Metodi Stocastici per la Finanza

Tiziano Vargiolu
vargiolu@math.unipd.it[1]

[1]Università degli Studi di Padova

Anno Accademico 2012-2013
Lezione 8

Indice

Indice

Indice

Modelli affini

Un modello affine è un modello per il tasso corto r tale che i prezzi degli zero-coupon bonds $p(t, T)$ al tempo t con maturità T possono essere espressi mediante

$$p(t, T) = \exp(A(t, T) - B(t, T) r_t)$$

Se il tasso corto r è un processo di Itô, questo accade se r ha dinamica

$$dr_t = (\alpha(t) r_t + \beta(t))\, dt + \sqrt{\gamma(t) r_t + \delta(t)}\, dW_t$$

dove $(\alpha, \beta, \gamma, \delta)$ sono opportune funzioni deterministiche di t.
Il legame tra queste e le funzioni A e B che compaiono nel prezzo dei bonds è dato dal fatto che A e B sono soluzioni delle seguenti equazioni differenziali ordinarie

$$\begin{cases} B_t(t, T) + \alpha(t) B(t, T) - \frac{1}{2}\gamma(t) B^2(t, T) = -1, & B(T, T) = 0, \\ A_t(t, T) = \beta(t) B(t, T) - \frac{1}{2}\delta(t) B^2(t, T), & A(T, T) = 0 \end{cases}$$

Modelli affini

Un modello affine è un modello per il tasso corto r tale che i prezzi degli zero-coupon bonds $p(t,T)$ al tempo t con maturità T possono essere espressi mediante

$$p(t,T) = \exp(A(t,T) - B(t,T)r_t)$$

Se il tasso corto r è un processo di Itô, questo accade se r ha dinamica

$$dr_t = (\alpha(t)r_t + \beta(t))\,dt + \sqrt{\gamma(t)r_t + \delta(t)}\,dW_t$$

dove $(\alpha, \beta, \gamma, \delta)$ sono opportune funzioni deterministiche di t.
Il legame tra queste e le funzioni A e B che compaiono nel prezzo dei bonds è dato dal fatto che A e B sono soluzioni delle seguenti equazioni differenziali ordinarie

$$\begin{cases} B_t(t,T) + \alpha(t)B(t,T) - \frac{1}{2}\gamma(t)B^2(t,T) = -1, & B(T,T) = 0, \\ A_t(t,T) = \beta(t)B(t,T) - \frac{1}{2}\delta(t)B^2(t,T), & A(T,T) = 0 \end{cases}$$

Modelli affini

Un modello affine è un modello per il tasso corto r tale che i prezzi degli zero-coupon bonds $p(t,T)$ al tempo t con maturità T possono essere espressi mediante

$$p(t,T) = \exp(A(t,T) - B(t,T)r_t)$$

Se il tasso corto r è un processo di Itô, questo accade se r ha dinamica

$$dr_t = (\alpha(t)r_t + \beta(t))\,dt + \sqrt{\gamma(t)r_t + \delta(t)}\,dW_t$$

dove $(\alpha, \beta, \gamma, \delta)$ sono opportune funzioni deterministiche di t.
Il legame tra queste e le funzioni A e B che compaiono nel prezzo dei bonds è dato dal fatto che A e B sono soluzioni delle seguenti equazioni differenziali ordinarie

$$\begin{cases} B_t(t,T) + \alpha(t)B(t,T) - \frac{1}{2}\gamma(t)B^2(t,T) = -1, & B(T,T) = 0, \\ A_t(t,T) = \beta(t)B(t,T) - \frac{1}{2}\delta(t)B^2(t,T), & A(T,T) = 0 \end{cases}$$

Modello di Vasicek

Nel modello di Vasicek (1978), il tasso corto è soluzione di

$$dr_t = a(b - r_t)\ dt + \sigma\ dW_t$$

e si può scrivere direttamente la soluzione esplicita

$$r_t = e^{-at} r_0 + b \int_0^t e^{-a(t-u)}\ du + \sigma \int_0^t e^{-a(t-u)}\ dW_u$$

e si possono calcolare esplicitamente molte quantità mediante questa formula esplicita.
In particolare, r è un processo gaussiano con media

$$\mathbb{E}[r_t] = e^{-at} r_0 + b \int_0^t e^{-a(t-u)}\ du$$

e varianza

$$\mathrm{Var}[r_t] = \sigma^2 \int_0^t e^{-2a(t-u)}\ du$$

Modello di Vasicek

Nel modello di Vasicek (1978), il tasso corto è soluzione di

$$dr_t = a(b - r_t)\ dt + \sigma\ dW_t$$

e si può scrivere direttamente la soluzione esplicita

$$r_t = e^{-at} r_0 + b \int_0^t e^{-a(t-u)}\ du + \sigma \int_0^t e^{-a(t-u)}\ dW_u$$

e si possono calcolare esplicitamente molte quantità mediante questa formula esplicita.
In particolare, r è un processo gaussiano con media

$$\mathbb{E}[r_t] = e^{-at} r_0 + b \int_0^t e^{-a(t-u)}\ du$$

e varianza

$$\mathrm{Var}[r_t] = \sigma^2 \int_0^t e^{-2a(t-u)}\ du$$

Modello di Vasicek come modello affine

In alternativa, seguendo la teoria generale dei modelli affini, poniamo

$$\alpha(t) \equiv -a, \quad \beta(t) \equiv ab, \quad \gamma(t) \equiv 0, \quad \delta_t \equiv \sigma^2$$

e quindi i due sistemi di equazioni ordinarie diventano

$$\begin{cases} B_t(t,T) = aB(t,T) - 1, & B(T,T) = 0, \\ A_t(t,T) = abB(t,T) - \frac{1}{2}\sigma^2 B^2(t,T), & A(T,T) = 0 \end{cases}$$

che hanno soluzioni esplicite

$$\begin{aligned} B(t,T) &= \frac{1 - e^{-a(T-t)}}{a}, \\ A(t,T) &= -b(T-t) + b\frac{1 - e^{-a(T-t)}}{a} - \frac{\sigma^2}{4a^3}(1 - e^{-a(T-t)})^2 + \\ &\quad + \frac{\sigma^2}{2a^2}\left(T - t - \frac{1 - e^{-a(T-t)}}{a}\right) \end{aligned}$$

Tasso yield sotto il modello di Vasicek

Calcoliamo ora il tasso continuo $R(t,T)$: siccome questo tasso soddisfa $p(t,T) = \exp(-(T-t)R(t,T))$, allora dobbiamo avere

$$
\begin{aligned}
R(t,T) &= \frac{1}{T-t}(B(t,T)r_t - A(t,T)) = \\
&= \frac{1-e^{-a(T-t)}}{a(T-t)} r_t + R_\infty \left(1 - \frac{1-e^{-a(T-t)}}{a(T-t)}\right) + \\
&\quad + \frac{\sigma^2}{4a^3} \frac{(1-e^{-a(T-t)})^2}{T-t}
\end{aligned}
$$

dove $R_\infty := b - \frac{\sigma^2}{2a^2}$ è il "tasso lungo": difatti, fissati t e r_t, si ha che

$$\lim_{T\to\infty} R(t,T) = 0 \cdot r_t + R_\infty(1-0) + 0 = R_\infty$$

La variabile r_t continua a conservare la sua interpretazione di tasso a breve: difatti fissati t e r_t, si ha che

$$\lim_{T\to t^+} R(t,T) = 1 \cdot r_t + R_\infty(1-1) + 0 = r_t$$

Tasso yield sotto il modello di Vasicek

Calcoliamo ora il tasso continuo $R(t,T)$: siccome questo tasso soddisfa $p(t,T) = \exp(-(T-t)R(t,T))$, allora dobbiamo avere

$$\begin{aligned} R(t,T) &= \frac{1}{T-t}(B(t,T)r_t - A(t,T)) = \\ &= \frac{1-e^{-a(T-t)}}{a(T-t)} r_t + R_\infty \left(1 - \frac{1-e^{-a(T-t)}}{a(T-t)}\right) + \\ &\quad + \frac{\sigma^2}{4a^3} \frac{(1-e^{-a(T-t)})^2}{T-t} \end{aligned}$$

dove $R_\infty := b - \frac{\sigma^2}{2a^2}$ è il "tasso lungo": difatti, fissati t e r_t, si ha che

$$\lim_{T\to\infty} R(t,T) = 0 \cdot r_t + R_\infty(1-0) + 0 = R_\infty$$

La variabile r_t continua a conservare la sua interpretazione di tasso a breve: difatti fissati t e r_t, si ha che

$$\lim_{T\to t^+} R(t,T) = 1 \cdot r_t + R_\infty(1-1) + 0 = r_t$$

Calibrazione con il modello di Vasicek

La calibrazione con un modello come quello di Vasicek, in cui il modello cioè prevede una specifica forma funzionale per i tassi di interesse in funzione della maturità, procede in questo modo.

Date determinate maturità $T_1, \ldots, T_n$ per cui si possano ricavare i tassi $\tilde{R}(t, T_i)$, $i = 1, \ldots, n$ dal mercato, per ogni a, b, σ si possono calcolare i tassi teorici $R(t, T) = R(t, T; a, b, \sigma)$ in base alle formule viste sopra.

Il metodo più usato per calibrare il modello è allora quello di minimizzare la somma degli scarti quadratici, cioè risolvere il problema

$$\min_{a,b,\sigma} \sum_{i=1}^{n} (\tilde{R}(t, T_i) - R(t, T_i; a, b, \sigma))^2 \tag{1}$$

che in effetti è un caso particolare di calibrazione parametrica.

Calibrazione con il modello di Vasicek

La calibrazione con un modello come quello di Vasicek, in cui il modello cioè prevede una specifica forma funzionale per i tassi di interesse in funzione della maturità, procede in questo modo. Date determinate maturità $T_1, \ldots, T_n$ per cui si possano ricavare i tassi $\tilde{R}(t, T_i)$, $i = 1, \ldots, n$ dal mercato, per ogni a, b, σ si possono calcolare i tassi teorici $R(t, T) = R(t, T; a, b, \sigma)$ in base alle formule viste sopra.

Il metodo più usato per calibrare il modello è allora quello di minimizzare la somma degli scarti quadratici, cioè risolvere il problema

$$\min_{a,b,\sigma} \sum_{i=1}^{n} (\tilde{R}(t, T_i) - R(t, T_i; a, b, \sigma))^2 \tag{1}$$

che in effetti è un caso particolare di calibrazione parametrica.

Calibrazione con il modello di Vasicek

La calibrazione con un modello come quello di Vasicek, in cui il modello cioè prevede una specifica forma funzionale per i tassi di interesse in funzione della maturità, procede in questo modo.
Date determinate maturità $T_1, \ldots, T_n$ per cui si possano ricavare i tassi $\tilde{R}(t, T_i)$, $i = 1, \ldots, n$ dal mercato, per ogni a, b, σ si possono calcolare i tassi teorici $R(t, T) = R(t, T; a, b, \sigma)$ in base alle formule viste sopra.
Il metodo più usato per calibrare il modello è allora quello di minimizzare la somma degli scarti quadratici, cioè risolvere il problema

$$\min_{a,b,\sigma} \sum_{i=1}^{n} (\tilde{R}(t, T_i) - R(t, T_i; a, b, \sigma))^2 \qquad (1)$$

che in effetti è un caso particolare di calibrazione parametrica.

Ottimizzazione non lineare

Per risolvere il problema

$$\begin{cases} \min \phi(x) \\ g(x) = 0, \\ h(x) \geq 0, \end{cases}$$

si usa l'istruzione

`[x, obj, info, iter, nf, lambda] = sqp(x0,@phi,@g,@h)`

L'input consiste in:

- `x0`: valore iniziale;
- `@phi`, `@g`, `@h`: funzioni predefinite. `@g` e `@h` possono essere sostituite da `[]`.

Output:

- `x`: ottimo;
- `obj`: valore ottimo;
- `info`: codice di funzionamento della routine;
- `iter`: n. iterazioni occorse;
- `lambda`: valore dei moltiplicatori di Lagrange.

Ottimizzazione non lineare

Per risolvere il problema

$$\begin{cases} \min \phi(x) \\ g(x) = 0, \\ h(x) \geq 0, \end{cases}$$

si usa l'istruzione

`[x, obj, info, iter, nf, lambda] = sqp(x0,@phi,@g,@h)`

L'input consiste in:

- `x0`: valore iniziale;
- `@phi`, `@g`, `@h`: funzioni predefinite. `@g` e `@h` possono essere sostituite da `[]`.

Output:

- `x`: ottimo;
- `obj`: valore ottimo;
- `info`: codice di funzionamento della routine;
- `iter`: n. iterazioni occorse;
- `lambda`: valore dei moltiplicatori di Lagrange

Ottimizzazione non lineare

Per risolvere il problema

$$\begin{cases} \min \phi(x) \\ g(x) = 0, \\ h(x) \geq 0, \end{cases}$$

si usa l'istruzione

`[x, obj, info, iter, nf, lambda] = sqp(x0,@phi,@g,@h)`

L'input consiste in:

- `x0`: valore iniziale;
- `@phi`, `@g`, `@h`: funzioni predefinite. `@g` e `@h` possono essere sostituite da `[]`.

Output:

- `x`: ottimo;
- `obj`: valore ottimo;
- `info`: codice di funzionamento della routine;
- `iter`: n. iterazioni occorse;
- `lambda`: valore dei moltiplicatori di Lagrange

Ottimizzazione non lineare

Per risolvere il problema

$$\begin{cases} \min \phi(x) \\ g(x) = 0, \\ h(x) \geq 0, \end{cases}$$

si usa l'istruzione

```
[x, obj, info, iter, nf, lambda] = sqp(x0,@phi,@g,@h)
```

L'input consiste in:

- `x0`: valore iniziale;
- `@phi`, `@g`, `@h`: funzioni predefinite. `@g` e `@h` possono essere sostituite da `[]`.

Output:

- `x`: ottimo;
- `obj`: valore ottimo;
- `info`: codice di funzionamento della routine;
- `iter`: n. iterazioni occorse;
- `lambda`: valore dei moltiplicatori di Lagrange.

Ottimizzazione parametrica

Per risolvere il problema, dipendente dal parametro a,

$$\begin{cases} \min \phi_a(x) \\ g(x) = 0, \\ h(x) \geq 0, \end{cases}$$

si usa l'istruzione

```
[x, obj, info, iter, nf, lambda] = sqp(x0,@(x) phi(x,a),@g,@h)
```

in questo modo:

```
> z0 = [0.03;-0.03;-0.02;0.5]
> [x, obj, info, iter, nf, lambda] = sqp(z0,@(z) phiVasicek(T,R,z),[],[])
```

Ottimizzazione parametrica

Per risolvere il problema, dipendente dal parametro a,

$$\begin{cases} \min \phi_a(x) \\ g(x) = 0, \\ h(x) \geq 0, \end{cases}$$

si usa l'istruzione

```
[x, obj, info, iter, nf, lambda] = sqp(x0,@(x) phi(x,a),@g,@h)
```

in questo modo:

```
> z0 = [0.03;-0.03;-0.02;0.5]
> [x, obj, info, iter, nf, lambda] = sqp(z0,@(z) phiVasicek(T,R,z),[],[])
```

Ottimizzazione parametrica

Per risolvere il problema, dipendente dal parametro a,

$$\begin{cases} \min \phi_a(x) \\ g(x) = 0, \\ h(x) \geq 0, \end{cases}$$

si usa l'istruzione

```
[x, obj, info, iter, nf, lambda] = sqp(x0,@(x) phi(x,a),@g,@h)
```

in questo modo:

```
> z0 = [0.03;-0.03;-0.02;0.5]
> [x, obj, info, iter, nf, lambda] = sqp(z0,@(z) phiVasicek(T,R,z),[],[])
```

File Vasicek.m

```
function obj = Vasicek(T,a,b,sigma,r0)

e = exp(-a*T);

Rinf = b - sigma^2/(2*a^2);

obj = (1 - e)*r0/(a*T) + Rinf * (1 - (1 - e)/(a*T)) + ...
    sigma^2/(4*a^3) * (1 - e)^2/T;

endfunction
```

File plotVasicek.m

```
function plotVasicek(a,b,sigma,r0,T,delta)

N = T/delta;

for u = 1:N
    g(u) = Vasicek(u*delta,a,b,sigma,r0);
    v(u) = u*delta;
end

plot(v,g,0)

endfunction
```

File phiVasicek.m

```
function obj = phiVasicek(T,R,z,r0)

obj = 0;

[n m] = size(R);

a = z(1);
b = z(2);
sigma = z(3);

for i = 2:n
 obj = obj + (R(i) - Vasicek(T(i) - T(1),a,b,sigma,r0))^2;
end

endfunction
```

Calibrazione "globale"

Siamo ora pronti per calibrare la curva zero-coupon. Per fare questo, si usa la funzione di minimizzazione sqp, e in particolare le seguenti istruzioni Octave:

- `z0 = [a;b;sigma]`
  inizializziamo il vettore di parametri poichè sqp ha bisogno di un punto di partenza;
- `[x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T,R,z,R(2)),[],[])`
  minimizza la funzione $z \to phiVasicek(T, R, z)$ partendo da z0 e senza vincoli

Disegnamo questa curva contro la curva originale:

```
> plotVasicek(x(1),x(2),x(3),R(2),50,0.1)
> plot(T,R,0)
```

Sono molto diverse: il modello di Vasicek riesce a catturare solo alcune curve di tassi "tipiche", ma ci possono essere particolari curve che non sono di quella forma.

Calibrazione "globale"

Siamo ora pronti per calibrare la curva zero-coupon. Per fare questo, si usa la funzione di minimizzazione `sqp`, e in particolare le seguenti istruzioni Octave:

- `z0 = [a;b;sigma]`
  inizializziamo il vettore di parametri poichè `sqp` ha bisogno di un punto di partenza;
- `[x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T,R,z,R(2)),[],[])`
  minimizza la funzione $z \to phiVasicek(T, R, z)$ partendo da `z0` e senza vincoli

Disegnamo questa curva contro la curva originale:

```
> plotVasicek(x(1),x(2),x(3),R(2),50,0.1)
> plot(T,R,0)
```

Sono molto diverse: il modello di Vasicek riesce a catturare solo alcune curve di tassi "tipiche", ma ci possono essere particolari curve che non sono di quella forma.

Calibrazione "globale"

Siamo ora pronti per calibrare la curva zero-coupon. Per fare questo, si usa la funzione di minimizzazione `sqp`, e in particolare le seguenti istruzioni Octave:

- `z0 = [a;b;sigma]`
  inizializziamo il vettore di parametri poichè `sqp` ha bisogno di un punto di partenza;
- `[x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T,R,z,R(2)),[],[])`
  minimizza la funzione $z \rightarrow phiVasicek(T, R, z)$ partendo da z0 e senza vincoli

Disegnamo questa curva contro la curva originale:

- `plotVasicek(x(1),x(2),x(3),R(2),50,0.1)`
- `plot(T,R,0)`

Sono molto diverse: il modello di Vasicek riesce a catturare solo alcune curve di tassi "tipiche", ma ci possono essere particolari curve che non sono di quella forma.

Calibrazione "globale"

Siamo ora pronti per calibrare la curva zero-coupon. Per fare questo, si usa la funzione di minimizzazione `sqp`, e in particolare le seguenti istruzioni Octave:

- `z0 = [a;b;sigma]`
  inizializziamo il vettore di parametri poichè `sqp` ha bisogno di un punto di partenza;
- `[x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T,R,z,R(2)),[],[])`
  minimizza la funzione $z \to phiVasicek(T, R, z)$ partendo da z0 e senza vincoli

Disegnamo questa curva contro la curva originale:

```
> plotVasicek(x(1),x(2),x(3),R(2),50,0.1)
> plot(T,R,0)
```

Sono molto diverse: il modello di Vasicek riesce a catturare solo alcune curve di tassi "tipiche", ma ci possono essere particolari curve che non sono di quella forma.

Calibrazione "globale"

Siamo ora pronti per calibrare la curva zero-coupon. Per fare questo, si usa la funzione di minimizzazione `sqp`, e in particolare le seguenti istruzioni Octave:

- `z0 = [a;b;sigma]`
  inizializziamo il vettore di parametri poichè `sqp` ha bisogno di un punto di partenza;
- `[x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T,R,z,R(2)),[],[])`
  minimizza la funzione $z \to phiVasicek(T, R, z)$ partendo da z0 e senza vincoli

Disegnamo questa curva contro la curva originale:

```
> plotVasicek(x(1),x(2),x(3),R(2),50,0.1)
> plot(T,R,0)
```

Sono molto diverse: il modello di Vasicek riesce a catturare solo alcune curve di tassi "tipiche", ma ci possono essere particolari curve che non sono di quella forma.

Calibrazione "per comparti": maturità corte

Si può però tentare una calibrazione "per comparti": ci si accontenta cioè di calibrare la curva solo su un sottointervallo $(a, b) \subseteq (0, 50)$. Questo può essere fatto fornendo come input alla funzione `phiVasicek` non tutti i vettori `T` ed `R`, ma solo alcune componenti.

Proviamo all'inizio con il comparto $(0, 6)$ (maturità fino a 6 anni): dato che $6 = T_{14}$, la sintassi è la seguente

```
> [x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T(1:14),R(1:14),z,R(2)),[],[])
> plotVasicek(x(1),x(2),x(3),R(2),6,0.1)
> plot(T(1:14),R(1:14),0)
```

Da qui si vede che la calibrazione è molto migliore di prima, ma ovviamente limitata alle maturità fino a 6 anni: in particolare, il vettore ottimale dei coefficienti è

`x = [2.628075;0.084058;0.730133]`.

Calibrazione "per comparti": maturità corte

Si può però tentare una calibrazione "per comparti": ci si accontenta cioè di calibrare la curva solo su un sottointervallo $(a, b) \subseteq (0, 50)$. Questo può essere fatto fornendo come input alla funzione `phiVasicek` non tutti i vettori `T` ed `R`, ma solo alcune componenti.
Proviamo all'inizio con il comparto $(0, 6)$ (maturità fino a 6 anni): dato che $6 = T_{14}$, la sintassi è la seguente

```
> [x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T(1:14),R(1:14),z,R(2)),[],[])
> plotVasicek(x(1),x(2),x(3),R(2),6,0.1)
> plot(T(1:14),R(1:14),0)
```

Da qui si vede che la calibrazione è molto migliore di prima, ma ovviamente limitata alle maturità fino a 6 anni: in particolare, il vettore ottimale dei coefficienti è
`x = [2.628075;0.084058;0.730133]`.

Calibrazione "per comparti": maturità corte

Si può però tentare una calibrazione "per comparti": ci si accontenta cioè di calibrare la curva solo su un sottointervallo $(a, b) \subseteq (0, 50)$. Questo può essere fatto fornendo come input alla funzione `phiVasicek` non tutti i vettori `T` ed `R`, ma solo alcune componenti.
Proviamo all'inizio con il comparto $(0, 6)$ (maturità fino a 6 anni): dato che $6 = T_{14}$, la sintassi è la seguente

```
> [x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T(1:14),R(1:14),z,R(2)),[],[])
> plotVasicek(x(1),x(2),x(3),R(2),6,0.1)
> plot(T(1:14),R(1:14),0)
```

Da qui si vede che la calibrazione è molto migliore di prima, ma ovviamente limitata alle maturità fino a 6 anni: in particolare, il vettore ottimale dei coefficienti è
`x = [2.628075;0.084058;0.730133]`.

Calibrazione "per comparti": maturità lunghe

Vediamo un altro esempio, con il comparto $(1, 50)$ (tutte le maturità tranne quelle fino a 1 anno). Dato che $1 = T_9$ e $50 = T_{42}$, la sintassi è la seguente:

```
> [x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T(9:42),R(9:42),z,R(2)),[],[])
> plotVasicek(x(1),x(2),x(3),R(2),50,0.1)
> plot(T,R,0)
```

Anche qui si osserva una calibrazione molto migliore della precedente, ma solo per periodi più lunghi di 1 anno: stavolta il vettore ottimale è `x = [2.745642;0.054769;0.304101]`.

Calibrazione "per comparti": maturità lunghe

Vediamo un altro esempio, con il comparto $(1, 50)$ (tutte le maturità tranne quelle fino a 1 anno). Dato che $1 = T_9$ e $50 = T_{42}$, la sintassi è la seguente:

```
> [x,obj,info,iter,nf,lambda] = sqp(z0,@(z) phiVasicek(T(9:42),R(9:42),z,R(2)),[],[])
> plotVasicek(x(1),x(2),x(3),R(2),50,0.1)
> plot(T,R,0)
```

Anche qui si osserva una calibrazione molto migliore della precedente, ma solo per periodi più lunghi di 1 anno: stavolta il vettore ottimale è `x = [2.745642;0.054769;0.304101]`.