Numerical Methods in Economics
MIT Press, 1998

Lecture **Notes: Finite-Difference Methods**

March 18, 2020

Classification of Ordinary Differential Equations

- A *first-order ordinary differential equation* (ODE) has the form

$$\frac{dy}{dx} = f(y, x), \tag{10.1.1}$$

  where $f : R^{n+1} \to R^n$ and the unknown is $y(x) : [a, b] \subset R \to R^n$.

  – When $n = 1$, we have a single differential equation

  – If $n > 1$, (10.1.1) is a system of differential equations.

- Need boundary conditions to fix unknown function $y(x)$.

  – *Initial value problem* (IVP): impose $y(x_0) = y_0$ for some $x_0 \in [a, b]$

  – *Two-point boundary value problem:* impose n conditions

$$g_i(y(a)) = 0, \quad i = 1, \cdots, n', \tag{10.1.2}$$

$$g_i(y(b)) = 0, \quad i = n' + 1, \cdots, n,$$

  where $g : R^n \to R^n$.

  – General BVP: impose

$$g_i(y(x_i)) = 0 \tag{10.1.3}$$

  for a set of points, x_i, $a \leq x_i \leq b$, $\quad 1 \leq i \leq n$.

- All problems have form (10.1.2). For example, replace

$$\frac{d^2y}{dx^2} = g(\frac{dy}{dx}, y, x)$$

with

$$\frac{dy}{dx} = z, \quad \frac{dz}{dx} = g(z, y, x),$$

- IVP and BVP definitions apply to discrete-time systems.

Finite-Difference Methods for IVPs

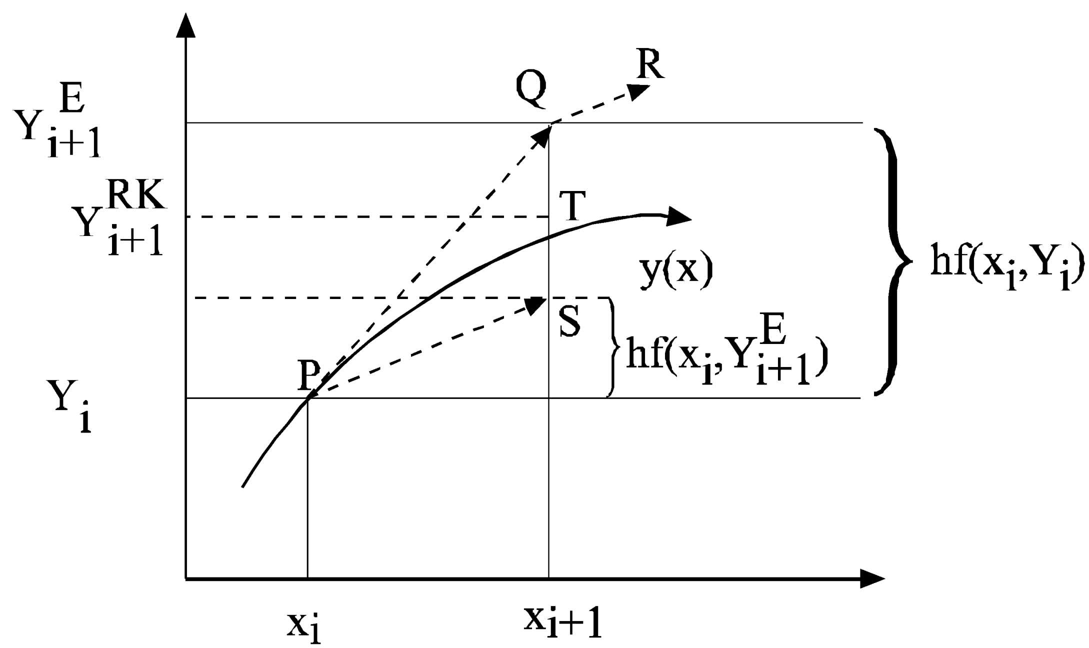
- Consider the IVP
$$y' = f(x, y), \quad y(x_0) = y_0. \tag{10.3.1}$$
- Specify a grid for x, $x_i = x_0 + ih$, $i = 0, 1, \cdots, N$
- Objective: find Y_i which approximates $y(x_i)$.
- Construct a difference equation on the grid
  - An explicit scheme $Y_{i+1} = F(Y_i, Y_{i-1}, \cdots, x_{i+1}, x_i, \cdots)$,
  - or an implicit scheme $Y_{i+1} = F(Y_{i+1}, Y_i, Y_{i-1}, \cdots, x_{i+1}, x_i, \cdots)$
- Y_0 is fixed by the initial condition, $Y_0 = y(x_0) = y_0$.
- Solve difference scheme, and hope that $Y_i \doteq y(x_i)$
- Find scheme using as few grid points as possible

Euler Method

- Algorithm:

$$Y_0 = y(x_0) = y_0; \qquad\qquad Y_{i+1} = Y_i + hf(x_i, Y_i)$$

- Geometry of Euler's method
  - Current iterate is $P = (x_i, Y_i)$; $y(x)$ is the true solution
  - At $P, y'(x_i)$ is the tangent vector $\vec{PQ}$. Euler's method follows $\vec{PQ}$ until $x = x_{i+1}$ at Q.
  - Euler estimate of $y(x_{i+1})$ is Y^E_{i+1}.



- Convergence Theorem:

Theorem 1 *Suppose that the solution to* $y'(x) = f(x, y(x))$, $y(0) = y_0$, *is* C^3 *on* $[a, b]$, *that* f *is* C^2, *and that* f_y *and* f_{yy} *are bounded for all* y *and* $a \leq x \leq b$. *Then the error of the Euler scheme with step size* h *is* $\mathcal{O}(h)$; *that is, it can be expressed as*

$$y(x_i) - Y_i = D(x_i)h + \mathcal{O}(h^2)$$

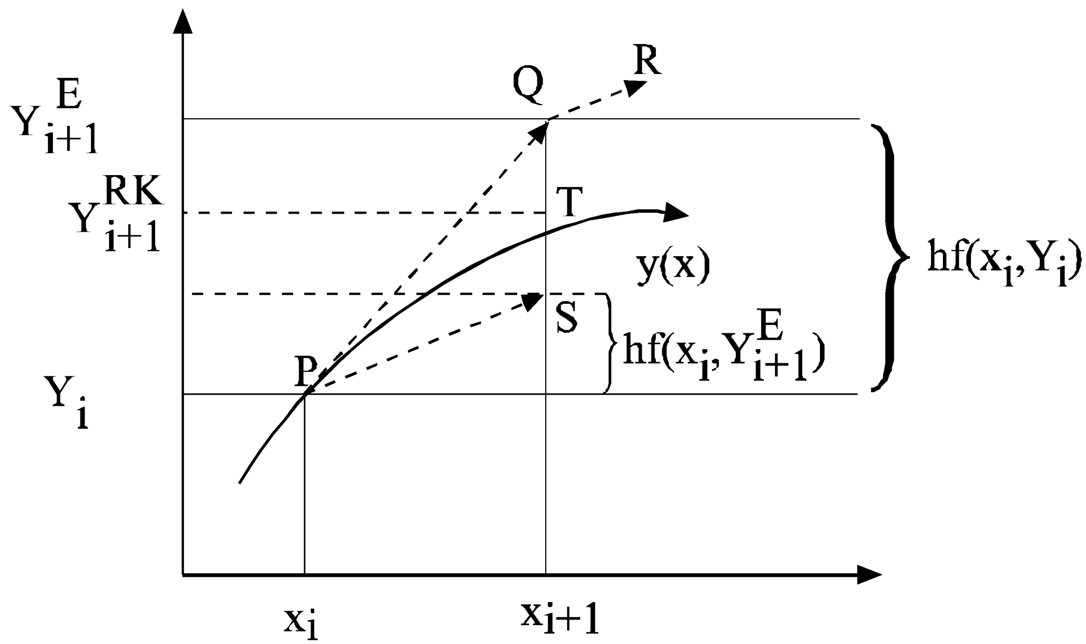
where $D(x)$ *is bounded on* $[a, b]$ *and solves the differential equation*

$$D'(x) = f_y(x, y(x))\, D(x) + \frac{1}{2}y''(x), \quad D(x_0) = 0$$

Runge-Kutta Methods

- First-order Runge-Kutta (RK1)
  - Euler estimate of $y(x_{i+1})$ is Y_{i+1}^E.
  - Slope of vector field at (x_i, Y_{i+1}^E) is $f(x_i, Y_{i+1}^E)$, not $f(x_i, Y_i^E)$ as assumed by Euler
  - Slope at (x_i, Y_{i+1}^E) says Y_{i+1} should be $Y_i + h\ f(x_i, Y_{i+1}^E)$, point S in figure 10.1
  - RK1 takes average of these two approximations:

$$Y_{i+1} = Y_i + \frac{h}{2}\left[f(x_i, Y_i) + f(x_{i+1}, Y_i + hf(x_i, Y_i))\right] \tag{10.3.9}$$



- RK1 asymptotic error is $\mathcal{O}(h^2)$
- RK1 evaluates f twice per step
- Typo: To the left of point S, the expression should be $hf\left(x_{i+1}, Y_{i+1}^E\right)$.
- Fourth-order Runge-Kutta (RK4)

$$\begin{aligned} z_1 &= f(x_i, Y_i), \\ z_2 &= f(x_i + \tfrac{1}{2}h,\ Y_i + \tfrac{1}{2}hz_1) \\ z_3 &= f(x_i + \tfrac{1}{2}h,\ Y_i + \tfrac{1}{2}hz_2) \\ z_4 &= f(x_i + h,\ Y_i + hz_3) \\ Y_{i+1} &= Y_i + \tfrac{h}{6}[z_1 + 2z_2 + 2z_3 + z_4] \end{aligned} \tag{10.3.10}$$

- RK4 asymptotic error is $\mathcal{O}(h^5)$
- RK4 evaluates f four times per step

Systems of Differential Equations

$$
\begin{aligned}
y_1'(x) &= f_1(x, y_1, y_2, \cdots y_n) \\
&\vdots \\
y_n'(x) &= f_n(x, y_1, y_2, \cdots, y_n)
\end{aligned}
\tag{10.3.11}
$$

- Euler:

$$Y_\ell^{i+1} = Y_\ell^i + hf_\ell\,(x_i, Y_1^i, \cdots, Y_n^i),\ \ \ell = 1, \cdots, n \tag{10.3.12}$$

- RK1:

$$Y^{i+1} = Y^i + \frac{h}{2}\,\left[f(x_i, Y^i) + f(x_{i+1}, Y^i + hf(x_i, Y^i))\right] \tag{10.3.14}$$

- RK4:

$$
\begin{aligned}
z^1 &= f(x_i, Y^i) \\
z^2 &= f(x_i + \tfrac{1}{2}h,\ Y^i + \tfrac{1}{2}hz^1) \\
z^3 &= f(x_i + \tfrac{1}{2}h,\ Y^i + \tfrac{1}{2}hz^2) \\
z^4 &= f(x_i + h,\ Y^i + hz^3) \\
Y^{i+1} &= Y^i + \tfrac{h}{6}[z^1 + 2z^2 + 2z^3 + z^4]
\end{aligned}
\tag{10.3.15}
$$

Spence Signaling Equilibrium

- Education signalling model implies the nonlinear equation

$$N'(y) = \frac{N(y)^{-1} - \alpha N(y) y^{\alpha-1}}{y^{\alpha}} \tag{10.4.3}$$

with initial condition $N(y_m) = n_m$, and closed-form solution

$$N(y) = y^{-\alpha} \left(\frac{2(y^{1+\alpha} + D)}{1+\alpha} \right)^{1/2}, \ D = \frac{1+\alpha}{2} \left(\frac{n_m}{y_m^{-\alpha}} \right)^2 - y_m^{1+\alpha} \tag{10.4.5}$$

Table 10.1: Signalling Model Errors

	Euler			RK1			RK4	
h:	.01	.001	.0001	.01	.001	.0001	.01	.001
$y - y_m$								
0.1	3(-2)	1(-3)	1(-4)	1(-3)	1(-4)	1(-6)	3(-3)	2(-6)
0.2	2(-2)	1(-3)	1(-4)	1(-3)	5(-4)	1(-6)	2(-3)	1(-6)
0.4	1(-2)	7(-4)	7(-5)	4(-4)	3(-4)	4(-7)	1(-3)	6(-7)
1.0	6(-3)	4(-4)	4(-5)	2(-4)	1(-4)	2(-7)	4(-4)	3(-7)
2.0	4(-3)	3(-4)	3(-5)	1(-4)	7(-5)	1(-7)	2(-4)	1(-7)
10.0	1(-3)	1(-4)	1(-5)	2(-5)	2(-6)	0(-7)	6(-5)	0(-7)
time:	.11	1.15	9.17	.16	1.49	14.4	.27	2.91

Boundary Value Problems for ODEs: Shooting

- Consider the BVP

$$\begin{aligned} \dot{x} &= f(x, y, t), \\ \dot{y} &= g(x, y, t), \\ x(0) &= x^0, \quad y(T) = y^T \\ x &\in R^n, \ y \in R^m \end{aligned} \tag{1}$$

- For any guess $y(0) = y^0$, solve IVP in (10.5.2)

$$\begin{aligned} \dot{x} &= f(x, y, t) \\ \dot{y} &= g(x, y, t) \\ x(0) &= x^0, \quad y(0) = y^0 \end{aligned} \tag{2}$$

  - Let $Y(T, y^0)$ denote the resulting value of $y(T)$
  - $Y(T, y^0)$ depends on y^0
  - Find correct value of y^0 by solving the nonlinear equation $Y(T, y^0) = y^T$

- Programming strategy

  - write procedure which computes $Y(T, y^0) - y^T$ given input y^0.
  - send that routine to a nonlinear equation solver to solve $Y(T, y^0) - y^T = 0$.

Life-Cycle Model of Consumption and Labor Supply

- Simple life-cycle model:

$$\begin{aligned} &\max_c \int_0^T e^{-\rho t} u(c)\, dt, \\ &\dot{A} \quad = f(A) + w(t) - c(t) \\ &A(0) = A(T) = 0 \end{aligned} \tag{10.6.10}$$

  - $u(c)$ is concave utility function over consumption c
  - $w(t)$ is wage rate at t
  - $A(t)$ is assets at time t
  - $f(A)$ is return on invested assets.

- Hamiltonian is $H = u(c) + \lambda(f(A) + w(t) - c)$.
  - Costate equation: $\dot{\lambda} = \rho\lambda - \lambda f'(A)$.
  - First-order condition: $0 = u'(c) - \lambda$, implying consumption function $c = C(\lambda)$.
  - The final system:

$$\begin{aligned} \dot{A} &= f(A) + w - C(\lambda) \\ \dot{\lambda} &= \lambda(\rho - f'(A)) \end{aligned} \tag{10.6.11}$$

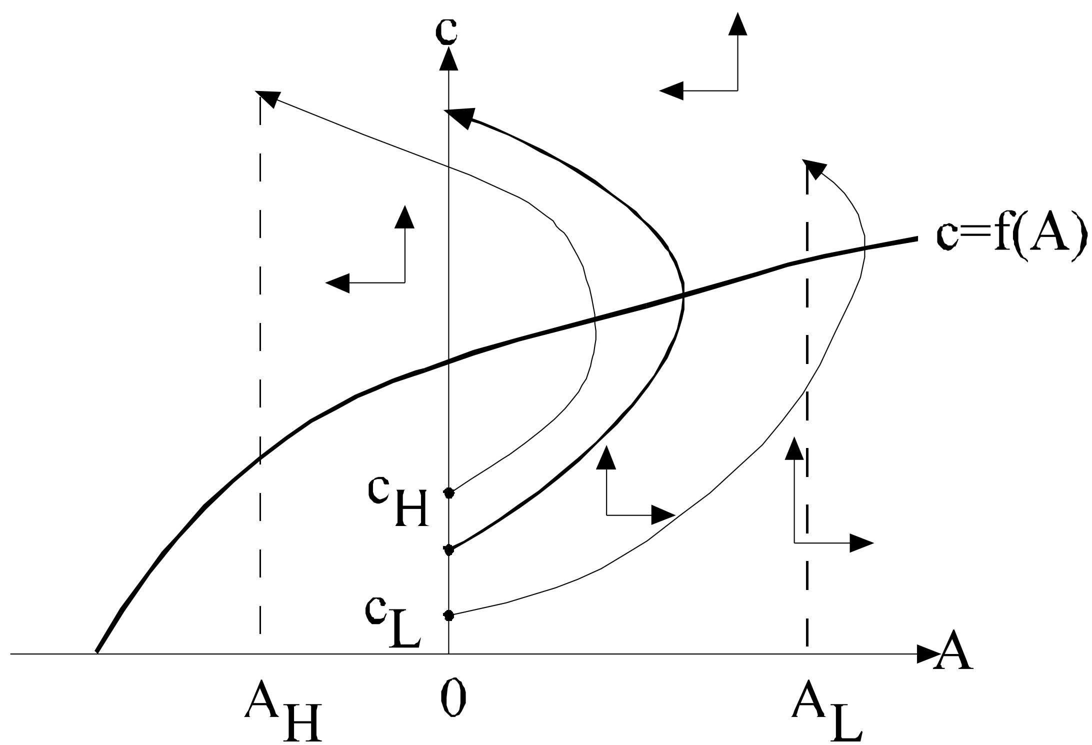
  with the boundary conditions

$$A(0) = A(T) = 0 \tag{10.6.12}$$

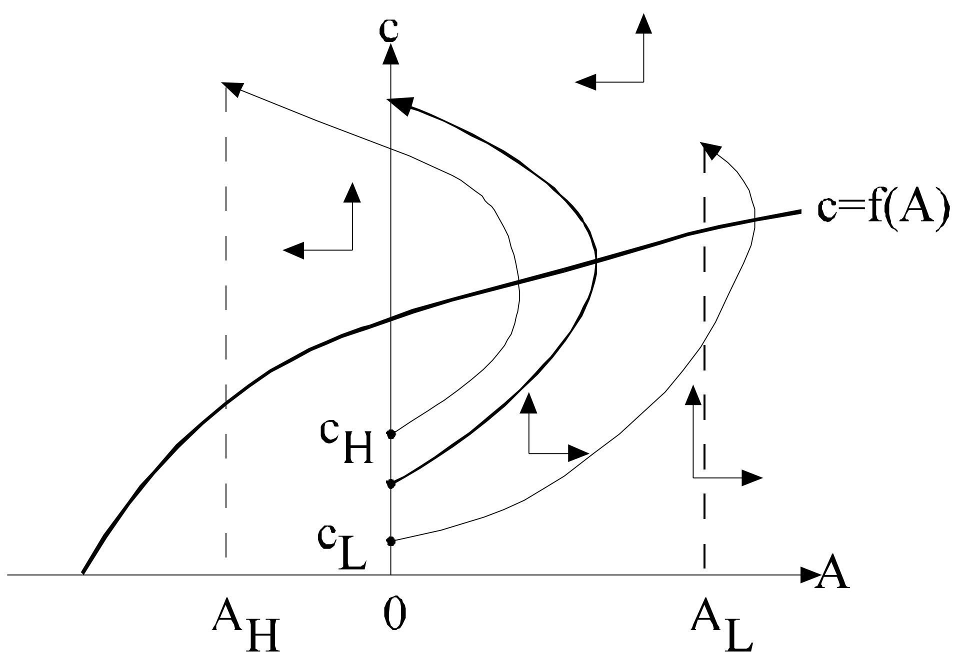
- Convert to a system for observable variables.
  - $u'(c) = \lambda$ implies that (10.6.11) can be replaced by

$$\begin{aligned} \dot{c} &= -\frac{u'(c)}{u''(c)}(f'(A) - \rho) \\ \dot{A} &= f(A) + w - c \end{aligned} \tag{10.6.13}$$

  - Phase diagram:



- Shooting: Consider implications of different $c(0)$.
  - If $A(T) < 0$ if $c(0) = c_H$, but $A(T) > 0$ if $c(0) = c_L$, then correct $c(0)$ lies between c_L and c_H.
  - Find true $c(0)$ by using the bisection method presented in algorithm 5.1.
- In general, if $A(T; c_0)$ is terminal wealth for initial consumption c_0, then find c_0 by solving nonlinear equation $A(T; c_0) = 0$.
- Phase diagram:



Optimal Growth

- Optimal control problem

$$\begin{aligned} &\max_c \int_0^\infty e^{-\rho t}\, u(c)\, dt \\ &\text{s.t.} \quad \dot{k} = f(k) - c \\ &\qquad\quad k(0) = k_0 \end{aligned} \tag{10.7.2}$$

  - k is the capital stock
  - c consumption
  - $f(k)$ the aggregate net production function

- $c(t)$ and $k(t)$ satisfy

$$\begin{aligned} \dot{c} &= \tfrac{u'(c)}{u''(c)} \ (\rho - f'(k)) \\ \dot{k} &= f(k) - c \end{aligned} \tag{10.7.3}$$

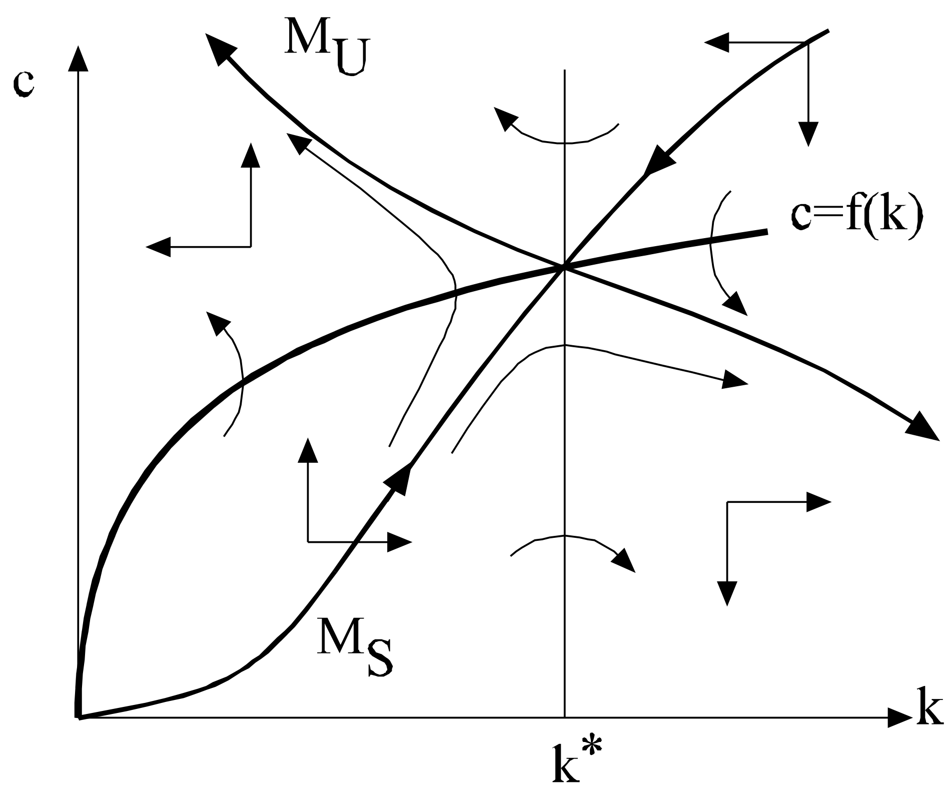
  and boundary conditions are

$$k(0) = k_0,\ 0 < \lim_{t\to\infty} \mid k(t) \mid < \infty$$

- Forward system:
  - We want to compute M_S, but shooting is numerically unstable.
  - Computing M_U would be easy since it attracts deviations
  - If we computed

$$\begin{aligned} \dot{c} &= \frac{u'(c)}{u''(c)} \ (\rho - f'(k)) \\ \dot{k} &= f(k) - c \end{aligned} \tag{10.7.3}$$

with initial condition near steady state, we would move along one of the branches of M_U



- Reversed system:
  - M_U in this system is numerically stable and is our consumption function
  - Find M_U by solving

$$\dot{c} = -\frac{u'(c)}{u''(c)} \ (\rho - f'(k))$$
$$\dot{k} = -(f(k) - c)$$

with initial conditions close to steady state.

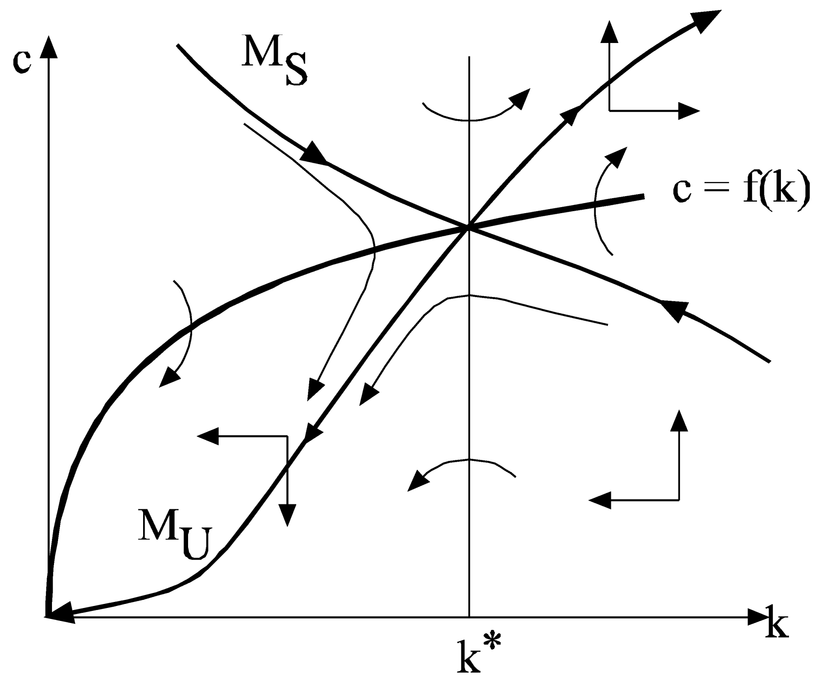


Table 10.2: Optimal Growth with Reverse Shooting

k	c		Errors	
		$h = 0.1$	$h = 0.01$	$h = 0.001$
0.2	0.10272	0.00034	3.1(-8)	3.1(-12)
0.5	0.1478	0.000025	3.5(-9)	4.1(-13)
0.9	0.19069	-0.001	-3.5(-8)	1.8(-13)
1.	0.2	0.	0.	0.
1.1	0.20893	-0.00086	-5.3(-8)	-1.2(-12)
1.5	0.24179	-0.000034	-1.8(-9)	-2.1(-14)
2.	0.2784	-9.8(-6)	-5.(-10)	3.6(-15)
2.5	0.31178	-5.(-6)	-2.6(-10)	-1.3(-14)
2.8	0.33068	-3.8(-6)	-1.9(-10)	-1.2(-14)