Pricing Synthetic CDOs Based on Exponential Approximations to the Payoff Function

Bachelier Congress
Toronto, Canada
June 24, 2010

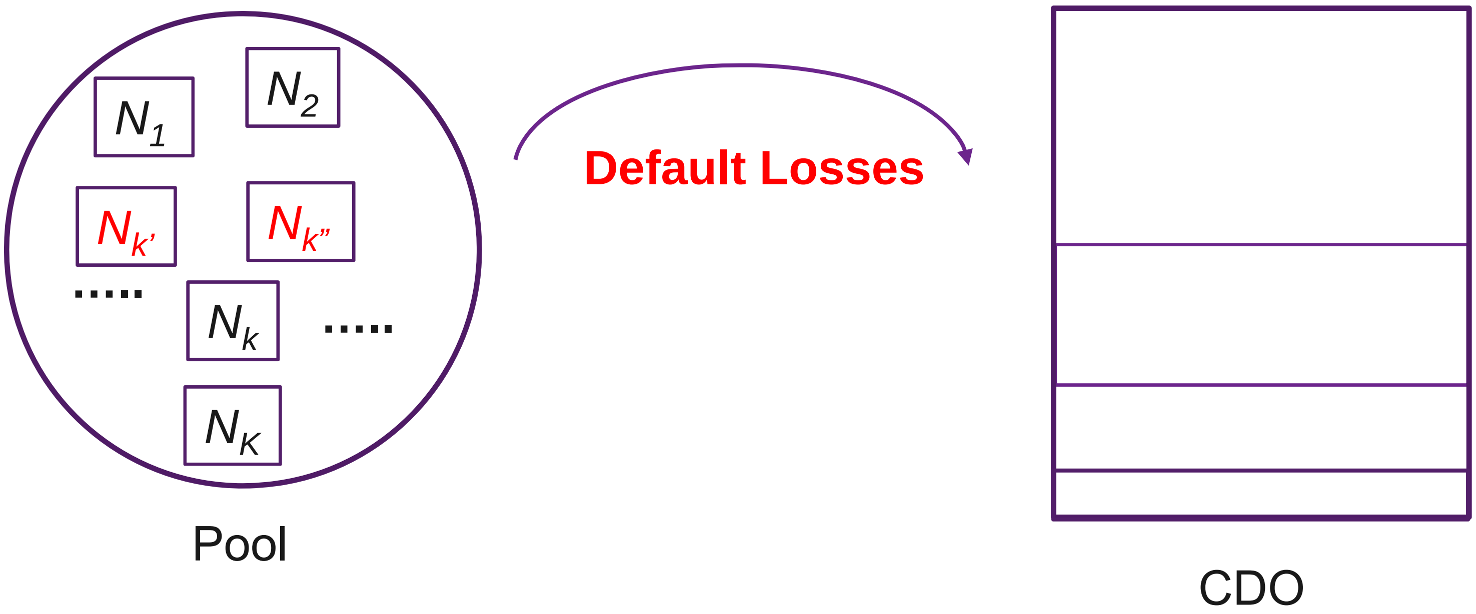
Ian Iscoe
Quantitative Research, Algorithmics, Inc.
Alex Kreinin
Quantitative Research , Algorithmics, Inc.

Ken Jackson
Dept. of Computer Science, University of Toronto
Xiaofang Ma
Bank of Montreal, Toronto

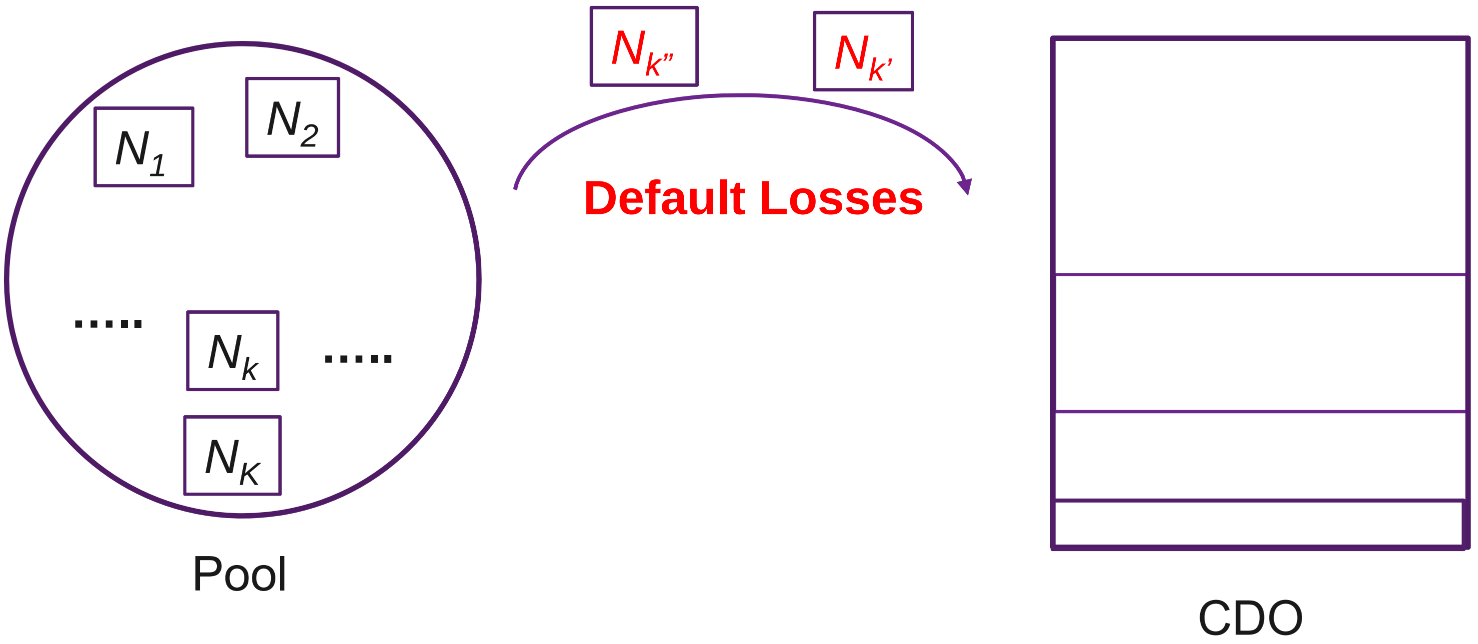
Outline

1. Introduction: brief review of CDO structure & pricing
2. Basic problem
3. Comparison of approaches: traditional vs EAP
4. Application to CDOs
5. Pros & Cons
6. Source of exponential approximation

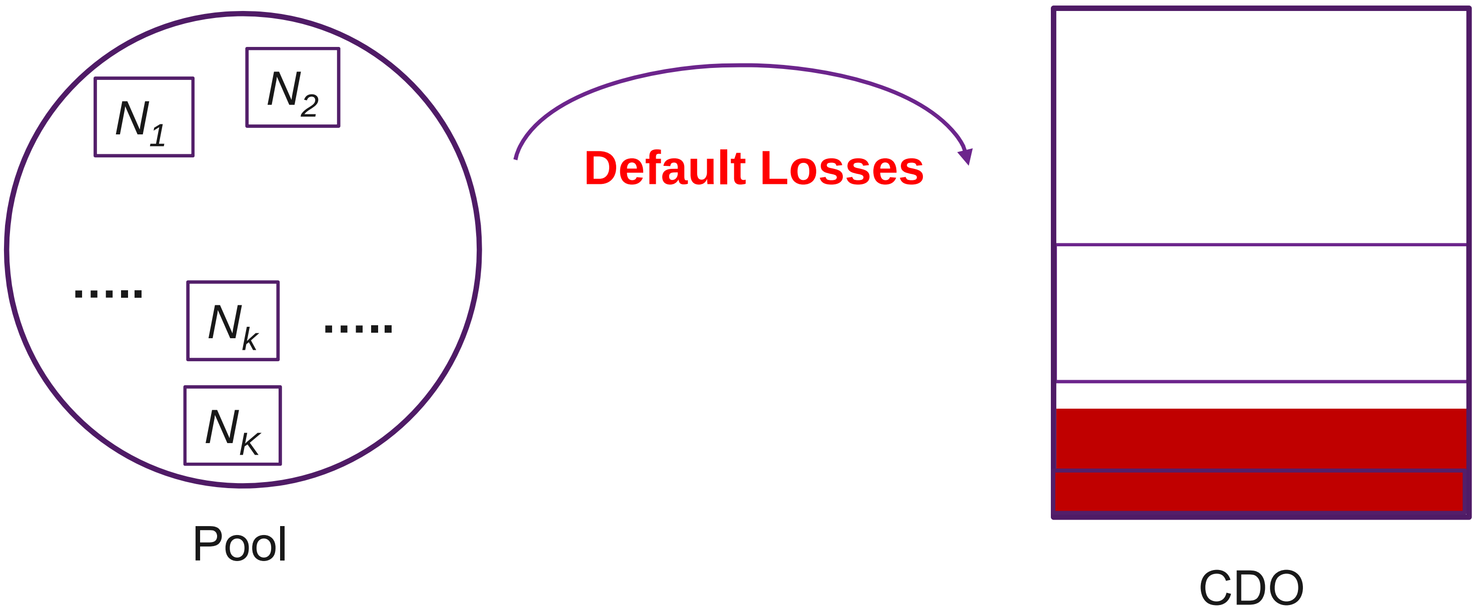
1.1 Synthetic CDO structure



1.1 Synthetic CDO structure

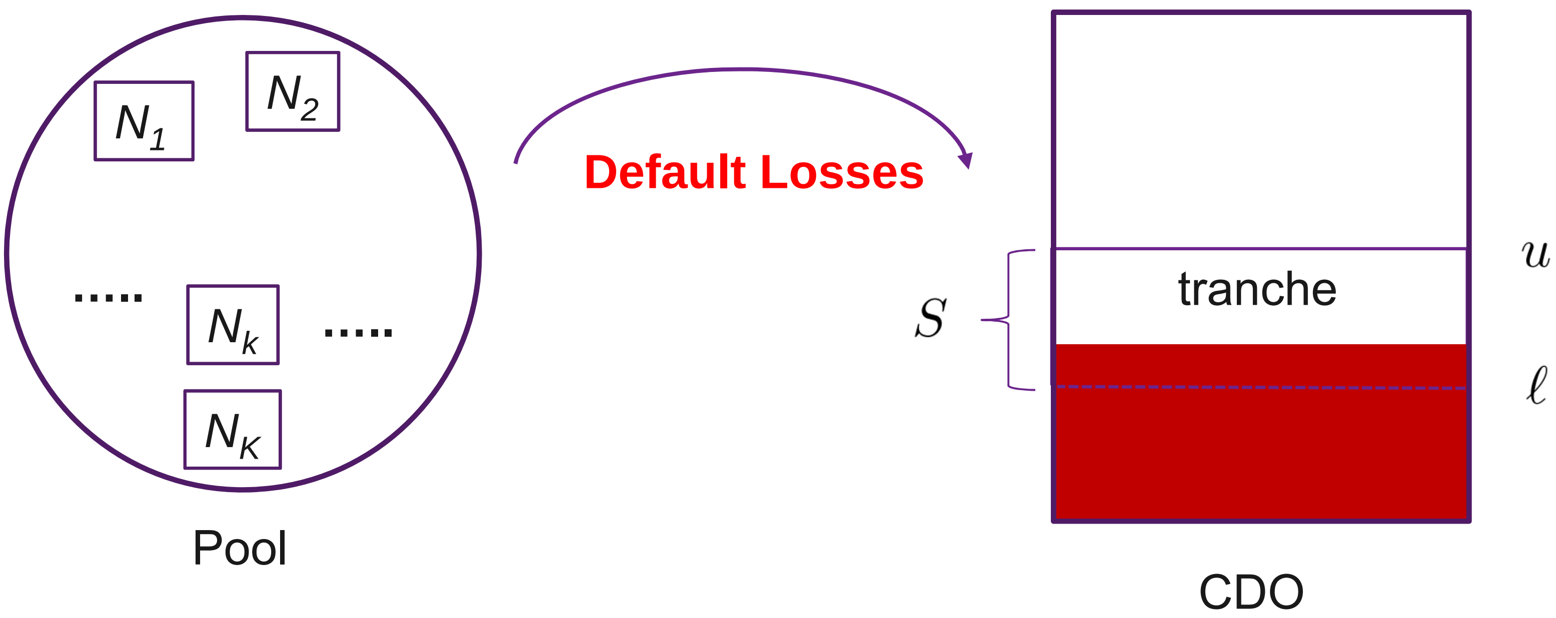


1.1 Synthetic CDO structure

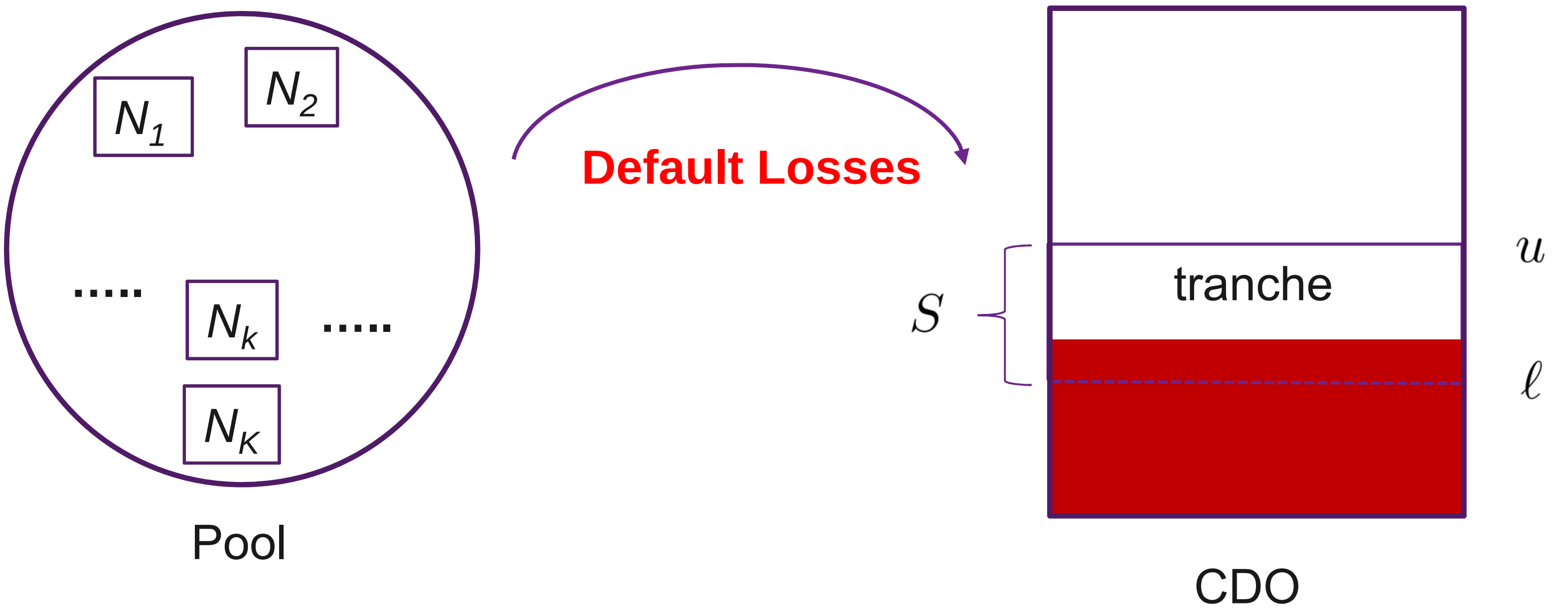


Algorithmics

1.1 Synthetic CDO structure



1.1 Synthetic CDO structure



Premiums

t

$t_0 = 0$ $\quad t_{i-1}$ $\quad t_i$ $\quad t_n = T$: Premium dates t_i, $1 \leq i \leq n$

Premium for ith period (due at t_i) $\propto$ (S − tranche losses up to t_i)

↑ $const. = s \times (t_i - t_{i-1})$; s : "spread"

Algorithmics

1.2 Synthetic CDO: Structure summary for pricing

General assumptions

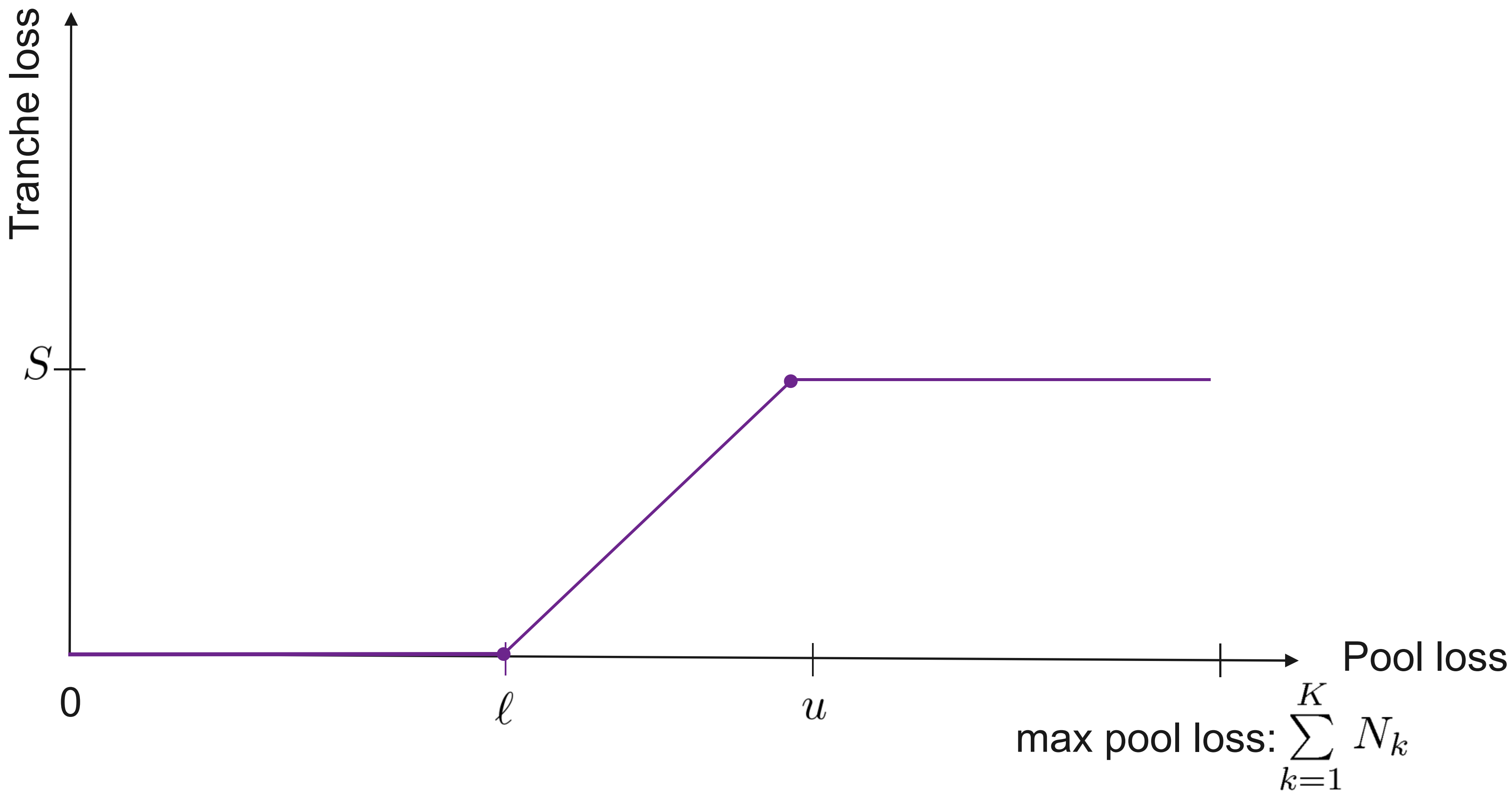
- Constant fair spread rate, s;
- Fixed premium times after today (t_0): $0 = t_0 < t_1 < t_2 < \cdots < t_n$;
- Deterministic discount factors, d_i , corresponding to t_i;
- Credit events occur only “at” each premium date;
- Static underlying pool.

Notation

- $\mathcal{L}_i^{(k)} :=$ loss on kth name, up to time t_i;
- $\mathcal{L}_i := \sum_{k=1}^{K} \mathcal{L}_i^{(k)}$: pool's cumulative losses up to time t_i;
- ℓ: attachment point of the tranche;
- u: detachment point of the tranche;
- $S := u - \ell$: thickness of the tranche;
- $L_i = \min\left(S, (\mathcal{L}_i - \ell)^+\right)$: tranche loss up to time t_i .

Algorithmics

1.3 CDO tranche payoff function



1.4 Synthetic CDO: Pricing equations

Swap Equations

$$\text{PV[Default leg]} = \sum_{i=1}^{n} \mathbf{E}[(L_i - L_{i-1})d_i]$$

$$\text{PV[Premium leg]} = s\sum_{i=1}^{n} \mathbf{E}[(S - L_i)(t_i - t_{i-1})d_i]$$

s from setting: PV[Default leg] = PV[Premium leg]

Value to protection seller = PV[Premium leg] − PV[Default leg]

Algorithmics

1.4 Synthetic CDO: Pricing equations

Swap Equations

$$\text{PV[Default leg]} = \sum_{i=1}^{n} \mathbf{E}[(L_i - L_{i-1})d_i] = \sum_{i=1}^{n} \left(\mathbf{E}[L_i] - \mathbf{E}[L_{i-1}]\right) d_i$$

$$\text{PV[Premium leg]} = s \sum_{i=1}^{n} \mathbf{E}[(S - L_i)(t_i - t_{i-1})d_i] = s \sum_{i=1}^{n} (S - \mathbf{E}[L_i])(t_i - t_{i-1})d_i]$$

s from setting: PV[Default leg] = PV[Premium leg]

Value to protection seller = PV[Premium leg] − PV[Default leg]

1.4 Synthetic CDO: Pricing equations

Swap Equations

$$\text{PV[Default leg]} = \sum_{i=1}^{n} \mathbf{E}[(L_i - L_{i-1})d_i] = \sum_{i=1}^{n} (\mathbf{E}[L_i] - \mathbf{E}[L_{i-1}])\, d_i$$

$$\text{PV[Premium leg]} = s \sum_{i=1}^{n} \mathbf{E}[(S - L_i)(t_i - t_{i-1})d_i] = s \sum_{i=1}^{n} (S - \mathbf{E}[L_i])(t_i - t_{i-1})d_i$$

s from setting: $\text{PV[Default leg]} = \text{PV[Premium leg]}$

$$\text{Value to protection seller} = \text{PV[Premium leg]} - \text{PV[Default leg]}$$

Essential Calculation

$$\mathbf{E}[L_i] \equiv \mathbf{E}[f(\mathcal{L}_i)] \equiv \mathbf{E}\left[f\left(\sum_{k=1}^{K} \mathcal{L}_i^{(k)}\right)\right]$$

where

$$f(z) = f(z; \ell, u) = \min(u - \ell, (z - \ell)^+)$$

Algorithmics

2. Basic problem (abstracted)

- **Setting:** conditional independence framework; i.e.,
  - family of non-negative r.v.'s Z_k which are conditionally independent, conditional on some auxiliary r.v. (possibly vectorial), $\mathcal{M}$, with distribution $\Phi(M)$.
  - payoff function f , evaluated on $Z := \sum_{k=1}^{K} Z_k$.

Algorithmics

2. Basic problem (abstracted)

- **Setting:** conditional independence framework; i.e.,
  - family of non-negative r.v.'s Z_k which are conditionally independent, conditional on some auxiliary r.v. (possibly vectorial), $\mathcal{M}$, with distribution $\Phi(M)$.
  - payoff function f , evaluated on $Z := \sum_{k=1}^{K} Z_k$.
- **Essential numerical aspect:** Efficient and accurate evaluation of

$$\mathbf{E}_M[f(Z)] = \mathbf{E}[f(Z) \,|\, \mathcal{M} = M] \tag{1}$$

leading to an evaluation of

$$\mathbf{E}[f(Z)] = \int \mathbf{E}_M[f(Z)]\, d\Phi(M).$$

Algorithmics

3.1 Comparison of approaches

- Two types of approaches
- Each addresses conditional expectation (1) differently
- Final integration (over M) is the same for both types

Algorithmics

3.1 Comparison of approaches

- Two types of approaches
- Each addresses conditional expectation (1) differently
- Final integration (over M) is the same for both types

Traditional approach

1. Compute the conditional distribution Ψ_M of Z , conditional on $\mathcal{M}$, using either FFT, recursion, or some approximation method.

2. Compute the conditional expectation $\mathbf{E}_M[f(Z)]$:

$$\mathbf{E}_M[f(Z)] = \int f(z)\, d\Psi_M(z)$$

3. (Integrate the conditional expectation over M.)

Algorithmics

3.2 Comparison of approaches (cont'd)

EAP approach

1. Approximate the non-smooth function f by a finite sum of exponentials.

2. Approximate the conditional expectation $\mathbf{E}_M[f(Z)]$ via explicit* evaluation of $\mathbf{E}_M[\exp(cZ_k)]$. (*Assumption!) **No Ψ_M is necessary.**

3. (Integrate the conditional expectation over M.)

Algorithmics

3.2 Comparison of approaches (cont'd)

EAP approach

1. Approximate the non-smooth function f by a finite sum of exponentials.

2. Approximate the conditional expectation $\mathbf{E}_M[f(Z)]$ via explicit* evaluation of $\mathbf{E}_M[\exp(cZ_k)]$. (*Assumption!) **No Ψ_M is necessary.**

3. (Integrate the conditional expectation over M.)

2. (reprise) Details:

$$f(z) \approx \sum_{n=1}^{N} w_n \exp(c_n z) \Longrightarrow$$

$$\begin{aligned} \mathbf{E}_M[f(Z)] &\approx \sum_{n=1}^{N} w_n \mathbf{E}_M[\exp(c_n Z)] \\ &= \sum_{n=1}^{N} w_n \mathbf{E}_M\left[\prod_{k=1}^{K} \exp(c_n Z_k)\right] = \sum_{n=1}^{N} w_n \prod_{k=1}^{K} \mathbf{E}_M[\exp(c_n Z_k)]. \end{aligned}$$

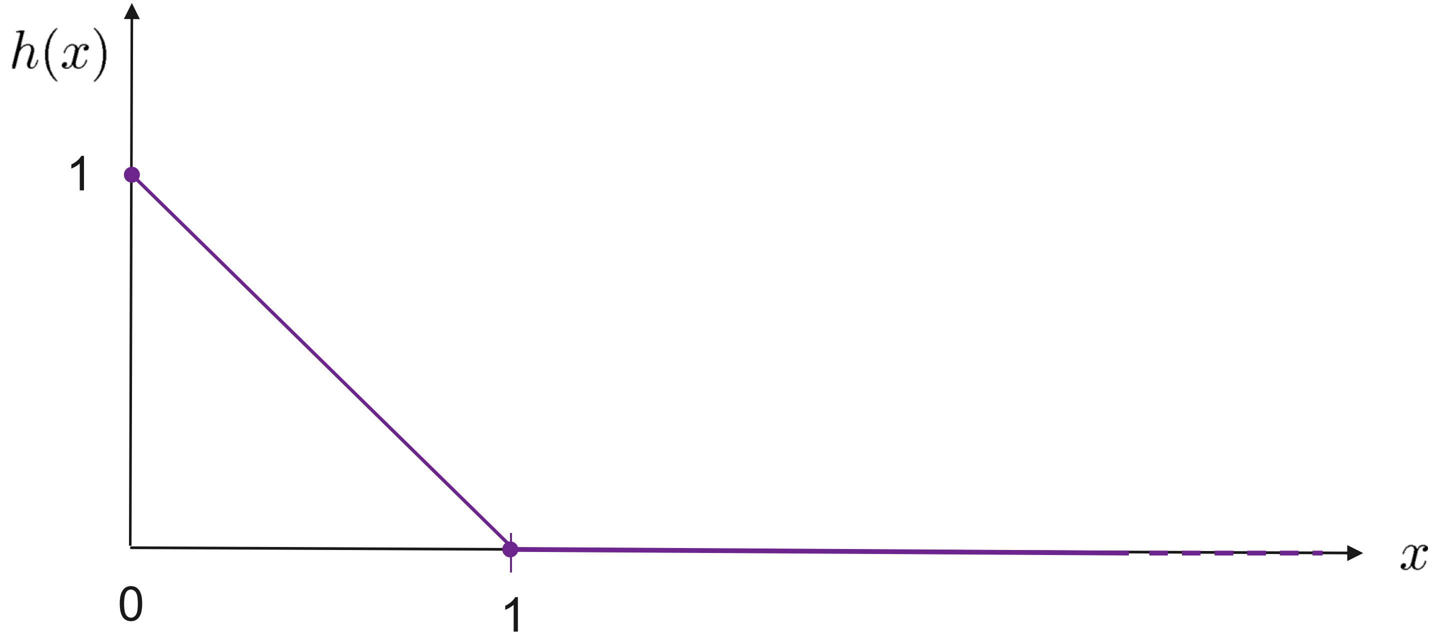
Algorithmics

4.1 EAP applied to CDO: Reduction of payoff function to hockey-stick function

For CDO,

$$f(z) = f(z; \ell, u) = u\left[1 - h\left(\tfrac{z}{u}\right)\right] - \ell\left[1 - h\left(\tfrac{z}{\ell}\right)\right],$$

where $h(x) = 1 - x$ if $x \leq 1$, 0 otherwise. ("Hockey-stick function")



4.2 EAP applied to CDO (recap)

Suppose

$$h(x) \approx \sum_{n=1}^{N} \omega_n \exp(\gamma_n x),$$

where ω_n and γ_n are (in general) complex numbers.

Then

$$\mathbf{E}_M[f(Z)]$$

$$\approx (u-\ell) - u \sum_{n=1}^{N} \omega_n \prod_{k=1}^{K} \mathbf{E}_M\left[\exp\left(\frac{\gamma_n}{u} Z_k\right)\right] + \ell \sum_{n=1}^{N} \omega_n \prod_{k=1}^{K} \mathbf{E}_M\left[\exp\left(\frac{\gamma_n}{\ell} Z_k\right)\right]$$

Note: Only $\mathbf{E}_M[\exp(cZ_k)]$ of individual names are computed, where $c = \frac{\gamma_n}{\ell}$ or $\frac{\gamma_n}{u}$.

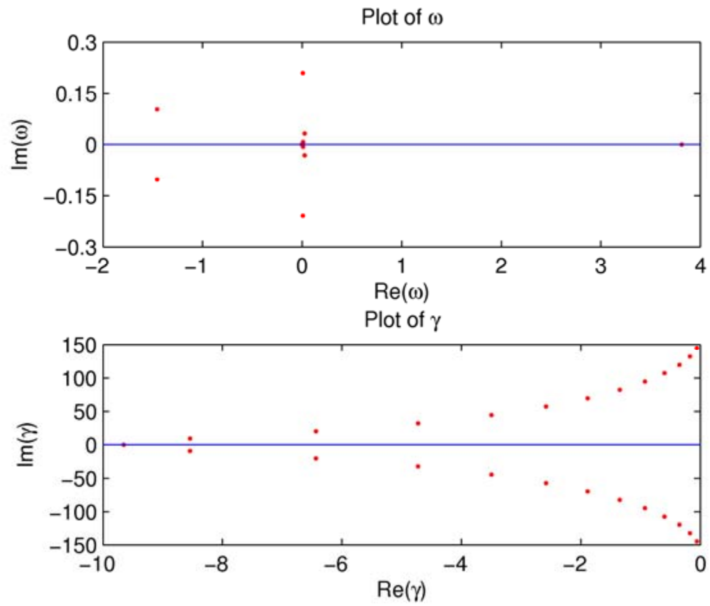
Algorithmics

4.3 EAP applied to CDO: Hockey-stick function's parameters

EAP approach reduces to the uniform approximation problem:

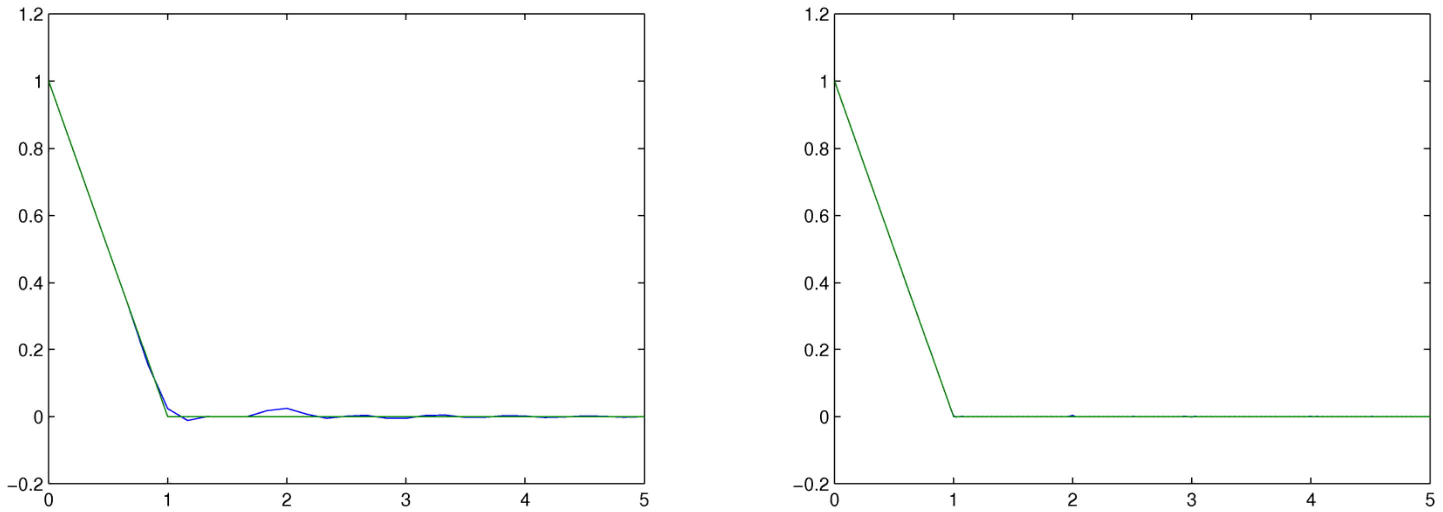
$$h(x) \approx \sum_{n=1}^{N} \omega_n \exp(\gamma_n x),$$

where ω_n and γ_n are complex numbers. E.g., with N = 25:



Parameters γ_n and ω_n for the 25-term approximation.

4.4 Plots of two approximations to h



Left panel: 5-term exponential approximation;
Right panel: 49-term exponential approximation

The maximum absolute error in the approximation is roughly proportional to $1/N$:

N	25	50	100	200	400
Max absolute error	6.4e-3	3.2e-3	1.6e-3	8e-4	4e-4

Algorithmics

5. Pros and cons of EAP approach

Pros

- Faster than traditional approach for:
  - single tranches
  - very heterogeneous pools
  - large pools

Ex. EAP-50: 10 x faster for first 4 tranches of one real CDO with 140-name, very heterogeneous* pool (*LGD varied from LGD_{min} to $LGD_{max} = 7 \times LGD_{min}$)

- Quite accurate (e.g., with 50 exp terms, spreads observed correct to within 1 bp; for all but highest tranche: < 0.5% rel error)
- No rounding of losses, as in many versions of the traditional approach
- EA can be calculated once, stored, then used for many pools
- Sensitivities (e.g., of spreads to PDs) are easily incorporated

Cons

- Slower than traditional approach for:
  - multiple tranches (> 3)
  - highest tranche (requires very large number [~200] of exp terms)
  - very homogeneous pools

Algorithmics

6.1 Source of Exponential Approximation

Revised notation

- M: $2M+1 = \#$ points in partition of $[0,1]$: $\left\{\frac{k}{2M} : 0 \le k \le 2M\right\}$
- h: any continuous function on $[0,1]$
- $h_k := h\left(\frac{k}{2M}\right)$

Discretisation

For $h(x) \approx \sum_{n=1}^{N} \omega_n \exp(\gamma_n x)$, set $\zeta_n = \exp(\gamma_n / 2M)$.

Consider discretised problem:

$$h_k = \sum_{n=1}^{N} \omega_n \zeta_n^k, \quad 0 \le k \le 2M, \qquad \text{(equality!)}$$

where N, ζ_n, ω_n TBD, $1 \le n \le N$.

Algorithmics

6.2 Source of EA (cont'd)

Gaspard de Prony (~1795)

3. Set $\{\zeta_1, \zeta_2, \ldots, \zeta_N\}$ to be roots of some polynomial equation

$$\sum_{k=0}^{N} q_k \zeta^k = 0.$$

4. Solve for $\omega_1, \omega_2, \ldots, \omega_N$ as solution to linear equations

$$h_k = \sum_{n=1}^{N} \omega_n \zeta_n^k, \quad 0 \leq k \leq N-1. \quad (*)$$

Require $(*)$ also holds (by induction) for $N \leq k \leq 2M$.

6.2 Source of EA (cont'd)

Gaspard de Prony (~1795)

1. Form $(M+1) \times (M+1)$ Hankel matrix H: $H_{kn} = h_{k+n}$.
2. Find $(M+1)$-vector q s.t. $Hq = 0$, with $q_N = -1$; $q_n = 0$, $n \geq N$. This is a recurrence relation of length N for h_k:

$$h_{N+k} = \sum_{m=0}^{N-1} q_m h_{k+m}.$$

3. Set $\{\zeta_1, \zeta_2, \ldots, \zeta_N\}$ to be roots of polynomial equation

$$\sum_{k=0}^{N} q_k \zeta^k = 0.$$

4. Solve for $\omega_1, \omega_2, \ldots, \omega_N$ as solution to linear equations

$$h_k = \sum_{n=1}^{N} \omega_n \zeta_n^k, \quad 0 \leq k \leq N-1. \quad (*)$$

Then $(*)$ also holds (by induction) for $N \leq k \leq 2M$.

Algorithmics

6.3 Source of EA (cont'd)

Shortcomings

- Numerical nullspace of H is usually very large → numerical instability.
- System $(*)$ can be extremely ill-conditioned.

Beylkin & Monzón (2005)

Replace equation $Hq = 0$ with $Hu = \sigma\bar{u}$ where $\sigma \equiv \sigma_N > 0$ and is small (entailing N large). It turns out that error of approximation

$$\max_k \left| h_k - \sum_{n=1}^{N} \omega_n \zeta_n^k \right|$$

is controlled by the smallest positive σ_N.

Algorithmics

6.4 Source of EA (cont'd)

Beylkin-Monzón Algorithm for hockey-stick function $(N \mapsto N+1 \equiv \mathcal{N},\ M = \mathcal{N})$

1. Input ϵ as given accuracy.
2. Find the smallest $\mathcal{N}$ such that $\mathcal{N} \geq \frac{1}{4\epsilon}$.
3. Compute the spectral decomposition of the matrix $\mathcal{H}_\mathcal{N}$: $\mathcal{H}_\mathcal{N} = U\Lambda U^T$. Let $u = (u_0, u_1, \ldots, u_{\mathcal{N}-1})^T$ be the last column of U. ($|\lambda| \downarrow$ down diag(Λ))
4. Find all roots $\zeta_1, \zeta_2, \ldots, \zeta_{\mathcal{N}-1}$ of the polynomial equation: $\sum_{m=0}^{\mathcal{N}-1} u_m \zeta^m = 0$.
5. Solve (least-squares) linear system, for ω_n: $h_m = \sum_{n=1}^{\mathcal{N}-1} \omega_n \zeta_n^m,\ 0 \leq m \leq 2\mathcal{N}$.
6. Compute γ_n according to $\gamma_n = 2\mathcal{N} \log \zeta_n$.

$$\mathcal{H}_\mathcal{N} := \begin{bmatrix} \mathcal{N} & \mathcal{N}-1 & \cdots & 1 \\ \mathcal{N}-1 & \mathcal{N}-2 & \cdots & 0 \\ \cdots & \cdots & & \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

Remarks:

- h considered on $[0,2]$, rescaled to $[0,1]$.
- $\frac{1}{N}\mathcal{H}_\mathcal{N}$ is upper right block of H; rest is 0s.

Algorithmics

References

1. G. BEYLKIN and L. MONZÓN, *On approximation of functions by exponential sums*. Applied and Computational Harmonic Analysis, 19 (2005), pp.17-48.

2. I. ISCOE, K. JACKSON, A. KREININ, and X. MA, *On exponential approximatons to the hockey-stick function*. Applied Numerical Mathematics (submitted; revised). http://www.cs.toronto.edu/pub/reports/na/IJKM.paper2.revised.pdf

3. I. ISCOE, K. JACKSON, A. KREININ, and X. MA, *Pricing synthetic CDOs based on exponential approximations to the payoff function*. Journal of Computational Finance (submitted; in revision). http://www.cs.toronto.edu/pub/reports/na/IJKM.paper3.pdf

4. K. JACKSON, A. KREININ, and X. MA, *Loss distribution evaluation for synthetic CDOs*. http://www.cs.toronto.edu/pub/reports/na/JKM.paper1.pdf

Algorithmics