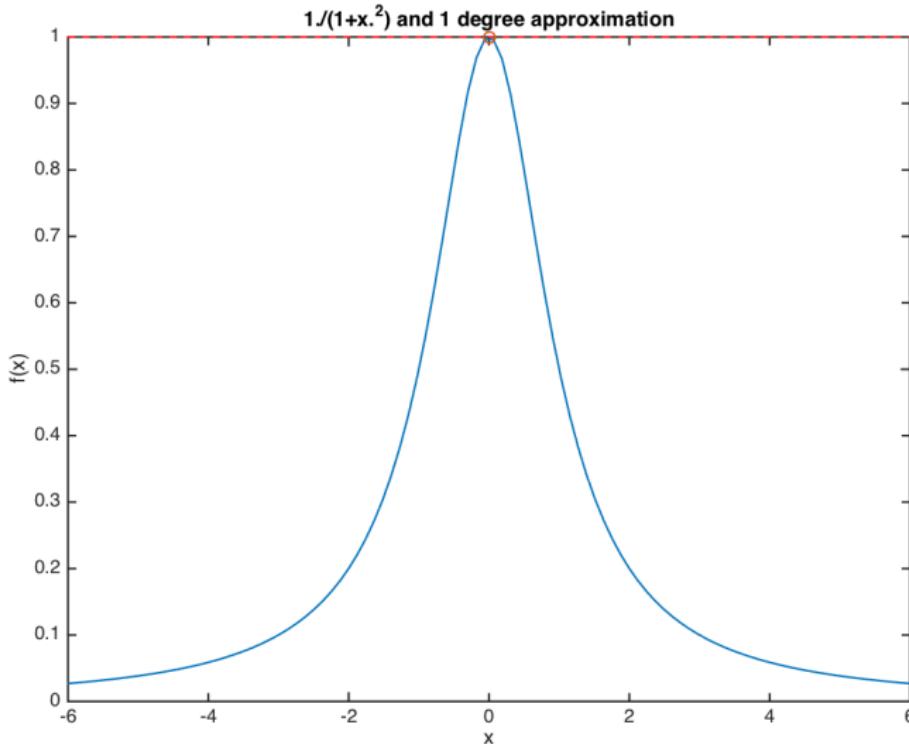
EXAMPLE: “REGRESSION”



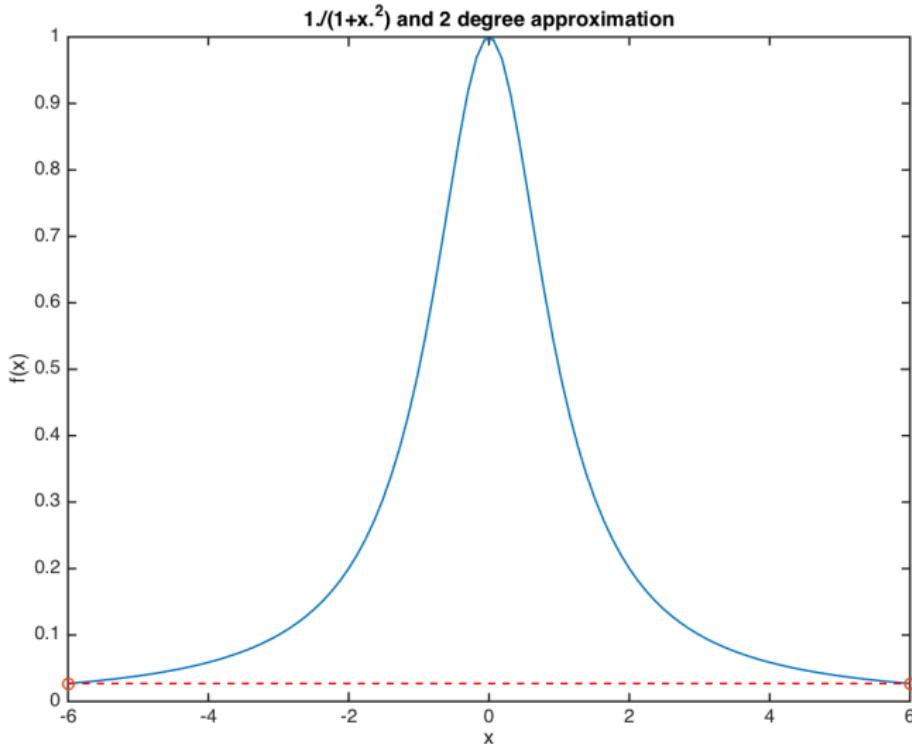
RUNGE'S PHENOMENON: $f(x) = \frac{1}{1+x^2}$



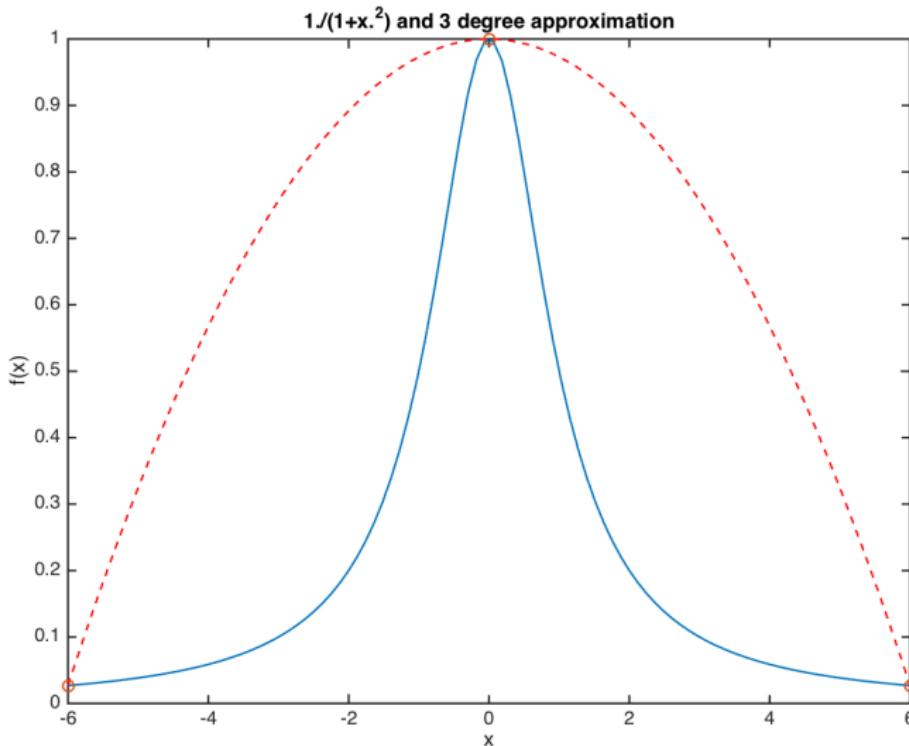
APPROX: $f(x) = \frac{1}{1+x^2}$: 1-DEG. APPROX



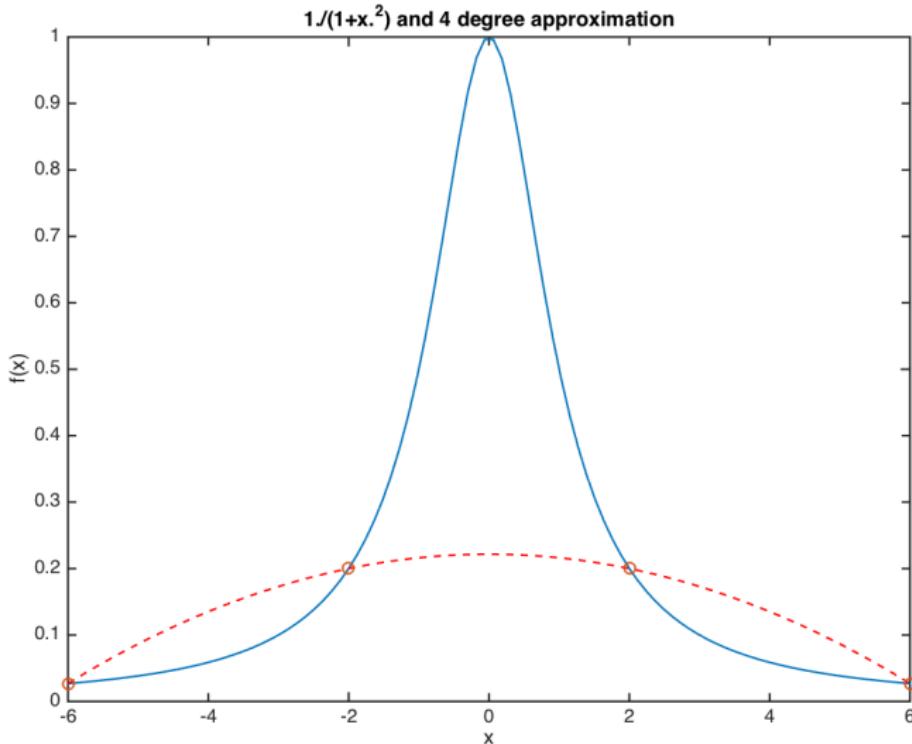
APPROX: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX



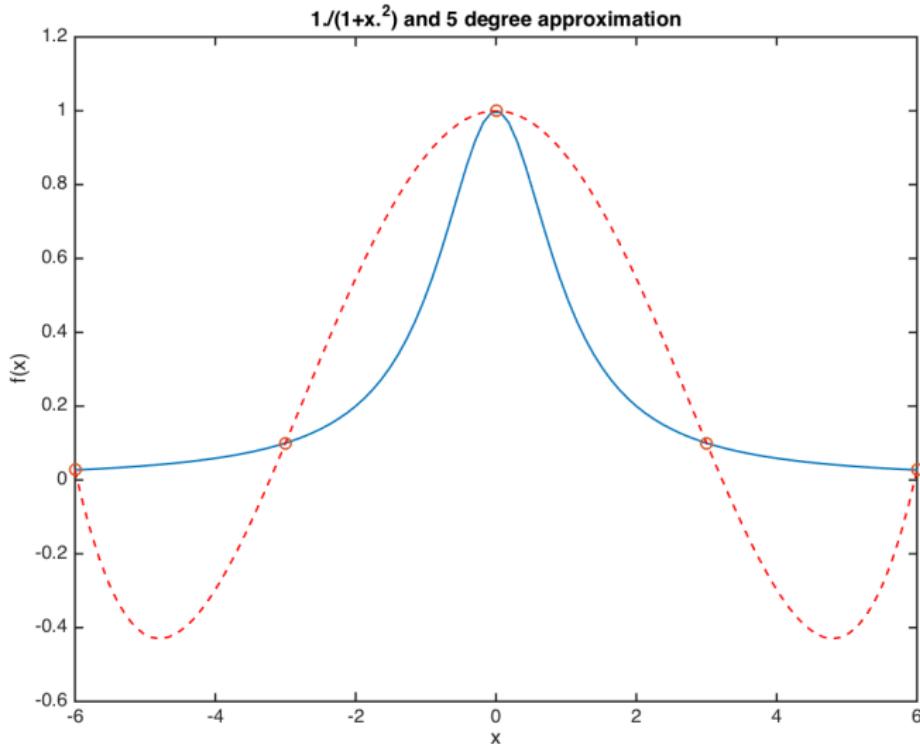
APPROX: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX



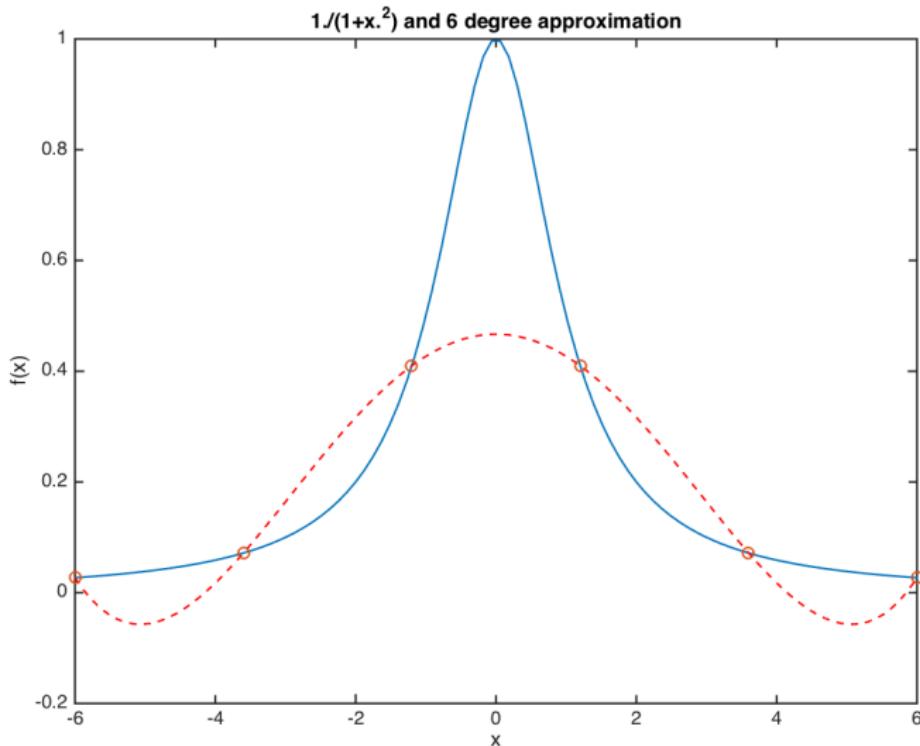
APPROX: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX



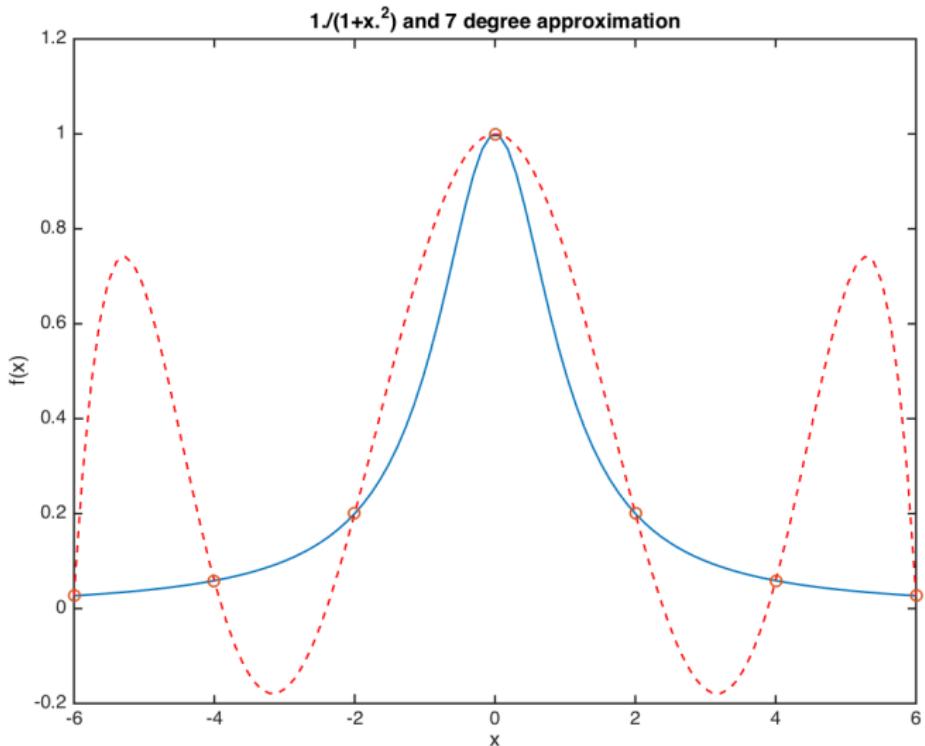
APPROX: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX



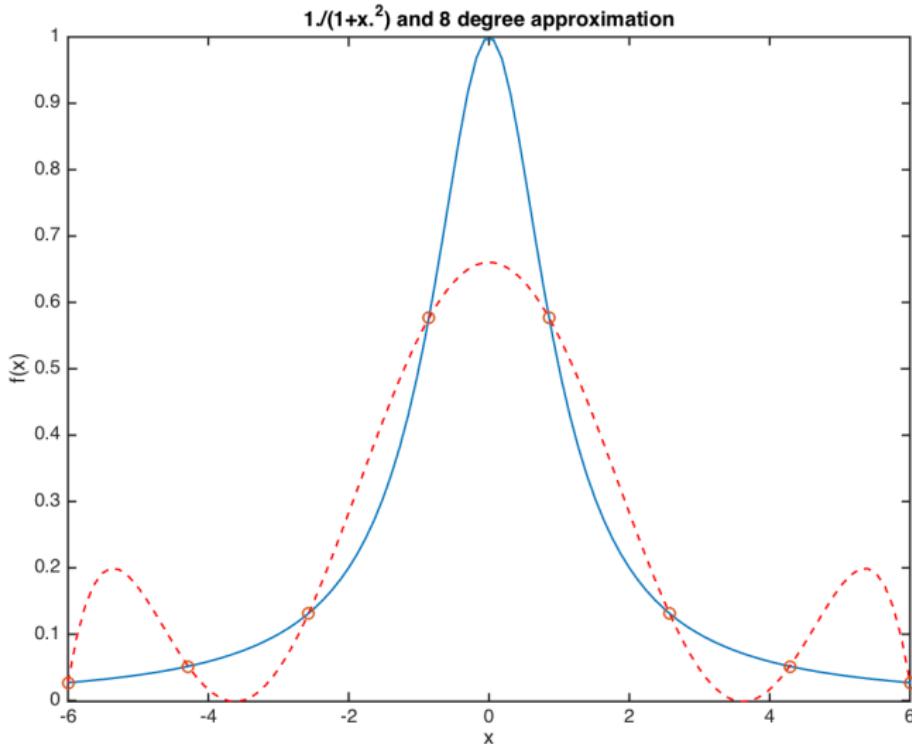
APPROX: $f(x) = \frac{1}{1+x^2}$: 6-DEG. APPROX



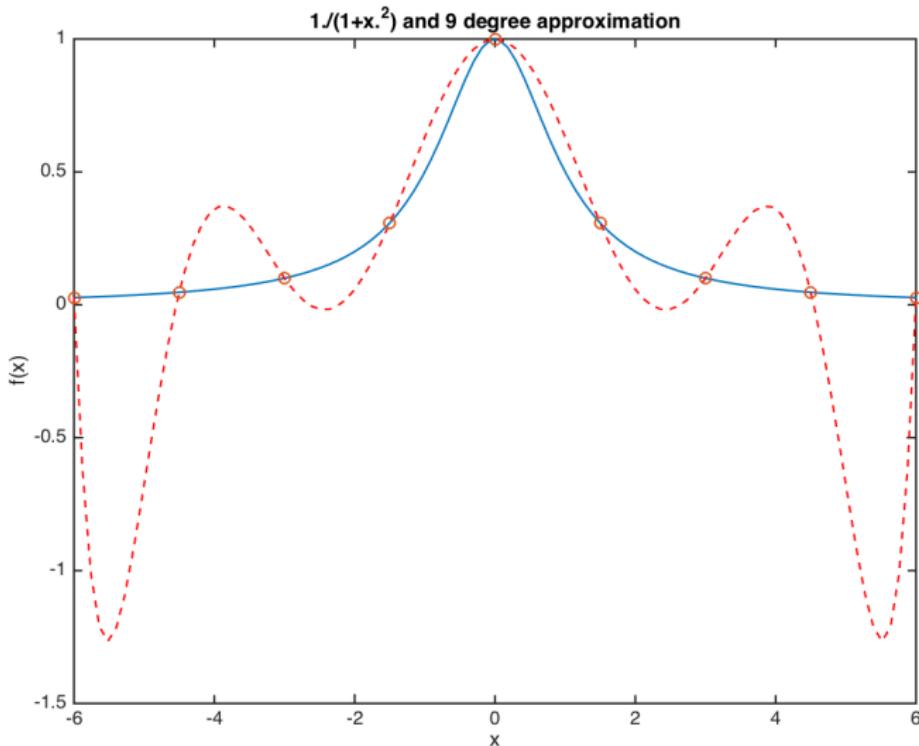
APPROX: $f(x) = \frac{1}{1+x^2}$: 7-DEG. APPROX



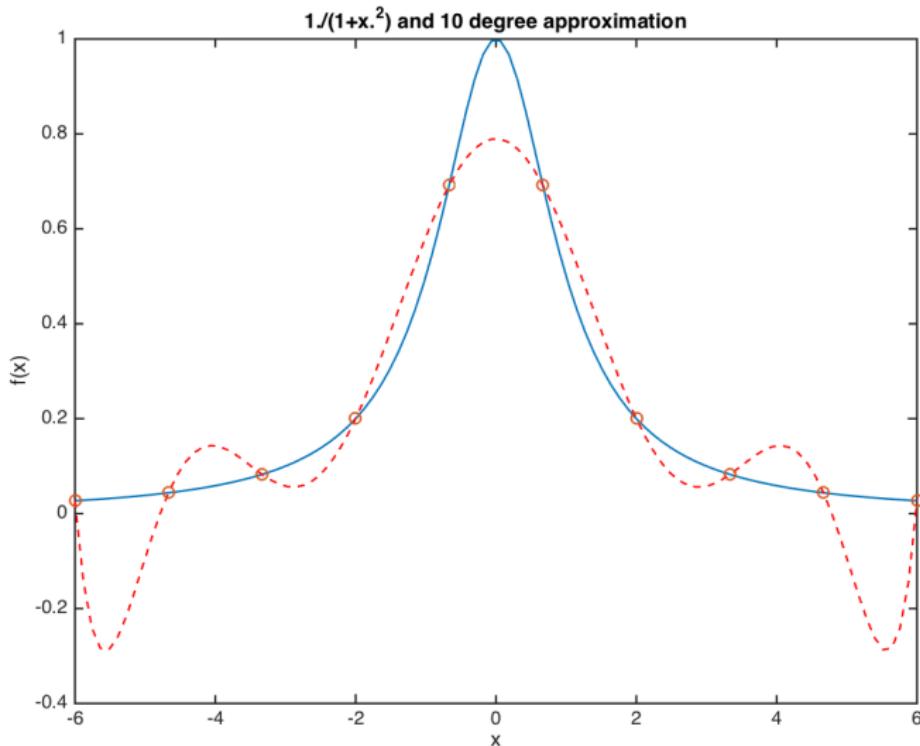
APPROX: $f(x) = \frac{1}{1+x^2}$: 8-DEG. APPROX



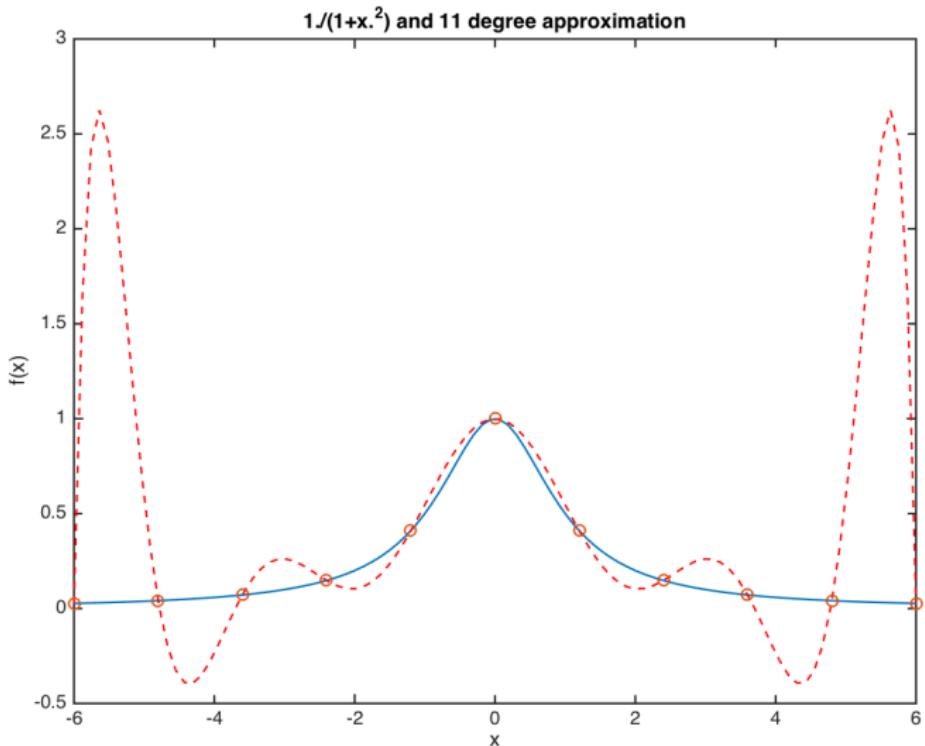
APPROX: $f(x) = \frac{1}{1+x^2}$: 9-DEG. APPROX



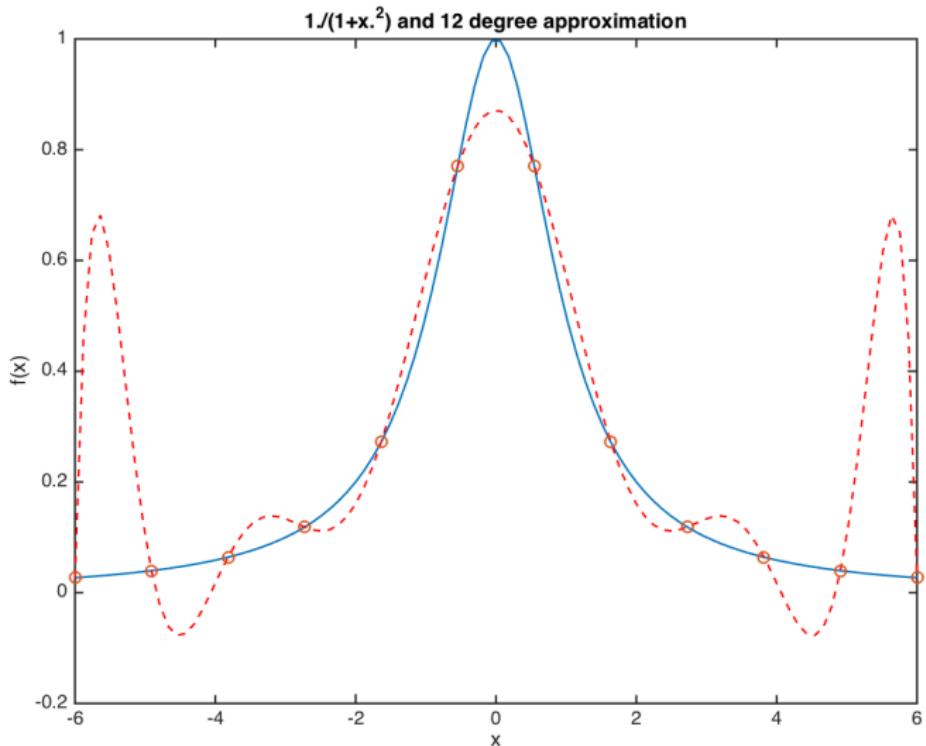
APPROX: $f(x) = \frac{1}{1+x^2}$: 10-DEG. APPROX



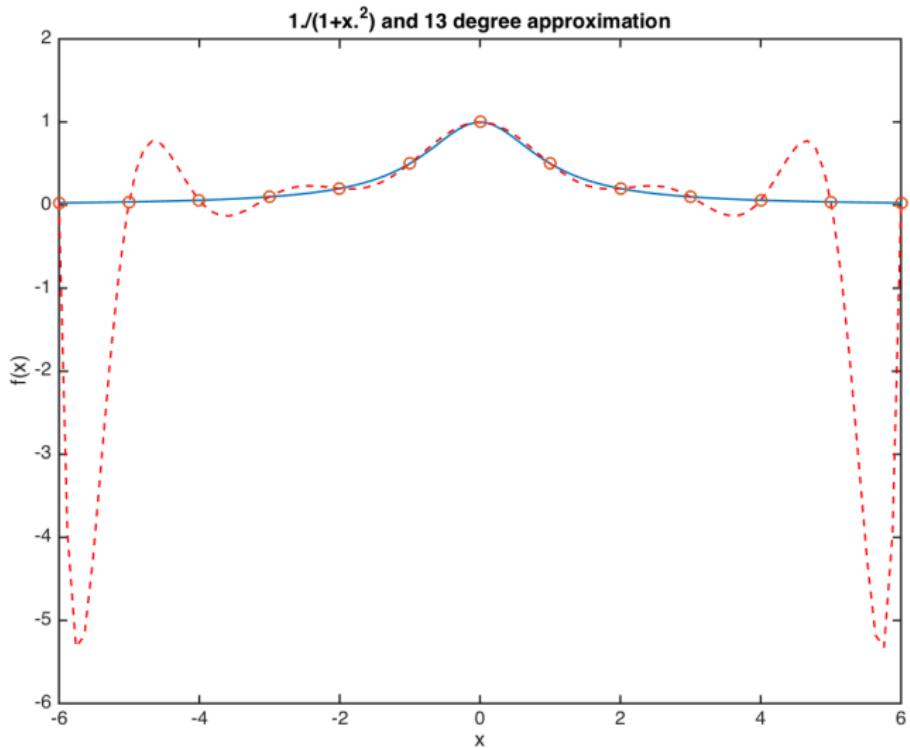
APPROX: $f(x) = \frac{1}{1+x^2}$: 11-DEG. APPROX



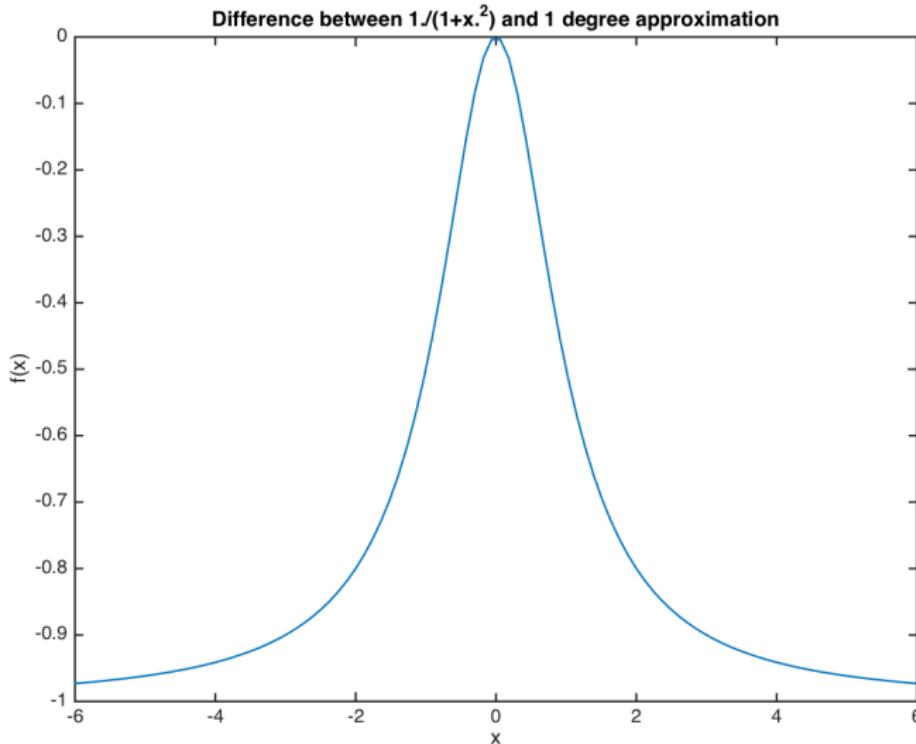
APPROX: $f(x) = \frac{1}{1+x^2}$: 12-DEG. APPROX



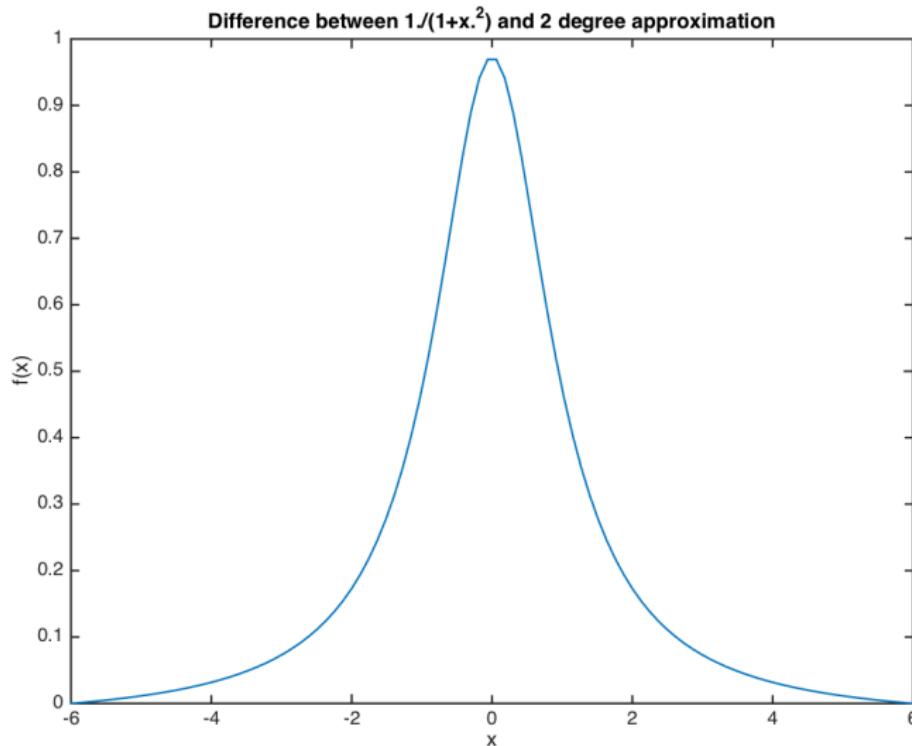
APPROX: $f(x) = \frac{1}{1+x^2}$: 13-DEG. APPROX



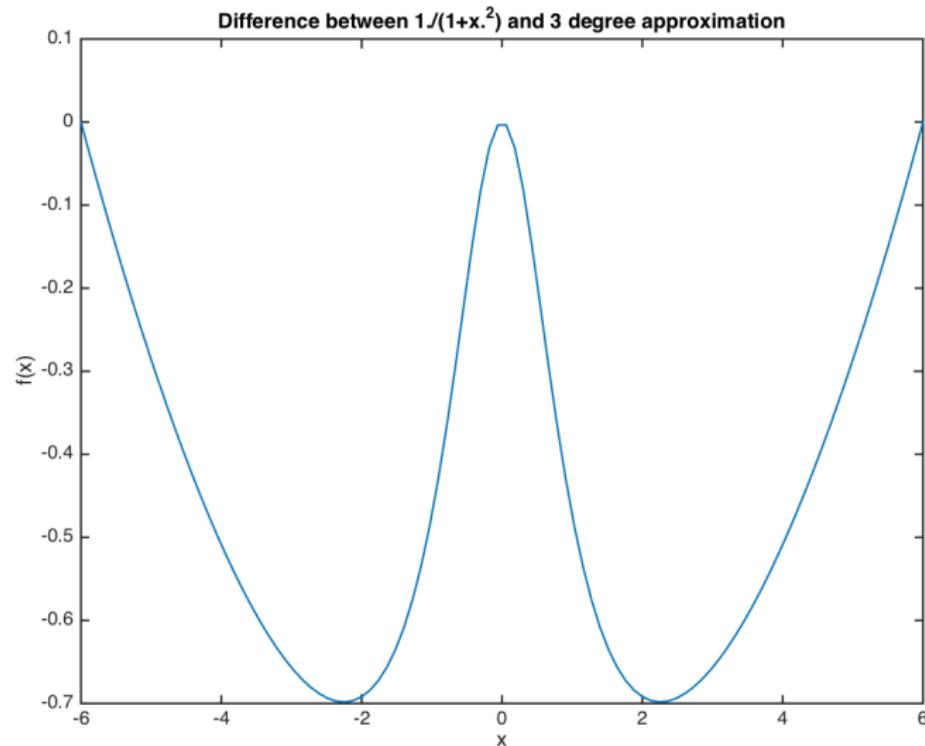
ERRORS: $f(x) = \frac{1}{1+x^2}$: 1-DEG. APPROX



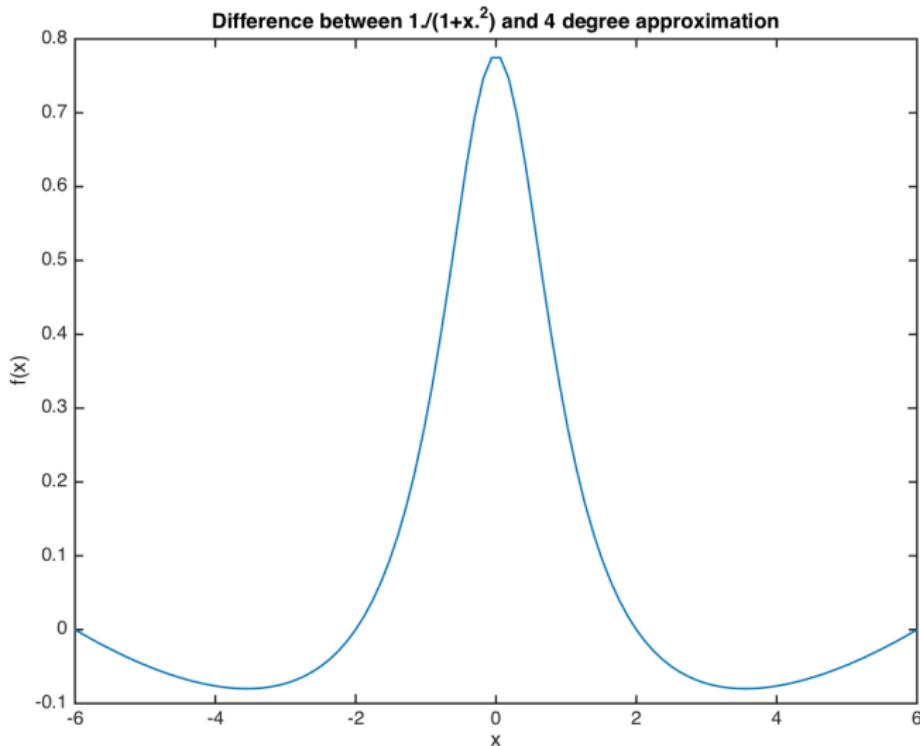
ERRORS: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX



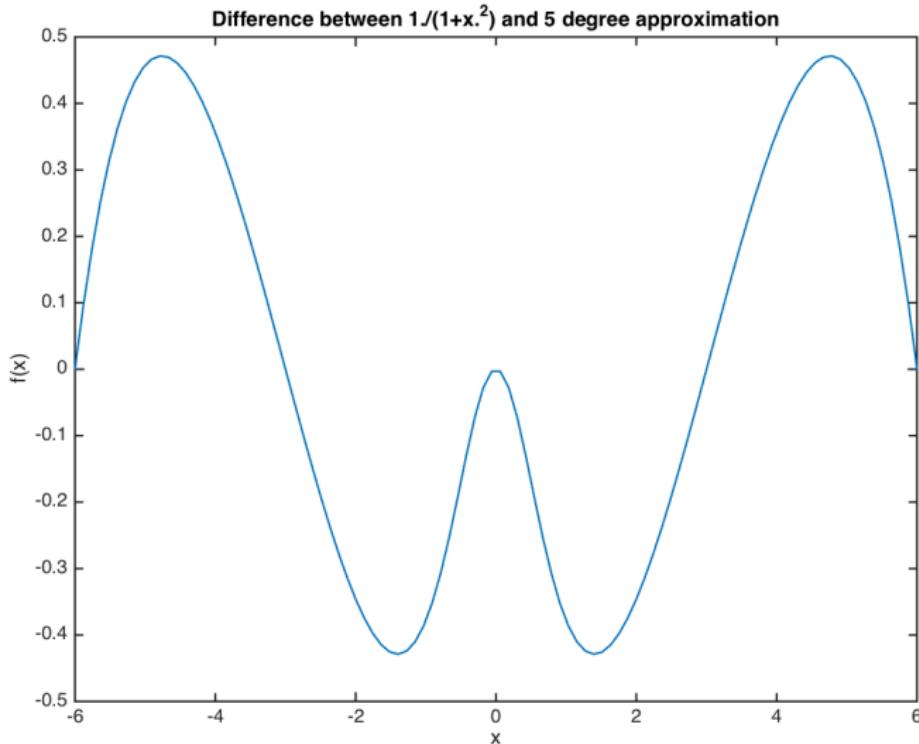
ERRORS: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX



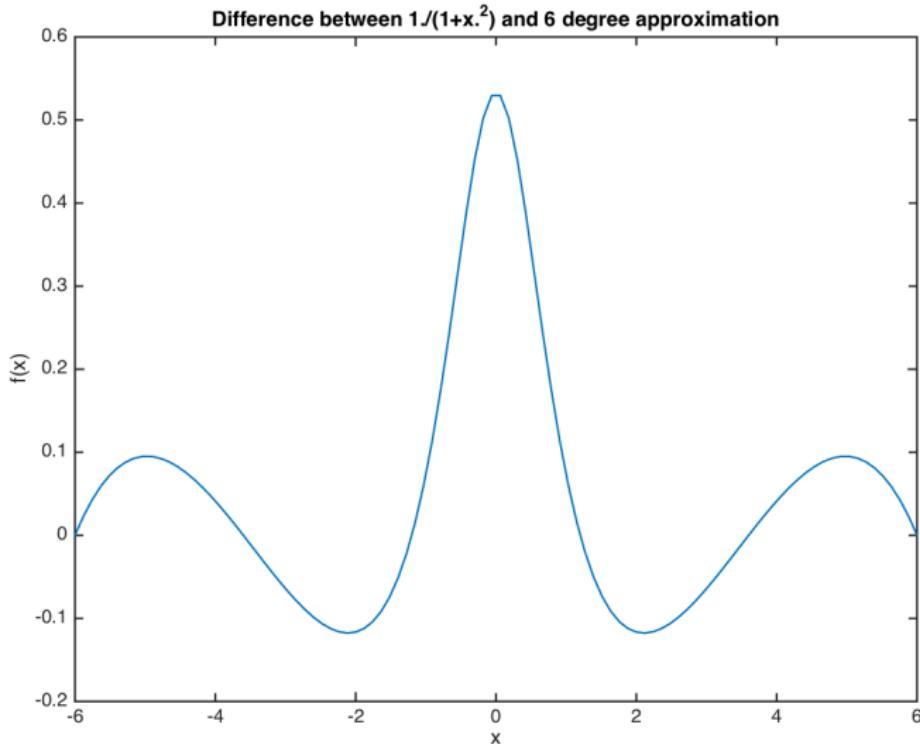
ERRORS: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX



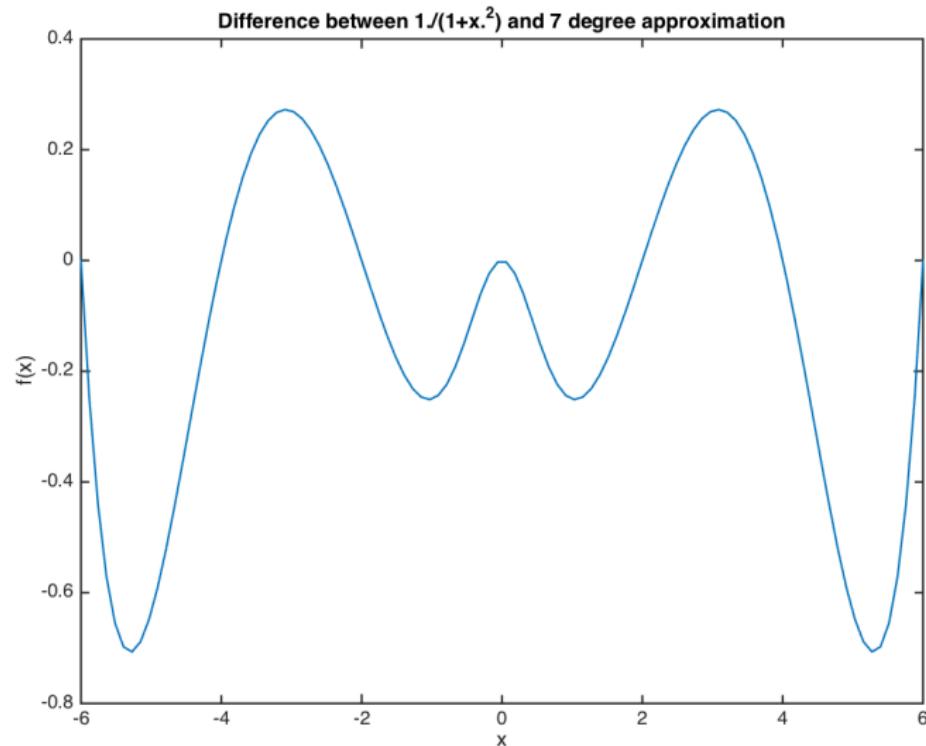
ERRORS: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX



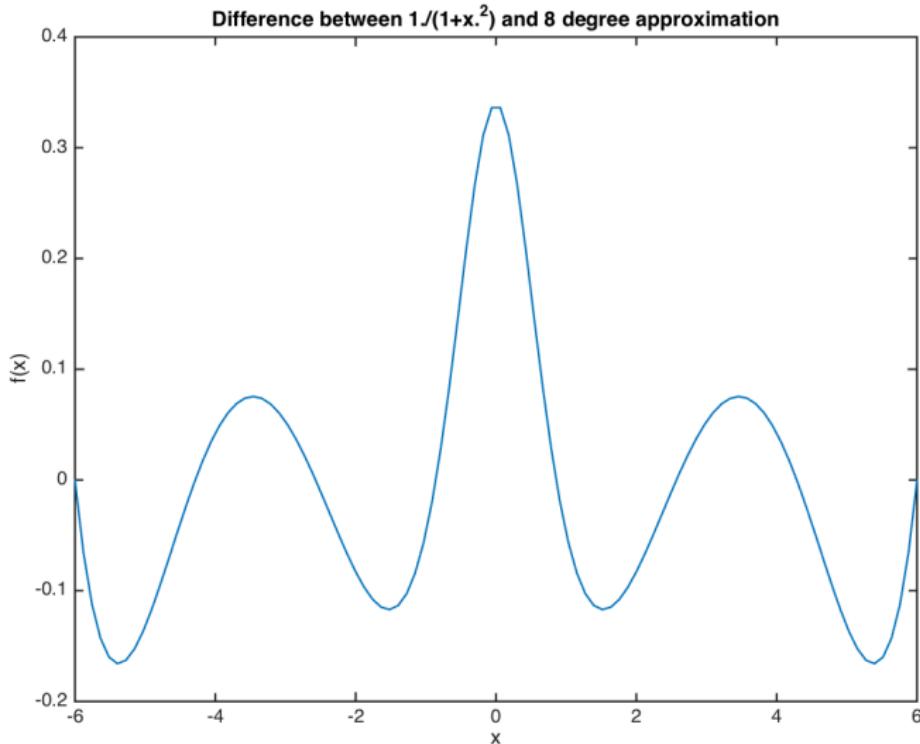
ERRORS: $f(x) = \frac{1}{1+x^2}$: 6-DEG. APPROX



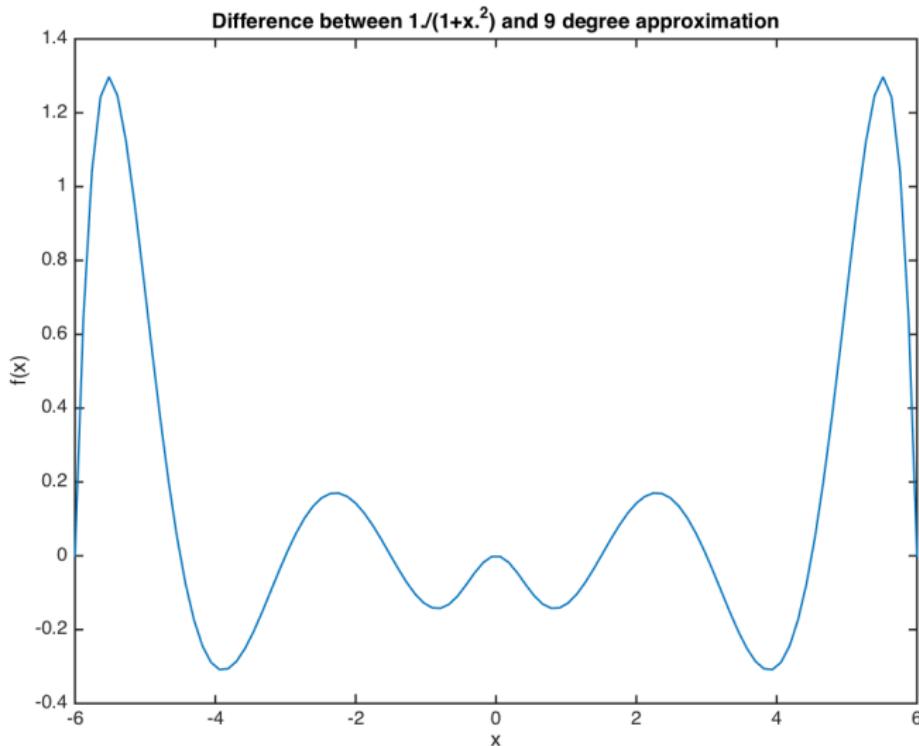
ERRORS: $f(x) = \frac{1}{1+x^2}$: 7-DEG. APPROX



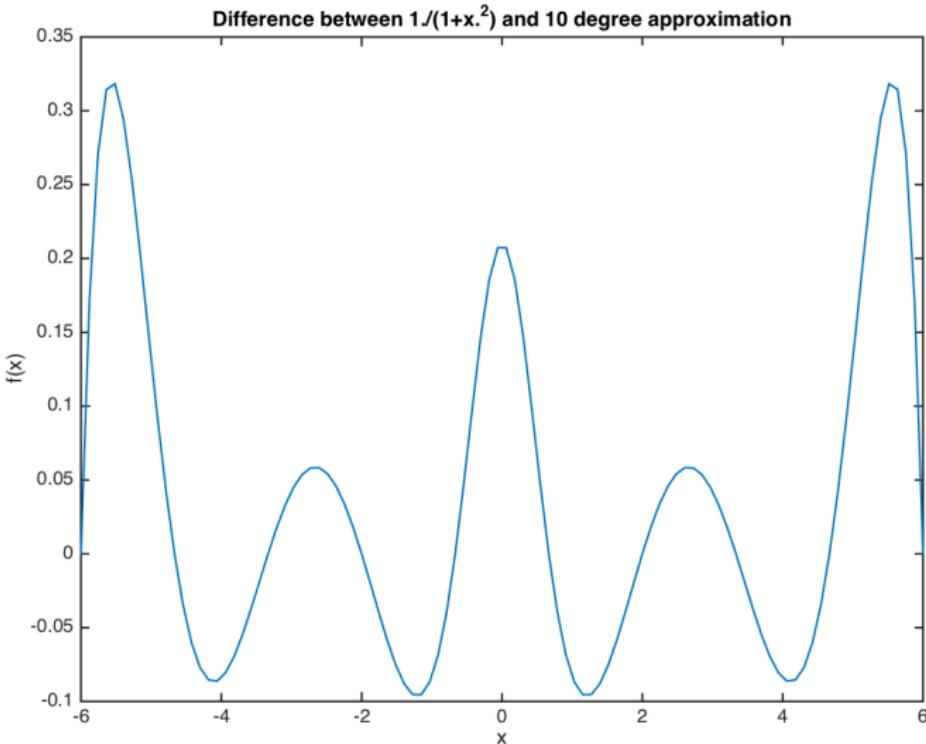
ERRORS: $f(x) = \frac{1}{1+x^2}$: 8-DEG. APPROX



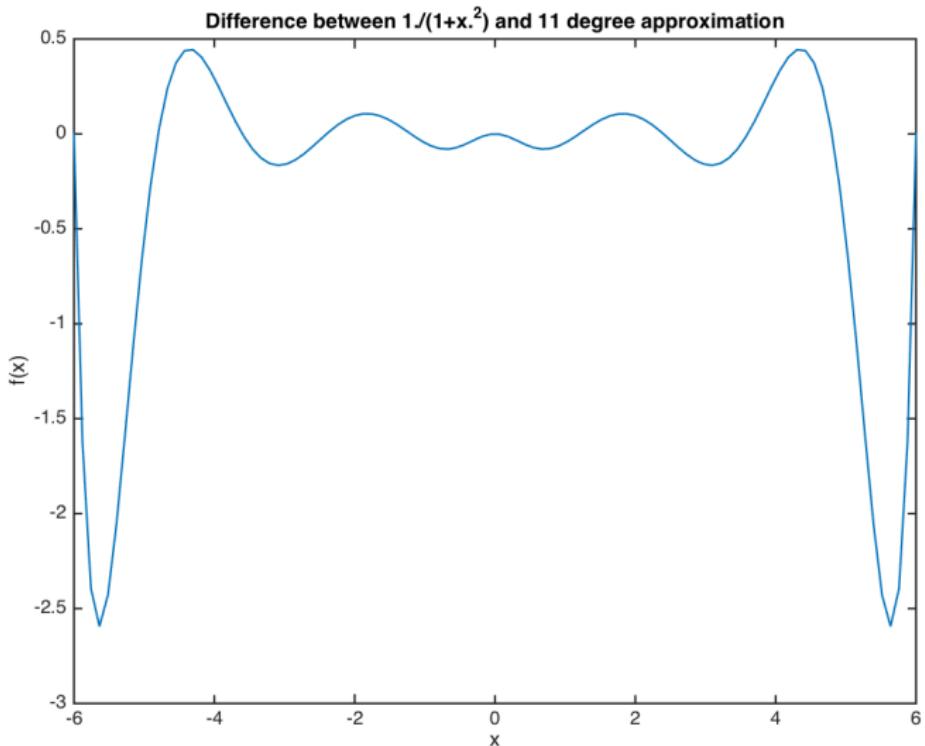
ERRORS: $f(x) = \frac{1}{1+x^2}$: 9-DEG. APPROX



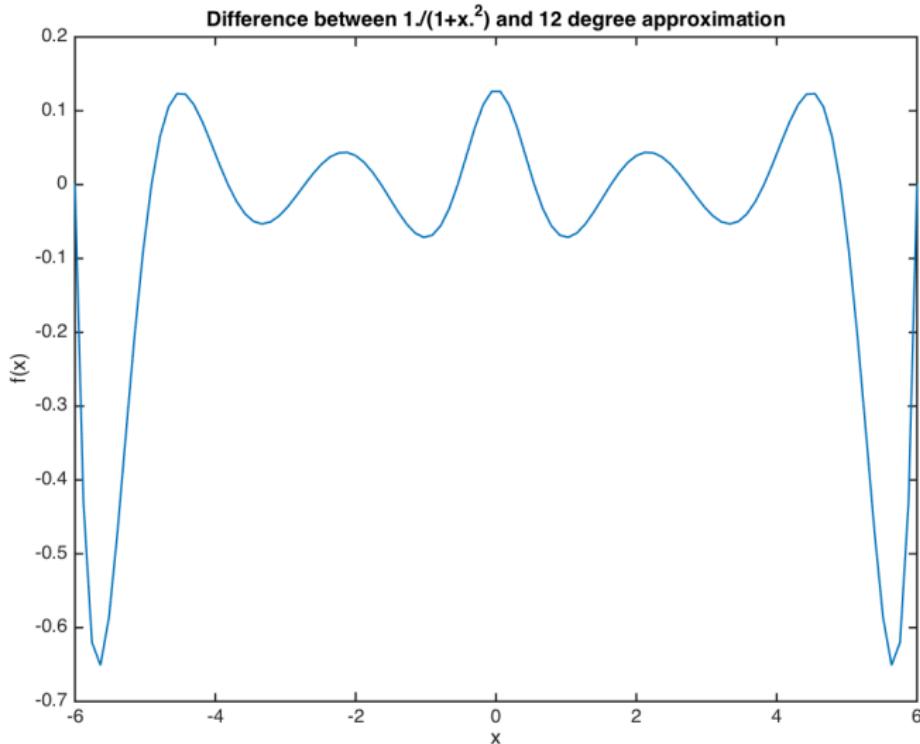
ERRORS: $f(x) = \frac{1}{1+x^2}$: 10-DEG. APPROX



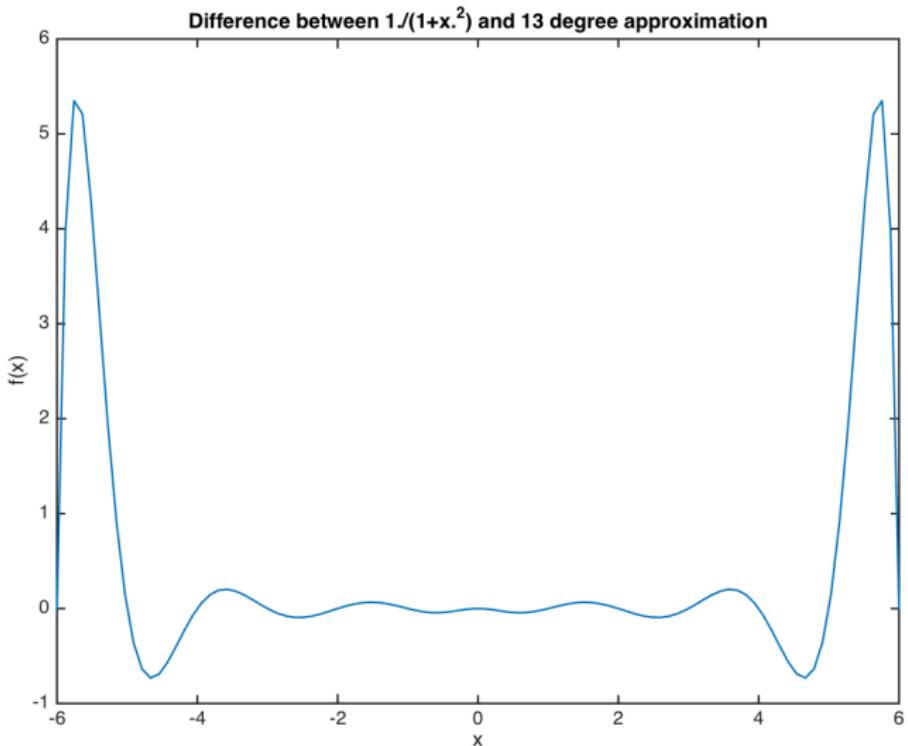
ERRORS: $f(x) = \frac{1}{1+x^2}$: 11-DEG. APPROX



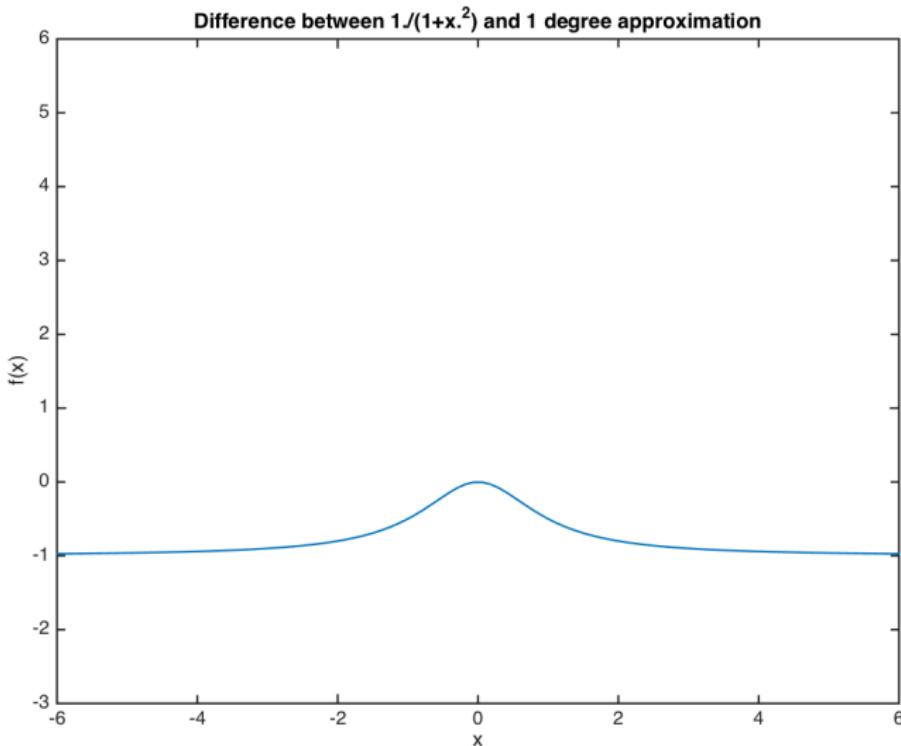
ERRORS: $f(x) = \frac{1}{1+x^2}$: 12-DEG. APPROX



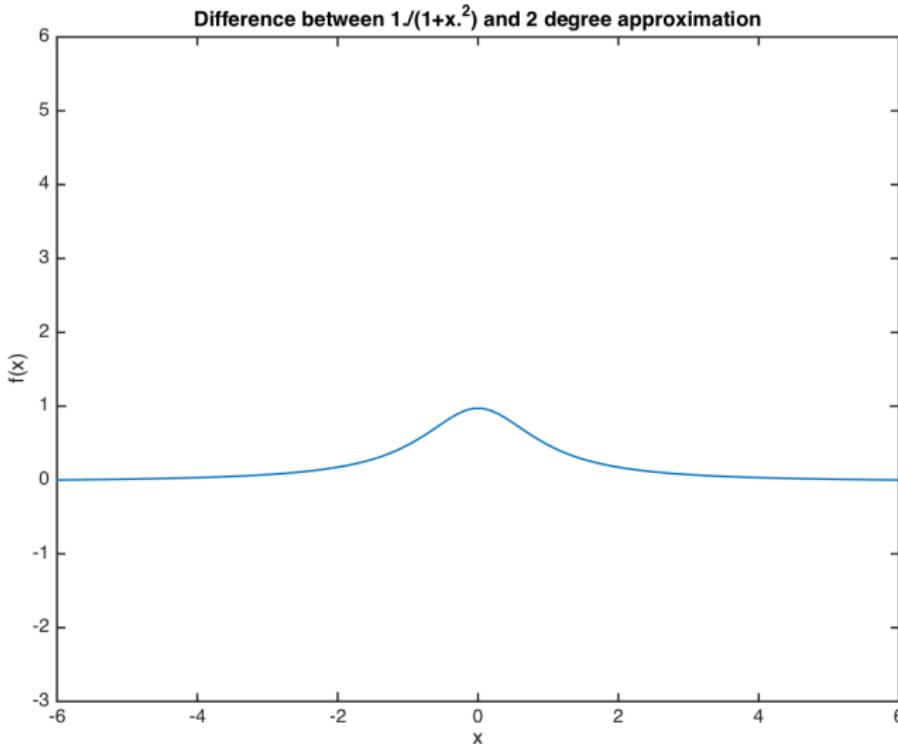
ERRORS: $f(x) = \frac{1}{1+x^2}$: 13-DEG. APPROX



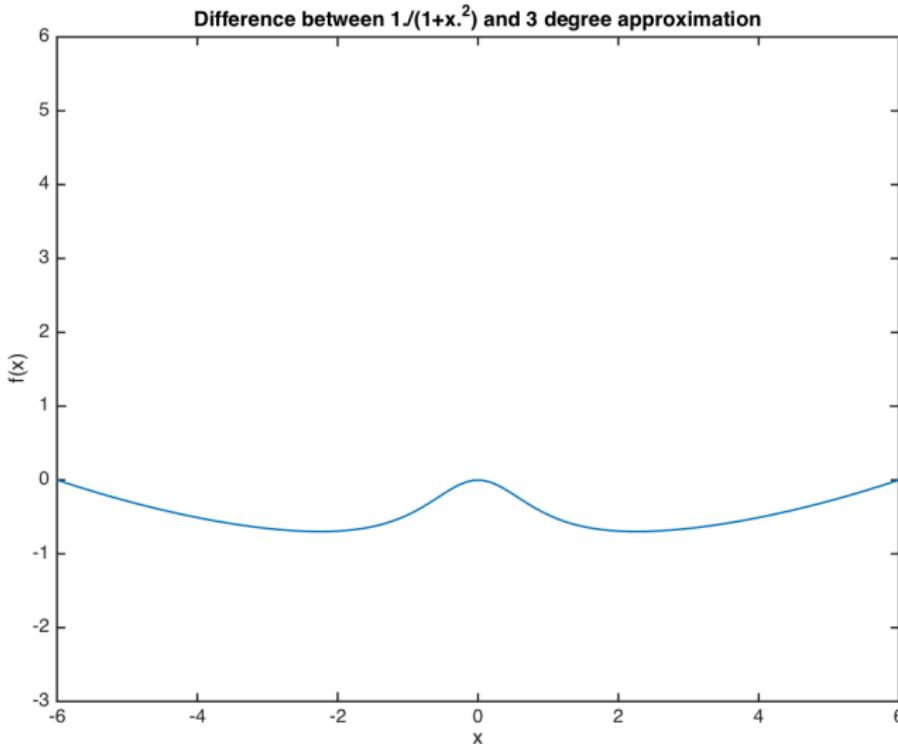
ERRORS: $f(x) = \frac{1}{1+x^2}$: 1-DEG. APPROX



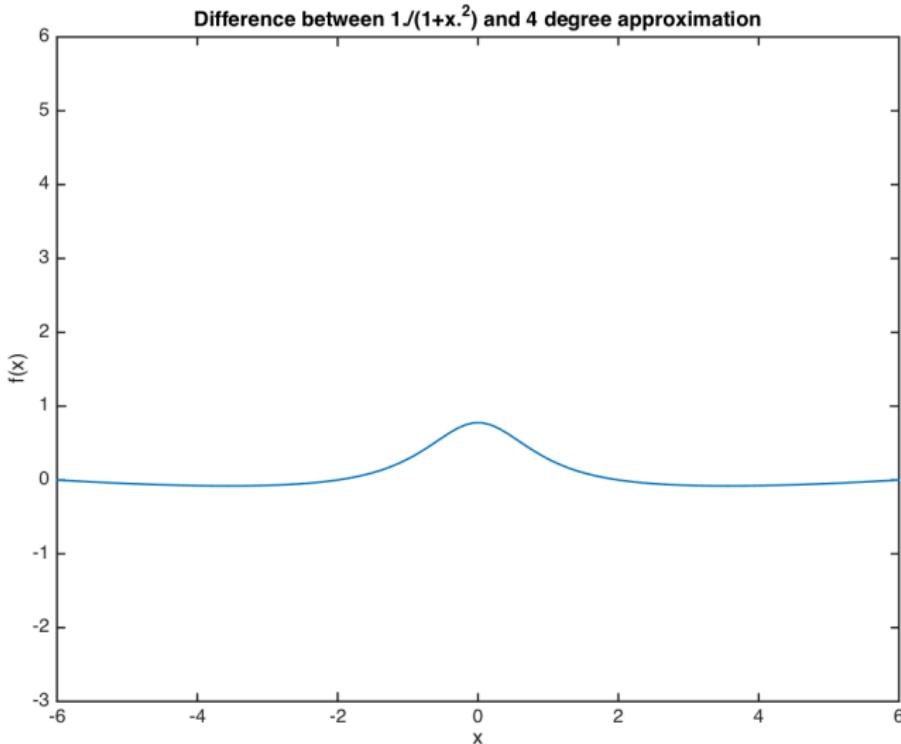
ERRORS: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX



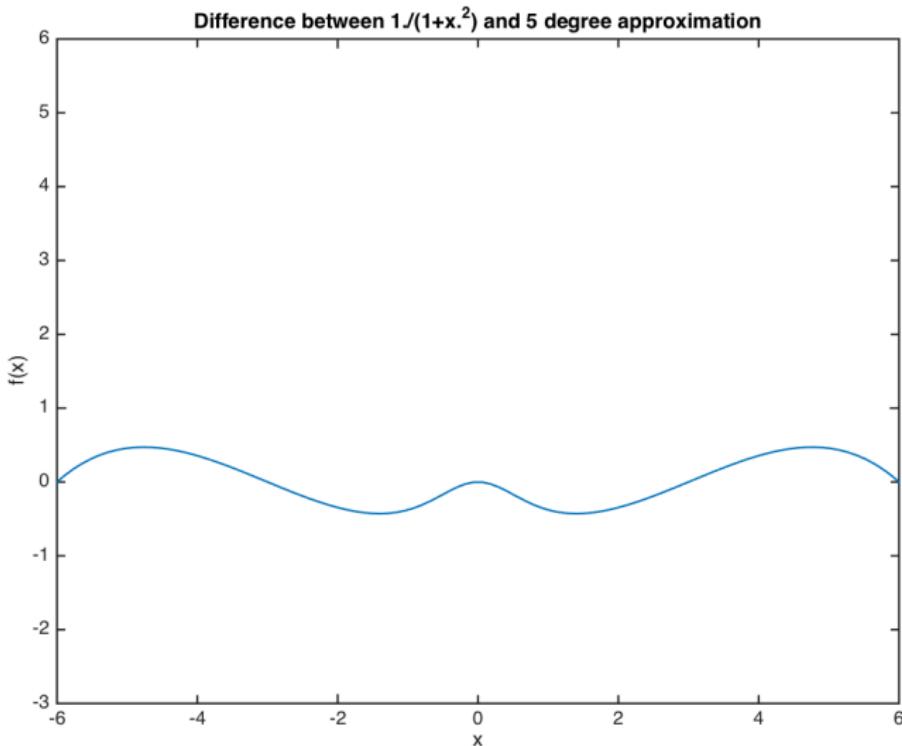
ERRORS: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX



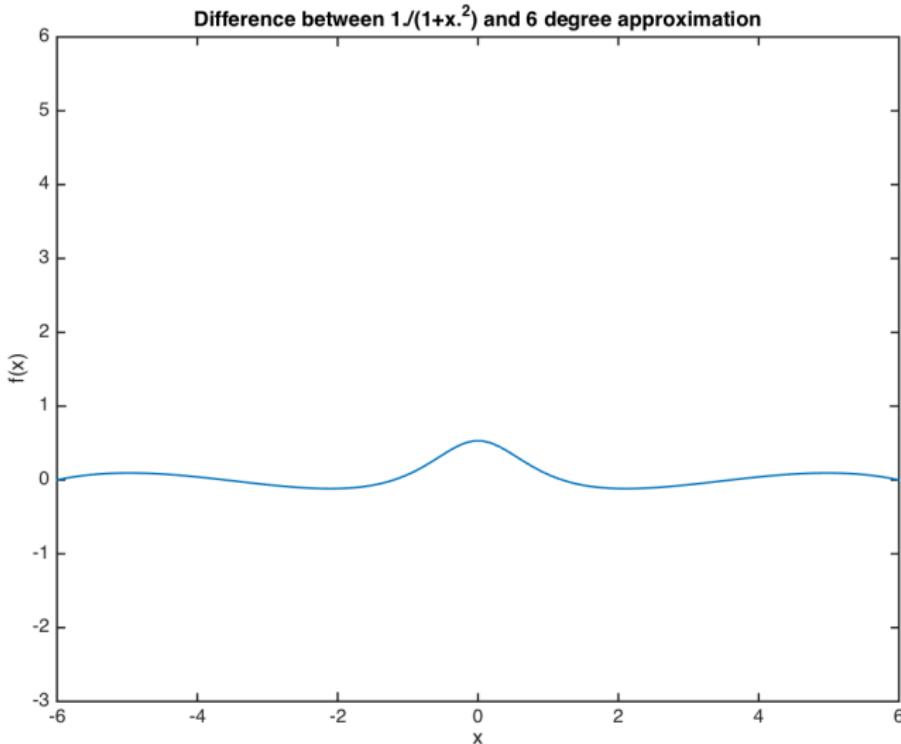
ERRORS: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX



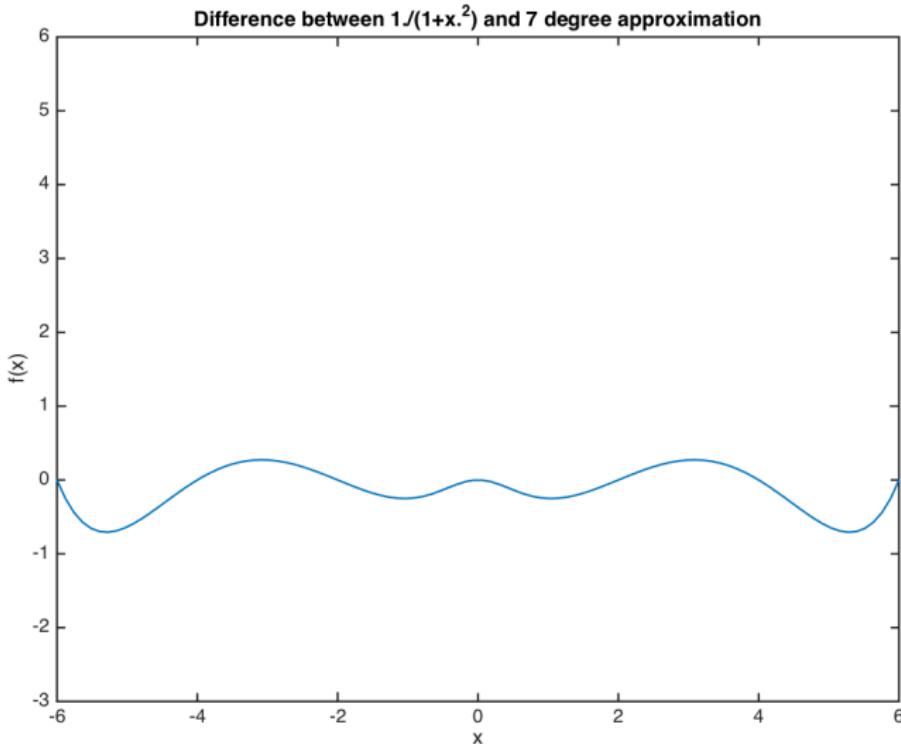
ERRORS: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX



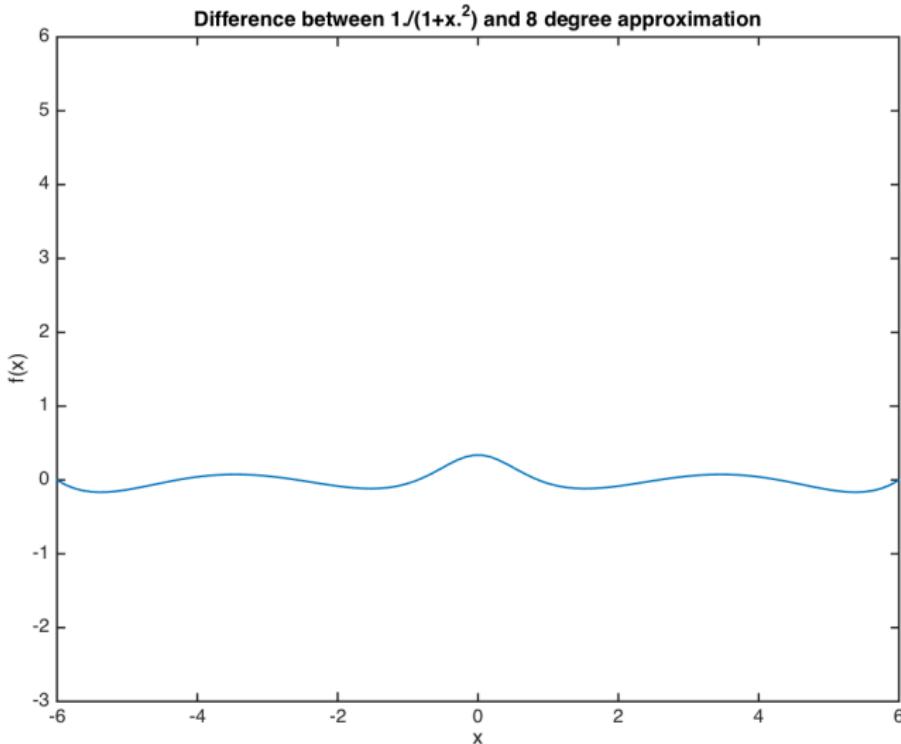
ERRORS: $f(x) = \frac{1}{1+x^2}$: 6-DEG. APPROX



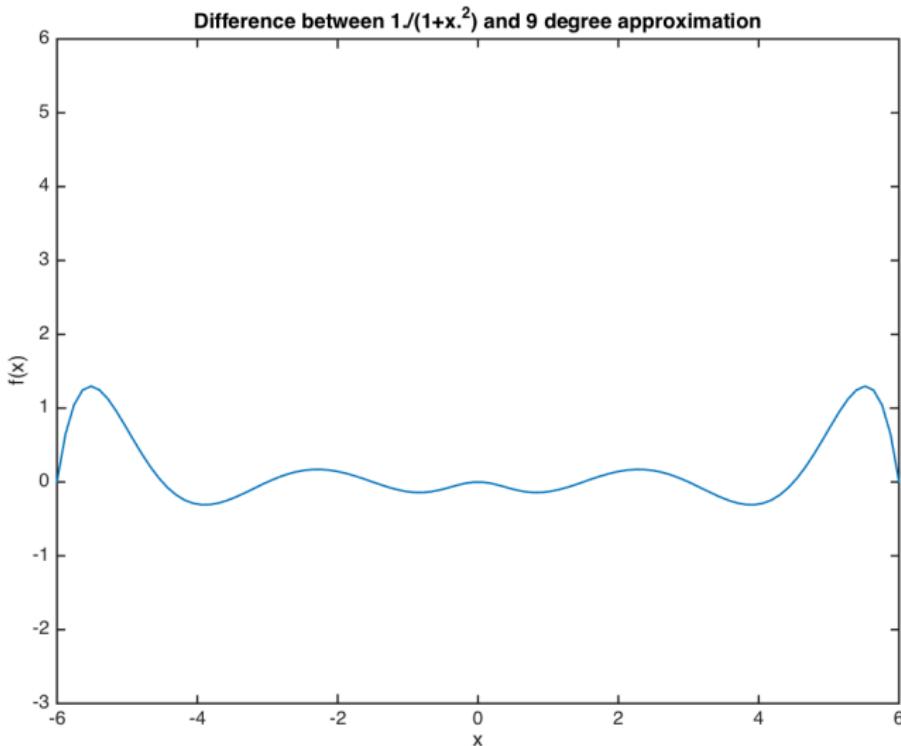
ERRORS: $f(x) = \frac{1}{1+x^2}$: 7-DEG. APPROX



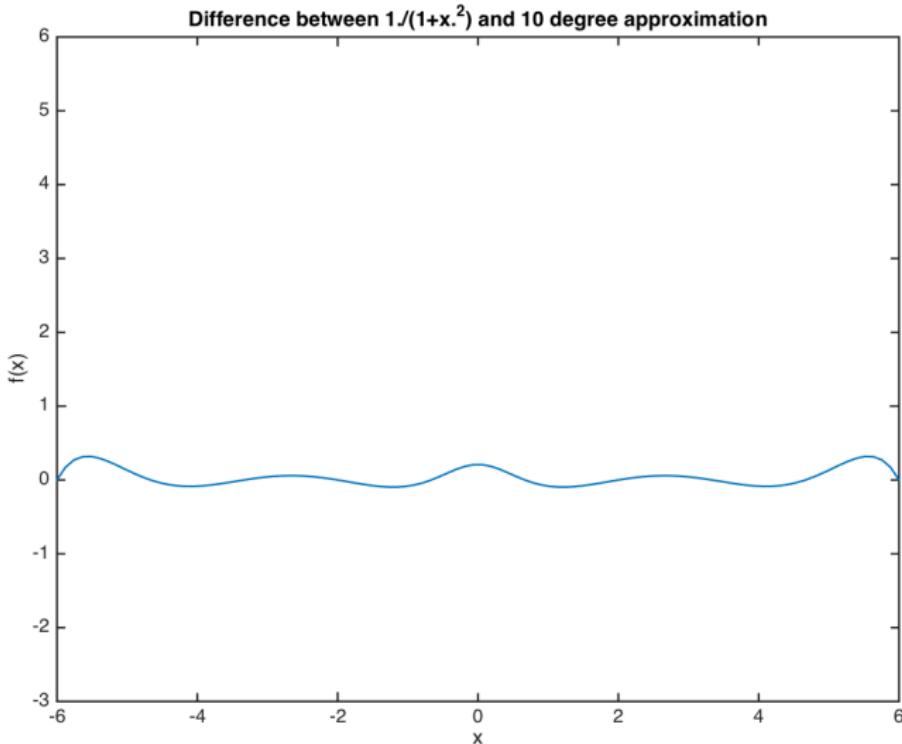
ERRORS: $f(x) = \frac{1}{1+x^2}$: 8-DEG. APPROX



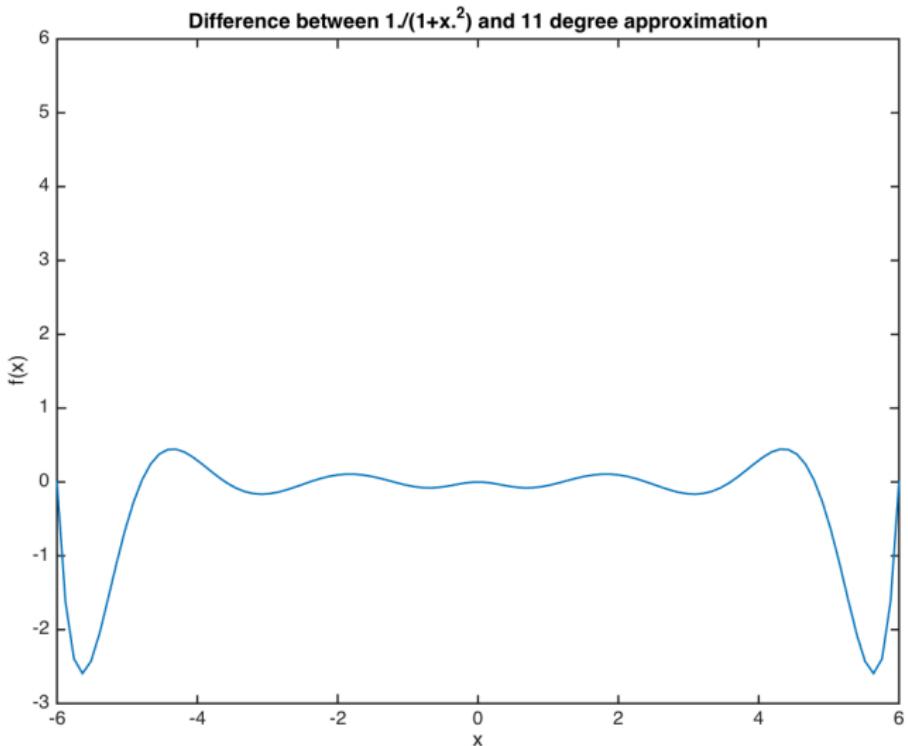
ERRORS: $f(x) = \frac{1}{1+x^2}$: 9-DEG. APPROX



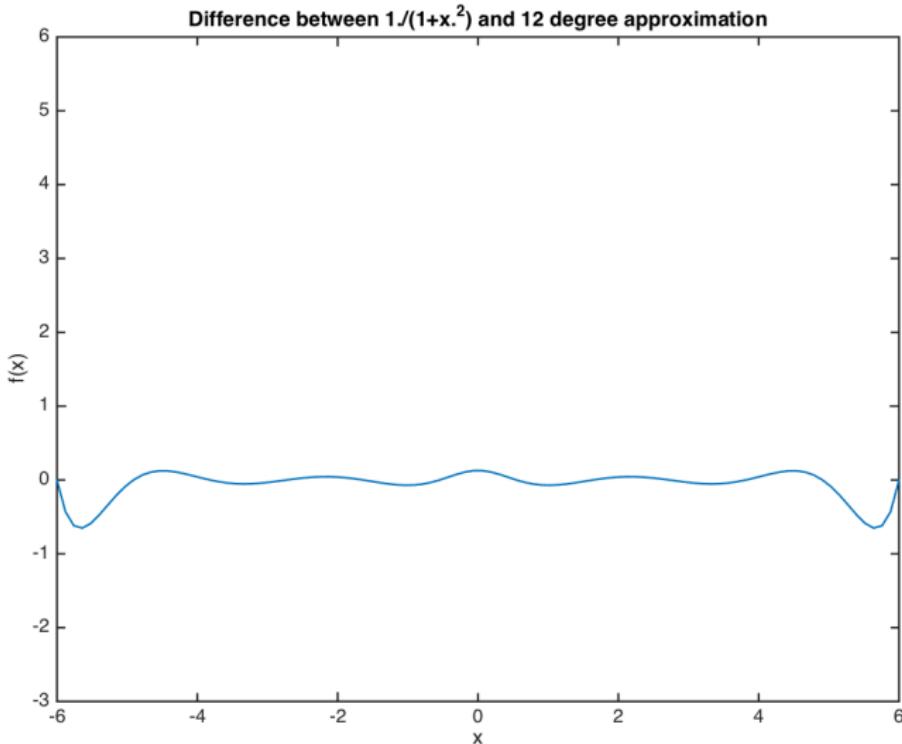
ERRORS: $f(x) = \frac{1}{1+x^2}$: 10-DEG. APPROX



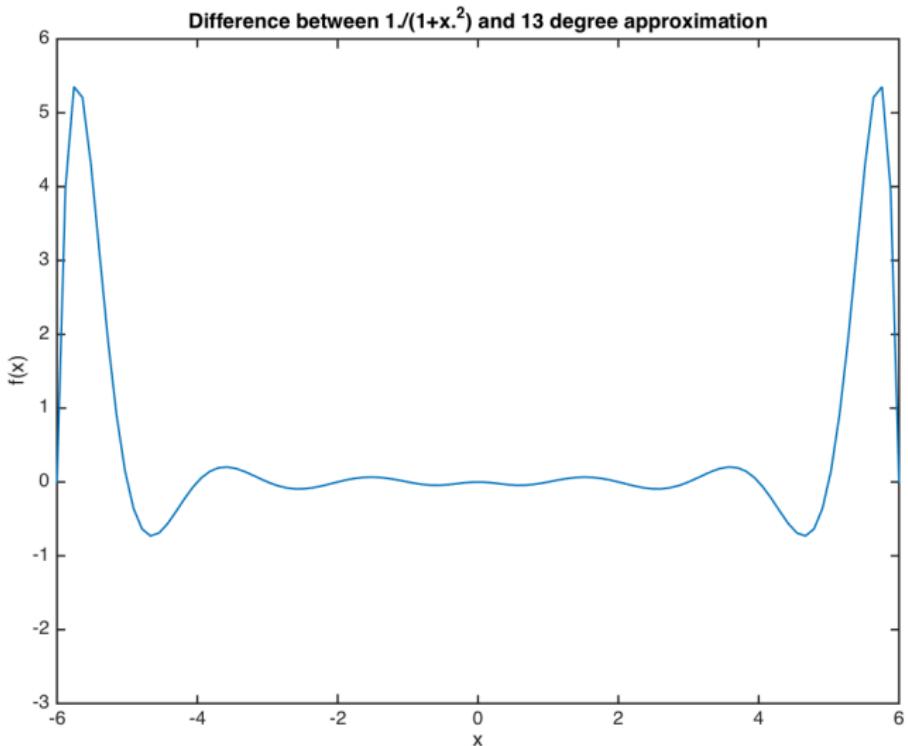
ERRORS: $f(x) = \frac{1}{1+x^2}$: 11-DEG. APPROX



ERRORS: $f(x) = \frac{1}{1+x^2}$: 12-DEG. APPROX



ERRORS: $f(x) = \frac{1}{1+x^2}$: 13-DEG. APPROX



CHEBYCHEV POLYNOMIALS

- ▶ Less violent and colinear polynomials are desirable
- ▶ We also want something that avoids Runge's phenomenon if possible
- ▶ We also want something that's pretty "easy" to integrate
- ▶ It just so happens Chebychev polynomials are all of these things
- ▶ Defined recursively:

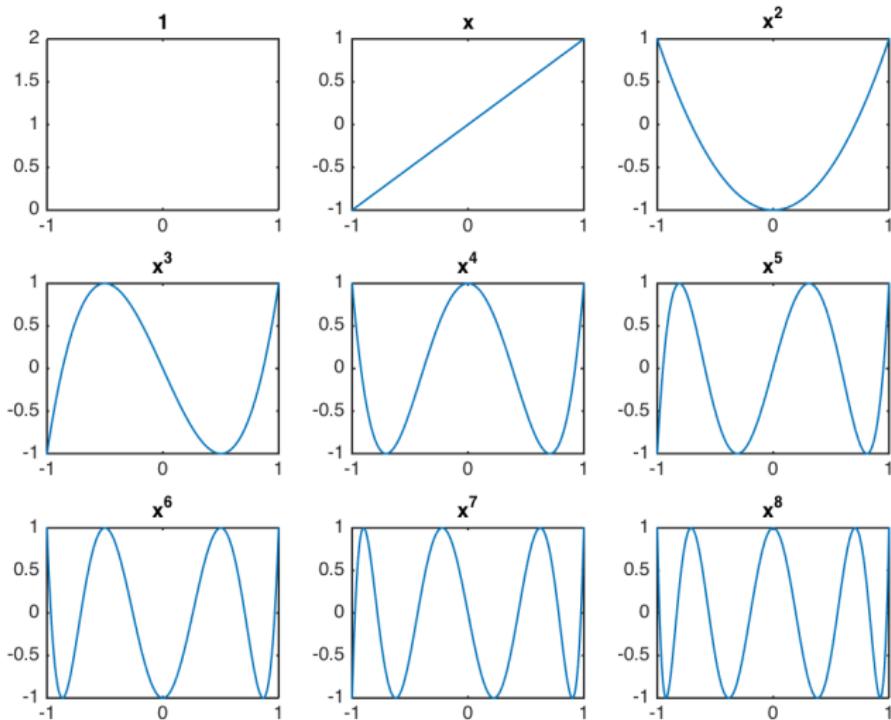
$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$$

- ▶ Defined over $[-1,1]$! Needs transformation.

CHEBYCHEV BASIS



PROPERTIES

- ▶ Also can show that:

$$T_n(x) = \cosh(n \cdot \text{arccosh}(x))$$

- ▶ And that the roots x_j^* T_n are:

$$x_j = \cos\left(\frac{2j-1}{2n}\pi\right) \quad j = 1, \dots, n$$

- ▶ It just so happens that sampling at Chebychev nodes help minimize the maximum error and make sure it's monotonically decreasing as n increases

INTERPOLATION: 1-D

- ▶ You have some function $f(x)$ that you want to interpolate over $[a, b]$.
- ▶ Define a bridge between your range and the interpolating polynomial's range:
- ▶ Define: $z = 2(x - a)/(b - a) - 1$, or $x = a + \frac{(b-a)(z+1)}{2}$
- ▶ Then
 - ▶ $x = a \Rightarrow z = -1$
 - ▶ $x = b \Rightarrow z = 1$
 - ▶ $x = (a + b)/2 \Rightarrow z = 0$.
- ▶ Our goal will be, for interpolating polynomial $g(z)$ over $[-1, 1]$ and $f(x)$ over $[a, b]$, that:

$$g(2(x - a)/(b - a) - 1) = f(x) \quad \forall x$$

INTERPOLATION: 1-D

1. Pick fcn. $f(x)$, degree n , bounds a, b for interpolation.
2. Find Chebychev nodes for $f(x)$ in "z" space.

$$z_j = -\cos \left(\frac{2j-1}{2n} \pi \right) \quad j = 1, \dots, n$$

3. Convert nodes into "x" space:

$$x_j = a + \frac{(b-a)(z_j + 1)}{2}$$

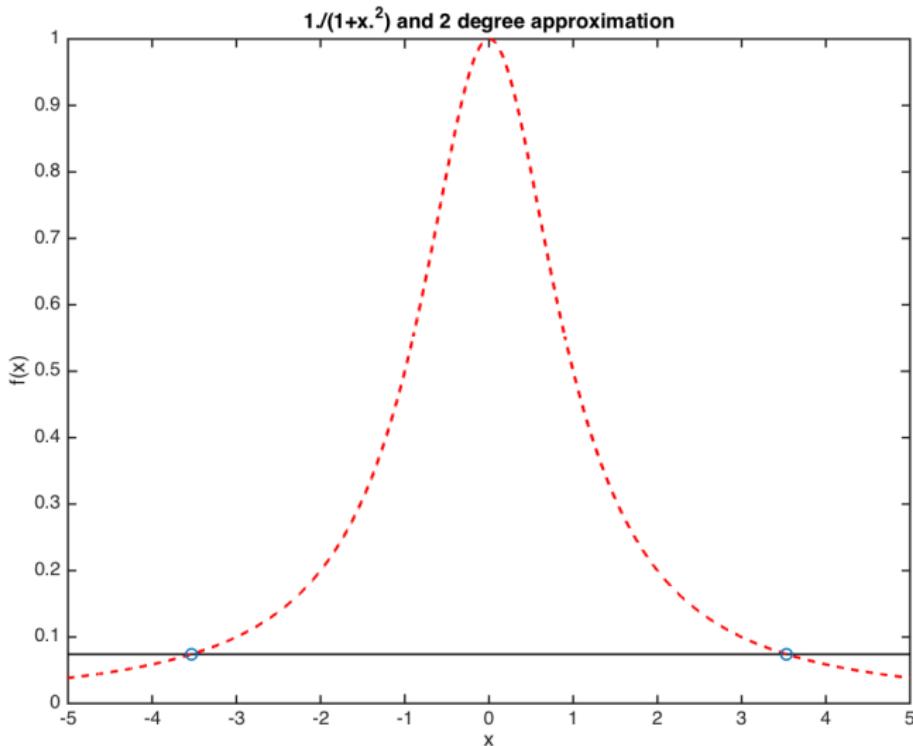
4. Calculate $y_j = f(x_j)$ at each x_j . This is the value of $g(z_j)$ at each of the corresponding nodes.
5. Given y_j , calculate the Chebychev coefficients c_j as:

$$c_j = \frac{2}{n+1} \sum_{k=0}^n y_j T_j(z_j)$$

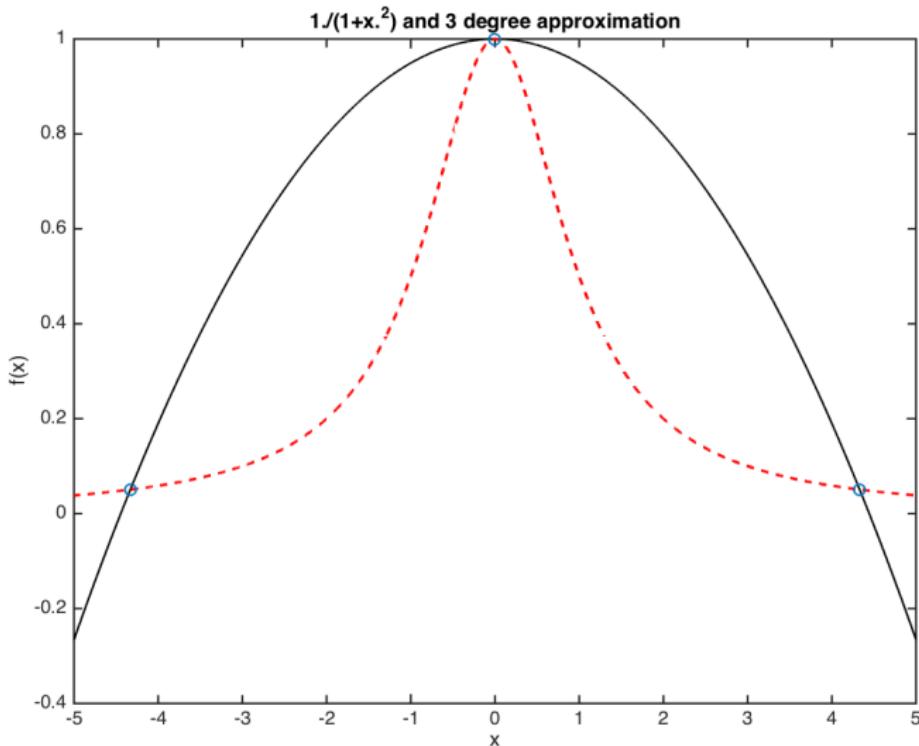
6. Your approximation is given by:

$$\hat{f}(x) = \sum_j c_j T_j(x) \approx f(x)$$

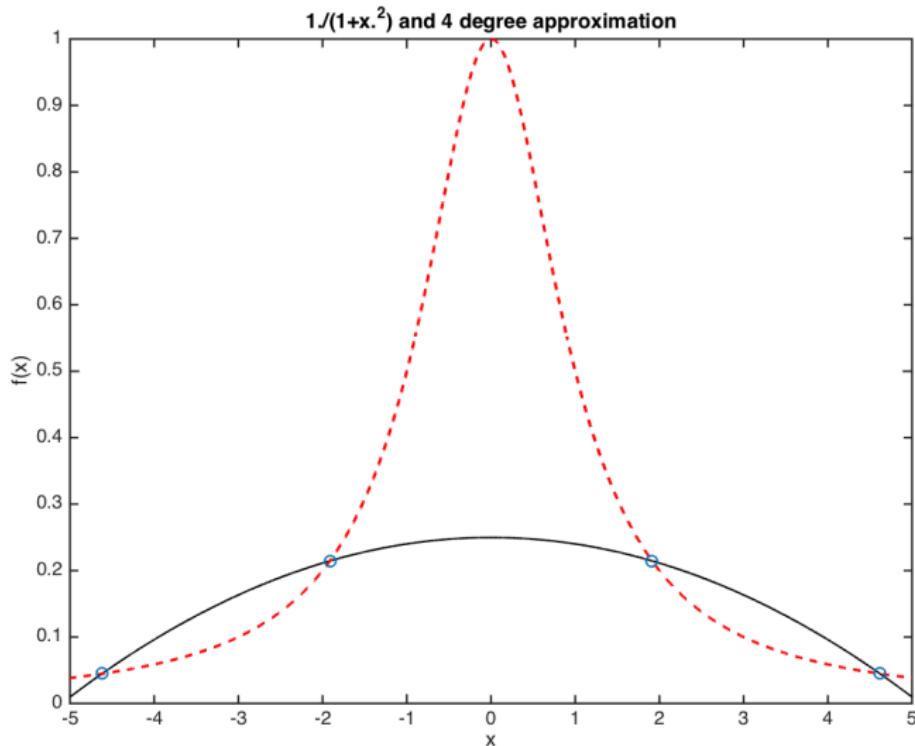
FIT: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX



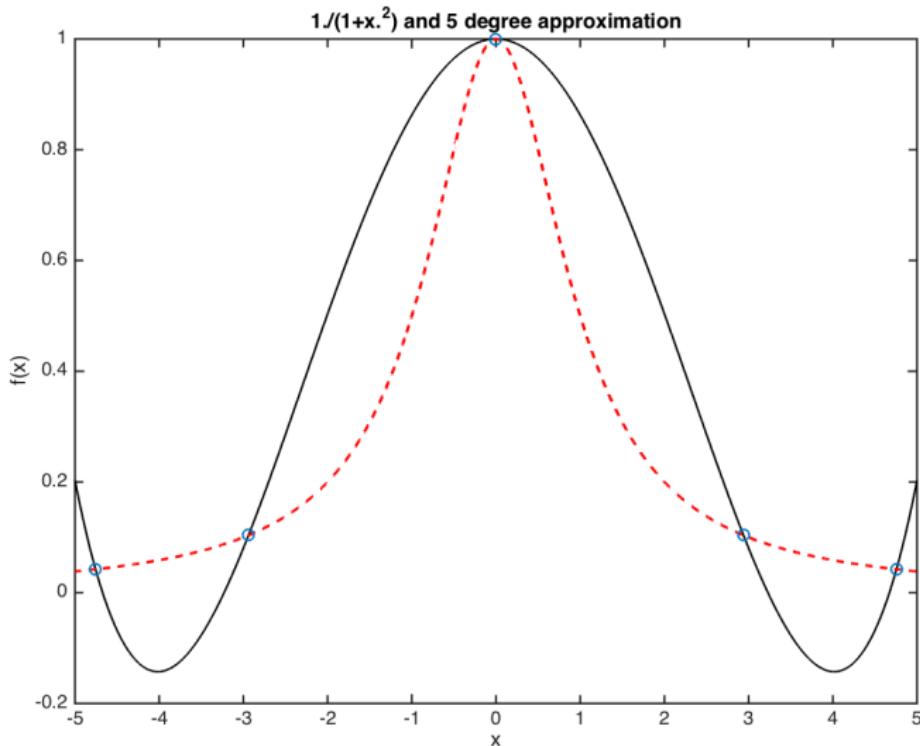
FIT: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX



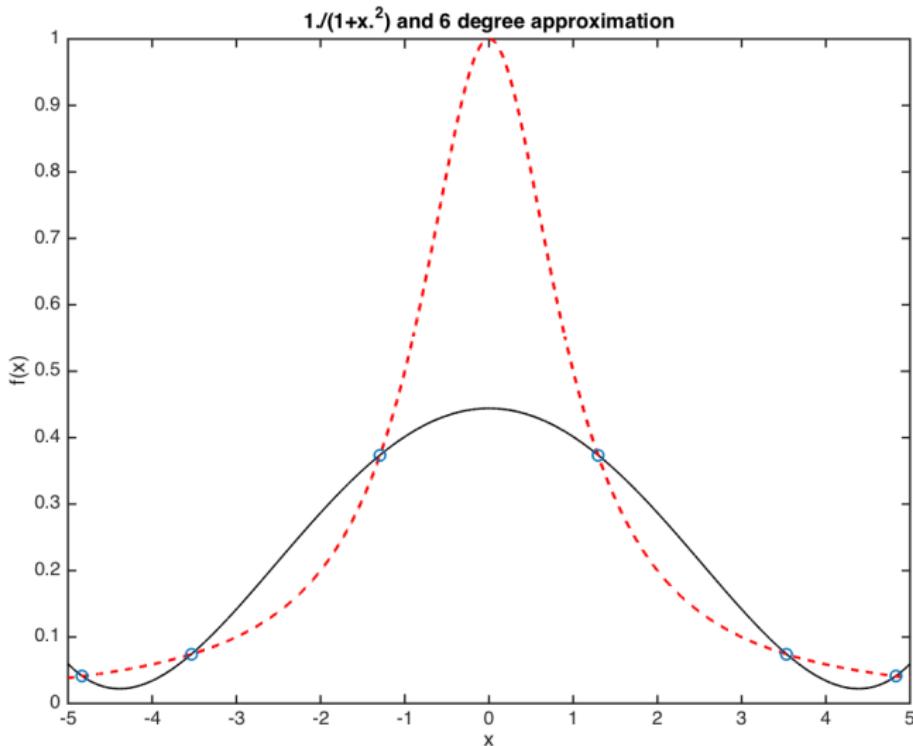
FIT: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX



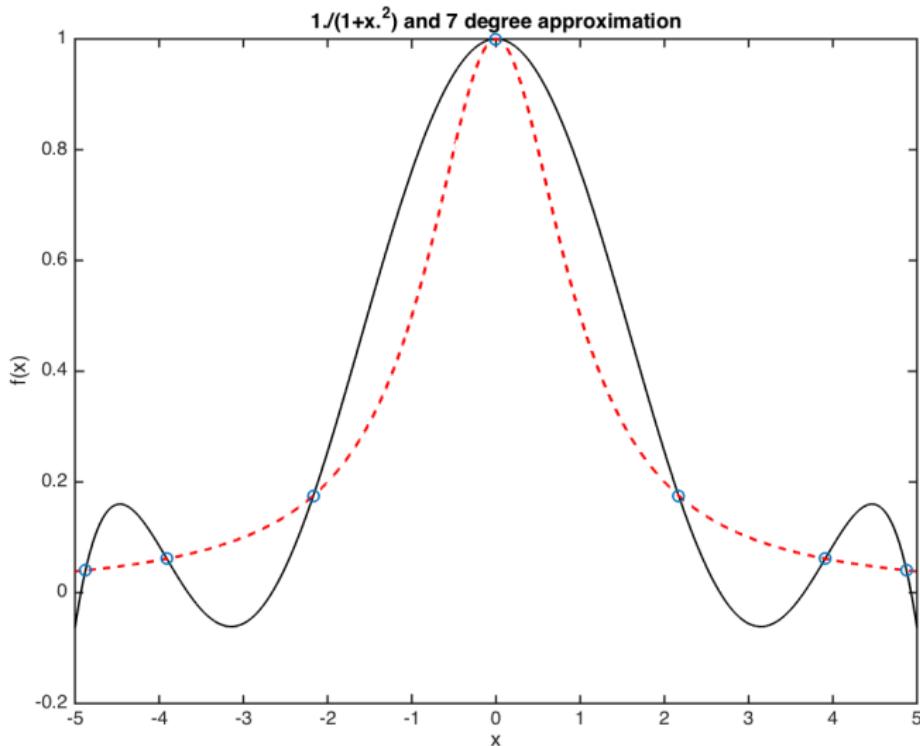
FIT: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX



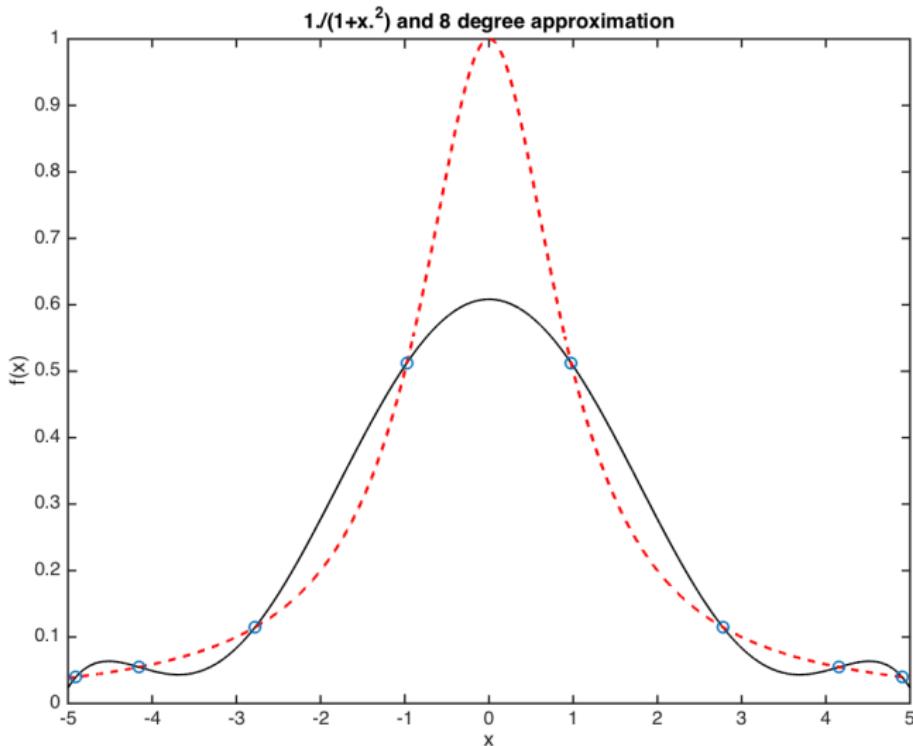
FIT: $f(x) = \frac{1}{1+x^2}$: 6-DEG. APPROX



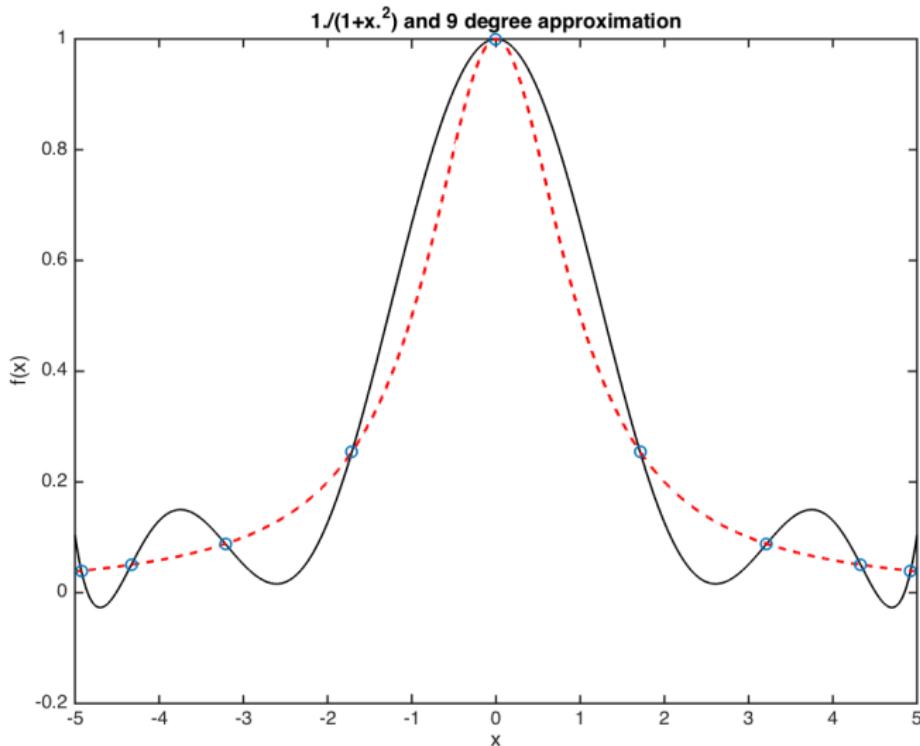
FIT: $f(x) = \frac{1}{1+x^2}$: 7-DEG. APPROX



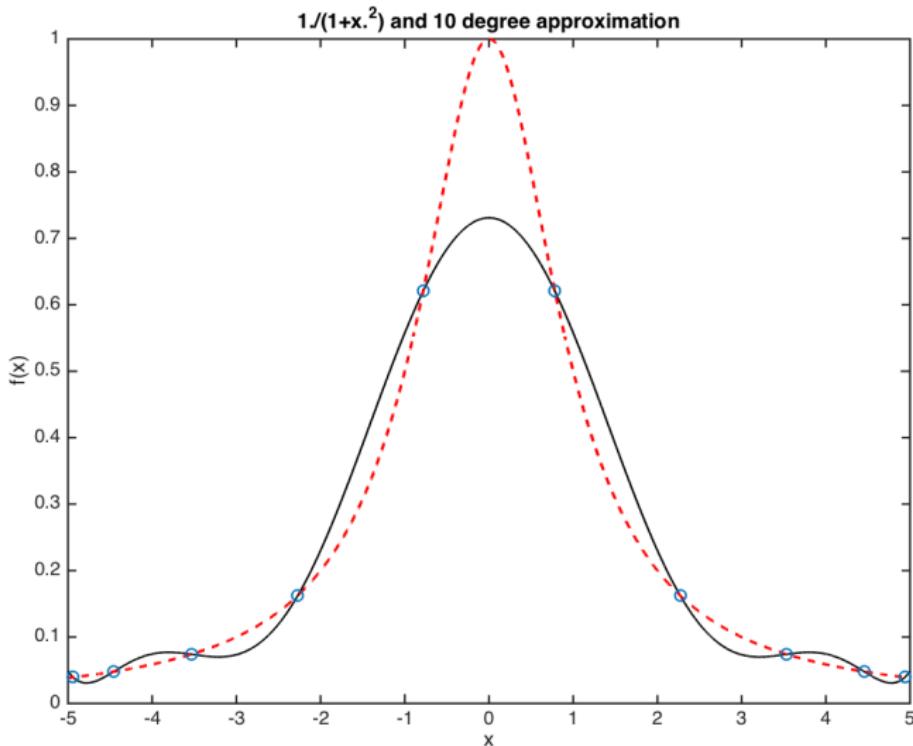
FIT: $f(x) = \frac{1}{1+x^2}$: 8-DEG. APPROX



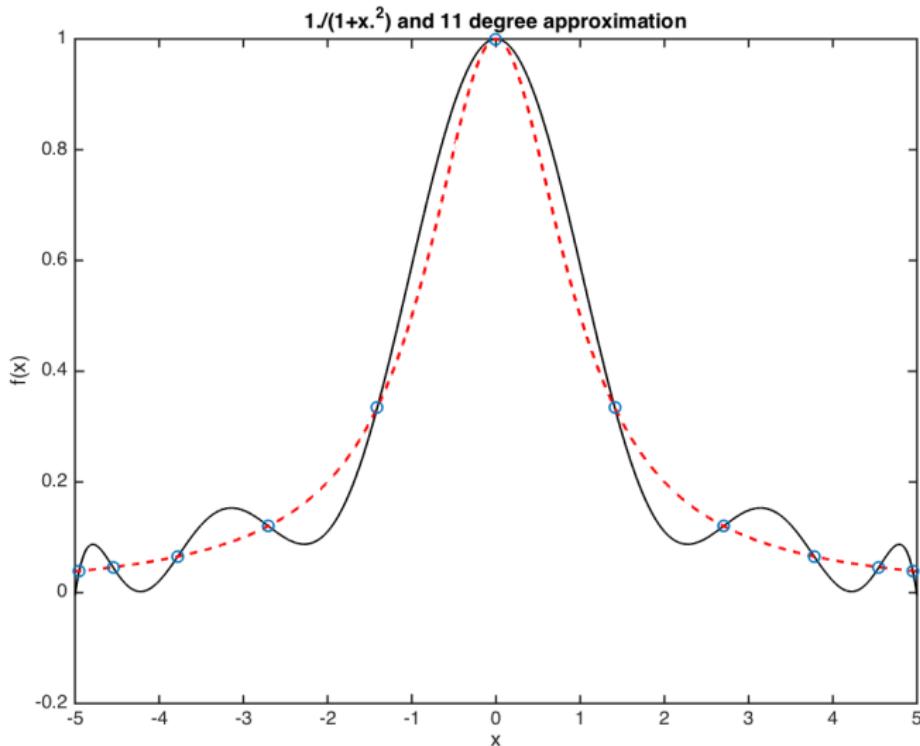
FIT: $f(x) = \frac{1}{1+x^2}$: 9-DEG. APPROX



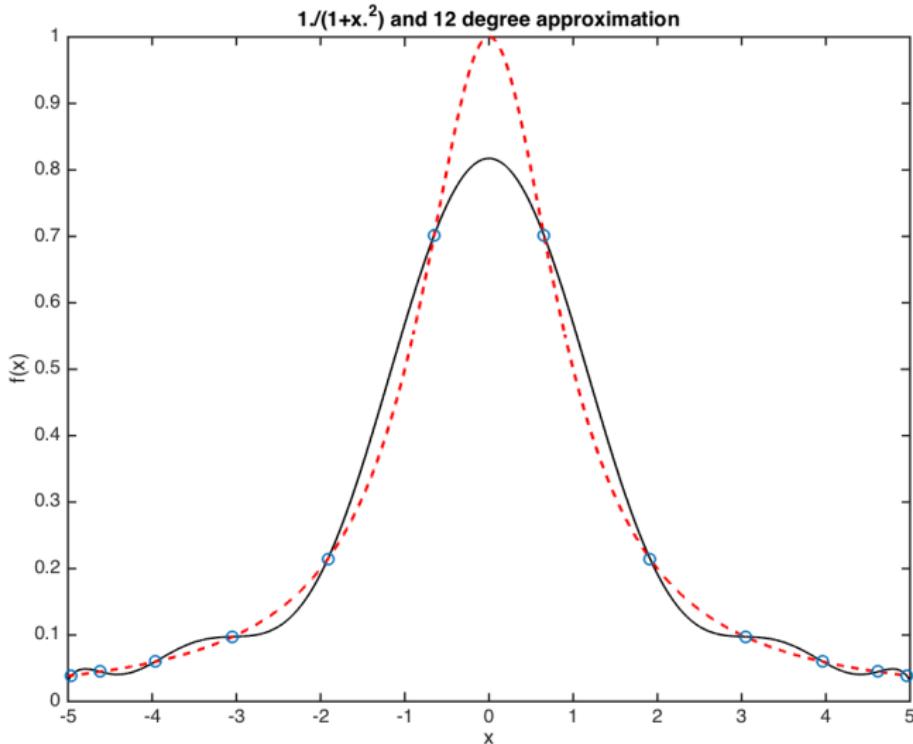
FIT: $f(x) = \frac{1}{1+x^2}$: 10-DEG. APPROX



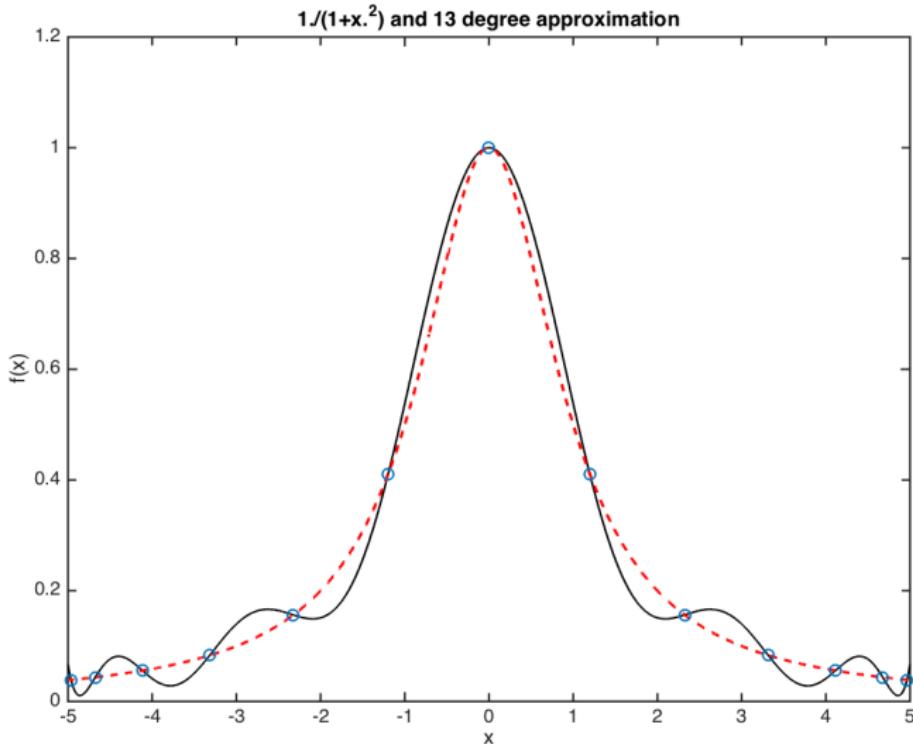
FIT: $f(x) = \frac{1}{1+x^2}$: 11-DEG. APPROX



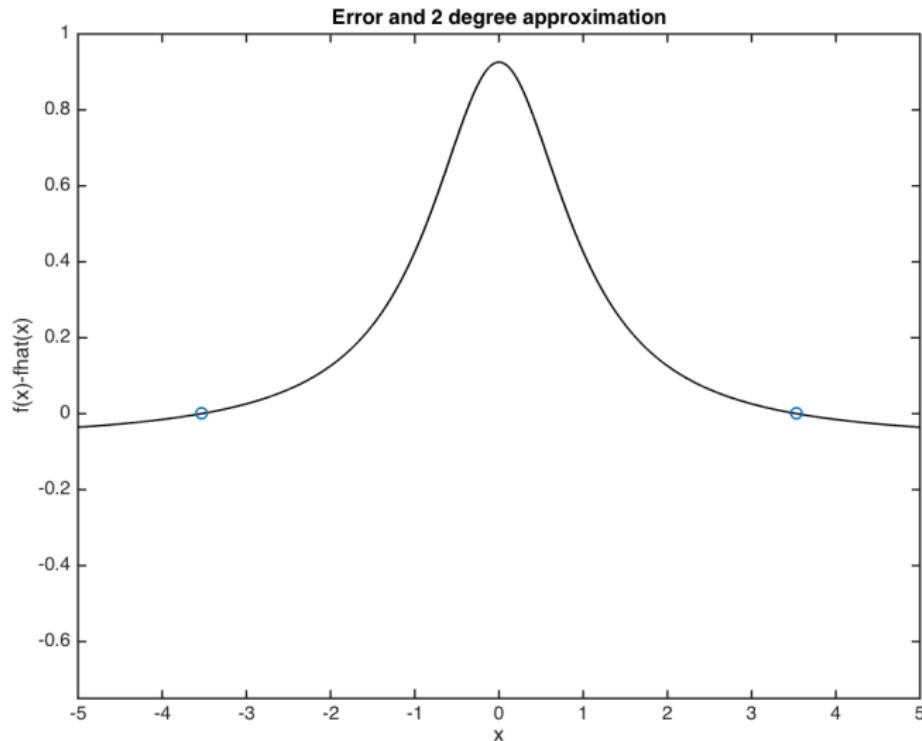
FIT: $f(x) = \frac{1}{1+x^2}$: 12-DEG. APPROX



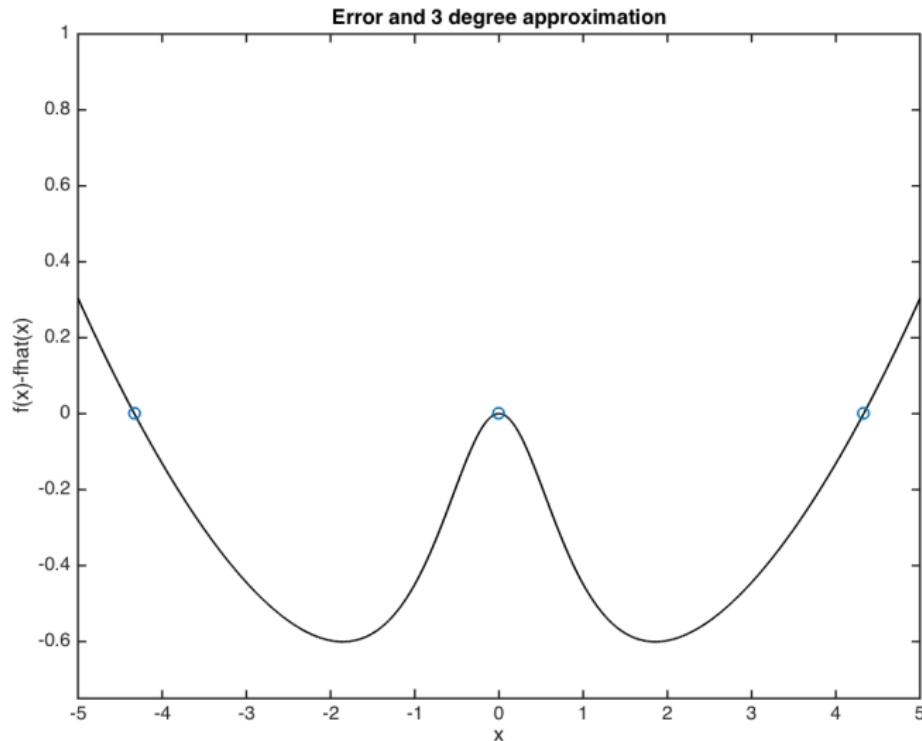
FIT: $f(x) = \frac{1}{1+x^2}$: 13-DEG. APPROX



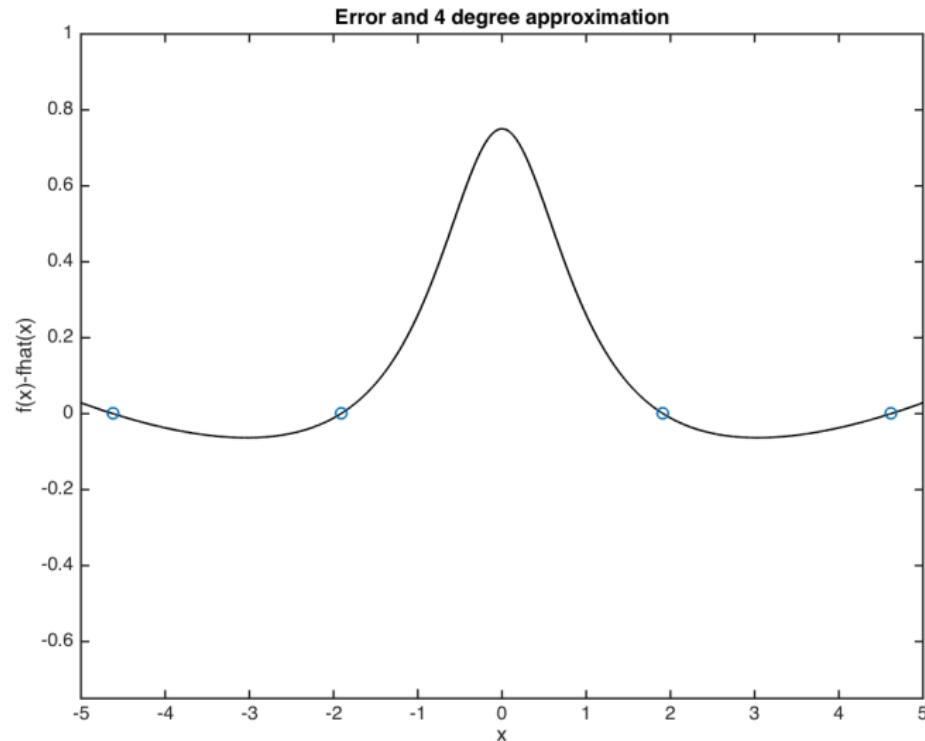
ERRORS: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX



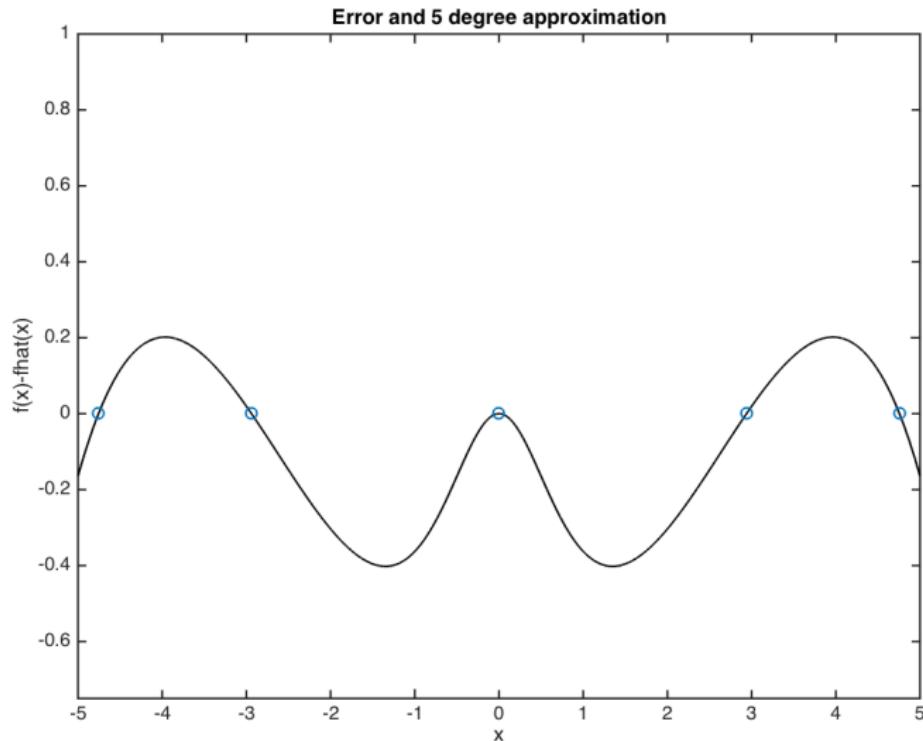
ERRORS: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX



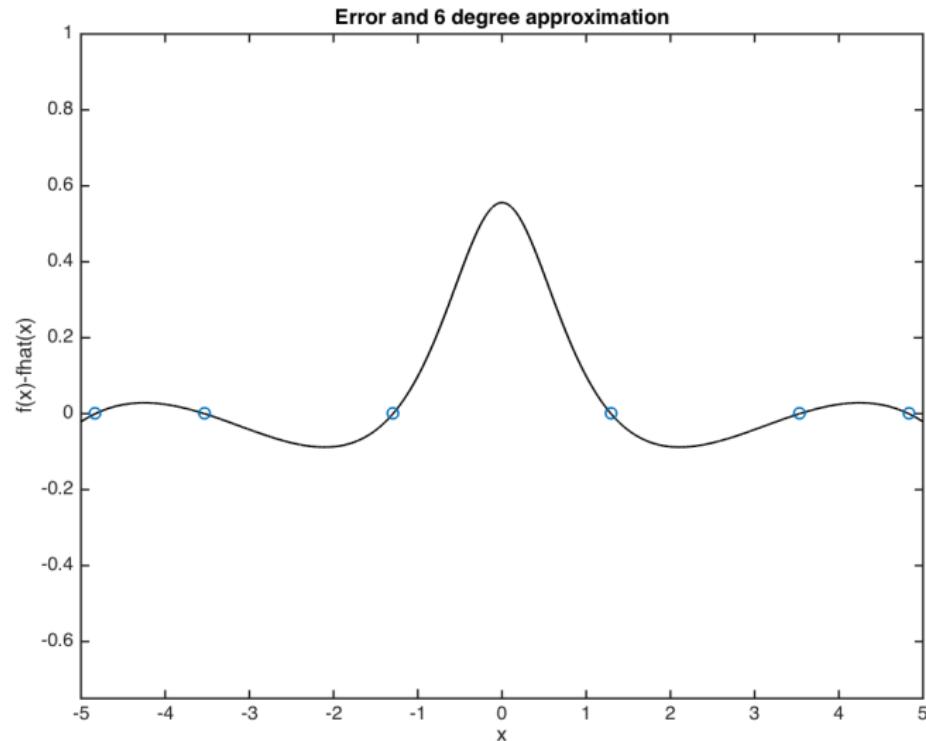
ERRORS: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX



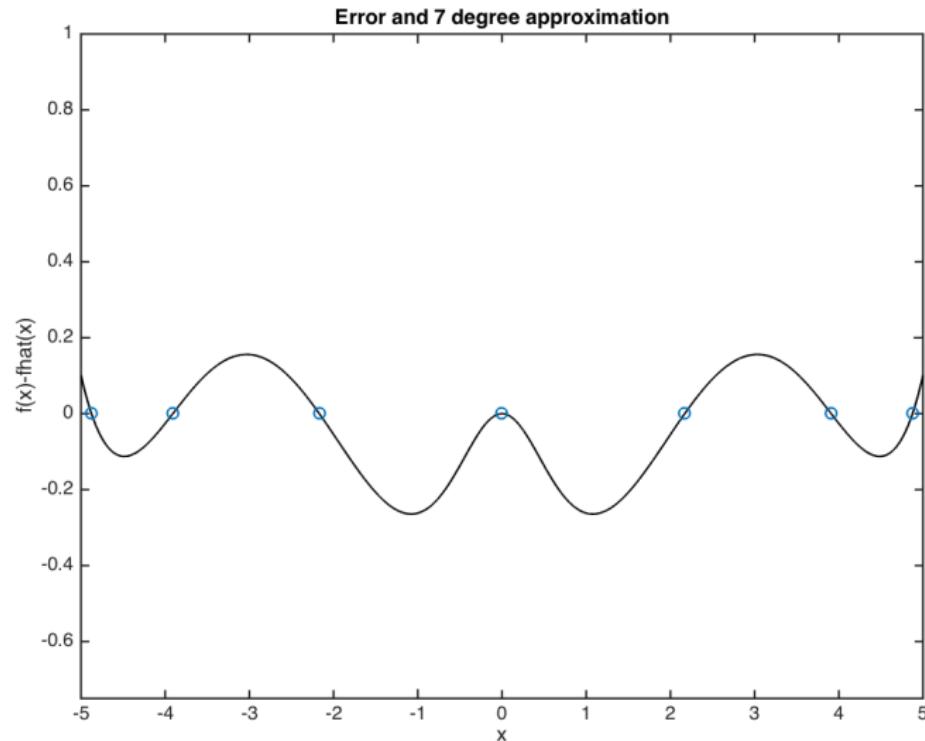
ERRORS: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX



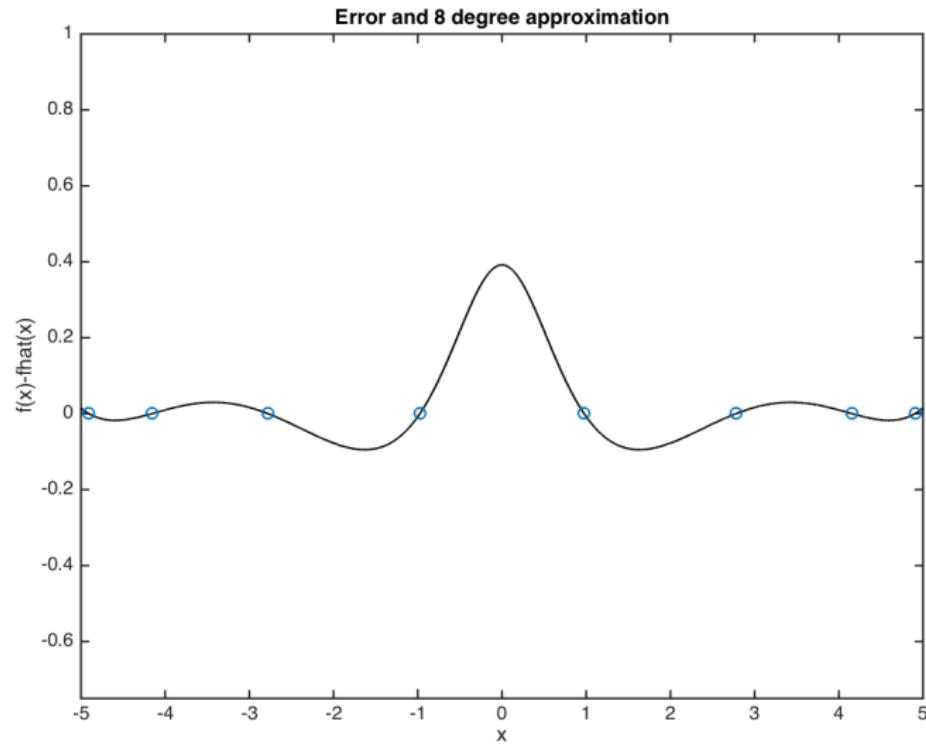
ERRORS: $f(x) = \frac{1}{1+x^2}$: 6-DEG. APPROX



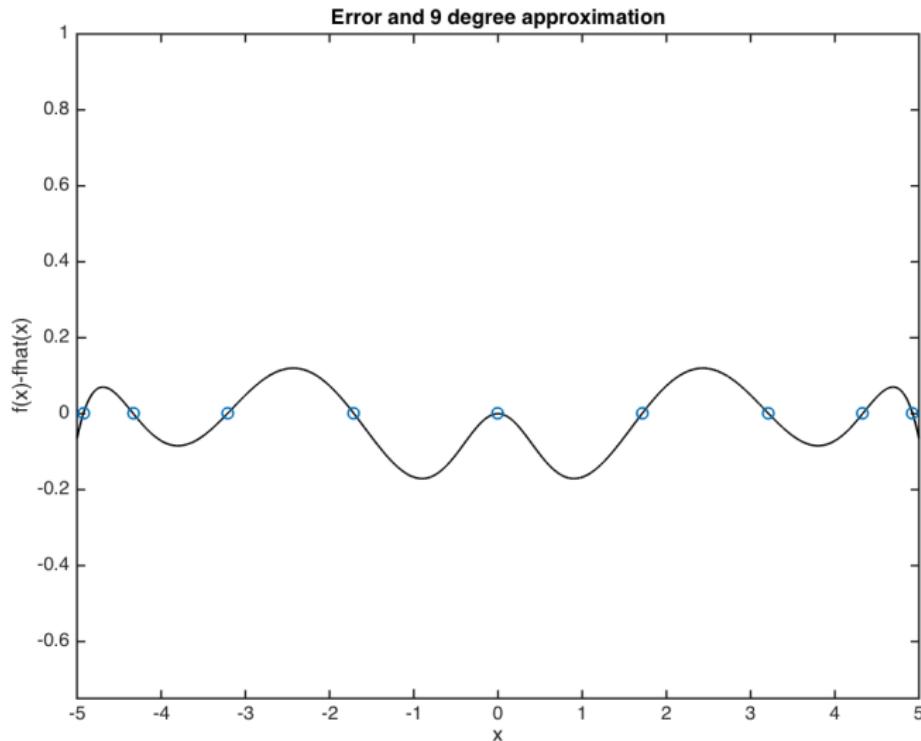
ERRORS: $f(x) = \frac{1}{1+x^2}$: 7-DEG. APPROX



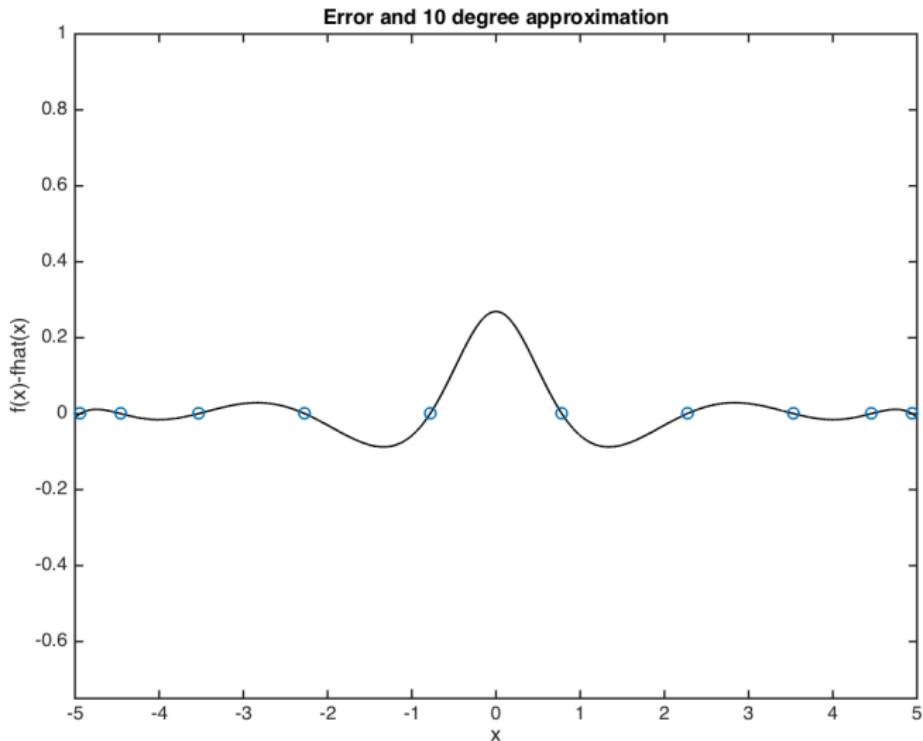
ERRORS: $f(x) = \frac{1}{1+x^2}$: 8-DEG. APPROX



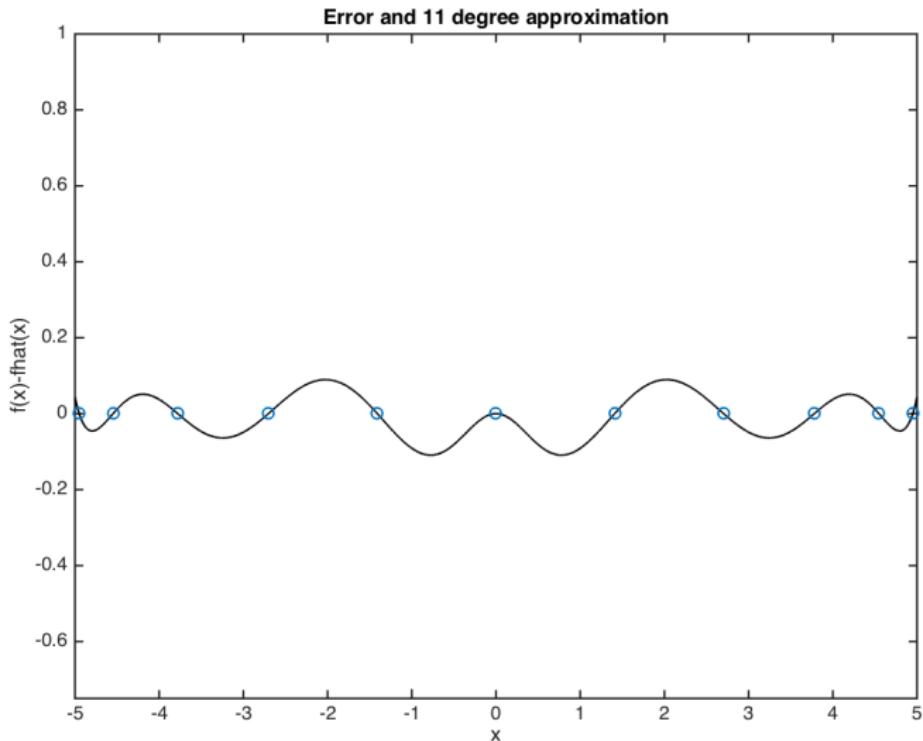
ERRORS: $f(x) = \frac{1}{1+x^2}$: 9-DEG. APPROX



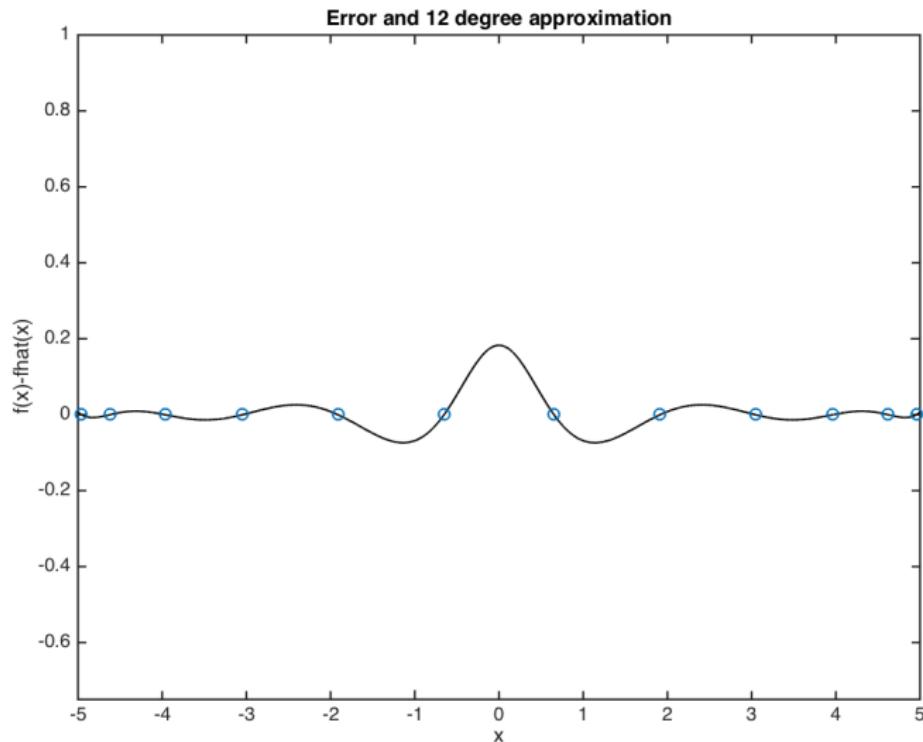
ERRORS: $f(x) = \frac{1}{1+x^2}$: 10-DEG. APPROX



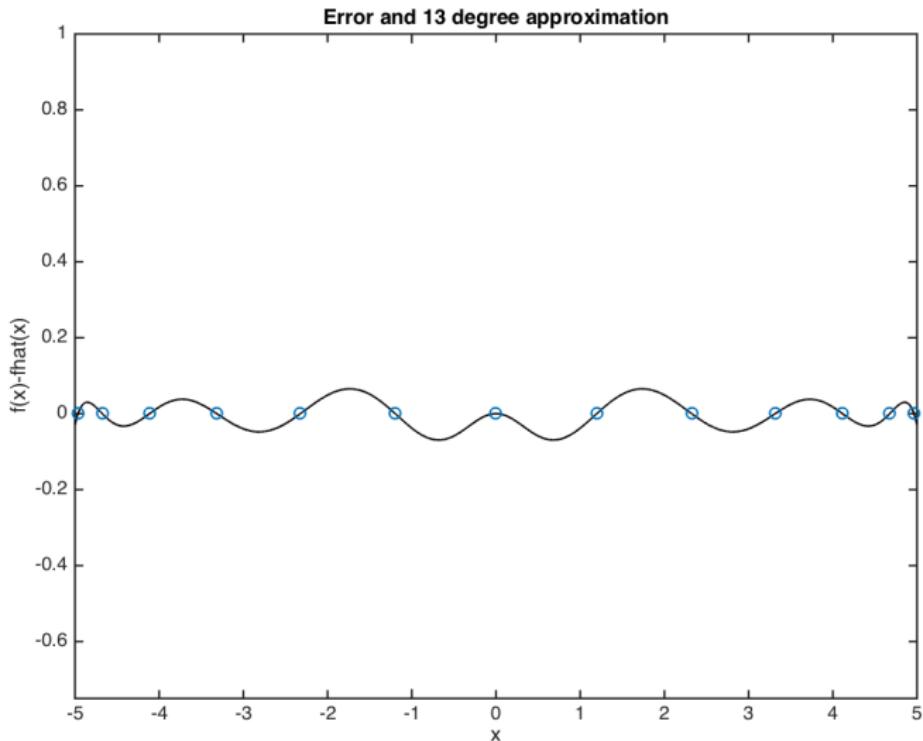
ERRORS: $f(x) = \frac{1}{1+x^2}$: 11-DEG. APPROX



ERRORS: $f(x) = \frac{1}{1+x^2}$: 12-DEG. APPROX



ERRORS: $f(x) = \frac{1}{1+x^2}$: 13-DEG. APPROX



INTERPOLATION: 2-D

1. Pick fcn $f(x, y)$ deg. n , bounds a, b and c, d for interpolation.
2. Find Chebychev nodes for $f(x, y)$ in " $z^{[1]}$ " and " $z^{[2]}$ " space.

$$z_j^{[1]} = \cos\left(\frac{2j-1}{2n}\pi\right), \quad z_j^{[2]} = \cos\left(\frac{2j-1}{2n}\pi\right), \quad j = 1, \dots, n$$

3. Convert nodes into "x" space:

$$x_j = a + \frac{(b-a)(z_j^{[1]} + 1)}{2}, \quad y_j = c + \frac{(d-c)(z_j^{[2]} + 1)}{2}$$

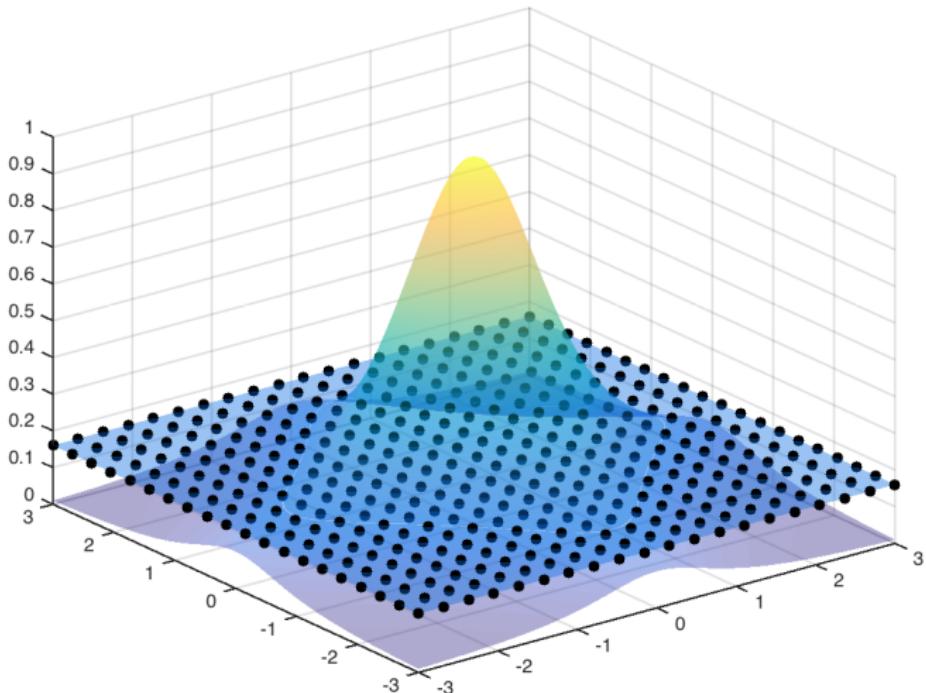
4. Calculate $\xi_{i,j} = f(x_i, y_j)$ for all points.
5. Given y_j , calculate the Chebychev coefficients c_j as:

$$c_{i,j} = \frac{\sum_{k=1}^n \sum_{l=1}^n \xi_{i,j} T_i(z_k^{[1]}) T_j(z_l^{[2]})}{\sum_{k=1}^n T_i(z_k^{[1]}) T_j(z_k^{[2]})}$$

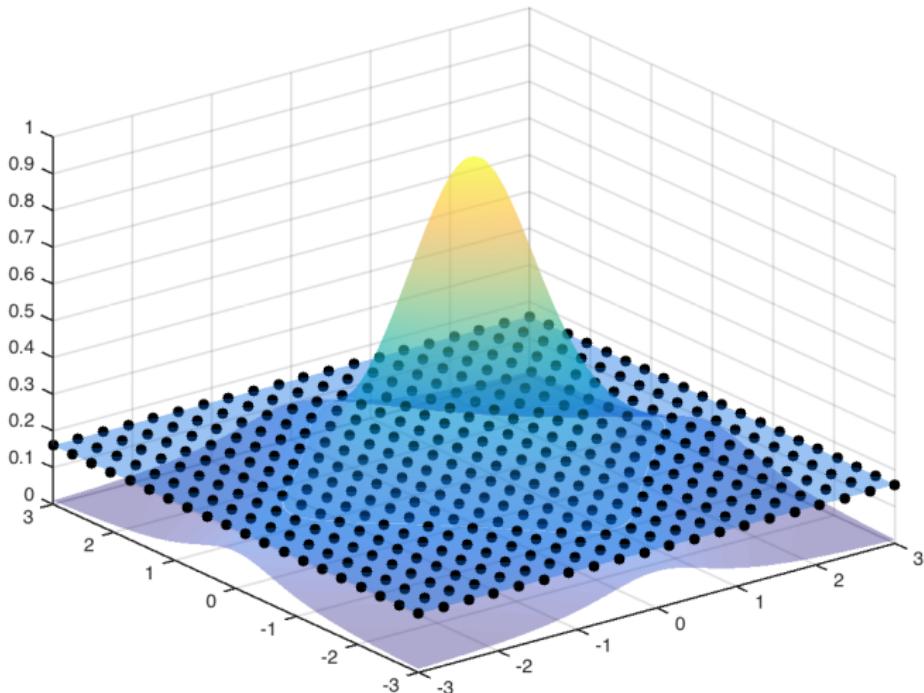
6. Approximation:

$$\hat{f}(x, y) \approx \sum_{i=1}^n \sum_{j=1}^n c_{ij} T_i\left(2\frac{x-a}{b-a} - 1\right) T_j\left(2\frac{y-c}{d-b} - 1\right)$$

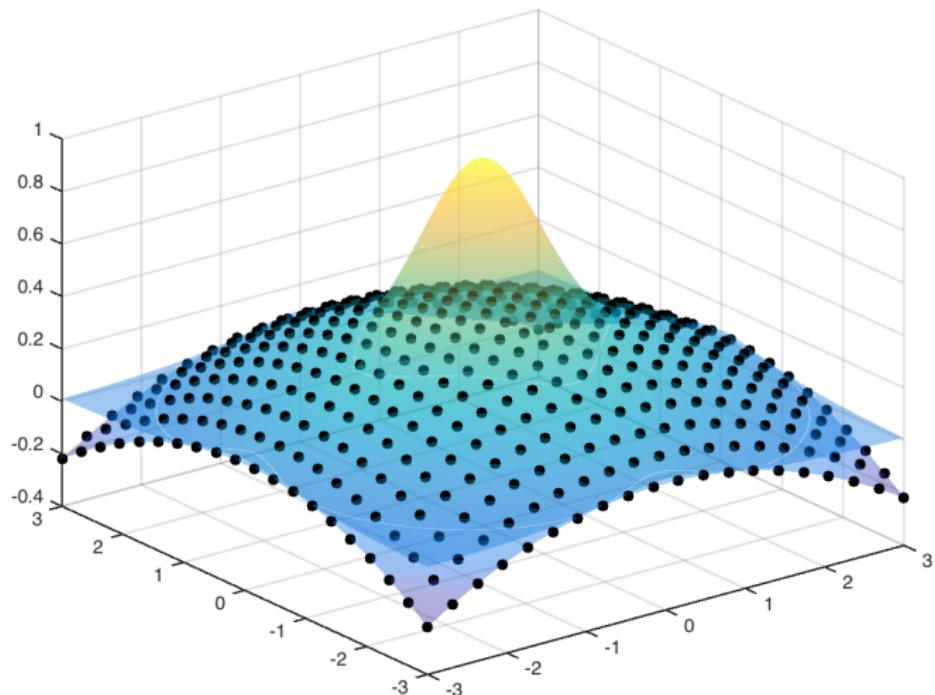
MONOM FIT: $f(x) = \frac{1}{1+x^2}$: 1-DEG. APPROX



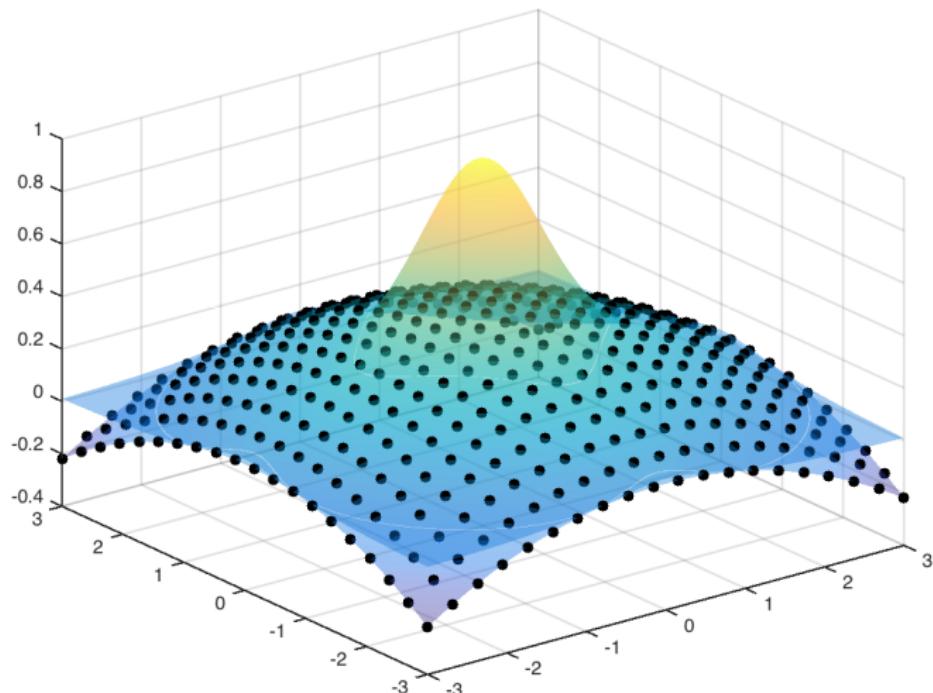
MONOM FIT: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX



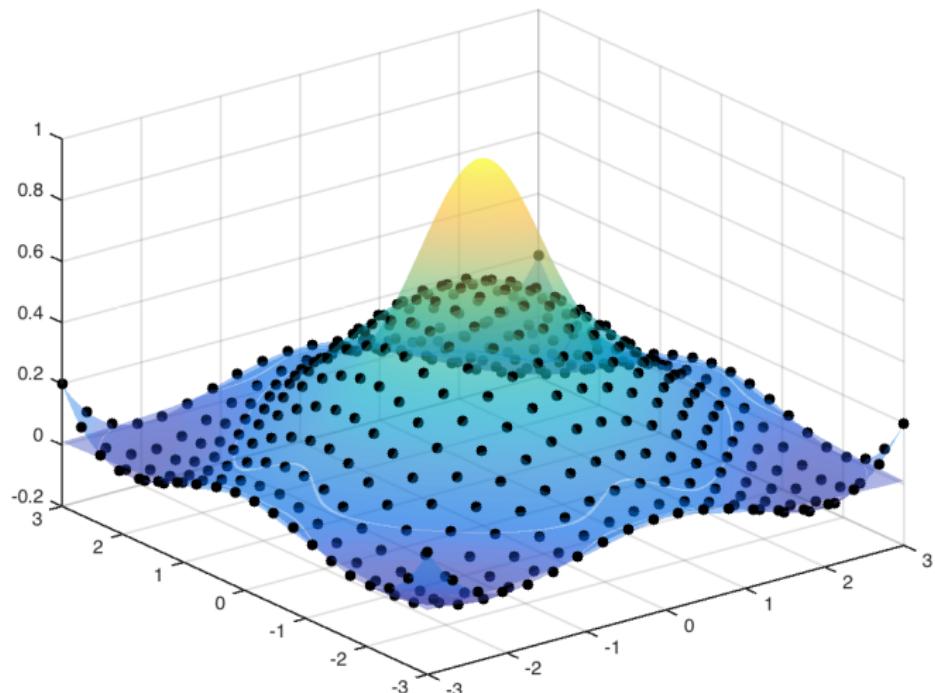
MONOM FIT: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX



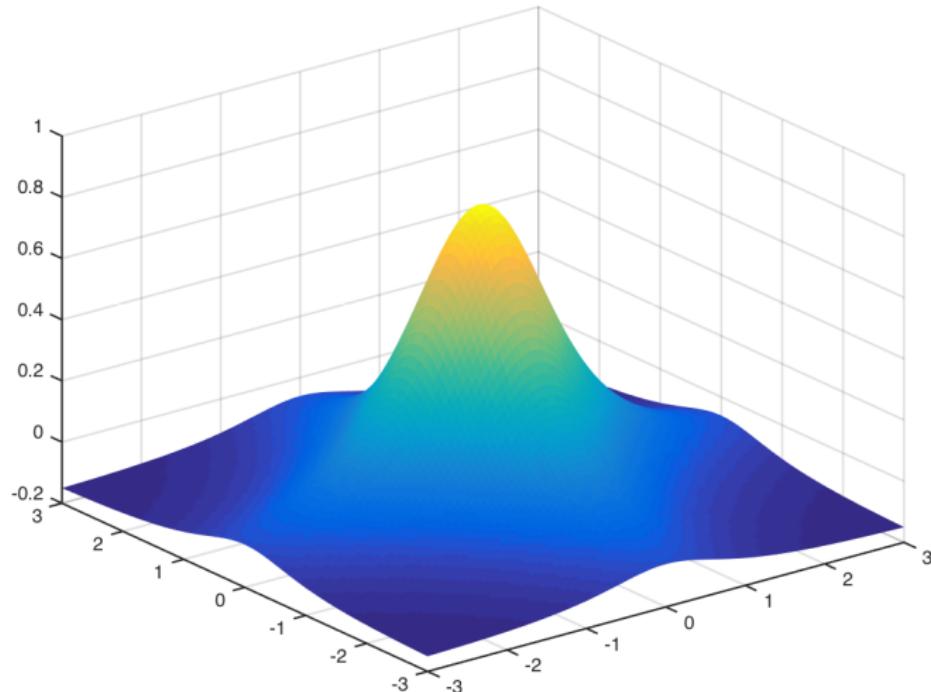
MONOM FIT: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX



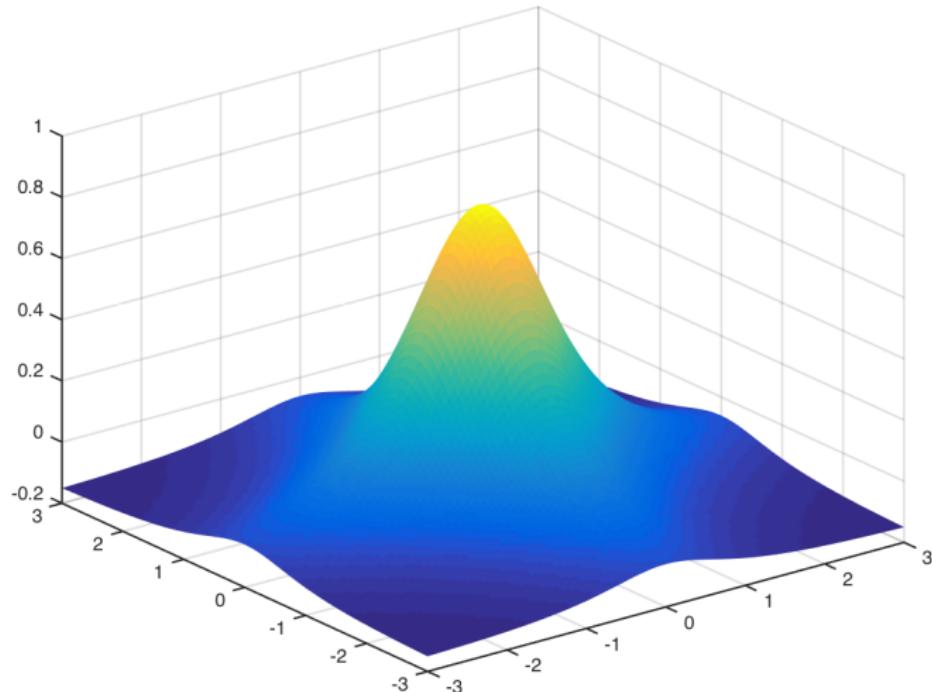
MONOM FIT: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX



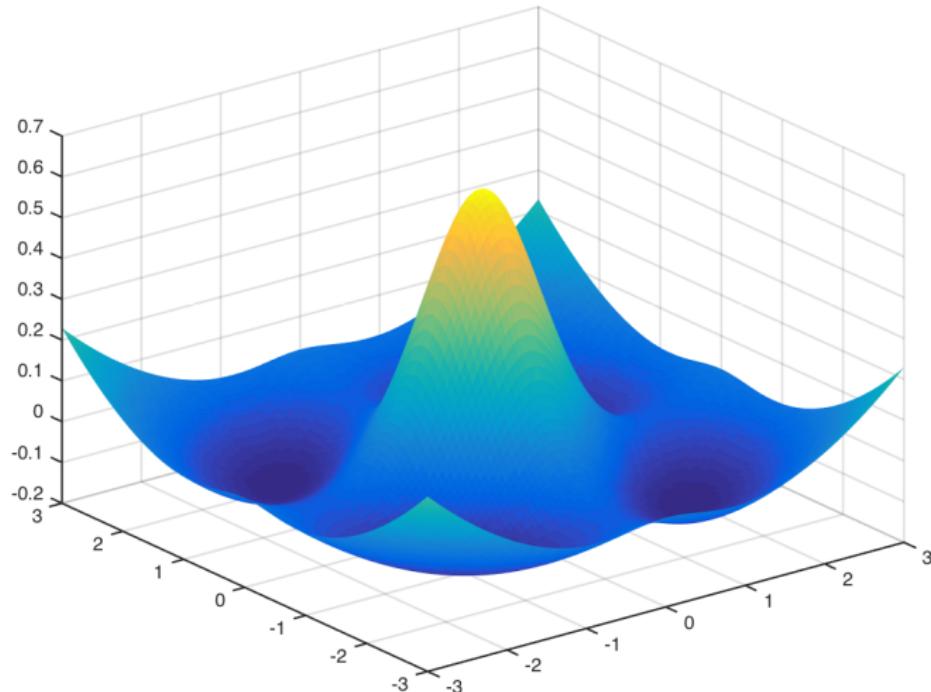
MONOM ERRORS: $f(x) = \frac{1}{1+x^2}$: 1-DEG. APPROX



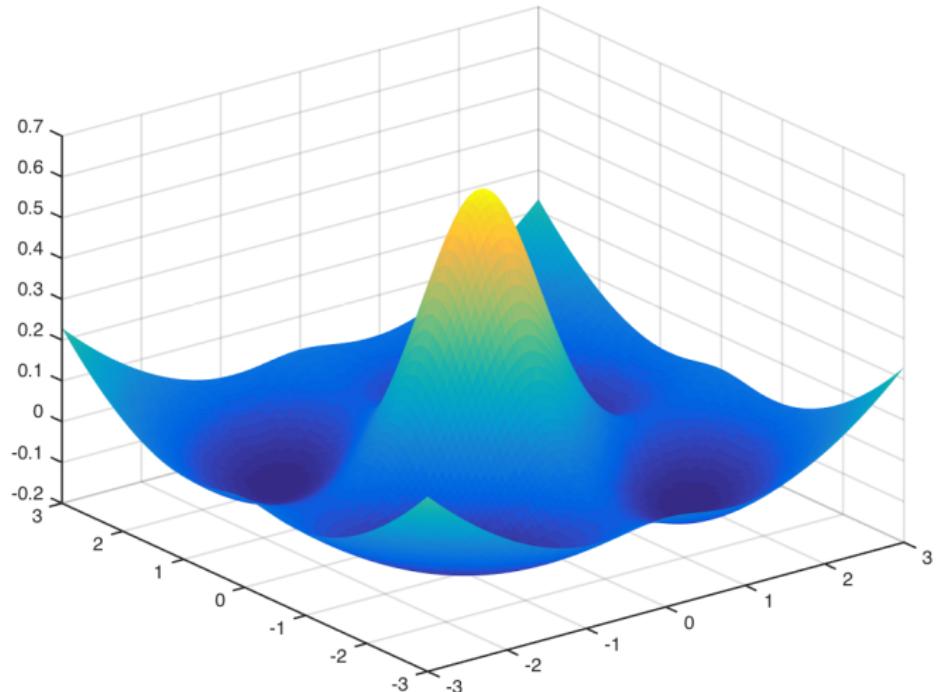
MONOM ERRORS: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX



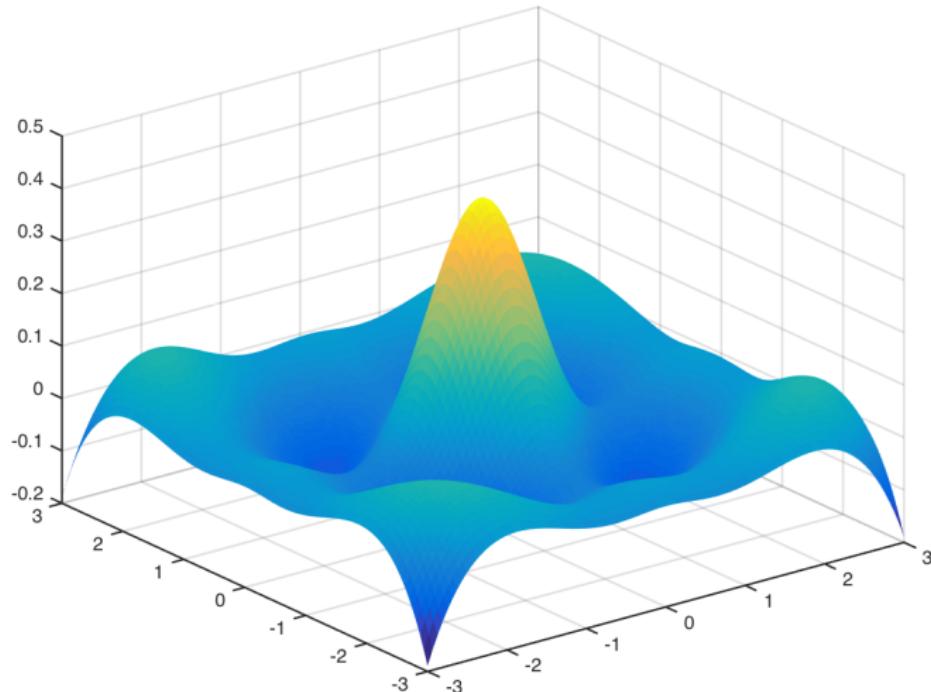
MONOM ERRORS: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX



MONOM ERRORS: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX

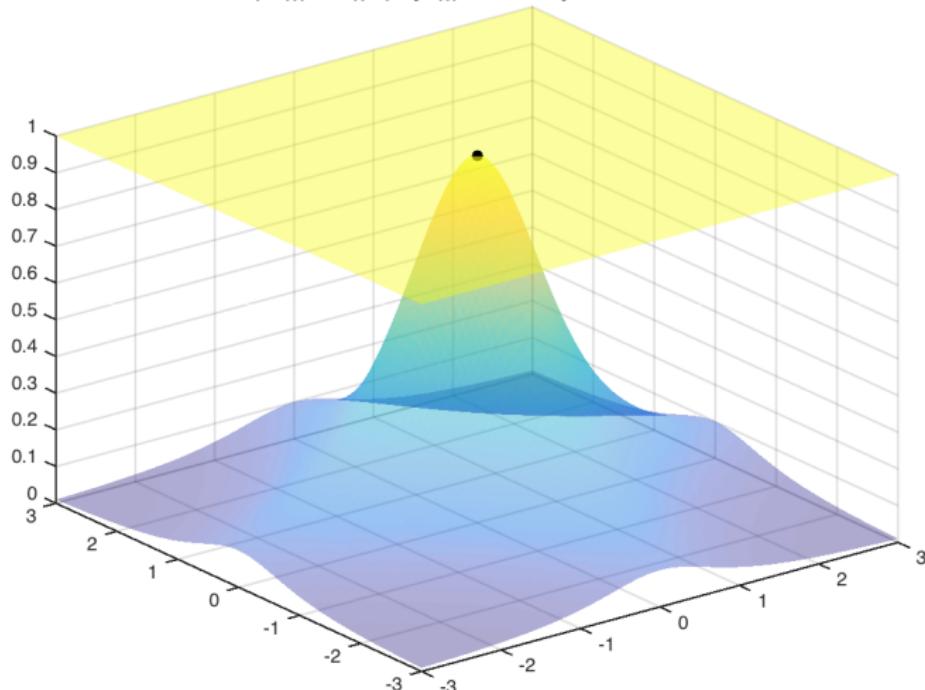


MONOM ERRORS: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX



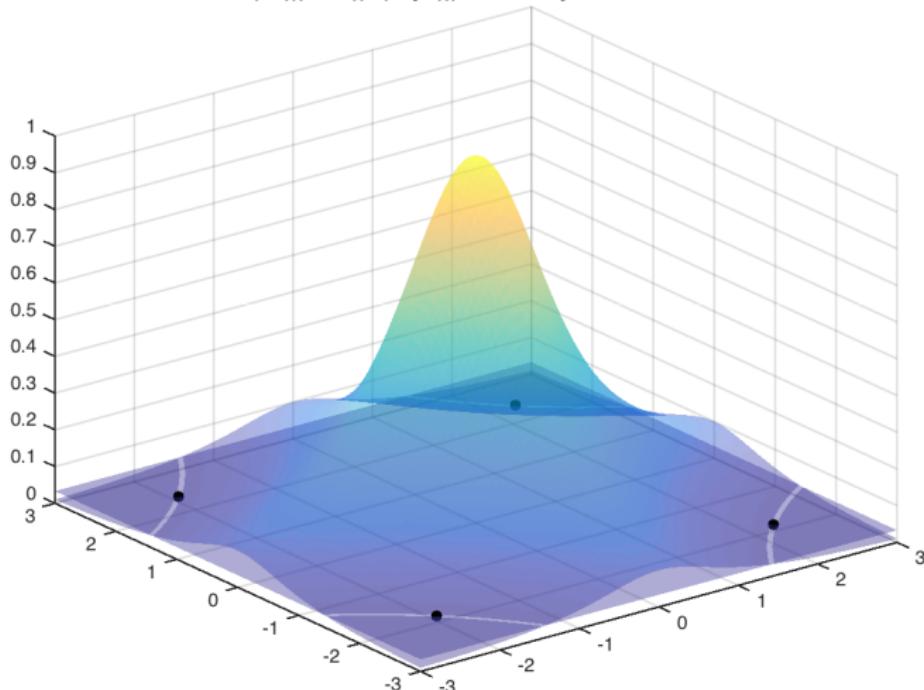
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 1-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=1



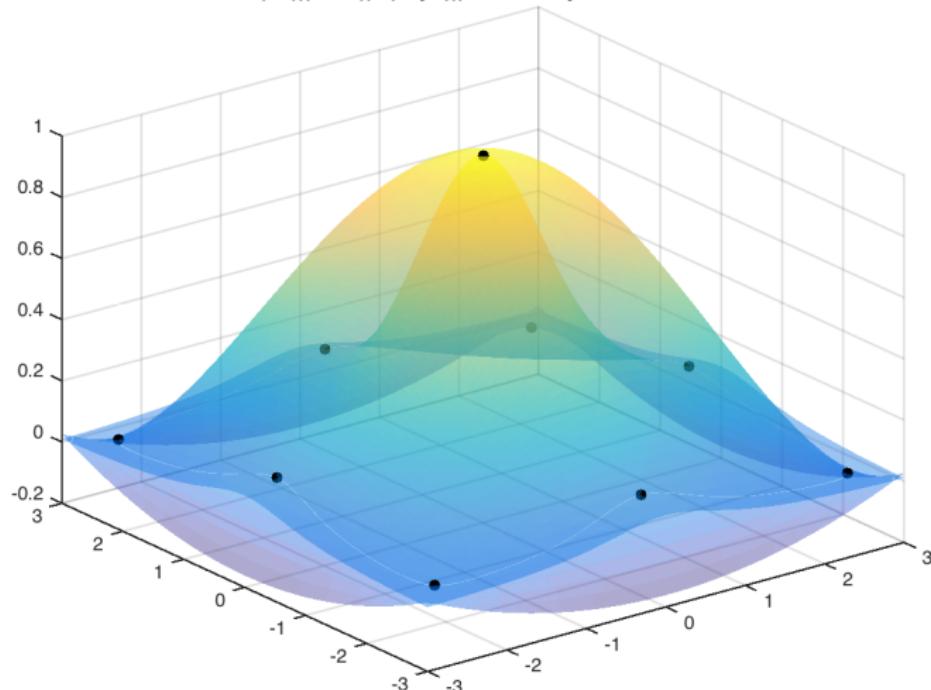
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=2



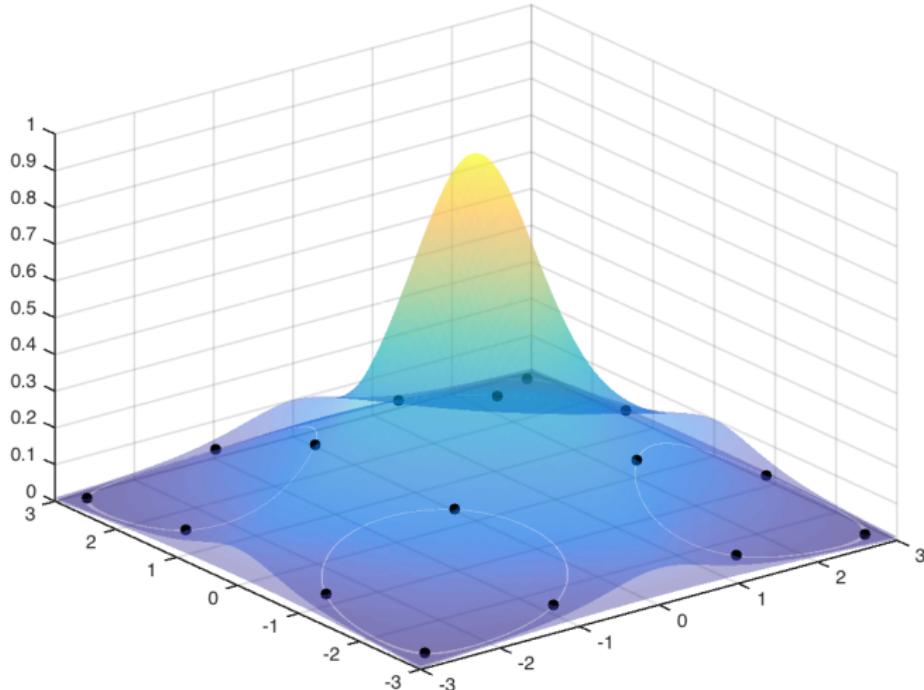
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=3



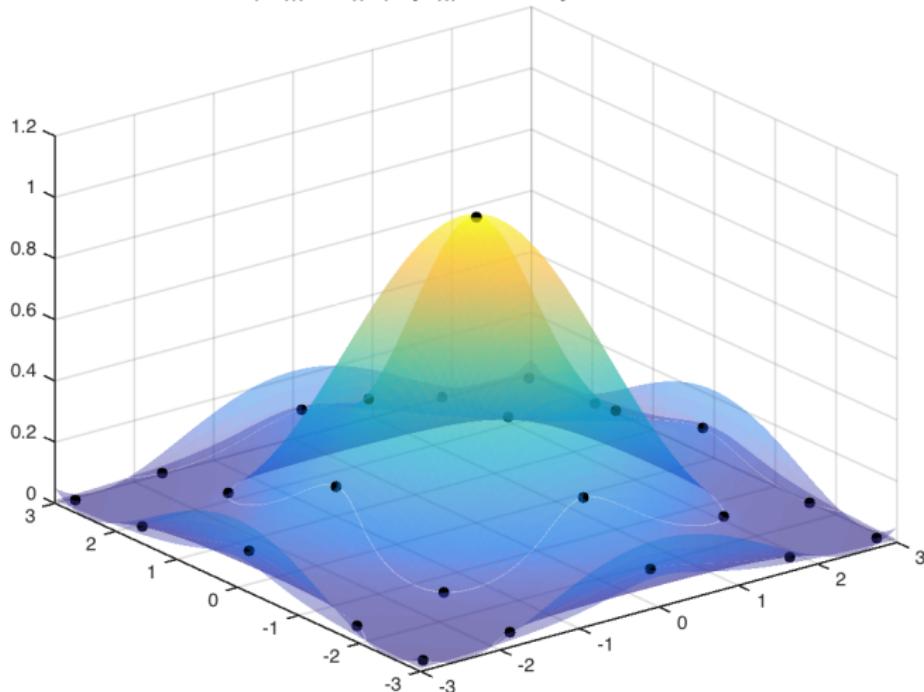
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=4



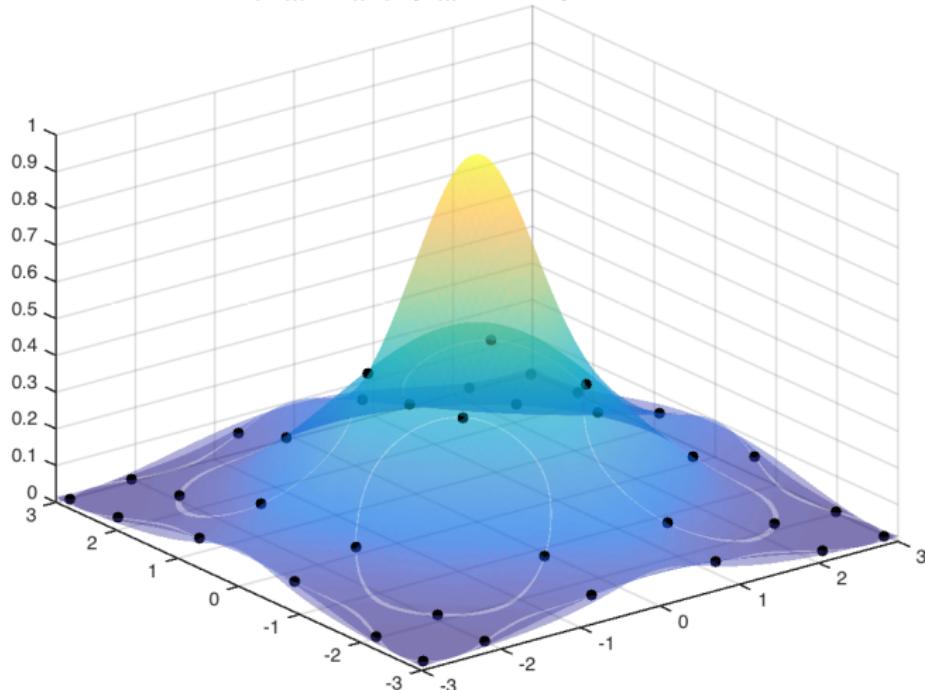
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=5



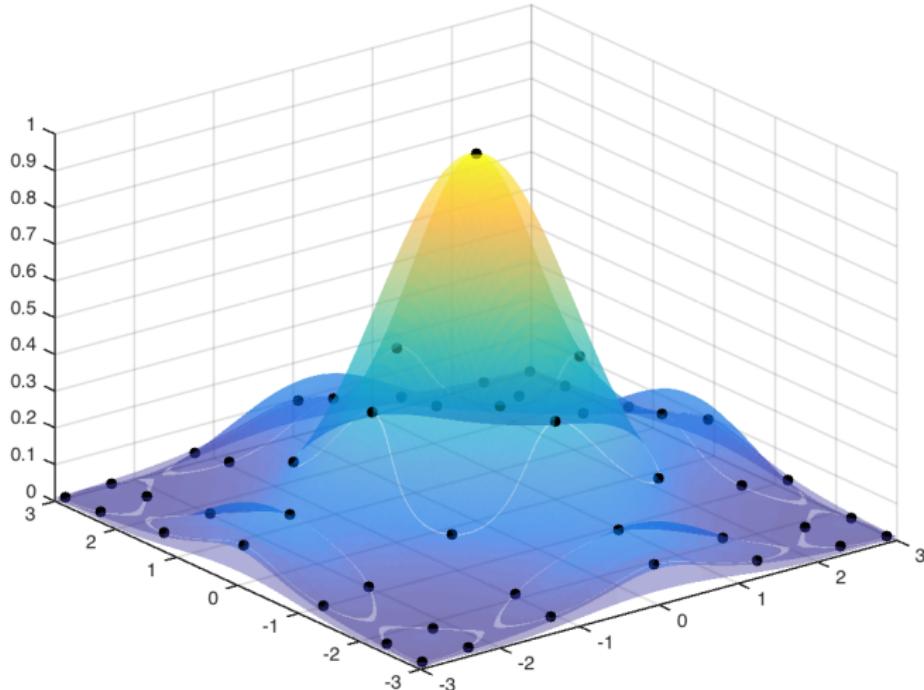
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 6-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=6



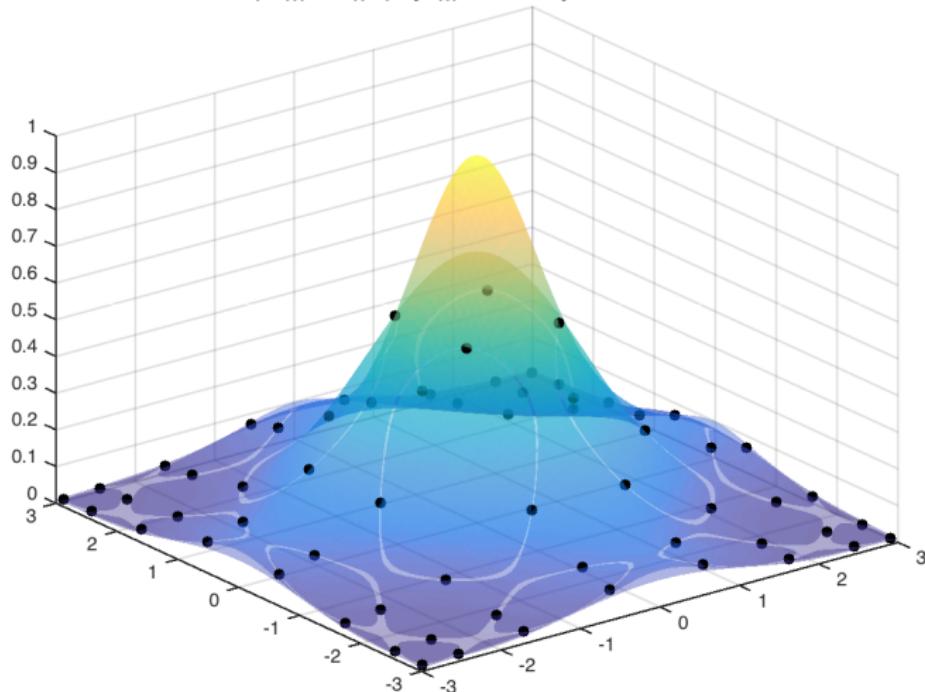
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 7-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=7



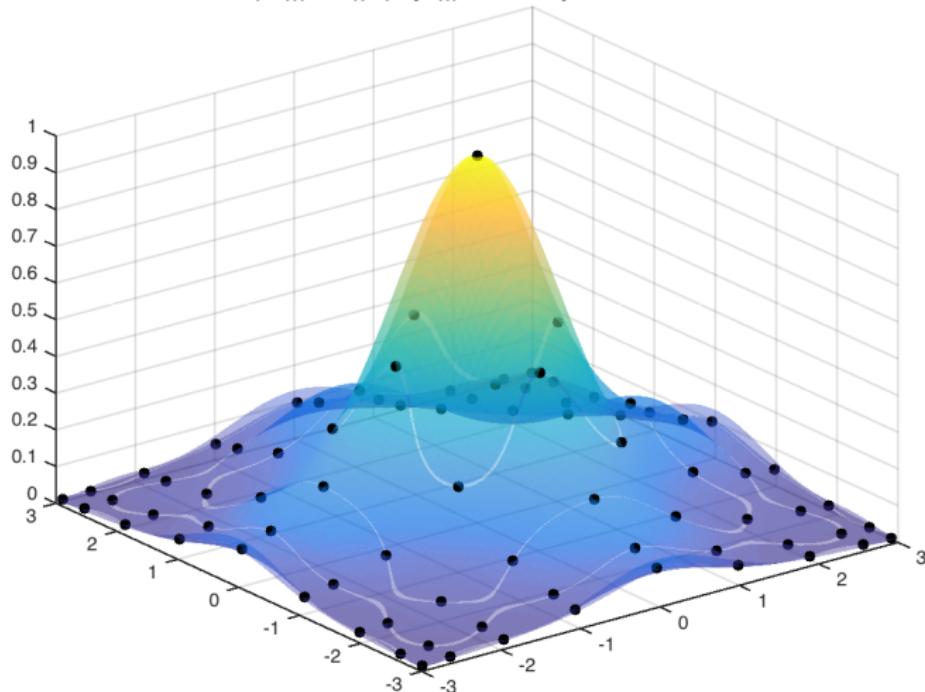
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 8-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=8



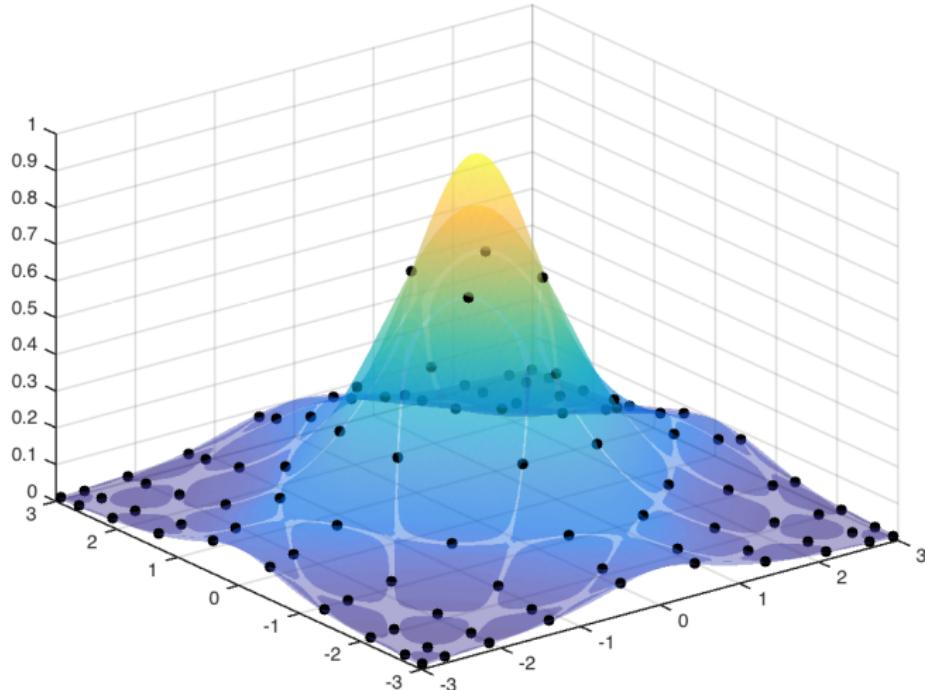
CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 9-DEG. APPROX

$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=9

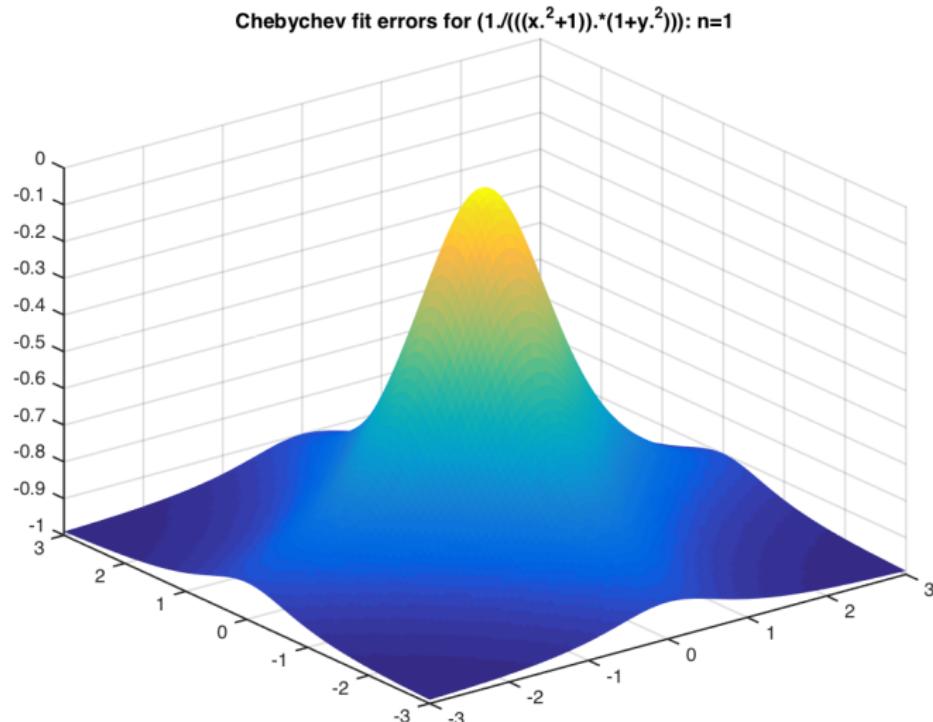


CHEBY FIT: $f(x) = \frac{1}{1+x^2}$: 10-DEG. APPROX

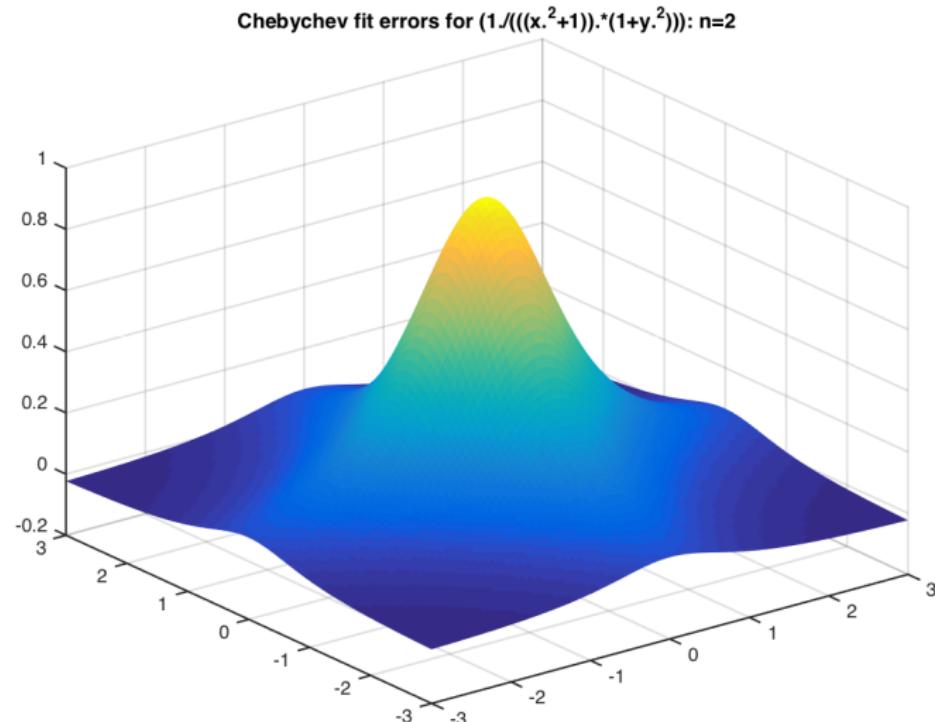
$(1./(((x.^2+1)).*(1+y.^2)))$ and Chebychev fit: n=10



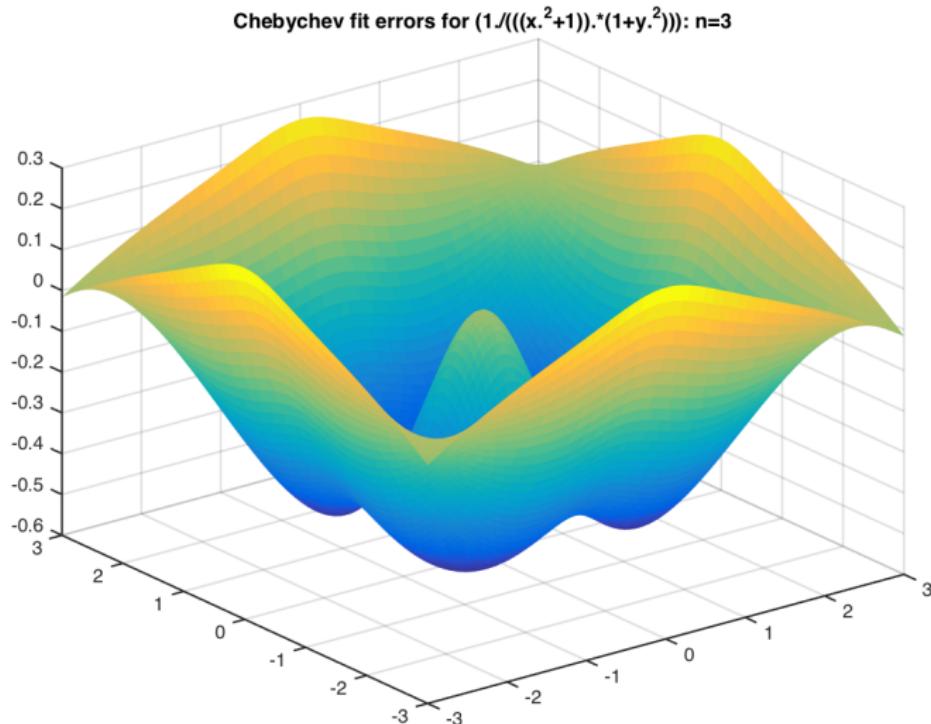
CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 1-DEG. APPROX



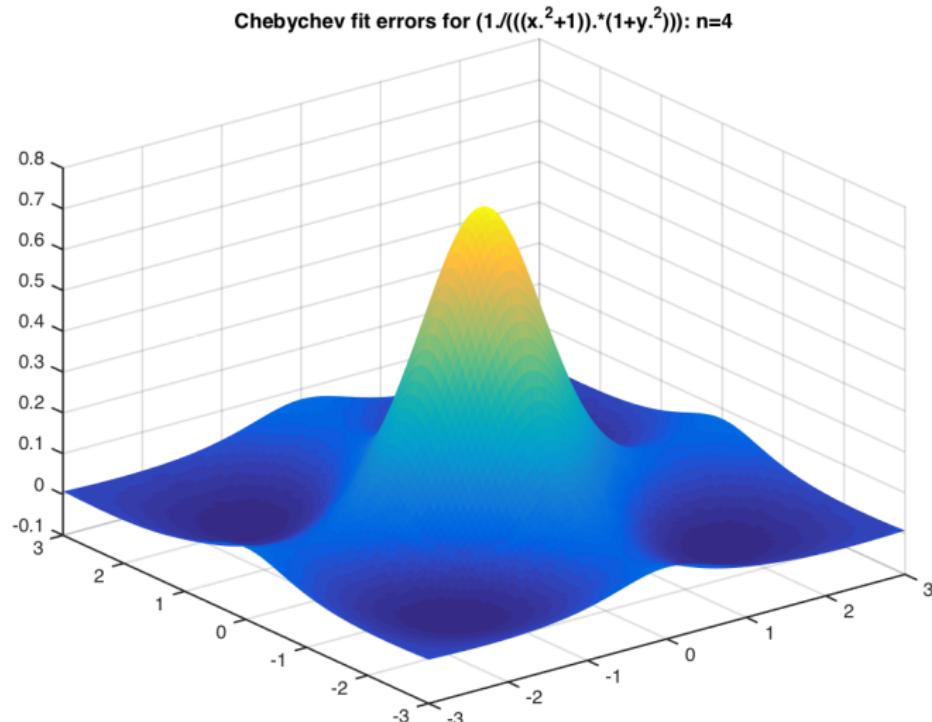
CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 2-DEG. APPROX



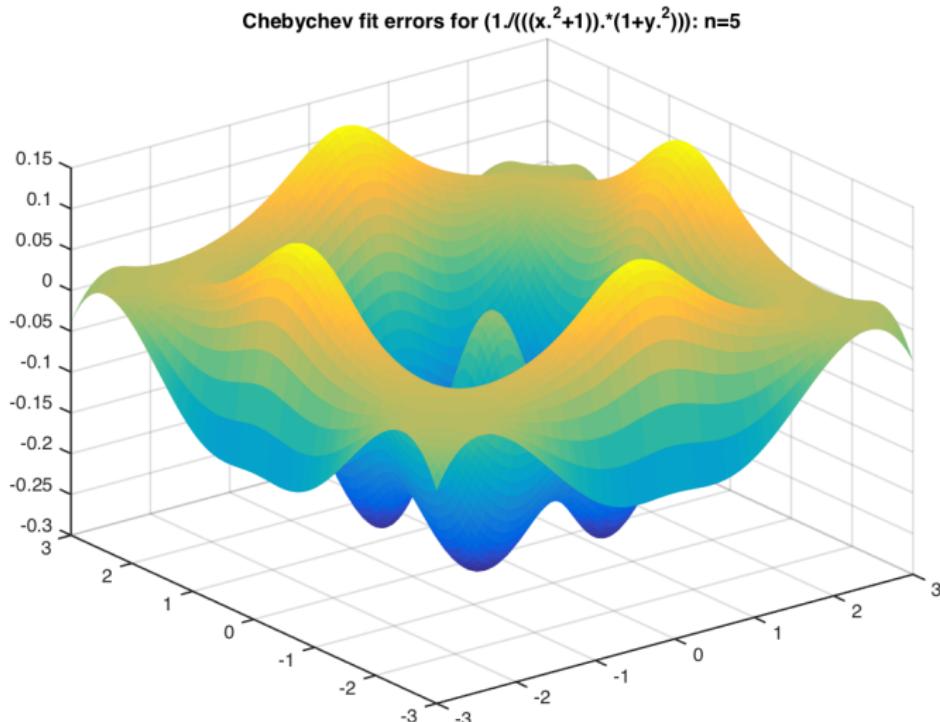
CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 3-DEG. APPROX



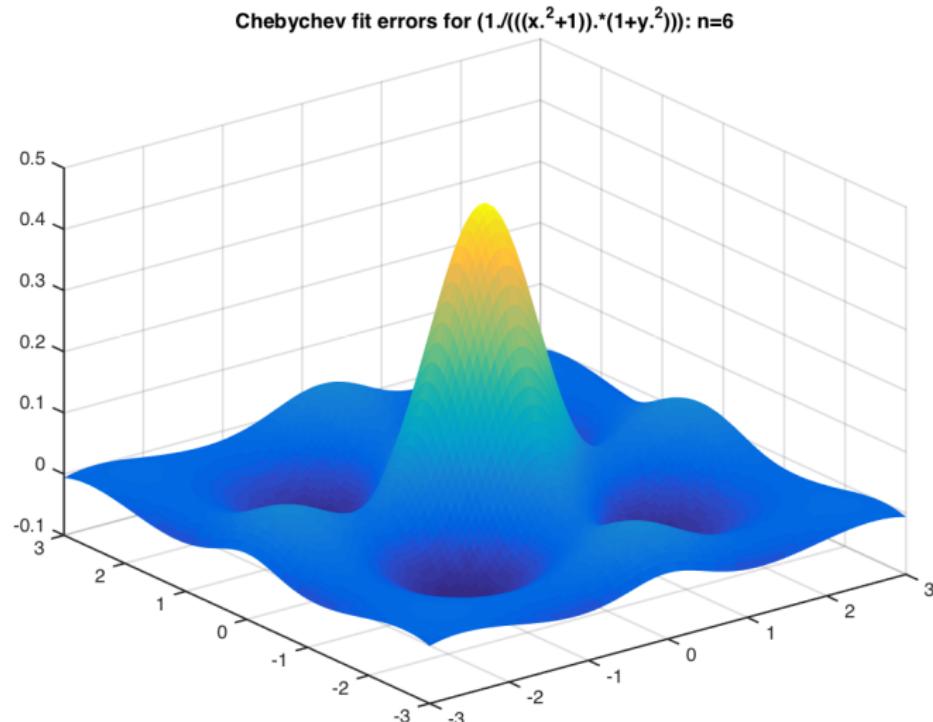
CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 4-DEG. APPROX



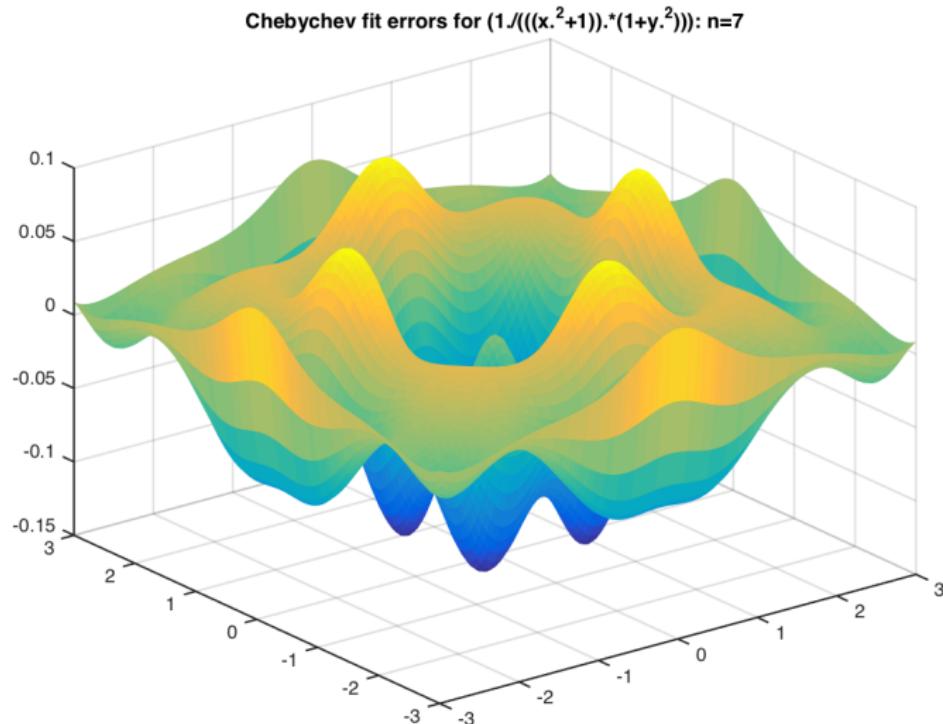
CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 5-DEG. APPROX



CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 6-DEG. APPROX

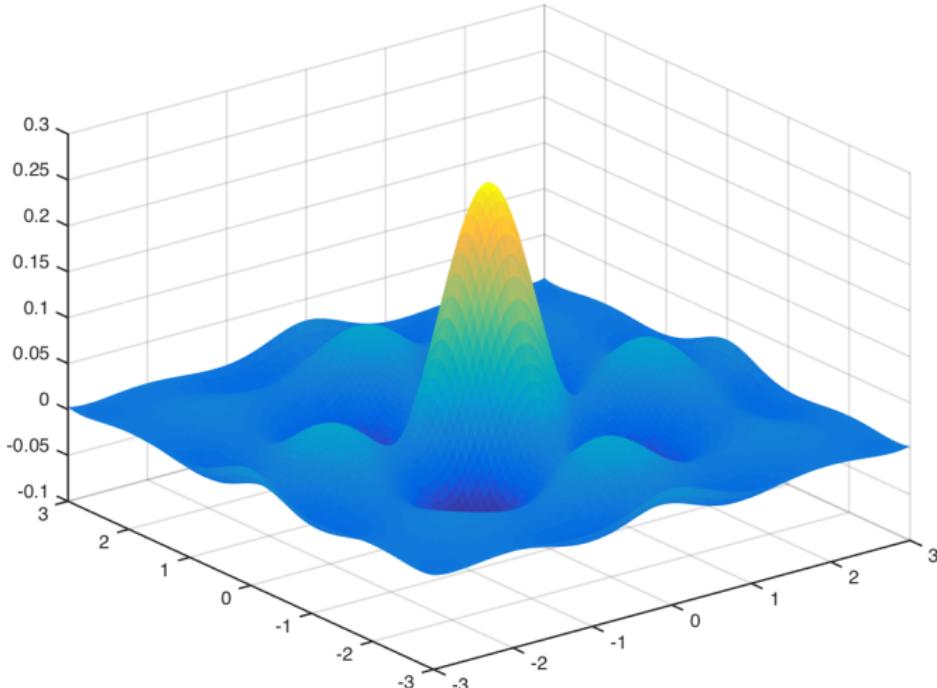


CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 7-DEG. APPROX

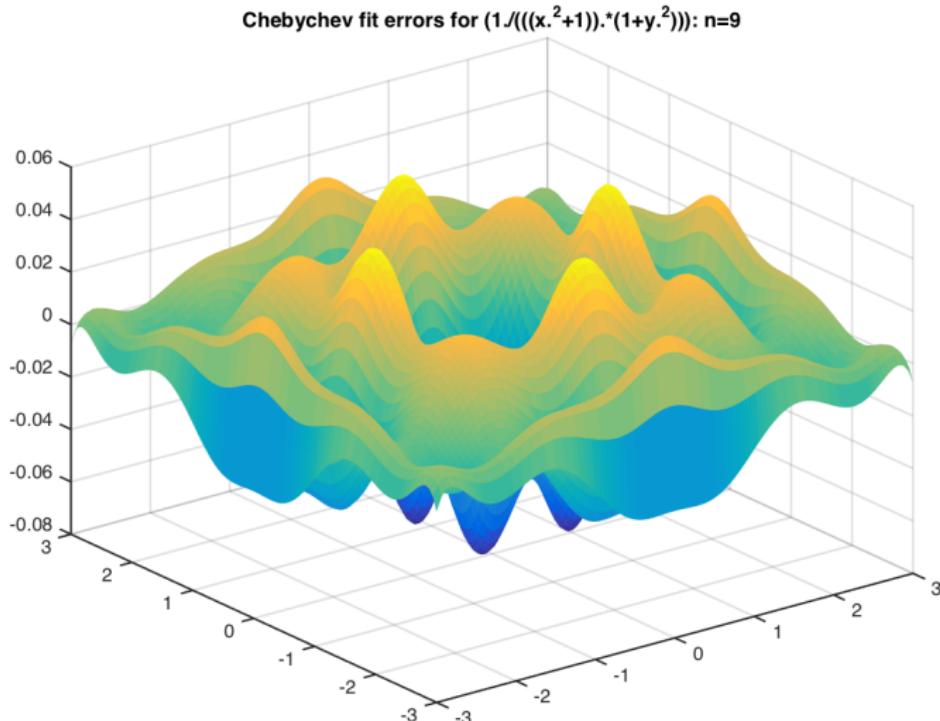


CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 8-DEG. APPROX

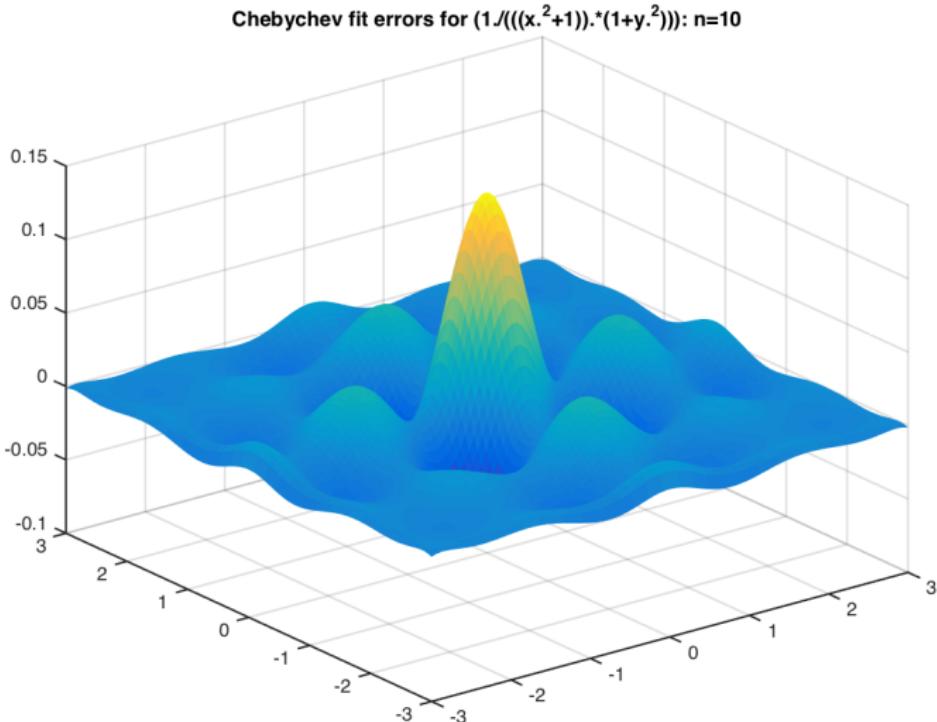
Chebychev fit errors for $(1./(((x.^2+1)).*(1+y.^2)))$: n=8



CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 9-DEG. APPROX



CHEBY ERRORS: $f(x) = \frac{1}{1+x^2}$: 10-DEG. APPROX



TASTE OF SOME TECHNICAL POINTS

- ▶ If we have some function $f(x)$ and some class of approximating functions, Ω
- ▶ We want to find the function $\omega(x) \in \Omega$ such that:

$$\omega = \arg \min_{\omega \in \Omega} \|f - g\|^2$$

- ▶ Interpolating at Chebychev nodes allows us to obtain a lower bound for the error:

$$|f(x) - \hat{f}_{n-1}(x)| \leq \frac{1}{2^{n-1} n!} \max_{\xi \in [-1,1]} |f^{(n)}(\xi)|$$

- ▶ For more see Judd Chapter 6: many ways to approximate functions.

PRACTICAL NOTE-I

- ▶ As mentioned, I've found that this discussion of interpolation is not frequently used by students in research
- ▶ Easier to use canned commands
- ▶ In matlab, interpolation is relatively easy
- ▶ Give it the basis for the function and the function, create an interpolating (and potentially extrapolating function)

PRACTICAL NOTE-II

- ▶ General Matlab interpolation methods
 - ▶ Linear - connect the dots with a line or plane
 - ▶ Nearest, next, previous - rarely used
 - ▶ Spline: Fit points with local polynomials of (typically) degree three
 - ▶ PChip: Spline but doesn't overshoot (good for non-smooth functions)
 - ▶ Makima: spline with orthogonal basis vectors

PRACTICAL NOTE-III

- ▶ How does it work?

- ▶ Let's say I want to define a value function in three dimensions:

```
wvec = linspace(0,10,5)
mvec = [0,1]
cvec = [0,1,2,3,4]
[mgrid,wgrid,cgrid] = ndgrid(wvec,mvec,cvec)
V=rand(size(wgrid))
Vfxn = griddedInterpolant(wgrid,mgrid,cgrid,V)
Vfxn(2,1,2)
```

- ▶ Could also use:

```
scatteredInterpolant(xvec1,xvec2,xvec3,yvec)
for scattered/non-gridded data.
```

PRACTICAL NOTE-IV

- ▶ Making grids in Matlab
- ▶ Two commands: `ndgrid` (interpolation) and `meshgrid` (ease of graphing)
- ▶ $[a, b] = \text{meshgrid}([1 : 2], [3 : 4])$ gives:

$$a = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \quad b = \begin{bmatrix} 3 & 3 \\ 4 & 4 \end{bmatrix}$$

- ▶ $[a, b] = \text{ndgrid}([1 : 2], [3 : 4])$ gives:

$$a = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \quad b = \begin{bmatrix} 3 & 4 \\ 3 & 4 \end{bmatrix}$$

- ▶ Matlab expects things to be in `ndgrid` format for interpolation (while `surf` uses `meshgrid`)
- ▶ Can always transpose, but if things aren't working, check if this is a problem. Useful to have different #'s of indices so interp crashes if things are in wrong order