
Gary S. Anderson
gary.anderson@frb.gov

Board of Governors of the Federal Reserve System
Division of Monetary Affairs

Society for Computational Economics
18th International Conference
Computing in Economics and Finance
Outline

1. Introduction
2. Authors
3. Abstract Classes
4. Summary
5. Bibliography
Outline

1 Introduction

2 Authors

3 Abstract Classes

4 Summary

5 Bibliography
Increased interest in solving models with constraints that occasionally bind
- multi-sector models with limits on inter-sectoral mobility of factors
- heterogeneous agent models with constraints on financial assets available to agents
- macro models with a zero lower bound on nominal interest rates

Long history of researchers developing and applying a variety of strategies

Challenging problems. Even without constraints must work with approximations
An Example

- As a concrete example (Christiano & Fisher, 2000) studied a simple stochastic growth model with irreversible investment.

\[
\max_E E_0 \sum_{t=0}^{\infty} \beta^t U(c_t)
\]

with

\[
c_t + e^{k_{t+1}} - (1 - \delta)e^{k_t} \leq f(k_t, \theta_t) \equiv e^{(\theta_t + \alpha k_t)}
\]

and gross investment non-negative

\[
e^{k_{t+1}} - (1 - \delta)e^{k_t} \geq 0
\]

- They provided several equation systems characterizing a solution
Policy and Lagrange Multiplier Functions

Find time invariant functions $g, h$ such that given

$$R(k, \theta; g, h) = U_c(k, g(k, \theta), \theta) - h(k, \theta) - \beta \int m(g(k, \theta), \theta'; g, h)p_\theta(\theta'|\theta)d\theta'$$

then

$$m(k', \theta'; g, h) = U_c(k', g(k', \theta', \theta') [f_k(k', \theta') + 1 - \delta] - h(k', \theta')(1 - \delta) \geq 0$$

$$R(k, \theta; g, h) = 0$$

$$e^{g(k, \theta)} - (1 - \delta)e^k \geq 0, h(k, \theta) \geq 0$$

$$h(k, \theta)[e^{g(k, \theta)} - (1 - \delta)e^k] = 0$$
A Parameterized Expectations Solution

The following is one of several PEA solutions in (Christiano & Fisher, 2000). Find $\gamma$ such that

$$\bar{R}(k, \theta; \gamma) = 0$$

where

$$\bar{R}(k, \theta; \gamma) = e^{\gamma(k, \theta)} - \int m(g(k, \theta), \theta'; g, h) p_\theta(\theta' | \theta) d\theta'$$

with $g, h$ implicitly defined by

$$U_c(k, \bar{g}(K, \theta), \theta) = \beta e^{\gamma(k, \theta)}$$

with

$$g(k, \theta) = \begin{cases} \bar{g}(k, \theta) & \text{if } \bar{g}(k, \theta) > \log(1 - \delta) + k \\ \log(1 - \delta) + k & \text{otherwise} \end{cases}$$
Some Authors Providing Downloadable Code

- (Haefke, 1998) FORTRAN, Gauss, MATLAB
- (Maliar & Maliar, 2005a; Maliar & Maliar, 2005b) MATLAB
- (Aruoba et al., 2006) FORTRAN90, MATLAB
- (Carroll, 2006) Mathematica, MATLAB.
- (Adam & Billi, 2006; Adam & Billi, 2007; Billi, 2007) MATLAB
- (Nakov, 2008) MATLAB
- (Hintermaier & Koeniger, 2010) MATLAB
- (Fella, 2011) FORTRAN95
- (Gordon, 2011) MATLAB
- (Fernández-Villaverde et al., 2012) FORTRAN90
- (Iskhakov et al., 2012) MATLAB Presented yesterday at this conference
Some Authors Providing Algorithms Only

- (Marcet & Marshall, 1994)
- (Krusell et al., 1997)
- (Christiano & Fisher, 2000)
- (Grüne & Semmler, 2004)
- (Dennis, 2007)
- (Benigno et al., 2009)
- (Brumm & Grill, 2010)
- (Judd et al., 2010)
- (Marcet & Marimon, 2011)
- (Malin et al., 2011)
- (Ludwig & Schön, 2012)
Generic Algorithm

1. Choose a function approximation method
2. Choose a metric for judging approximation quality
3. Guess parameters characterizing functions solving the equation system
4. Use equation system to generate an improved guess
   - identify strategic ordering of subsets of the equation system to facilitate solution
   - identify function evaluation points
   - compute expected values
   - solve for new parameters characterizing functions
5. If functions changed significantly repeat 4
Little Apparent Code Reuse

- Very limited code reuse
  - Notable exception – several authors use the COMPECON tools (Miranda & Fackler, 2002)
- Down-loadable code typically written in very model specific ways
- Similarities in goals and methods hidden by idiosyncratic model differences and coding conventions
- Author’s with algorithmic innovations typically chose to build a complete DSGE solution framework
Benefits from Interchangeable Components

- Interchangeable components would facilitate experimentation with alternative algorithmic designs
- Instrumented with timers and memory monitors, could help guide production code development
- Design patterns (Gamma et al., 1995)
  
  **Template Method** Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Template Method lets subclasses redefine certain steps of an algorithm without changing the algorithm’s structure.

  **Strategy** Define a family of algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithm vary independently from clients that use it.
Outline

1. Introduction
2. Authors
3. Abstract Classes
4. Summary
5. Bibliography

Anderson

Generic Tools – Occasionally Binding Constraints
Outline

1. Introduction

2. Authors
   - Parametrized Expectations Algorithms (PEAs)
   - Miranda and Fackler
   - Endogenous Grid Method
   - Other Improved Grids
   - Common Components
     - Function Approximation
     - Approximation Metric
     - Evaluation Points
     - Expectations

3. Abstract Classes
(Christiano & Fisher, 2000)

### The Generic Algorithm

<table>
<thead>
<tr>
<th>Component</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Approximation</td>
<td>Chebyshev Polynomials; Finite element piecewise linear</td>
</tr>
<tr>
<td>Approximation Metric</td>
<td>Approximation Parameter Fixed Point; Galerkin and Collocation variants of Weighted Residuals</td>
</tr>
<tr>
<td>Evaluation Points</td>
<td>Chebyshev Nodes</td>
</tr>
<tr>
<td>Expectations</td>
<td>Gaussian quadrature, Monte Carlo Integration</td>
</tr>
</tbody>
</table>

- Implemented several PEA variants
- Piecewise linear approximation methods for policy function iteration (PFI)
Outline

1. Introduction

2. Authors
   - Parametrized Expectations Algorithms (PEAs)
   - Miranda and Fackler
   - Endogenous Grid Method
   - Other Improved Grids
   - Common Components
     - Function Approximation
     - Approximation Metric
     - Evaluation Points
     - Expectations

3. Abstract Classes
The authors provide down-loadable MATLAB code for solving

\[ f[s_t, x_t, E_t h(s_{t+1}, x_{t+1})] = \phi_t \]

where

\[ s_{t+1} = g(s_t, x_t, \epsilon_{t+1}) \]

and

\[ a(s_t) \leq x_t \leq b(s_t), x_{jt} > a_j(s_t) \Rightarrow \phi_{jt} \leq 0, x_{jt} < b_j(s_t) \Rightarrow \phi_{jt} \geq 0, \]
### The Generic Algorithm

<table>
<thead>
<tr>
<th>Component</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Approximation</td>
<td>piecewise linear</td>
</tr>
<tr>
<td>Approximation Metric</td>
<td>Approximated Function at Nodes Fixed Point; Collocation; Time Iteration</td>
</tr>
<tr>
<td>Evaluation Points</td>
<td>uniform grid</td>
</tr>
<tr>
<td>Expectations</td>
<td>Gaussian quadrature</td>
</tr>
</tbody>
</table>

- performs policy function iteration (PFI)
(Nakov, 2008)

<table>
<thead>
<tr>
<th>Component</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Approximation</td>
<td>piecewise linear</td>
</tr>
<tr>
<td>Approximation Metric</td>
<td>Approximation parameter fixed point</td>
</tr>
<tr>
<td>Evaluation Points</td>
<td>uniform grid</td>
</tr>
<tr>
<td>Expectations</td>
<td>Gaussian Quadrature</td>
</tr>
</tbody>
</table>

- performs policy function iteration (PFI)
Outline

1 Introduction

2 Authors
   - Parametrized Expectations Algorithms (PEAs)
   - Miranda and Fackler
   - Endogenous Grid Method
   - Other Improved Grids
   - Common Components
     - Function Approximation
     - Approximation Metric
     - Evaluation Points
     - Expectations

3 Abstract Classes
The Endogenous Grid Method (EGM)

- Originally developed in (Carroll, 2006) – Mathematica and MATLAB code available. Extended to perform value function iteration by (Barillas & Fernandez-Villaverde, 2007) – Fortran90 code available.

- (Krueger & Ludwig, 2007; Rendahl, 2006) show that time iteration, nesting EGM, is often applicable and useful

- Applied to a model with occasionally binding constraints in (Hintermaier & Koeniger, 2010) – MATLAB code

- Extended to a class of non-concave problems in (Fella, 2011; Iskhakov et al., 2012) – Fortran95 and MATLAB code available.
(Barillas & Fernandez-Villaverde, 2007)

<table>
<thead>
<tr>
<th>Component</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Approximation</td>
<td>Piecewise linear</td>
</tr>
<tr>
<td>Approximation Metric</td>
<td>Approximated Function Fixed Point at Nodes</td>
</tr>
<tr>
<td>Evaluation Points</td>
<td>uniform grid</td>
</tr>
<tr>
<td>Expectations</td>
<td>Tauchen –41 discrete states(^a)</td>
</tr>
</tbody>
</table>

\(^a\)Adapted from code for (Ljungqvist & Sargent, 2004; Miranda & Fackler, 2002)

- Extends EGM to value function iteration (VFI)
### The Generic Algorithm

<table>
<thead>
<tr>
<th>Component</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Approximation</td>
<td>Piecewise linear</td>
</tr>
<tr>
<td>Approximation Metric</td>
<td>Approximated Function at Nodes Fixed Point</td>
</tr>
<tr>
<td>Evaluation Points</td>
<td>7 Uniformly spaced points for discrete durable; double exponential grid for assets</td>
</tr>
<tr>
<td>Expectations</td>
<td>Tauchen – 49 discrete states&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<sup>a</sup>Adapted from code for (Barillas & Fernandez-Villaverde, 2007)

- Extends the EGM to non-convex problems including discrete state space
Outline

1 Introduction

2 Authors
   - Parametrized Expectations Algorithms (PEAs)
   - Miranda and Fackler
   - Endogenous Grid Method
   - Other Improved Grids
   - Common Components
     - Function Approximation
     - Approximation Metric
     - Evaluation Points
     - Expectations

3 Abstract Classes
## Parametrized Expectations Algorithms (PEAs)

<table>
<thead>
<tr>
<th>Component</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Approximation</td>
<td>Polynomial interpolation for subset of endogenous state variables.</td>
</tr>
<tr>
<td>Approximation Metric</td>
<td>Approximated Function Fixed Point at Nodes</td>
</tr>
<tr>
<td>Evaluation Points</td>
<td>Endogenous solution domain determined by extent of ergodic set</td>
</tr>
<tr>
<td>Expectations</td>
<td>Non product monomial and one point quadrature rules. Gaussian quadrature for accuracy tests.</td>
</tr>
</tbody>
</table>

- Cluster Grid approach chooses grid points endogenously based on the extent of ergodic set

(Judd et al., 2010)
### Component | Implementation
--- | ---
Function Approximation | Piecewise linear – Adaptive Simplicial Interpolation (ASI)
Approximation Metric | Approximated Function Fixed Point at Nodes
Evaluation Points | Uniform grid augmented with endogenously determined grids points at function kinks
Expectations | Discrete Markov Process

- Adaptive Simplicial Interpolation endogenously places grid points at “kinks” and uses Delaunay interpolation
(Fernández-Villaverde et al., 2012)

<table>
<thead>
<tr>
<th>Component</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function Approximation</td>
<td>Smolyak Projections and Interpolation; Complete polynomials</td>
</tr>
<tr>
<td>Approximation Metric</td>
<td>Selected Approximated Function Value at Node points</td>
</tr>
<tr>
<td>Evaluation Points</td>
<td>Time Iteration guess at collocation point. use them as t+1 functions to compute time t functions, repeat till no change</td>
</tr>
<tr>
<td>Expectations</td>
<td>Smolyak points</td>
</tr>
</tbody>
</table>

- One of a number of authors using Smolyak points along with complete polynomials
Outline

1. Introduction

2. Authors
   - Parametrized Expectations Algorithms (PEAs)
   - Miranda and Fackler
   - Endogenous Grid Method
   - Other Improved Grids
   - Common Components
     - Function Approximation
     - Approximation Metric
     - Evaluation Points
     - Expectations

3. Abstract Classes
Function Approximation Components

- Chebyshev Polynomials (Christiano & Fisher, 2000)
- Generic Polynomial interpolation (Judd et al., 2010)
- Finite element piecewise linear (Christiano & Fisher, 2000; Billi, 2011; Nakov, 2008; Fella, 2011; Barillas & Fernandez-Villaverde, 2007)
- Piecewise linear Adaptive Simplicial Interpolation (ASI) (Brumm & Grill, 2010)
- Smolyak Projections and Interpolation; Complete polynomials (Fernandez-Villaverde et al., 2012)
Approximation Metric

- Approximation Parameter Fixed Point Galerkin (Christiano & Fisher, 2000)
- Approximation Parameter Fixed Point Collocation (Fella, 2011)
- Approximated Function at Nodes Fixed Point; Collocation (Billi, 2011; Nakov, 2008; Barillas & Fernandez-Villaverde, 2007; Judd et al., 2010; Brumm & Grill, 2010)
- Selected Approximated Function Value at Node points (Fernández-Villaverde et al., 2012)
Evaluation Points

- Endogenous grid points (Carroll, 2006; Barillas & Fernandez-Villaverde, 2007; Krueger & Ludwig, 2007; Rendahl, 2006; Hintermaier & Koeniger, 2010; Fella, 2011)
- Smolyak points (Fernandez-Villaverde et al., 2012)
- Uniform grid (Billi, 2011; Nakov, 2008)
- Uniform grid augmented with endogenously determined grids points at function kinks (Brumm & Grill, 2010)
- Chebyshev Points (Christiano & Fisher, 2000)
- Cluster grid points determined by estimate of ergodic set (Judd et al., 2010)
Expectations

- **Discretized Tauchen Matrix Multiplication** (Fella, 2011; Barillas & Fernandez-Villaverde, 2007)
- **Gaussian Quadrature** (Christiano & Fisher, 2000; Billi, 2011; Nakov, 2008)
- **Non product monomial and one point quadrature rules** (Judd et al., 2010)
A General Framework

Following (Hintermaier & Koeniger, 2010)

First Order Conditions

\[ F(x_-, y_-, \lambda_-, x_0, y_0, \lambda_0, x_+, y_+, \lambda_+, b) = 0 \]

Equality Constraints

\[ Q(x_-, y_-, x_0, y_0, x_+, y_+, b) = 0 \]

Occasionally Binding Constraints

\[ O(x_-, y_-, x_0, y_0, x_+, y_+, b) \geq 0 \]

Complementary Slackness Conditions

\[ O(x_-, y_-, x_0, y_0, x_+, y_+, b) \land (x_-, y_-, x_0, y_0, \lambda_0, x_+, y_+, \lambda_+, b) = 0 \]

with

\[ \land (x_-, y_-, \lambda_-, x_0, y_0, \lambda_0, x_+, y_+, \lambda_+, b) \geq 0 \]
Object oriented implementation of the algorithms requires committing to collection of inter-operating classes. Existing code provides guidance for designing classes:
- Program data lead to fields
- Program operations lead to methods
Abstract Classes for Dynamic Models with OBCs

- Variable
  - StateVariable
  - NonStateVariable
  - LagrangeMultiplier
- Grid
- BooleanFunction
- ApproximateFunction
  - ValueFunction
  - PolicyFunction
- Expression

- Relation
  - Equation
  - Inequality
- System
  - EquationSystem
  - InequalitySystem
- Report
  - CPUUsageReport
  - MemoryUsageReport
Grid Abstract Class Tentative Example

- **Grid Fields**
  - evaluationPts
  - variableSpecs

- **Grid Methods**
  - `getEvaluationPts()` returns list of points – abstract
  - `evaluateAtPts(Function ff)` returns a list of points – abstract
  - `pointsWhereTrue(BooleanFunction qq)` returns a list of points – default implemented
  - `cpuUsageReport()` returns a CPUUsageReport – abstract
  - `memoryReport()` returns a MemoryReport – abstract
Outline

1. Introduction
2. Authors
3. Abstract Classes
4. Summary
5. Bibliography
Summary

- Two decades of solving dynamic models with occasionally binding constraints
  - Dozens of proposed solution algorithms
  - Broadly similar structure
  - Composed from a few common components
  - But generally incompatible implementations
  - Existing code and algorithms could provide guide to good API

- Synergy would be enhanced if components were more interchangeable
  - Experimentation in prototyping would be useful
  - Interoperability could facilitate experimentation and algorithm design
Adam, Klaus, & Billi, Roberto M. 2006.
Optimal monetary policy under commitment with a zero bound on nominal interest rates.
*Journal of money, credit and banking, 38*(7), 1877–1905.

Adam, Klaus, & Billi, Roberto M. 2007.
Discretionary monetary policy and the zero lower bound on nominal interest rates.
*Journal of monetary economics, 54*(3), 728–752.
Bibliography II

Comparing solution methods for dynamic equilibrium economies.
*Journal of economic dynamics and control, 30*(12), 2477–2508.

A generalization of the endogenous grid method.
*Journal of economic dynamics and control, 31*(8), 2698–2712.
Benigno, Gianluca, Chen, Huigang, Otrok, Christopher, Rebucci, Alessandro, & Young, Eric R. 2009 (April).
Optimal policy with occasionally binding credit constraints.

Billi, Roberto M. 2007 (April).
Optimal inflation for the u.s.

Billi, Roberto M. 2011.
Optimal inflation for the us economy.
Brumm, Johannes, & Grill, Michael. 2010 (July). 
Computing equilibria in dynamic models with occasionally binding constraints.

Carroll, Christopher D. 2006.
The method of endogenous gridpoints for solving dynamic stochastic optimization problems.
Economics letters, 91(3), 312–320.


Design patterns: Elements of reusable object-oriented software. 
Reading, MA: Addison-Wesley.

Gordon, Grey. 2011 (June). 
Computing dynamic heterogeneous-agent economies: Tracking the distribution. 


Ludwig, Alexander, & Schöhn, Matthias. 2012 (May). Endogenous grid methods in higher dimensions: Delaunay interpolation and hybrid methods. CMR, University of Cologne, Germany.
*Computational economics, 25*(3), 269–274.

*Economics letters, 87*(1), 135–140.
Solving the multi-country real business cycle model using a smolyak-collocation method.
*Journal of economic dynamics and control, 35*(2), 229–239.

Marcet, Albert, & Marimon, Ramon. 2011 (June).
*Recursive contracts.*
CEP Discussion Papers dp1055. Centre for Economic Performance, LSE.
