Lynx Market - process chls

GaussYaml - formats calibrated data in a Yaml friendly way
- even populated fields are coming from otherplaces - their values are not fixed

Historical Calibration

allow to do calibration for different COB dates, to check stability of the calibration process.
- generated date list via Freq/start/end
- either save the file for historical calib [not sure if it works] or run the calibration in Excel.

Rates - lynx specific, used to retrieve data from lynx + manipulate data
TrancheQuotes
CDSSpreads ⇒ transform into CDSspreads, base corr surfaces, def i tranches
MarketTranchePrices ⇒ market tranche spreads [calib targets].
- Use Triton to generate tranche prices

Rates - will get rate curve from lynx
- only uses USD Cash swap curve
- only calibrating CDX Index - this is USD only index
- (A) in the future, will have a currency per index
- (A) this is ccy of tranche instrument itself
  - there's also ccy in which CDS is quoted + name in that index (it's typicall all will be same ccy, if not, need generalize).

- We use the curve to do 2 things:
  - bootstrap CDS curves [calib assets]
  - discount cash flows [pricing - instrument specific]

(A) in the future, we may want to separate it. (calibrating assets vs instrument)

out of order

CDS spreads

name Lynx Mkt ToB

- Names of components of the index we calibrating.
- Obtained from Lynx through data acquisition
- each name has wt, recovery, CDS spreads
- how that we construct CDS curve (column Q)
  → make a Triton obj out of that
  → bootstrap it, column R

There are 2 workflows with S/S – one is to work with raw mkt curve (base mkt data) or to work with tweaked (to all sensitivity calculation – 20% shock in underlying names).

Handle B/O Lynx Mkt Tweaked Run %
Tweak Size (20%) = 0.2

- we tweak the CDS curve and then it gets bootstrapped by Triton
- incase of tweaked run, we get a diff't yaml.
- "Calibrate" does both but you could run manually to get each side to run ~~independent~~ by itself.
- typical prod setting – we produce 2 yamls (~~each~~ we combine them in 1 xml).

Tranche Quotes

- Retrieve base correl surfaces. (default corrs for components of the index)
- Tranche prices are liquidly traded
- Tranche protection derivative (like CDS) when losses in the underlying portfolio (0% = no one df, 100% = all df w/o regard)
  – you get paid for any losses in (K_1, K_2)
  (payoff like a call spread)
  – 2 observables → (0, n%) tranche – any loss hits up to n%
  ⇒ = ΣCDSs depends only on CDSs

  just depend on E[loss], any point loss, not correlation

  - once $K_1 > 0$, it's also a function of how correlated names are.
  - we use a Gaussian copula pricer <needmore> ⇒ base corr surfaces
    ↘ assume all names have same corr
    ⇒ prices for tranche
  - done by middle office
  - ~~[illegible] corr of [illegible] and (K_1, K_2)~~ = C(K_1, K_2)

- We write Tranche (K_1, K_2) as $Tr(0,K_2) - Tr(0,K_1)$ and price each of their coms.
  Base corr surf = $(K_1, K_2) \Rightarrow$ corr.
- Tranche breaks are standard, it's also from Lynx.
- Base corr surf are assumed as input - already calibrated

Tab
- Attachment (K_1) and Detachment (K_2) points
- ~~for each mat~~ reverse exercise - gives BaseCorrSurf, and CDS spread of the index, what are Tranche prices.
  → we want not only observed tranche prices, but also in hypothetical scenarios

- ~~Maturity table: Compute tranche prices~~
- EG2-68/FG2-68 - settings to drive the calc
  most inp are FG6-68.
  - Use Tr fcn (base corr surf, CDS Index spr) ⇒ mkt spreads
  - it computes ~~dt~~ loss distr. for portfolio
    $= P[\text{loss} \leq K] \; \forall K$. (pdf[loss])
- ~~once~~ this is computed,
  can go back to Maturity Rows [18-55], (each row represents a definition tranche)
  Col. 0 is tranche spreads
  → this gets copied into Market Tranche Prices tab.
  (E18→P55) → [load Market] → A51:...
  The copying gets done by [load Market] button,
  ~~because~~ because want to decouple Lynx from Qlib,
  So once the MTP tab gets populated, you don't need Lynx to work for you - just the Qlib price.

Market Tranche Prices compress data to B2 - F30 (final mkt tranche spreads - 1 # for each tranche / T).

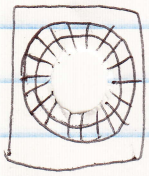
Qlib Swap Curve - compute the Qlib interest rate curve there
- inputs/rates copied from Lynx sig curve
- also a transition tab - copy done by VBA code

CDX another transition tab
- news of same index components names/data - wts, recovery, spreadcurve

Simpathi To price CDS by the CID model, we construct a QLib calibrator object, with market spreads marked as calibration targets.

Calibrator ← Simpathi, CIDParams, Tranche Mktg ⇒ pathConfig

Step, data – constructed later in Fluri tab (B39, B76, B114)
- each step is its own calibrator
- diff which param were trying to calibrate. [≤3 parms/step], to avoid a high-D optim problem.



~~Market from Simpathi tab (J2)~~

C16
Market also an input, from Simpathi tab J2
- s/s defaults to determ IR model (saves calib time) [doesn't add much IRdiffus].

MktSpreads table #17 - 019 from MktTranchePrices
pathConfig Simpathi
IndexTranche – MC instrument in QLib to price IndexTranches called MCIndexTranche Simpathi C4 = C6.

TranchePricer – what do we price
- control ⇒ ~~what requests~~
- requests ⇒ what to do

CIDParams similar to SimpathI, but also has an initial guess, solver would take CID config and use it as initial guess.
- Stochastic Freq CIR Jump is the initial guess (all of them)
- reorganizes data from CalibRation Results + other controls and settings

Calibration Results

- initial guess
- model spreads (under weights)
- mkt spreads from MKT TranPrice tab
- error computations

Buttons

- price – chg init guess and see the impact on model spreads (7-9 secs)
- Calibcurr – (init guess, target spreads) → calibrator addin → takes a while

  ctrled by flag. Calibrator!D2
- load Calib FromFile
  - load prev done calibs (~~mktg~~ chg Lynx Mkt tab – precalibrate)
  - same ~~tab~~ row 76

    (step6 is a placeholder, we only use 5 or less)
  - step final – final set of parameters used to reset init guess
- copy calib parms
  - reset init guess to step final

Jump Impacts – exper add term vol error fcn. for smoothness.

Remote Calib – produce XML to combine both Excels.