Optimization of Functions

Brent’s Method and Powell’s Method

Mahbuba Perveen
CSE-5400
November 10, 2020

Inverse Parabolic Interpolation

- A Golden section search is designed to handle the worst possible case of function minimization
- However, generally, sufficiently smooth functions are nicely parabolic near to the minimum
- Therefore, the parabola fitted through any three points may take us in a single leap to the minimum, or at least very near it.

Inverse Parabolic Interpolation (cont.)

- The formula for the abscissa x, that is the minimum of a parabola through three points f(a), f(b) and f(c) is:

$$x = b - \frac{1}{2}\frac{(b-a)^2[f(b)-f(c)] - (b-c)^2[f(b)-f(a)]}{(b-a)[f(b)-f(c)] - (b-c)[f(b)-f(a)]}$$

- This formula fails only if the three points are collinear
  - The denominator becomes zero = the minimum of the parabola is infinitely far away
- However, it is possible for x obtained using this equation to be either the parabolic maximum or minimum
  - Therefore, minimization scheme depending solely on this equation is not practical

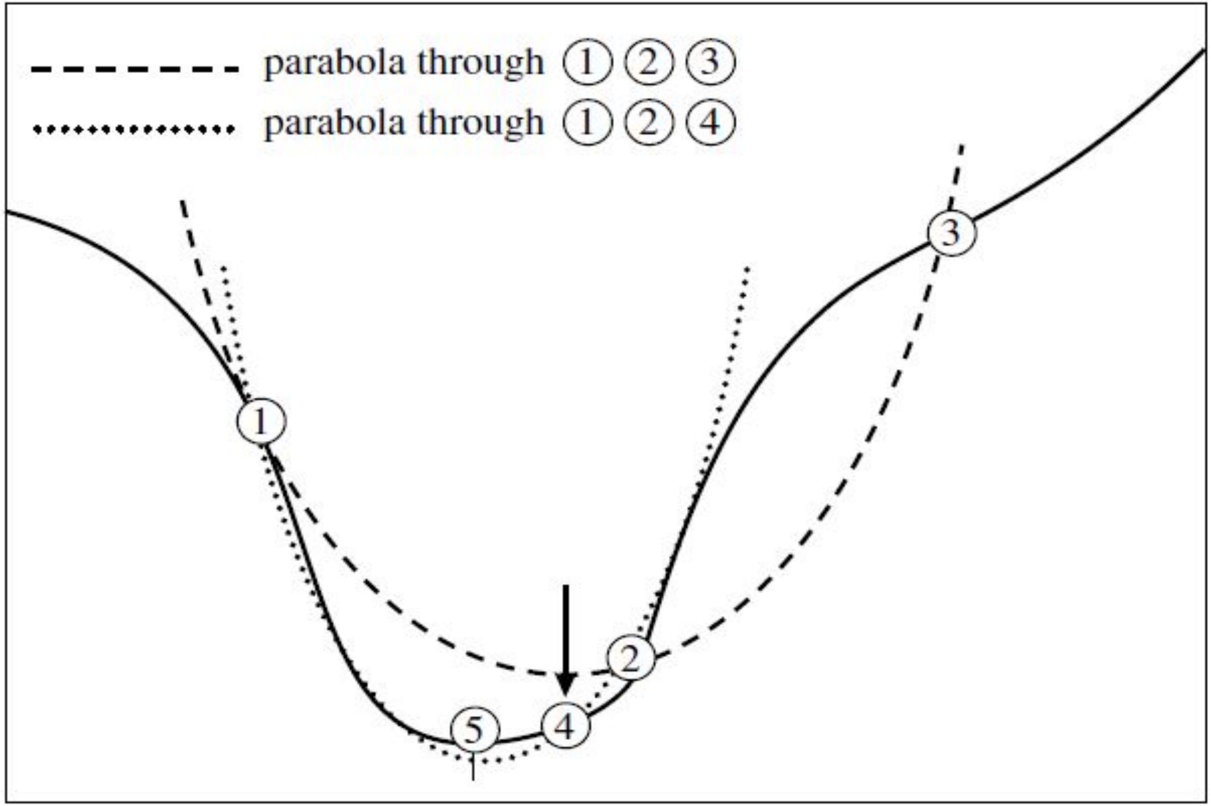


Figure 10.3.1. Convergence to a minimum by inverse parabolic interpolation. A parabola (dashed line) is drawn through the three original points 1,2,3 on the given function (solid line). The function is evaluated at the parabola's minimum, 4, which replaces point 3. A new parabola (dotted line) is drawn through points 1,4,2. The minimum of this parabola is at 5, which is close to the minimum of the function.

Brent’s Method

Brent's Method

- It's a hybrid method
- It combines root bracketing, bisection and inverse quadratic/parabolic interpolation
- The method relies on a sure-but-slow technique, like golden section search, when the function is not cooperative, but switches over to inverse quadratic interpolation when the function allows
- Careful attention is given when the function is being evaluated very near to the roundoff limit of inverse parabolic interpolation equation
- It's very robust in detecting a cooperative vs noncooperative function

Brent’s Method

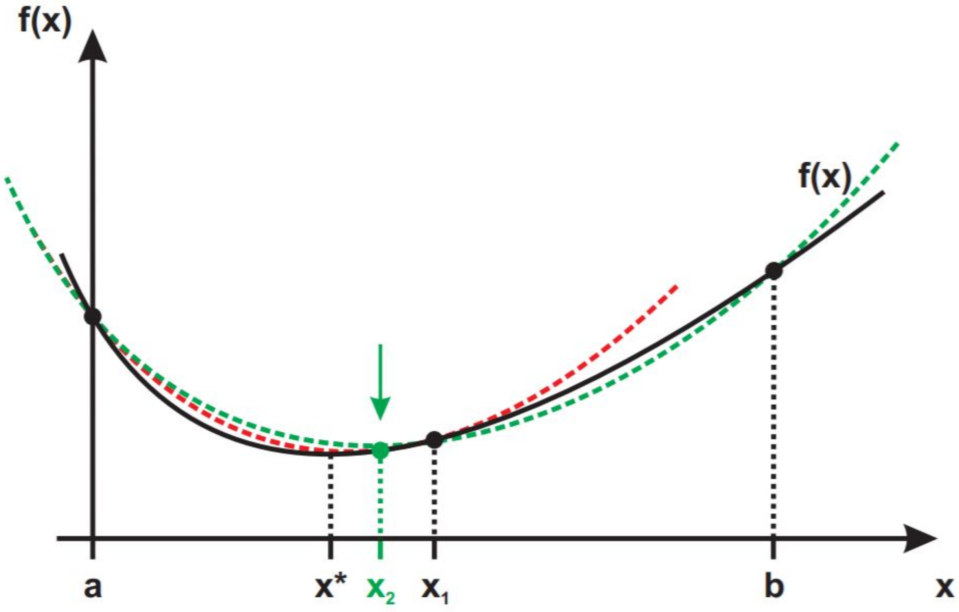


Figure: Brent’s method for finding minima

Brent's Method

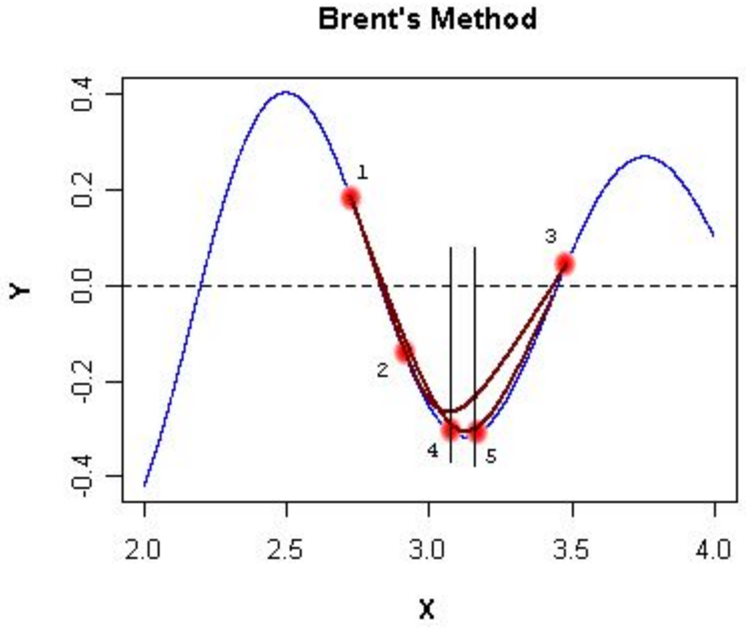
- We start with the initial interval [a,b] and compute the intersection point x_1
  - $x_1 = (a+b)/2$
- Then we compute a parabola that exactly contains the three points (a,f(a)), (b,f(b)) and $(x_1, f(x_1))$
  - The minimum of this parabola can be calculated and is denoted with x_2
- Then we replace b with x_2 and again compute a parabola through our new points
- The method is repeated until we reach convergence

Example

function f interpolating parabola and a bracketing triplet of abscissas (a b c)

a<b<c & f(a) >f(b) <f(c)

(1 2 3) (1 2 4) (1 2 5)



Powell’s Method

Line Methods in Multidimensions

We know (10.2 –10.4) how to minimize a function of one variable.

- If we start at a point P in N-dimensional space, and proceed from there in some vector direction n, then any function of N variables f(P) can be minimized along the line n by our one-dimensional methods.
- Methods: differ in choice of next direction n for next step.
- Black-box routine (linmin) minimizes a function along 1 direction

`linmin`: Given as input the vectors $\mathbf{P}$ and $\mathbf{n}$, and the function f, find the scalar λ that minimizes $f(\mathbf{P} + \lambda\mathbf{n})$. Replace $\mathbf{P}$ by $\mathbf{P} + \lambda\mathbf{n}$. Replace $\mathbf{n}$ by $\lambda\mathbf{n}$. Done.

A simple method for general N-d minimization

- Take the unit vectors e_0, e_1, … , e_{N-1} as a set of directions.
- We start at an initial guess x_1
- Then take our first optimization direction, represented by the arrow leaving x_1
- Using linmin, move along the first direction to its minimum
- Then from there along the second direction to its minimum
- Keep cycling through the whole set of directions as many times as necessary
- Stop when the function converges/stops decreasing

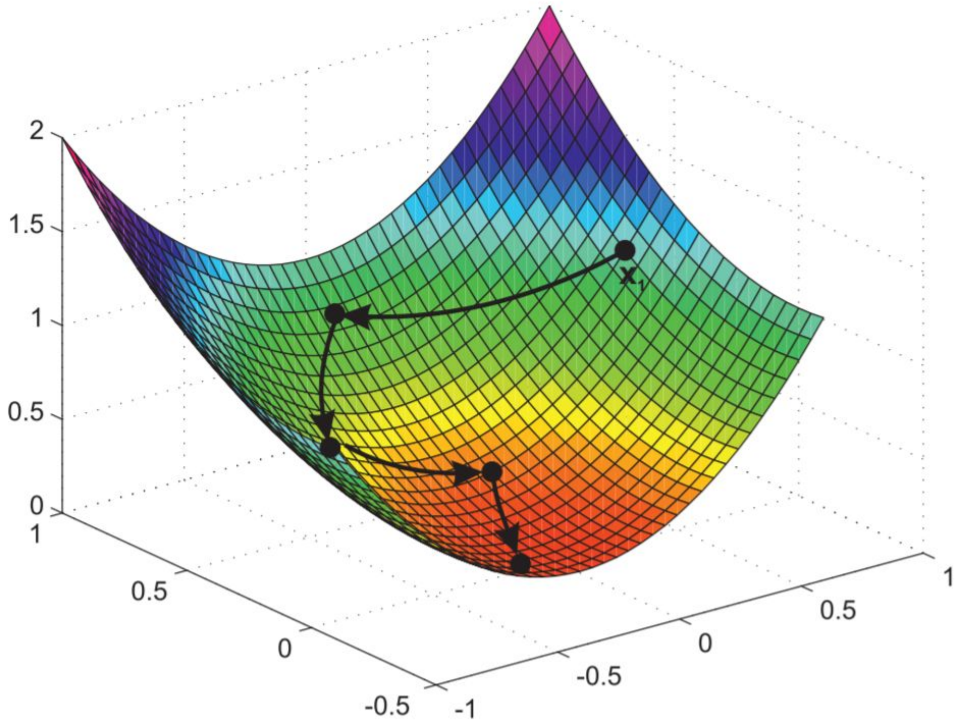
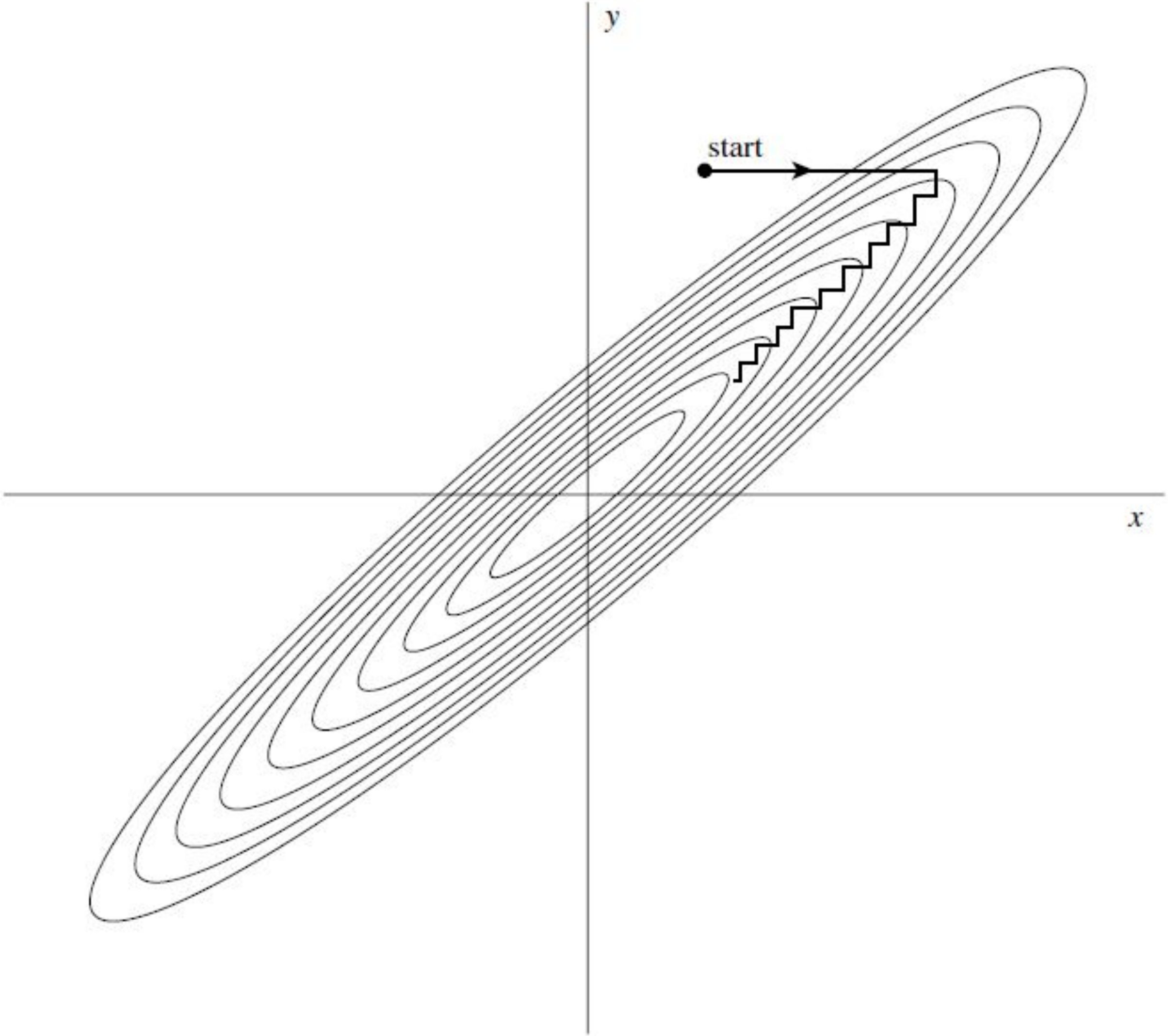


Figure: Finding minima in multidimensions

(cont.)

- **Drawback:**
    - For some functions this simple method will be very inefficient
    - Consider a function of two dimensions whose contour map (level lines) happens to define a long, narrow valley at some angle to the coordinate basis vectors.
    - (example figure in next slide)

    - Might fail: conjugate directions


y
start
x

Direction Set (Powell's) Methods in Multidimensions

- We need a better set of directions than the e_i's
- All **direction set methods** consist of prescriptions for updating the set of directions as the method proceeds
- They attempt to come up with a set that either:
  - Includes some very good directions that will take us far along narrow valleys
  - Or, includes some number of non-interfering directions
    - Special property: minimization along one is not spoiled by subsequent minimization along another
    - Thus, interminable cycling through the set of directions can be avoided

(cont.)

- Conjugate Direction
  - First, note that if we minimize a function along some direction u, then the gradient of the function must be perpendicular to u at the line minimum; if not, then there would still be a nonzero directional derivative along u.
- Next take some particular point P as the origin of the coordinate system with coordinates x.
- Then any function f can be approximated by its Taylor series

$$
\begin{aligned}
f(\mathbf{x}) &= f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots \\
&\approx c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}
\end{aligned}
\qquad (10.7.1)
$$

(cont.)

$$f(\mathbf{x}) = f(\mathbf{P}) + \sum_i \frac{\partial f}{\partial x_i} x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j} x_i x_j + \cdots$$
$$\approx c - \mathbf{b} \cdot \mathbf{x} + \frac{1}{2} \mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x} \qquad (10.7.1)$$

Where, $$c \equiv f(\mathbf{P}) \qquad \mathbf{b} \equiv -\nabla f|_{\mathbf{P}} \qquad [\mathbf{A}]_{ij} \equiv \left.\frac{\partial^2 f}{\partial x_i \partial x_j}\right|_{\mathbf{P}} \qquad (10.7.2)$$

In the approximation of (10.7.1), the gradient of f is easily calculated as

$$\nabla f = \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \qquad (10.7.3)$$

(cont.)

The gradient ∇f changes as we move along some direction

$$\delta(\nabla f) = \mathbf{A} \cdot (\delta \mathbf{x}) \tag{10.7.4}$$

Suppose that we have moved along some direction u to a minimum and now propose to move along some new direction v. The condition that motion along v not spoil our minimization along u is just that the gradient stay perpendicular to u, i.e., that the change in the gradient be perpendicular to u. By equation (10.7.4) this is just

$$0 = \mathbf{u} \cdot \delta(\nabla f) = \mathbf{u} \cdot \mathbf{A} \cdot \mathbf{v} \tag{10.7.5}$$

(cont.)

$$0 = \mathbf{u} \cdot \delta(\nabla f) = \mathbf{u} \cdot \mathbf{A} \cdot \mathbf{v} \qquad (10.7.5)$$

When (10.7.5) holds for two vectors u and v, they are said to be **conjugate**. When the relation holds pairwise for all members of a set of vectors, they are said to be a **conjugate set**.

Powell's Quadratically Convergent Method

Powell first discovered a direction set method that does produce N mutually conjugate directions.

Steps:

1. Initialize the set of directions $\boldsymbol{u}_i$ to the basis vectors,

$$\mathbf{u}_i = \mathbf{e}_i \qquad i = 0, \ldots, N-1$$

2. Then repeat the following steps until function stops decreasing:
   - Save starting point as $\boldsymbol{P}_0$.
   - For i = 0,...,N-1, move $\boldsymbol{P}_i$ to the minimum along direction $\boldsymbol{u}_i$ and call this point $\boldsymbol{P}_{i+1}$.
   - For i = 0,...,N-2, set $\boldsymbol{u}_i \leftarrow \boldsymbol{u}_{i+1}$.
   - Set $\boldsymbol{u}_{N-1} \leftarrow \boldsymbol{P}_N$-$\boldsymbol{P}_0$.
   - Move $\boldsymbol{P}_N$ to the minimum along direction $\boldsymbol{u}_{N-1}$ and call this point $\boldsymbol{P}_0$.

Example with N=2

1. $\boldsymbol{u}_0 = \boldsymbol{e}_0$, $\boldsymbol{u}_1 = \boldsymbol{e}_1$
2. inintialize $\boldsymbol{P}_0$

Repeat (until function stops decreasing)**{**

3. Move $\boldsymbol{P}_0$ to the minimum along **u0**, set $\boldsymbol{P}_1 = \boldsymbol{P}_0 + \lambda_0 \cdot \boldsymbol{u}_0$
4. Move $\boldsymbol{P}_1$ to the minimum along **u1**, set $\boldsymbol{P}_2 = \boldsymbol{P}_1 + \lambda_1 \cdot \boldsymbol{u}_1$
5. set $\boldsymbol{u}_0 = \boldsymbol{u}_1$
6. set $\boldsymbol{u}_1 = \boldsymbol{P}_2 - \boldsymbol{P}_0$
7. Move $\boldsymbol{P}_2$ to the minimum along **u1**, set $\boldsymbol{P}_0 = \boldsymbol{P}_2 + \lambda_2 \cdot \boldsymbol{u}_1$

}

Problem of Powell's Method

The procedure of throwing away at each stage, $\boldsymbol{u}_0$ in favor of $\boldsymbol{P}_N$-$\boldsymbol{P}_0$ tends to produce sets of directions that "fold up on each other" and become linearly dependent. Once this happens, the procedure finds the minimum of the function f only over a subspace of the full N-dimensional case; in other words, it gives the wrong answer.

For previous example, if $\boldsymbol{u}_1$ =$\boldsymbol{P}_2$-$\boldsymbol{P}_0$ from step 6 has the same direction as $\boldsymbol{u}_0$ from step 5, linearly dependence happens for latter loops.

Discarding the Direction of Largest Decrease

The basic idea of modified Powell's method is still to take $\boldsymbol{P}_N$-$\boldsymbol{P}_0$ as a new direction;

For a valley whose long direction is twisting slowly, this direction is likely to give us a good run along the new long direction. The change is to discard the old direction along which the function f made its largest decrease. This seems paradoxical, since that direction was the best of the previous iteration. However, it is also likely to be a major component of the new direction that we are adding, so dropping it gives us the best chance of avoiding a buildup of linear dependence.

(cont.)

There are a couple of exceptions to this basic idea. Sometimes it is better not to add a new direction at all. Define

$$f_0 \equiv f(\mathbf{P}_0) \qquad f_N \equiv f(\mathbf{P}_N) \qquad f_E \equiv f(2\mathbf{P}_N - \mathbf{P}_0) \qquad (10.7.7)$$

Here f_E is the function value at an "extrapolated" point somewhat further along the proposed new direction. Also define Δf to be the magnitude of the largest decrease along one particular direction of the present basic procedure iteration. (Δf is a positive number.) Then:

1. If $f_E \geq f_0$, then keep the old set of directions for the next basic procedure, because the average direction $\boldsymbol{P}_N$-$\boldsymbol{P}_0$ is all played out.

2. If $(f_0\text{-}2f_N\text{+}f_E)\ [(f_0\text{-}f_N)\text{-}\ \Delta f\ ]^2 \geq (f_0 - f_E)^2\Delta f$, then keep the old set of directions for the next basic procedure, because either (i) the decrease along the average direction was not primarily due to any single direction's decrease, or (ii) there is a substantial second derivative along the average direction and we seem to be near to the bottom of its minimum.

References

Press, William H., and William T. Vetterling. Numerical Recipes. Cambridge Univ. Press, 2007.

https://mathworld.wolfram.com/BrentsMethod.html

https://users.wpi.edu/~walker/MA3257/HANDOUTS/brents_algm.pdf

https://cs.fit.edu/~dmitra/SciComp/13Spr/HuiYunfei-Ch10.3.4.pptx

https://cs.fit.edu/~dmitra/SciComp/13Spr/ShiBo10.7.pptx

https://www.wiwi.uni-wuerzburg.de/fileadmin/12010500/user_upload/skripte/ss10/CompEcon/cechap33.pdf