Perturbation and Projection Methods for Solving DSGE Models

Lawrence J. Christiano

Discussion of projections taken from Christiano-Fisher, ‘Algorithms for Solving Dynamic Models with Occasionally Binding Constraints’, 2000, *Journal of Economic Dynamics and Control.*
Discussion of perturbations primarily taken from Judd’s textbook. Also:
Wouter J. den Haan and Joris de Wind, ‘How well-behaved are higher-order perturbation solutions?’ DNB working paper number 240, December 2009.
For pruning, see Kim, Jinill, Sunghyun Kim, Ernst Schaumburg, and Christopher A. Sims, 2008, “Calculating and using second-order accurate solutions of discrete time dynamic equilibrium models,” *Journal of Economic Dynamics and Control*, 32(11), 3397–3414. Also, Lombardo, 2011, ‘On approximating DSGE models by series expansions,’ European Central Bank.

Outline

- A Toy Example to Illustrate the basic ideas.
  – Functional form characterization of model solution.
  – Projections and Perturbations.

- Neoclassical model.
  – Projection methods
  – Perturbation methods

- Stochastic Simulations and Impulse Responses
  – Focus on perturbation solutions of order two.
  – The need for pruning.

Simple Example

- Suppose that *x* is some exogenous variable and that the following equation implicitly defines *y*:

$$h(x,y) = 0, \ \text{for all}\ x \in X$$

- Let the solution be defined by the 'policy rule', *g*:

$$y = g(x)$$

'Error function'

- satisfying

$$R(x;g) \equiv h(x,g(x)) = 0$$

- for all $x \in X$

The Need to Approximate

- Finding the policy rule, *g*, is a big problem outside special cases
  - ‘Infinite number of unknowns (i.e., one value of *g* for each possible *x*) in an infinite number of equations (i.e., one equation for each possible *x*).’
- Two approaches:
  - projection and perturbation

Projection

- Find a parametric function, $\hat{g}(x;\gamma)$, where γ is a vector of parameters chosen so that it imitates the property of the exact solution, i.e., $R(x;g) = 0$ for all $x \in X$, as well as possible.
- Choose values for γ so that

$$\hat{R}(x;\gamma) = h(x, \hat{g}(x;\gamma))$$

- is close to zero for $x \in X$.

- The method is defined by the meaning of ‘close to zero’ and by the parametric function, $\hat{g}(x;\gamma)$, that is used.

Projection, continued

- Spectral and finite element approximations
  - **Spectral functions**: functions, $\hat{g}(x;\gamma)$, in which each parameter in γ influences $\hat{g}(x;\gamma)$ for all $x \in X$ example:

$$\hat{g}(x;\gamma) = \sum_{i=0}^{n} \gamma_i H_i(x),\ \gamma = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{bmatrix}$$

$$H_i(x) = x^i \text{ ~ordinary polynominal (not computationaly efficient)}$$

$$H_i(x) = T_i(\varphi(x)),$$

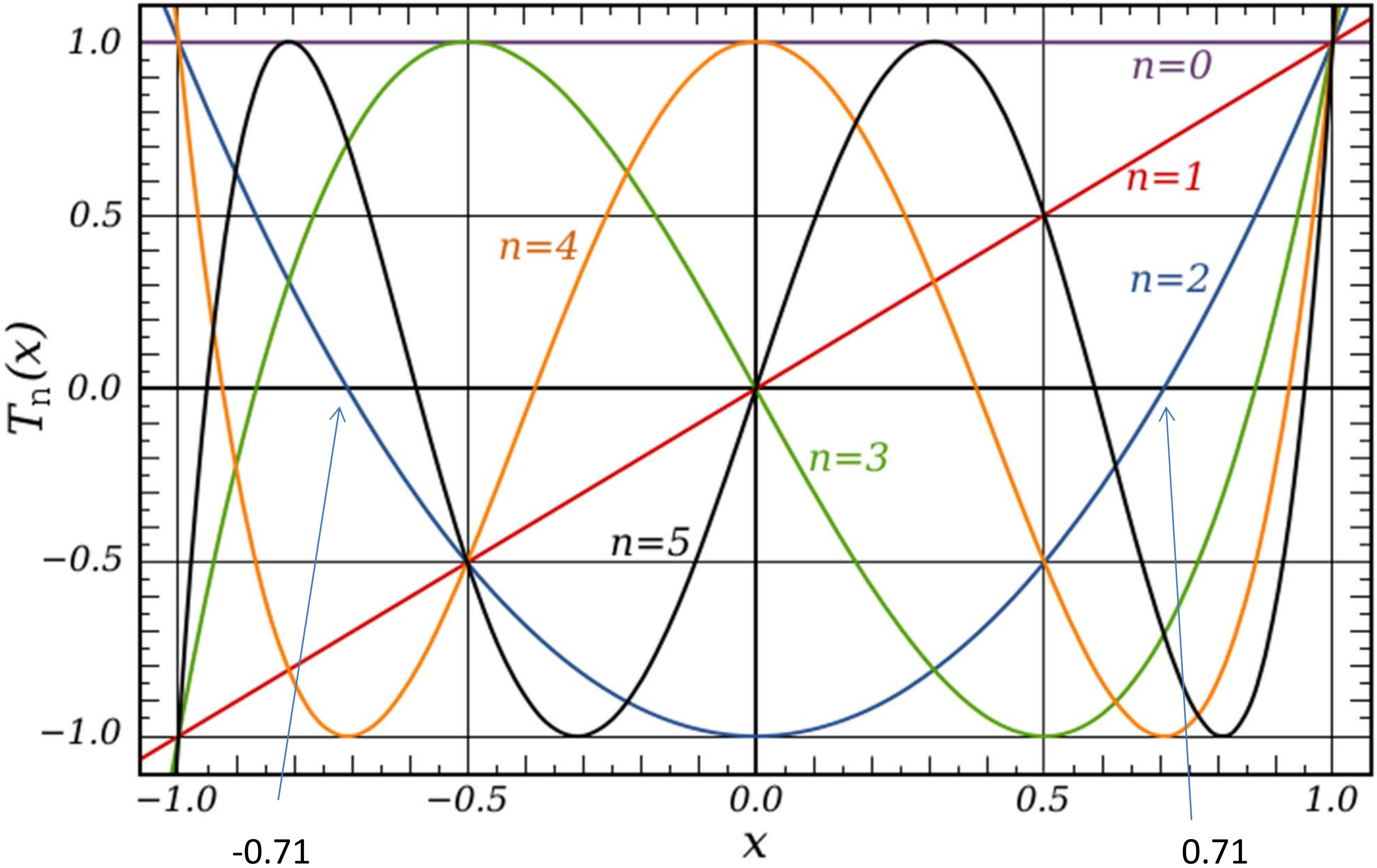
$$T_i(z) \ : \ [-1,1] \to [-1,1],\ i^{th} \text{ order Chebyshev polynomial}$$

$$\varphi \ : \ X \to [-1,1]$$

$$T_0(x) = 1 \quad T_2(x) = 2x^2 - 1$$
$$T_1(x) = x \quad T_3(x) = 4x^3 - 3x$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

Zeros: $x_k = \cos\left(\frac{\pi}{2}\,\frac{2k-1}{n}\right), \quad k = 1, \ldots, n.$

Chebyshev polynomials

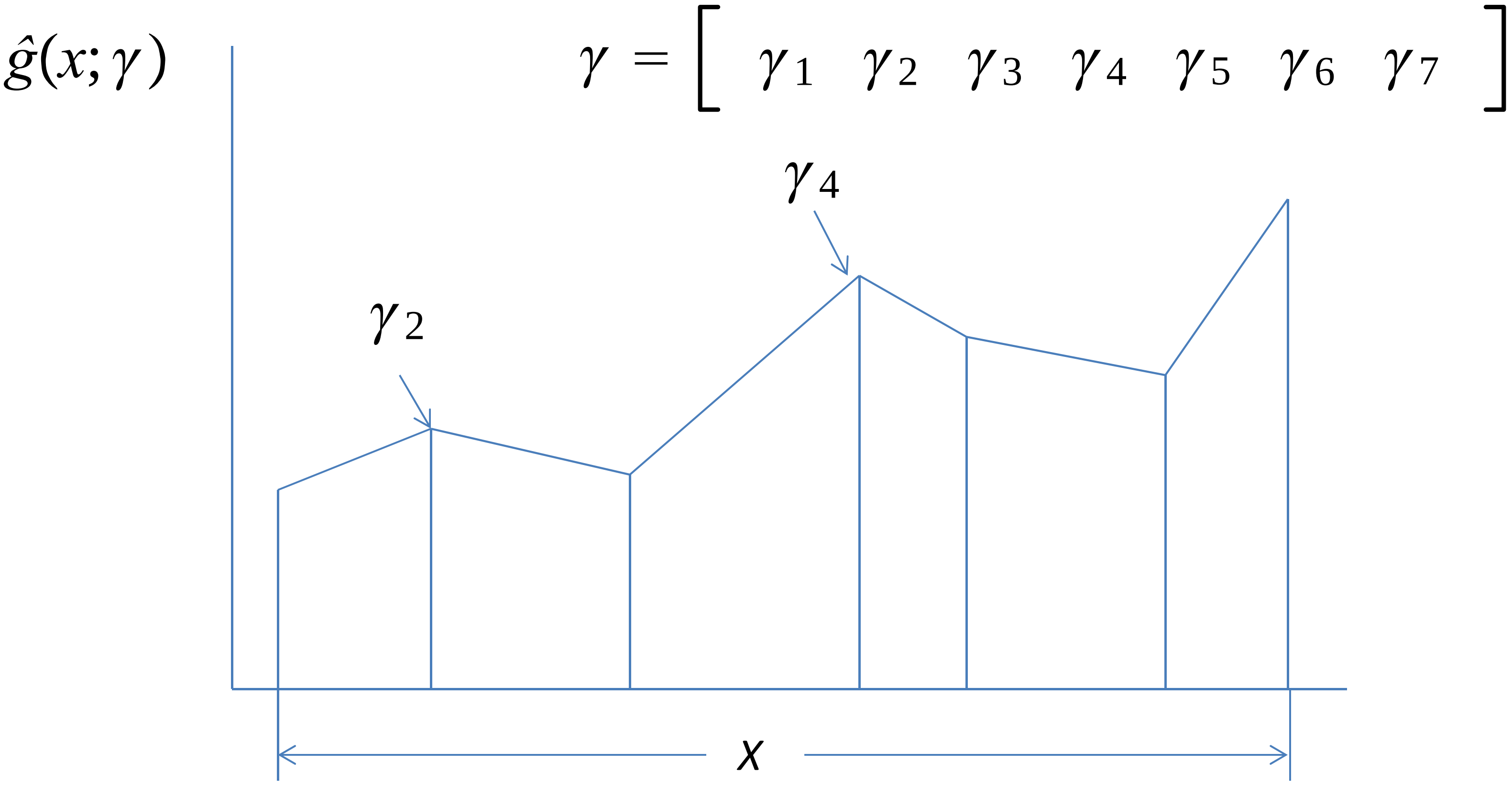


-0.71

0.71

Projection, continued

– Finite element approximations: functions, $\hat{g}(x;\gamma)$, in which each parameter in γ influences $\hat{g}(x;\gamma)$ over only a subinterval of $x \in X$



$$\gamma = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 & \gamma_7 \end{bmatrix}$$

Projection, continued

- ‘Close to zero’: two methods
- **Collocation**, for *n* values of $x: \ x_1, x_2, \ldots, x_n \in X$ choose *n* elements of $\gamma = \begin{bmatrix} \gamma_1 & \cdots & \gamma_n \end{bmatrix}$ so that

$$\hat{R}(x_i;\gamma) = h(x_i, \hat{g}(x_i;\gamma)) = 0, \ i = 1, \ldots, n$$

  - how you choose the grid of x_i’s matters…

Example of Importance of Grid Points

- Here is an example, taken from a related problem, the problem of *interpolation*.
  - You get to evaluate a function on a set of grid points that you select, and you must guess the shape of the function between the grid points.

- Consider the 'Runge' function,

$$f(k) = \frac{1}{1+k^2},\ k \in [-5,5]$$

This is called the 'Runge phenomenon', discovered by Carl Runge in 1901.

- Next slide shows what happens when you select 11 equally-spaced grid points and interpolate by fitting a 10th order polynomial.
  - As you increase the number of grid points on a fixed interval grid, oscillations in tails grow more and more violent.

How You Select the Grid Points Matters

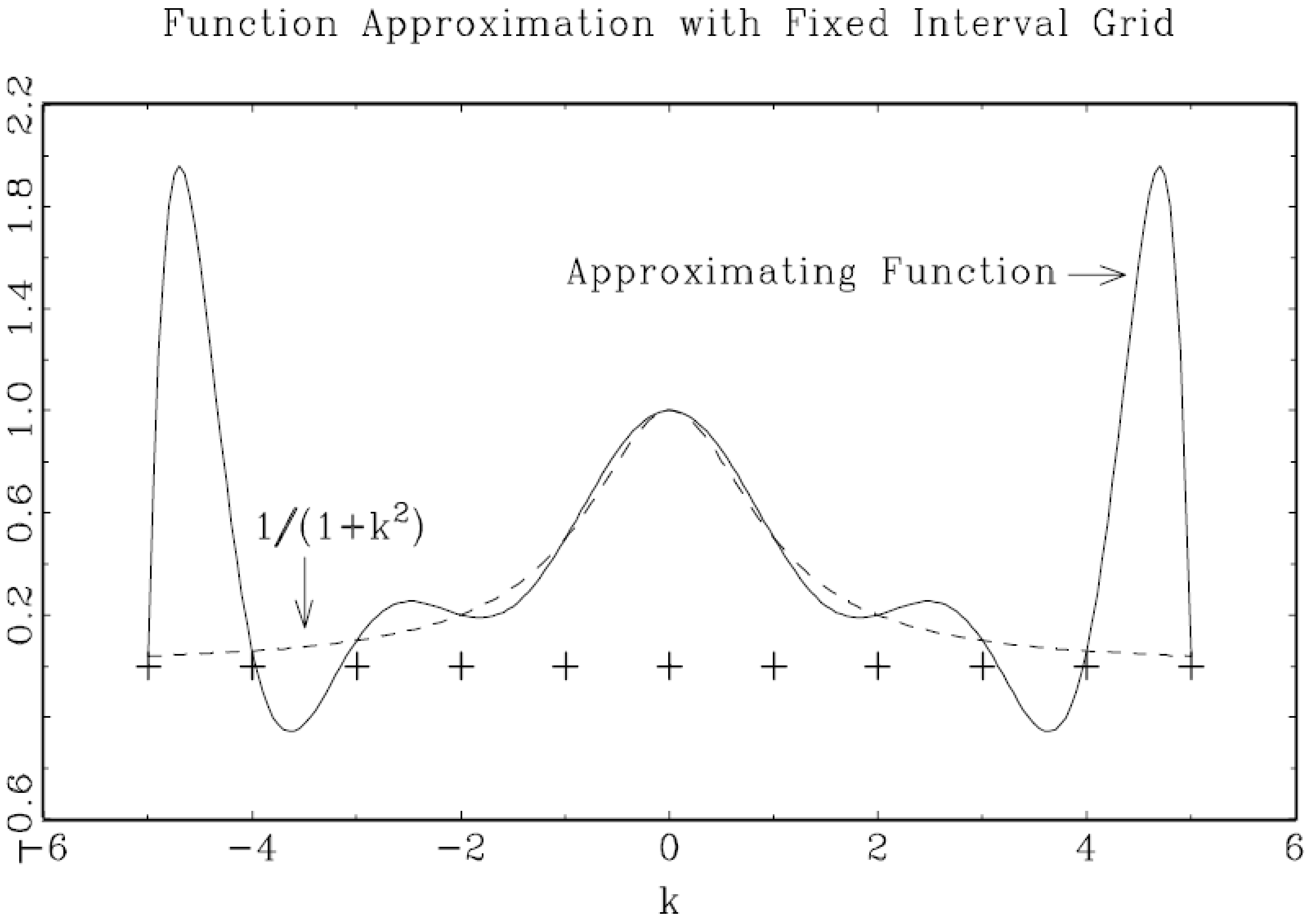


Figure from Christiano-Fisher, JEDC, 2000

Example of Importance of Grid Points

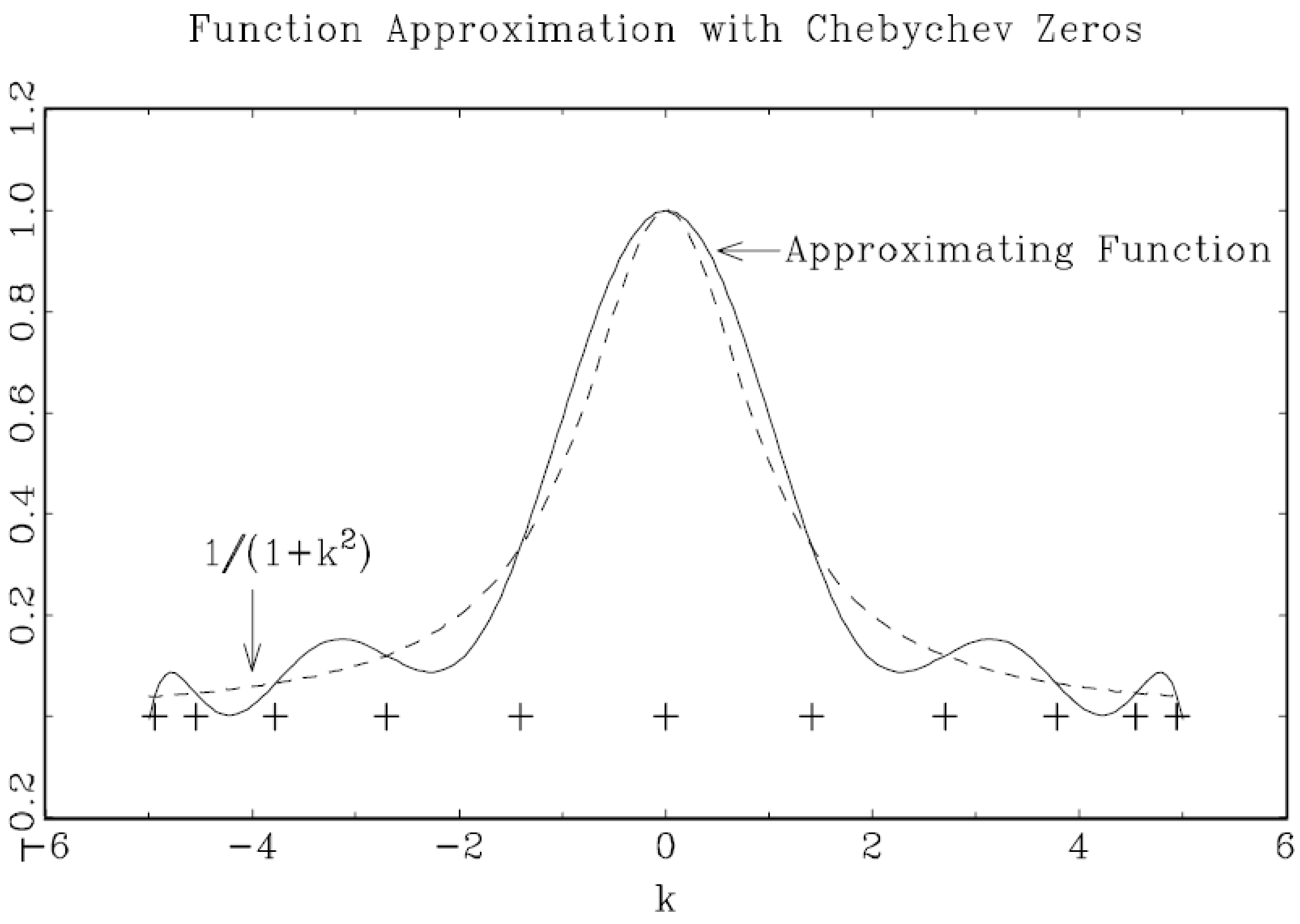
- Here is an example, taken from a related problem, the problem of *interpolation*.
  - You get to evaluate a function on a set of grid points that you select, and you must guess the shape of the function between the grid points.

- Consider the 'Runge' function,

$$f(k) = \frac{1}{1+k^2},\ k \in [-5,5]$$

- Next slide shows what happens when you select 11 equally-spaced grid points and interpolate by fitting a 10th order polynomial.
  - As you increase the number of grid points on a fixed interval grid, oscillations in tails grow more and more violent.

- Chebyshev approximation theorem: distribute more points in the tails (by selecting zeros of Chebyshev polynomial) and get convergence in sup norm.


Function Approximation with Chebychev Zeros
Approximating Function
1/(1+k^2)
1.2
1.0
0.8
0.6
0.4
0.2
0.2
-6
-4
-2
0
2
4
6
k

Projection, continued

- ‘Close to zero’: two methods
- **Collocation**, for *n* values of $x\ :\ x_1, x_2, \ldots, x_n \in X$ choose *n* elements of $\gamma = \begin{bmatrix} \gamma_1 & \cdots & \gamma_n \end{bmatrix}$ so that

$$\hat{R}(x_i;\gamma) = h(x_i, \hat{g}(x_i;\gamma)) = 0,\ i = 1,\ldots,n$$

  - how you choose the grid of x_i’s matters…

- **Weighted Residual**, for *m>n* values of $x\ :\ x_1, x_2, \ldots, x_m \in X$ choose the *n* γ_i’s

$$\sum_{j=1}^{m} w_j^i h(x_j, \hat{g}(x_j;\gamma)) = 0,\ i = 1,\ldots,n$$

Perturbation

- Projection uses the ‘global’ behavior of the functional equation to approximate solution.
  - Problem: requires finding zeros of non-linear equations. Iterative methods for doing this are a pain.
  - Advantage: can easily adapt to situations the policy rule is not continuous or simply non-differentiable (e.g., occasionally binding zero lower bound on interest rate).
- Perturbation method uses Taylor series expansion (computed using implicit function theorem) to approximate model solution.
  - Advantage: can implement procedure using non-iterative methods.
  - Possible disadvantages:
    - Global properties of Taylor series expansion not necessarily very good.
    - Does not work when there are important non-differentiabilities (e.g., occasionally binding zero lower bound on interest rate).

Taylor Series Expansion

- Let $f : R \rightarrow R$ be *k+1* differentiable on the open interval and continuous on the closed interval between *a* and *x*.
  - Then,

$$f(x) = P_k(x) + R_k(x)$$

  - where

Taylor series expansion about $x = a$:

$$P_k(x) = f(a) + f^{(1)}(a)(x-a) + \tfrac{1}{2!} f^{(2)}(a)(x-a)^2 + \ldots + \tfrac{1}{k!} f^{(k)}(a)(x-a)^k$$

remainder:

$$R_k(x) = \tfrac{1}{(k+1)!} f^{(k+1)}(\zeta)(x-a)^{k+1}, \text{ for some } \zeta \text{ between } x \text{ and } a$$

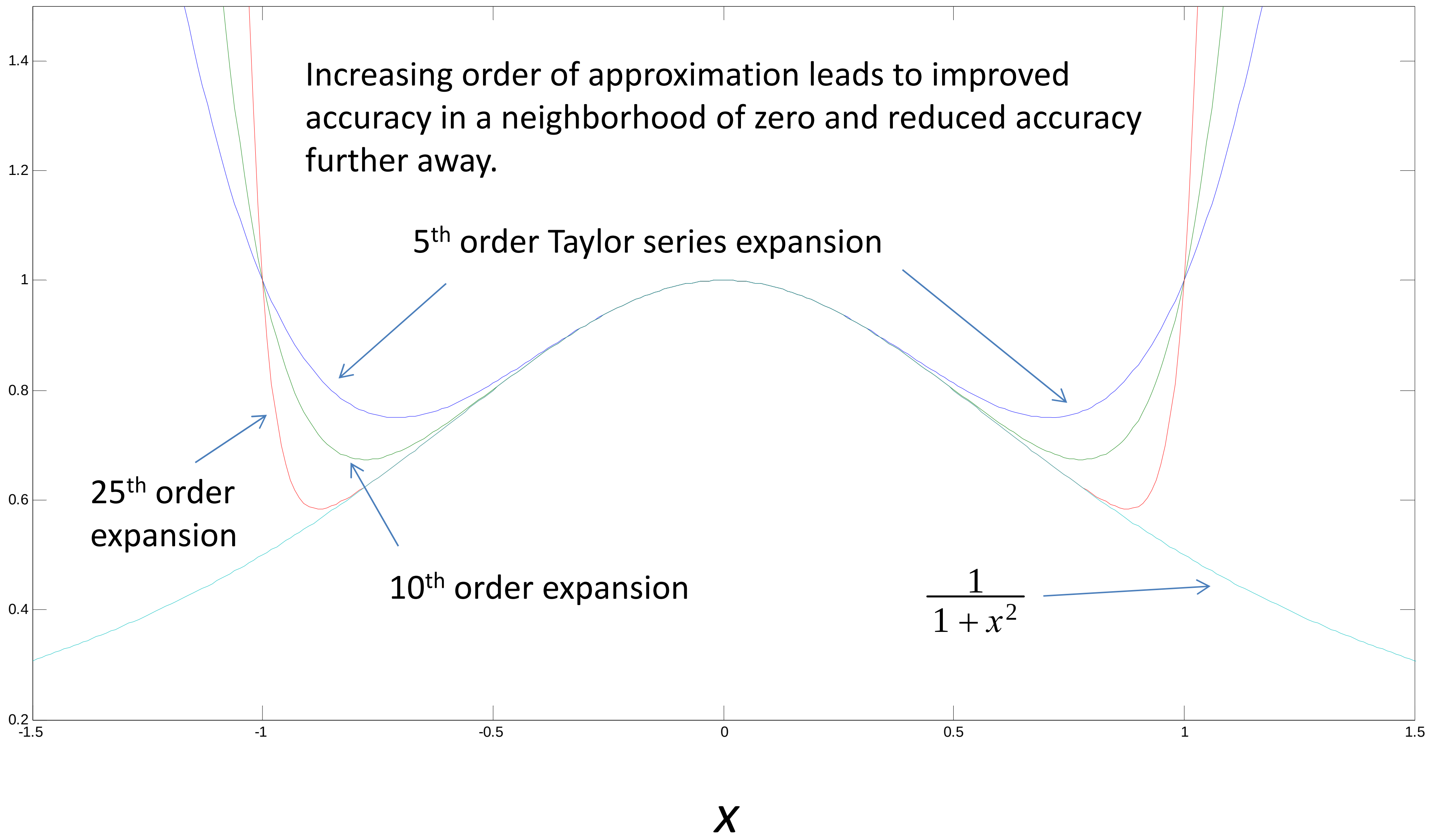
  - Question: is the Taylor series expansion a good approximation for *f*?

Taylor Series Expansion

- It’s not as good as you might have thought.

- The next slide exhibits the accuracy of the Taylor series approximation to the Runge function.
  - In a small neighborhood of the point where the approximation is computed (i.e., *0*), higher order Taylor series approximations are increasingly accurate.
  - Outside that small neighborhood, the quality of the approximation deteriorates with higher order approximations.

Taylor Series Expansions about 0 of Runge Function



Increasing order of approximation leads to improved accuracy in a neighborhood of zero and reduced accuracy further away.

5th order Taylor series expansion

25th order expansion

10th order expansion

$$\frac{1}{1+x^2}$$

Taylor Series Expansion

- Another example: the log function
  - It is often used in economics.
  - Surprisingly, Taylor series expansion does not provide a great global approximation.

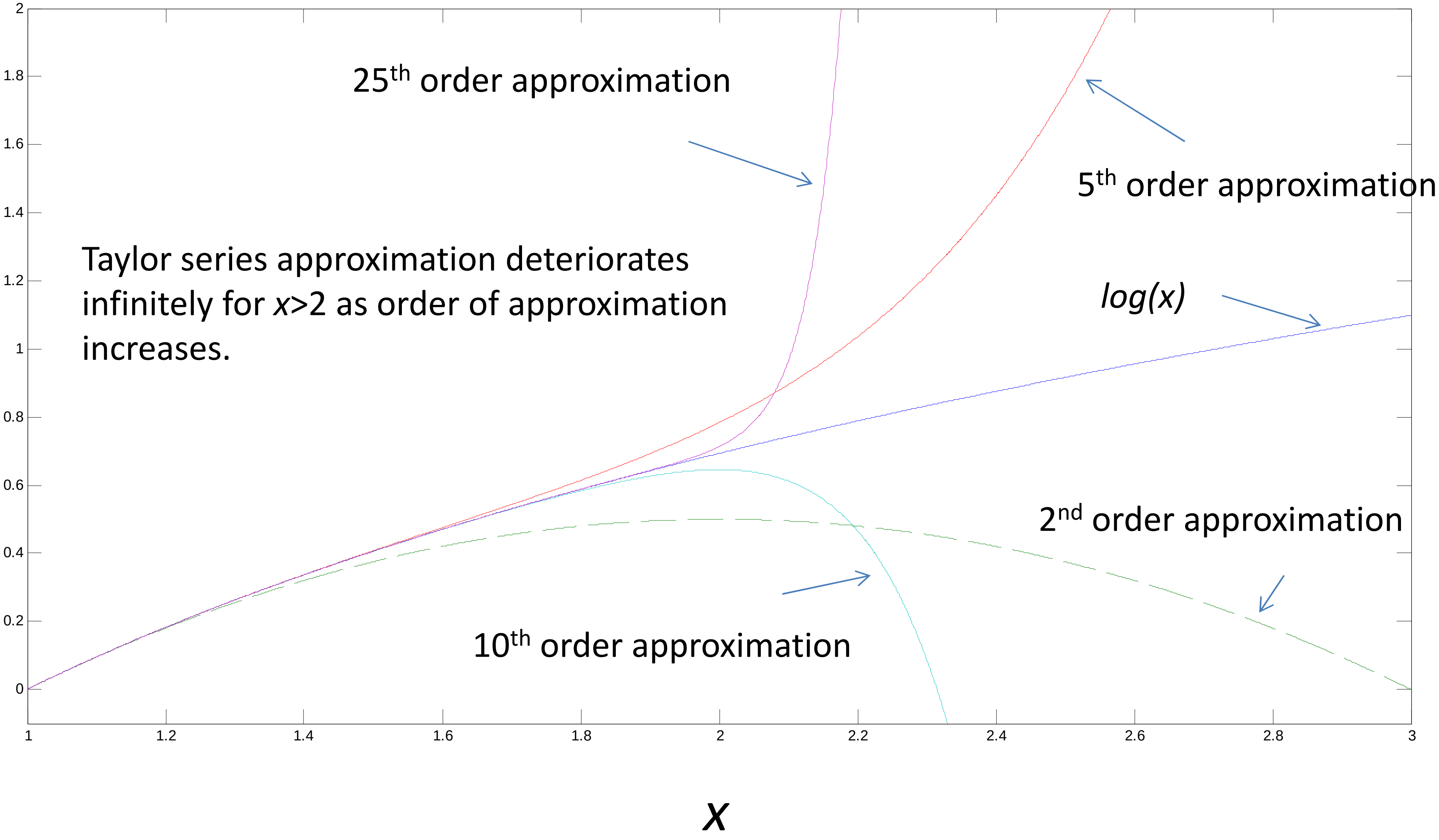
- Approximate *log(x)* by its k^{th} order Taylor series approximation at the point, *x=a*:

$$\log(x) = \log(a) + \sum_{i=1}^{k} (-1)^{i+1} \frac{1}{i} \left(\frac{x-a}{a} \right)^{i}$$

  - This expression diverges as *N*→∞ for

$$x \text{ such that } \left| \frac{x-a}{a} \right| \geq 1$$

Taylor Series Expansion of Log Function About *x=1*



25th order approximation

5th order approximation

log(x)

Taylor series approximation deteriorates infinitely for *x*>2 as order of approximation increases.

2nd order approximation

10th order approximation

Taylor Series Expansion

- In general, cannot expect Taylor series expansion to converge to actual function, globally.
  - There are some exceptions, e.g., Taylor's series expansion of $f(x) = e^x, \cos(x), \sin(x)$ about *x=0* converges to *f(x)* even for *x* far from *0*.
  - Problem: in general it is difficult to say for what values of *x* the Taylor series expansion gives a good approximation.

Taylor Versus Weierstrass

- Problems with Taylor series expansion does not represent a problem with polynomials *per se* as approximating functions.

- Weierstrass approximation theorem: for every continuous function, *f(x)*, defined on *[a,b]*, and for every $\varepsilon > 0$, there exists a finite-ordered polynomial, *p(x)*, on *[a,b]* such that

$$|f(x) - p(x)| < \varepsilon, \text{ for all } x \in [a,b]$$

- Weierstrass – polynomials may approximate well, even if sometimes the Taylor series expansion is not very good.
  - Our analysis of the Runge function illustrates the point: The Taylor series expansion of the Runge function diverges over a large portion of the domain, while the sequence of polynomials associated with the zeros of the Chebyshev polynomials converges to the Runge function in the sup norm.

- Ok, we’re done with the digression on the Taylor series expansion.

- Now, back to the discussion of the perturbation method.
  - It approximates a solution using the Taylor series expansion.

Perturbation Method

- Suppose there is a point, $x^* \in X$, where we know the value taken on by the function, *g*, that we wish to approximate:

$$g(x^*) = g^*, \text{ some } x^*$$

- Use the implicit function theorem to approximate *g* in a neighborhood of x^*
- Note:

$$R(x;g) = 0 \text{ for all } x \in X$$

$$\rightarrow$$

$$R^{(j)}(x;g) \equiv \frac{d^j}{dx^j} R(x;g) = 0 \text{ for all } j, \text{ all } x \in X.$$

Perturbation, cnt'd

- Differentiate *R* with respect to x and evaluate the result at x^* :

$$R^{(1)}(x^*) = \frac{d}{dx}h(x,g(x))|_{x=x^*} = h_1(x^*,g^*) + h_2(x^*,g^*)g'(x^*) = 0$$

$$\rightarrow g'(x^*) = -\frac{h_1(x^*,g^*)}{h_2(x^*,g^*)}$$

- Do it again!

$$R^{(2)}(x^*) = \frac{d^2}{dx^2}h(x,g(x))|_{x=x^*} = h_{11}(x^*,g^*) + 2h_{12}(x^*,g^*)g'(x^*)$$

$$h_{22}(x^*,g^*)[g'(x^*)]^2 + h_2(x^*,g^*)g''(x^*).$$

$\rightarrow$ Solve this linearly for $g''(x^*)$.

Perturbation, cnt'd

- Preceding calculations deliver (assuming enough differentiability, appropriate invertibility, a high tolerance for painful notation!), recursively:

$$g'(x^*), g''(x^*), \ldots, g^{(n)}(x^*)$$

- Then, have the following Taylor's series approximation:

$$\begin{aligned} g(x) &\approx \hat{g}(x) \\ \hat{g}(x) &= g^* + g'(x^*) \times (x - x^*) \\ &\quad + \frac{1}{2} g''(x^*) \times (x - x^*)^2 + \ldots + \frac{1}{n!} g^{(n)}(x^*) \times (x - x^*)^n \end{aligned}$$

Perturbation, cnt'd

- Check….
- Study the graph of

$$R(x;\hat{g})$$

  - over $x \in X$ to verify that it is everywhere close to zero (or, at least in the region of interest).

Example: a Circle

- Function:

$$h(x,y) = (x^2 + y^2) - 4 = 0.$$

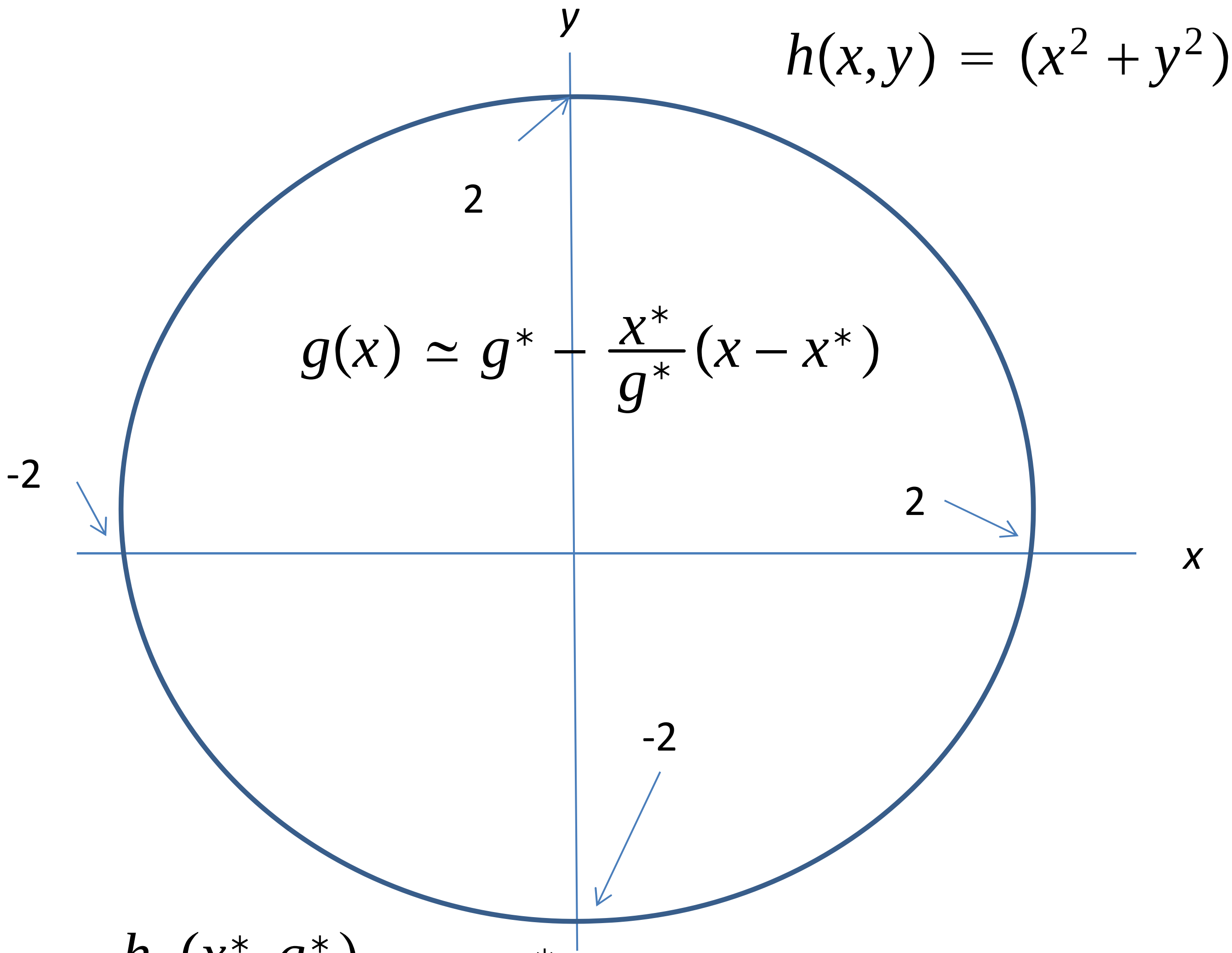
- For each *x* except *x=-2, 2*, there are two distinct *y* that solve *h(x,y)=0*:

$$y = g_1(x) \equiv +\sqrt{4 - x^2},\ y = g_2(x) \equiv -\sqrt{4 - x^2}.$$

- The perturbation method does not require that the function g that solves *h(x,g(x))=0* be unique.
  - When you specify the value of the function, *g*, at the point of the expansion, you select the function whose Taylor series expansion is delivered by the perturbation method.

Example of Implicit Function Theorem

$$h(x,y) = (x^2 + y^2) - 4 = 0.$$



$$g(x) \simeq g^* - \frac{x^*}{g^*}(x - x^*)$$

$$g'(x^*) = -\frac{h_1(x^*, g^*)}{h_2(x^*, g^*)} = -\frac{x^*}{g^*} \quad (h_2 \text{ had better not be zero!})$$

Outline

- A Toy Example to Illustrate the basic ideas.
  – Functional form characterization of model solution.
  – Projections and Perturbations.

Done!

- Neoclassical model.
  – Projection methods
  – Perturbation methods

- Stochastic Simulations and Impulse Responses
  – Focus on perturbation solutions of order two.
  – The need for pruning.

Neoclassical Growth Model

- Objective:

$$E_0 \sum_{t=0}^{\infty} \beta^t u(c_t),\ u(c_t) = \frac{c_t^{1-\gamma} - 1}{1-\gamma}$$

- Constraints:

$$c_t + \exp(k_{t+1}) \leq f(k_t, a_t),\ t = 0, 1, 2, \ldots.$$

$$a_t = \rho a_{t-1} + \varepsilon_t,\ \varepsilon_t \sim E\varepsilon_t = 0,\ E\varepsilon_t^2 = V_\varepsilon$$

$$f(k_t, a_t) = \exp(\alpha k_t)\exp(a_t) + (1-\delta)\exp(k_t)$$

Why *Log* Capital?

- Might hope to get an accurate solution.
- Consider the special case, $\alpha = \gamma = 1$
  - In this case, we know the solution is given by:

$$K_{t+1} = \beta\alpha \exp(a_t) K_t^{\alpha},\ K_t \equiv \exp(k_t)$$

  - So, in terms of log capital, the solution is *exactly* linear:

$$k_{t+1} = \log(\beta\alpha) + a_t + \alpha k_t$$

  - Solution methods often work with polynomials (the perturbation method always does!) , so in case, $\alpha = \gamma = 1$, you would get exactly the right answer.

Efficiency Condition

$$E_t[u'\left(\overbrace{f(k_t,a_t)-\exp(k_{t+1})}^{c_t}\right)$$

$$-\beta u'\left(\overbrace{f(k_{t+1},\rho a_t+\sigma\varepsilon_{t+1})-\exp(k_{t+2})}^{c_{t+1}}\right)\overbrace{f_K(k_{t+1},\rho a_t+\sigma\varepsilon_{t+1})}^{\text{period } t\text{+1 marginal product of capital}}\quad]\ =0.$$

- Here,

  k_t, a_t ~given numbers

  ε_{t+1} ~random variable

  time t choice variable, k_{t+1}

- Parameter, σ , indexes a set of models, with the model of interest corresponding to

$$\sigma=1$$

Solution

- A policy rule,

$$k_{t+1} = g(k_t, a_t, \sigma).$$

- With the property:

$$R(k_t, a_t, \sigma; g) \equiv E_t\{u'\left(\overbrace{f(k_t, a_t) - \exp[g(k_t, a_t, \sigma)]}^{c_t}\right)$$

$$-\beta u'\left(\overbrace{f\left(\overbrace{g(k_t, a_t, \sigma)}^{k_{t+1}}, \overbrace{\rho a_t + \sigma\varepsilon_{t+1}}^{a_{t+1}}\right) - \exp\left[g\left(\overbrace{g(k_t, a_t, \sigma)}^{k_{t+1}}, \overbrace{\rho a_t + \sigma\varepsilon_{t+1}}^{a_{t+1}}, \sigma\right)\right]}^{c_{t+1}}\right)$$

$$\times f_K\left(\overbrace{g(k_t, a_t, \sigma)}^{k_{t+1}}, \overbrace{\rho a_t + \sigma\varepsilon_{t+1}}^{a_{t+1}}\right)\} = 0,$$

- for all $a_t, k_t, \sigma.$

Projection Methods

- Let

$$\hat{g}(k_t, a_t, \sigma; \gamma)$$

  - be a function with finite parameters (could be either spectral or finite element, as before).

- Choose parameters, γ, to make

$$R(k_t, a_t, \sigma; \hat{g})$$

  - as close to zero as possible, over a range of values of the state.
  - use weighted residuals or Collocation.

Occasionally Binding Constraints

- Suppose we include a non-negativity constraint on investment.
  - Lagrangian approach. Add a multiplier, λ_t, to the set of functions to be computed, and add the ('complementary slackness') equation:

Non-negativity constraint on investment

$$\overbrace{\lambda_t}^{\geq 0}\ \overbrace{\left[\exp(g(k_t, a_t, \sigma)) - (1-\delta)\exp(k_t)\right]}^{\geq 0} = 0$$

- Conceptually straightforward to apply preceding solution method. For details, see Christiano-Fisher, 'Algorithms for Solving Dynamic Models with Occasionally Binding Constraints', 2000, *Journal of Economic Dynamics and Control.*
  - This paper describes a wide range of strategies, including those based on parameterizing the expectation function, that may be easier, when constraints occasionally bind.

Perturbation

- Conventional application of the perturbation approach, as in the toy example, requires knowing the value taken on by the policy rule at a point.

- The overwhelming majority of models used in macro do have this property.

  - In these models, can compute non-stochastic steady state without any knowledge of the policy rule, *g*.
  - Non-stochastic steady state is k^* such that

$$k^* = g\left(k^*, \overbrace{0}^{a=0 \text{ (nonstochastic steady state in no uncertainty case)}}, \overbrace{0}^{\sigma=0 \text{ (no uncertainty)}} \right)$$

  - and

$$k^* = \log\left\{\left[\frac{\alpha\beta}{1-(1-\delta)\beta}\right]^{\frac{1}{1-\alpha}}\right\}.$$

Perturbation

- Error function:

$$R(k_t, a_t, \sigma; g) \equiv E_t\{u'\left(\overbrace{f(k_t, a_t) - \exp[g(k_t, a_t, \sigma)]}^{c_t}\right)$$

$$-\beta u'\left[\overbrace{f(g(k_t, a_t, \sigma), \rho a_t + \sigma\varepsilon_{t+1}) - \exp[g(g(k_t, a_t, \sigma), \rho a_t + \sigma\varepsilon_{t+1}, \sigma)]}^{c_{t+1}}\right]$$

$$\times f_K(g(k_t, a_t, \sigma), \rho a_t + \sigma\varepsilon_{t+1})\} = 0,$$

  – for all values of k_t, a_t, σ.

- So, all order derivatives of *R* with respect to its arguments are zero (assuming they exist!).

Four (Easy to Show) Results About Perturbations

- Taylor series expansion of policy rule:

$$g(k_t, a_t, \sigma) \simeq \overbrace{k + g_k(k_t - k) + g_a a_t + g_\sigma \sigma}^{\text{linear component of policy rule}}$$

$$\overbrace{+\frac{1}{2}[g_{kk}(k_t - k)^2 + g_{aa}a_t^2 + g_{\sigma\sigma}\sigma^2] + g_{ka}(k_t - k)a_t + g_{k\sigma}(k_t - k)\sigma + g_{a\sigma}a_t\sigma + \ldots}^{\text{second and higher order terms}}$$

  - $g_\sigma = 0$: to a first order approximation, ‘certainty equivalence’
  - All terms found by solving linear equations, except coefficient on past endogenous variable, g_k ,which requires solving for eigenvalues
  - To second order approximation: slope terms certainty equivalent – $$g_{k\sigma} = g_{a\sigma} = 0$$
  - Quadratic, higher order terms computed recursively.

First Order Perturbation

- Working out the following derivatives and evaluating at $k_t = k^*, a_t = \sigma = 0$

$$R_k(k_t, a_t, \sigma; g) = R_a(k_t, a_t, \sigma; g) = R_\sigma(k_t, a_t, \sigma; g) = 0$$

- Implies:

‘problematic term’

Source of certainty equivalence
In linear approximation

$$R_k = u''(f_k - e^g g_k) - \beta u' f_{Kk} g_k - \beta u''(f_k g_k - e^g g_k^2) f_K = 0$$

$$R_a = u''(f_a - e^g g_a) - \beta u'[f_{Kk} g_a + f_{Ka} \rho] - \beta u''(f_k g_a + f_a \rho - e^g [g_k g_a + g_a \rho]) f_K = 0$$

$$R_\sigma = -[u' e^g + \beta u''(f_k - e^g g_k) f_K] g_\sigma = 0$$

Absence of arguments in these functions reflects they are evaluated in $k_t = k^*, a_t = \sigma = 0$

Technical notes for next slide

$$\begin{aligned}
u''(f_k - e^g g_k) - \beta u' f_{Kk} g_k - \beta u''(f_k g_k - e^g g_k^2) f_K &= 0 \\
\frac{1}{\beta}(f_k - e^g g_k) - u' \frac{f_{Kk}}{u''} g_k - (f_k g_k - e^g g_k^2) f_K &= 0 \\
\frac{1}{\beta} f_k - \left[\frac{1}{\beta} e^g + u' \frac{f_{Kk}}{u''} + f_k f_K \right] g_k + e^g g_k^2 f_K &= 0 \\
\frac{1}{\beta} \frac{f_k}{e^g f_K} - \left[\frac{1}{\beta f_K} + \frac{u'}{u''} \frac{f_{Kk}}{e^g f_K} + \frac{f_k}{e^g} \right] g_k + g_k^2 &= 0 \\
\frac{1}{\beta} - \left[1 + \frac{1}{\beta} + \frac{u'}{u''} \frac{f_{Kk}}{e^g f_K} \right] g_k + g_k^2 &= 0
\end{aligned}$$

- Simplify this further using:

$$\begin{aligned}
& \beta f_K \sim \text{steady state equation} \\
f_K &= \alpha K^{\alpha-1} \exp(a) + (1-\delta),\ K \equiv \exp(k) \\
&= \alpha \exp[(\alpha-1)k + a] + (1-\delta) \\
f_k &= \alpha \exp[\alpha k + a] + (1-\delta)\exp(k) = f_K e^g \\
f_{Kk} &= \alpha(\alpha-1)\exp[(\alpha-1)k + a] \\
f_{KK} &= \alpha(\alpha-1)K^{\alpha-2}\exp(a) = \alpha(\alpha-1)\exp[(\alpha-2)k + a] = f_{Kk} e^{-g}
\end{aligned}$$

- to obtain polynomial on next slide.

First Order, cont'd

- Rewriting $R_k = 0$ term:

$$\frac{1}{\beta} - \left[1 + \frac{1}{\beta} + \frac{u'}{u''} \frac{f_{KK}}{f_K} \right] g_k + g_k^2 = 0$$

- There are two solutions, $0 < g_k < 1,\ g_k > \frac{1}{\beta}$
  - Theory (see Stokey-Lucas) tells us to pick the smaller one.
  - In general, could be more than one eigenvalue less than unity: multiple solutions.
- Conditional on solution to g_k, g_a solved for linearly using $R_a = 0$ equation.
- These results all generalize to multidimensional case

Numerical Example

- Parameters taken from Prescott (1986):

$$\beta = 0.99,\ \gamma = 2(20),\ \alpha = 0.36,\ \delta = 0.02,\ \rho = 0.95,\ V_{\varepsilon} = 0.01^2$$

- Second order perturbation:

$$\hat{g}(k_t, a_{t-1}, \varepsilon_t, \sigma) = \overbrace{k^*}^{3.88} + \overbrace{g_k}^{0.98\ (0.996)} (k_t - k^*) + \overbrace{g_a}^{0.06\ (0.07)} a_t + \overbrace{g_\sigma}^{0} \sigma$$

$$+ \frac{1}{2}[\overbrace{g_{kk}}^{0.014\ (0.00017)} (k_t - k^*)^2 + \overbrace{g_{aa}}^{0.067\ (0.079)} a_t^2 + \overbrace{g_{\sigma\sigma}}^{0.000024\ (0.00068)} \overbrace{\sigma^2}^{1}]$$

$$+ \overbrace{g_{ka}}^{-0.035\ (-0.028)} (k_t - k^*) a_t + \overbrace{g_{k\sigma}}^{0} (k_t - k^*) \sigma + \overbrace{g_{a\sigma}}^{0} a_t \sigma$$

Comparing 1st and 2nd Order Approximation

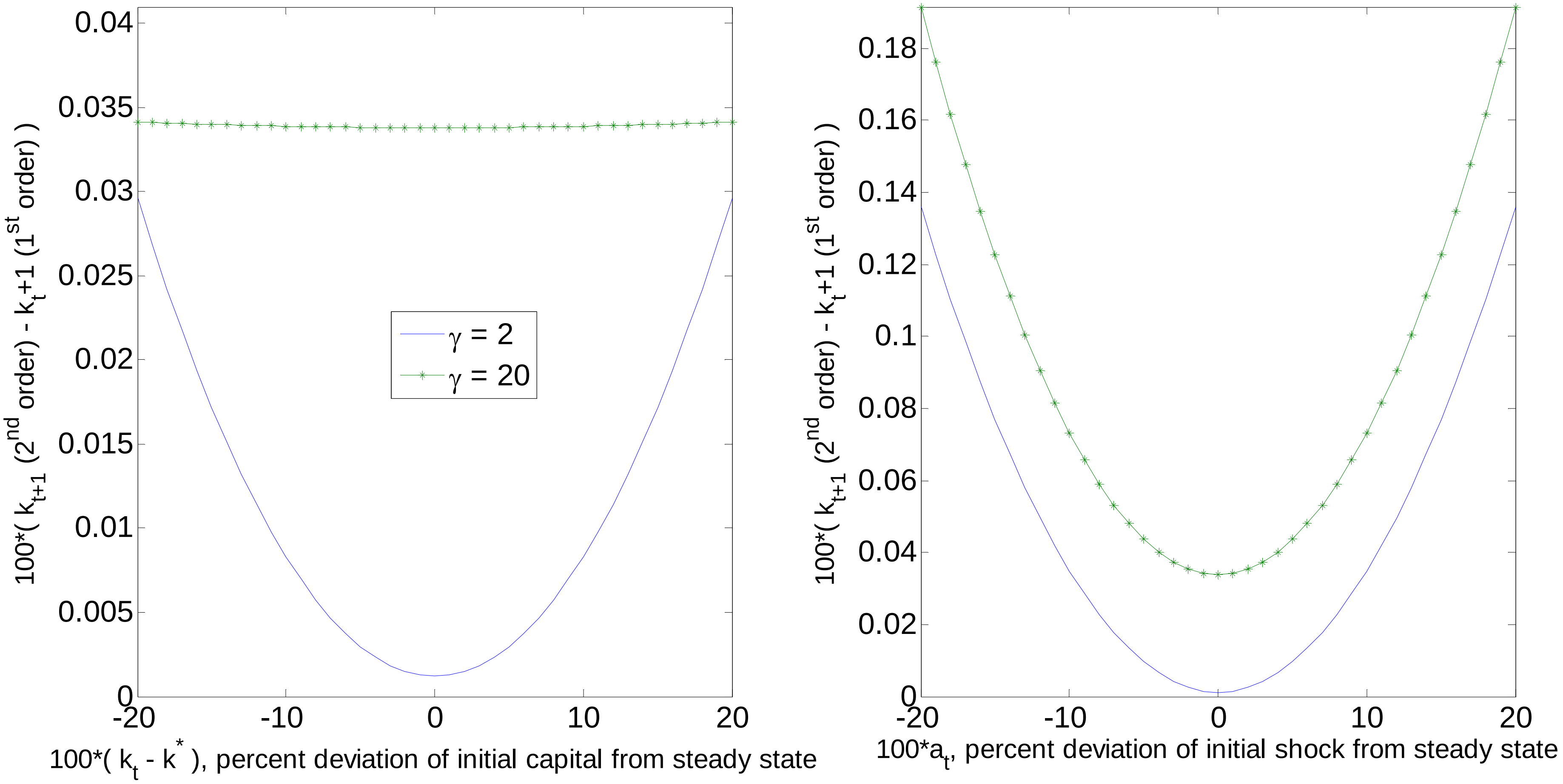
- In practice, 1st order approximation is often used. Want to know if results change much, going from 1st to 2nd order.
  - They appear to change very little in our example.

- Following is a graph that compares the policy rules implied by the first and second order perturbation.

- The graph itself corresponds to the baseline parameterization, and results are reported in parentheses for risk aversion equal to 20.

‘If initial capital is 20 percent away from steady state, then capital choice differs by 0.03 (0.035) percent between the two approximations.’

‘If shock is 6 standard deviations away from its mean, then capital choice differs by 0.14 (0.18) percent between the two approximations’



Number in parentheses at top correspond to γ = 20.

Outline

Done!

- A Toy Example to Illustrate the basic ideas.
  - Functional form characterization of model solution.
  - Projections and Perturbations.

- Neoclassical model.
  - Projection methods
  - Perturbation methods

Done!

Next

- Stochastic Simulations and Impulse Responses
  - Focus on perturbation solutions of order two.
  - The need for pruning.

Next

- Stochastic simulations
  - Mapping from shocks and initial conditions to data.
  - 2nd order approximation and simulation.
  - Pruning: a way to do 2nd order approximation.
  - Naïve simulation versus pruning.
  - Spurious steady state.
  - Extended example of simulation and pruning.

- Impulse response functions
  - Conditional impulse response function (IRF).
  - Unconditional IRF.

Simulations

- Artificial data simulated from a model can be used to compute second moments and other model statistics.
  - These can be compared with analog statistics computed in the data to evaluate model fit.
- Simulated model data can be compared with actual data, to evaluate model fit.
- The computation of impulse responses, when the solution is nonlinear, requires simulations.

Shock and initial conditions

Simulation

data

- Mapping from $k_t, a_t, \sigma,\ \varepsilon_{t+1}$ to k_{t+2} :

$$k_{t+2} = g(g(k_t, a_t, \sigma), \rho a_t + \sigma\varepsilon_{t+1}, \sigma)$$

- Mapping from $k_t, a_t, \sigma,\ \varepsilon_{t+1}, \varepsilon_{t+2}$ to k_{t+3} :

$$k_{t+3} = g\left(g\left(\overbrace{g(k_t, a_t, \sigma)}^{k_{t+1}}, \overbrace{\rho a_t + \sigma\varepsilon_{t+1}}^{a_{t+1}}, \sigma \right), \overbrace{\rho^2 a_t + \rho\sigma\varepsilon_{t+1} + \sigma\varepsilon_{t+2}}^{a_{t+2}}, \sigma \right)$$

- Similarly obtain mapping from $k_t, a_t, \sigma, \varepsilon_{t+1}, \ldots, \varepsilon_{t+j}$ to y_{t+j+1}, for $j=1,2,\ldots$.

2nd order Taylor expansion and simulation

- We do not have the exact policy rule, *g*, and so must do some sort of approximation in computing the simulations.

- Consider the 2nd order expansion of the mapping from $k_t, a_t, \varepsilon_{t+1}, \sigma$ to k_{t+2} :

$$g(g(k_t, a_t, \sigma), \rho a_t + \sigma \varepsilon_{t+1}, \sigma)$$

  – After some (painful!) algebra ($y_t = k_t - k^*$):

$$\begin{aligned} y_{t+2} = {} & g_k^2 y_t + (g_k g_a + g_a \rho) a_t + g_a \sigma \varepsilon_{t+1} \\ & + \frac{1}{2}[(g_{kk} g_k^2 + g_k g_{kk}) y_t^2 + (g_k g_a^2 + 2 g_{ka} g_a \rho + g_k g_{aa} + g_{aa} \rho^2) a_t^2 + g_{aa} \sigma^2 \varepsilon_{t+1}^2 + g_{\sigma\sigma}(g_k + 1)\sigma^2] \\ & + g_k [g_{kk} g_a + g_{ka}(\rho + 1)] y_t a_t + g_{ka} \sigma g_k y_t \varepsilon_{t+1} + \sigma [g_{ka} g_a + g_{aa} \rho] a_t \varepsilon_{t+1} + g_a \varepsilon_{t+1} \sigma \end{aligned}$$

Technical note to previous slide

- Taylor series expansion of $g(g(k_t,a_t,\sigma),\rho a_t+\sigma\varepsilon_{t+1},\sigma)$ about $k_t=k^*,a_t=\varepsilon_{t+1}=\sigma=0$
  - First order terms:

$$k_t\ :\ g_k(g(k_t,a_t,\sigma),\rho a_t+\sigma\varepsilon_{t+1},\sigma)g_k(k_t,a_t,\sigma)=g_k^2$$

$$a_t\ :\ g_k(g(k_t,a_t,\sigma),\rho a_t+\sigma\varepsilon_{t+1},\sigma)g_a(k_t,a_t,\sigma)+g_a(g(k_t,a_t,\sigma),\rho a_t+\sigma\varepsilon_{t+1},\sigma)\rho=g_kg_a+g_a\rho$$

$$\varepsilon_{t+1}\ :\ g_a(g(k_t,a_t,\sigma),\rho a_t+\sigma\varepsilon_{t+1},\sigma)\sigma=g_a\sigma$$

$$\sigma\ :\ g_k(g(k_t,a_t,\sigma),\rho a_t+\sigma\varepsilon_{t+1},\sigma)g_\sigma(k_t,a_t,\sigma)+g_\sigma(g(k_t,a_t,\sigma),\rho a_t+\sigma\varepsilon_{t+1},\sigma)=g_kg_\sigma+g_\sigma=0$$

  - Second order terms:

$$k_t^2\ :\ g_{kk}g_k^2+g_kg_{kk}$$

$$a_t^2\ :\ g_kg_a^2+g_{ka}g_a\rho+g_kg_{aa}+g_{ak}g_a\rho+g_{aa}\rho^2$$

$$\varepsilon_{t+1}^2\ :\ g_{aa}\sigma^2$$

$$\sigma^2\ :\ g_{kk}g_\sigma^2+g_{k\sigma}g_\sigma+g_kg_{\sigma\sigma}+g_{\sigma k}g_\sigma+g_{\sigma\sigma}=g_kg_{\sigma\sigma}+g_{\sigma\sigma}$$

$$k_ta_t\ :\ g_{kk}g_ag_k+g_{ka}\rho g_k+g_kg_{ka}$$

$$k_t\varepsilon_{t+1}\ :\ g_{ka}\sigma g_k$$

$$k_t\sigma\ :\ g_{kk}g_\sigma g_k+g_{k\sigma}g_k+g_kg_{k\sigma}=0$$

$$a_t\varepsilon_{t+1}\ :\ g_{ka}\sigma g_a+g_{aa}\sigma\rho$$

$$a_t\sigma\ :\ g_{kk}g_ag_\sigma+g_kg_{a\sigma}+g_{k\sigma}g_a+g_{ak}g_\sigma\rho+g_{a\sigma}\rho=0$$

$$\varepsilon_{t+1}\sigma\ :\ g_{ak}g_\sigma\sigma+g_{a\sigma}\sigma+g_a=g_a$$

Pruning: a way to do 2nd order approximation

- Second order approximation, mapping to k_{t+3} from $k_t, a_t, \sigma,\ \varepsilon_{t+1}, \varepsilon_{t+2}$ is even more algebra-intensive. Mapping to k_{t+4}, k_{t+5}, … worse still.
- Turns out there is a simple way to compute these mappings, called *pruning*[1]
  - First, draw $\varepsilon_{t+1}, \varepsilon_{t+2}, \varepsilon_{t+3}, \ldots$
  - Then, solve linear system:

Brute force substitution verifies that pruning delivers second order approximation on previous slide.

$$\tilde{y}_{t+j+1} = g_k \tilde{y}_{t+j} + g_a a_{t+j},\ a_{t+j} = \rho a_{t+j-1} + \varepsilon_{t+j},\ j = 1,2,3,\ldots$$

  - Finally,

$$y_{t+j+1} = g_k y_{t+j} + g_a a_{t+j} + \frac{1}{2}[g_{kk}\tilde{y}_{t+j}^2 + g_{aa}a_{t+j}^2 + g_{\sigma\sigma}] + g_{ka}\tilde{y}_{t+j}a_{t+j}$$

[1] Kim, Kim, Schaumburg and Sims, 2008.

Pruning: a way to do 2nd order approximation

- Second order approximation, mapping to k_{t+3} from $k_t, a_t, \sigma,\ \varepsilon_{t+1}, \varepsilon_{t+2}$ is even more algebra-intensive. Mapping to k_{t+4}, k_{t+5}, … worse still.
- Turns out there is a simple way to compute these mappings, called *pruning*[1]
  - First, draw $\varepsilon_{t+1}, \varepsilon_{t+2}, \varepsilon_{t+3}, \ldots$
  - Then, solve linear system:

$$\tilde{y}_{t+j+1} = g_k \tilde{y}_{t+j} + g_a a_{t+j},\ a_{t+j} = \rho a_{t+j-1} + \varepsilon_{t+j},\ j = 1,2,3,\ldots$$

  - Finally,

$$y_{t+j+1} = g_k y_{t+j} + g_a a_{t+j} + \frac{1}{2}\left[g_{kk} \tilde{y}_{t+j}^2 + g_{aa} a_{t+j}^2 + g_{\sigma\sigma}\right] + g_{ka} \tilde{y}_{t+j} a_{t+j}$$

Note that if y_{t+j} were used here, then higher order powers than two would appear, and would not be second order expression.

[1] Kim, Kim, Schaumburg and Sims, 2008.

Pruning: a way to do 2nd order approximation

- Second order approximation, mapping to k_{t+3} from $k_t, a_t, \sigma,\ \varepsilon_{t+1}, \varepsilon_{t+2}$ is even more algebra-intensive. Mapping to $k_{t+4,}$ $k_{t+5,}$ … worse still.
- Turns out there is a simple way to compute these mappings, called *pruning*[1]
  - First, draw $\varepsilon_{t+1}, \varepsilon_{t+2}, \varepsilon_{t+3}, \ldots$
  - Then, solve linear system:

$$\tilde{y}_{t+j+1} = g_k \tilde{y}_{t+j} + g_a a_{t+j},\ a_{t+j} = \rho a_{t+j-1} + \varepsilon_{t+j},\ j = 1,2,3,\ldots$$

  - Finally,

$$y_{t+j+1} = g_k y_{t+j} + g_a a_{t+j} + \frac{1}{2}\left[g_{kk}\tilde{y}_{t+j}^2 + g_{aa}a_{t+j}^2 + g_{\sigma\sigma}\right] + g_{ka}\tilde{y}_{t+j}a_{t+j}$$

By using $\tilde{y}_t$ instead of y_t to remove higher order terms, you are in effect doing what a gardener who removes (prunes) diseased or unwanted parts of plants.

[1] Kim, Kim, Schaumburg and Sims, 2008.

Pruning: a way to do 2nd order approximation

- Second order approximation, mapping to k_{t+3} from $k_t, a_t, \sigma,\ \varepsilon_{t+1}, \varepsilon_{t+2}$ is even more algebra-intensive. Mapping to $k_{t+4,}$ $k_{t+5,}$ … worse still.
- Turns out there is a simple way to compute these mappings, called *pruning*
  - First, draw $\varepsilon_{t+1}, \varepsilon_{t+2}, \varepsilon_{t+3}, \ldots$
  - Then, solve linear system:

$$\tilde{y}_{t+j+1} = g_k \tilde{y}_{t+j} + g_a a_{t+j},\ a_{t+j} = \rho a_{t+j-1} + \varepsilon_{t+j},\ j = 1,2,3,\ldots$$

By using $\tilde{y}_{t+j}$ instead of y_{t+j}
To remove higher order terms,
you guarantee stability of
simulated paths.

  - Finally,

$$y_{t+j+1} = g_k y_{t+j} + g_a a_{t+j} + \frac{1}{2}\left[g_{kk}\tilde{y}_{t+j}^2 + g_{aa}a_{t+j}^2 + g_{\sigma\sigma}\right] + g_{ka}\tilde{y}_{t+j}a_{t+j}$$

Naïve simulation versus pruning

- Naïve simulation (no pruning):

$$y_{t+j+1} = g_k y_{t+j} + g_a a_{t+j} + \frac{1}{2}\left[g_{kk} y_{t+j}^2 + g_{aa} a_{t+j}^2 + g_{\sigma\sigma}\right] + g_{ka} y_{t+j} a_{t+j},$$

$$j = 1, 2, 3, \ldots.$$

- Two problems
  - #1: Implied mapping from $k_t, a_t, \sigma, \varepsilon_{t+1}, \ldots, \varepsilon_{t+j}$ to *y*$_{t+j+1}$, for *j=1,2,…* contains terms of higher order than 2 and so is not a second order approximation. Those higher order terms do not correspond to the ones in the higher order Taylor series expansion (which makes use of derivatives of *g* of higher order than 2) and so they do not confer the desirable (local) accuracy properties of a Taylor series.

Naïve simulation versus pruning

- Naïve simulation (no pruning):

$$y_{t+j+1} = g_k y_{t+j} + g_a a_{t+j} + \frac{1}{2}[g_{kk} y_{t+j}^2 + g_{aa} a_{t+j}^2 + g_{\sigma\sigma}] + g_{ka} y_{t+j} a_{t+j},$$

$$j = 1, 2, 3, \ldots.$$

- Two problems
  - #2:
    - The second order approximation to the policy rule has a second (definitely spurious, in the case of the neoclassical growth model) steady state.
    - Spurious steady state marks a transition into explosive dynamics, profoundly different from actual *g*, in the case of the neoclassical model.

Spurious Steady State

- Setting $a_t=0$ and ignoring $g_{\sigma\sigma}$ (it's small anyway), the 2nd order approximation to the policy rule is:

$$y_{t+1} = g_k y_t + \frac{1}{2} g_{kk} y_t^2$$

- This has two steady states: $y_t=0$ and

$$y = \frac{2(1-g_k)}{g_{kk}} = \frac{2(1-0.98)}{0.014} = 2.86$$

- This corresponds to the following value of the capital stock:

$$y = k - k^* = \log(K/K^*)$$

This is very large, probably not worth worrying about in neoclassical model, but the analog object in medium-sized New Keynesian models is closer to actual steady state.

$$K = \exp(y+k^*) \overbrace{\approx}^{\text{because } g_{\sigma\sigma} \text{ ignored}} \exp(2.86+3.9) = 790.3 \text{ (after rounding)}$$

The Spurious Steady State in 2nd Order Approximation of Neoclassical Model

- Because of scale problem, it is hard to see the policy rule when graphed in the ‘natural way’, k_{t+1} against k_t .

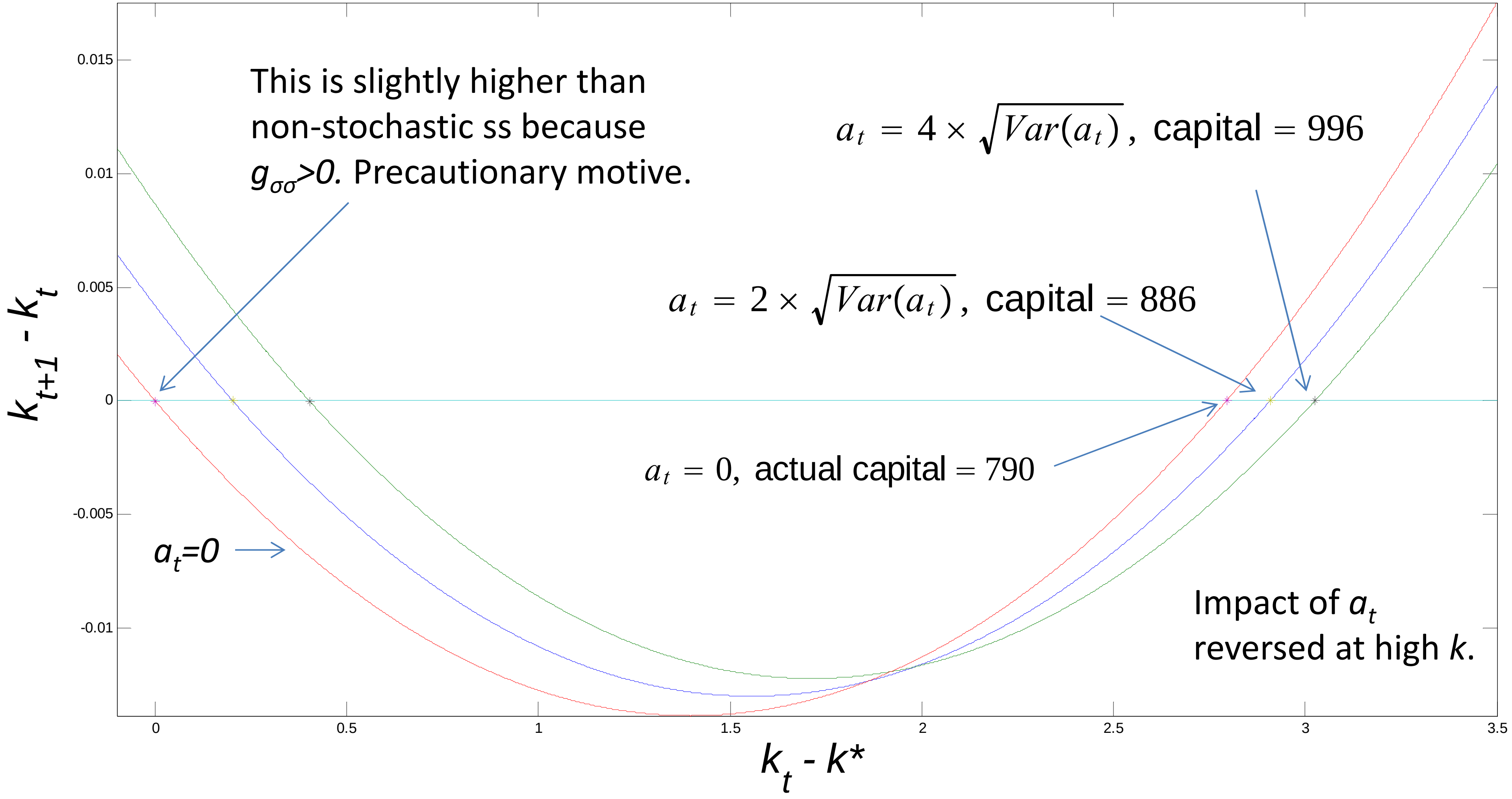
- Instead, will graph:

  – k_{t+1} - k_t against k_t –$k*$ (recall, k_t = log(K_t)).

$$y_{t+1} = \overbrace{g_k}^{0.98} y_t + \overbrace{g_a}^{0.063} a_t + \frac{1}{2}\left[\overbrace{g_{kk}}^{0.014} \times y_t^2 + \overbrace{g_{aa}}^{0.067} a_t^2 + \overbrace{g_{\sigma\sigma}}^{0.000024}\right] + \overbrace{g_{ka}}^{-0.035} \times y_t a_t$$

$y_t \equiv k_t - k^*$, k_t ~log, capital stock

Second order approximation of policy rule



This is slightly higher than non-stochastic ss because $g_{\sigma\sigma}>0$. Precautionary motive.

$a_t = 4 \times \sqrt{Var(a_t)}$, capital $= 996$

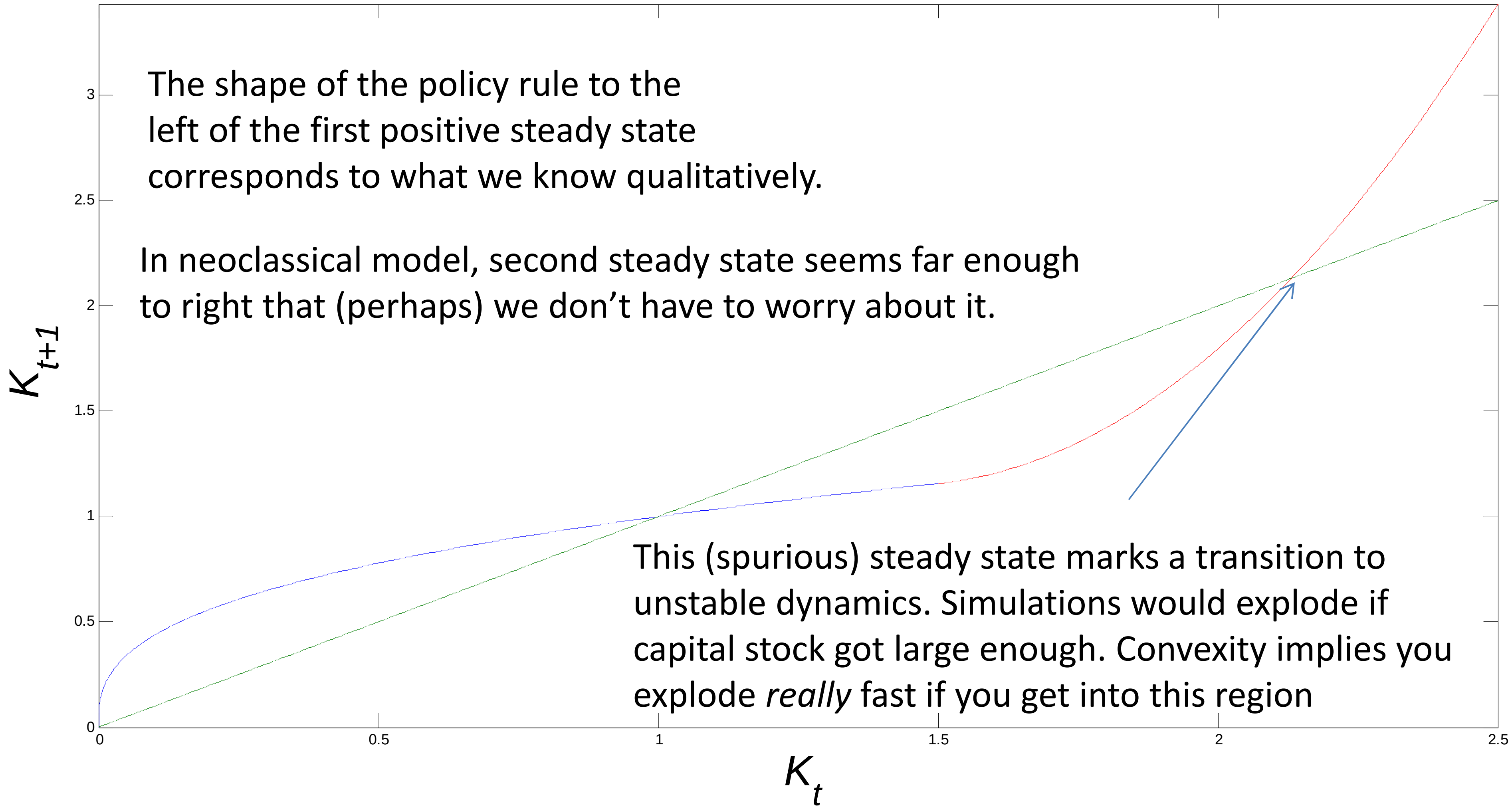
$a_t = 2 \times \sqrt{Var(a_t)}$, capital $= 886$

$a_t = 0$, actual capital $= 790$

$a_t=0$ →

Impact of a_t reversed at high k.

Stylized Representation of 2nd Order Approximation of Policy Rule, *a=0*



The shape of the policy rule to the left of the first positive steady state corresponds to what we know qualitatively.

In neoclassical model, second steady state seems far enough to right that (perhaps) we don’t have to worry about it.

This (spurious) steady state marks a transition to unstable dynamics. Simulations would explode if capital stock got large enough. Convexity implies you explode *really* fast if you get into this region

Next

Done

- Stochastic simulations
  - Mapping from shocks and initial conditions to data.
  - 2nd order approximation and simulation.
  - Pruning: a way to do 2nd order approximation.
  - Naïve simulation versus pruning.
  - Spurious steady state.
  - Extended example of simulation and pruning.

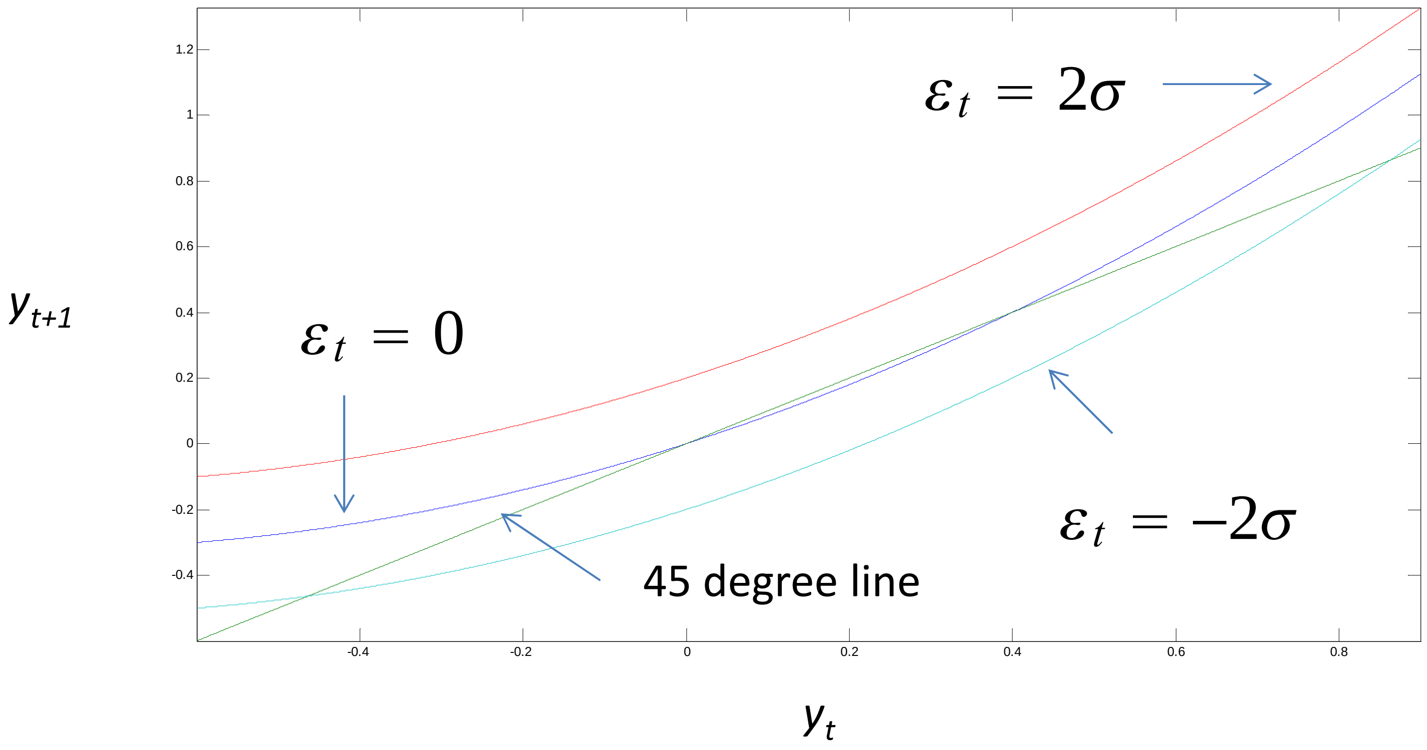
Next

- Impulse response functions
  - Conditional impulse response function (IRF).
  - Unconditional IRF.

Extended Example of Simulation and Pruning

- Simplified version of 2nd order approximation to neoclassical model solution:

$$y_t = \rho y_{t-1} + \alpha y_{t-1}^2 + \varepsilon_t,\ \rho = 0.8,\ \alpha = 0.5,\ E\varepsilon_t^2 = \sigma,\ \sigma = 0.10$$



$\varepsilon_t = 2\sigma$

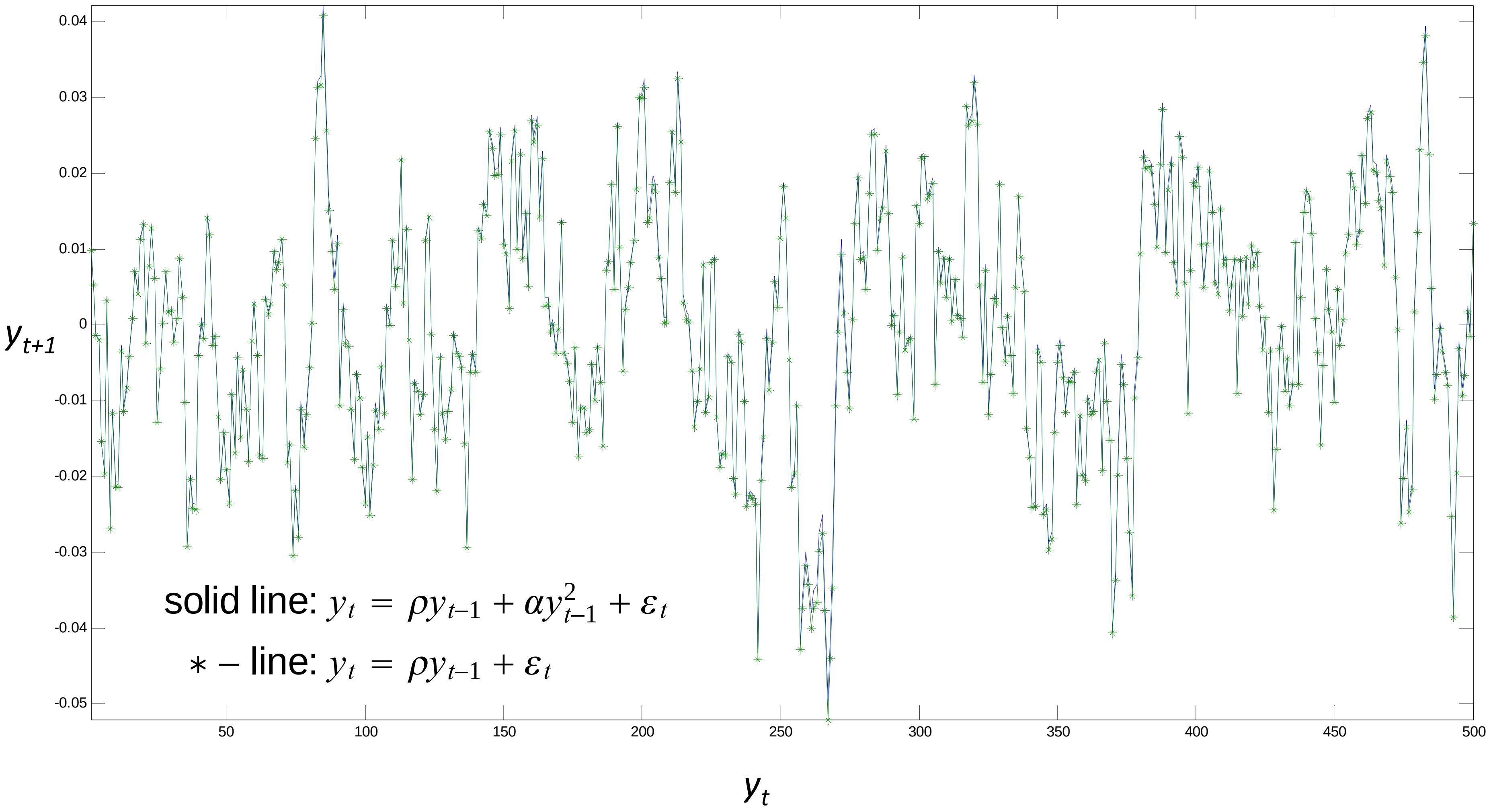
$\varepsilon_t = 0$

$\varepsilon_t = -2\sigma$

45 degree line

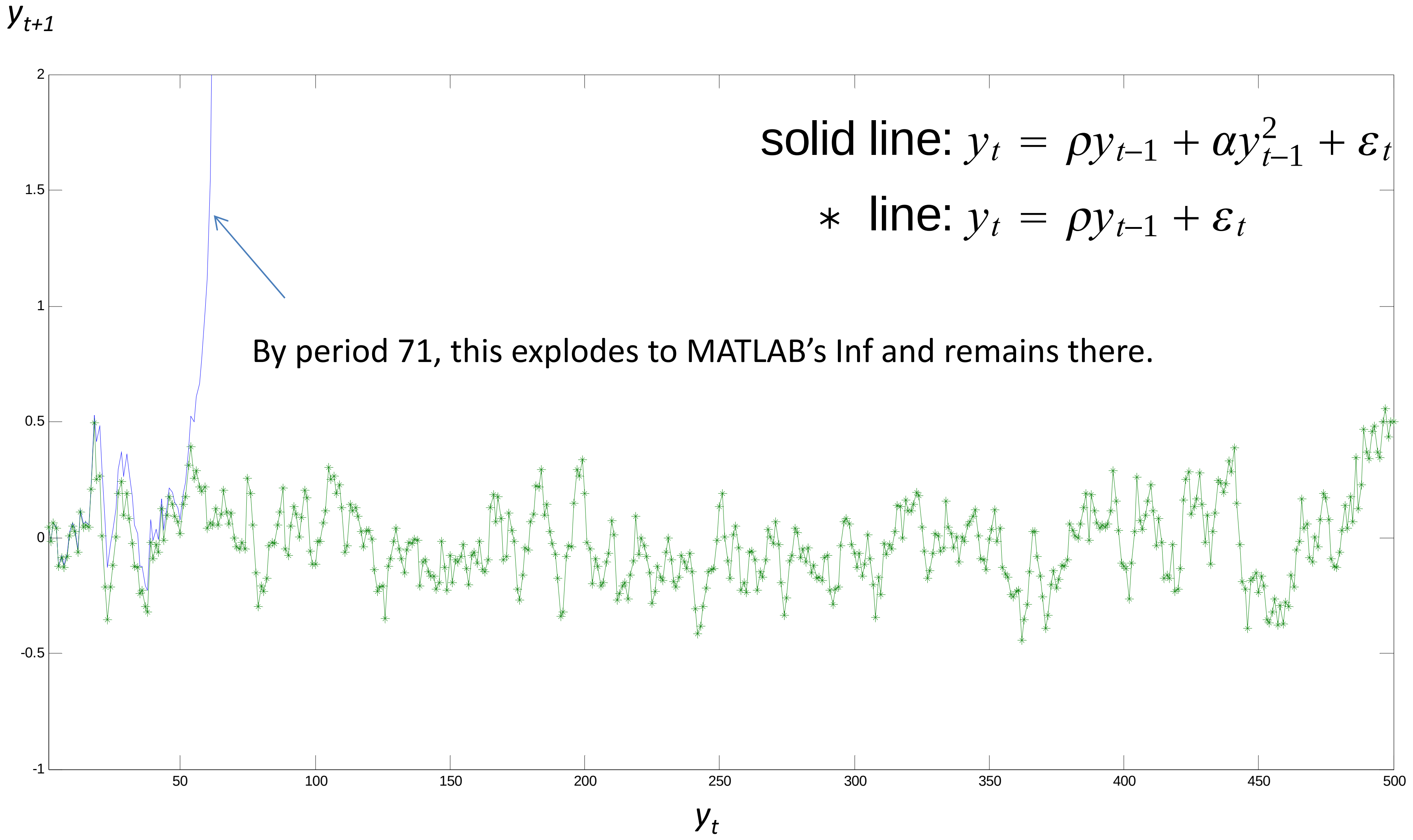
After a few big positive shocks, this process will explode. Explosion big because of convexity.

Simulations, std dev = σ/10



Two lines virtually the same for small shocks.

Simulations , std dev = σ



solid line: $y_t = \rho y_{t-1} + \alpha y_{t-1}^2 + \varepsilon_t$

$*$ line: $y_t = \rho y_{t-1} + \varepsilon_t$

By period 71, this explodes to MATLAB’s Inf and remains there.

Two lines wildly different for large shocks…….naive simulations explode!

Pruning

- Procedure for simulating artificial data, using second order Taylor series expansion of simulated data as a function of the shocks and initial conditions[1].

- First, draw a sequence, $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$

- Next, solve for $\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_T$ in the linear component of the process:

$$\tilde{y}_t = \rho \tilde{y}_{t-1} + \varepsilon_t$$

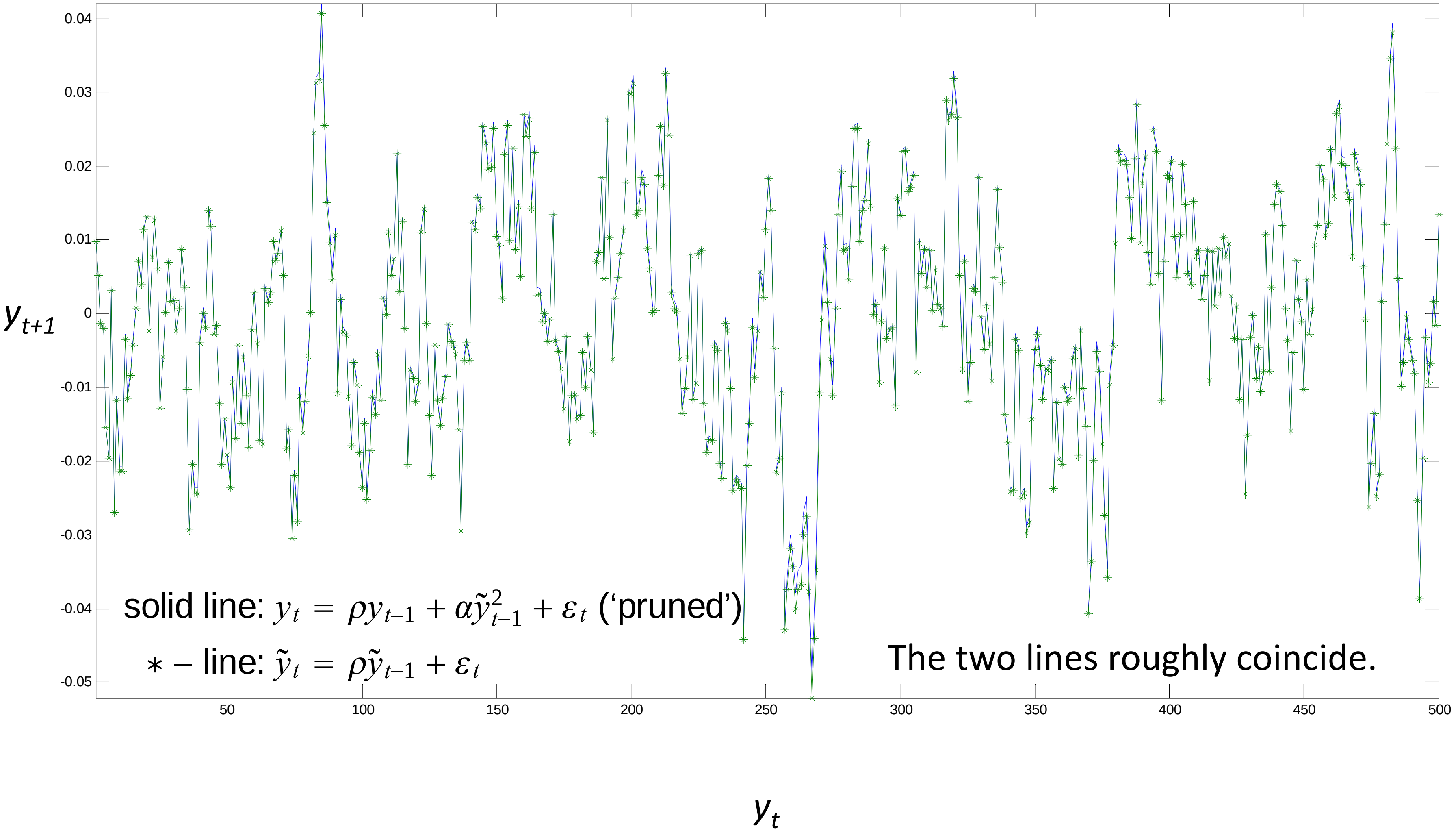
- The ('pruned') solution to the 2nd order difference equation is $y_1, y_2, \ldots, y_T$ in

$$y_t = \rho y_{t-1} + \alpha \tilde{y}_{t-1}^2 + \varepsilon_t$$

- Note that the y_t's cannot explode.

[1] For pruning with higher order perturbations, see Lombardo (2011).

Simulations, std dev = σ/10

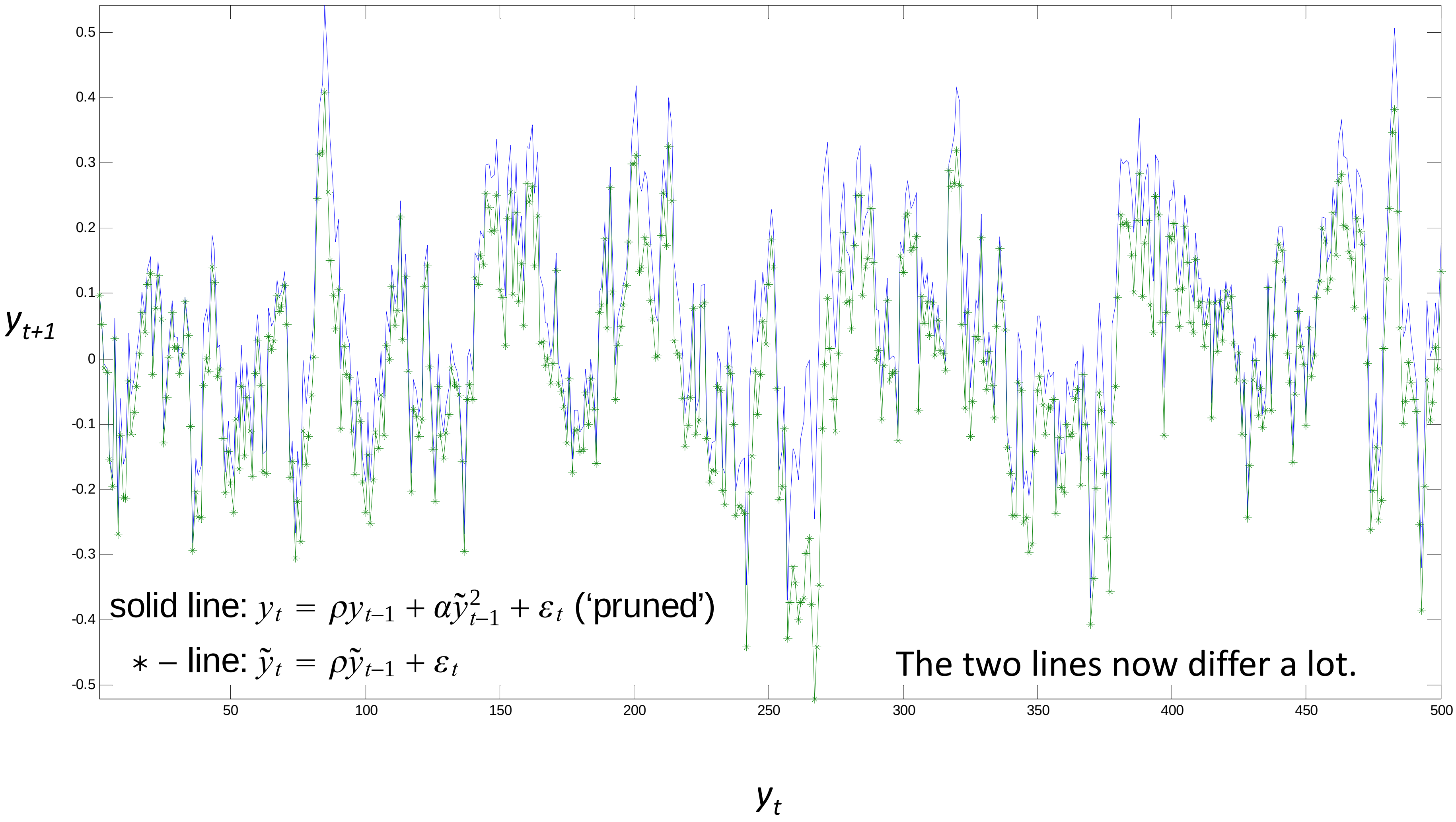


solid line: $y_t = \rho y_{t-1} + \alpha \tilde{y}_{t-1}^2 + \varepsilon_t$ (‘pruned’)

$* -$ line: $\tilde{y}_t = \rho \tilde{y}_{t-1} + \varepsilon_t$

The two lines roughly coincide.

Simulations, std dev = σ



The two lines now differ a lot.

Next

done

- Stochastic simulations
  - Mapping from shocks and initial conditions to data.
  - 2nd order approximation and pruning.
  - Naïve simulation of 2nd order approximation, versus pruning.
  - Spurious steady state.
  - Extended example.

done

- Impulse response functions
  - Conditional impulse response function (IRF).
  - Unconditional IRF.

Impulse Response Function

- (Conditional) impulse response function:
  - Impact of a shock on expectation of future variables.

$$E[y_{t+j}|\Omega_{t-1}, \text{shock}_t \neq 0] - E[y_{t+j}|\Omega_{t-1}, \text{shock}_t = 0],\ j = 0, 1, 2, \ldots$$

  - Impulse responses are useful for building intuition about the economic properties of a model.
  - When you condition on initial information, can compare responses in recessions versus booms.
  - Can also used for model estimation, if you have the empirical analogs from VAR analysis.

Impulse Response Function, cnt'd

- Example:

$$y_t = \overbrace{\rho y_{t-1}}^{\Omega_{t-1}} + \varepsilon_t$$

- Obviously:

$$\overbrace{E[y_t | \Omega_{t-1}, \varepsilon_t \neq 0]}^{\rho y_{t-1} + \varepsilon_t} - \overbrace{E[y_t | \Omega_{t-1}, \varepsilon_t = 0]}^{\rho y_{t-1}} = \varepsilon_t$$

- Also

$$\begin{aligned} y_{t+1} &= \rho y_t + \varepsilon_{t+1} \\ &= \rho^2 y_{t-1} + \varepsilon_{t+1} + \rho \varepsilon_t \end{aligned}$$

- So that:

$$E[y_{t+1} | \Omega_{t-1}, \varepsilon_t] = \rho^2 y_{t-1} + \rho \varepsilon_t, \; E[y_{t+1} | \Omega_{t-1}, \varepsilon_t = 0] = \rho^2 y_{t-1}$$

$$\rightarrow E[y_{t+1} | \Omega_{t-1}, \varepsilon_t] - E[y_{t+1} | \Omega_{t-1}, \varepsilon_t = 0] = \rho \varepsilon_t$$

- In general:

$$E[y_{t+j} | \Omega_{t-1}, \varepsilon_t \neq 0] - E[y_{t+j} | \Omega_{t-1}, \varepsilon_t = 0] = \rho^j \varepsilon_t$$

Impulse Responses, cnt'd

- Easy in the linear system!
  - Impulse responses not even a function of Ω_{t-1}
- Different story in our 2nd order approximation, especially because of the spurious steady state..
- Example:

Same form as our 2nd order approximation

$$y_t = \rho y_{t-1} + \alpha y_{t-1}^2 + \varepsilon_t$$

- Obviously:

Easy…

$$\overbrace{E[y_t|\Omega_{t-1},\varepsilon_t]}^{\overbrace{\rho y_{t-1}+\alpha y_{t-1}^2}^{\Omega_{t-1}}+\varepsilon_t} - \overbrace{E[y_t|\Omega_{t-1},\varepsilon_t = 0]}^{\rho y_{t-1}+\alpha y_{t-1}^2} = \varepsilon_t$$

IRF, cont'd

- One-period-out IRF
  - Note:

$$y_{t+1} = \rho \overbrace{[\rho y_{t-1} + \alpha y_{t-1}^2 + \varepsilon_t]}^{y_t} + \alpha \overbrace{[\rho y_{t-1} + \alpha y_{t-1}^2 + \varepsilon_t]^2}^{y_t^2} + \varepsilon_{t+1}$$

  - Then,

$$E[y_{t+1}|\Omega_{t-1}, \varepsilon_t \neq 0] = \rho[\rho y_{t-1} + \alpha y_{t-1}^2 + \varepsilon_t] + \alpha\Big[(\rho y_{t-1} + \alpha y_{t-1}^2)^2 + \varepsilon_t^2 + 2(\rho y_{t-1} + \alpha y_{t-1}^2)\varepsilon_t\Big] + \overbrace{0}^{E[\varepsilon_{t+1}|\Omega_{t-1},\varepsilon_t]}$$

$$E[y_{t+1}|\Omega_{t-1}, \varepsilon_t = 0] = \rho[\rho y_{t-1} + \alpha y_{t-1}^2] + \alpha(\rho y_{t-1} + \alpha y_{t-1}^2)^2$$

  - So,

$$E[y_{t+1}|\Omega_{t-1}, \varepsilon_t] - E[y_{t+1}|\Omega_{t-1}, \varepsilon_t = 0] = \rho\varepsilon_t + \alpha\varepsilon_t^2 + 2\alpha(\rho y_{t-1} + \alpha y_{t-1}^2)\varepsilon_t$$

  - Ouch! Much more complicated…is a function of elements of Ω_{t-1}

IRF’s, cnt’d

- Too hard to compute IRF’s by analytic formulas, when equations are not linear.

$$y_t = \rho y_{t-1} + \alpha y_{t-1}^2 + \varepsilon_t$$

- What we need:
  – Fix a value for $\Omega_{t-1} = \underbrace{\rho y_{t-1} + \alpha y_{t-1}^2}_{\text{our example}}$
  – Compute:

$$E[y_{t+j}|\Omega_{t-1}, \varepsilon_t],\ j = 1, 2, 3, \ldots, T, \text{ for a given value of } \varepsilon_t > 0$$

$$E[y_{t+j}|\Omega_{t-1}, \varepsilon_t = 0],\ j = 1, 2, 3, \ldots, T.$$

  – Subtract:

$$E[y_{t+j}|\Omega_{t-1}, \varepsilon_t] - E[y_{t+j}|\Omega_{t-1}, \varepsilon_t = 0],\ j = 1, 2, 3, \ldots, T$$

IRF’s, cnt’d

- Computational strategy
  - From a random number generator, draw:

$$\varepsilon_{t+1}^{(1)}, \varepsilon_{t+2}^{(1)}, \ldots\ldots, \varepsilon_{t+T}^{(1)}$$

  - Using the stochastic equation, $\rho y_{t-1} + \alpha y_{t-1}^2$ and the given ε_t compute (by pruning)

$$y_{t+1}^{(1)}, y_{t+2}^{(1)}, \ldots, y_{t+T}^{(1)}$$

  - Repeat this, over and over again, *R* (big) times, to obtain

$$y_{t+1}^{(1)}, y_{t+2}^{(1)}, \ldots, y_{t+T}^{(1)}$$

$$\ldots$$

$$y_{t+1}^{(R)}, y_{t+2}^{(R)}, \ldots, y_{t+T}^{(R)}$$

  - Finally,

$$E[y_{t+j}|\Omega_{t-1}, \varepsilon_t] = \frac{1}{R}\sum_{l=1}^{R} y_{t+j}^{(l)},\ j = 1, 2, \ldots, T$$

IRF's, cnt'd

- To get $E[y_{t+j}|\Omega_{t-1}, \varepsilon_t = 0], \, j = 1, 2, 3, \ldots, T$, just repeat the preceding calculations, except set $\varepsilon_t = 0$

- To do the previous calculations, need *R* and *T*.
  - Dynare will do these calculations.
  - In the stoch_simul command,
    - *R* is set by including the argument, *replic=R*.
    - *T* is set by including *irf=T*.

Next

done

- Stochastic simulations
  - Mapping from shocks and initial conditions to data.
  - 2nd order approximation and pruning.
  - Naïve simulation of 2nd order approximation, versus pruning.
  - Spurious steady state.
  - Extended example.

done

- Impulse response functions
  - Conditional impulse response function (IRF).

done

  - Unconditional IRF.

A Different Type of Impulse Response Function

- The previous concept of an impulse response function required specifying the information set, Ω_{t-1} .
  - How to specify this is not often discussed…in part because with linear solutions it is irrelevant.
  - With nonlinear solutions, Ω_{t-1} makes a difference.
  - How to choose Ω_{t-1} ?
  - One possibility: nonstochastic steady state.
  - Another possibility: stochastic mean.

Unconditional Impulse Response Functions

Unconditional IRF

- Note that

$$E[y_{t+j}|\Omega_{t-1},\varepsilon_t] - E[y_{t+j}|\Omega_{t-1},\varepsilon_t = 0],\ j = 1,2,3,\ldots,T$$

  - is a function of Ω_{t-1} (i.e., it is a random variable)
  - Evaluate the mean of this random variable as follows:
    - Suppose there is date *0*, date *t* and date *T*, where *T>t* and *t* is itself large.
    - Draw R sets of shocks (no need to draw ε_t)

$$\varepsilon_0^{(1)},\varepsilon_1^{(1)},\ldots,\varepsilon_{t-1}^{(1)},\varepsilon_{t+1}^{(1)},\varepsilon_{t+2}^{(1)},\ldots\ldots,\varepsilon_{t+T}^{(1)}$$

$$\ldots$$

$$\varepsilon_0^{(R)},\varepsilon_1^{(R)},\ldots,\varepsilon_{t-1}^{(R)},\varepsilon_{t+1}^{(R)},\varepsilon_{t+2}^{(R)},\ldots\ldots,\varepsilon_{t+T}^{(R)}$$

Unconditional IRF

- Using $\varepsilon_t \neq 0, \; \varepsilon_t = 0$ together with

$$
\begin{aligned}
&\varepsilon_0^{(1)},\varepsilon_1^{(1)},\ldots,\varepsilon_{t-1}^{(1)},\varepsilon_{t+1}^{(1)},\varepsilon_{t+2}^{(1)},\ldots\ldots,\varepsilon_{t+T}^{(1)}\\
&\ldots\\
&\varepsilon_0^{(R)},\varepsilon_1^{(R)},\ldots,\varepsilon_{t-1}^{(R)},\varepsilon_{t+1}^{(R)},\varepsilon_{t+2}^{(R)},\ldots\ldots,\varepsilon_{t+T}^{(R)}
\end{aligned}
$$

- Compute *two* sets (by pruning)

$$
\begin{aligned}
&y_0^{(1)},y_1^{(1)},\ldots,y_{t-1}^{(1)},y_t^{(1)},y_{t+1}^{(1)},y_{t+2}^{(1)},\ldots,y_{t+T}^{(1)}\\
&\ldots\\
&y_0^{(R)},y_1^{(R)},\ldots,y_{t-1}^{(R)},y_t^{(R)},y_{t+1}^{(R)},y_{t+2}^{(R)},\ldots,y_{t+T}^{(R)}
\end{aligned}
$$

- The period *t+j* IRF is computed by averaging across *l=1,…,R*, for given *t+j, j=0,1,…,T*. Then, subtract, as before.
- In Dynare, *t* is set with *drop=t* parameter in stoch_simul command.

Conclusion

- For modest US-sized fluctuations and for aggregate quantities, it may be reasonable to work with first order perturbations.
  - This assumption deserves much further testing.
  - Can do this by studying the error function.
  - Also, try fancier approximations and see if it changes your results.

- One alternative to first order perturbations is higher order perturbations.
  - These must be handled with care, as they are characterized by spurious steady states, which may be the transition point to unstable dynamics.
  - Must do some sort of pruning to compute IRF’s, or just to simulate data.

- An alternative is to apply projection methods.
  - Perhaps these have less problems with spurious steady states.
  - Computation of solutions is more cumbersome in this case.

- First order perturbation: linearize (or, log-linearize) equilibrium conditions around non-stochastic steady state and solve the resulting system.
  - This approach assumes ‘certainty equivalence’. Ok, as a first order approximation.

List of endogenous variables determined at *t*

Solution by Linearization

- (log) Linearized Equilibrium Conditions:

$$E_t[\alpha_0 z_{t+1} + \alpha_1 z_t + \alpha_2 z_{t-1} + \beta_0 s_{t+1} + \beta_1 s_t] = 0$$

- Posit Linear Solution:

$$s_t - P s_{t-1} - \epsilon_t = 0.$$

$$z_t = A z_{t-1} + B s_t$$

Exogenous shocks

- To satisfy equil conditions, *A* and *B* must:

$$\alpha_0 A^2 + \alpha_1 A + \alpha_2 I = 0, \;\; F = (\beta_0 + \alpha_0 B)P + [\beta_1 + (\alpha_0 A + \alpha_1)B] = 0$$

- If there is exactly one *A* with eigenvalues less than unity in absolute value, that’s the solution. Otherwise, multiple solutions.

- Conditional on *A*, solve linear system for *B*.