Cont'-ing w/ the estimation

25 July 2020

- despite adding SPF, still not identified so I'm really thinking it must be a code issue

E.g. the avg moments still depend on shocks when N=100? Can that be?

Also start taking notes on cleaning out the TIPS from a liq. premium in blue,

and notes on how to include the welfare mistake you make when you ass. RE instead of anchoring to the paper in brown.

Let's define some terms.

26 July 2020

Breakeven inflation = difference btw nominal & real yields of the same maturity

Fisher: $r = i - \pi \Rightarrow \pi = i - r$

$\Rightarrow \pi^{be} = i^{Tbill} - r^{TIPS}$

Andreasen et al say that

1. pos. bias in r^{TIPS} $\rightarrow$ neg. bias in π^{be}.

So the idea is

$$\pi^{be, true} = i^{Tbill} - (r^{TIPS} - \text{liq premium}^{TIPS})$$

Now, the FRED π^{be} series (T10YIE) is constructed as

$$T10YIE = WGS10YR - DFII10$$

$\uparrow$ 10-year Treasury

$\uparrow$ 10-Year Treasury Inflation-Indexed Security

(both constant maturity)

The question is if DFII10 is filtered for the liq. premium or not. Likely not.

A quick check for the liq premium could be what Andreasen's model is able to match well: the "model-free measure of the TIPS liq. premium, ... the difference between inflation-swap rates and TIPS-breakeven inflation."

<u>inflation-swap</u> := an agreement btwn 2 counter-parties to swap fixed-rate payments for a floating-rate payment linked to inflation. i.e. to swap a fixed-rate to an inflation-indexed rate payment!

It sounds like the swap rate is a measure of expected inflation too, which is why

$$\pi^{be} \overset{!}{=} \pi^{swap}$$

so that if they're not equal, then

$$(\pi^{be} + \text{liq prem}) = \pi^{swap} \Rightarrow \pi^{swap} - \pi^{be}$$

$= $ liq. premium.

This would be great if π^{swap} was a good measure of π-Exp. But it's not b/c the swap market is, although growing, very small. Still it's quite liquid (ho?) (Fleming & Sporn, 2013)

simplest form

Investopedia: a zero-coupon inflation swap (ZCIS) is also known as a breakeven inflation swap

But it doesn't even seem like π-swap data is publicly accessible. (A Cleveland Fed paper has it from Bloomberg, but I guess you need an account.)

27 July 2020

Haubrich et al emphasize that risk premia may also contaminate things, and I'm scared that Andreasen et al ignore that.

So cont. w/ Andreasen et al.

liq. risk = identified as the difference between prices of principal & coupon payments

The Andreasen et al ATSM model [Section 3]

(affine term structure model

$$r_t^N = \rho_0^N + (\rho_x^N)' X_t$$

nominal short rate — scalar — $N \times 1$ — N pricing factors

X_t evolves as

$$dX_t = \underbrace{K_x^Q}_{N \times N} (\underbrace{\theta_x^Q}_{N \times 1} - X_t)\,dt + \underbrace{\Sigma_x}_{N \times N} \underbrace{\sqrt{S_{x,t}}}_{N \times N}\, d\tilde{W}_t^Q$$

where

$\tilde{W}_t^Q$ is a standard Wiener process

$$[S_{x,t}]_{k,k} = \underbrace{\delta_{0\,k,k}}_{N \times 1} + \underbrace{\delta_{x,k}'}_{N \times 1} X_t$$

Price of nominal zero-coupon bond maturing at time $t+\tau$

$$P_t^N = \exp\{\underbrace{A^N(\tau)}_{\downarrow} + \underbrace{B^N(\tau)}_{\swarrow}{}' X_t\} \quad (3)$$

some known ODEs

In principle, TIPS (or other real bonds) could be priced like this, but that's not a good assumption given the low liquidity of the tips market.

So instead they ass that liq. costs are present, and in fact in the following form:

$$r_t^{R,i} = \rho_0^R + (\rho_x^R)' X_t + \underbrace{h(t - t_0; i)}_{\text{increasing fct of time since issuance, to (latent factor)}} X_t^{liq} \quad (4)$$

a time-varying liq costs

$Z_t \equiv [X_t', X_t^{liq}]$ we have an extended state vector which evolves according to a Wiener process. (5)

→ Price of a real zero-coupon bond maturing at T is:

$$P^{R,i}(t_0, t, T) = \exp\{A^{R,i}(t_0, t, T) + B^{R,i}(t_0, t, T)' Z_t\} \quad (6)$$

where A & B are given ODEs.

⇒ implied breakeven inflation rate from (3) & (6)

$$-\frac{1}{\tau} \log P_t^N(\tau) - \Big(-\frac{1}{\tau} \log P_t^{R,i}(t_0, t, \underbrace{t+\tau}_{=:T})\Big)$$

(which is a fancy way of saying $E\pi = i - r$)

$$= \frac{1}{\tau}\left[A^{R,i}(t_0, t, T) - A(\tau) - B(\tau)' X_t + B^{R,i}(t_0, t, T)' \begin{bmatrix} X_t \\ X_t^{liq} \end{bmatrix}\right]$$

Section 3.2 A Gaussian version of the ATSM w/ liquidity risk w/ closed-form expressions (!) for liquidity adjusted real prices

$$r_t^N = \underbrace{L_t^N}_{\text{level factor}} + \underbrace{S_t}_{\text{slope factor}} \quad (9)$$

for the real rate:

$$r_t^{R,i} = L_t^R + \underset{\substack{\uparrow \\ \text{scales EIR}}}{\alpha^R} S_t + \underbrace{\underset{>0}{\beta^i}\left(1 - e^{-\underset{\geq 0}{\lambda^{L,i}}(t-t_0)}\right)}_{\text{functional form for } h(t-t_0;\, i)} X_t^{liq} \quad (10)$$

Interpretation of β^i and $\lambda^{L,i}$

Trading of TIPS happens in 2 phases:

Phase 1: bond i just issued, high supply but also high demand (low liquidity risk)

Phase 2: buy-and-hold investors have acquired their share of TIPS i, are sitting on them contently and the supply of bonds i for trading is scarce (high liq. risk)

$\lambda^{L,i}$ = determines length of Phase 1; a low $\lambda^{L,i}$ implies a long Phase 1, less exposure to X_t^{liq}.

β^i = determines maximal exposure of i to X_t^{liq} in Phase 2.

Now, $z_t \equiv \left[L_t^N, S_t, C_t, L_t^R, X_t^{liq} \right]'$

$$dz_t = \underbrace{\begin{bmatrix} K_X^Q & 0_{4\times 1} \\ 0_{1\times 4} & K_{liq}^Q \end{bmatrix}}_{\substack{K_z^Q \\ 4\times 5}} \left(\underbrace{\begin{bmatrix} 0_{4\times 1} \\ \theta_{liq}^Q \end{bmatrix}}_{\substack{\theta_z^Q \\ 5\times 1}} - z_t \right) + \Sigma_z \, dW_t^Q \quad (11)$$

$$\underset{4\times 4}{K_X^Q} = \begin{bmatrix} \underline{0} & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{\lambda} & \underline{-\lambda} & \underline{0} \\ \underline{0} & \underline{0} & \underline{\lambda} & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & \underline{0} \end{bmatrix}$$

Actually, then the model (β^i, $\lambda^{L,i}$ and a bunch of other things for all TIPS $i = 1, \ldots n_{TIPS}$) is estimated w/ an extended Kalman filter where you back out X_t^{liq} as a filtered state..

na ne!

→ So let's try to understand the behavior of the liq premium and try to argue that it's not driving the dynamics of my figure 1.

p. 26 Mac

① The liq premium on TIPS is low and falling in general b/c i) the market is growing ii) dealers are expanding their TIPS-trading

② The TIPS liq premium is higher in recessions, e.g. 2001 9/11 and 2008, but stabilizes afterwards (also partly thanks to QE)

③ The 10-year TIPS exhibits a lower average liq premium and a less volatile one than the rest of the TIPS market

$$\text{mean}(\psi_t^{10y}) = 30 \text{ basis points} \quad \text{vs} \quad \text{mean}(\psi_t) = 38$$

$$sd(\psi_t^{10y}) = 13\text{ bp} \qquad sd(\psi_t) = 34$$

④ Older TIPS have a higher liq premium, but QE mainly lowers their liq premium since the Fed mainly bought TIPS which were issued long ago.

Their dataset goes till Dec. 27, 2013.

Basis points $= \frac{1}{100}\%$, i.e. 0.01%

So mean$(\chi_t^{10y}) = 30$ bp $\rightarrow 0.3\%$

If I add a sd $= 13$ bp to it, 43 bp $= 0.43\%$

So supp at 2020 Covid-shock, χ^{10y} rose by 2sd, i.e by 26 bp to 56 bp $= 0.56\%$, then π^E is downward biased by 0.56 pp, so instead of π-Exp$(\cdot)$ of 0.5%, it'd be 1%.

1pp $= 100$ bp

Since int. rates are usually changed by 25 bp, this liq premium is significant, but it doesn't change the message.

(5) $Corr(\Psi, \text{VIX index}) = 0.67$

avg. liq premium of Andreasen et al

between 0 & 100.

And: regressing Ψ on VIX gives a sig. effect of 0.85^{**}

$1 \uparrow$ in VIX $\rightarrow$ 0.85 bp $\uparrow$ in Ψ

In 2020: VIX $\uparrow$ by 60 $\rightarrow$ 51 bp $\uparrow$ in Ψ.

Here is a VAR(1) <u>28 July 2020</u>

$$y_t = \rho y_{t-1} + \varepsilon_t$$

What's its autocovariance at lag k?

$$E(y_t y_{t-k}) = ?$$

$$E(y_t y_{t-k}) = E(\rho y_{t-1} + \varepsilon_t)(\rho y_{t-1-k} + \varepsilon_{t-k})$$

$$E(y_t y_{t-k}) = \rho^2 E(y_{t-1} y_{t-1-k}) + \underbrace{E(\varepsilon_t \varepsilon_{t-k})}_{=0 \text{ if iid}}$$

$$\Gamma_k = \rho^2 \Gamma_k \quad \ddot{\sim}$$

Its VC-matrix is

$$E(y_t y_t) = (\rho y_{t-1} + \varepsilon_t)(\rho y_{t-1} + \varepsilon_t)$$

$$= \rho^2 E(y_{t-1} y_{t-1}) + E(\varepsilon_t \varepsilon_t)$$

$$\Sigma = \rho^2 \Sigma + Q$$

In this scalar case $\Sigma = (1-\rho^2)^{-1} Q$

→ Hamilton p. 67 Mac: j^{th} autocor of AR(1)

$$\gamma_j = E(Y_t - \mu)(Y_{t-j} - \mu)$$

Hamilton uses the MA(∞)-representation as

$$Y_t = \rho Y_{t-1} + \varepsilon_t$$

$$= \rho^2 Y_{t-2} + \rho \varepsilon_{t-1} + \varepsilon_t$$

$$\ldots = \varepsilon_t + \rho \varepsilon_{t-1} + \rho^2 \varepsilon_{t-2} + \ldots$$

$$\gamma_j = (\varepsilon_t + \rho \varepsilon_{t-1} + \rho^2 \varepsilon_{t-2} + \ldots) \qquad j=1$$

$$(\varepsilon_{t-1} + \rho \varepsilon_{t-2} + \ldots)$$

$$= (\varepsilon_t + \rho(\varepsilon_{t-1} + \rho \varepsilon_{t+2} + \ldots))(\varepsilon_{t-1} + \rho \varepsilon_{t-2} + \ldots)$$

$$= 0 + \rho(\varepsilon_{t-1} + \rho \varepsilon_{t+2} + \ldots)^2$$

$$= \rho^j(\sigma^2 + \rho\sigma^2 + \rho^2\sigma^2 + \ldots) = \rho^j \sigma^2 (1 + \rho + \rho^2 + \ldots)$$

$$= \frac{\rho^j \sigma^2}{1-\rho^2}$$

For AR(1), the j^{th} autocor is

$$\gamma_j = \rho^j \left(\frac{\sigma^2}{1-\rho^2} \right) = F^j \cdot \Sigma \quad \checkmark$$

in the VAR(1)-notation of Hamilton, eq. [10.2.21]

→ so it seems again that my Matlab code is fine.

→ in the multivariate VAR(1) case this is

$$\Big(I_{np^2} - \underbrace{F}_{\uparrow\,\rho} \otimes F\Big)^{-1} \cdot \underbrace{vec(Q)}_{\uparrow\,\sigma^2}$$

the point is, the moments seem to be computed properly.

Peter meeting

28 July 2020

1) F has to have $eig(F) < 1$

2) synthetic data: might be an issue of scaling

The logic of GMM $W^{-1} = \begin{bmatrix} 2 \cdot 10^{-6} & & \\ & 1 \cdot 10^{-6} & \\ & & \ddots \end{bmatrix}$

then what matters is that 1 is two times that of 2. But in Matlab, numerically, this can be so close that it screws up. Take the smallest order of magnitude on diag and rescale w/ that so all diag el's are bigger > 1.

Could also scale the moment vector, altho careful there b/c it might cancel the multiple of W.

3) Take artificial data

4) Try a VAR(1) b/c then the VAR is more misspecied

5) Can instead of estimating the autocov & fit a parametric model to the data

↳ paper by Hansen, Hodrick, Singleton about how to compute the asymptotic std. error (i.e. W) w/o bootstrapping

↳ would require a lot of coding

6) Try again to plot the loss fct as a fct of a single α, when the others are fixed at the truth.

Then you can iteratively to do it with two, and for three just do the blue-line-yellow-line plots.

7) W^{-1}

In Matlab $\left(\begin{pmatrix} 1^{-10} & 0 \\ 0 & 100 \end{pmatrix}^{-1}\right)^{-1} \neq \begin{pmatrix} 1^{-10} & 0 \\ 0 & 100 \end{pmatrix}$

haha! So that could easily cause problems.

<u>Ryan meeting</u> <u>29 July 2020</u>

(Materials 38)

- Makes sense to explore the calibration of measurement error b/c they shouldn't be visible on the ACF.

  Shut off the moments for $E(\pi)$ and just add the meas. e. $\rightarrow$ compare ACF and see if meas. er. made things worse on its own.

- Maybe the meas. error causes a wedge between N_{estim} & N_{simul}

- Fig 7. Don't reflect (re-)scaling of W in certain respects, e.g. 10^{-5} is smaller in magn. than the others, vs. bottom panel.

Top panel rejects that model is ID b/c of flat regions.
Bottom panel ?!

Do Fig 7. of Mat 38 w/o expectations
- Why does the loss hit a hard 0?
- Why does rescaling change the shape?

↳ based on this picture and the others: a bug.

- Can only have as many moments as parameters in F & Q → so p=4 is better than p=1

2 reasons why stoch. singularity may not happen:

- Model is non-linear, so when you est a VAR w/ fixed coeffs it might <u>not</u> be stoch. singular

The other reason is that observables depend on past lags (of shocks) which the VAR doesn't see, and those act like new shocks

What could happen though is that F is very volatile. (And close to singular.)

What you can do?

Ridge $\lambda = 0.01$ small compared to $X'X$
then singularity

meas error: add a little.

How to hunt for bugs

4 August 2020

1) Investigate whether the measurement error is behind it.

1a) shut off the Expectations moments and look at autocovariogram w/ and w/o meas. error.

→ autocovariograms shouldn't change

1b) same: plot loss for changing α's w/ and w/o meas. error and w/ and w/o rescaling

If the meas. error is behind it, then try cutting it out by reverting to a ridge regression approach if stock singularity does become an issue.

No, keep Nestimations so you're comparable.
~~Use NSimulations as a default instead of Nestimations.~~

2) I want to do a general combing thru of various codes.

1a): If I take out the expectations-moments, W is no longer tiny, W^{-1} no longer explodes... → the expectations moments are the ones responsible for the "scaling W" issue.

Another thing I note is that the moment I use the true data w/ meas. error, residuals become super-small. This is true even if you don't use meas. error in the true data, only in the estimation.

The measurement error does show up in the ACFs, → it changes the initial moments, but that's not a surprise. It shouldn't for Nsimul though, and it doesn't really either.

- Should it change the truth? Yes, I think so b/c that's just one simul.

1b) Losses indicate that

i) the rescaling thing is really an issue when E() are used → all losses look like the rescaled one.

ii) meas. error is scaling down losses a lot! why?

iii) α's at the edges look ID'd; but at the wrong place. (a little too low)

iv) Middle α's are not ID'd: flat in the correct region.

v) 0-α looks ID'ed at zero.

↳ why opposite of last week's?

↳ if ID'ed at zero, why isn't estim giving a zero there?

⇒ most likely b/c interaction w/ the ones that aren't ID'ed.

↳ again, why is meas. error changing the loss?

⇒ I'm also wondering if actually this means that w/ more info that wouldn't be screwed up, we could ID everything?

I wonder why $E(\cdot)$-moments have no variance. Try setting $K = 1$ (use only 2 lags, 0 and 1) as the $E(\cdot)$-moments have lowest variance at longer lags. Why?

W^{-1} still is e+05.

↳ not saving these figs since W^{-1} is still exploding, and then I prefer to keep K the same.

Doesn't seem to improve a lot. The problem is that it isn't representative b/c W^{-1} exploding.

Check out the same for $K=4$.

⇒ In both case Matlab doesn't complain when I generate the data, but it does complain maybe six times while estimating that stuff is close to singular.

W^{-1} is on the order of e06.

k=4 suggest (Nsimul) that the moments are matched very well, so previously it was indeed the meas. error that made them unmatched.

As for the loss, $\alpha - 0$ seems to be reversed when you add $E(\cdot)$.

<u>Checking codes</u> <u>5 Aug 2020</u>

One thing I notice for Nsimul is that (N=100)

$loss(\alpha^{true}) > loss(\hat{\alpha})$

1736 > 944

$N = 1000$?

resnorm$(\alpha^{true}) = 1821$

$N = 10\,000$?

resnorm$(\alpha^{true}) = 1834$

$N = 10$?

resnorm$(\alpha^{true}) = 1686$

→ Is the loss increasing in N? It seems so!

Why?

I've checked and obj_GMM_Compair_univariate_mean.m computes the moments exactly as command_acf_sim-data_univariate.m does. The only diffs are that

i) command_acf... also computes the lag p, which is an input to obj_GMM...

ii) obj_GMM... does the computation N times and then takes a mean.

→ why is $res = (Om^{data} - mean(Om))\, W^{-1}$
increasing in N? → can only be if if mean(Om) moves away from Om^{data} as N↑.

Check that acf saves the right things! ✓ Done.

obj_GMM_LOMgain_univariate.m is also correct; it also does exactly the same thing that command_acf... does.

The last thing to comb thru is sim_LamLH_dean_approx_univariate.m

→ I still wonder if taking no inverse gains would help "unclutter" the learning code. Especially as $k^{-1} = 0$ sometimes, which set $k = \infty$, which, I believe, is penalized in obj_GMM_LOMgain_univariate.m

"Uninverting k"

↳ sim_learnLH_clean_approx_univariate_uninvertk.m

↳ fk_CEMP_uninvertk.m

↳ fk_CUSUM_vector_uninvertk.m

↳ fk_smooth_approx_univariate_uninvertk.m

Test it out in uninvertk.m

Haven't implemented "uninverted IRFs" but I obtain the same exog states, jumps, k, shocked k, and convergence diffs.

Now estimate using the uninverting learning code

Nestimations: surprisingly, I get pretty much the same thing (see 1.2.3 in Materials 39).

Nsimulations: same :(, even more exactly the same thing! The loss is exactly the same.

↳ for this at least I got a loss of 740 instead of 780.

So far my discussions w/ Ryan & Peter:

1) Measurement error does impair the ability to match moments. (Fig 1 & Fig 2)

- Both for Neshin & Nsimul
- The loss becomes lots smaller if "data" has meas. error in it.
- The m.e. seems to, strangely, give some ID-info. (seen nicely on Fig. 3)

2) Loss

- The "W explodes"-issue is only there if E(.) are included. (Fig 4)
- Fig 5 again: meas. error adds some ID info
- Fig 6: adding E(.) adds huge problems: loss↑, directions reverse.

Peter meeting   Materials 38   5 Aug 2020

- Fig 4: Shape shouldn't change shape from top to bottom.

scalar, matrix (VC matrix of moments)

$$Y = \alpha X$$

Check: $Y^{-1} = \frac{1}{\alpha} X^{-1}$

Check: $X^{-1} ./ Y^{-1}$ = matrix of α's

You are inverting a matrix at one point. If the inversion is done correctly then you should get $X X^{-1} = I$.
Need to confirm that this isn't happening at one place.
→ that's where the error is.

- Measurement error: loss goes down

Could multiply by sthg to make it go up.

• Fig 4. Supp Panel (a) is an accurate pic of loss fct.

1.) Ignoring the flat parts, the loss takes on a value very close to zero close to the true value

2) loss steeply sloped below truth

↳ do we see a U-shape when increasing the neighborhood of true?

If yes, then ID but large std error b/c it converges to a diff values every time.

⇒ check again

ID problem: check also "absolutely flat" vs reaching a global min at the truth and being slightly bigger off to the right.

ID, but pretty flat

Meas. error:

Loss falls. Typically we think that a lower loss as a better fit.

But a model w/ & w/o m.e. doesn't correspond to each other.

Even if you add noise to data & loss ↓, even that can happen b/c it's a different dataset! Est-ing the same model on different data is not comparable.

2 possibilities:

1) params of model are ID-ed, but shallow bowls on the surface mean that the estim is challenging numerically, std errors large

↳ then let it run 10 000 times, and say in an App. all this. w/ diff starting values → std errors tell us that estimated w/ uncertainty

2) not ID-ed w/ this data

then Bayesian est. of DSGEs the prior is doing a lot of the work.

Ryan meeting 5 Aug 2020

BosFed meetings 10.30 Tuesday.

meas error making est variances of moments larger, and therefore fitting the moments worse but giving a smaller loss,

The sense in which something is going wrong is

- if the solver quits too soon
- if the σ_{measerr} is too high & me. is playing too large a role

Check Fig 2: take α(row 1), plug into row 2 w/ m.e. → moments should look just as good as in row 1 just Matlab exited too soon.

∘ true data has no action in E(·)
→ do a sim w/ more action.

<u>Fig 5</u>: most troublesome: loss is huge for the 0.025 params.

· not-locally: extremely curvy
· locally: very flat

↳ Need to zoom in and see what's happening

- The meas. error isn't behind any of this.
- We still don't know what is.