Package 'capn'

October 12, 2022

Type Package

Title Capital Asset Pricing for Nature

Version 1.0.0

Date 2017-06-04

Author Seong Do Yun [aut, cre],
Eli P. Fenichel [aut, ctb],
Joshua K. Abbott [aut, ctb]

Maintainer Seong Do Yun <`seongdo.yun@yale.edu`>

Description
Implements approximation methods for natural capital asset prices suggested by Fenichel and Abbott (2014) <doi:10.1086/676034> in Journal of the Associations of Environmental and Resource Economists (JAERE), Fenichel et al. (2016) <doi:10.1073/pnas.1513779113> in Proceedings of the National Academy of Sciences (PNAS), and Yun et al. (2017) in PNAS (accepted), and their extensions: creating Chebyshev polynomial nodes and grids, calculating basis of Chebyshev polynomials, approximation and their simulations for: V-approximation (single and multiple stocks, PNAS), P-approximation (single stock, PNAS), and Pdot-approximation (single stock, JAERE). Development of this package was generously supported by the Knobloch Family Foundation.

Depends R (>= 3.0.1)

License GPL (>= 2)

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-05 17:48:02 UTC

R topics documented:

`aproxdef` *Defining Approximation Space*

Description

The function defines an approximation space for all three approximation apporoaches (V, P, and Pdot).

Usage

```
aproxdef(deg, lb, ub, delta)
```

Arguments

`deg`	An array of degrees of approximation function: degrees of Chebyshev polynomials
`lb`	An array of lower bounds
`ub`	An array of upper bounds
`delta`	discount rate

Details

For the i-th dimension of $i = 1, 2, \cdots, d$, suppose a polynomial approximant s_i over a bounded interval $[a_i, b_i]$ is defined by Chebysev nodes. Then, a d-dimensional Chebyshev grids can be defined as:

$\mathbf{S} = \{(s_1, s_2, \cdots, s_d) | a_i \leq s_1 \leq b_i, i = 1, 2, \cdots, d\}$.

Suppose we impletement n_i numbers of polynomials (i.e., $(n_i - 1)$-th order) for the i-th dimension. The approximation space is defined as:

deg = c($n_1, n_2, \cdots, n_d$),
lb = c($a_1, a_2, \cdots, a_d$), and
ub = c($b_1, b_2, \cdots, b_d$).

`delta` is the given constant discount rate.

Value

A list containing the approximation space

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`vaprox`, `vsim`, `paprox`, `psim`, `pdotaprox`, `pdotsim`

Examples

```
## Reef-fish example: see Fenichel and Abbott (2014)
delta <- 0.02
upper <- 359016000     # upper bound on approximation space
lower <- 5*10^6        # lower bound on approximation space
myspace <- aproxdef(50,lower,upper,delta)
## Two dimensional example
ub <- c(1.5,1.5)
lb <- c(0.1,0.1)
deg <- c(20,20)
delta <- 0.03
myspace <- aproxdef(deg,lb,ub,delta)
```

`catch` *catch function of GOM dataset*

Description

The function calulates the catchment in the reef-fishy example of GOM dataset (Fenichel and Abbott, 2014).

Usage

```
catch(s,Z)
```

Arguments

`s`	stock
`Z`	parameter vector

Details

This catch function is adopted in `GOM` dataset.

Value

Quantity of catchment

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`chebbasisgen` *Generating Unidimensional Chebyshev polynomial (monomial) basis*

Description

The function calculates the monomial basis of Chebyshev polynomials for the given unidimensional nodes, s_i, over a bounded interval [a,b].

Usage

```
chebbasisgen(stock, npol, a, b, dorder = NULL)
```

Arguments

`stock`	An array of Chebyshev polynomial nodes s_i (an array of stocks in `capn`-packages)
`npol`	Number of polynomials (n polynomials = (n-1)-th degree)
`a`	The lower bound of inverval [a,b]
`b`	The upper bound of inverval [a,b]
`dorder`	Degree of partial derivative of the basis; Default is NULL; if dorder = 1, returns the first order partial derivative

Details

Suppose there are m numbers of Chebyshev nodes over a bounded interval [a,b]:

$s_i \in [a, b]$, for $i = 1, 2, \cdots, m$.

These nodes can be nomralized to the standard Chebyshev nodes over the domain [-1,1]:

$z_i = \frac{2(s_i - a)}{(b-a)} - 1$.

With normalized Chebyshev nodes, the recurrence relations of Chebyshev polynomials of order n is defined as:

$T_0(z_i) = 1$,
$T_1(z_i) = z_i$, and
$T_n(z_i) = 2z_i T_{n-1}(z_i) - T_{n-2}(z_i)$.

The interpolation matrix (Vandermonde matrix) of (n-1)-th Chebyshev polynomials with m nodes, Φ_{mn} is:

$$\Phi_{mn} = \begin{bmatrix} 1 & T_1(z_1) & \cdots & T_{n-1}(z_1) \\ 1 & T_1(z_2) & \cdots & T_{n-1}(z_2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & T_1(z_m) & \cdots & T_{n-1}(z_m) \end{bmatrix}.$$

The partial derivative of the monomial basis matrix can be found by the relation:

$(1 - z_i^2)T'_n(z_i) = n[T_{n-1}(z_i) - z_i T_n(z_i)]$.

The technical details of the monomial basis of Chebyshev polynomial can be referred from Amparo et al. (2007) and Miranda and Fackler (2012).

Value

A matrix (number of nodes (m) x npol (n)) of (monomial) Chebyshev polynomial basis

References

Amparo, Gil, Javier Segura, and Nico Temme. (2007) *Numerical Methods for Special Functions*. Cambridge: Cambridge University Press.
Miranda, Mario J. and Paul L. Fackler. (2002) *Applied Computational Economics and Finance*. Cambridge: The MIT Press.

See Also

`chebnodegen`

Examples

```
## Reef-fish example: see Fenichel and Abbott (2014)
data("GOM")
nodes <- chebnodegen(param$nodes,param$lowerK,param$upperK)
## An example of Chebyshev polynomial basis
chebbasisgen(nodes,20,0.1,1.5)
## The partial derivative of Chebyshev polynomial basis with the same function
chebbasisgen(nodes,20,0.1,1.5,1)
```

`chebgrids` *Generating Chebyshev grids*

Description

This function generates a grid of multi-dimensional Chebyshev nodes.

Usage

```
chebgrids(nnodes, lb, ub, rtype = NULL)
```

Arguments

`nnodes`	An array of numbers of nodes
`lb`	An array of lower bounds
`ub`	An array of upper bounds
`rtype`	A type of results; default is NULL that returns a list class; if rtype = list, returns a list class; if rtype = grid, returns a matrix class.

Details

For the i-th dimension of $i = 1, 2, \cdots, d$, suppose a polynomial approximant s_i over a bounded interval $[a_i, b_i]$ is defined by Chebysev nodes. Then, a d-dimensional Chebyshev grids can be defined as:

$$\mathbf{S} = \{(s_1, s_2, \cdots, s_d) | a_i \leq s_1 \leq b_i, i = 1, 2, \cdots, d\}.$$

This is all combinations of s_i. Two types of results are provided. '`rtype = list`' provides a list of d dimensions wherease '`rtype = grids`' creates a $\left(\prod_{i=1}^{d} n_i\right) \times d$ matrix.

Value

A list with d elements of Chebyshev nodes or a $\left(\prod_{i=1}^{d} n_i\right) \times d$ matrix of Chebyshev grids

See Also

chebnodegen

Examples

```
## Chebyshev grids with two-dimension
chebgrids(c(5,3), c(1,1), c(2,3))
# Returns the same results
chebgrids(c(5,3), c(1,1), c(2,3), rtype='list')
## Returns a matrix grids with the same domain
chebgrids(c(5,3), c(1,1), c(2,3), rtype='grid')
## Chebyshev grids with one-dimension
chebgrids(5,1,2)
chebnodegen(5,1,2)
## Chebyshev grids with three stock
chebgrids(c(3,4,5),c(1,1,1),c(2,3,4),rtype='grid')
```

`chebnodegen` *Unidimensional Chebyshev nodes*

Description

The function generates uni-dimensional chebyshev nodes.

Usage

```
chebnodegen(n, a, b)
```

Arguments

n	A number of nodes
a	The lower bound of inverval [a,b]
b	The upper bound of interval [a,b]

Details

A polynomial approximant s_i over a bounded interval [a,b] is constructed by:

$s_i = \frac{b+a}{2} + \frac{b-a}{2} cos(\frac{n-i+0.5}{n}\pi)$ for $i = 1, 2, \cdots, n.$

More detail explanation can be refered from Miranda and Fackler (2002, p.119).

Value

An array n Chebyshev nodes

References

Miranda, Mario J. and Paul L. Fackler. (2002) *Applied Computational Economics and Finance*. Cambridge: The MIT Press.

Examples

```
## 10 Chebyshev nodes in [-1,1]
chebnodegen(10,-1,1)
## 5 Chebyshev nodes in [1,5]
chebnodegen(5,1,5)
```

`dsdotds` *first derivative function of sdot in GOM dataset*

Description

`dsdotds` evaluated $\frac{dsdot}{ds}$ in the reef-fishy example of GOM dataset (Fenichel and Abbott, 2014).

Usage

```
dsdotds(s,Z)
```

Arguments

`s`	stock
`Z`	parameter vector

Details

This function is adopted in `GOM` dataset.

Value

The first derivative of sdot with respect to s

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`dsdotdss` *second derivative function of sdot in GOM dataset*

Description

`dsdotdss` evaluated $\frac{d}{ds}(\frac{dsdot}{ds})$ in the reef-fishy example of GOM dataset (Fenichel and Abbott, 2014).

Usage

```
dsdotdss(s,Z)
```

Arguments

`s`	stock
`Z`	parameter vector

Details

This function is adopted in `GOM` dataset.

Value

The second derivative of sdot with respect to s

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`dwds` *first derivative function of profit in GOM dataset*

Description

`dwds` evaluated $\frac{dw}{ds}$ in the reef-fishy example of GOM dataset (Fenichel and Abbott, 2014).

Usage

```
dwds(s,Z)
```

Arguments

`s`	stock
`Z`	parameter vector

Details

This function is adopted in `GOM` dataset.

Value

The first derivative of w with respect to s

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`dwdss` *second derivative function of profit in GOM dataset*

Description

`dwdss` evaluated $\frac{d}{ds}(\frac{dw}{ds})$ in the reef-fishy example of GOM dataset (Fenichel and Abbott, 2014).

Usage

```
dwdss(s,Z)
```

Arguments

`s`	stock
`Z`	parameter vector

Details

This function is adopted in `GOM` dataset.

Value

The second derivative of w with respect to s

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`effort` *effort function of GOM dataset*

Description

The function calulates the catchment effort in the reef-fishy example of GOM dataset (Fenichel and Abbott, 2014).

Usage

```
effort(s,Z)
```

Arguments

`s`	stock
`Z`	parameter vector

Details

This effort function is adopted in `GOM` dataset.

Value

catchment effort values

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`GOM` *Reef Fish example: one dimensional stock*

Description

The `GOM` provides data to replicate the Gulf of Mexico Reef Fish example in Fenichel and Abbott (2014). This dataset is consisted of parameters and functions.From Fenichel and Abboott(2014),

catch effort: $x(s) = ys^{\gamma}$,

harvest: $h(s,x) = q((ys^{\gamma})^{\alpha})s = q(y^{\alpha})(s^{\gamma\alpha})$,

profit: $w(s,x) = price \cdot h(s,x) - cost \cdot x(s)$, and

sdot: $\dot{s} = rs\left(1 - \frac{s}{k}\right) - q(y^{\alpha})(s^{\gamma\alpha+1})$.

The parameters in detal are in below.

Usage

```
## Load dataset
data("GOM")
## Demonstration of example
# demo(GOM, package="capn")
## R-script location
# system.file("demo", "GOM.R", package = "capn")
```

Format

param: a data.frame of parameters

- `r` intrinsic growth rate (=0.3847)
- `k` carrying capacity (=359016000)
- `q` catchability coefficient (=0.00031729344157311126)
- `price` price (=2.70)
- `cost` cost (=153.0)
- `alpha` technology parameter (=0.5436459179063678)

- `gamma` pre-ITQ management parameter (=0.7882)
- `y` system equivalence parameter (=0.15745573410462155)
- `delta` discount rate (=0.02)
- `order` Chebyshev polynomial order (=50)
- `upperK` upper bound of Chebyshev polynomial nodes (=k)
- `lowerK` lower bound of Chebyshev polynomial nodes (=5*10^6)
- `nodes` the number of Chebyshev polynomial nodes (=50)

functions: functions for generate simulation data for each nodes

- `effort` effort function
- `catch` catch function
- `profit` profit function (w in Fenichel and Abbott (2014))
- `sdot` evaluated $\frac{dst}{dt}$
- `dsdotds` evaluated $\frac{dsdot}{ds}$
- `dsdotdss` evaluated $\frac{d}{ds}(\frac{dsdot}{ds})$
- `dwds` evaluated $\frac{dw}{ds}$
- `dwdss` evaluated $\frac{d}{ds}(\frac{dw}{ds})$

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

`LV` *Prey-Predator (Lotka-Volterra) example: two stocks*

Description

The `LV` provides the data and functions to simulate prey-predator (Lotka-Volterra) model. The original code was written by Joshua Abbott in MATLAB and Seong Do Yun adapted it to a package example. The prey-predator model is:

Prey (X): $\dot{X} = rX\left(1 - \frac{X}{K}\right) - aXY - \theta X$, and

Predator (Y): $\dot{Y} = bXY - mY - \gamma Y$.

The parameters are given as:
$r = 0.025$: intrinsic growth rate for prey,
$K = 1$: carrying capacity for prey,
$a = 0.08$: predator-related mortality parameter for prey,
$b = 0.05$: predator/prey uptake parameter for predator,
$m = 0.01$: natural mortality for predator,

$\gamma = 0.005$: slope for linear predator harvest control rule, and
$\theta = 0.005$: slope for linear prey harvest control rule

The predator with no economic value (unharvested) is designed for the economic program as:

$W = harv.prey(p.prey - c.prey/X)\theta X + harv.pred * (p.pred - c.pred/Y)\gamma Y$.

The paramters are:
$p.pred = 0$: price per unit harvest of predator,
$p.prey = 25$: price per unit harvest of prey,
$c.prey = 0.1p_prey$: cost /per unit of prey effort in Schaefer model (really c/q with q=1), and
$c.pred = c_prey$: cost per unit of predator effort in Schaefer model (really c/q with q=1).

Usage

```
## Load dataset
data("lvdata")
## Demonstration of example
# demo(LV, package="capn")
## R-script location
# system.file("demo", "LV.R", package = "capn")
```

Format

lvaproxdata: a data.frame for approximation (evaluated on (20 x 20) Chebyshev nodes)

- `xs` prey stock
- `ys` predator stock
- `xdot` evaluated xdot $\frac{dx}{dt}$
- `ydot` evaluated ydot $\frac{dy}{dt}$
- `wval` profit (W in Fenichel and Abtott (2014))

lvsimdata.time: a data for time simulation (101 ODE solution)

- `tseq` time sequence from 0 to 100
- `xs` prey stock
- `ys` predator stock

`lvaproxdata` *Prey-Predator (Lotka-Volterra) example in LV dataset*

Description

The `lvaproxdata` provides the data in `LV` dataset to simulate prey-predator (Lotka-Volterra) model. The original code was written by Joshua Abbott in MATLAB and Seong Do Yun adapted it to a package example. The prey-predator model is:

Prey (X): $\dot{X} = rX\left(1 - \frac{X}{K}\right) - aXY - \theta X$, and

Predator (Y): $\dot{Y} = bXY - mY - \gamma Y$.

The parameters are given as:
$r = 0.025$: intrinsic growth rate for prey,
$K = 1$: carrying capacity for prey,
$a = 0.08$: predator-related mortality parameter for prey,
$b = 0.05$: predator/prey uptake parameter for predator,
$m = 0.01$: natural mortality for predator,
$\gamma = 0.005$: slope for linear predator harvest control rule, and
$\theta = 0.005$: slope for linear prey harvest control rule

The predator with no economic value (unharvested) is designed for the economic program as:

$W = harv.prey(p.prey - c.prey/X)\theta X + harv.pred * (p.pred - c.pred/Y)\gamma Y.$

The paramters are:
$p.pred = 0$: price per unit harvest of predator,
$p.prey = 25$: price per unit harvest of prey,
$c.prey = 0.1p_prey$: cost /per unit of prey effort in Schaefer model (really c/q with q=1), and
$c.pred = c_prey$: cost per unit of predator effort in Schaefer model (really c/q with q=1).

Usage

```
## Load dataset
data("lvdata")
```

Format

lvaproxdata: a data.frame for approximation (evaluated on (20 x 20) Chebyshev nodes)

- `xs` prey stock
- `ys` predator stock
- `xdot` evaluated xdot $\frac{dx}{dt}$
- `ydot` evaluated ydot $\frac{dy}{dt}$
- `wval` profit (W in Fenichel and Abtott (2014))

See Also

`LV`,`vsim`

`lvsimdata.time` *Prey-Predator (Lotka-Volterra) example in LV dataset*

Description

The `lvsimdata.time` provides the time simulation data in `LV` dataset to simulate prey-predator (Lotka-Volterra) model. The original code was written by Joshua Abbott in MATLAB and Seong Do Yun adapted it to a package example. The prey-predator model is:

Prey (X): $\dot{X} = rX\left(1 - \frac{X}{K}\right) - aXY - \theta X$, and

Predator (Y): $\dot{Y} = bXY - mY - \gamma Y$.

The parameters are given as:
$r = 0.025$: intrinsic growth rate for prey,
$K = 1$: carrying capacity for prey,
$a = 0.08$: predator-related mortality parameter for prey,
$b = 0.05$: predator/prey uptake parameter for predator,
$m = 0.01$: natural mortality for predator,
$\gamma = 0.005$: slope for linear predator harvest control rule, and
$\theta = 0.005$: slope for linear prey harvest control rule

The predator with no economic value (unharvested) is designed for the economic program as:

$W = harv.prey(p.prey - c.prey/X)\theta X + harv.pred * (p.pred - c.pred/Y)\gamma Y.$

The paramters are:
$p.pred = 0$: price per unit harvest of predator,
$p.prey = 25$: price per unit harvest of prey,
$c.prey = 0.1p_prey$: cost /per unit of prey effort in Schaefer model (really c/q with q=1), and
$c.pred = c_prey$: cost per unit of predator effort in Schaefer model (really c/q with q=1).

Usage

```
## Load dataset
data("lvdata")
```

Format

lvsimdata.time: a data for time simulation (101 ODE solution)

- `tseq` time sequence from 0 to 100
- `xs` prey stock
- `ys` predator stock

See Also

`LV`, `vsim`

`paprox` *Calculating P-approximation coefficients*

Description

The function provides the P-approximation coefficients of the defined Chebyshev polynomials in `aproxdef`. For now, only unidimensional case is developed.

Usage

```
paprox(aproxspace, stock, sdot, dsdotds, dwds)
```

Arguments

`aproxspace`	An approximation space defined by `aproxdef` function
`stock`	An array of stock, s
`sdot`	An array of ds/dt, $\dot{s} = \frac{ds}{dt}$
`dsdotds`	An array of d(sdot)/ds, $\frac{d\dot{s}}{ds}$
`dwds`	An array of dw/ds, $\frac{dW}{ds}$

Details

The P-approximation is finding the shadow price of a stock, p from the relation:

$p(s) = \frac{W_s(s)+\dot{p}(s)}{\delta-\dot{s}_s}$,

where $W_s = \frac{dW}{ds}$, $\dot{p}(s) = \frac{dp}{ds}$, $\dot{s}_s = \frac{d\dot{s}}{ds}$, and δ is the given discount rate.

Consider approximation $p(s) = \mu(s)\beta$, $\mu(s)$ is Chebyshev polynomials and β is their coeffcients. Then, $\dot{p} = diag(\dot{s})\mu_s(s)\beta$ by the orthogonality of Chebyshev basis. Adopting the properties above, we can get the unknown coefficient vector β from:

$\mu\beta = diag\left(\delta - \dot{s}_s\right)^{-1}(W_s + diag(\dot{s})\mu_s\beta)$, and thus,

$\beta = \left(diag\left(\delta - \dot{s}_s\right)\mu - diag(\dot{s})\mu_s\right)^{-1} W_s$.

In a case of over-determined (more nodes than approaximation degrees),

$$\left(\left(diag\left(\delta - \dot{s}_s\right)\mu - diag(\dot{s})\mu_s\right)^T \left(diag\left(\delta - \dot{s}_s\right)\mu - diag(\dot{s})\mu_s\right)\right)^{-1} \left(diag\left(\delta - \dot{s}_s\right)\mu - diag(\dot{s})\mu_s\right)^T W_s$$

For more detils see Fenichel et al. (2016).

Value

A list of approximation resuts: deg, lb, ub, delta, and coefficients. Use `results$item` (or `results[["item"]]`) to import each result item.

`degree`	degree of Chebyshev polynomial
`lowerB`	lower bound of Chebyshev nodes
`upperB`	upper bound of Chebyshev nodes
`delta`	discount rate
`coefficient`	Chebyshev polynomial coefficients

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.
Fenichel, Eli P., Joshua K. Abbott, Jude Bayham, Whitney Boone, Erin M. K. Haacker, and Lisa Pfeiffer. (2016) "Measuring the Value of Groundwater and Other Forms of Natural Capital." *Proceedings of the National Academy of Sciences* .113:2382-2387.

See Also

`aproxdef`,`psim`

Examples

```
## 1-D Reef-fish example: see Fenichel and Abbott (2014)
data("GOM")
nodes <- chebnodegen(param$nodes,param$lowerK,param$upperK)
simuDataP <- cbind(nodes,sdot(nodes,param),
                   dsdotds(nodes,param),dwds(nodes,param))
Aspace <- aproxdef(param$order,param$lowerK,param$upperK,param$delta)
pC <- paprox(Aspace,simuDataP[,1],simuDataP[,2],
             simuDataP[,3],simuDataP[,4])
```

`param` *the parameter vector adopted in GOM dataset*

Description

The `GOM` provides data to replicate the Gulf of Mexico Reef Fish example in Fenichel and Abbott (2014).

Usage

```
## Load dataset
data("GOM")
```

Format

param: a data.frame of parameters

- `r` intrinsic growth rate (=0.3847)
- `k` carrying capacity (=359016000)
- `q` catchability coefficient (=0.00031729344157311126)
- `price` price (=2.70)
- `cost` cost (=153.0)
- `alpha` technology parameter (=0.5436459179063678)
- `gamma` pre-ITQ management parameter (=0.7882)
- `y` system equivalence parameter (=0.15745573410462155)
- `delta` discount rate (=0.02)
- `order` Chebyshev polynomial order (=50)
- `upperK` upper bound of Chebyshev polynomial nodes (=k)
- `lowerK` lower bound of Chebyshev polynomial nodes (=5*10^6)
- `nodes` the number of Chebyshev polynomial nodes (=50)

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`pdotaprox` *Calculating Pdot-approximation coefficients*

Description

The function provides the Pdot-approximation coefficients of the defined Chebyshev polynomials in `aproxdef`. For now, only unidimensional case is developed.

Usage

```
pdotaprox(aproxspace, stock, sdot, dsdotds, dsdotdss, dwds, dwdss)
```

Arguments

`aproxspace`	An approximation space defined by `aproxdef` function
`stock`	An array of stock, s
`sdot`	An array of ds/dt, $\dot{s} = \frac{ds}{dt}$
`dsdotds`	An array of d(sdot)/ds, $\frac{d\dot{s}}{ds}$
`dsdotdss`	An array of d/ds(d(sdot)/ds), $\frac{d}{ds}\left(\frac{d\dot{s}}{ds}\right)$
`dwds`	An array of dw/ds, $\frac{dW}{ds}$
`dwdss`	An array of d/ds(dw/ds), $\frac{d}{ds}\left(\frac{dW}{ds}\right)$

Details

The Pdot-approximation is finding the shadow price of a stock, p from the relation:

$$p(s) = \frac{W_s(s)+\dot{p}(s)}{\delta-\dot{s}_s},$$

where $W_s = \frac{dW}{ds}$, $\dot{p}(s) = \frac{dp}{ds}$, $\dot{s}_s = \frac{d\dot{s}}{ds}$, and δ is the given discount rate.

In order to operationalize this approach, we take the time derivative of this expression:

$$\dot{p} = \frac{((W_{ss}\dot{s}+\ddot{p})(\delta-\dot{s}_s)+(W_s+\dot{p})(\dot{s}_{ss}\dot{s}))}{(\delta-\dot{s}_s)^2}$$

Consider approximation $\dot{p}(s) = \mu(s)\beta$, $\mu(s)$ is Chebyshev polynomials and β is their coeffcients. Then, $\ddot{p} = \frac{d\dot{p}}{ds}\frac{ds}{dt} = diag(\dot{s})\mu_s(s)\beta$ by the orthogonality of Chebyshev basis. Adopting the properties above, we can get the unknown coefficient vector β from:

$$\mu\beta = diag\left(\delta-\dot{s}_s\right)^{-2}\left[\left(W_{ss}\dot{s}+diag(\dot{s})\mu_s\beta\right)\left(\delta-\dot{s}_s\right)+diag\left(\dot{s}_{ss}\dot{s}\right)\left(W_s+\mu\beta\right)\right], \text{ and}$$

$$\beta = \left[diag\left(\delta-\dot{s}_s\right)^2\mu - diag\left(\dot{s}\left(\delta-\dot{s}_s\right)\right)\mu_s - diag(\dot{s}_{ss}\dot{s})\mu\right]^{-1}\left(W_{ss}\dot{s}\left(\delta-\dot{s}_s\right)+W_s\dot{s}_{ss}\dot{s}\right).$$

If we suppose $A = \left[diag\left(\delta-\dot{s}_s\right)^2\mu - diag\left(\dot{s}\left(\delta-\dot{s}_s\right)\right)\mu_s - diag(\dot{s}_{ss}\dot{s})\mu\right]$ and $B = \left(W_{ss}\dot{s}\left(\delta-\dot{s}_s\right)+W_s\dot{s}_{ss}\dot{s}\right)$, then over-determined case can be calculated:

$$\beta = \left(A^TA\right)^{-1}A^TB.$$

For more detils see Fenichel and Abbott (2014).

Value

A list of approximation results: deg, lb, ub, delta, and coefficients. Use `results$item` (or `results[["item"]]`) to import each result item.

`degree`	degree of Chebyshev polynomial
`lowerB`	lower bound of Chebyshev nodes
`upperB`	upper bound of Chebyshev nodes
`delta`	discount rate
`coefficient`	Chebyshev polynomial coefficients

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`aproxdef`, `pdotsim`

Examples

```
## 1-D Reef-fish example: see Fenichel and Abbott (2014)
data("GOM")
nodes <- chebnodegen(param$nodes,param$lowerK,param$upperK)
simuDataPdot <- cbind(nodes,sdot(nodes,param),
                      dsdotds(nodes,param),dsdotdss(nodes,param),
                      dwds(nodes,param),dwdss(nodes,param))
Aspace <- aproxdef(param$order,param$lowerK,param$upperK,param$delta)
pdotC <- pdotaprox(Aspace,simuDataPdot[,1],simuDataPdot[,2],
                   simuDataPdot[,3],simuDataPdot[,4],
                   simuDataPdot[,5],simuDataPdot[,6])
```

`pdotsim` *Simulation of Pdot-approximation*

Description

The function provides the Pdot-approximation simulation.

Usage

```
pdotsim(pdotcoeff, stock, sdot, dsdotds, wval, dwds)
```

Arguments

`pdotcoeff`	An approximation result from `pdotaprox` function
`stock`	An array of stock
`sdot`	An array of ds/dt, $\dot{s} = \frac{ds}{dt}$
`dsdotds`	An array of d(sdot)/ds, $\frac{d\dot{s}}{ds}$
`wval`	An array of W-value
`dwds`	An array of dw/ds, $\frac{dW}{ds}$

Details

Let $\hat{\beta}$ be the vector of approximation coefficents from the results of `pdotaprox` function. The estimated shadow price (accounting) price of stock over the given approximation interval of $s \in [a, b]$, $\hat{p}$ can be calculated as:

$\hat{p} = \frac{W_s + \mu\beta}{\delta - \dot{s}_s}$.

The estimated value function is:

$\hat{V} = \frac{1}{\delta}\left(W + \hat{p}\dot{s}\right)$.

For more detils see Fenichel and Abbott (2014) and Fenichel et al. (2016).

Value

A list of approximation resuts: shadow (accounting) prices, inclusive wealth, and value function, stock, and W values. Use `results$item` (or `results[["item"]]`) to import each result item.

`shadowp`	Shadow price
`vfun`	Value function
`stock`	Stock
`wval`	W-value

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.
Fenichel, Eli P., Joshua K. Abbott, Jude Bayham, Whitney Boone, Erin M. K. Haacker, and Lisa Pfeiffer. (2016) "Measuring the Value of Groundwater and Other Forms of Natural Capital." *Proceedings of the National Academy of Sciences* .113:2382-2387.

See Also

`pdotaprox`

Examples

```
## 1-D Reef-fish example: see Fenichel and Abbott (2014)
data("GOM")
nodes <- chebnodegen(param$nodes,param$lowerK,param$upperK)
simuDataPdot <- cbind(nodes,sdot(nodes,param),
                      dsdotds(nodes,param),dsdotdss(nodes,param),
                      dwds(nodes,param),dwdss(nodes,param))
Aspace <- aproxdef(param$order,param$lowerK,param$upperK,param$delta)
pdotC <- pdotaprox(Aspace,simuDataPdot[,1],simuDataPdot[,2],
                   simuDataPdot[,3],simuDataPdot[,4],
                   simuDataPdot[,5],simuDataPdot[,6])
GOMSimPdot <- pdotsim(pdotC,simuDataPdot[,1],simuDataPdot[,2],simuDataPdot[,3],
                      profit(nodes,param),simuDataPdot[,5])
```

```
# Shadow Price
plotgen(GOMSimPdot, xlabel="Stock size, s", ylabel="Shadow price")

# Value function and profit
plotgen(GOMSimPdot,ftype="vw",
        xlabel="Stock size, s",
        ylabel=c("Value Function","Profit"))
```

`plotgen` *Plot Generator for Shadow Price or Value Function*

Description

The function draws shadowp or vfun-w plot from the simulation results of `vsim`, `psim`, or `pdotsim`.

Usage

```
plotgen(simres, ftype = NULL, whichs = NULL, tvar = NULL, xlabel = NULL,
  ylabel = NULL)
```

Arguments

`simres`	A simulation results from `vsim`, `psim`, or `pdotsim`
`ftype`	Plot type (ftype=NULL (default) or ftype="p" for shadow price; ftype="vw" for vfun-w plot)
`whichs`	A pisitive integer for indicating a specific stock for multi-sotck cases (ftype=NULL (default) or 1<= whichs <= the number of stocks)
`tvar`	An array of time variable if simulation result is a time-base simulation
`xlabel`	A character for x-label of a plot (xlabel=NULL (default); "Stock" or "Time")
`ylabel`	An array of characters for y-label of a plot (ylabel=NULL (default); "Shadow Price", "Value Function" or "W-value")

Details

This function provides an one-dimensional plot for "shadow price-stock", "shadow price-time", "Value function-stock", "Value function-time", "Value function-stock-W value", or "Value function-time-W value" depending on input arguments.

Value

A plot of approximation resuts: shadow (accounting) prices, inclusive wealth, and Value function

See Also

`vsim`, `psim`, `pdotsim`

Examples

```
## 1-D Reef-fish example: see Fenichel and Abbott (2014)
data("GOM")
nodes <- chebnodegen(param$nodes,param$lowerK,param$upperK)
simuDataP <- cbind(nodes,sdot(nodes,param),
                   dsdotds(nodes,param),dwds(nodes,param))
Aspace <- aproxdef(param$order,param$lowerK,param$upperK,param$delta)

# p-approximation
pC <- paprox(Aspace,simuDataP[,1],simuDataP[,2],
             simuDataP[,3],simuDataP[,4])

# Without prividing W-value
GOMSimP <- psim(pC,simuDataP[,1])
# With W-value
GOMSimP2 <- psim(pC,simuDataP[,1],profit(nodes,param),simuDataP[,2])

# Shadow price-Stock plot
plotgen(GOMSimP)
plotgen(GOMSimP,ftype="p")
plotgen(GOMSimP,xlabel="Stock Size, S", ylabel="Shadow Price (USD/Kg)")

# Value-Stock-W plot
plotgen(GOMSimP2,ftype="vw")
plotgen(GOMSimP2,ftype="vw",xlabel="Stock Size, S", ylabel="Value Function")
plotgen(GOMSimP2,ftype="vw",xlabel="Stock Size, S", ylabel="Value Function")

## 2-D Prey-Predator example
data("lvdata")
aproxdeg <- c(20,20)
lower <- c(0.1,0.1)
upper <- c(1.5,1.5)
delta <- 0.03
lvspace <- aproxdef(aproxdeg,lower,upper,delta)
lvaproxc <- vaprox(lvspace,lvaproxdata)
lvsim <- vsim(lvaproxc,lvsimdata.time[,2:3])

# Shadow price-Stock plot
plotgen(lvsim)
plotgen(lvsim,ftype="p")
plotgen(lvsim,whichs=2,xlabel="Stock Size, S",ylabel="Shadow Price (USD/Kg)")

# Shadow price-time plot
plotgen(lvsim,whichs=2,tvar=lvsimdata.time[,1])

# Value Function-Stock plot
plotgen(lvsim,ftype="vw")
plotgen(lvsim,ftype="vw",whichs=2,
        xlabel="Stock Size, S",ylabel="Shadow Price (USD/Kg)")

# Value Function-time plot
plotgen(lvsim,ftype="vw",tvar=lvsimdata.time[,1])
```

```
plotgen(lvsim,ftype="vw",whichs=2,tvar=lvsimdata.time[,1],
        xlabel="Stock Size, S",ylabel="Shadow Price (USD/Kg)")
```

`profit` *profit function in GOM dataset*

Description

profit (w) function in the reef-fishy example of GOM dataset (Fenichel and Abbott, 2014).

Usage

```
profit(s,Z)
```

Arguments

`s`	stock
`Z`	parameter vector

Details

This function is adopted in `GOM` dataset.

Value

profit

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`psim` *Simulation of P-approximation*

Description

The function provides the P-approximation simulation.

Usage

```
psim(pcoeff, stock, wval = NULL, sdot = NULL)
```

Arguments

`pcoeff`	An approximation result from `paprox` function
`stock`	An array of stock variable
`wval`	(Optional for vfun) An array of W-value (need `sdot` simultaneously)
`sdot`	(Optional for vfun) An array of ds/dt, $\dot{s} = \frac{ds}{dt}$ (need `W` simultaneously)

Details

Let $\hat{\beta}$ be the vector of approximation coefficents from the results of `paprox` function. The estimated shadow price (accounting) price of stock over the given approximation interval of $s \in [a, b]$, $\hat{p}$ can be calculated as:

$\hat{p} = \mu(s)\hat{\beta}$.

The estimated value function is:

$\hat{V} = \frac{1}{\delta}(W + \hat{p}\dot{s})$.

For more detils see Fenichel and Abbott (2014) and Fenichel et al. (2016).

Value

A list of approximation resuts: shadow (accounting) prices, inclusive wealth, value function, stock, and W values. Use `results$item` (or `results[["item"]]`) to import each result item.

`shadowp`	Shadow price
`vfun`	Value function
`stock`	Stock
`wval`	W-value if `wval` is provided

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.
Fenichel, Eli P., Joshua K. Abbott, Jude Bayham, Whitney Boone, Erin M. K. Haacker, and Lisa Pfeiffer. (2016) "Measuring the Value of Groundwater and Other Forms of Natural Capital." *Proceedings of the National Academy of Sciences* .113:2382-2387.

See Also

`aproxdef`, `paprox`

Examples

```
## 1-D Reef-fish example: see Fenichel and Abbott (2014)
data("GOM")
nodes <- chebnodegen(param$nodes,param$lowerK,param$upperK)
simuDataP <- cbind(nodes,sdot(nodes,param),
                   dsdotds(nodes,param),dwds(nodes,param))
Aspace <- aproxdef(param$order,param$lowerK,param$upperK,param$delta)
pC <- paprox(Aspace,simuDataP[,1],simuDataP[,2],
             simuDataP[,3],simuDataP[,4])
GOMSimP <- psim(pC,simuDataP[,1],profit(nodes,param),simuDataP[,2])

# Shadow Price
plotgen(GOMSimP, xlabel="Stock size, s", ylabel="Shadow price")

# Value function and profit
plotgen(GOMSimP,ftype="vw",
        xlabel="Stock size, s",
        ylabel=c("Value Function","Profit"))
```

`sdot` *growth function of GOM dataset*

Description

The function calulates the growth rate in the reef-fishy example of GOM dataset (Fenichel and Abbott, 2014).

Usage

```
sdot(s,Z)
```

Arguments

`s`	stock
`Z`	parameter vector

Details

This function is adopted in `GOM` dataset.

Value

growth rate

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.

See Also

`GOM`

`unigrids` *Generating unifrom grids*

Description

This function generates a grid of multi-dimensional uniform grids.

Usage

```
unigrids(nnodes, lb, ub, rtype = NULL)
```

Arguments

`nnodes`	An array of numbers of nodes
`lb`	An array of lower bounds
`ub`	An array of upper bounds
`rtype`	A type of results; default is NULL that returns a list class; if rtype = list, returns a list class; if rtype = grid, returns a matrix class.

Details

For the i-th dimension of $i = 1, 2, \cdots, d$, suppose a polynomial approximant s_i over a bounded interval $[a_i, b_i]$ is defined by evenly gridded nodes. Then, a d-dimensional uniform grids can be defined as:

$$\mathbf{S} = \{(s_1, s_2, \cdots, s_d) | a_i \leq s_1 \leq b_i, i = 1, 2, \cdots, d\}.$$

This is all combinations of s_i. Two types of results are provided. '`rtype = list`' provides a list of d dimensions wherease '`rtype = grids`' creates a $\left(\prod_{i=1}^{d} n_i\right) \times d$ matrix.

Value

A list with d elements of Chebyshev nodes or a $\left(\prod_{i=1}^{d} n_i\right) \times d$ matrix of uniform grids

Examples

```
## Uniform grids with two-dimension
unigrids(c(5,3), c(1,1), c(2,3))
## Returns the same results
unigrids(c(5,3), c(1,1), c(2,3), rtype='list')
## Returns a matrix grids with the same domain
unigrids(c(5,3), c(1,1), c(2,3), rtype='grid')
## Uniform grid with one-dimension
unigrids(5,1,2)
## Uniform grids with three stock
unigrids(c(3,4,5),c(1,1,1),c(2,3,4),rtype='grid')
```

`vaprox` *Calculating V-approximation coefficients*

Description

The function provides the V-approximation coefficients of the defined Chebyshev polynomials in `aproxdef`.

Usage

```
vaprox(aproxspace, sdata)
```

Arguments

`aproxspace`	An approximation space defined by `aproxdef` function
`sdata`	A data.frame or matrix of [stock,sdot,benefit]=[$\mathbf{S}$,$\dot{\mathbf{S}}$,W]

Details

The V-approximation is finding the shadow price of i-th stock, p_i for $i = 1, \cdots, d$ from the relation:

$\delta V = W(\mathbf{S}) + p_1\dot{s}_1 + p_2\dot{s}_2 + \cdots + p_d\dot{s}_d,$

where δ is the given discount rate, V is the intertemporal welfare function, $\mathbf{S} = (s_1, s_2, \cdots, s_d)$ is a vector of stocks, $W(\mathbf{S})$ is the net benefits accruing to society, and $\dot{s}_i$ is the growth of stock s_i. By the definition of the shadow price, we know:

$p_i = \frac{\partial V}{\partial s_i}.$

Consider approximation $V(\mathbf{S}) = \mu(\mathbf{S})\beta$, $\mu(\mathbf{S})$ is Chebyshev polynomials and β is their coefficients. Then, $p_i = \mu_{s_i}(\mathbf{S})\beta$ by the orthogonality of Chebyshev basis. Adopting the properties above, we can get the unknown coefficient vector β from:

$$\delta\mu(\mathbf{S})\beta = W(\mathbf{S}) + \sum_{i=1}^{d} diag(\dot{s}_i)\mu_{s_i}(\mathbf{S})\beta, \text{ and thus,}$$

$$\beta = \left(\delta\mu(\mathbf{S}) - \sum_{i=1}^{d} diag(\dot{s}_i)\mu_{s_i}(\mathbf{S})\right)^{-1} W(\mathbf{S}).$$

In a case of over-determined (more nodes than approaximation degrees),

$$\beta = \left(\left(\delta\mu(\mathbf{S}) - diag(\dot{s}_i)\sum_{i=1}^{d}\mu_{s_i}(\mathbf{S})\right)^T \left(\delta\mu(\mathbf{S}) - \sum_{i=1}^{d} diag(\dot{s}_i)\mu_{s_i}(\mathbf{S})\right)\right)^{-1}$$

$$\times \left(\delta\mu(\mathbf{S}) - \sum_{i=1}^{d} diag(\dot{s}_i)\mu_{s_i}(\mathbf{S})\right)^T W(\mathbf{S}).$$

For more detils see Fenichel and Abbott (2014), Fenichel et al. (2016), and Yun et al. (2017).

Value

A list of approximation resuts: deg, lb, ub, delta, and coefficients. Use `results$item` (or `results[["item"]]`) to import each result item.

`degree`	degree of Chebyshev polynomial
`lowerB`	lower bound of Chebyshev nodes
`upperB`	upper bound of Chebyshev nodes
`delta`	discount rate
`coefficient`	Chebyshev polynomial coefficients

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.
Fenichel, Eli P., Joshua K. Abbott, Jude Bayham, Whitney Boone, Erin M. K. Haacker, and Lisa Pfeiffer. (2016) "Measuring the Value of Groundwater and Other Forms of Natural Capital." *Proceedings of the National Academy of Sciences*.113:2382-2387.
Yun, Seong Do, Barbara Hutniczak, Joshua K. Abbott, and Eli P. Fenichel. (2017) "Ecosystem Based Management and the Welath of Ecosystems" *Proceedings of the National Academy of Sciences*. (forthcoming).

See Also

`aproxdef`, `vsim`

Examples

```
## 1-D Reef-fish example: see Fenichel and Abbott (2014)
data("GOM")
nodes <- chebnodegen(param$nodes,param$lowerK,param$upperK)
simuDataV <- cbind(nodes,sdot(nodes,param),profit(nodes,param))
Aspace <- aproxdef(param$order,param$lowerK,param$upperK,param$delta)
vC <- vaprox(Aspace,simuDataV)

## 2-D Prey-Predator example
data("lvdata")
aproxdeg <- c(20,20)
lower <- c(0.1,0.1)
upper <- c(1.5,1.5)
delta <- 0.03
lvspace <- aproxdef(aproxdeg,lower,upper,delta)
vaproxc <- vaprox(lvspace,lvaproxdata)
```

`vsim` *Simulation of V-approximation*

Description

The function provides the V-approximation simulation by adopting the results of `vaprox`. Available for multiple stock problems.

Usage

```
vsim(vcoeff, adata, wval = NULL)
```

Arguments

`vcoeff`	An approximation result from `varpox` function
`adata`	A data.frame or matrix of [stock]=[$\mathbf{S}$]
`wval`	(Optional for `plotgen`) An array of W-value

Details

Let $\hat{\beta}$ be the approximation coefficent from the results of `vaprox` function. The estimated shadow (accounting) price of i-th stock over the given approximation intervals of $s_i \in [a_i, b_i]$, $\hat{p}_i$ can be calcuated as:

$\hat{p}_i = \mu(\mathbf{S})\hat{\beta}$ where $\mu(\mathbf{S})$ Chebyshev polynomial basis.

The value function is:

$\hat{V} = \delta\mu(\mathbf{S})\hat{\beta}$.

For more detils see Fenichel and Abbott (2014), Fenichel et al. (2016a), Fenichel et al. (2016b), and Yun et al. (2017).

Value

A list of simulation resuts: shadow (accounting) prices, inclusive wealth, Value function, stock, and W values. Use `results$item` (or `results[["item"]]`) to import each result item.

`shadowp`	Shadow price
`iweach`	Inclusive wealth for each stock for multi-stock case
`vfun`	Value function
`stock`	Stock
`wval`	W-value if `wval` is provided

References

Fenichel, Eli P. and Joshua K. Abbott. (2014) "Natural Capital: From Metaphor to Measurement." *Journal of the Association of Environmental Economists*. 1(1/2):1-27.
Fenichel, Eli P., Joshua K. Abbott, Jude Bayham, Whitney Boone, Erin M. K. Haacker, and Lisa Pfeiffer. (2016a) "Measuring the Value of Groundwater and Other Forms of Natural Capital." *Proceedings of the National Academy of Sciences*.113:2382-2387.
Fenichel, Eli P., Simon A. Levin, Bonnie McCay, Kevin St. Martin, Joshua K. Abbott, and Malin L. Pinsky. (2016b) "Wealth Reallocation and Sustainability under Climate Change." *Nature Climate change*.6:237-244.
Yun, Seong Do, Barbara Hutniczak, Joshua K. Abbott, and Eli P. Fenichel. (2017) "Ecosystem Based Management and the Welath of Ecosystems" *Proceedings of the National Academy of Sciences*. (forthcoming).

See Also

`aproxdef`, `vsim`

Examples

```
## 1-D Reef-fish example: see Fenichel and Abbott (2014)
data("GOM")
nodes <- chebnodegen(param$nodes,param$lowerK,param$upperK)
simuDataV <- cbind(nodes,sdot(nodes,param),profit(nodes,param))
Aspace <- aproxdef(param$order,param$lowerK,param$upperK,param$delta)
vC <- vaprox(Aspace,simuDataV)
# Note vcol function requries a data.frame or matrix!
GOMSimV <- vsim(vC,as.matrix(simuDataV[,1],ncol=1),profit(nodes,param))

# plot shadow (accounting) price: Figure 4 in Fenichel and Abbott (2014)
plotgen(GOMSimV, xlabel="Stock size, s", ylabel="Shadow price")

## 2-D Prey-Predator example
data("lvdata")
aproxdeg <- c(20,20)
lower <- c(0.1,0.1)
upper <- c(1.5,1.5)
delta <- 0.03
lvspace <- aproxdef(aproxdeg,lower,upper,delta)
```

```
lvaproxc <- vaprox(lvspace,lvaproxdata)
lvsim <- vsim(lvaproxc,lvsimdata.time[,2:3])

# plot Biomass
plot(lvsimdata.time[,1], lvsimdata.time[,2], type='l', lwd=2, col="blue",
     xlab="Time",
     ylab="Biomass")
lines(lvsimdata.time[,1], lvsimdata.time[,3], lwd=2, col="red")
legend("topright", c("Prey", "Predator"), col=c("blue", "red"),
       lty=c(1,1), lwd=c(2,2), bty="n")

# plot shadow (accounting) prices
plot(lvsimdata.time[,1],lvsim[["shadowp"]][,1],type='l', lwd=2, col="blue",
     ylim = c(-5,7),
     xlab="Time",
     ylab="Shadow price")
lines(lvsimdata.time[,1],lvsim[["shadowp"]][,2], lwd=2, col="red")
legend("topright", c("Prey", "Predator"), col=c("blue", "red"),
       lty=c(1,1), lwd=c(2,2), bty="n")

# plot inclusive weath and value function
plot(lvsimdata.time[,1],lvsim[["iw"]],type='l', lwd=2, col="blue",
     ylim = c(-0.5,1.2),
     xlab="Time",
     ylab="Inclusive Wealth / Value Function ($)")
lines(lvsimdata.time[,1],lvsim[["vfun"]], lwd=2, col="red")
legend("topright", c("Inclusive Wealth", "Value Function"),
       col=c("blue", "red"), lty=c(1,1), lwd=c(2,2), bty="n")
```

Index