User Guide 01
Krusell-Smith Model

Felipe Alves

March 28, 2017

Abstract

These notes presents an step-by-step on Reiter's Projection+Perturbation approach to solve Heterogeneous Agents model with aggregate uncertainty. The method is discussed on the context of Krusell et al. (1998) model. The presentation draws heavily on Winberry (2016), although the implementation differs on some relevant aspects.

1. Model

Households There is a continuum of households indexed by $j \in [0,1]$, each with preferences over consumption c_{jt} represented by the utility function

$$\mathbb{E}\sum_{t=0}^{\infty}\beta^t\frac{c_{jt}^{1-\sigma}-1}{1-\sigma}$$

where β is the subjective discount factor and $\frac{1}{\sigma}$ is the elasticity of intertemporal substitution. Each household supplies inelastically ϵ_{jt} efficiency units of labor in the labor market at market price w_t. This idiosyncratic shock is distributed independently across households but for each one particular household it follows a two-state Markov process with values $\epsilon \in E := \{0,1\}$ and transition probabilities Π. Therefore households with idiosyncratic shock $\epsilon_{j,t}=0$ don't supply labor but receive unemployment benefits in the value of bw_t, while households with $\epsilon_{j,t}=1$ supply their whole unit of efficiency labor and enjoy after-tax labor earnings of $w_t(1-\tau)$. Let π denote the stationary distribution associated with Π. Assume that at period 0 ϵ_{j0} is distributed according to this stationary distribution. In particular, this imples that total labor endowments in the economy (in efficiency units) is constant every period and given by

$$\bar{L} = \pi(1)$$

Asset markets are incomplete. Agents have access only to one type of asset with rate of return r_t, potentially stochastic but independent of the individual state. Moreover, they asset holdings are subject to a borrowing constraint.

Therefore, the flow **budget constraint** at time t reads

$$\begin{aligned} c_{j,t} + a_{j,t+1} &= \Big(1+r_t\Big)a_t + w_t\Big(\epsilon_{j,t}(1-\tau)+(1-\epsilon_{j,t})b\Big) \\ a_{j,t+1} &\geq \underline{a} \end{aligned}$$

State Variables Before proceeding let's think what are the relevant states for the household problem and define some notation while we are at it. Idiosyncratic levels of asset holdings and employment shock (a,ϵ) must

be in the state, since they enter on the budget constraint. These are the *idiosyncratic states* of the agent. But note that this is not enough.

The price vector (r_t, w_t) also appears on the budget constraint and, in the recursive competitive equilibrium definition to be given below, must be themselves a function of the household's state variables. The assumptions on supply side make them a function of aggregate capital K_t and technological shock z in equilibrium. So why not have (K_t, z_t) themselves in the state?

It turns out that we need to keep track of more than just aggregate capital to compute household's policies. Note that when deciding how much to consume/save agents must also predict next period prices distribution - therefore next period capital - which in this heterogeneous agent model is not determined only by the aggregate level of present capital stock. Since agents with have different savings rates depending on their idiosyncratic states, predicting next period capital requires knowledge of the whole distribution of agents over *idiosyncratic states*. When we are solving only for a stationary equilibrium, the distribution over *idiosyncratic states* is invariant, making aggregate capital and hence prices constant. But in the presence of aggregate shocks, as is the case here, aggregate capital and hence the prices will vary which forces household to keep track of the whole distribution to be able to predict next period capital/prices. Therefore, the *aggregate states* of household problem are given by (z_t, μ_t) where μ_t denotes the time t measure of households across individual states.

Government Government finances unemployment insurance payments by taxing labor earnings with constant tax τ. Government's budget constraint is balanced each period implying

$$\tau \int \mu_t(da, 1) = b \int \mu_t(da, 0)$$

or simply, $\tau = \frac{b(1-L)}{L}$ where $L := \mu_t(1, A)$.

Firms There is a representative firm which produces output Y_t according to the production function

$$Y_t = e^{z_t} K_t^{\alpha} L_t^{1-\alpha}$$

where z_t is an aggregate productivity shock, K_t is the aggregate capital stock and L_t is aggregate labor.

In equilibrium, factor prices are determined competitively and given by the marginal products of inputs

$$\begin{aligned} r_t &= \alpha e^{z_t} K_t^{\alpha-1} L_t^{1-\alpha} - \delta \\ w_t &= (1-\alpha) e^{z_t} K_t^{\alpha} L_t^{-\alpha} \end{aligned}$$

where δ is the depreciation rate. Aggregate TFP follows an AR(1) process

$$z_{t+1} = \rho z_t + \sigma_z \omega_{t+1}, \qquad \omega_{t+1} \sim \mathcal{N}(0,1) \tag{1.1}$$

Equilibrium

A **Recursive Competitive Equilibrium** is a list of functions

$$\Big\{ c\big(a, \epsilon; z, \mu\big), a'\big(a, \epsilon; z, \mu\big), r\big(z, \mu\big), w\big(z, \mu\big), \Psi(\mu, z) \Big\}$$

1. (Household Optimization)
   Taking price functions $r(\cdot), w(\cdot)$ and the law of motion $\Psi(\cdot)$ as given, $\Big\{c(a,\epsilon;z,\mu), a'(a,\epsilon;z,\mu)\Big\}$ must satisfy

$$c(a,\epsilon;z,\mu)^{-\sigma} \geq \beta\mathbb{E}\Bigg\{\Big(1+\tilde{r}\Big(z',\Psi(z,\mu)\Big)\Big)c\Big(a'(a,\epsilon;z,\mu),\epsilon';z',\Psi(z,\mu)\Big)^{-\sigma}\Bigg\} \tag{1.2}$$

$$c(a,\epsilon;z,\mu) + a'(a,\epsilon;z,\mu) = \big(1+\tilde{r}(z,\mu)\big)a + w\big(z,\mu\big)\tau(\epsilon) \tag{1.3}$$

   with the euler holding with equality if $a'\big(a,\epsilon;z,\mu\big) > \underline{a}$.

2. (Firm Optimization)
   Given price function $r(\cdot), w(\cdot)$, firms choose capital K and labor L according to

$$r(z,\mu) = \alpha e^z K^{\alpha-1}L^{1-\alpha} - \delta \tag{1.4}$$

$$w(z,\mu) = (1-\alpha)e^z K^{\alpha}L^{-\alpha} \tag{1.5}$$

3. (Market Clearing)

   Labor market clears:

$$\bar{L} = \int \epsilon\mu\big(d[a\times\epsilon]\big) \tag{1.6}$$

   Asset market clears:

$$K' = \int a'\mu\big(d[a\times\epsilon]\big) \tag{1.7}$$

   Good market clear:

$$\int c(a,\epsilon;z,\mu)\mu\big(d[a\times\epsilon]\big) + \int a'(a,\epsilon;z,\mu)\mu\big(d[a\times\epsilon]\big) = \exp(z)F(K,L) + (1-\delta)K \tag{1.8}$$

4. Government budget constraint is satisfied

5. (Evolution of distribution)
   For any $\epsilon' \in E$ and measurable set $\mathcal{A} \subseteq A$, next period distribution implied by the law of motion Ψ is consistent with the households' policies

$$\Psi(z,\mu)\Big(\mathcal{A}\times\{\epsilon'\}\Big) = \int \mathbf{1}\Big\{a'(a,\epsilon;z,\mu)\in\mathcal{A}\Big\}\,\pi(\epsilon'|\epsilon)\,\mu\big(d[a\times\epsilon]\big) \tag{1.9}$$

2. Method

At a broad level, the solution method includes three steps

1. Approximate the infinite dimensional equilibrium objects, which in our example involves the policy functions of the household and cross-sectional distribution of individual states, by some finite dimensional object. This allows us to have a finite parametrization of the model that we shall refer to as the *discrete model*.
2. Compute the stationary equilibrium of the *discrete* model. This will take into account the idiosyncratic uncertainty but **not** the uncertainty coming from aggregate shocks.
3. Linearize the discrete model equations with respect to our finite parametrization around steady-state and compute the rational expectation solution of the linearized system taking into account aggregate shocks.

2.1. Finite-Dimensional Approximation

Looking at the equilibrium definition we see that there are two infinite-dimensional objects that need to be approximated if we intend to solve the model using Reiter (2009) method.

First, we need a finite-dimensional representation of the distribution over individual state variables (a, ϵ). The seems to be two main options here. One is to keep track of histogram defined on finite number of points over the state space. The other is to try to approximate the distribution by a density function of a finite dimensional family. Second, household's decision rules characterized by (1.2) are the solution to a functional equation and must also be approximated[1]

Distribution I The first option comes from Reiter (2009) and involves approximating the distribution of households by a finite number of mass points on a predefined grid for $\mathcal{A} := \{a_j\}_{j=1}^{N_a}$ and productivity E. Let $\Phi(a_j, \epsilon)$ denote the fraction of households with productivity level ϵ and asset holdings a_j. The evolution of this distribution is implied by the decision rules as follows. For any pair $(a_{j'}, \epsilon') \in \mathcal{A} \times E$, next period mass at the point is given by

$$\Phi'(a_{j'}, \epsilon') = \sum_{\epsilon \in E} \pi(\epsilon, \epsilon') \sum_{j=1}^{N_a} \omega_{j,\epsilon,j'} \times \Phi(a_j, \epsilon) \tag{2.1}$$

where $\omega_{j,\epsilon,j'}$ is a function of savings of the household

$$\omega_{j,\epsilon,j'} = \begin{cases} \frac{a' - a_{j'-1}}{a_{j'} - a_{j'-1}} & \text{if } a'(a_j, \epsilon; z, \mu) \in \left[a_{j'-1}, a_{j'}\right] \\ \frac{a_{j'+1} - a'}{a_{j'+1} - a_{j'}} & \text{if } a'(a_j, \epsilon; z, \mu) \in \left[a_{j'}, a_{j'+1}\right] \\ 0 & \text{o.w.} \end{cases}$$

Here's something I might need to add.

[1] Note that this is a separate concern from the fact that household decision rules in equilibrium depend on the whole distribution, which leads global solution methods unfeasible. The fact that policy solves a functional equation and needs to be approximated by a finite-dimensional object is true even when computing the stationary equilibrium

Distribution II The second alternative follows Winberry (2016) and tries to approximate the distribution over assets conditional on a realization of labor productivity ϵ - $\mu(da, \epsilon)$ - using the following parametric family

$$g_\epsilon\Big(a; g_\epsilon^0, \{g_\epsilon^i, m_\epsilon^i\}_{i=1}^{n_g}\Big) = g_\epsilon^0 \exp\left\{g_\epsilon^1(a - m_\epsilon^1) + \sum_{i=1}^{n_g} g_\epsilon^1\Big[(a - m_\epsilon^1)^i - m_\epsilon^i\Big]\right\} \tag{2.2}$$

where n_g denotes the order of approximation, $\{g_\epsilon^i\}_{i=0}^{n_g}$ are the parameters of the distribution which must be consistent with the centralized moments $\mathbf{m}_\epsilon := \{m_\epsilon^i\}_{i=1}^{n_g}$, that is,

$$\begin{aligned} m_\epsilon^1 &= \int a g_\epsilon(a) da \\ m_\epsilon^i &= \int \left(a - m_\epsilon^1\right)^i g_\epsilon(a) da \end{aligned} \tag{2.3}$$

Turns out that finding a solution of this non-linear system of equation is hard and without an good initial guess convergence is uncertain. To solve this issue we follow the suggestion by Algan et al. (2008) and find the coefficients - except for g_ϵ^0 - as the solution to the following minimization problem

$$\min_{\{\rho_\epsilon^i\}_{i=1}^{n_g}} \int g\Big(a; 1, \{\rho_\epsilon^i, m_\epsilon^i\}_{i=1}^{n_g}\Big) da$$

To see that, note that the first order conditions of the problem correspond exactly to consistency conditions (2.3). Turns out that solving this minimization problem is way easier than solving the system (2.3) and it works even without a good initial condition.

This approximation reduces the infinite-dimensional distribution μ to a set of moments $\{\mathbf{m}_\epsilon\}_{\epsilon \in E}$. But note that we still need to derive the law of motion in terms of this approximate aggregate state. Although current density together with decision rules pin down the distribution for next period, the distribution has no reason to be an element of the parametric family (2.2). That in principle presents us with a problem. But note that our reduced state only requires us to keep track of next period moments, from which we can derive next period approximated density through consistency conditions. Next period moments in turn are determined as

$$\begin{aligned} m_\epsilon^{1\prime} &= \sum_{\tilde{\epsilon} \in E} \frac{\pi(\tilde{\epsilon})\pi(\tilde{\epsilon}, \epsilon)}{\pi(\epsilon)} \int a'\Big(a, \tilde{\epsilon}; z, \mathbf{m}\Big) g_{\tilde{\epsilon}}(a) da \\ m_\epsilon^{i\prime} &= \sum_{\tilde{\epsilon} \in E} \frac{\pi(\tilde{\epsilon})\pi(\tilde{\epsilon}, \epsilon)}{\pi(\epsilon)} \int \Big(a'\Big(a, \tilde{\epsilon}; z, \mathbf{m}\Big) - m_\epsilon^{1\prime}\Big)^i g_{\tilde{\epsilon}}(a) da \end{aligned}$$

In practice, I compute the integrals using a Gauss-Legendre quadrature, which specifies nodes $\{a_j\}_{j=1}^{n_q}$ and weights $\{\omega_j\}_{j=1}^{n_q}$ and replaces the integrals with finite sums

$$\int a'(a, \tilde{\epsilon}; z, \mathbf{m}) g_{\tilde{\epsilon}}(a) da \approx \sum_{j=1}^{n_q} \omega_j a'(a_j, \tilde{\epsilon}; z, \mathbf{m}) g_{\tilde{\epsilon}}(a_j)$$

Household Decision Rules

Optimality conditions in terms of the new (finite-dimensional) aggregate state are given by

$$c\Big(a,\epsilon;z,\Phi\Big)^{-\sigma} \geq \beta\mathbb{E}_{z'|z}\left\{\big(1+\tilde{r}(z',\Phi')\big)\sum_{\epsilon'\in E}\pi(\epsilon'|\epsilon)c\Big(a'(a,\epsilon;z,\Phi),\epsilon';z',\Phi'\Big)^{-\sigma}\right\}$$
$$c\big(a,\epsilon;z,\Phi\big)+a'\big(a,\epsilon;z,\Phi\big)=\big(1+r(z,\Phi)\big)a+w(z,\Phi)\tau(\epsilon)$$

where conditional expectation of future marginal utility has been broken into two pieces: (i) the expectation with respect to idiosyncratic shocks is taken explicitly, (ii) expectation with respect to aggregate shocks is only implicitly in the expectation operator and will be taken into account only during the perturbation step.

Household's savings behavior is characterized by a critical level of assets χ_ϵ at which the borrowing constraint starts binding and a smooth function for $a > \chi_\epsilon$. I follow Reiter (2009) and approximate the savings rule by a piecewise linear spline with knot points $a_{i,\epsilon} = \chi_\epsilon + x_i,\ i = 1,\ldots,n_s$ with $0 = x_1 < \ldots < x_{n_s}$.

For a given aggregate state (z,Φ), household's savings policy is then represented by $n_\epsilon \times n_s$ coefficients giving for each idiosyncratic shock ϵ the critical level χ_ϵ and savings at $a_{2,\epsilon},\ldots,a_{n_s,\epsilon}$ - note that savings at $a_{1,\epsilon}$ equals $\underline{a}$ by construction.

These coefficients are collected into the vector $\boldsymbol{\theta}(z,\Phi)$ and the approximated consumption function is then written as $\hat{c}\big(a,\epsilon;\ \boldsymbol{\theta}(z,\Phi)\ \big)$. Given this approximation choice, household's optimality conditions are approximated using collocation, which forces (1.2)-(1.3) to hold exactly on the set of nodes $\big\{a_{i,\epsilon},\epsilon\big\}_{i\in\{1,\ldots,n_s\},\epsilon\in E}$

$$\hat{c}\Big(a_{i,\epsilon},\epsilon;\ \boldsymbol{\theta}(z,\Phi)\ \Big)^{-\sigma} = \beta\mathbb{E}_{z'|z}\left\{\big(1+\tilde{r}(z',\Phi')\big)\sum_{\epsilon'\in E}\pi(\epsilon'|\epsilon)\ \hat{c}\Big(\hat{a}'(a_{i,\epsilon},\epsilon;\ \boldsymbol{\theta}(z,\Phi)\ ),\epsilon';\ \boldsymbol{\theta}(z',\Phi')\ \Big)^{-\sigma}\right\}$$
$$\hat{c}\Big(a_{i,\epsilon},\epsilon;\ \boldsymbol{\theta}(z,\Phi)\ \Big)+\hat{a}'\Big(a_{i,\epsilon},\epsilon;\ \boldsymbol{\theta}(z,\Phi)\ \Big)=\big(1+\tilde{r}(z,\Phi)\big)a_{i,\epsilon}+w(z,\Phi)\tau(\epsilon)$$

The notation is intended to make clear how differently we treat the dependence of policy on idiosyncratic and aggregate states. Policy dependence on individual states is explicitly consider and parametrized by finite-dimensional vector $\boldsymbol{\theta}(z,\Phi)$, whose value at steady state $\boldsymbol{\theta}^*$ is determined by our collocation scheme. Policy depends on aggregates states only insofar as that parametrization varies with (z,Φ), and that dependence is solved at the perturbation step only.

> Mkcay and Reis: We approximate the policy rules for savings by piece-wise linear splines with 100 knot point each. We deal with the borrowing constraint in the approximation of the policy functions by parameterizing the point at which the borrowing constraint starts binding

Approximate Equilibrium Conditions

Substituting our approximations into our **recursive competitive equilibrium** definition we reduced the theoretical model with a continuum of agents and infinite dimensional state space to a discrete model with a high but finite-dimensional state vector. From now own, we concentrate on solving the latter. To ease notation, stack the equilibrium conditions of the *discrete model* in a function f as

$$\mathbb{E}_{z'|z}\Big[f(\mathbf{y}',\mathbf{y},\mathbf{x}',\mathbf{x},\omega')\Big]=0 \tag{2.4}$$

where $\mathbf{y} = (\boldsymbol{\theta}, r, w)$ of size $2 + \big(n_s \times n_\epsilon\big)$ is the $n_y \times 1$ vector of controls, $\mathbf{x} = (\Phi, K, z)$ is the $n_x \times 1$ vector of states. For example, in the case we choose to represent the distribution by a histogram (**Distribution I**), f is given by

$$
\begin{cases}
\Phi'(a_{j'}, \epsilon') - \displaystyle\sum_{\epsilon \in E} \pi(\epsilon'|\epsilon) \sum_{a_j \in \mathcal{A}} \Big(\omega_{j,\epsilon,j'} \Phi(a_j, \epsilon)\Big) \quad a_{j'} \in \mathcal{A}, \epsilon' \in E \\
K' \; - \displaystyle\sum_{\epsilon \in E} \sum_{j=1}^{N_a} a_j \Phi'(a_j, \epsilon) \\
z' - \rho_z z - \sigma_z \omega_z' \\
r - \alpha e^z K^{\alpha - 1} L^{1-\alpha} + \delta \\
w - (1-\alpha) e^z K^{\alpha} L^{-\alpha} \\
\hat{c}\Big(a_{i,\epsilon}, \epsilon;\ \boldsymbol{\theta}\ \Big)^{-\sigma} - \beta \Bigg\{ (1 + r') \displaystyle\sum_{\epsilon' \in E} \pi(\epsilon'|\epsilon)\ \hat{c}\Big(\hat{a}'(a_{i,\epsilon}, \epsilon; \boldsymbol{\theta}), \epsilon';\ \boldsymbol{\theta}'\ \Big)^{-\sigma} \Bigg\} \\
\hat{c}(a_{i,\epsilon}, \epsilon;\ \boldsymbol{\theta}\ ) + \hat{a}'(a_{i,\epsilon}, \epsilon; \boldsymbol{\theta}) - (1 + r) a_{i,\epsilon} - w\tau(\epsilon), \qquad i = 1, \ldots, n_s \text{ and } \epsilon \in E
\end{cases}
$$

A solution to model (2.4) is a function g that gives the controls in terms of states - $\mathbf{y}_t = g(x_t)$ - and a function h describing the law of motion of the states - $\mathbf{x}_{t+1} = h(\mathbf{x}_t) + \eta \omega_{t+1}$.

IMPORTANT: It is crucial to have the consistency condition in terms of next period assets. Problems with stable/unstable roots otherwise.

2.2. Stationary Equilibrium

In terms of the notation presented above, a *stationary equilibrium* of the *discrete* model are values $\mathbf{x}^*, \mathbf{y}^*$ such that $f(\mathbf{y}^*, \mathbf{y}^*, \mathbf{x}^*, \mathbf{x}^*, 0) = 0$. Note that this is just a nonlinear system with as many equations ans unknowns - $6 + \big((N_a + n_s) \times n_\epsilon\big)$. However, the system is too large to be solved with usual numerical algorithms and we need a more stable scheme to search for the equilibrium.

The following algorithm seems to work in practice. Set up an initial guess for capital K^0

1. Back out market-clearing factor prices $r^0 = \alpha K^{\alpha-1} L^{1-\alpha} - \delta$ and $w^0 = (1-\alpha) K^{\alpha} L^{-\alpha}$
2. Given fixed prices $\big(r^0, w^0\big)$, we can solve the dynamic problem of the agent for $\boldsymbol{\theta}_0^*$
3. Using implied decisions, solve for *invariant distribution*, i.e.

$$
\Phi^0(a_{j'}, \epsilon') - \sum_{\epsilon \in E} \pi(\epsilon, \epsilon') \sum_{a_j \in \mathcal{A}} \Big(\omega_{j,\epsilon,j'}\big(\boldsymbol{\theta}_0^*\big) \times \Phi^0(a_j, \epsilon)\Big)
$$

4. Compute the aggregate supply of capital deduced the invariant distribution

$$A^0 = \sum_{\epsilon \in E} \sum_{j=1}^{N_a} a_j \ \Phi^0(a_j, \epsilon)$$

5. Check $|A^0 - K^0| < \epsilon$. If not update the guess for capital K^1.[2]

2.3. Linearization using Automatic Differentiation

As is stated on *wiki*

> *Automatic differentiation (AD) is a set of techniques to numerically evaluate the derivative of a function specified by a computer program. AD exploits the fact that every computer program, no matter how complicated, executes a sequence of elementary arithmetic operations (addition, subtraction, multiplication, division, etc.) and elementary functions (exp, log, sin, cos, etc.). By applying the* ***chain rule*** *repeatedly to these operations, derivatives of arbitrary order can be computed automatically, accurately to working precision, and using at most a small constant factor more arithmetic operations than the original program.*

The key idea of AD in to use basic derivative rules from calculus, such as the chain rule, in a numerical environment. The derivatives are then computed together with the evaluation steps and combined with other derivatives using these rules.

As an example, consider the function f defined by $x \to x \sin x^2$. When evaluating this function at $x = 3$ you computer will compute the sequence of values on the left of table

$x = 3$	$\dot{x} = 1$
$w_1 = x^2$	$\dot{w}_1 = 2x\dot{x}$
$w_2 = \sin(w_1)$	$\dot{w}_2 = \cos(w_1)\dot{w}_1$
$w_3 = xw_2$	$\dot{x}w_2 + x\dot{w}_2$

But for each computation in the left, we can compute the derivatives that aggregate up to the derivative of the function at last line. What AD does is make derivative computations to happen automatically when evaluating or interpreting the values on the left side.

These ideas are implemented in Matlab using object-oriented programming (OOP) features to define a new class of value-and-derivative objects. Methods are then associated to such objects in order to implement standard calculus derivative rules with built-in chain rule. In particular, methods overload the definitions of standard operations and functions, such as $*$ and sin, to compute and return not only the value of the expression but also its derivative.

[2] The updating rule may come from a bisection or newton algorithm.

To see a simple example, to compute the value and derivative of our function f at 3, we create a value-and-derivative object `x` with attributes $v = 3$ (value) and $d = 1$ (derivative). When we evaluate the expression `x * sin( x^2 )` both the left and right hand side of the table are executed and the return value is also a value-and-derivative object with attributes $v = 1.2363$ and $d = 15.9883$.

Julia Implementation

Using AD is made easy on *Julia* environment due to the ForwardDiff package which offers an implementation of forward mode automatic differentiation (AD) in Julia. The package is built upon the `Dual` type

```
type DualNumber{N, T<:Real} <: Real
value::T
partials::Partials{N,t}
end
```

where `Partials` is defined

```
type Partials{N,T}
values::NTuple{N,T}
end
```

The idea here is to enhance any real number x by an additional term $x'\epsilon$, where $x' \in \mathbb{R}$ and ϵ is an *infinitesimal* with the property that $\epsilon^2 = 0$. All arithmetic operators must then be extended for the augmented algebra. This new arithmetic will consist of ordered pairs, elements written (x, x'), with ordinary arithmetics on the first component, and first order differentiation arithmetic on the second component.

Defining a new arithmetic to our `Dual` types is natural in Julia because of its *multiple dispatch* approach. We just need to add a new method to deal with `Dual` numbers on all elementary numerical functions that evaluates both the original function and the derivative of the function. For example Base.sin should be handled as

```
Base.sin(d::Dual) = Dual(sin(value(d)), cos(value(d)) * partials(d))
```

ForwardDiff defines the `Dual` type and does all the overloading so that we don't need to worry about it when computing derivatives. Since our function f, no matter how complicated, ends up being composed entirely of these elementary functions, then the chain rule enables our derivatives to compose as well. Thus we can differentiate f by passing in a Dual number and looking at the output.

ADD simple example on the jupyter ...

2.4. Solving Linearized System

Klein method. Linearizing (2.4) around the steady-state yields a first-order linear expectational difference equation system of the form

$$f_{\mathbf{y}'} E_t\Big[\underbrace{(\mathbf{y}_{t+1} - \bar{\mathbf{y}})}_{\tilde{\mathbf{y}}_{t+1}}\Big] + f_{\mathbf{y}}\tilde{\mathbf{y}}_t + f_{\mathbf{x}'} E_t\Big[\tilde{\mathbf{x}}_{t+1}\Big] + f_{\mathbf{x}}\tilde{\mathbf{x}}_t = 0 \tag{2.5}$$

which can be set on Klein's form letting $\Gamma_0 := \begin{bmatrix} f_{\mathbf{x}'} & f_{\mathbf{y}'} \end{bmatrix}$ and $\Gamma_1 := -\begin{bmatrix} f_{\mathbf{x}} & f_{\mathbf{y}} \end{bmatrix}$ so that

$$\Gamma_0 E_t \left\{ \begin{bmatrix} \mathbf{x}_{t+1} \\ \mathbf{y}_{t+1} \end{bmatrix} \right\} = \Gamma_1 \begin{bmatrix} \mathbf{x}_t \\ \mathbf{y}_t \end{bmatrix} \tag{2.6}$$

SIMS METHOD. Let $X_t := (\tilde{\mathbf{x}};\ \tilde{\mathbf{y}})$. To put the model into Sims' format, we need to get rid of the expectational terms in

$$\Gamma_0 \mathbb{E}_t \Big[X_{t+1} \Big] = \Gamma_1 X_t$$

where Γ_0, Γ_1 are defined as before. To do so we substitute $E_t\big\{X_{t+1}\big\}$ by the combination of its *ex post* realizations plus the appropriate forecast errors.[3] Once we do that, we get to Sims' canonical form

$$\Gamma_0 X_{t+1} = \Gamma_1 X_t + \Psi \omega_{t+1} + \Pi \eta_{t+1} \tag{2.7}$$

which is on Sims' canonical form and can be solved using Sim's `gensys` algorithm.[4].

[3] For each $y_{i,t+1}$, whenever we have $E_t\{y_{i,t+1}\}$ we substitute to $y_{i,t+1} - \eta_{i,t+1}$, where $\eta_{i,t+1}$ are **endogenous** expectational errors determined as part of the solution. Endogenous states are all pre-determined, so $E_t\{x_{1,t+1}\} = x_{t+1}$, while exogenous states have exogenous forecast error ω_{t+1}, so we can write $E_t\{\mathbf{x}_{2,t+1}\} = \mathbf{x}_{2,t+1} - \omega_{t+1}$

[4] I actually use a modified version of the gensys algorithm that treats the existence and uniqueness more carefully. The code accompanies the `Reiter.jl`.

References

Yann Algan, Olivier Allais, and Wouter J. Den Haan. Solving heterogeneous-agent models with parameterized cross-sectional distributions. *Journal of Economic Dynamics and Control*, 32(3):875–908, March 2008. URL https://ideas.repec.org/a/eee/dyncon/v32y2008i3p875-908.html.

Per Krusell, Anthony A. Smith, and Jr. Income and Wealth Heterogeneity in the Macroeconomy. *Journal of Political Economy*, 106(5):867–896, October 1998. URL https://ideas.repec.org/a/ucp/jpolec/v106y1998i5p867-896.html.

Michael Reiter. Solving heterogeneous-agent models by projection and perturbation. *Journal of Economic Dynamics and Control*, 33(3):649–665, March 2009.

Thomas Winberry. A toolbox for solving and estimating heterogeneous agent macro models. 2016.