## Dynamic Programming with Piecewise Linear Interpolation Dynamic Programming with Piecewise Linear<br>Interpolation<br>Kenneth Judd and Yongyang Cai<br>April 20, 2011<br>0.1 Piecewise Linear Interpolation<br>If Lagrange data  $\{(x_i, v_i) : i = 1, ..., m\}$  is given, then its piecewise linear inter-Programming with Piecew<br>
Interpolation<br>
Kenneth Judd and Yongyang Cai<br>
April 20, 2011<br>
<br>
inear Interpolation<br>
<br>  $\hat{v}_i, v_i) : i = 1, ..., m\}$  is given, then its p<br>  $\hat{V}(x) = b_{j,0} + b_{j,1}x,$  if  $x \in [x_j, x_{j+1}],$ be interpolation<br>
by Judd and Yong<br>
h Judd and Yong<br>
h Judd and Yong<br>
h Judd and Yong<br>
terpolation<br>  $\text{tr}_1, \ldots, m\}$  is given<br>  $\text{tr}_i, 0 + b_{j,1}x, \quad \text{if } x \in$ <br>  $b_{j,1} = \frac{v_{j+1} - v_j}{x_{j+1} - x_j}$ g with<br>olation<br>nd Yong<br>20, 2011<br>tion<br>} is given<br>if  $x \in$ <br> $\frac{v_{j+1} - v_j}{x_{j+1} - x_j}$

Kenneth Judd and Yongyang Cai

April 20, 2011

## 0.1 Piecewise Linear Interpolation

polation is **0.1 Piecewise L**<br>If Lagrange data {(*x*<br>polation is<br>where<br>for  $j = 1, \ldots, m - 1$ .

$$
\hat{V}(x) = b_{j,0} + b_{j,1}x, \quad \text{if } x \in [x_j, x_{j+1}],
$$

where

Continuting with the Judd and Yongys

\nApril 20, 2011

\ninterpolation

\n
$$
= 1, \ldots, m
$$
 is given,

\n
$$
b_{j,0} + b_{j,1}x, \quad \text{if } x \in
$$

\n
$$
b_{j,1} = \frac{v_{j+1} - v_j}{x_{j+1} - x_j},
$$

\n
$$
b_{j,0} = v_i - b_{j,1}x_i,
$$

In the maximization step of numerical DP algorithms, one could directly solve the maximization problem tions  $v_i^t$ merical DP algori $u_t(x_i, a_i) + \beta \hat{V}$  ( thr merical DP a<br> $u_t(x_i, a_i) + \beta$ <br> $x^+ = g(x, a)$ In the maximization step of numerical DP algorithms, one could directly solve<br>
maximization problem<br>  $v_i^t = \max_{a_i \in \mathcal{D}(x_i, t)} u_t(x_i, a_i) + \beta \hat{V}(x_i^+; \mathbf{b}^{t+1})$ <br>
are<br>  $x^+ = g(x, a)$ <br>
Problem:  $\hat{V}(x; \mathbf{b}^{t+1})$  is not differ

$$
v_i^t = \max_{a_i \in \mathcal{D}(x_i, t)} u_t(x_i, a_i) + \beta \hat{V}(x_i^+; \mathbf{b}^{t+1})
$$

where

$$
x^+ = g(x, a)
$$

In the maximization step<br>the maximization problem<br> $v_i^t = \max_{a_i \in \mathcal{C}}$ <br>where<br>Problem:  $\hat{V}(x; \mathbf{b}^{t+1})$  is<br>optimization problem for a.

## 0.2 Min-Function Approach  $\mathbf{u}$ -**I**<br> $v_i^t$

The differentiability problem is solved as follows:

**1-Function Approach**

\nntiability problem is solved as follows:

\n
$$
v_i^t = \max_{a_i \in \mathcal{D}(x_i, t), y_{ik}, x_{ik}^+} u_t(x_i, a_i) + \beta y_i
$$
\n
$$
\text{s.t.} \quad x_i^+ = g(x_i, a)
$$
\n
$$
y_i \le b_{j,0}^{(t+1)} + b_{j,1}^{(t+1)} x_i^+, \quad 1 \le j < m
$$

The objective function is smooth and inequality constraints are linear and sparse so that we can apply fast Newton type optimization algorithms to solve this prob-**0.2** Min-Function Approach<br>
The differentiability problem is solved as follows:<br>  $v_i^t = \max_{a_i \in \mathcal{D}(x_i, t), y_{ik}, x_{ik}^+} u_t(x_i, a_i) + \beta y_i$ <br>
s.t.  $x_i^+ = g(x_i, a)$ <br>  $y_i \leq b_{j,0}^{(t+1)} + b_{j,1}^{(t+1)} x_i^+, \quad 1 \leq j < m$ <br>
The objective funct constraints, few of them will be active, such that optimization softwares can still solve the new model quickly. **Min-Function Approach**<br>
edifferentiability problem is solved as follows:<br>  $v_i^t = \max_{a_i \in \mathcal{D}(x_i, t), y_{ik}, x_{ik}^+} u_t(x_i, a_i) + \beta y_i$ <br>
s.t.  $x_i^+ = g(x_i, a)$ <br>  $y_i \leq b_{j,0}^{(t+1)} + b_{j,1}^{(t+1)} x_i^+, \quad 1 \leq j <$ <br>
e objective function is smoo

Moreover, this way does not need to find the interval where  $x_{ik}^{+}$  locates, while the naive way has to.

## 0.3 Convex-Set Approach

Both previous methods need to calculate coefficients; this is very complicated for multi-dimensional piecewise linear interpolation.  $\begin{array}{c} \text{See} \ \text{at} \ \text{l} \ \text{in} \ \text{in} \ \text{v}_i^t \end{array}$ ficients; this is very complicated for<br>ion.<br>ipute the coefficients explicitly:<br> $u_t(x_i, a_i) + \beta y_i,$  (1)

The following method has no need to compute the coefficients explicitly:

Set Approach  
athods need to calculate coefficients; this is very complicated for  
l piecewise linear interpolation.  
; method has no need to compute the coefficients explicitly:  

$$
v_i^t = \max_{a_i \in \mathcal{D}(x_i, t), \mu_j \ge 0, y_i, x_i^+} u_t(x_i, a_i) + \beta y_i,
$$
(1)  
s.t. 
$$
x_i^+ = g(x_i, a),
$$

$$
x_i^+ \le \sum_{j=1}^m \mu_j x_j^{(t+1)}
$$

$$
y_i = \sum_{j=1}^m \mu_j v_j^{(t+1)}
$$

$$
\sum_{j=1}^m \mu_j = 1
$$