Finn Marten Körner

Wechselkurse und globale Ungleichgewichte

Wirtschaftsentwicklung und Stabilität in Bretton Woods I und II

12. November 2013

Springer

Inhaltsverzeichnis

Akronyme und Abkürzungen

BIS	Bank für Internationalen Zahlungsausgleich
FRED2	Datenbank der Federal Reserve Bank of St. Louis
FDI	ausländische Direktinvestitionen
IMF	*International Monetary Fund*
NBS	*National Bureau of Statistics of China*
NIPA	*National Income and Product Accounts*
OECD	*Organization for Economic Cooperation and Development*
PBC	*People's Bank of China*
SAFE	*State Administration on Foreign Exchange*
TFP	totale Faktorproduktivität, *total factor productivity*
TOT	*terms of trade*
US	US-amerikanisch
XR	Wechselkurs, *exchange rate*

Kapitel 1
OnlinePlus

1.1 Mathematischer Anhang

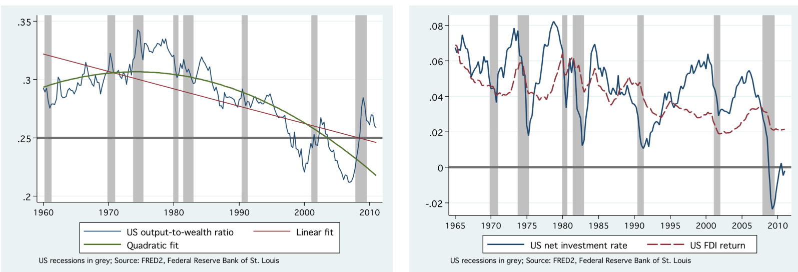
1.1.1 Updated Calibration

The model simulation uses calibrated starting values analogous to Caballero, Farhi und Gourinchas (2008) which date back to 2004. Some of the value need to be reviewed and updated, however,

Parameter	θ	g	δ	x_0^R	μ_{0-}^{RU}	NA_{0-}^U	σ	γ	g^z	g^n
Value	0.25	0.03	0.24	0.30	0.05	0.0	4	0.9	0.0	0.03
Parameter	κ	κ_P	r_{aut}							
Value	0.04	0.08	0.06							

Tabelle 1.1 Updated calibrated and starting values for exogenous model parameters.

The parameter θ is approximated by GDP over the net worth of the household sector according to the US Flow of Funds. From Figure 1.1(a) it is apparent that the dot-com bubble and the housing market bubble both significantly lowered θ compared with the rather stable higher values observed before 1995. Assuming a value of $\theta = .25$ seems warranted as the mean over the volatile past fifteen or so years. It is depicted in Figure 1.1(a).



(a) Evolution of the parameter θ (1960–2010).

(b) Evolution of investment (1965–2010).

Abb. 1.1 Calibrating new values for θ, κ and κ_P.

Gross domestic investment is defined by the US *National Income and Product Accounts* (NIPA) guide as consisting of “fixed investment and the change in private inventories. Fixed investment consists of both non-residential fixed investment and residential fixed investment. It is measured without a deduction for CSC and includes replacements and additions to the capital stock. [...] It excludes investment by US residents in other countries."(BEA, 2006, 7) Since the definition of investment in Caballero, Farhi und Gourinchas (2008, 377) reads that "[p]lanting the $g^n N_t^i$ new trees consumes resources $I_t^i = \kappa q_t^i X_t^i$", we need net investment, not gross investment. Duca (1997, 5) proposes to deduct NIPA's "consumption of fixed capital" in order to get the net investment rate of Figure 1.1(b). Being extremely volatile in nature, the parameter κ was around 5.2% before 1990, and thereafter exactly 4% until 2007 when it contracted dragging the 20-year average to 3.3%. It seems warranted to assume a long-term net investment rate $\kappa = .04$.

We can get a benchmark for κ_P as the bargaining price for carrying out ausländische Direktinvestitionen (FDI) by looking at the net capital outflows and the resulting stock of US assets abroad. Approximating investment income by "Income Receipts on US Assets Abroad" and "Other Private Income Receipts on US Assets Abroad" we can calculate the return on investment for FDI as shown in Figure 1.1(b). Since 1965 it has slowly decreased to a mean just below 4%. In the model's language, this is the bargaining power of US investors since they derive positive income from abroad. κ_P is therefore 4 percentage points below the value which would render all FDI gains to the foreign country. κ_P is therefore not 12% but $.12 - .04 = .08$.

1.1.2 Nested model properties

All model extensions are based on the same underlying model specification. It is therefore possible to construct a model set-up which incorporates all sub-models by separatating the general equilibrium procedure into four cases: (i) a baseline case without FDI and exchange rates, (ii) the investment case with FDI, (iii) the multiple goods case with exchange rates but no FDI and (iv) a joint model with both FDI and exchange rates. A separation can be achieved by specifying non-adjustable terms of trade $q_t^R = const.$ for $\forall t$. An exchange rate is thus not excluded but the mechanism is disabled as a rebalancing channel and the trade balance and the real interest rate – in (i) – and the interest rate and returns on FDI as in (ii) need to allow for rebalancing. Disabling the FDI component is somewhat more complex since a larger set of equations is affected. In particular, the need to solve for asset values by reverse calculation as shown in appendix 1.1.5 becomes unnecessary. The joint model in (iv) is employed as specified in section 6.

The figure one the baseline model in the results section is the graphical exposition of the property that all sub-models are nested within the baseline model. The equations underlying the six panels are given here for comparison:

Current account $\quad CA_t^U = TB_t^U + r_t(\alpha_t^{UR} V_t^R - \alpha_t^{RU} V_t^U) \equiv TB_t^U + r_t NA_t^U.$

Net foreign assets $NA_t^U = \alpha_t^{UR} V_t^R - \alpha_t^{RU} V_t^U \equiv W_t^U - V_t^U$

$$\text{Baseline:}\quad NA_t^U/X_t^U = \frac{(1-\delta)}{(\theta+g-r_t)} - \frac{\delta}{(r_t-g)}$$

$$\text{Pure FDI:}\quad NA_t^U/X_t^U = \frac{(1-\delta-\kappa) + g^n \frac{\delta+\delta x_t^R/x_t^U}{(r_t-g^z)} - (\kappa+\kappa_P) x_t^R/x_t^U}{(\theta+g-r_t)} - \frac{\delta}{(r_t-g)}$$

$$\text{Pure XR:}\quad NA_t^U/X_t^U = \frac{(1-\delta)}{(\theta+g-r_t)} - \frac{\delta}{(r_t-g)}$$

$$\text{Joint model:}\quad NA_t^U/X_t^U = \frac{(1-\delta-\kappa) + g^n \frac{\delta+\delta x_t^R/x_t^U}{(r_t-g^z)} - (\kappa+\kappa_P) x_t^R/x_t^U}{(\theta+g-r_t)} - \frac{\delta}{(r_t-g)}$$

Trade balance $TB_t^U = X_t^U - I_t^U - \theta W_t^U$.

$$\text{Baseline:}\quad TB_t^U/X_t^U = 1 - \theta \frac{(1-\delta)}{(\theta+g-r_t)}$$

$$\text{Pure FDI:}\quad TB_t^U/X_t^U = (1-\kappa) - \theta \frac{(1-\delta-\kappa) + g^n \frac{\delta+\delta x_t^R/x_t^U}{(r_t-g^z)} - (\kappa+\kappa_P) x_t^R/x_t^U}{(\theta+g-r_t)}$$

$$\text{Pure XR:}\quad TB_t^U/X_t^U = 1 - \theta \frac{(1-\delta)}{(\theta+g-r_t)}$$

$$\text{Joint model:}\quad TB_t^U/X_t^U = (1-\kappa) - \theta \frac{(1-\delta-\kappa) + g^n \frac{\delta+\delta x_t^R/x_t^U}{(r_t-g^z)} - (\kappa+\kappa_P) x_t^R/x_t^U}{(\theta+g-r_t)}$$

Real interest rate with $\hat{v}_t^{Ro} q_t^R X_t^{Ro} = V_t^{Ro}$

$$\text{Baseline:}\quad r_t = g^z + \theta\left[\delta - (\delta-\delta^R) x_t^R)\right]$$

$$\text{Pure FDI:}\quad r_t = g^z + \frac{\theta}{1-\kappa}\left(\left[\delta - (\delta-\delta^R) x_t^R\right] - g^n \hat{v}_t^{Ro} x_t^R \left[\frac{\delta}{\delta^R} - 1\right]\right)$$

$$\text{Pure XR:}\quad r_t = g^z + \theta\left[\delta - (\delta-\delta^R) x_t^R)\right] + x_t^R \frac{\dot{q}_t^R}{q_t^R}$$

$$\text{Joint model:}\quad r_t = g^z + \frac{\theta}{1-\kappa}\left(\left[\delta - (\delta-\delta^R) x_t^R\right] - g^n \hat{v}_t^{Ro} x_t^R \left[\frac{\delta}{\delta^R} - 1\right]\right) + x_t^R \frac{\dot{q}_t^R}{q_t^R}$$

Real exchange rate $1 = \theta\gamma w_t \left(\gamma + (1-\gamma) q_t^{R(1-\sigma)}\right)^{-1} + (1-\gamma)\left(\frac{(1-\kappa)}{x_t} - \theta w_t\right)\left(\gamma q_t^{R(1-\sigma)} + (1-\gamma)\right)^{-1}$

- Baseline: $q_t^R = const.$, adjustment via r_t and V_{0+}^U.
- Pure FDI: $q_t^R = const.$, adjustment via r_t, FDI and V_{0+}^U.
- Pure XR: $q_t^R = flex.$, adjustment via r_t, q_t^R and V_{0+}^U.
- Joint model: $q_t^R = flex.$, adjustment via r_t, q_t^R, FDI and V_{0+}^U.

Global portfolio share U's assets in R's portfolio holdings as a share of R's overall wealth W_t^R:

$$\mu_t^{RU} = \alpha_t^{RU} \frac{V_t^U}{W_t^R} = \frac{\sum_{s=0}^{t-1} CA_s^R}{V_t^U} \frac{V_t^U}{W_t^R} = \frac{\sum_{s=0}^{t-1} CA_s^R}{W_t^R} \equiv \frac{\sum_{s=0}^{t-1} \dot{V}_s^U - \dot{W}_s^U}{W_t^R}$$

1.1.3 Simulation sequence

The source code of the calibrated simulation of the joint model is rather extensive. It is available from the author upon request. A simplified version of the model of a wholly descriptive nature is given in what follows:

Pre-shock $(t = 0^-)$

- Starting values and definitions:
  1. Define calibrated parameters as in 1.1.4.
  2. Derive pre-shock values for: $X_{0^-}^U$ and r_{aut}.
- Iterative solution to $q_{0^-}^R$:
  1. Make initial guess for shock to $V_{0^+}^R$, e.g. a drop by 25%.
  2. Use guess in initial portfolio allocation to derive wealth: $W_{0^+}^i = (1 - \alpha_{0^-}^{ji})V_{0^+}^i + \alpha_{0^-}^{ij} V_{0^+}^j$.
  3. Use shooting algorithm for relative demand equation to solve for (known) $X_{0^-}^U$.
  4. Thereby derive $q_{0^-}^R$, $X_{0^-}^R$ and $W_{0^-}^R$ using 1.1.6.
- Define pre-shock values for all remaining variables requiring terms of trade $q_{0^-}^R$.

Shock $(t = 0^+)$

- Start of main loop from $0^+ \to \infty$ (approximated at $N = 200$ periods/years):
- Solve system $\{w_t, x_t, q_t^R\}$ for $t = 0^+$ using equations in 1.1.6:
  - Use post-shock wealth share $w_{0^+} = W_{0^+}^U / X_0^U$ from portfolio allocation estimate.
  - Calculate values for $q_{0^+}^R$ and $x_{0^+}^U$ iteratively until (1.10) converges to 1.
- Solving r_t and $\hat{v}_t^{Ro}$ reversely for $t = \infty \to 0^+$ using the adjusted FDI model in 1.1.5.

Post-shock $(0^+ < t \leq \infty)$

- Solve system $\{w_t, x_t, q_t^R\}$ for $t = 1$ using 1.1.6:
  - Calculate value for q_1^R and x_1^U iteratively until (1.10) converges to 1.

  – Calculate post-shock rate of change of terms of trade as $\dot{q}_{0^+}/q_{0^+} = (q_1^R - q_{0+}^R)/q_{0+}^R$.
  – Initial appreciation at $t = 0^+$ feeds into 'pre-shock' model since $(q_{0+}^R - q_{0-}^R)/q_{0-}^R$ only in $t = 0^-$.

  $\Rightarrow$ No 'over-shooting' of the interest rate at impact because the initial appreciation of q_{0+}^R does not affect post-shock dynamics.

- Repeat system for $t = 2 \rightarrow \infty$ and calculate all other values for remaining variables using one-step iterations.
- End first run-through.

  Repeated loop characteristics
- Repeat loop for $(0^+ < t \leq \infty)$ with updated values:
  – Update estimate for V_{0+}^R using 1.1.5.
  – Update estimate for δ^R to produce desired shock: $\delta^R = \delta^R \times (1 - shock) \times V_{0-}^R/V_{0+}^R$
- End loop if there is no further change in the guesses for V_{0+}^U and V_{0+}^R . The system converges.

1.1.4 Initial values before the shock

Initial values for all variables can be created using the equilibrium conditions before the asset market shock to δ^R in $t = 0^+$. The pre-shock period is labelled $t = 0^-$. If a variable is just denoted $t = 0$ then this particular variable is not affected by the shock and remains at the pre-shock value. The whole model is scalable so that absolute values are irrelevant. Only values relative to each region's or total output need to be considered. We can therefore make the following assumptions for the variables describing the real economy $X_t^i = N_t^i Z_t^i$ for all regions $i = \{U, Ro, Rn, R\}$:

$$\begin{aligned} \text{Output:} \quad & X_0^i = x_{0-}^i \\ \text{Number of trees:} \quad & N_0^i = x_{0-}^i \\ \text{Productivity:} \quad & Z_0^i = X_0^i/N_0^i = 1 \end{aligned}$$

The calculation of wealth and asset values requires an initial value for the terms of trade q_{0-}^R. The shooting algorithm of the system $\{x_t, w_t, q_t\}$ starts with an initial guess for q_{0-}^R and thus relative output $x_{0^-} = X_{0-}^U/(\sum_i q_{0-}^i X_{0-}^i)$ and wealth $w_{0^-} = W_{0-}^U/X_{0-}^U$ in U. The algorithm is repeated until pasting the previously obtained shooting value for $\tilde{q}_{0-}^R$ delivers the exogenously determined value of X_{0-}^U:

$$\text{Relative demand guess:} \quad \tilde{X}_t^U = \frac{\theta\gamma\tilde{W}_t^U}{\left(\gamma+(1-\gamma)\tilde{q}_t^{R(1-\sigma)}\right)} + \frac{\theta(1-\gamma)\tilde{W}_t^R}{\left(\gamma\tilde{q}_t^{R(1-\sigma)}+(1-\gamma)\right)}$$

$$\text{Updated terms of trade guess:} \quad \tilde{q}_t^R = (X_t^U/\tilde{X}_t^U)q_t^R$$

$$\text{Updated wealth guess:} \quad \tilde{W}_t^R = \frac{(1-\delta-\kappa)\tilde{q}_t^R\tilde{X}_t^U}{(\theta-g^z-r_{aut})}$$

The shooting value for $\tilde{q}_t^R$ is adjusted so that larger deviations from the target value for X_t^U produce a greater change in the guessed value. The change in the terms of trade are defined as the future period's shooting value minus its current value: $\dot{q}_t^R = q_{t+1}^R - q_t^R$.[1] After obtaining the shooting value for the terms of trade we can then determine initial asset values, wealth and output relations as well as price indices in the respective regions and the real exchange rate between regions i and j:

$$\text{Wealth:} \quad \frac{W_{0^-}^i}{X_0^i} = \frac{1-\delta-\kappa}{\theta-g^z-r_{aut}}$$

$$\text{Asset values:} \quad \frac{V_{0^-}^i}{X_0^i} = \frac{\delta}{r_{aut}-g^z}$$

$$\text{Price indices:} \quad P_{0^-}^i = \left(\gamma q_{0^-}^{i(1-\sigma)} + (1-\gamma)q_{0^-}^{j(1-\sigma)}\right)^{\frac{1}{1-\sigma}}$$

$$\text{Real exchange rate:} \quad \lambda_{0^-}^{ij} = P_{0^-}^j / P_{0^-}^i$$

In contrast to the pure FDI model, investment in the joint model needs to be adjusted for changes in the terms of trade, too. The bargaining price for executing investment options I_t^{Rn} is susceptible to exchange rate changes according to:

$$\text{Investment:} \quad I_{0^-}^i = \kappa q_{0^-}^i X_0^i$$

$$\text{Price of FDI:} \quad P_{0^-} = \kappa_P q_{0^-}^i X_0^i$$

All other variables do not require a particular adjustment to their initial values since their definitions depend only on the above derived parameters. Valuation effects in asset holdings in the α_0^{ij} and $\mu_{0^-}^{ij}$ parameters are already accounted for in the asset value and wealth equations.

[1] The exact definition of $\dot{q}_t^i$ is left unspecified by the authors in the final 2008 and all previous working paper versions 2006. None of the original authors replied to requests by the present author to validate the above assumption.

1.1.5 Derivation of post-shock asset values

In the joint model with FDI and exchange rates, there are three ways of calculating the initial asset values after the shock. All three should ideally give the same result. Due to the iterative nature of the simulation model, small deviations are likely. The derivation of asset values is not as straightforward as in the single model extensions. In the FDI model extension we determined the initial asset value in $V^R_{0^+}$ by backward integration using the property that the relative importance of old assets dimishes with time due to non-investment. Using asymptotic asset values for individual assets – $v^{Ro}_t / v^{Rn}_t = \delta^R/\delta$ – one could show that $\hat{v}^{Ro}_t = V^{Ro}_t / q^R_t X^{Ro}_t$ would converge to:

$$\hat{v}^{Ro}_\infty = \frac{\delta^R}{\delta}\frac{1}{\theta} \tag{1.1}$$

The ratio between old and new asset values, $v^{Ro}_t / v^{Rn}_t = \delta^R/\delta$, is constant over time so that diverging aggregate asset values for V^{Ro}_t and V^{Rn}_t are only driven by investment into the number of assets they are comprised of, $V^{Ro}_t = N^R_0 v^{Ro}_t$ and $V^{Rn}_t = (N^R_t - N^R_0) v^{Rn}_t$, respectively. The latter assumes that all investment after the shock to δ^R is undertaken with the aid of U (labelled 'know-how export'). This extreme assumption ensures an upper bound to the effect of FDI in the present model and is maintained for illustrative purposes alone. A partial substitution with effective investment resulting in a mixture of δ^R and δ assets in R is likely in reality but needs additional assumptions for calibrated starting values. The dynamics between this asymptotic value and the time of the shock at $t = 0^+$ are given by:

$$\frac{d\hat{v}^{Ro}_t}{dt} = \frac{\theta}{1-\kappa}\left[\delta(1-x^{Ro}_t) + \delta^R x^{Ro}_t - g^n \hat{v}^{Ro}_t x^{Ro}_t (\delta/\delta^R - 1)\right]\hat{v}^{Ro}_t - \delta^R \tag{1.2}$$

and may be re-written using the interest rate equation derived in (1.11) as:

$$\frac{d\hat{v}^{Ro}_t}{dt} = (r_t - g^z)\hat{v}^{Ro}_t - \delta^R \tag{1.3}$$

The main change is for the reversely solved FDI model to be re-written using exchange rates. In the joint model, the equilibrium interest rate not only depends upon the current value of V^{Ro}_t / X^{Ro}_t which itself depends upon the entire sequence of future interest rates Caballero et al. (2006, 50). Now, the solution is additionally complicated by incorporating the rate of change of the terms of trade, $\dot{q}^R_t / q^R_t$, for the entire sequence.

Following Caballero, Farhi und Gourinchas (2008, 391), the derivation starts at $t = \infty$ with $x^{Ro}_\infty \approx 0$ and $\hat{v}^{Ro}_t = \hat{v}^{Ro}_\infty$. The value of V^{Ro}_t decreases at the nominal rate g^n due to the non-investment in old assets – $X_t^{Ro} = N_0^R Z_t^R$ for $\forall t$ – so that $x_t^{Ro} = q_t^R N_0^R Z_t^R / X_t$ evolves taking logs-and-derivatives of $x_t^{Ro} = q_t^R X_t^{Ro} / \sum q_t^i X_t^i$ as:

$$\begin{aligned} \dot{x}_t^{Ro}/x_t^{Ro} &= \dot{q}_t^R/q_t^R + g^z - (g x_t^U + g^R x_t^R + x_t^R \dot{q}_t^R/q_t^R) \\ \dot{x}_t^{Ro} &= \left[-g^n + (1-x_t^R)\dot{q}_t^R/q_t^R\right] x_t^{Ro} \\ \frac{\partial x_t^{Ro}}{\partial q_t^R} &= (1-x_t^R)\frac{X_t^{Ro}}{q_t^R X_t^R + X_t^U} = (1-x_t^R)\frac{x_t^{Ro}}{q_t^R} \end{aligned} \tag{1.4}$$

The system is backwards soluble until $x_t^{Ro} = x_{0^+}^{Ro}$ which is known once we have determined the post-shock real exchange rate. The post-shock output share is then calculated as $x_{0^+}^{Ro} = q_{0^+}^R X_0^{Ro} / X_{0^+}$. It is therefore necessary to employ the shooting algorithm in 1.1.6 to solve for the real exchange rate and use the derived value for $q_{0^+}^R$ in the above model until it converges. Convergence of the model is assumed once a changes in the real exchange rate do not affect post-shock asset values and *vice versa*.

The necessary condition for convergence is for the FDI-driven process to produce the same post-shock asset value $V_{0^+}^R$ as the one using the present value derivation from the exchange rate model. For this, we need two conditions to be met: $V_{0^+}^R$ needs to be equal in both cases and δ^R needs to be calibrated so as to produce the desired shock at $t = 0^+$, in our case $\Delta V_{0^+}^R / V_{0^-}^R = -25\%$.

By definition: $$V_{0^+}^R = N_0^R v_{0^+}^{Ro} = N_0^R \delta^R q_{0^+}^R Z_{0^+}^R \int_{0^+}^{\infty} e^{-\int_{0^+}^{s}(r_u - g^z)du} ds \tag{1.5}$$

From the XR model: $$\begin{aligned} V_{0^+}^R &= V_{0^+}^{Ro} + V_{0^+}^{Rn} \\ &= \delta^R \int_0^{\infty} q_t^R X_t^{Ro} e^{-\int_0^s r_u du} ds + \delta \int_0^{\infty} q_t^R X_t^{Rn} e^{-\int_0^s r_u du} ds \end{aligned} \tag{1.6}$$

From the FDI model: $$V_{0^+}^R = \hat{v}_{0^+}^{Ro} q_{0^+}^R X_{0^+}^{Ro} \tag{1.7}$$

All three approaches to evaluate $V_{0^+}^{Ro}$ should therefore yield the same result. The financial market parameter δ^R is the factor which brings this equality about since it is iteratively adjusted until $V_{0^+}^{Ro} = (1 - shock) \times V_{0^-}^{Ro}$.

1.1.6 Post-shock exchange rate dynamics

The shock in $t = 0^+$ needs to be calibrated so as to produce a decline in asset values of 25% in response to a reduction in δ^R. The simulations of the FDI and exchange rate extensions each separately posed the additional difficulty of having to solve for initial asset values $V_{0^+}^i$ using backward and forward integration respectively. In the joint model, we still know that in $x_\infty^{Ro} = 0$ and additionally $\dot{q}_\infty^R / q_\infty^R = 0$ but we cannot postulate a sequence for $\dot{x}_t^{Ro}$ as in the separate model due to incomplete knowledge of the future path of the terms of trade affecting relative output. Similarly, we cannot find the post-shock value of $V_{0^+}^R$ for lack of knowledge of the evolution of relative output and the rate of interest. We therefore have to solve the model – which is nonetheless uniquely identified and only lacks starting values – by using a shooting algorithm.

The model is iteratively solved back and forth until a coherent, i.e. non divergent, path for r_t, q_t^R and V_t^i is found. These paths include the solution to the system (w_t, x_t, q_t^i) from the exchange rate extension. The system includes the shorthands for $w_t = W_t^U / X_t^U$, $x_t = X_t^U / \sum q_t^i X_t^i$ and $\gamma = \gamma_{UU} = 1 - \gamma_{UR}$ from U's perspective. The major change manifests itself in wealth dynamics and in the interest rate equations which is additionally dependent upon the evolution of old and new asset values in R:

$$\text{Wealth dynamics:} \quad \dot{w}_t = (r_t - \theta - g) w_t + (1 - \delta - \frac{\kappa}{x_t}) + \\ + g^n \left[\frac{(1-\kappa)}{\theta x_t} + \hat{v}_t^{Ro} \frac{x_t^{Ro}}{x_t} \left(\frac{\delta}{\delta^R} - 1 \right) \right] - \kappa_p \frac{x_t^{Rn}}{x_t} \tag{1.8}$$

$$\text{Terms of trade:} \quad 1 = \theta \gamma w_t P_t^{U(\sigma-1)} + (1-\gamma) \left(\frac{(1-\kappa)}{x_t} - \theta w_t \right) P_t^{R(\sigma-1)} \tag{1.9}$$

$$\text{Output dynamics:} \quad \dot{x}_t = x_t (1 - x_t) \left(g - g^R - \frac{\dot{q}_t^R}{q_t^R} \right) \tag{1.10}$$

$$\text{Interest rate:} \quad r_t = g^z + x_t^R \frac{\dot{q}_t^R}{q_t^R} + \frac{\theta}{(1-\kappa)} [\delta - (\delta - \delta^R) x_t^{Ro} - \\ - g^n \hat{v}_t^{Ro} x_t^R \left(\frac{\delta}{\delta^R} - 1 \right)] \tag{1.11}$$

The wealth dynamics equation (1.8) is a variation of the global asset demand equation (**??**). It can be derived as follows:

$$\begin{aligned}
\dot{W}_t^U &= (r_t - \theta) W_t^U + (1-\delta) X_t^U + g^n (V_t^U + N_t^R v_t^{Rn}) - \kappa_p q_t^R X_t^{Rn} - \kappa X_t \\
\dot{w}_t &= (r_t - \theta - g) w_t + (1-\delta) + g^n (V_t^U + N_t^R v_t^{Rn}) / \dot{X}_t^U - \kappa_p x_t^{Rn} / x_t - \kappa / x_t \\
&= (r_t - \theta - g) w_t + (1-\delta) + g^n (V_t - N_0^R (v_t^{Ro} - v_t^{Rn})) / \dot{X}_t^U - \kappa_p x_t^{Rn} / x_t - \kappa / x_t \\
&= (r_t - \theta - g) w_t + (1-\delta) + g^n (V_t + N_0^R v_t^{Ro} (\delta / \delta^R - 1)) / \dot{X}_t^U - \kappa_p x_t^{Rn} / x_t - \kappa / x_t \\
&= (r_t - \theta - g) w_t + (1-\delta) + g^n (V_t + V_t^{Ro} (\delta / \delta^R - 1)) / \dot{X}_t^U - \kappa_p x_t^{Rn} / x_t - \kappa / x_t \\
\dot{w}_t &= (r_t - \theta - g) w_t + (1-\delta) + g^n ((1-\kappa)/(\theta x_t) + \hat{v}_t^{Ro} x_t^{Ro} / x_t (\delta / \delta^R - 1)) - \\
&\quad \kappa_p x_t^{Rn} / x_t - \kappa / x_t
\end{aligned}$$

The terms of trade equation (1.9) utilises the equilibrium condition on the goods market using $P^iC^i = \theta W^i$:

$$
\begin{aligned}
X_t^U &= \sum_i \gamma_{ij} C^i \left(\frac{q^j}{Pi}\right)^{-\sigma} \\
&= \gamma\theta W_t^U P_t^{U(\sigma-1)} + (1-\gamma)\theta W_t^R P_t^{R(\sigma-1)} \\
&= \gamma\theta W_t^U P_t^{U(\sigma-1)} + (1-\gamma)\theta (W_t - W_t^U) P_t^{R(\sigma-1)} \\
&= \gamma\theta W_t^U P_t^{U(\sigma-1)} + (1-\gamma)\theta \left(\frac{(1-\kappa)}{\theta} X_t - W_t^U\right) P_t^{R(\sigma-1)} \\
\frac{X_t^U}{X_t^U} &= \theta\gamma w_t P_t^{U(\sigma-1)} + (1-\gamma)\left(\frac{(1-\kappa)}{x_t} - \theta w_t\right) P_t^{R(\sigma-1)} \\
1 &= \theta\gamma w_t \left(\gamma + (1-\gamma) q_t^{R(1-\sigma)}\right)^{-1} + (1-\gamma)\left(\frac{(1-\kappa)}{x_t} - \theta w_t\right) \\
&\quad \left(\gamma q_t^{R(1-\sigma)} + (1-\gamma)\right)^{-1}
\end{aligned}
$$

with $P_t^i = \left(\sum_j \gamma_{ij} q_t^{j(1-\sigma)}\right)^{1/(1-\sigma)}$ for $j = \{U,R\}$. A shooting mechanism is used to find the value for q_t^R for values of w_t and x_t derived using the dynamic equations of the preceding period. Final values are obtained using the above equations while the guess for V_{0+}^i is iteratively updated every time the shooting algorith reaches the end of the loop at $t = \infty$:

$$
\begin{aligned}
\text{Guess for asset values: } V_{0+}^i &= \delta^i \int_0^\infty q_t^i X_t^i e^{-\int_0^s r_u du} ds \\
&= q_0^i X_0^i \delta^i \int_0^\infty e^{-\theta \int_0^s \bar{\delta}_u du} \frac{x_s^i}{x_0^i} ds
\end{aligned}
$$

The average capitalisation ratio in this equation is time-varying since relative output is constantly changing, $\bar{\delta} = \sum_i x_t^i \delta^i$, and it is also dependent upon exchange rate changes. For the combined case, we need a new estimation for V_{0+}^R since it now consists of the old and new trees exhibiting different capitalisation rates δ^i. The initial guess for their aggregate asset value is:

$$
\begin{aligned}
V_{0+}^R &= V_{0+}^{Ro} + V_{0+}^{Rn} \\
&= \delta^R \int_0^\infty q_t^R X_t^{Ro} e^{-\int_0^s r_u du} ds + \delta \int_0^\infty q_t^R X_t^{Rn} e^{-\int_0^s r_u du} ds
\end{aligned}
\tag{1.12}
$$

The values for $V_{0^+}^{Ro}$ and $V_{0^+}^{Rn}$ have to correspond to the values obtained in the reverse solution using the asymptotic property $\lim_{t \to \infty} x^{Ro} = 0$. The derivation is described in section 1.1.5.

1.2 Graphischer Anhang

Abb. 1.2 Variation des Finanzmarktparameters δ^R.

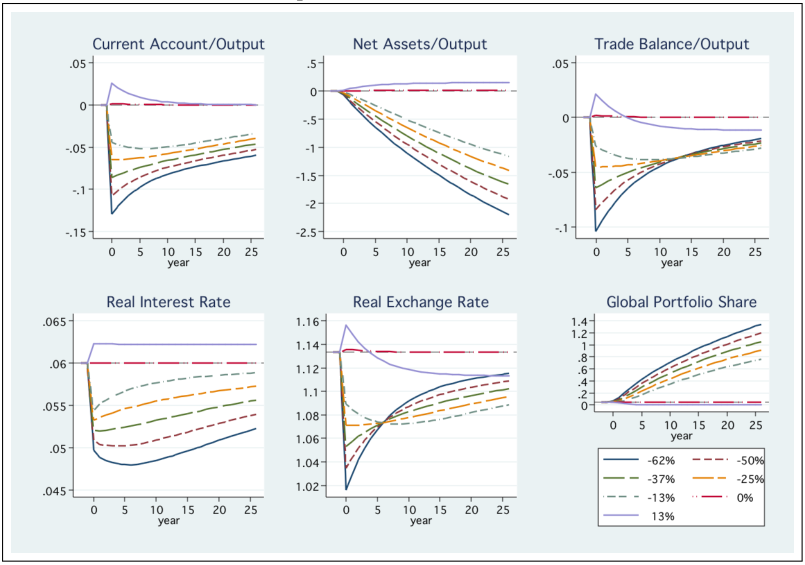


Abb. 1.3 Variation des Anpassungsparameters der TOT σ.

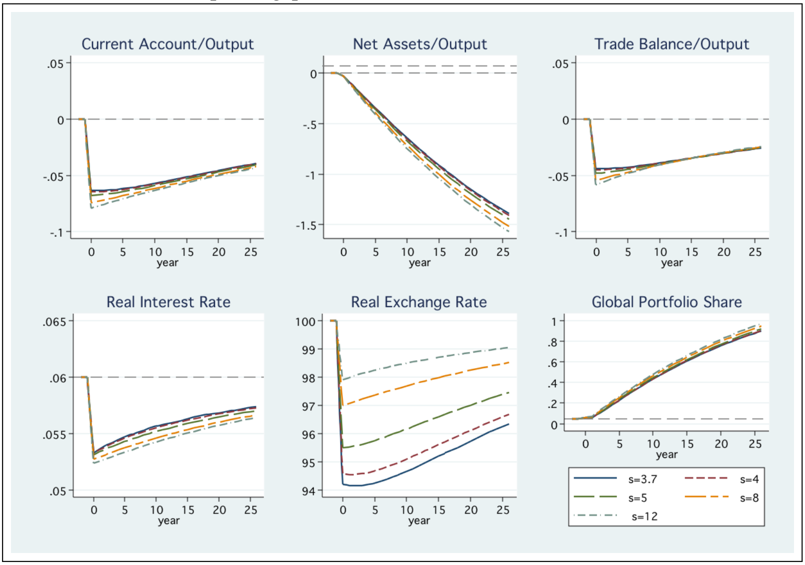


Abb. 1.4 Variation des Investitionskostenparameters κ.

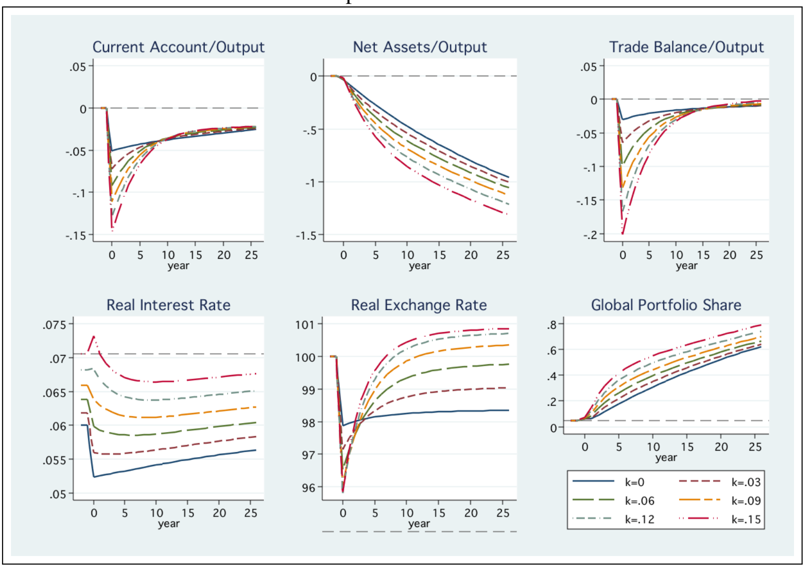


Abb. 1.5 Variation des FDI-Verhandlungspreisparameters κ_P.

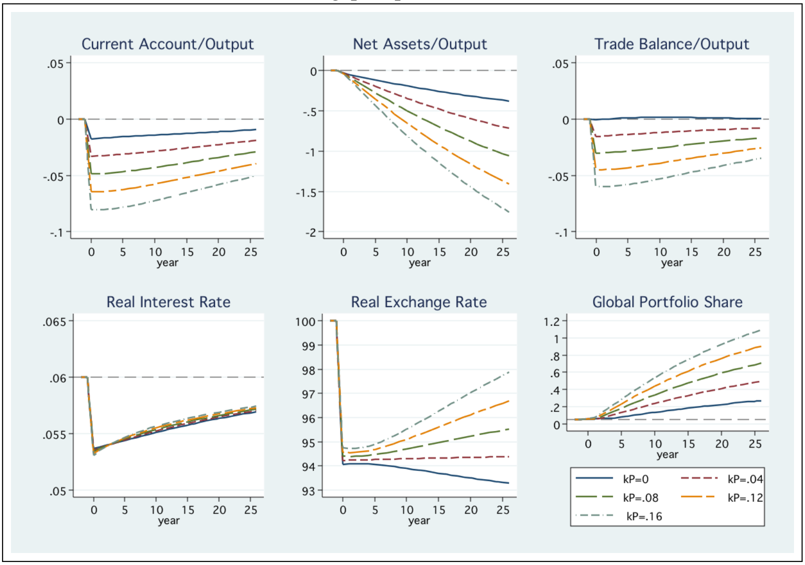


Abb. 1.6 Variation der Inzidenz der gesamten FDI-Kosten für $\kappa_P + \kappa = .12$.

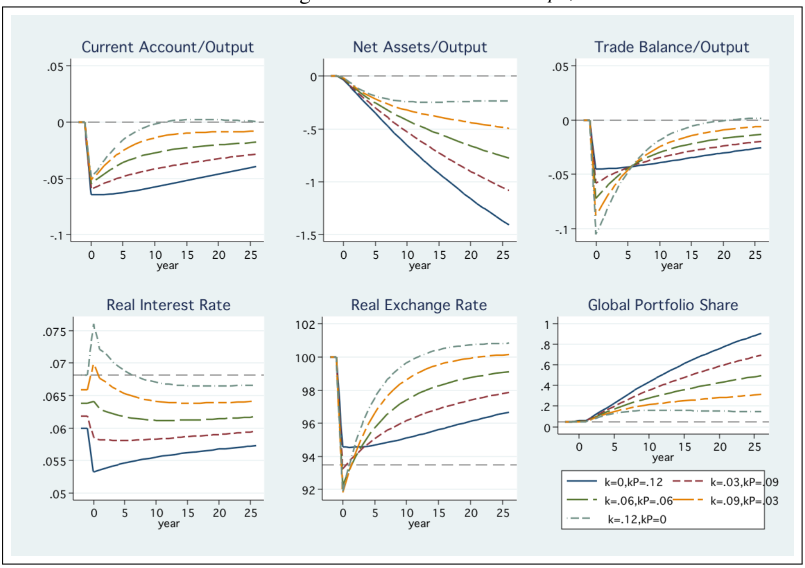


Abb. 1.7 Variation der totalen Faktorproduktivität g^z in R.

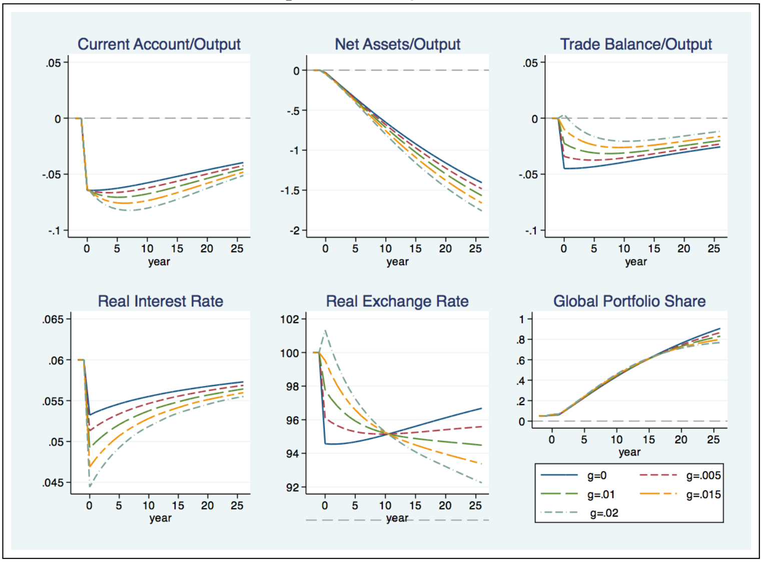
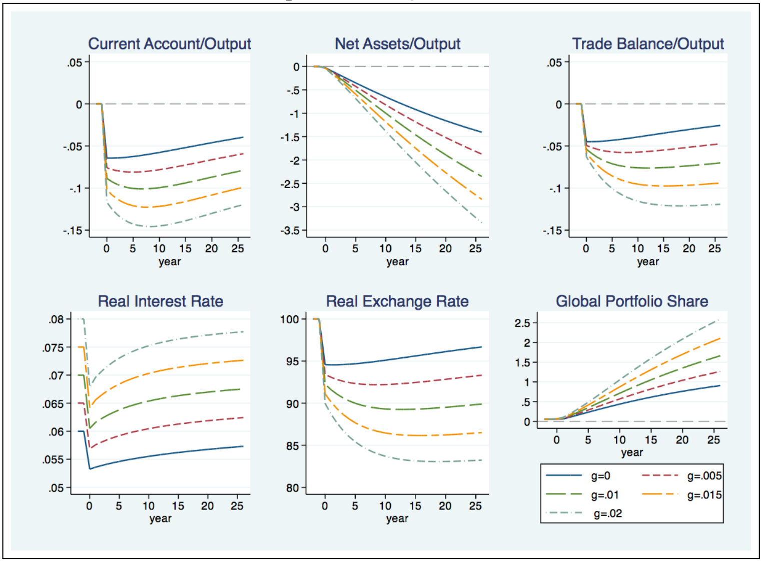


Abb. 1.8 Variation der totalen Faktorproduktivität g^z in U.



1.3 Statistischer Anhang

1.3.1 Datenquellen

BIS Bank für Internationalen Zahlungsausgleich, Statistics
http://www.bis.org/statistics/
Bundesbank Deutsche Bundesbank
http://www.bundesbank.de/Navigation/DE/Statistiken/statistiken.html
destatis Deutsches Statistisches Bundesamt Genesis-Online-Datenbank
https://www-genesis.destatis.de/genesis/online
Eurostat Europan Commission Eurostat Statistics Database
http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home
FRED2 Datenbank der Federal Reserve Bank of St. Louis, https://research.stlouisfed.org/fred2/
IMF *International Monetary Fund*, Data and Statistics http://www.imf.org/external/data.htm
Maddison 2010 Angus Maddison, Historical Statistics
http://www.ggdc.net/MADDISON/oriindex.htm
NBS *National Bureau of Statistics of China* China Statistical Database
http://www.stats.gov.cn/english/statisticaldata/yearlydata/
OECD iLibrary *Organization for Economic Cooperation and Development* OECD iLibrary
http://www.oecd-ilibrary.org/
PBC *People's Bank of China* Survey & Statistics
http://www.pbc.gov.cn/publish/english/984/index.html
SAFE *State Administration on Foreign Exchange* Data and Statistics
http://www.safe.gov.cn/model_safe_en/tjsj_en/tjsj_list_en.jsp
Statista Statista Statistik–Portal
http://de.statista.com/
SourceOECD The OECD's Online Library of Statistical Databases, Books and Periodicals
http://caliban.sourceoecd.org/vl=89783355/cl=16/nw=1/rpsv/home.htm

1.3.2 Weitere Modellergebnisse

In den folgenden Tabellen werden die Ergebnisse der kalibrierten Simulationen des Ausgangsmodells sowie der Sensitivitätsanalyse der Parameter präsentiert. Für folgende Parameterwerte werden die Ergebnisse dargestellt:

Tabelle	**1**	**2**	**3**	**4**	**5**	**Beschreibung**
Tab. 1.3	FDI+XR	FDI	XR	**Basis**		Basismodell und Erweiterungen
Tab. 1.4	**9**	12	15	18	21-27	Finanzmarktparameter δ^R
Tab. 1.5	3,7	**4**	5	8	12	CES-Parameter σ
Tab. 1.6	**0**	3	6	9	12-15	Nettoinvestitionskosten κ
Tab. 1.7	0	4	8	**12**	16	FDI-Parameter κ_P
Tab. 1.8	**0/12**	3/9	6/6	9/3	12/0	Variation von κ und κ_P
Tab. 1.9	**0**	0,5	1,0	1,5	2,0	TFP-Wachstum in R
Tab. 1.10	**0**	0,5	1,0	1,5	2,0	TFP-Wachstum in U
Tab. 1.11	100	150	200			Simulationshorizont in Perioden
Tab. 1.12	**0/12**	4/8	4/4	8/8		Aktualisierte Investitionskosten

Anmerkung. Alle Werte in Prozent; Werte des Ausgangsmodell von Caballero, Farhi und Gourinchas (2008) hervorgehoben.

Tabelle 1.2 Übersicht der Ergebnistabellen

Tabelle 1.3: Ergebnistabelle für Variation des Basismodells

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
Joint model, FDI, XR							
	-2	0	0	0	.06	100	.05
	0	-.064441	-.032315	-.044972	.053239	94.568283	.057158
	3	-.063865	-.225319	-.044399	.054138	94.578491	.144091
	10	-.057735	-.651972	-.03931	.05552	95.099136	.436608
	15	-.051964	-.922049	-.034775	.056197	95.60862	.613499
	50	-.020443	-2.04959	-.012697	.058673	98.297272	1.258188
pure FDI, noXR							
	-2	0	0	0	.06	100	.05
	0	-.048704	-.036479	-.012857	.051038	100	.058779
	3	-.062838	-.19958	-.030625	.051824	100	.121789
	10	-.065658	-.656524	-.036562	.053393	100	.410001
	15	-.060342	-.96666	-.032785	.054321	100	.600987
	50	-.024206	-2.265196	-.01076	.05801	100	1.352625
pure XR, noFDI							
	-2	0	0	0	.06	100	.05
	0	-.048476	-.027935	-.0406	.055767	95.985626	.054674
	3	-.025194	-.146422	-.010632	.055577	99.525291	.086095
	10	-.004939	-.245635	.015804	.055425	102.58612	.13635
	15	-.001185	-.261467	.020954	.055397	103.16672	.141637
	50	.000151	-.257653	.02373	.055381	103.47815	.13661
baseline, noFDI, noXR							
	-2	0	0	0	.06	100	.05
	0	-.023694	-.030596	0	.055381	100	.055348
	3	-.01487	-.091593	.011334	.055381	100	.053463
	10	-.00734	-.166818	.021007	.055381	100	.091263
	15	-.005938	-.200119	.022809	.055381	100	.108866
	50	-.005219	-.38648	.023731	.055381	100	.209608

Tabelle 1.4: Results summary for variations of deltaR.

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
δ^R =9	-2	0	0	0	.06	100	.05
	0	-.129492	-.071014	-.104168	.049707	89.667755	.062406
	3	-.107271	-.43568	-.077378	.048232	92.797485	.279495
	10	-.082106	-1.104198	-.044609	.0485	96.337318	.708437
	15	-.073239	-1.497039	-.033011	.049596	97.396133	.941837
	50	-.034314	-3.140224	-.007305	.056209	99.2537	1.872805
δ^R =12	-2	0	0	0	.06	100	.05
	0	-.10729	-.05797	-.084057	.050903	91.315163	.060504
	3	-.092495	-.363997	-.066024	.050215	93.414551	.232999
	10	-.073892	-.949004	-.042463	.050906	95.945045	.616955
	15	-.066052	-1.299075	-.033287	.051851	96.821556	.83181
	50	-.029443	-2.769258	-.009121	.057035	98.940964	1.667109
δ^R =15	-2	0	0	0	.06	100	.05
	0	-.085769	-.045152	-.064427	.052066	92.945137	.058787
	3	-.078132	-.294374	-.055071	.052175	94.007515	.188247
	10	-.065816	-.799371	-.040691	.053242	95.535782	.527128
	15	-.059021	-1.108892	-.033887	.054038	96.226181	.723349
	50	-.024849	-2.40803	-.010908	.057854	98.623497	1.463619
δ^R =18	-2	0	0	0	.06	100	.05
	0	-.064712	-.032474	-.045199	.053228	94.548759	.057182
	3	-.064038	-.226188	-.044512	.054112	94.574234	.144648
	10	-.057809	-.653814	-.039289	.055506	95.108788	.437772
	15	-.052077	-.924473	-.034784	.056162	95.613411	.61496
	50	-.020494	-2.054063	-.012675	.058663	98.301224	1.260785
δ^R =21	-2	0	0	0	.06	100	.05
	0	-.043914	-.019872	-.026308	.054409	96.116493	.055626
	3	-.05009	-.15896	-.034351	.056052	95.109871	.101746
	10	-.049858	-.510988	-.038303	.05768	94.656479	.34773
	15	-.045132	-.743878	-.035933	.058252	94.987709	.505095
	50	-.01635	-1.704834	-.014433	.059466	97.974815	1.056981
δ^R =24	-2	0	0	0	.06	100	.05
	0	.00155	.000333	.002039	.06	100.23644	.049956
	3	.000824	.003993	.001084	.06	100.12723	.047682
	10	.000188	.007205	.000248	.06	100.02969	.045766
	15	.000066	.007826	.000086	.06	100.01083	.045415
	50	0	.008158	0	.06	100.00002	.045241
δ^R =27	-2	0	0	0	.06	100	.05
	0	.02543	.014237	.021547	.062313	102.05757	.047667
	3	.013652	.074258	.005883	.062266	100.29894	.011415
	10	.0033	.127743	-.007773	.062226	98.703262	0
	15	.001263	.138699	-.010397	.062218	98.39399	0

Continued on next page...

... table 1.4 continued

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
	50	.000027	.147386	-.011791	.062214	98.228844	0

Tabelle 1.5: Ergebnisübersicht für Variation von sigma

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
σ=3,7	-2	0	0	0	.06	100	.05
	0	-.063318	-.032254	-.043779	.053273	94.201508	.056738
	3	-.063043	-.222198	-.043584	.054223	94.160782	.142013
	10	-.057254	-.64415	-.039027	.055673	94.654938	.433578
	15	-.051513	-.911862	-.03464	.056319	95.205086	.610328
	50	-.020232	-2.030273	-.012837	.058725	98.118965	1.249827
σ=4,0	-2	0	0	0	.06	100	.05
	0	-.064441	-.032315	-.044972	.053239	94.568291	.057158
	3	-.063865	-.225318	-.044399	.054138	94.578453	.144091
	10	-.057735	-.651969	-.039309	.055522	95.09893	.436605
	15	-.051964	-.922053	-.034775	.056196	95.608704	.613503
	50	-.020443	-2.04959	-.012697	.058673	98.297264	1.258188
σ=5,0	-2	0	0	0	.06	100	.05
	0	-.067674	-.032553	-.048238	.053107	95.500305	.05834
	3	-.066174	-.234258	-.046511	.053879	95.620964	.150083
	10	-.059071	-.673805	-.039933	.055187	96.161964	.445657
	15	-.053054	-.950204	-.034904	.055887	96.597954	.623388
	50	-.020993	-2.101665	-.012346	.058549	98.710075	1.281788
σ=8,0	-2	0	0	0	.06	100	.05
	0	-.074226	-.033254	-.054287	.052726	97.003639	.060672
	3	-.070659	-.252312	-.049963	.053354	97.220703	.162338
	10	-.061602	-.716189	-.04063	.054618	97.691483	.465232
	15	-.055126	-1.004156	-.034865	.055365	97.987427	.645692
	50	-.021947	-2.196604	-.011801	.058359	99.258179	1.327997
σ=12,0	-2	0	0	0	.06	100	.05
	0	-.079194	-.033936	-.058483	.05236	97.916473	.062426
	3	-.073957	-.265984	-.052058	.052959	98.131271	.171736
	10	-.063465	-.747372	-.040862	.054254	98.500412	.480843
	15	-.056659	-1.043482	-.034701	.055045	98.704025	.663922
	50	-.022577	-2.262859	-.011479	.058249	99.528885	1.362032

Tabelle 1.6: Ergebnisübersicht für Variation von kappa

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
κ=0	-2	0	0	0	.06	100	.05
	0	-.050839	-.03388	-.030509	.05239	97.87207	.062537
	3	-.045415	-.180204	-.023891	.052893	98.065834	.116685
	10	-.037808	-.470803	-.016216	.054137	98.250427	.307478
	15	-.033673	-.647027	-.01336	.054936	98.300346	.422379
	50	-.011752	-1.347712	-.006015	.058216	98.403954	.874183
κ=3	-2	0	0	0	.061856	100	.05
	0	-.072201	-.029508	-.06499	.055983	97.14431	.063115
	3	-.053968	-.22553	-.041802	.055751	97.99955	.151382
	10	-.036474	-.538846	-.019773	.05637	98.758247	.351454
	15	-.031297	-.707933	-.01416	.057033	98.92141	.457825
	50	-.012226	-1.390942	-.004798	.060136	99.118156	.887793
κ=6	-2	0	0	0	.06383	100	.05
	0	-.092149	-.02552	-.098813	.059837	96.5121	.064086
	3	-.062674	-.268926	-.059979	.058788	98.04528	.187434
	10	-.035727	-.609225	-.023373	.058739	99.336723	.396443
	15	-.029439	-.774668	-.014803	.059263	99.595505	.495645
	50	-.012903	-1.452096	-.003261	.06218	99.860458	.9107
κ=9	-2	0	0	0	.065934	100	.05
	0	-.111062	-.021886	-.132559	.063977	96.047775	.065693
	3	-.071751	-.311616	-.078672	.06192	98.204872	.226108
	10	-.035554	-.683985	-.026821	.061208	99.855782	.443922
	15	-.028105	-.849336	-.015105	.061615	100.16175	.537501
	50	-.0138	-1.534117	-.001339	.064357	100.44504	.945703
κ=12	-2	0	0	0	.068182	100	.05
	0	-.129143	-.018574	-.166664	.068439	95.807251	.068249
	3	-.081367	-.354385	-.098018	.065067	98.448967	.268661
	10	-.035946	-.764616	-.029982	.063768	100.22785	.495103
	15	-.027301	-.933511	-.014968	.064091	100.5313	.584638
	50	-.014914	-1.638808	.001015	.066677	100.79386	.99409
κ=15	-2	0	0	0	.070588	100	.05
	0	-.146341	-.015538	-.201368	.07328	95.837883	.072184
	3	-.09159	-.397623	-.118034	.068177	98.738197	.316351
	10	-.0369	-.851918	-.032826	.066433	100.43659	.550677
	15	-.027026	-1.028103	-.014376	.066699	100.70654	.637595
	50	-.016238	-1.767018	.003845	.069151	100.92847	1.055737

Tabelle 1.7: Ergebnisübersicht für Variation von kappaP

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
κ_P=0	-2	0	0	0	.070588	100	.05
	0	-.127165	-.01551	-.182469	.073309	95.814247	.072267
	3	-.071274	-.338187	-.097963	.068143	98.707626	.269628
	10	-.018028	-.64866	-.014934	.066381	100.3498	.423872
	15	-.010034	-.730067	.000941	.06665	100.57568	.461063
	50	-.008135	-1.047546	.006757	.069135	100.58404	.659818
κ_P=.04	-2	0	0	0	.070588	100	.05
	0	-.142504	-.015533	-.197585	.073286	95.833229	.0722
	3	-.087525	-.385731	-.114018	.06817	98.732132	.307018
	10	-.033121	-.811249	-.029243	.066423	100.41939	.525525
	15	-.023621	-.968458	-.011307	.06669	100.68074	.602777
	50	-.014608	-1.622773	.00443	.069148	100.86222	.979653
κ_P=.08	-2	0	0	0	.070588	100	.05
	0	-.157859	-.015554	-.212726	.073263	95.851593	.072136
	3	-.10379	-.433314	-.13009	.068198	98.75631	.344319
	10	-.048251	-.973973	-.043588	.066464	100.48772	.625517
	15	-.03726	-1.207143	-.023601	.066728	100.78284	.74062
	50	-.021152	-2.20078	.002081	.069159	101.12014	1.27539
κ_P=.12	-2	0	0	0	.070588	100	.05
	0	-.173231	-.015576	-.227889	.073241	95.869377	.072074
	3	-.120069	-.480933	-.14618	.068225	98.779945	.381533
	10	-.063416	-1.136831	-.057966	.066505	100.55478	.723887
	15	-.05095	-1.446119	-.035938	.066765	100.88206	.874745
	50	-.027763	-2.781464	-.000289	.06917	101.35915	1.549638
κ_P=.16	-2	0	0	0	.070588	100	.05
	0	-.188619	-.015596	-.243073	.073219	95.886612	.072014
	3	-.136361	-.528588	-.162287	.068253	98.803574	.41866
	10	-.078616	-1.299818	-.072377	.066544	100.6209	.820673
	15	-.06469	-1.685382	-.048318	.066801	100.97887	1.005299
	50	-.034435	-3.364714	-.00268	.069179	101.58144	1.804643

Tabelle 1.8: Ergebnisübersicht für Variation der Investitionskostenparameter κ und κ_P

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
$\kappa = 0$	-2	0	0	0	.06	100	.05
$\kappa_P = .12$	0	-.079194	-.033936	-.058483	.05236	97.916473	.062426
	3	-.073957	-.265984	-.052058	.052959	98.131271	.171736
	10	-.063465	-.747371	-.040862	.054254	98.500412	.480843
	15	-.056659	-1.043482	-.034701	.055045	98.704025	.663922
	50	-.022577	-2.262858	-.011479	.058249	99.528885	1.362032
$\kappa = .03$	-2	0	0	0	.061856	100	.05
$\kappa_P = .09$	0	-.08816	-.029539	-.08071	.055964	97.172134	.063043
	3	-.070205	-.273935	-.05782	.05579	98.039719	.183554
	10	-.051224	-.697091	-.033923	.056441	98.907188	.450721
	15	-.044536	-.936106	-.026413	.057098	99.16111	.596436
	50	-.018469	-1.924416	-.007806	.060155	99.772377	1.175927
$\kappa = .06$	-2	0	0	0	.06383	100	.05
$\kappa_P = .06$	0	-.096088	-.025527	-.102691	.059832	96.519142	.064067
	3	-.066724	-.280929	-.063975	.058797	98.055275	.195713
	10	-.039439	-.648944	-.026927	.058756	99.371895	.421361
	15	-.032775	-.832239	-.017875	.059279	99.651192	.530492
	50	-.014479	-1.58818	-.003977	.062184	100.00989	.985204
$\kappa = .09$	-2	0	0	0	.065934	100	.05
$\kappa_P = .03$	0	-.103265	-.021873	-.124882	.063987	96.03466	.065731
	3	-.063657	-.287741	-.070681	.061903	98.186554	.208968
	10	-.028086	-.604065	-.019691	.061178	99.795868	.393781
	15	-.021387	-.732964	-.00896	.061587	100.06854	.46732
	50	-.01062	-1.256648	9.000e-06	.064348	100.19715	.79171
$\kappa = .12$	-2	0	0	0	.068182	100	.05
$\kappa_P = 0$	0	-.109823	-.018544	-.147632	.068466	95.778519	.068338
	3	-.061106	-.294896	-.078007	.065027	98.41021	.224045
	10	-.017176	-.563265	-.012122	.063705	100.11005	.368741
	15	-.010412	-.639216	.000371	.064031	100.35121	.407774
	50	-.006903	-.932568	.004159	.066657	100.313	.597064

Tabelle 1.9: Ergebnisübersicht des Produktivitätswachstums in R g^R

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
$g^R = 0$	-2	0	0	0	.06	100	.05
	0	-.064438	-.032314	-.044972	.053239	94.568367	.057157
	3	-.063864	-.22531	-.0444	.054138	94.57827	.144086
	10	-.057736	-.651963	-.039315	.055512	95.099586	.436602
	15	-.051969	-.922005	-.03478	.056204	95.606636	.61346
	50	-.020443	-2.049548	-.012697	.058673	98.297302	1.258162
$g^R = .5$	-2	0	0	0	.06	100	.05
	0	-.064075	-.035943	-.03417	.051311	96.109444	.05903
	3	-.066519	-.231236	-.037077	.052691	95.492416	.145699
	10	-.062335	-.682398	-.035427	.054654	95.145912	.43971
	15	-.056271	-.971284	-.031713	.055548	95.244987	.610532
	50	-.020822	-2.13883	-.0105	.058416	95.715958	1.080637
$g^R = 1.0$	-2	0	0	0	.06	100	.05
	0	-.063642	-.040271	-.022493	.049188	97.767105	.061263
	3	-.069633	-.238247	-.029125	.051125	96.462311	.147792
	10	-.067606	-.717596	-.031041	.053759	95.194839	.444913
	15	-.061099	-1.027452	-.028138	.054859	94.878685	.610523
	50	-.020863	-2.228172	-.007624	.058113	93.178032	.92884
$g^R = 1.5$	-2	0	0	0	.06	100	.05
	0	-.063022	-.045401	-.009967	.046887	99.519402	.063895
	3	-.073224	-.246258	-.020569	.049459	97.473572	.150286
	10	-.073597	-.75771	-.026111	.052826	95.239212	.452106
	15	-.066478	-1.090606	-.023976	.054154	94.497025	.613054
	50	-.020363	-2.312866	-.003843	.05776	90.688911	.797182
$g^R = 2.0$	-2	0	0	0	.06	100	.05
	0	-.062068	-.051434	.003352	.044441	101.33709	.066953
	3	-.077295	-.255097	-.011451	.04772	98.506454	.153056
	10	-.08034	-.802733	-.020613	.051839	95.269127	.461099
	15	-.072467	-1.160468	-.019151	.053487	94.086258	.617552
	50	-.019039	-2.386481	.001148	.057357	88.252296	.681388

Tabelle 1.10: Ergebnisübersicht des Produktivitätswachstums in U g^U

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
$g^U = 0$	-2	0	0	0	.06	100	.05
	0	-.064438	-.032314	-.044972	.053239	94.568367	.057157
	3	-.063864	-.22531	-.0444	.054138	94.57827	.144086
	10	-.057736	-.651963	-.039315	.055512	95.099586	.436602
	15	-.051969	-.922005	-.03478	.056204	95.606636	.61346
	50	-.020443	-2.049548	-.012697	.058673	98.297302	1.258162
$g^U = .05$	-2	0	0	0	.065	100	.05
	0	-.075957	-.034895	-.049446	.056853	93.409035	.058799
	3	-.080426	-.267945	-.05528	.058484	92.596611	.173411
	10	-.07824	-.821768	-.057355	.060444	92.207619	.567836
	15	-.072646	-1.188362	-.054847	.061271	92.454681	.821818
	50	-.038236	-2.923283	-.035435	.063746	94.833176	1.878998
$g = 1.0$	-2	0	0	0	.07	100	.05
	0	-.088532	-.037564	-.05401	.060494	92.244629	.060502
	3	-.098028	-.313804	-.066389	.062835	90.613167	.20554
	10	-.099566	-1.000026	-.076034	.065374	89.307846	.71341
	15	-.09396	-1.464772	-.075746	.066324	89.27916	1.054023
	50	-.055962	-3.798668	-.059237	.068814	91.258797	2.563765
$g = 1.5$	-2	0	0	0	.075	100	.05
	0	-.102199	-.040337	-.058579	.064155	91.089874	.062276
	3	-.11675	-.363054	-.077707	.067198	88.650734	.240641
	10	-.121816	-1.187425	-.095449	.070299	86.451347	.874679
	15	-.116003	-1.752202	-.097605	.07137	86.151253	1.312584
	50	-.07364	-4.677481	-.084201	.073876	87.682526	3.320415
$g = 2.0$	-2	0	0	0	.08	100	.05
	0	-.117011	-.043233	-.063067	.067831	89.959251	.064132
	3	-.13669	-.415908	-.089205	.071579	86.730576	.27891
	10	-.145088	-1.38468	-.115687	.075233	83.687576	1.052913
	15	-.138847	-2.051587	-.120538	.076416	83.133553	1.599697
	50	-.091288	-5.561126	-.110442	.078933	84.212456	4.154188

Tabelle 1.11: Ergebnisübersicht der Variation der Anzahl der Perioden N

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
$N = 100$	-2	0	0	0	.068182	100	.05
	0	-.102361	-.019974	-.134464	.067187	95.758186	.069293
	3	-.056395	-.276277	-.070026	.064229	98.149269	.214746
	10	-.017205	-.526274	-.012577	.063574	99.658089	.361473
	15	-.011056	-.601597	-.002285	.064163	99.850845	.405189
	50	-.003848	-.836548	.000935	.067138	99.77285	.567995
$N = 200$	-2	0	0	0	.068182	100	.05
	0	-.103247	-.019762	-.136744	.067374	95.653069	.069526
	3	-.056702	-.278102	-.071346	.064356	98.103027	.216903
	10	-.016954	-.528344	-.012936	.063635	99.647934	.363267
	15	-.010749	-.602473	-.002453	.064208	99.846069	.405993
	50	-.003762	-.831792	.000906	.067148	99.772156	.56481
$N = 300$	-2	0	0	0	.068182	100	.05
	0	-.103249	-.019762	-.136749	.067374	95.652802	.069526
	3	-.056703	-.278107	-.071349	.064356	98.102905	.216908
	10	-.016953	-.528349	-.012937	.063635	99.647911	.363272
	15	-.010748	-.602476	-.002453	.064209	99.846069	.405995
	50	-.003761	-.831781	.000906	.067148	99.772156	.564802
$N = 400$	-2	0	0	0	.068182	100	.05
	0	-.103249	-.019762	-.136749	.067374	95.652802	.069526
	3	-.056703	-.278107	-.071349	.064356	98.102913	.216908
	10	-.016953	-.528349	-.012937	.063635	99.647911	.363272
	15	-.010748	-.602476	-.002453	.064209	99.846069	.405995
	50	-.003761	-.831782	.000906	.067148	99.772156	.564803
$N = 500$	-2	0	0	0	.068182	100	.05
	0	-.103248	-.019761	-.136749	.067374	95.652817	.069526
	3	-.056702	-.278104	-.071349	.064356	98.102913	.216906
	10	-.016953	-.528343	-.012937	.063635	99.647911	.363268
	15	-.010748	-.602468	-.002453	.064209	99.846069	.40599
	50	-.003761	-.831768	.000906	.067148	99.772156	.564794

Tabelle 1.12: Ergebnisübersicht der Variation mit neu kalibrierten Investitionskosten

Variable	Periode	CAU_{XU}	NAU_{XU}	TBU_{XU}	r_t	λ_{UR}	μ_{RU}
$\kappa = 0$	-2	0	0	0	.06	100	.05
$\kappa_P = .12$	0	-.064441	-.032315	-.044972	.053239	94.568283	.057158
	3	-.063865	-.225319	-.044399	.054138	94.578491	.144091
	10	-.057735	-.651972	-.03931	.05552	95.099136	.436608
	15	-.051964	-.922049	-.034775	.056197	95.60862	.613499
	50	-.020443	-2.04959	-.012697	.058673	98.297272	1.258188
$\kappa = .04$	-2	0	0	0	.0625	100	.05
$\kappa_P = .08$	0	-.058072	-.023462	-.062916	.060378	92.855072	.053091
	3	-.048754	-.187868	-.050362	.059479	94.483627	.12399
	10	-.036739	-.489526	-.033239	.05926	96.553505	.327989
	15	-.031956	-.664363	-.026422	.059533	97.332184	.439036
	50	-.013932	-1.421665	-.008072	.061353	99.33799	.882921
$\kappa = .04$	-2	0	0	0	.0625	100	.05
$\kappa_P = .04$	0	-.042976	-.023237	-.048797	.06054	92.674995	.052977
	3	-.033094	-.141293	-.035541	.059466	94.309052	.093462
	10	-.02191	-.3346	-.019269	.059116	96.14772	.226744
	15	-.018563	-.438166	-.014068	.059302	96.672501	.295211
	50	-.00762	-.872255	-.004719	.061292	97.412209	.575431
$\kappa = .08$	-2	0	0	0	.065217	100	.05
$\kappa_P = .08$	0	-.067232	-.016709	-.096304	.067802	92.041847	.0505
	3	-.05201	-.203396	-.072649	.064979	95.118584	.138832
	10	-.033324	-.507937	-.040834	.06329	98.61412	.335544
	15	-.027891	-.669908	-.02971	.063114	99.706932	.429879
	50	-.014639	-1.407095	-.006112	.064305	101.82074	.8351

1.4 Technischer Anhang

Der folgende Abschnitt enthält den Quellcode des Hauptprogramms in STATA, das in OnlinePlus als Download ebenfalls zur Verfügung steht. Der Code ist zwischen den Zeilen kommentiert und mit den entsprechenden Gleichungen des Ausgangsmodells referenziert.

Listing 1.1 Kommentierter STATA-Code des Hauptprogramms

```
/*****************************************
*
* dofile: CFG_04_combined_model_calibration.do
* author: Finn Marten Körner
* created: 04/05/2011
*
*****************************************/

clear all

// Define variables
if "`model'" == "04_combined" {
        local theta = .25
        local gn = .03
        local gz = .00
        local g = `gn' + `gz'
        local gR = `g'
        local delta = .24
        local xE  = .5
        local xRo = .3
        local xRn = .0
        local xR  = `xRo' + `xRn'
        local xU = 1 - `xR'
        local muiU = .05
        local g_gE = .0111
        local deltaR = .24
        local r_aut = .06
        local kappa  = .00
        local kappaP = .12
        local sigma = 4
        local gamma = .9
        local shock = .75
        local N = 100
        set obs `N'

        #delimit ;
        foreach var in XU r_t qR qRdotqR XRo XRn XR NU NR ZU ZR
            X xU xRo xRn xR WU
        WU_XU WR WR_XR W_t W_V_eq VU VU_XU vhatRo VRo vRo vRn
            VRn VR VR_XR V_t W_V
```

```
        IU IR P PU PR lambdaUR aRU aUR muRU muUR TBU TBU_XU CAU CAU_XU CAR
        CAR_XR NAU NAU_XU NINV NINV_XU ChiU ChiU_XU
        w x r q qdotq lambda V0U V0R q3 deltaR r_vhatRo vhatU r_vhatU vhatRn
        VU_XU_asympt VR_XR_asympt WU_XU_asympt WR_XR_asympt TB_XU_asympt
        CAU_XU_asympt CAR_XR_asympt
        {;
                qui gen `var' = .;
        };
        #delimit cr

// Possibly increase number of observations
set obs `N'

gen year = _n - 3
tsset year

local to_precision = .000001

// SELECT model variation (select a model by setting = 1, otherwise = 0)

local model_only_joint                          = 0
local model_comparison                          = 1
local model_sensitivity_delta           = 0
local model_sensitivity_sigma           = 0
local model_sensitivity_kappa           = 0
local model_sensitivity_kappaP          = 0
local model_sensitivity_kappa_kappaP= 0
local model_sensitivity_growth          = 0
local model_sensitivity_sizeN           = 0
local model_sensitivity_misc            = 0

// SELECT additional features (graph_normalisation creates r_t = .06 and lambda = 100 common base)
local graph_normalisation_lambda        = "on" // "off" //
local graph_normalisation_r_t           = "off" // "on" //
// local output_file                                    = "_full" // "" //

// Derive list of selected models to loop over.
foreach submodel in model_only_joint model_comparison model_sensitivity_delta
model_sensitivity_sigma model_sensitivity_kappa model_sensitivity_kappaP
model_sensitivity_kappa_kappaP model_sensitivity_growth model_sensitivity_sizeN
```

```
model_sensitivity_misc {
        if ‘‘submodel’’ != 1 continue
        local models "‘models’‘submodel’ "
        local ‘submodel’ = 0
}
local mod_loops ‘"1(1)‘=wordcount("‘models’")’"’
if wordcount("‘models’") == 0 {
        di as error "Process aborted. "as result"No model selected. Please select a model and restart."
}

// START loop for submodel.
local z = 0
foreach mod_loop of numlist ‘mod_loops’ {

        di _newline as result "Start loop for model variations: "as input"‘models’"as result"."
        // SET previous submodel to zero in order to discard old loop values.
        if ‘z’ > 0 local ‘=word("‘models’",‘z’)’ = 0 //"
        local ++z
        // SET next submodel to one in order to fetch new loop values.
        local submodel ‘"‘=word("‘models’",‘z’)’"’
        di _newline as result "Choose model variations for "as input"‘submodel’"as result"."
        local ‘submodel’ = 1

// START model variations of each submodel by setting values to loop over.
if ‘model_sensitivity_delta’ == 1               local loop = ".09(.03).27"
else if ‘model_sensitivity_sigma’ == 1  local loop = "3.7 4 5 8 12" // 1.7 2 2.5 3 4 5 { //2
else if ‘model_sensitivity_kappa’ == 1  local loop = "0(.03).15"
else if ‘model_sensitivity_kappaP’ == 1 local loop = "0(.04).16"
else if ‘model_sensitivity_kappa_kappaP’ == 1 local loop = "0(.03).12"
else if ‘model_sensitivity_growth’ == 1 local loop = ".0(.01).04"
else if ‘model_sensitivity_sizeN’ == 1  local loop = "100(100)500"
else if ‘model_comparison’ == 1                 local loop = "1(1)4"
else if ‘model_sensitivity_misc’ == 1   local loop = "1(1)4"
else if ‘model_only_joint’ == 1                 local loop = "1"

di as result "Start loop for "as input"‘=word("‘models’",‘z’)’" as result" over values "
```

```
as input"`loop'"as result"."
foreach modvar of numlist `loop' {
di as result "Start loop for "as input"`=word("`models'",`z')'" as result" = "
as input"`modvar'"as result"."

// DEFINE model variations
if `model_sensitivity_delta' == 1 {
//      local loop = ".09(.03).27"
        local modvargraph = "deltaR"
        local modvaracr = "dR"
        local modvaracr2 = "V0+"
        local `modvargraph' = `modvar'
        local modvars = "`modvars'`modvar' "
        local modvarname = "_`modvaracr'_`=round(`modvar'*100,1)'"
        local modvarlabel = "`modvaracr'=`modvar'"
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
        local exrates = "on"
        local FDI = "on"
//      local Force_FDI = "on"
}
else if `model_sensitivity_sigma' == 1 {
//      local loop = "3 4 5 8 12"  //1.7 2 2.5 3 4 5 { //2
        local modvargraph = "sigma"
        local modvaracr = "s"
        local `modvargraph' = `modvar'
        local modvars = "`modvars'`modvar' "
        local modvarname = "_`modvaracr'_`=round(`=`modvar'*10',1)'" // "_`modvaracr'_`modvar'" //
        local modvarlabel = "`modvaracr'=`modvar'"
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
        local FDI = "on"
        local Force_FDI = "on"
}
else if `model_sensitivity_kappa' == 1 {
//      local loop = "0(.03).15"
        local kappaP = .05
        local modvargraph = "kappa"
        local modvaracr = "k" //"{bf:{&kappa}{subscript:P}}=`modvar'"
        local `modvargraph' = `modvar'
        local modvars = "`modvars'`modvar' "
        local modvarname = "_`modvaracr'_`=round(`=`modvar'*100',1)'"
        local modvarlabel = "`modvaracr'=`modvar'"
```

```
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
        local FDI = "on"
        local Force_FDI = "on"
}
else if `model_sensitivity_kappaP' == 1 {
//      local loop = "0 .02 .05 .09 .12 .15"
        local modvargraph = "kappaP"
        local modvaracr = "kP"
        local `modvargraph' = `modvar'
        local modvars = "`modvars'`modvar' "
        local modvarname = "_`modvaracr'_`=round(`=`modvar'*100',1)'"
        local modvarlabel = "`modvaracr'=`modvar'"
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
        local FDI = "on"
        local Force_FDI = "on"
}
else if `model_sensitivity_kappa_kappaP' == 1 {
//      local loop = "0(.03).12"
        local modvargraph = "kappa_kappaP"
        local modvaracr1 = "k"
        local modvaracr2 = "kP"
        local kappa = `modvar'
        local kappaP = .12 - `modvar'
        local modvars = "`modvars'`modvar' "
        local modvarname = "_`modvaracr1'_`=round(`=`modvar'*100',1)'_`modvaracr2'_`=round(`=12-`modvar'*100',1)'"
        local modvarlabel = "`modvaracr1'=`modvar',`modvaracr2'=`=.12-`modvar''"
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
        local FDI = "on"
        local Force_FDI = "on"
}
else if `model_sensitivity_growth' == 1 {
//      local loop = "0(.01).04"
        local modvargraph = "growth"
        local modvaracr = "g"
        local gn = `modvar'
        local g = `gz' + `gn'
        local gR = `g'
        local shock = .75
        local modvars = "`modvars'`modvar' "
        local modvarname = "_`modvaracr'_`=round(`=`modvar'*100',1)'"
        local modvarlabel = "`modvaracr'=`modvar'"
```

```
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
        local exrates = "on"
        local FDI = "on"
        local Force_FDI = "on"
}
else if `model_sensitivity_sizeN' == 1 {
//      local loop = "200(100)500"
        local modvargraph = "sizeN"
        local modvaracr = "N"
        local `modvaracr' = `modvar'
        local modvars = "`modvars'`modvar' "
        local modvarname = "_`modvaracr'_`=`modvar''"
        local modvarlabel = "`modvaracr'=`modvar'"
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
        set obs `modvar'
        replace year = _n - 3
}
else if `model_comparison' == 1 {
//      local loop = "1(1)4"
        local modvargraph = "baseline"
        if `modvar' == 1 {
                local modvaracr = "joint"
                local exrates = "on"
                local FDI = "on"
        }
        if `modvar' == 2 {
                local modvaracr = "pureFDI"
                local exrates = "off"
                local FDI = "on"
        }
        if `modvar' == 3 {
                local modvaracr = "pureXR"
                local exrates = "on"
                local FDI = "off"
        }
        if `modvar' == 4 {
                local modvaracr = "baseline"
                local exrates = "off"
                local FDI = "off"
        }
        local modvars = "`modvars'`modvar' "
        local modvarname = "_model_`modvar'"
        local modvarlabel = "`modvaracr'"
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
}
else if `model_sensitivity_misc' == 1 {
```

```
//      local loop = "1(1)2"
        if `modvar' == 1 {
                local kappa = .00
                local kappaP = .12
                local modlabel "baseline"
        }
        if `modvar' == 2 {
                local kappa = .04
                local kappaP = .08
                local modlabel "low-high"
        }
        if `modvar' == 3 {
                local kappa = .04
                local kappaP = .04
                local modlabel "low-low"
        }
        if `modvar' == 4 {
                local kappa = .08
                local kappaP = .08
                local modlabel "high-high"
        }
        local modvargraph = "new_kappas"
        local modvaracr1 = "k"
        local modvaracr2 = "kP"
        local modvars = "`modvars'`modvar' "
        local modvarname = "_`modvaracr1'_`=round(`=`kappa'*100'
            ,1)'_`modvaracr2'_`=round(`=`kappaP'*100',1)'"
        local modvarlabel = "`modlabel'"
//      local modvarlabel = "`modvaracr1'=`kappa',`modvaracr2'=`
    kappaP'"
        local modvarlegend "label(`=wordcount("`modvars'")' `
            modvarlabel') "
        local FDI = "on"
        local Force_FDI = "on"
}

else if `model_only_joint' == 1 {
//      local loop = ""
}


// ############################################
// Solving the model with exchange rates
// Initial values
// ############################################

qui replace XU = 1 - `xR' in 1/3
label variable XU "Output of U"
qui replace r_t = `gz' + `theta'/(1-`kappa')*`delta' in 1/3
```

```
label variable r_t "Real interest rate"
local r_aut = r_t[1]

// ###############################################
// Exchange rate extensions
// Using iterative approach to solve for qR.
local i = 2 // start in period 2
local j = 1 // start with iteration 1
local qR = 1
local XR =     `xR'
local XU = 1 - `xR'
local WR = `qR'*`XR'*(1-`kappa')/`theta' // initial XR needs to be iteratively updated
local WU =      `XU'*(1-`kappa')/`theta'
while `j' > 0 {
        local XU = `theta'*`gamma'*`WU'/(`gamma'+(1-`gamma')*`qR'^(1-`sigma')) +
        + `theta'*(1-`gamma')*`WR'/(`gamma'*`qR'^(1-`sigma') + (1-`gamma'))
        local XU`i'_`j' = round(`XU',.000000001)
        if `j' > 1 {
                if `XU`i'_`=`j'-1'' == round(XU[1],.000000001) continue, break
        }
        local qR = XU[`i']/`XU'*`qR'
        local XR = `XR'/(`XU' + `qR'*`XR') // defined so that XU + qR*XR = 1
        local WR = `qR'*`XR'*(1-`kappa')/`theta'
        local WU =      `XU'*(1-`kappa')/`theta'
        local ++j
}
di as text"Iterative value for "as input"XR[`i']"as text" is "as result `XR' as text"."
di as text"Iterative value for "as input"XU[`i']"as text" is "as result `XU' as text"."
di as text"Iterative value for "as input"qR[`i']"as text" is "as result `qR' as text"."
di as text"Iterative value for "as input"WR[`i']"as text" is "as result `WR' as text"."
di as text"Iterative value for "as input"WU[`i']"as text" is "as result `WU' as text"."
qui replace qR = `qR' in 1/3
label variable qR "Real exchange rate"
qui replace qRdotqR = (qR[_n+1] - qR)/qR in 1/3
label variable qRdotqR "Change of real exchange rate"
qui replace XRo =    `XR' in 1/3 // R output (old) -- not adjusted for qR
label variable XRo "Output of Ro"
```

```
qui replace XRn =    0 in 1/3 // R output (new) -- not adjusted for qR
label variable XRn "Output of Rn"
qui replace XR  =    `XR' in 1/3
label variable XR "Output of R" // R output (old+new) -- not adjusted for qR
qui replace NU = XU in 1/3 // R trees in 0
label variable NU "Number of trees in U"
qui replace NR = XR in 1/3 // R trees in t
label variable NR "Number of trees in R"
local N0R = NR[3] // R trees in 0
qui replace ZU = XU/NU in 1/3 // Productivity of U trees (constant = 1)
label variable ZU "Productivity of trees in U"
qui replace ZR = XR/NR in 1/3 // Productivity of R trees (constant = 1)
label variable ZR "Productivity of trees in R"
qui replace X   =    XU + qR*XR in 1/3 // World output
label variable X "World output"
qui replace xU = XU / (XU + qR*XR) in 1/3
label variable xU "Output share of U"
qui replace xRo = qR*XRo / (XU + qR*XR) in 1/3
label variable xRo "Output share of Ro"
qui replace xRn = qR*XRn / (XU + qR*XR) in 1/3
label variable xRn "Output share of Rn"
qui replace xR = xRo + xRn in 1/3
label variable xR "Output share of R"

// ###############################################
// Wealth
qui replace WU = `WU' in 1/3
label variable WU "Asset demand of U"
qui replace WU_XU = WU / XU in 1/3
label variable WU_XU "Asset demand output share of U"
qui replace WR = `WR' in 1/3
label variable WR "Asset demand of R"
qui replace WR_XR = WR / (qR*XR) in 1/3
label variable WR_XR "Asset demand output share of R"
qui replace W_t = WR + WU in 1/3
label variable W_t "Asset demand overall"
qui replace W_V_eq = (1-`kappa')*(XU + qR*XR)/`theta' in 1/3
label variable W_V_eq "Asset market equilibrium"

// Asset value
qui replace VU = `delta' * XU  / (`r_aut' - `gz') in 1/3
label variable VU "Asset supply in U"
qui replace VU_XU = VU / XU in 1/3
label variable VU_XU "Asset supply output share in U"
qui replace vhatRo = . in 1/3
```

```
label variable vhatRo "Asset value of Ro trees over output"
qui replace VRo = `delta' *qR*XR  / (`r_aut' - `gz') in 1/3
label variable VRo "Asset supply by old trees Ro"
qui replace vRo = . in 1/3
label variable vRo "Asset value of Ro tree"
qui replace vRn = . in 1/3
label variable vRn "Asset value of Rn tree"
qui replace VRn = 0 in 1/3
label variable VRn "Asset supply by new trees Rn"
qui replace VR  = VRo + VRn in 1/3
label variable VR "Asset supply in R"
qui replace VR_XR = VR / (qR*XR) in 1/3
label variable VR_XR "Asset supply output share in R"
qui replace V_t = VU + VR in 1/3
label variable V_t "Asset supply overall"
qui replace W_V = W_t - V_t in 1/3
label variable W_V "Net asset share overall (=0)"

//Investment
qui replace IU = `kappa' * XU in 1/3 // investment in U
label variable IU "Investment in U"
qui replace IR = `kappa' *qR*XRn in 1/3 // investment in R
label variable IR "Investment in R"
qui replace P = `kappaP' *qR*XRn in 1/3 // costs of FDI
label variable P "Price of FDI"

// Prices and exchange rates
qui replace PU = (`gamma' + (1-`gamma')*qR^(1-`sigma'))^(1/(1-`sigma')) in 1/3
label variable PU "Price index of country U"
qui replace PR = (`gamma'*qR^(1-`sigma') + (1-`gamma'))^(1/(1-`sigma')) in 1/3
label variable PR "Price index of country R"
qui replace lambdaUR = PR/PU in 1/3
label variable lambdaUR "Real exchange rate between U and R"

// Cross-holdings of assets
qui replace aRU = `muiU'*WR[2]/VU[2] in 1/3
// qui replace aRU = `muiU'*`xR'/(1-`xR') in 1/3 // possibly wrong
label variable aRU "Share of U assets held by R"
qui replace aUR = `muiU' in 1/3 // since initial net asset position is assumed to be zero.
label variable aUR "Share of R assets held by U"
qui replace muRU = `muiU' in 1/3 // global portfolio share
label variable muRU "Global portfolio share of U assets in R's portfolio"
qui replace muUR = `muiU'*`xR'/(1-`xR') in 1/3 // global portfolio share
```

```
label variable muUR "Global portfolio share of R assets in U’s portfolio"

// National accounts
qui replace TBU = XU - ‘theta’ * WU - IU in 1/3
label variable TBU "Trade balance of U"
qui replace TBU_XU = TBU/XU in 1/3
label variable TBU_XU "Trade balance output share of U"
qui replace CAU = XU - ‘theta’ * WU - IU + r_t * (WU - VU) in 1/3
label variable CAU "Current account of U"
qui replace CAU_XU = CAU/XU in 1/3
label variable CAU_XU "Current account output share of U"
qui replace CAR = qR*XR - ‘theta’ * WR + r_t * (WR - VR) in 1/3
label variable CAR "Current account of R"
qui replace CAR_XR = CAR / (qR*XR) in 1/3
label variable CAR_XR "Current account output share of R"
qui replace NINV = r_t * (WU - VU) in 1/3
label variable NINV "Net investment income of U"
qui replace NINV_XU = NINV/XU in 1/3
label variable NINV_XU "Net investment income output share of U"
qui replace NAU = WU - VU in 1/3
label variable NAU "Net assets of U"
qui replace NAU_XU = NAU / XU in 1/3
label variable NAU_XU "Net assets output share of U"
qui replace ChiU = (‘gn’*NR*VRn/(qR*XR) - (‘kappaP’+‘kappa’)*qR*XRn - ‘kappa’*qR*XRo) / WU in ‘i’
label variable ChiU "Net investment income of U"
qui replace ChiU_XU = ChiU / XU in ‘i’
label variable ChiU_XU "Net investment income share of U"

// #############################################
// Shooting algorithm
// Definitions
// #############################################

qui replace w = WU / XU in 1/2
qui replace x = XU / (XU + qR*XR) in 1/2
qui replace r = ‘gz’ + ‘delta’*‘theta’/(1-‘kappa’) in 1/2 // = r_aut = .06
qui replace q = qR in 1/3
qui replace qdotq = (q-q[_n-1])/q[_n-1] in 1/3
qui replace lambda = PR/PU in 1/3
qui replace V0U = .
qui replace V0R = .
qui replace q3 = .
qui replace deltaR = .
qui replace r_vhatRo = r

```

```
// Guess for V0+
local V0R = ‘shock’*VR[2]
local V0U = VU[2] + (1-‘shock’)*VR[2] // yields V = (1-kappa)/theta X
qui replace V0U = ‘V0U’ in 1
qui replace V0R = ‘V0R’ in 1
local one = .

// ##############################################
// CFG: "We start with a guess for the asset values V0+i immediately after the shock."
local j = 0
while (round(V0U[‘j’]-V0U[‘j’-1],‘to_precision’) != 0 | round(V0U[‘j’]-V0U[‘j’-2],‘to_precision’) != 0 )  {
        local ++j
        if ‘j’ == ‘N’ + 1 {
                di as error"Loop number larger than number of observations."as text" Restart."
                continue, break
        }
        if "‘modvars’" != "" di _newline as result"Model variation for ‘modvargraph’ with "
        as input "‘modvaracr’=‘modvar’" as result"."
        di as result"Iteration no: "as input ‘j’ as result"."
        // CFG: "Given the initial portfolio allocation, we infer the initial wealth distribution W0+i."
        //              Problem to ensure that total wealth remains unchanged (W0+ = X0+/theta)
        qui replace w  = ( (1-aRU[3])*‘V0U’ + aUR[3]*‘V0R’ )/XU in 3
        di as text"Initial guesses are "as input"V0U = "as result ‘V0U’ as text" and "
        as input"V0R = "as result ‘V0R’ as text" at "as result" iteration  "as input ‘j’ as text"."
        di as text"Initial wealth uses "as input"aRU[0] = "as result aRU[3] as text" and "
        as input"aUR[0] = "as result aUR[3] as text"."
        di as text"Initial wealth is "as input"w[0+] = "as result w[3] as text
        " from initial portfolio distribution."
        di as text"Total wealth is "as input"W[0+] = "as result ‘=‘V0U’+‘V0R’’ as text
        " from initial portfolio distribution."

        // ##############################################
        // XR t=0+: Calculating value for q0+ for known wealth distribution and absolute output.
        // ##############################################
        if ‘j’ == 1 local q0 = q[2]
```

```
        if `j'  > 1 local q0 = q[3]
        local k = 1
        while `k' > 0 & round(`one',.0000001) != 1 {
                local qdotq = (`q0'-q[2])/q[2] // q2dot/q2
                local x = XU[2]/(`q0'*XR[2] + XU[2])  // for x[3
                    ] equation (41)/(62)
                local PU = ( `gamma'
                      + (1-`gamma')*`q0'^(1-`sigma') )^(1/(1-`
                    sigma'))
                local PR = ( `gamma'*`q0'^(1-`sigma') + (1-`
                    gamma')                             )^(1/(
                    1-`sigma'))
                if "`exrates'" != "off" { // adjustment via qR
                        local one = ( `theta'*`gamma'*w[3]*`PU'^
                            (`sigma'-1) ) +
                        ((1-`gamma')*((1-`kappa')/`x' - `theta'*
                            w[3])*`PR'^(`sigma'-1) )
                        local q0 = 1/`one'*`q0'
                }
                if "`exrates'" == "off" { // adjustment via w
                (indirectly via V0U appreciation, r_t decrease
                    as consequence not cause!)
                        if `k' >  1 local V0U = 1/`one'*`V0U'
                        local w = ( (1-aRU[3])*`V0U' + aUR[3]*`
                            V0R' )/XU
                        local one = ( `theta'*`gamma'*`w'*`PU'^(
                            `sigma'-1) ) +
                        + ((1-`gamma')*((1-`kappa')/`x' - `theta
                            '*`w')*`PR'^(`sigma'-1) )
                        di as text"Shooting value for "as input"
                            w[0+]"as text" is "as result `w' as
                            text
                        " yielding one = "as input `one' as text
                            "."
                }
                local ++k
        }
        di as text"Wealth in R is approximated as "as input"WR[
            0+]/XU[0+]"as text" as "
        as result `=(((1-aUR[3])*`V0R' + aRU[3]*`V0U' )/XU[3])'
            as text" (portfolio) or "
        as result `=(1-`kappa')/(`theta'*`x') - w[3]' as text "
            (W-WU)."
        di as text"Control value for "as input"one"as text" is "
            as result `one' as text"."
        di as text"Shooting value for "as input"qR[0+]"as text"
            is "as result `q0' as text
        " after "as input `=`k'-1' as text" iterations."
        qui replace x = `x' in 3
```

```
        if "`exrates'" == "off" {
                qui replace w = `w' in 3 // NEW adjustment via w/VU
                qui replace VU = `V0U' in 3 // NEW adjustment via w/VU
        }
        qui replace q  = `q0' in 3
        qui replace qdotq = (q[3]-q[2])/q[2] in 2
        qui replace qR = `q0' in 3
        qui replace qRdotqR = qdotq in 2
        qui replace q3 = `q0' in `j'

        // #############################################
        // deltaR: Updating value for deltaR in order to produce desired shock
        // #############################################

        if "`modvargraph'" != "deltaR" {
                local deltaR = `deltaR'*`shock'*VR[2]/VR[3]
        }
        qui replace deltaR = `deltaR' in `j'
        di as text"Shooting value for "as input"deltaR"as text" is "as result `deltaR' as text
        " with "as input" shock = "as result `shock' as text"."

        // #############################################
        // Condition on bilateral private gains from FDI
        // #############################################
        local upper_bound = round(`gn'*`delta' /(`r_aut'-`gz'),.001)
        local lower_bound = round(`gn'*`deltaR'/(`r_aut'-`gz'),.001)
        di as text"Condition on bilateral privat gains demands " as input"FDI costs"as text"
        are between " as result `upper_bound' as text" >= k+kP=" as input
        `=`kappa'+`kappaP'' as text" >= "as result `lower_bound' as text"."
        if "`FDI'" != "off" & "`Force_FDI'" != "on" { // if not already switched off by model default
                if              `upper_bound' >= `kappa' + `kappaP' & `kappa' + `kappaP' >= `lower_bound' {
                        local FDI = "on"
                }
                if `upper_bound' < `kappa' + `kappaP' | `kappa' + `kappaP' < `lower_bound' | `upper_bound' == `lower_bound' {
                        local FDI = "off"
```

```
                }
        }
        di as text"Bilateral private gains set "as input"FDI"as text" are " as result "`FDI'" as text"."

        // ############################################
        // FDI: Solving r_t and vRo reversely
        // ############################################

//      // Value of qR to use as starting value (first run without XR change).
        if `j' == 1 {
                qui replace xR  = xR[2]
                qui replace xRo = xRo[2]
                qui replace qRdotqR = qRdotqR[1] in 3/`N'
                qui replace r_vhatRo = `r_aut' in 3/`N'
        }
        if `j'  > 1 {
                replace qRdotqR = qRdotqR[`N'-1] in `N'
        }
        forvalues i= 3/`N' {
                // reverse counting of `i': sequence changed from 3/N to N/3
                local n = `i'
                local i = `N' - `i' + 3
//              // Update using new exchange rate values
                qui replace vhatRo = (`deltaR'/`delta') * ((1-`kappa')/`theta') in `i' if `i' == `N' // was (1/`theta')
                if "`FDI'" != "off" {
//                      qui replace xRo = xRo[`i']*(1 + (`gR'-`g') + `gn' - `xRoqR'*`qRdotqR') in `=`i'-1' if `i' > 3
                        if "`exrates'" != "off" qui replace r_vhatRo   = `gz' + `theta'/(1-`kappa')*
                        (`delta' - (`delta'-`deltaR')*xRo - `gn'*vhatRo*xRo*(`delta'/`deltaR'-1))
                        + xR*qRdotqR[`i'] in `i'
                        if "`exrates'" == "off" qui replace r_vhatRo   = `gz' + `theta'/(1-`kappa')*
                        (`delta' - (`delta'-`deltaR')*xRo - `gn'*vhatRo*xRo*(`delta'/`deltaR'-1)) in `i'
                }
                if "`FDI'" == "off" {
                        if "`exrates'" != "off" qui replace r_vhatRo   = `gz' + `theta'/
```

```
                    (1-`kappa')*(`delta' - (`delta'-`deltaR')*xRo) in `i'
                    if "`exrates'" == "off" qui replace r_vhatRo   = `gz' +
                    `theta'/(1-`kappa')*(`delta' - (`delta'-`deltaR')*xRo) in `i'
                }
                qui replace vhatRo = (vhatRo[`i'] + `deltaR')/(1 + r_vhatRo[`i'] - `gz') in `='i'-1' if `i' > 3 // transitory
                qui replace vhatU  = `delta'/(r_vhatRo - `gz') in `n' if `n' == 3 // initial value
                // // sets V0+/X = (1-kappa)/theta
                qui replace vhatU  = (vhatU[`n'-1] + `delta')/(1 + r_vhatRo[`n'-1] - `gz') in `n' if `n' > 3 // transitory
                qui replace vhatRn = `delta'/(r_vhatRo - `gz') in `n' if `n' == 3 // initial value as for vhatU
                qui replace vhatRn = (vhatRn[`n'-1] + `delta')/(1 + r_vhatRo[`n'-1] - `gz') in `n' if `n' > 3 // transitory
        }
        local vhatRo`k' = vhatRo[3]
        local  vhatU`k' = vhatU[3]
        di as text"New estimate for "as input"vhatRo[0+]"as text" is "as result
         `vhatRo`='k''' as text" for iteration no. "as input `k' as text"."
        di as text"New estimate for "as input"vhatU[0+]"as text" is "as result
        `vhatU`='k''' as text" for iteration no. "as input `k' as text"."
        local VRo`k' = vhatRo[3]*qR[3]*XRo[3]
        local VU`k' = vhatU[3]*XU[3]
        di as text"New estimate for "as input"VRo[0+]"as text" is "as result `='VRo`k''' as text"."
        di as text"New estimate for "as input"VU[0+]"as text" is "as result `='VU`k''' as text"."
        local V`k' = `='VRo`k''' + `='VU`k'''
        di as text"New estimate for "as input"V[0+]"as text" is "as result `='V`k''' as text"."

        pause

        // #############################################
        // t>0+: Calculating shooting algorithm equation (39) to (42)
        // #############################################
```

```
        forvalues i = 3/`N' {
                di _newline as result"Calculating model
                    variation for: "as input"`modvargraph'"
                as result"="as input"`modvar'"as result"."
                di                      as result"Calculating
                    shooting values in period: "as input"`i'"as
                    result"/"as input"`N'"as result"."
                // Finding the starting value for q0+ by
                    reiteration - first guess for q is preceding
                     period's value.
                local q0 = q[`i']
                local qdotq = (`q0'-q[`i'])/q[`i']
                // di as text"Calculating shooting value for "as
                     input "q[i+1]" as text
                " with a starting value of "as result `q0' as
                    text":" //, _continue

                // CFG: "We then use (40) to solve for the
                    initial terms of trade q0+."
                local count = 1
                local one = .
                // q0+: While-loop runs until XU/XU = 1 which is
                     obtained when q0 converges.
                while round(`one',.00001) != 1 { // was .000001
                        // equation (40) -- need to update x_t
                            accordingly since it incorporates x
                            = XU/sum(qi*Xi)
                        if `j' == 1 local x = x[`i'] // *(1 + (
                            1-x[`i'])*(`g' - `gR' - `qdotq')) //
                             equation (41)/(62)
                        if `j'  > 1 local x = xU[`i']
                        if "`FDI'" != "off" {
                                local xRo = xRo[`i'] // *(1 + x[
                                    `i']*(`gR' - `g' - `gn' + `
                                    qdotq'))
                                local r = r_vhatRo[`i'] //
                                local w = w[`i']*(1 + `r' - `
                                    theta' - `g') + (1 - `delta'
                                     - `kappa'/`x') +
                                 `gn'*((1-`kappa')/(`theta'*`x')
                                      + vhatRo[`i']*`xRo'/`x'*(`
                                     delta'/`deltaR' -1)) -
                                 `kappaP'*(1-`x'-`xRo')/`x' //
                                     equation (39)/60)
                        }
                        if "`FDI'" == "off" {
                                if "`exrates'" != "off" {
                                        local r = r_vhatRo[`i']
                                }
```

```
                                if "`exrates'" == "off" { // rebalancing channel (t=0+) wealth/asset values or (t=1/N) real interest rates
//                                      if `count' == 1 local r = r_vhatRo[`i']
//                              if `count' >  1 local r = 1/`one'*`r'
                                }
                                local w = w[`i']*(1 + `r' - `theta' - `gz') + (1 - `delta')  // equation (39)/(60)
                        }
                        local PU = ( `gamma'
                                        + (1-`gamma')*`q0'^(1-`sigma') )^(1/(1-`sigma'))
                        local PR = ( `gamma'*`q0'^(1-`sigma') + (1-`gamma')
                                        )^(1/(1-`sigma'))
                        local lambda = `PR'/`PU'
                        if `i' <  `N' local one = ( `theta'*`gamma'*`w'*`PU'^(`sigma'-1) ) +
                        ((1-`gamma')*((1-`kappa')/`x' - `theta'*`w')*`PR'^(`sigma'-1) )
                        if `i' == `N' local one = 1
                        local one = `one'
                        local one`count' = round(`one',.000001)
                        if "`exrates'" != "off" {
                                // Larger decrease (increase) of q0 the further `one' is above (below) 1.
                                local q`count' = `q0'
                                local q0 = 1/`one'*`q0'
                        }
                        if "`exrates'" == "off" {
                                local q`count' = q[`i']
                                local q0 = `q0'
                        }
                        local qdotq = (`q0'-q[`i'])/q[`i']
                        // di as text"Restart iterative process after"as input" iteration "as result `count' as text"." _newline
                        local ++count
                        if `count' == 60 {
                                di as error"Iteration has reached "as input"60" as error" steps. "
                                as input"one=`one'" as error" Stop and break loop here."
```

```
                    pause
                    continue, break
            }
    }
    di as text"Shooting value for "as input"w[`i']"
        as text" is "as result `w'
    as text" yielding one = "as input `one' as text
        "."
            local one = .
    di as text"Shooting value for "as input"q[`=`i'
        +1']"as text" is "as result"`q0'"
    as text" after "as result"`count'"as text" runs
        ."
    if `i' < `N' qui replace q = `q0' in `=`i'+1'
    if `i' <  `N' qui replace qdotq = `qdotq' in `i'
    di as text"Shooting value for "as input "qdotq[`
        i']" as text" is "as result"`qdotq'"
    as text" after "as result"`count'"as text" runs
        ."

    // ############################################
    // CFG: "Finally, we integrate (39)/(60)-(42)/(
        63) forward to construct the
    // path of future interest rates and terms of
        trade, rt, qt
    //            consistent with equilibrium in
        the goods markets."
    // ############################################
    if `i' < `N' qui replace x = `x' in `=`i'+1' //
        same as above: x[`i']*(1 +
    (1-x[`i'])*(`g' - `gR' - qdotq[`i']))
    qui replace r = `r' in `i' //x*(`gz' + `delta'*`
        theta') + (1-x)*
    (`gz' + qdotq + `deltaR'*`theta') in `i' //
        equation (42)/(63)
    if `i' < `N' qui replace w = `w'  in `=`i'+1' //
         w[`i']*(1 + r[`i'] - `theta') +
    (1 - `delta') in `=`i'+1' // same as above:
    qui replace lambda = `PR'/`PU' in `i'
    di as text"Final shooting value for "as input"x[
        `i']"as text" is "as result x[`i'] as text
        "."
    di as text"Final shooting value for "as input"x[
        `=`i'+1']"as text" is "as result x[`i'+1] as
         text"."
    di as text"Final shooting value for "as input"r[
        `i']"as text" is "as result r[`i'] as text
        "."
```

```
			di as text"Final shooting value for "as input"w[`i']"as text" is "as result w[`i'] as text"."
			di as text"Final shooting value for "as input"w[`=`i'+1']"as text" is "as result w[`i'+1] as text"."
			di as text"Final shooting value for "as input"lambda[`i']"as text" is "as result lambda[`i'] as text"."
			di as text"Final shooting value for "as input"q[`i']"as text" is "as result q[`i'] as text"."
			di as text"Final shooting value for "as input"q[`=`i'+1']"as text" is "as result q[`i'+1] as text"."
			di as text"Final shooting value for "as input"qdotq[`i']"as text" is "as result qdotq[`i'] as text"."

//			pause

			// ###############################################
			// Calculating all other values from shooting mechanism.
			quietly {
				replace qR = q in `i'
				//replace r_t = r in `i'

				// Real economy (for next period)
				if `i'> 3 {
					replace NU  = (1+`gn')*NU[`i'-1] in `i'
					replace NR  = (1+`gR')*NR[`i'-1] in `i' // gR to incorporate possibly faster growth
					replace ZU  = (1+`gz')*ZU[`i'-1] in `i'
					replace ZR  = (1+`gz')*ZR[`i'-1] in `i'
					replace XU  = NU[`i'] * ZU[`i'] in `i' // normal growth rate
					if "`FDI'" != "off" {
						replace XRo = `N0R' * ZR[`i'] in `i' // no growth
						replace XRn = (NR[`i'] - `N0R')*ZR[`i'] in `
```

```
                        i’ // normal growth rate
                }
                if "‘FDI’" == "off" {
                        replace XRo = NR[‘i’]*ZR[‘i’] in ‘i’ // normal growth rate
                        replace XRn = 0 in ‘i’ // no growth
                }
                replace XR  = XRo[‘i’] + XRn[‘i’] in ‘i’ // normal growth rate
        }
        replace PU = ‘PU’ in ‘i’
        replace PR = ‘PR’ in ‘i’
        replace qRdotqR = qdotq[‘i’] in ‘i’
        replace lambdaUR = PR/PU in ‘i’
        replace xU =    XU / (XU + qR*XR) in ‘i’
        replace X   = XU + qR*XR in ‘i’
        replace xRo = qR*XRo / (XU + qR*XR) in ‘i’ // decreasing
        replace xRn = qR*XRn / (XU + qR*XR) in ‘i’ // increases
        replace xR  = qR*XR  / (XU + qR*XR) in ‘i’ // time-varying
        replace r_t = r in ‘i’ // value after derivation: used to be r_vhatRo
        // replace r_t = ‘gz’ + qRdotqR*(xRn+xRo) + ‘theta’/(1-‘kappa’)*(‘delta’*(xU+xRn) + ‘deltaR’*xRo - ‘gn’*(‘N0R’*vhatRo*(‘delta’/‘deltaR’-1))/X) in ‘i’ // r[‘i’] in ‘i’
        // replace r_t = (‘g’ + ‘delta’*‘theta’) + xR*(qRdotqR - ‘theta’*(‘delta’-‘deltaR’)) in ‘i’
        // replace r_t = ‘g’*xU + ‘delta’*‘theta’/(1-‘kappa’) + xR* (‘gR’ + qRdotqR - ‘theta’*(‘delta’ - ‘deltaR’)) in ‘i’ // shock at t=0
        // replace r_t   = ‘gz’ + ‘theta’/(1-‘kappa’)*(‘delta’*(1-xRo) + ‘deltaR’*xRo - ‘gn’*vRo*xRo*(‘delta’/‘deltaR’-1)) in ‘i’

        // Asset supply (asset value)
        // Shooting values for V0+i from above derivation already in ‘i’ == 3
```

```
// Using portfolio home-bias to determine wealth in t=0+.
replace WU = w*XU in `i' if `i' == 3 // (1-aRU[3])*`V0U' + aUR[3]*`V0R' in `i' if `i' == 3 // using shooting values for t=0+
replace WR = (1-`kappa')/(`theta'*X) - w*XU in `i' if `i' == 3 // (1-aUR[3])*`V0R' + aRU[3]*`V0U' in `i' if `i' == 3 // using shooting values for t=0+

// Investment and FDI costs after shock (for current period)
replace IU  = `kappa' * XU[`i']  in `i' if `i' > 3
replace IR  = `kappa' *qR[`i']* XRn[`i']  in `i' if `i' > 3
replace P   = `kappaP'*qR[`i']* XRn[`i']  in `i' if `i' > 3

// Asset supply (asset value)
// Shooting values for V0+i from above derivation already in `i' == 3
replace vRo = vhatRo*qR*XRo/`N0R' in `i'  //`deltaR'*ZR / (r_t - `gz') in `i'
replace VRo = vhatRo*qR*XRo in `i'
if "`FDI'" != "off" {
        replace vRn = `delta'*ZR / (r_t - `gz') in `i' // vRo*`delta'/`deltaR' in `i' //        // alternatively: vRn = vRo*ZR*`delta'/`deltaR'
        replace VRn = (NR - `N0R')*vRn in `i' if `i' == 3      // transitory // alternatively: VRn = `delta'*(NR/`N0R' - 1)*XR[`i'-1]
}
if "`FDI'" == "off" {
        replace vRn = 0 in `i' //`delta'*ZR / (r_t - `gz') in `i'          // alternatively: vRn = vRo*ZR*`delta'/`deltaR'
        replace VRn = 0  in `i' if `i' == 3     // transitory //
```

```
                                        alternatively: VRn = `delta' *(NR/`N0R' - 1)*XR[`i'-1]
                        }
                        replace VRn = vhatRn*qR*XRn in `i' if `i' > 3
//                      replace VRn = (1 + r_t[`i'-1])*VRn[`i'-1] -`delta' *qR[`i'-1]*XRn[`i'-1] +`gn'*NR[`i'-1]*vRn[`i'-1] in `i' if `i' > 3 // NR[`i'-1]*vRn[`i'-1] =? (1-`N0R'/(NR[`i'-1]-`N0R'))*VRn[`i'-1]
                        replace VR = VRo + VRn  in `i' // transitory
                        replace VR_XR = VR/(qR*XR) in `i'
//                      replace VU = (1-`kappa')*X/`theta' - VR in `i'
                        replace VU = vhatU*XU in `i'
//                      replace VU = (1 + r_t[`i'-1] + `g'-`gz')*VU[`i'-1] -`delta'*XU[`i'-1]  in `i' if `i' > 3 // transitory
                        replace VU_XU = VU / XU in `i'
                        replace V_t = VR + VU in `i' // if `i' <= 4

                        // Asset demand (Wealth)
                        // Shooting values for W0+i from portfolio home-bias using above derivation in `i' == 3
//                      local w = w[`i']*(1 + `r' - `theta' - `g') +            (1 - `delta') +                 `gn'*((1-`kappa')/(`theta'*`x') - vhatRo[`i']*`xRo'/`x') - `kappaP'*(1-`x'-`xRo')/`x' - `kappa'*(1-`xRo')/`x' // equation (39)/60) ??? BUT wrong wealth equation!!! (p. 49)
//                      replace WU = (1 + r_t[`i'-1] - `theta')*WU[`i'-1] + (1-`delta')*XU[`i'-1] + `gn'*(VU[`i'-1] + NR[`i'-1]*vRn[`i'-1]) - (IU[`i'-1]+IR[`i'-1]) - P[`i'-1] in `i' if `i' > 3
//      not like w              replace WU = (1 + r_t[`i'-1] - `theta')*WU[`i'-1] + (1-`delta')*XU[`i'-1] + `gn'*(VU[`i'-1] + NR[`i'-1]*vRn[`i'-1]) - (IU[`i'-1]+IR[`i'-1]) - P[`i'-1] in `i' if `i' > 3
//                      replace WU_XU = WU / XU in `i'
                        replace WU = w*XU in `i' if `i' > 3
                        replace WU_XU = w in `i'
//                      replace WR = (1-`kappa')*X/`theta' - WU in `i'
                        if "`FDI'" != "off" replace WR = (1 + r_t[`i'-1] - `theta')*WR[`i'-1] + (1-`delta')*qR[`i'-1]*XRn[`i'-1] + (1-`deltaR')*qR[`i'-1]*XRo[`i'-1] + P
```

```
                        [`i'-1] in `i' if `i' > 3 // transitory -- from p. 45
                if "`FDI'" == "off" replace WR = (1 + r_t[`i'-1] - `theta' + `gR'-`gz')*WR[`i'-1] + (1-`deltaR')*qR[`i'-1]*XRo[`i'-1] in `i' if `i' > 3 // transitory -- from p. 45
                replace WR_XR = WR / (qR*XR) in `i'
                replace W_t = WR + WU in `i'

                // Open economy
                replace W_V = W_t - V_t in `i'
                replace W_V_eq = (1-`kappa')/`theta'*(XU + qR*XR) in `i'

                if "`FDI'" != "off" replace TBU =     XU - `theta'*WU - IU  + `gn'*VR - (`kappa' + `kappaP')*qR*XR in `i' // transitory
                if "`FDI'" == "off" replace TBU =     XU - `theta'*WU - IU in `i' // transitory  ( + `gn'*VR - (`kappa' + `kappaP')*qR*XR)
                replace TBU_XU = TBU/XU in `i' // transitory
                replace NAU = aUR*VR - aRU*VU in `i' // WU - VU in `i' //
//              replace NAU = WU - VU in `i' // aUR*VR - aRU*VU in `i' //
                replace NAU_XU = NAU / XU in `i'
                replace CAU = TBU + r_t*(WU-VU) in `i' // transitory // or: r_t*(WU-VU)
//              replace CAU = TBU + r_t*NAU in `i' // transitory // or: r_t*(WU-VU)
//              replace CAU = TBU + r_t*(aUR*VR - aRU*VU) in `i'
//          replace CAU = (w[`i'+1]*XU - WU) - (vhatU[`i'+1]*XU[`i'+1] - VU) in `i' // CAU = \dot{W} - \dot{V}
//              replace CAU = (WU[`i'+1] - WU) - (VU[`i'+1] - VU) in `i' // CAU = \dot{W} - \dot{V}
                replace CAU_XU = CAU/XU in `i'
                replace CAR = qR*XR - `theta'*WR - IR + r_t*(aRU*VU - aUR*VR) in `i'
                replace CAR_XR = CAR/(qR*XR) in `i'
                replace NINV = r_t * (WU - VU) in `i'
                replace NINV_XU = NINV/XU in `i'

//              // OLD: Mechanism for net foreign asset holdings
```

```
//                      if `i' < `N' replace aRU  = aRU[`i'] - CAU[`i']/VU[`i'] in `=`i'+1' // from p. 57
//                      if `i' < `N' replace aUR  = aUR[`i'] + CAU[`i']/VR[`i'] in `=`i'+1' // from p. 57

                        // NEW: Mechanism for net foreign asset holdings
                        // if CA DEFICIT of U in `i'
                        if CAU[`i'] < 0 & `i' < `N' {
                                local aUR_diff  = aUR[`i'] + CAU[`i']/VR[`i'] // existing asset sufficient to cover deficit?
                                if `aUR_diff'  >= 0 { // U's existing holdings sufficient
                                        replace aUR = aUR[`i'] + CAU[`i']/VR[`i']          in `=`i'+1' // aUR > 0
                                        replace aRU = aRU[`i']          in `=`i'+1' // aRU = aRU[`i']
                                }
                                if `aUR_diff' <  0 { // U's existing holdings insufficient (`aUR_diff' -ve)
                                        replace aUR = aUR[`i'] + CAU[`i']/VR[`i'] - `aUR_diff' in `=`i'+1' // aUR = 0
                                        replace aRU = aRU[`i']          - `aUR_diff'*VR[`i']/VU[`i'] in `=`i'+1' // aRU > aRU[`i']
                                }
                        }
                        // if CA SURPLUS of U in `i'
                        if CAU[`i'] > 0 & `i' < `N' {
                                local aRU_diff  = aRU[`i'] - CAU[`i']/VU[`i'] // existing asset sufficient to cover deficit?
```

```
                        if ‘aRU_diff’  >= 0 { // R’s existing holdings sufficient
                            replace aRU = aRU[‘i’] - CAU[‘i’]/VU[‘i’] in ‘=‘i’+1’ // aUR > 0
                            replace aUR = aUR[‘i’] in ‘=‘i’+1’ // aRU = aRU[‘i’]
                        }
                        if ‘aRU_diff’ <  0 { // R’s existing holdings insufficient (‘aRU_diff’ -ve)
                            replace aRU = aRU[‘i’] - CAU[‘i’]/VU[‘i’] - ‘aRU_diff’ in ‘=‘i’+1’ // aUR = 0
                            replace aUR = aUR[‘i’] - ‘aRU_diff’*VU[‘i’]/VR[‘i’] in ‘=‘i’+1’ // aUR > aUR[‘i’]
                        }
                }

//              MECHANISM for net assets as W-V.
//              if ‘i’ < ‘N’ replace aRU  = (WU-VU)/VU[‘i’] in ‘=‘i’+1’ // from p. 57
//              if ‘i’ < ‘N’ replace aUR  = (WR-VR)/VR[‘i’] in ‘=‘i’+1’ // from p. 57

//              Net international position
//              replace muRU = (aRU*VU - aUR*VR) / WR in ‘i’
//              replace muUR = (aUR*VR - aRU*VU) / WU in ‘i’

//              Gross international position
                replace muRU = aRU * VU / WR in ‘i’
                replace muUR = aUR * VR / WU in ‘i’
```

```
                replace ChiU = (`gn'*NR*vRn - (`kappaP'+`kappa')*qR*XRn - `kappa'*qR*XRo) / WU in `i'
                replace ChiU_XU = ChiU / XU in `i'
        }
}


// ##############################################
// CFG: "We then use V0+i to update our guess for Vo+i, where delta_bar is the average (time-varying) capitalization ratio."
di _newline as result"Updating shooting value for "as input"V0+"as result" in "as input"U"as result" and "as input"R"as result"."

// u creates a range for the integral from 0 to u = N-2
forvalues max = `N'/`N'{
        //local max = 200
        qui gen u = _n - 3 if _n >= 3 & _n <= `max'

        // Second equation V0+i = q0i*X0i*deltai
        qui gen dbar = (xU+xRn)*`delta' + xRo*`deltaR' if tin(0,`max')
        qui integ dbar u, gen(int_dbar)
        qui gen exp_dbarU = exp(-`theta'*int_dbar)*xU/xU[3]
        qui integ exp_dbarU u, gen(int_exp_dbarU)
        qui gen exp_dbarRo = exp(-`theta'*int_dbar)*xR/xR[3]
        qui integ exp_dbarR u, gen(int_exp_dbarR)
        qui gen exp_dbar = exp(-`theta'*int_dbar)*X/X[3]
        qui integ exp_dbar u, gen(int_exp_dbar)
        local V0U =           XU[3]*`delta' * int_exp_dbarU[`max']
        local V0R = qR[3]*(XRo[3]*`deltaR' + XRn[3]*`delta')*int_exp_dbarR[`max']
        local V0 = (XU[3] + qR[3]*(XRo[3]*`deltaR' + XRn[3]*`delta'))*int_exp_dbar

        // First equation V0+i = deltai
        qui integ r_t u, gen(int_r)
        qui gen exp_intU  =     XU*exp(-int_r)
        qui gen exp_intRo = qR*XRo*exp(-int_r)
        qui gen exp_intRn = qR*XRn*exp(-int_r)
        qui gen exp_int   =      X*exp(-int_r)
        qui integ exp_intU  u, gen(int_exp_intU)
        qui integ exp_intRo u, gen(int_exp_intRo)
        qui integ exp_intRn u, gen(int_exp_intRn)
```

```
                qui integ exp_int   u, gen(int_exp_int)
//              qui gen V0Ro = ‘deltaR’*int_exp_intRo[‘=‘max’-1’]
//              qui gen V0Rn = ‘delta’*int_exp_intRn[‘=‘max’-1’]
//              qui gen V0U2 = ‘delta’*int_exp_intU[‘=‘max’-1’]
                local V0Ro = ‘deltaR’*int_exp_intRo[‘=‘max’-1’]
                local V0Rn = ‘delta’ *int_exp_intRn[‘=‘max’-1’]
//              local V0U  = ‘delta’ *int_exp_intU[‘=‘max’-1’] // not working !
//              local V0R  = ‘=qR[3]*XRo[3]*vhatRo[3]’ //‘V0Ro’ + ‘V0Rn’
//              local V0  = (‘delta’*(xU+xRn)+‘deltaR’*xRo)*int_exp_int[‘=‘max’-1’]

                // Update existing values and restart loop.
                qui replace V0U = ‘V0U’ in ‘j’
                di as text"Shooting value for "as input"V0+U"as text" is "as result round(‘V0U’,.001) as text" in iteration no: "as input"‘j’"as text"."

                qui replace V0R = ‘V0R’ in ‘j’
                di as text"Shooting value for "as input"V0+R"as text" is "as result round(‘V0R’,.001) as text" in iteration no: "as input"‘j’"as text"."
                di as text"Shooting value for "as input"V0+"as text" is "as result round(‘V0’,.001) as text" in iteration no: "as input"‘j’"as text"."
                //browse u dbar exp* int* V0*
                di as text"Decline in "as input"VR"as text" is "as result round(‘=(VR[2]-‘V0R’)/VR[2]’*100,.001) as text" % with " as input"deltaR = "as result ‘deltaR’ as text"."
                di _newline
                di as text"Shooting value for "as input"V0+R"as text" is "as result round(‘V0Ro’,.001) as text" in iteration no: "as input"V0+R"as text"."
                di as text"Shooting value for "as input"V0+R"as text" is "as result round(‘=‘N0R’*vRo[3]’,.001) as text" in iteration no: "as input"N0R*vRo"as text"."
                di as text"Shooting value for "as input"V0+R"as text" is "as result round(‘=qR[3]*XRo[3]*vhatRo[3]’,.001) as text" in iteration no: "as input"qR*XRo*vhatRo"as text"."
                qui replace VRo = ‘V0R’ in 3

```

```
                //browse u dbar exp* int* V0* X*

                drop u dbar exp* int*
        }

        if `j' > 1      {
                local dV0U = V0U[`j'] - V0U[`j'-1]
                local dV0R = V0R[`j'] - V0R[`j'-1]
        }
        else    {
                local dV0U = V0U[`j']
                local dV0R = V0R[`j']
        }
        di _newline as result"Change in "as input"V0U"as result"
             is "as input `dV0U' as result"."
        di          as result"Change in "as input"V0R"as result"
             is "as input `dV0R' as result"."
        di          as result"Present value global asset values
            sum to "as input"V0+ = W0+ = "as result"  "as input
            `=`V0U'+`V0R'' as result"."

//      qui replace VU = `V0U' in 3  // shooting value
        //qui replace VR = `V0R' in 3 // shooting value

        //qui replace V0U = VU[3] in `j'  // shooting value
        //qui replace V0R = VR[3] in `j' // shooting value
        //pause
}
di as text"End shooting algorithm after iteration no: "as input"
    `j'"as text"."
if "`modvars'" != "" di _newline as result"End model variation
    for `modvargraph' with "as input "`modvaracr'=`modvar'" as
    result"."

// END: while-loop for improving estimate of V0+i.

// ##############################################
// Generate final asymptotic values for joint model
local max = min(100,`N'-1)
qui replace VU_XU_asympt = `delta' /(r[`max']-`gz')
qui replace VR_XR_asympt = (xRn[`max']/(1-xU[`max'])*`delta'+xRo
    [`max']/(1-xU[`max'])*`deltaR')/(r[`max']-`gz')
qui replace WU_XU_asympt = (1-`delta' - `kappa' + `gn'*
    VU_XU_asympt + `gn'*NR[`max']*vRn[`max']/XU[`max'] - `kappaP
    '*xRn[`max']/xU[`max'])/(`theta'+`gz'-r[`max'])
qui replace WR_XR_asympt = (1-`deltaR'*xRo[`max'] - `delta'*xRn[
    `max']
            + `kappaP'*xRn[`max']/xU[`max'])/(`theta'+`gz'-r[`
    max'])
```

```
qui replace TB_XU_asympt = (1 - `kappa') - `theta'*(1 - `delta' - `kappa' + `gn'*(VU[`max'] + NR[`max']*vRn[`max'])/XU[`max'] - `kappaP'*xRn[`max']/xU[`max'])/(`theta'+`gz' -r[`max'])
qui replace CAU_XU_asympt = `g' *( (1-`delta' - `kappa'  + `gn'*VU_XU_asympt + `gn'*NR[`max']*vRn[`max'] - `kappaP'*xRn[`max']/xU[`max'])/(`theta'+`gz'-r[`max']) - `delta' /(r[`max']-`gz' ) )
qui replace CAR_XR_asympt = `gR'*( (1-`deltaR'*xRo[`max'] - `delta'*xRn[`max']
          + `kappaP'*xRn[`max']/xU[`max'])/(`theta'+`gz'-r[`max']) - (`deltaR'*xRo[`max'] - `delta'*xRn[`max'])/(r[`max']-`gz') )

// if wordcount("`modvars'") <= 1 { // only create variables on first run-through.
//      // Generate growth variables.
//      foreach var in WU VU WR VR w W_V TBU NAU CAU {
//              qui gen g`var' = (`var'[_n]-`var'[_n-1])/`var'[_n-1]
//      }
// }
//browse WU_* WR_* VU_* VR_* TB_* CAU_*

// browse
//browse year XU XR r_t qR P* lambdaUR VU* VR* V_t WU* WR* W_t CA* NA* IIP*

if "`modvargraph'" == "deltaR" {
        local modvarlabel = "`=round(`=(VR[3]-VR[2])/VR[2]'*100,1)'%" //"{bf:{&kappa}{subscript:P}}=`modvar'"
        local modvarlegend "label(`=wordcount("`modvars'")' `modvarlabel') "
}

// ##############################################
// Generate graphs
di _newline as result"Generating output graphs."
local vars "CAU_XU NAU_XU TBU_XU r_t lambdaUR muRU" /*VR ChiU_XU aRU*/
foreach var in `vars' {
        local initial = `var'[2]
        local postshock = `var'[3]
        local asympt  = `var'[`max']
        local yline = "yline(`initial', lpattern(dash) lcolor(gs8))" //  `asympt'
        di as text"Initial, post-shock and asymptotic values for "as input "`var'" as text" are "as result `initial' as text ", "as result `postshock' as text " and " as result `asympt' as text " respectively."
```

```
		if "`var'" == "CAU_XU"  local options "yscale(r(-.10 .05)) title(Current Account/Output)"
		if "`var'" == "CAU2_XU" local options "yscale(r(-.10 .05)) title(Current Account/Output)"
		if "`var'" == "NAU_XU"  local options "yscale(r(.10 -.5)) title(Net Assets/Output)"
		if "`var'" == "r_t"     local options "yscale(r(.05 .065)) title(Real Interest Rate)"
		if "`var'" == "TBU_XU"  local options "yscale(r(-.1 .05)) title(Trade Balance/Output)"
		if "`var'" == "muRU"    local options "yscale(r(0 .5)) title(Global Portfolio Share)"
		if "`var'" == "aRU"     local options "yscale(r(0 .5)) title(Foreign Asset Share)"
		if "`var'" == "lambdaUR" local options "yscale(r(1.0 1.2)) title(Real Exchange Rate)"
		if "`var'" == "VR"              local options "yscale(log) title(Post-shock asset value)"
		if "`var'" == "ChiU_XU" local options "title(Net Investment Income)"
		tsline `var' if tin(-2,26), `options' ylabel(#7, angle(0)) `yline' ytitle("") ///
			name(`var', replace) lwidth(medthick ...) nodraw

		// Creating new variables for model variations
		if "`modvars'" != "" {
			qui gen `var'_`modvargraph'`modvarname' = `var'
			label variable `var'_`modvargraph'`modvarname' "`modvarlabel'"
			if "`graph_normalisation_lambda'" == "on" {
				// Normalisation to lambda0- = 100
				if "`var'" == "lambdaUR" {
					rename `var'_`modvargraph'`modvarname'   `var'2_`modvargraph'`modvarname'
					gen    `var'_`modvargraph'`modvarname' = `var'2_`modvargraph'`modvarname'/`var'2_`modvargraph'`modvarname'[2]*100
				}
			}
			if "`graph_normalisation_r_t'" == "on" {
				// Normalisation to r_0- = 6%
				if "`var'" == "r_t" {
					rename `var'_`modvargraph'`modvarname'   `var'2_`modvargraph'`modvarname'
```

```
                                gen `var'_`modvargraph'`modvarname' = `var'2_`modvargraph'`modvarname'/`var'2_`modvargraph'`modvarname'[2]*.06
                        }
                }
        }
}
di as result"Output graphs successfully created."
if "`modvars'" == "" { // only draw if only one model variation.
        graph combine `vars' ///
                , note("Parameter values: deltaR=`=round(`deltaR',.01)', kappa=`kappa', kappaP=`kappaP', sigma=`sigma', shock=`=round(`=(VR[2]-`V0R')/VR[2]'*100,.001)'%")

        graph save    ~/diss/output/CFG/`model'/figures/`model'_baseline.gph, replace
        graph export  ~/diss/output/CFG/`model'/figures/`model'_baseline.png, replace
        graph export "~/diss/output/CFG/`model'/figures/`model'_baseline (`=c(current_date)').png", replace
}

// END model variations of each submodel.
di _newline as text"End model variation of submodel "as input"`submodel'" as text"."
}

// Generate graphs for comparison if more than one model variation was tested.
if "`modvars'" != "" {
        di _newline as result"Generating output graphs for model variations."
        foreach var in `vars' {
                local initial = `var'[2]
                local asympt  = `var'[`max']
                local yline = "yline(`initial', lpattern(dash) lcolor(gs8))" // `asympt'
                local label `: var label `var''
                if "`var'" == "CAU_XU"  local options "yscale(r(-.10 .05)) title(Current Account/Output)"
                if "`var'" == "NAU_XU"  local options "yscale(r(.10 -.5)) title(Net Assets/Output)"
                if "`var'" == "r_t"     local options "yscale(r(.05 .065)) title(Real Interest Rate)"
                if "`var'" == "TBU_XU"  local options "yscale(r(-.1 .05)) title(Trade Balance/Output)"
```

```
			if "`var'" == "muRU"    local options "yscale(r(0 .5)) title(Global Portfolio Share)"
			if "`var'" == "aRU"     local options "yscale(r(0 .5)) title(Foreign Asset Share)"
			if "`var'" == "lambdaUR" local options " title(Real Exchange Rate)" // yscale(r(1.0 1.2))
			if "`var'" == "VR"              local options " yscale(log) title(Post-shock asset value)"
			if "`var'" != word("`vars'", wordcount("`vars'")) local legend "legend(off)"
			if "`var'" == word("`vars'", wordcount("`vars'")) local legend "legend(`modvarlegend')" // only display legend in last chart
			tsline `var'_`modvargraph'_* if tin(-2,26), `options' ylabel(#7, angle(0)) `yline' ///
				name(`var', replace) `legend' lpattern(l - _ _- .-- .._) lwidth(medthick ...) nodraw
		}
		di as result"Output graphs for model variations successfully created."
		graph combine `vars', iscale(.6) ///
//			note("Parameter values: deltaR=`=round(`deltaR',.01)', kappa=`kappa', kappaP=`kappaP', sigma=`sigma', shock=`=round(`=(VR[2]-`V0R')/VR[2]'*100,.001)'%")

		graph save    ~/diss/output/CFG/`model'/figures/`model'_`modvargraph'`output_file'.gph, replace
		graph export  ~/diss/output/CFG/`model'/figures/`model'_`modvargraph'`output_file'.png, replace width(1200)
		graph export  `"~/Desktop/Dropbox/Projects/07 Convertibility/Elsevier/`model'_`modvargraph'`output_file'.png"', replace width(1200) // "
		graph export  `"~/Desktop/Dropbox/Projects/07 Convertibility/Elsevier/`model'_`modvargraph'`output_file'.eps"', replace mag(200) logo(off) // "
		graph export "~/diss/output/CFG/`model'/figures/`model'_`modvargraph'`output_file' (`=c(current_date)').png", replace

		// PRESERVE normalised variables.
		foreach var in r_t lambdaUR {
			if "`graph_normalisation_`var''" == "on" {
				drop   `var'_`modvargraph'`modvarname'
				rename `var'2_`modvargraph'`modvarname' `var'_`modvargraph'`modvarname'
			}
		}

```

```
        // EXPORT variables for comparison.
        preserve
        keep *`modvaracr'_*
        save ~/diss/output/CFG/`model'/tables/`model'_`modvargraph'`output_file', replace
        di _newline as result "Results saved to " as input"~/diss/output/CFG/`model'/tables/`model'_`modvargraph'`output_file'" as result"."
        restore
        // Delete temporary variables from model variations
        // drop *`modvarname'

}

// END timer
local time_end = c(current_time)
local elapsed_sec = real(substr("`time_end'",7,2)) - real(substr("`time_start'",7,2))
local elapsed_min = real(substr("`time_end'",4,2)) - real(substr("`time_start'",4,2))
if `elapsed_sec' < 0 {
        local elapsed_sec = `elapsed_sec' + 60
        local elapsed_min = `elapsed_min' - 1
}
if `elapsed_min' < 0 {
        local elapsed_min = `elapsed_min' + 60
}
di _newline as text"Model converged after "as result `elapsed_min' as input" min(s)"as text" and " as result `elapsed_sec' as input" sec(s)" as text"."

// END loop for submodel.
di _newline as text"End model variations of submodel "as input"`submodel'" as text"."
}
```

Kapitel 2

Literaturverzeichnis

BEA (2006). A Guide to the National Income and Product Accounts of the United States. Bureau of Economic Analysis, Department of Commerce, http://www.bea.gov/national/pdf/nipaguid.pdf.

Caballero, R. J., E. Farhi und P.-O. Gourinchas (2006). An equilibrium model of “global imbalances” and low interest rates. Institute of Business and Economic Research, Working Paper C06-149, University of California, Berkeley.

Caballero, R. J., E. Farhi und P.-O. Gourinchas (2008). An Equilibrium Model of “Global Imbalances” and Low Interest Rates. *American Economic Review 98*(1), 358–393.

Duca, J. V. (1997). Has long-run profitability risen in the 1990s. *Economic and Financial Policy Review Q IV*, 2–14.

http://www.springer.com/978-3-658-04054-3

Wechselkurse und globale Ungleichgewichte
Wirtschaftsentwicklung und Stabilität Deutschlands und
Chinas in Bretton Woods I und II
Körner, F.M.
2014, XVIII, 328 S. 45 Abb., Softcover
ISBN: 978-3-658-04054-3