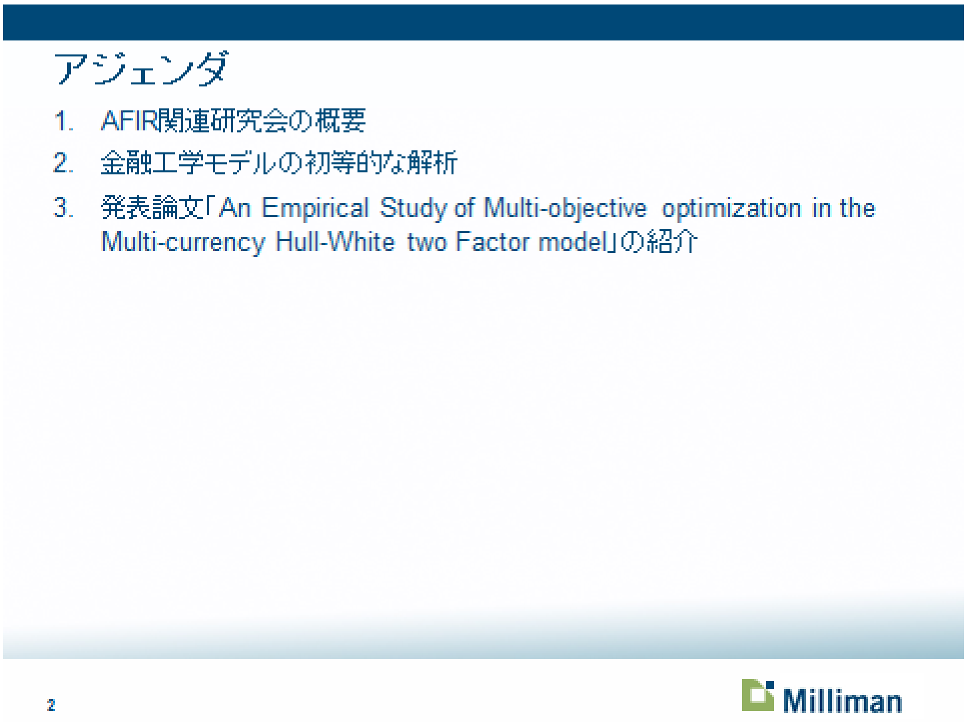
複数経済の Hull-White 2 Factor モデルにおける多目的最適化の試み

ミリマン　コンサルタント　大塚裕次朗

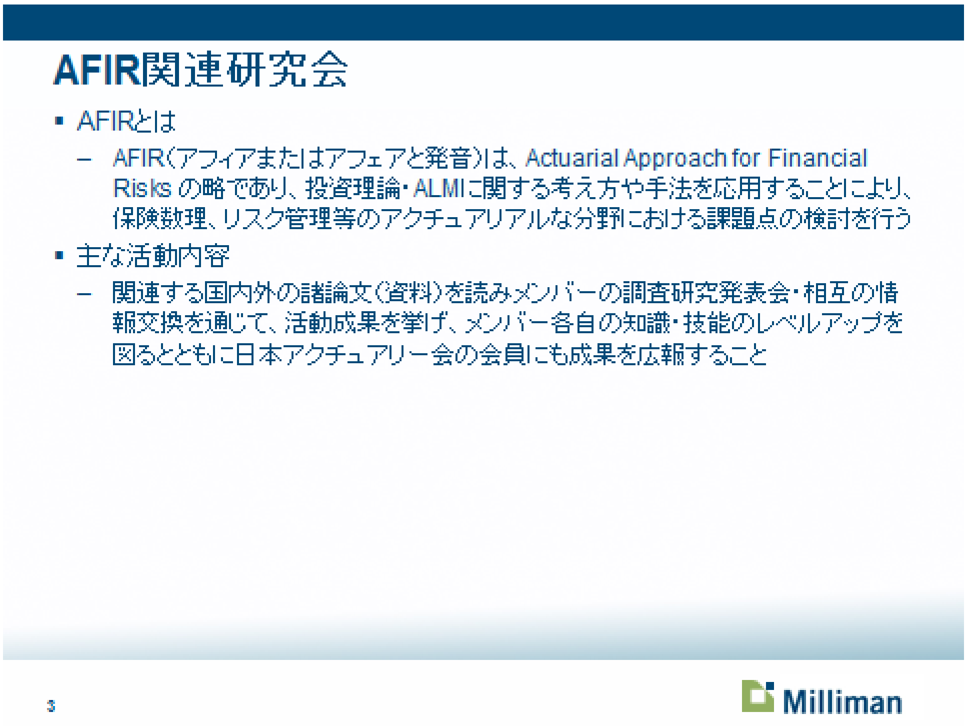
大塚　ただ今、ご紹介にあずかりました、ミリマンの大塚です。本日はＡＦＩＲ関連の研究会の代表として、「複数経済の Hull-White 2 Factor モデルにおける多目的最適化の試み」と題しまして、お話をさせていただきます。

まず先に、簡単に自己紹介をさせていただきます。ミリマンという会社は世界最大規模の独立系コンサルティング・アクチュアリー会社で、世界の主要な各都市にオフィスを設けております。私はミリマンに入って、もう６年ぐらいになるのですけれども、特にＥＶや変額年金関連の責準評価から負債経済価値の評価とヘッジ、ＦＲＭというところに関わっていて、主にプロジェクトの関わりとしては、金融工学モデルを使うというような方面からプロジェクトに関わるということが主たる業務になっております。その中で普段、課題認識しているもののうち、どちらかというと、今まで目をつぶってきたといいますか、とても難しい課題だという認識があった部分について、今回、研究発表をさせていただきます。

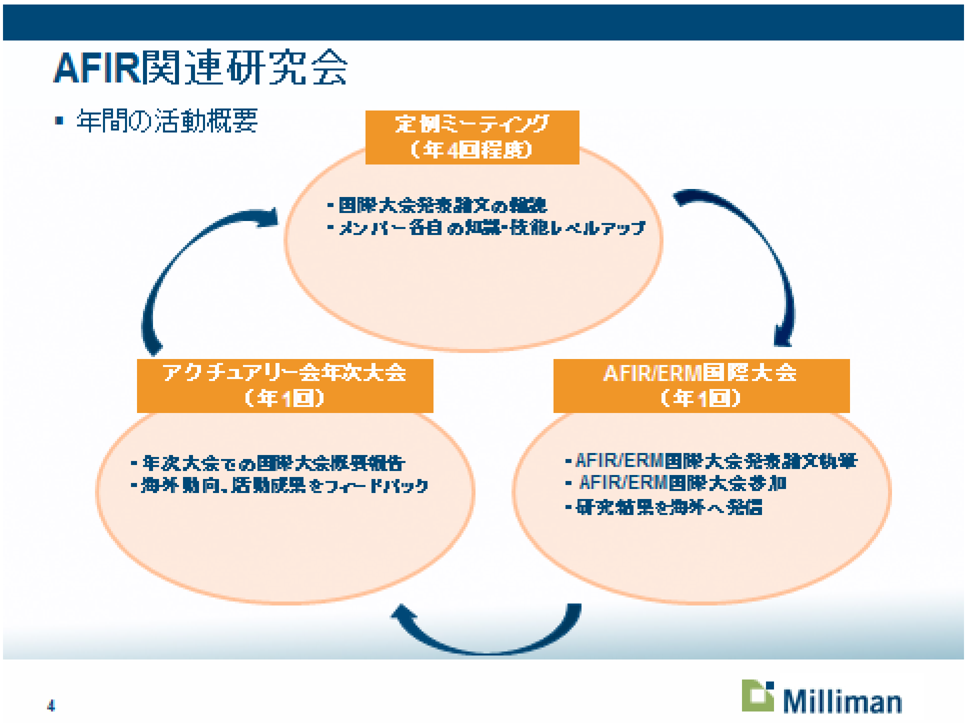


まず先に、ＡＦＩＲ関連の研究会の紹介をさせていただきます。今日の発表では最初にＡＦＩＲ関連研究会の紹介と１年間の活動報告をさせていただきます。その後、二つめに金融工学モデルの初等的な解析ということで、普段いろいろとモデルに接していると、どうしても踏み込めない部分というものがあって、ＥＶでも何でも評価をするときに、どのようなモデルを使えばいいのかということが一つ問題にあります。なおかつ、世の中にはいろいろなモデルがあって、結局どのようなモデルがいいのかよく分からない。そこに迫っていこうかということです。ここでは、わざわざ初等的な解析と書いてあるのですけれども、モデルをより抽象化して、パラメータの関数と考えたときに、そこに何か初等的な解析、要するに微分ですが、細かく切って調べてみると何か見えるのではないだろうかというような試みについて紹介します。最後に、発表論文の紹介として、ここでは英語の表記ですけれども、日本語で言えばこのプレゼンの題名の「複数経済のhull-White ２ Factorモデルにおける多目的最適化の試み」ということで、内容を紹介させていただきます。

分量がきっと多いので、途中に数式などが出てくるのですけれども、その辺はさらさらと流して、目的や背景や結論で重要な所を中心に紹介したいと思います。また、２と３は独立していますので、いったんそこで切って、質疑等があればそこでその都度受け付けたいと思います。発表の合間でも、何かご質問があれば、挙手いただければ質疑に応じます。

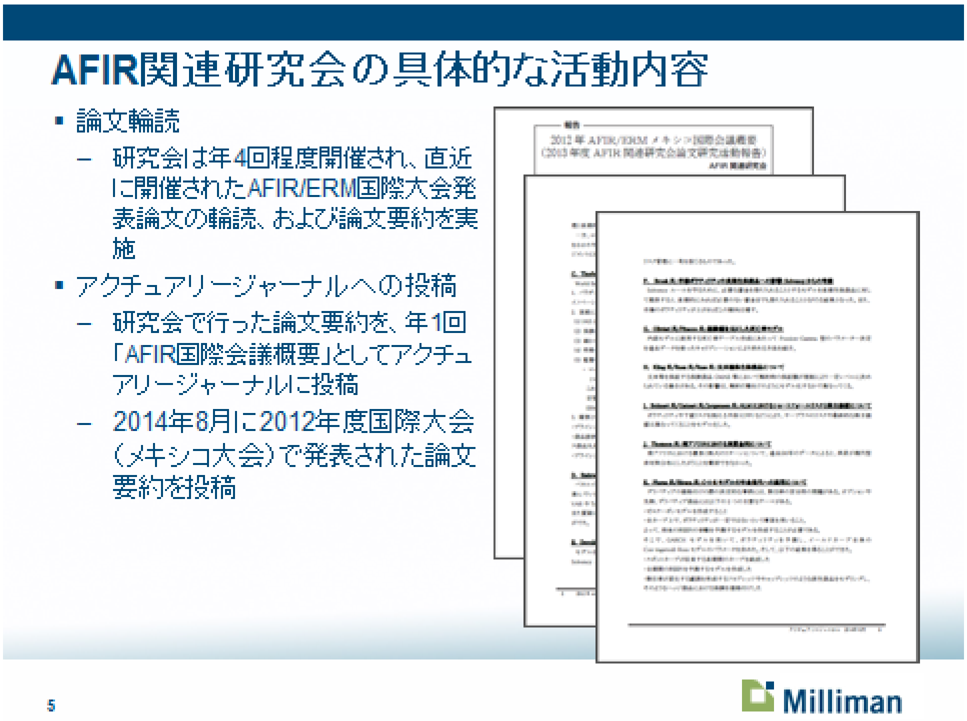


最初に、ＡＦＩＲ関連研究会です。ＡＦＩＲというのは言ってみれば投資理論、ＡＬＭに関する考え方・手法を応用することで、われわれの業界の業務の課題の検討を行うというような組織です。日本アクチュアリー会にもＡＦＩＲ関連研究会というものがありまして、そこでは主に国内外を含めて、そのような試みをした文献等を読んで、会員と共有するということをしております。



年間の活動の概要は、年４回程度の定例のミーティングが主たる活動です。この場でメンバー各自が主に前年度に行われた国際大会の論文を輪読して、メンバー同士で紹介をして、ディスカッションをする。その要約をアクチュアリー・ジャーナルに投稿するというのが主な活動です。それから、右下のＡＦＩＲ/ＥＲＭ国

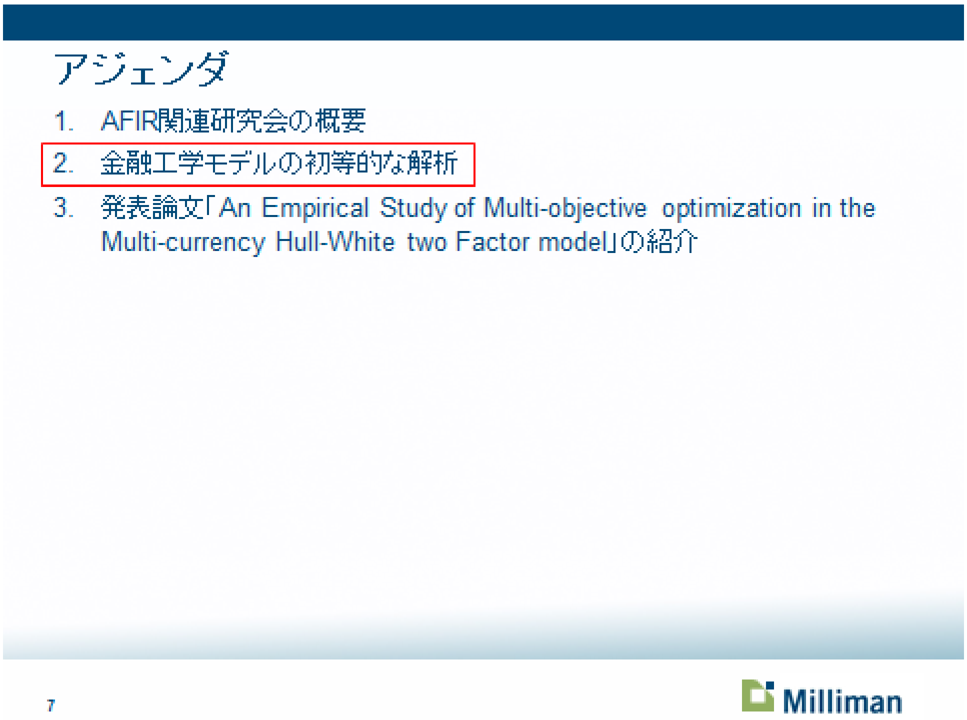
際大会が年１回あるのですけれども、ここに論文を投稿するということもします。実際には、来年度のシドニー大会で、今日発表するプレゼンテーションを投稿するということになっています。そのような１年の活動成果を左下の年次大会、それは今日ですけれども、そこで報告して、あるいはアクチュアリー・ジャーナルに投稿するというようなサイクルになっています。



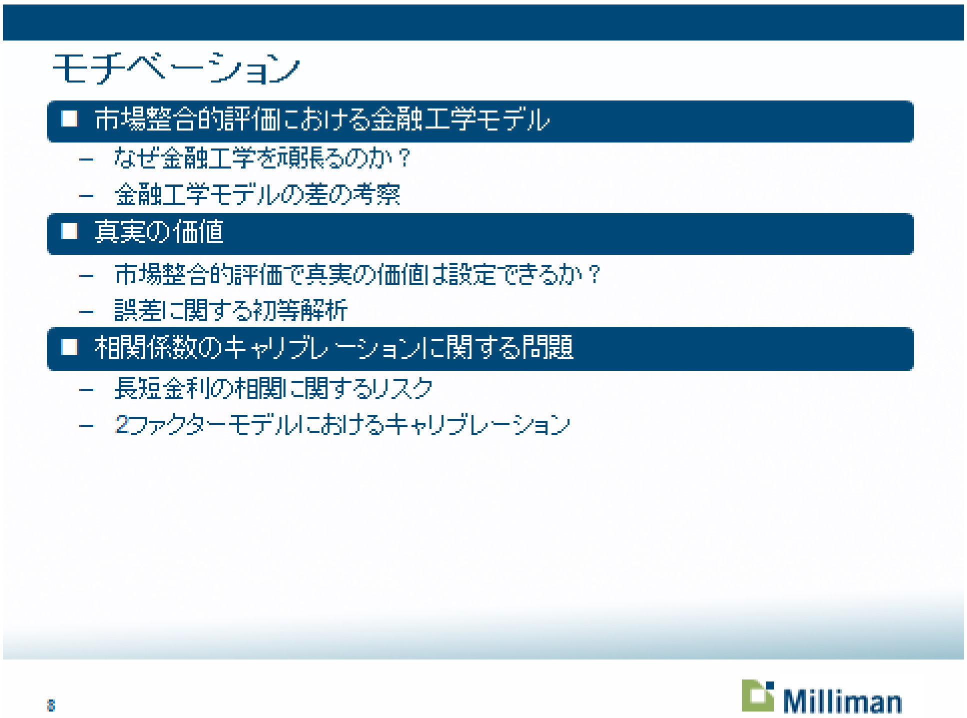
こちらは具体的に投稿した内容です。年４回のミーティングで要約を紹介して、それをまとめてジャーナルに投稿するということをしています。

ＡＦＩＲ関連研究会というのは、投資理論やＡＬＭに関してそれをどのように応用していくかというよう

なところに課題認識を持っているので、当然、ファイナンシャル・リスクマネジメントや、金融工学モデルに関する関心は、引き続き強くて、これは昨年度のリヨン大会の発表論文の中から、それに関連した題名のみを挙げています。国際的にも引き続き、このような分野に関する関心が高いということがうかがえます。



大変簡単に済ませてしまいましたけれども、中身に入っていきたいと思います。まず、次の「金融工学モデルの初等的な解析」という所に入っていきますけれども、最初にどのようなモチベーションでやっているかということをお話ししたいと思います。ここで考えたいのはまず、「なぜ、それほど金融工学を頑張るのでしょうか、われわれアクチュアリーが」というところです。

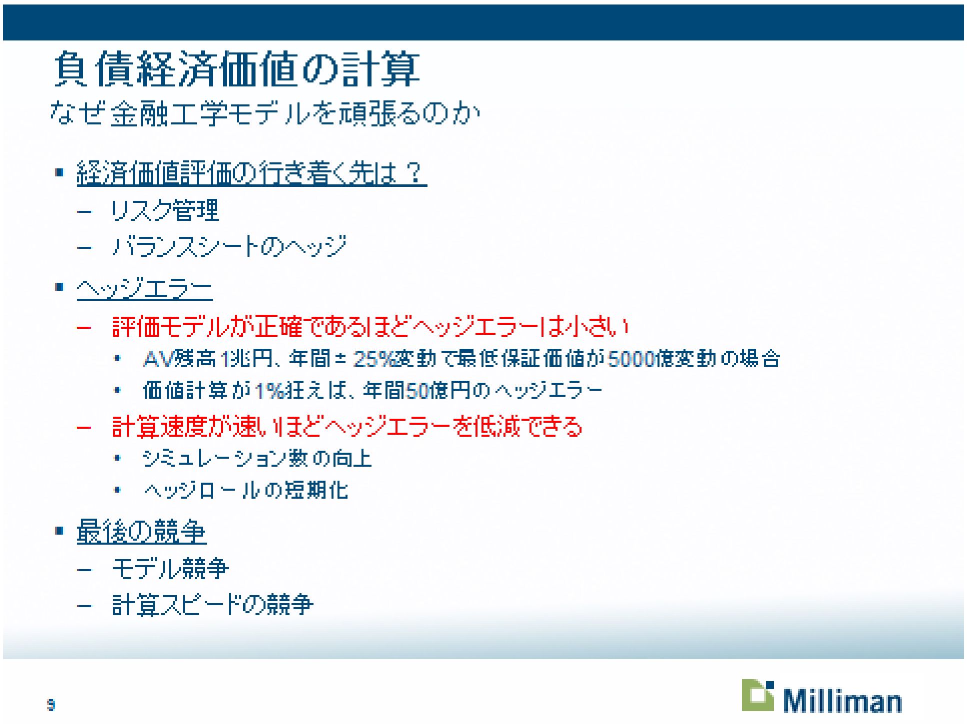


どの会社さんにも大抵１人ないし２人は、この関連の分野に非常に詳しくて、その話題になると大変口数が多くなるというような方がいらっしゃると思います。僕もミリマンでは恐らくそのようなタイプだと思います。金融工学は見た目が大変えぐいといいますか、何か難解に感じるので、それが余計に趣味的な要素というか、「あの人は好きでやってるんだろうな」というように思われてしまいがちです。もちろん、やっている方からすれば「面白いからやる」というのが唯一のモチベーションですけれども、ただ、仕事でやるからには単に面白いからということでは済まされなくて、そもそもなぜ僕たちが金融工学モデルを頑張らなければいけないのかということを考えてみたいと思います。

世の中にはいろいろな金融工学モデルがありますが、それがそもそもどのような違いがあるか。一つ一つの論文を読むと、「これはこういうモデルで、こういう所が優れていて、このモデルの方がいい」というようなことはもちろん書かれています。結局、それを「じゃあＥＶに使ってみよう」というときに、果たして本当にそのモデルがいいのか。あるいは、あるモデルとあるモデルを比較したときに、どちらがいいということを数学的に厳密な意味で言い切れるかというと、それは無理なのです。無理なので、結局どれがいいのか分からないということになります。そのような中で、またどんどん「こういうモデルがいい」というものがいろいろ出てきて、結局、本質的に状況が変わらないので、だんだん辟易してしまうといいますか、何か議論が不毛に見えてしまいます。

そこを考えていこうとすると、結局そのような不毛なサイクルに陥るのは真実の価値ということが、例えばＥＶで言えば、本当のＥＶの値とは何なのだろうかということが分からないままでいるので、それに近いか遠いかということを数字として定量的に評価できないのが原因であろうということで、その真実の価値というものは何らかの設定ができるものなのかどうかということを考えます。その上で誤差があるなら、その誤差の解析ということをしていきます。

最後の三つめのポチは、発表の方につながっていくのですけれども、相関係数に関してキャリブレーションするという問題について扱います。この詳細は後ほど述べます。



まず、金融工学モデルをなぜ頑張るのか。ずっとここ数年、負債経済価値を評価するなどといわれていま

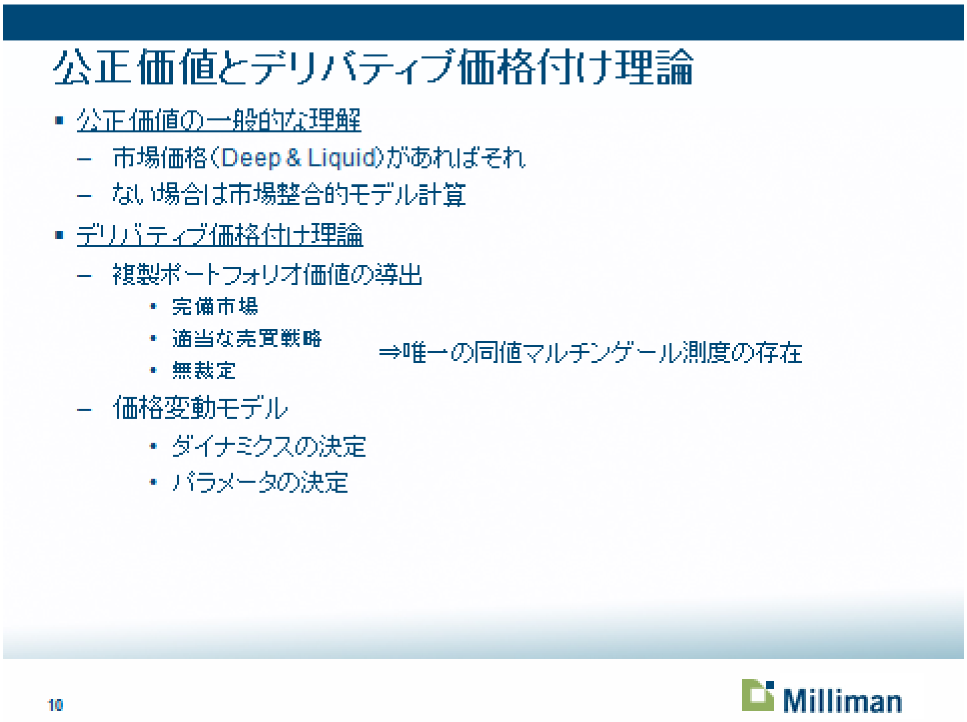
すけれども、結局、経済価値評価の行き着く先というのは何だろうかということを考えると、一つにはリスク管理ということがあると思います。いろいろなよく分からない複雑なものを、もう全部経済価値という１本の物差しでやってしまえということです。

では、その先その先というように、結局行き着くところは何かというと、要するにヘッジであろうと思います。そういう前提で、ではヘッジをするにあたって金融工学モデルというものがどのようにかかわるかというと、そのヘッジのエラーです。ヘッジには必ずエラーがあって、今はダイナミック・ヘッジのようなものを想定しますけれども、負債のグリークスにあたるものを反対サイドで複製するというのが、ヘッジの一般的な考え方です。そのヘッジ資産の評価というものは、結局負債の評価と結び付いていて、負債の評価で起こる誤差というものがヘッジ資産の誤差になります。ヘッジ資産の誤差がヘッジ・エラーになります。

そうすると、ここでは大変極端な非現実的な例ですけれども、言いたいことは、評価モデルが例えば１％正確になるというときに、恐らくそれに伴う努力は結構大変だと思います。大変というのは、いろいろな小難しいモデルを実装して、ああだこうだということが必要ですけれども、ここで言いたいのは、難しいモデルを実装しようとするときに、それは単にいたずらに難しいことをやって、個人の趣味的なところで自己満足のような気持ち悪いことをしたいわけではなくて、オーダーとして数億、年間、ヘッジのエラー損益が実際に起こるか起きないかというコンテキストで考えれば、その重要性というものが認識されるのではないでしょうか。

もう一つは、これはまたちょっと金融工学とは関係ないですけれども、こちらもわれわれアクチュアリーが世の中的にはとても遅れているのかなと思う所ですけれども、計算速度の問題です。これは計算科学という意味で、どのようにして速いコードを書くか、また、ある計算をいかに早く終わらせるかというところですが、基本的にわれわれが書くコードは大変複雑なアルゴリズムが多いので、それを速くしようとすると、コードがより複雑になってしまって、誰もメンテできなくなってしまう。そのような事情もあって、コードを速くするという努力は、なかなかわれわれはやらないのですけれども、この辺についてもゆくゆくはきっと考えていかなければいけないと思います。これについては後ほど、少し時間があれば紹介したいと思います。

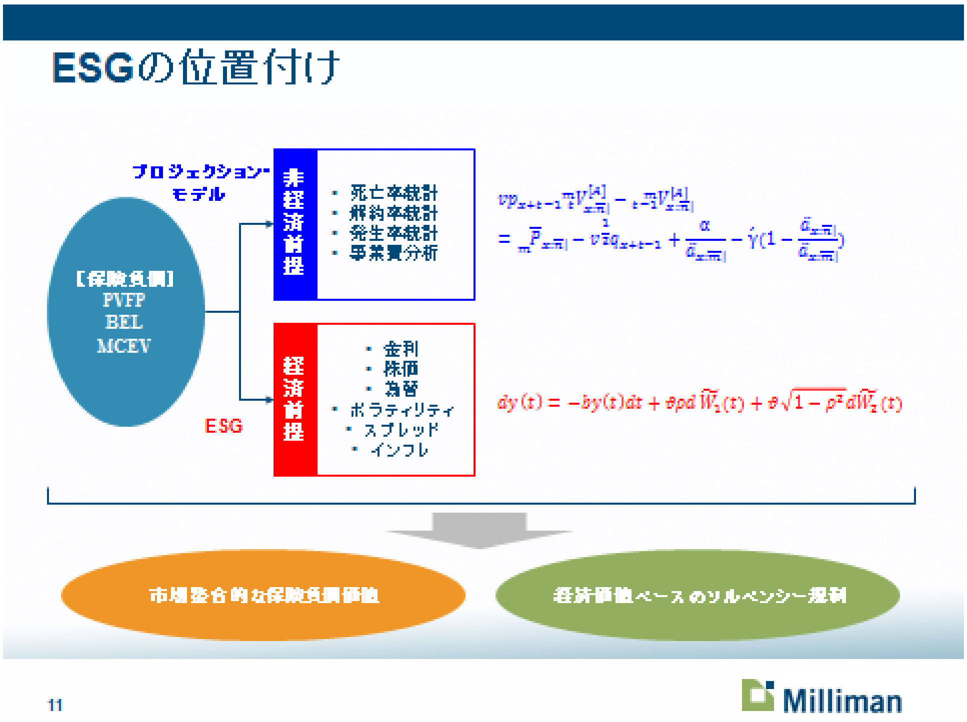
最終的にはモデルの競争と計算スピードの競争に行き着くのだろうと思います。ここで注意しなければいけないのは、金融危機の起こった原因といいますか、結局金融工学は正しいのだと過信して、あのようなことが起きたということも踏まえて、ブレーキは踏みながら、しかし進んでいくというようなことを念頭に置かなければいけません。



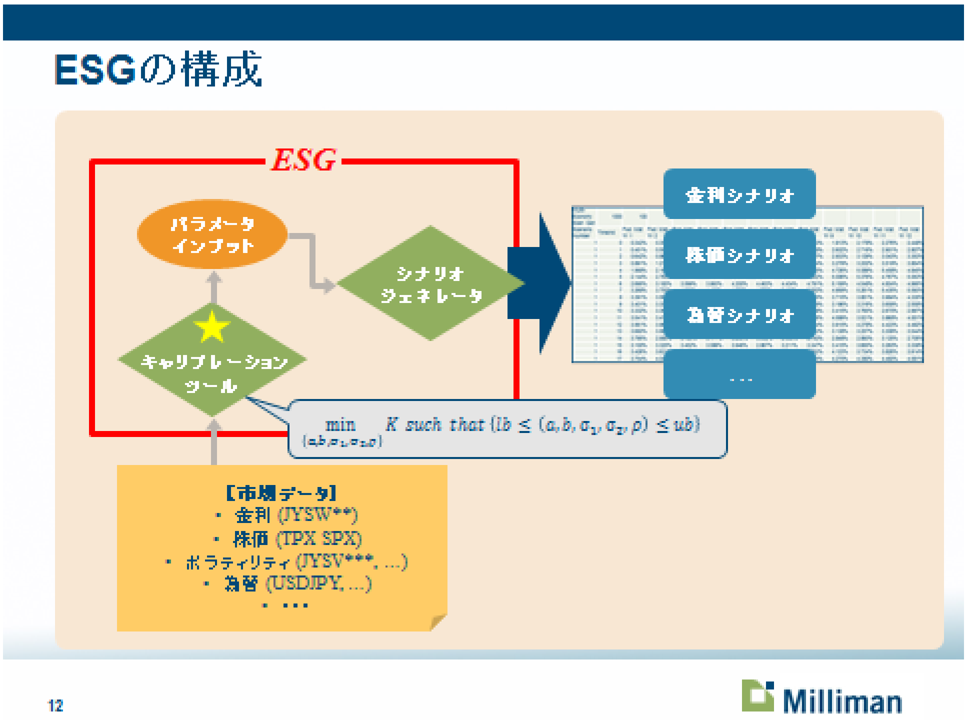
中身に入っていきます。最初に、負債の経済価値の評価というものをわれわれがどのようにやっているかということを、簡単におさらいします。ご存じの方がたくさんいらっしゃると思うので、少し冗長かもしれないですけれども、簡単におさらいします。まず、経済価値といった場合には公正価値のことです。公正価値というのは市場価格があればそれで、ただし、deep and liquid でなければいけない。ない場合には市場整合的なモデルで計算したものです。

その市場整合的なモデルで計算したものというのは何かというと、デリバティブ価格付け理論に基づいて計算されたものです。そして、デリバティブ価格付け理論というのはどのようなことかというと、まず完備な市場があって、適当な売買戦略が仮定できて、そうすると、デリバティブのペイオフは現物の資産で完全に複製できる。市場が無裁定であるとすれば、同じペイオフを持つポートフォリオは同じ価格でなければいけないから、その複製ポートフォリオの構成するためのコストがデリバティブの価格だということです。

そこでまた便利な話があって、そこに唯一の同値マルチンゲール測度というものがあって、何やらこの測度によってペイオフの期待値をとると、どういうわけか、複製ポートフォリオの価格に一致するのだという、とても便利な定理があって、結論としては、何でもかんでもリスク中立測度で評価してしまえばいいのだという、大変明快な結論が出てくる。その上で、いろいろな価格変動モデルを仮定してやっていくということになります。



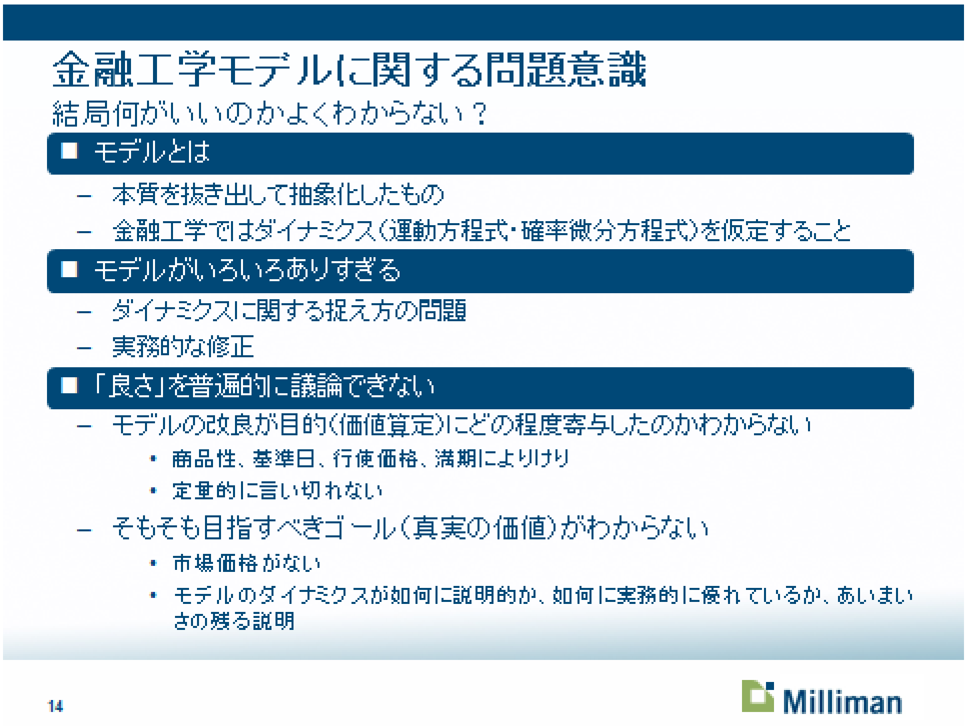
それをわれわれの業務に適用しようとするとどのようなことになるかというと、まずプロジェクションモデルです。これは伝統的な保険数理のプロジェクションモデルです。これとＥＳＧ、先ほどのリスク中立測度で確率論的なモンテカルロシュミレーションのシナリオを作るツールがあって、これと二つを組み合わせて市場整合的な価値というものを計算するというような構造になっています。



ＥＳＧに恐らくなじみのない方がほとんどだと思いますので、どのようになっているかということを簡単にご説明すると、まず市場データ、金利や株やボラティリティ、為替の情報を Bloomberg なり、証券会社なりから取得します。この市場の状況を再現できるように、パラメータを推計するということをします。それ

がキャリブレーション・ツールというもので、ここは大変難しい作業です。そこから推計されたパラメータをシナリオ・ジェネレーター本体にインプットしてボタンを押すと、中で乱数を発生させて、金利なり株価なり為替のモンテカルロ・シナリオが生成される。

その使われるモデルにどのようなものがあるかというと、われわれ的には金利モデルが一番重要なものですけれども、金利モデルを見ていくと、今日扱う Hull-White モデルや LIBOR マーケットモデルはフォワード LIBOR が対数正規分布に従うというモデルです。ブラック・カラシンスキーモデルという、金利のログとったものが Hull-White と同じカテイに従うというようなモデルなど、いろいろあります。もちろん、これだけではないです。最近では、その LIBOR マーケットモデル・プラスやシフティッド・LIBOR マーケットモデルや、ＣＩＲプラスなにがしというようなモデルが出ているなど、いろいろあって、何がどう違って、結局どれがいいのかということがよく分からない。やっている方もよく分かっていないです。



そこで、このような問題意識があります。まず、そもそもモデルというのは一体何かというと、これは考え方もあると思いますけれども、いろいろな複雑な世の中の現象から本質の部分だけを抜き出して抽象化したものである。このいい例は、ニュートン方程式です。物体の運動は質量と位置と速度で完全に記述できるという、いたって抽象化した物のとらえ方をして運動を決める。結局それが物理的には本質であったのだということです。

われわれの金融工学の中では、では、どのようにするのかというと、基本的にはダイナミクスを仮定します。ダイナミクスと言っているのは、物理で言うところの運動方程式ですし、確率微分方程式のことです。ダイナミクスをどのように仮定するかというのは、それは人によりけりなので、そのようなとらえ方の違いなどでモデルはいろいろと出てきます。

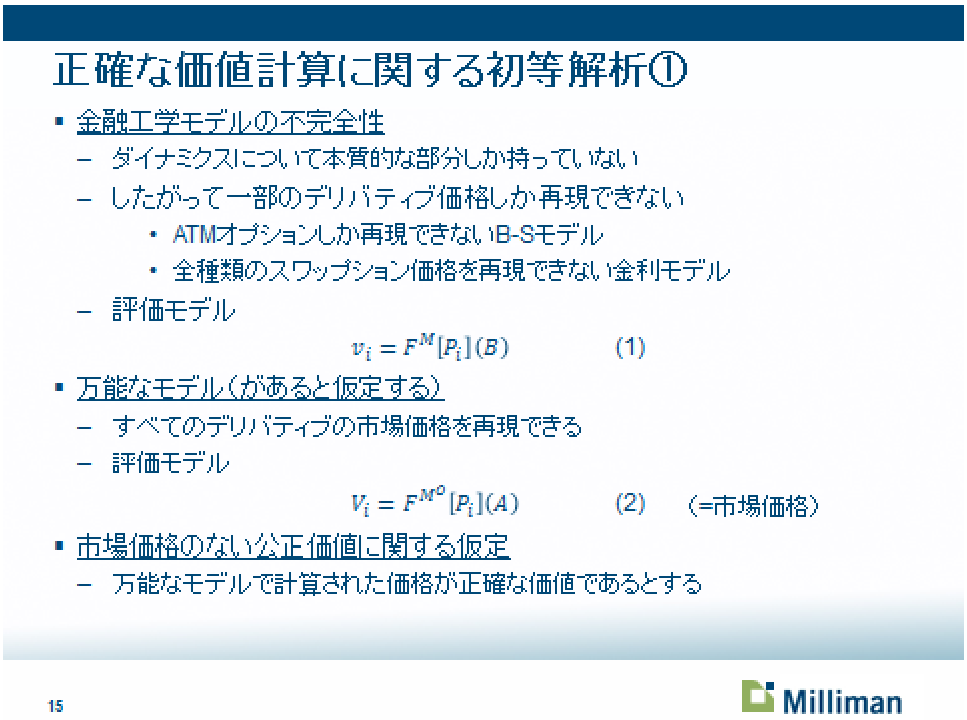
ここは、僕はそれほど好きではないですけれども、例えば「LIBOR マーケットモデルは低金利だとうまくいかないから、金利の下限を低くして、もっと０より下にしてしまえ。そうすると、うまくいくようになりますよ」というような実務的な修正というもので、モデルがいろいろとまた出てくることがあります。

そうすると、結局われわれが持つ疑問としては、モデルの良さというものを普遍的に議論することができないでいる。例えば、「ＣＩＲプラス、すごくいいですよ」と言われても、何がどのように本当にいいのかということはよく分からないです。例えば、「スワップションの再現性が、こっちのモデルの方が非常に高いから、だからこのモデルがいいんです」という言い方は一つありだと思いますけれども、では、それが本当かというと、半分うそが入っています。半分うそということは、それは数学的にはうそということになります。

これには簡単な反例があって、例えば、何らかの金利モデルをスワップションにキャリブレーションします。それを使って、何でもいいですが、キャップなりフロアなり、別のデリバティブを評価するというときには、そのような状況を考えて、今スワップションにキャリブレーションしてあるから、そちらの意味では誤差が最小化されている。ところが、では一方で、キャップやフロアの再現性を上げるためにそこからキャリブレーションしましょうというと、スワップションの再現性は下がるということになります。何が言いたいかというと、スワップションの再現性を上げるということが、イコール何か今評価したいデリバティブの

本当の価値に近づくということにはならないということです。それは簡単な反例です。要するに、評価したいＥＶというのはスワップションではないので、スワップションの再現性が上がったから正確かというと、そうではないということです。

結局、そこで何が問題を難しくしているかというと、ＥＶの本当の価値というものが、これから求めるものなのでまずモデルを検討する際にその情報がない。そうすると、そこにどれぐらい近づいたかということを議論ができないということです。結局はモデルのダイナミクスがこのような金利の動きというのはよりリアルである、あるいはスワップションの再現性がいかに優れているかというような、少しあいまいさが残ったまま、何とか「このモデルがいいですよ」と言うしかなくなってしまう。そうすると、聞く方も結局は「うーん、何とも言えないですね」というところで終わってしまう。

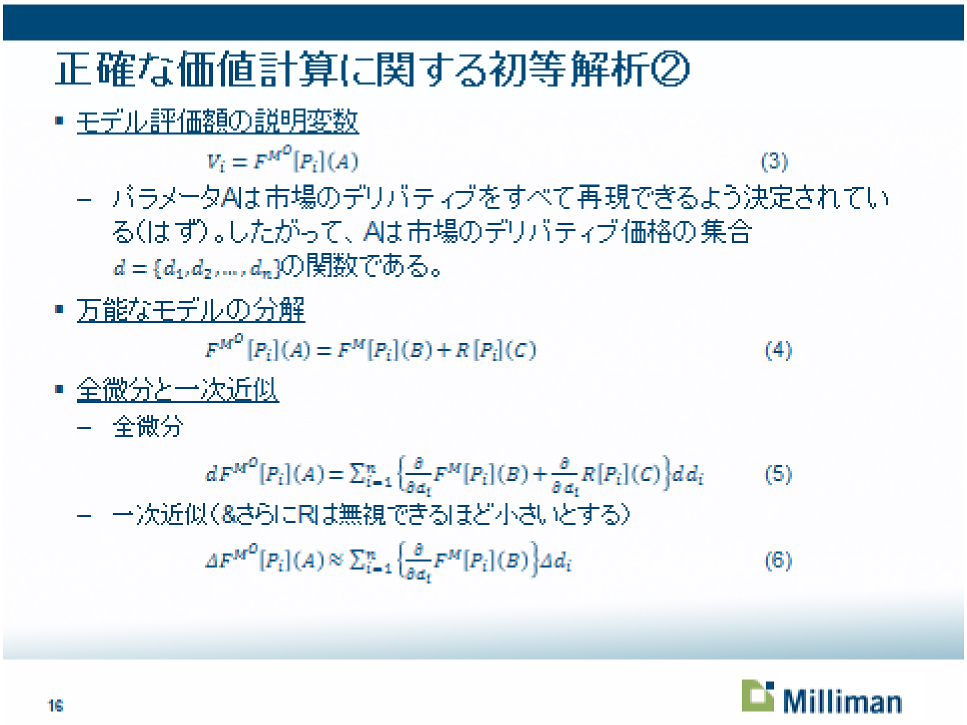


そのようなことを数式で考えてみたいのです。まず、金融工学モデルの不完全性ということで、何らかのデリバティブをモデルを使って評価するということを（1）のように表現します。説明すると、Ｆというのはフェアバリュー評価という意味のＦです。添え字のＭというのがモデルです。これは適当にＭという文字だけで置いてしまっています。Ｐはペイオフ関数です。デリバティブのペイオフ関数です。Ｂというのがモデル M のパラメータです。これだけそろえばフェアバリューの評価はできますので、そこで計算される価格を v と書いています。

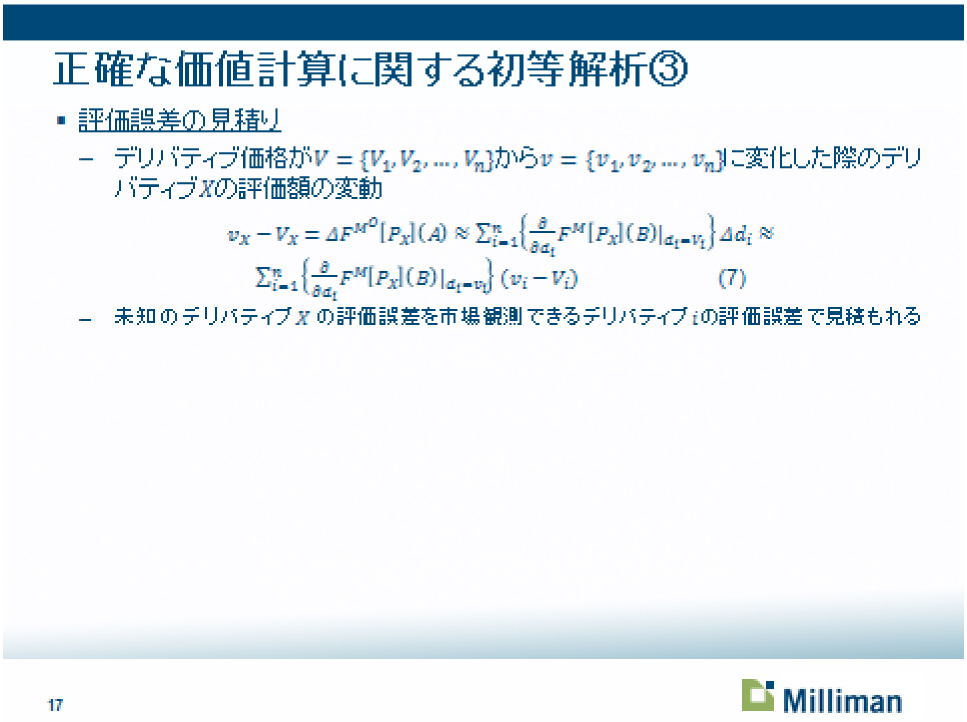
これが普段、われわれがやっていることです。ここで万能なモデルというものがあると仮定します。まず、どのようなモデルかというと、世の中のありとあらゆるデリバティブの価格を完全に再現できるというモデルが仮にあったとします。これを作れるかどうかということなどは、とりあえずさておき、一次方程式で何かよく分からない数を x と置くというようなものと同じ感覚で、とりあえずこのようなモデルがあると思うと、今度は、パラメータはモデルが違うのでパラメータのセットも違いますから、パラメータの B の集合を A で置き換えています。この A も、有限なのか無限なのか、もし無限だった場合に計算を実行できるかどうかというようなことは、ここでは全く考えていなくて、とりあえずできたとしてそこで出てきた価格を V と置きます。万能なモデルなので、もしこれがマーケットにあるデリバティブであれば、これはイコール市場

価格ということになります。

　ここで一つの仮定をします。この万能なモデルで計算されたＥＶなり何なり市場で取引のない商品の価格というのは、これが真実の価格であると仮定します。



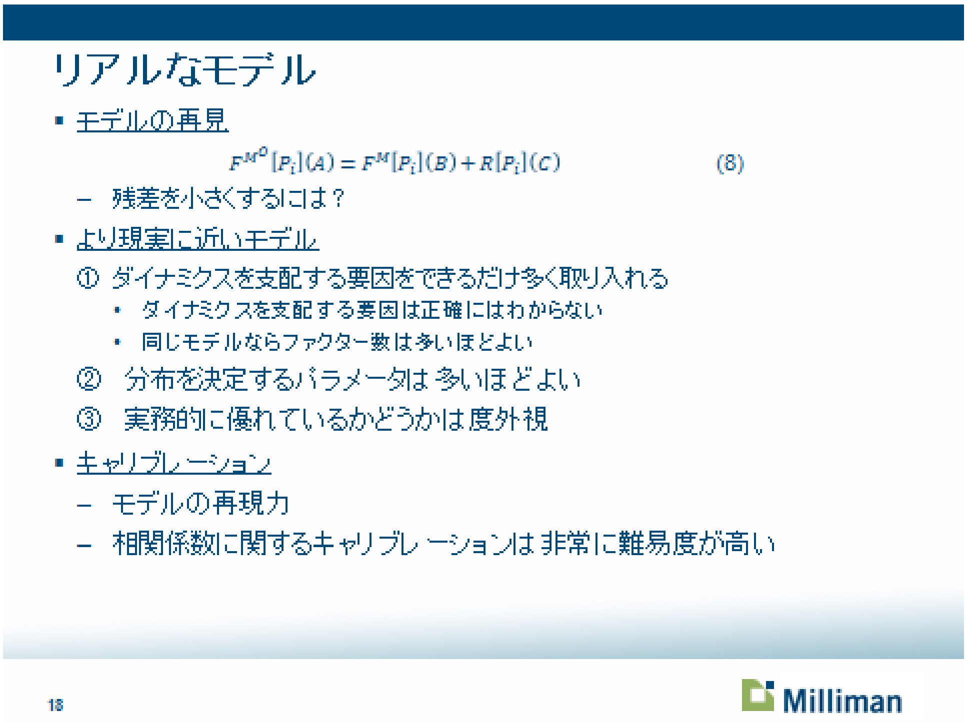
その後は普通の解析です。万能なモデルをパラメータＡの関数だと見て、でも、Ａというのはマーケットのデリバティブの関数とも考えられるから、まず万能なモデルを普通のわれわれが使うモデルと残差に分解をして、これの全微分を取るとこのように書けて、従って一次の近似としてこのように書けます。ここでちょっとずるいのですけれども、残差は無視できるということにする。この仮定は果たしていいかというと、ここはややりすぎの感はありますけれども、ただ、モデルを使う方としてはこれで金利なら金利の動きの大部分を本質的にはとらえていると信じてやるわけですから、まあいいかなと。Ｒを残したままではこれ以上、話が進まないので、Ｒを消します。



すみません、すごく飛ばしてしまいましたけれども、結論の方が大事で、まず今から評価したいあるデリバティブをｘとします。これは何でもいいです。ＥＶだと思ってください。ＥＶのＶの方なので、これは真実の価格で、ｖは何でもいいですが、Hull-White モデルなどを使って計算したＥＶだとして、その誤差というのは最後の式です。その残差は Hull-White モデルを使ってＥＶ評価した額のマーケットのデリバティブに対する感応度にキャリブレーション誤差を掛けて足し合わせたものとして近似できるというところまでいくわけです。

では、マーケットのデリバティブに対する感応度はどのように計算するのかというのは一つ問題です。これについてはいろいろと試してはみていて、今日の発表ではやらないですけれどもこれが求められたとして、そうすれば何らかの近似的な真実の価格からの誤差というものを評価できるのではないでしょうか。

もう少し簡単に、これは何をしているかというと、要するにキャリブレーション誤差がＥＶに与える影響を計算しているにすぎません。つまり、スワップションの価格、本当のマーケットの価格はＶだけれども、モデルでキャリブレーションすると、モデルの価格の方がそこから少しずれるので、例えばスワップションの価格がマーケットよりも少し上に出てしまったならば、ＥＶも少しこれぐらいはずれるかなというような影響を全部線形だと思って足し合わせているという式です。これはここまでです。



もう一度よく見てみますと、評価誤差を小さくする、イコール残差を小さくするということですけれども、そのようにするためにはどのようにモデルというのは進んでいけばいいのだろうかということを考えてみます。それは恐らく現実により近いモデルである必要があろう。例えばダイナミクスを支配するように、できるだけ多く採り入れる。あるいは分布決定するパラメータは多い方がいい。

これは一体どういうことかと言えば、例えば、株価のモデルを本当に真面目に、くそ真面目に作りたいという場合には、取引のモデルから作らなければいけないです。つまり、取引所のモデルを作って、そこに板を、「売ります」「買います」を作って、そこで何人かの人がこのような価格を付けて、その折り合いがついたところで価格が変動していく。あるいは、それに参加する人々のモデルです。トレーダーのモデルと、その人の気持ちに影響を与える要因すべてです。それは例えば天候であったり、経済であったり、社会であったり、人間関係であったり、そのような世の中すべてをモデルしなければ結局は現実のモデルとは言えなくなる。要するに、やや非現実的な話をしています。結局、モデルというのはそのようなところから本質的なところだけ抜いてきたものにすぎません。

それから、もう一つの方法があってそれはキャリブレーションの精度を上げるということです。こちらの式で言うと、この部分を小さくすれば誤差が小さくなるという論法です。結局はモデルの再現力をいかに担保するかということになります。これについては数値最適化の問題なので、それは数学の分野にお任せするとして、それとはもう少し違ったアプローチで、デリバティブの誤差を小さくするというのはそうですけれども、それとまた別の要因です。相関係数について、より現実に近い設定になるようにモデルをキャリブレーションするということも、表には出てこないですけれども、非常に重要なことです。重要なことですけれども、非常に難しいので見ないふりをするということが多いです。

金利モデルのキャリブレーション

- 無裁定金利モデルのキャリブレーションとは
  - 金利モデルが市場データを再現できるよう較正（キャリブレーション）すること

$$\min_B K = \sum \{F^M[P_i](B) - D_i\}^2 \quad (F^M[P_i](B)\text{: モデル価格}, D_i\text{: 市場価格})$$

- 実装上の論点
  - デリバティブ価格の計算負荷
    - 近似式
    - 数値積分
  - 数値最適化アルゴリズム
    - 通常は制約付き非線形最適化問題に落ち着く。
    - 最適解は多次元空間（目に見えない）、いくつ極値があるかもわからない。
    - EVに与える影響もこの時点でわからないので許容誤差もわからない。
  - ➢ **早く正確なデリバティブ価格の計算**
  - ➢ **説明可能なアルゴリズムの採用**

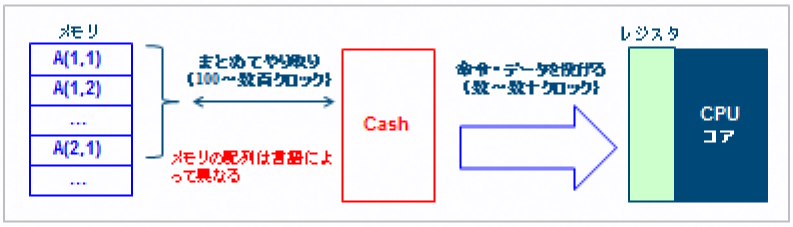
19 Milliman

ここでは金利モデルのキャリブレーションといってもよく分からない人もいらっしゃると思うので、簡単に解説します。文字が変わってDになってしまっていますけれども、マーケットのデリバティブの価格とモデルで評価した価格の誤差が最小化するように、モデルのパラメータ B を数値最適化するというのが、金利モデルのキャリブレーションです。

参考：コードの速度

処理速度向上のために

- メモリ最適化
  - 行のループと列のループ



  - VBAの場合
    - ループの行と列の入れ替えでおよそ1.6倍速度が異なる
  - C++の場合
    - ループの行と列の入れ替えでおよそ3倍速度が異なる

20 Milliman

ここまでが一応、前半です。少し時間がありそうなので、少し横道にそれた話をさせていただきます。先ほど、われわれの業務の中でいかに速く計算するかということも考えなければいけませんということを申し上げました。速く計算させるためには、結局計算するのはコンピュータなので、コンピュータがどのように

計算をしているのかが分からなければ速くしようがない。そうすると、何か工学的な知識というものも必要になってきて、なかなかとっつきづらいということがあるのですけれども、基本的なことはそれほど大した話ではないので、ちなみにということで紹介させていただきます。

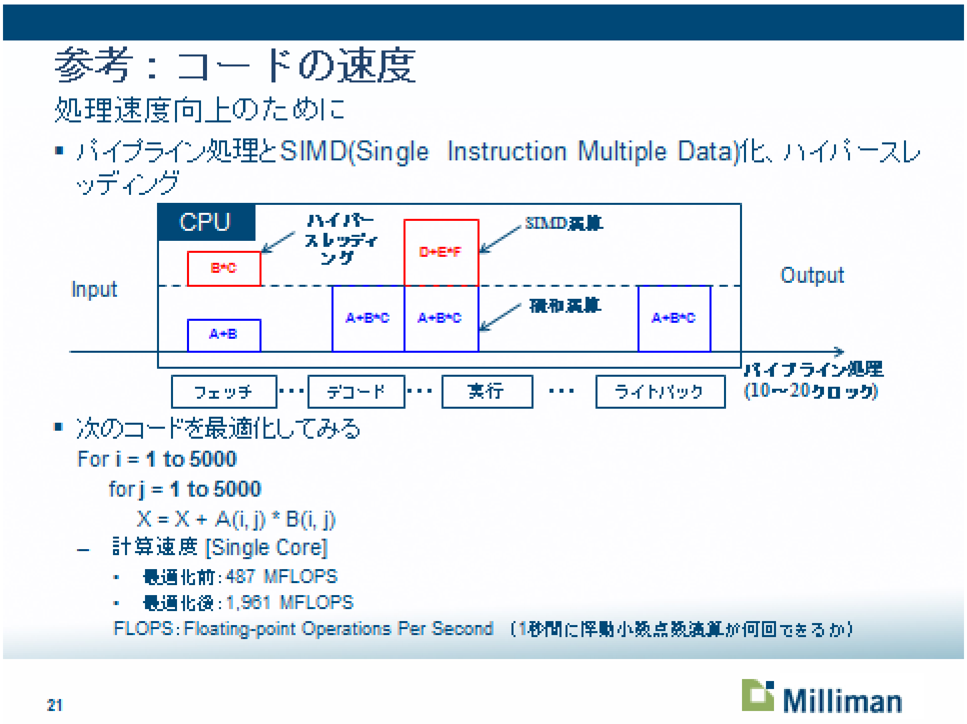
まず、コードを速くするために一番初めに考えるのは、メモリーの最適化です。例えば、まずコンピュータはどのように計算をするかというと、ＣＰＵという所で計算をするのですけれども、ＣＰＵの中にレジスタという、数字を入れるデータを入れておく所があります。ここにキャッシュメモリーという所から命令が来て、ここでデコードして、計算をして、レジスタにバックするというようなことをします。このキャッシュメモリーはＣＰＵのチップの中にあって、ここにまた外付けのメモリーからデータが運ばれます。レジスタとキャッシュの間のやり取りというのはそれなりに、クロックというのはＣＰＵの刻みですが、最近では３ギガというのは１秒間に３ギガ回のクロックがあるということですけれども、数十クロックでここのやり取りはできます。

ところが、メモリーからキャッシュに読み取るというときには、バルクでごそっとコピーをするのですけれども、ここはまとめてやり取りし、なおかつ、ここの線が弱いということがいわれていて、大抵100から数百クロックかかります。つまり、ここのやり取りに比べると、ここのやり取りは非常に重たいので、いかにこれを減らすか。このやり取りをキャッシュミスなどといいますけれども、いかにそのキャッシュミスを減らして計算するかということが、速さのポイントになります。

というのは、ここでやり取りしている間にＣＰＵは何をするかというと、何もできないのです。タスクマネージャー上では動いているようには見えるのですけれども、実際には遊んでいます。遊んでいて、ちょっと命令が来て、ちょっとやって、また、その何倍もの時間休むということをしています。だから、われわれが普通にコードを書くと、ＣＰＵの理論的な性能の数%、普通は１％、２％で、よく書けていても３％、４％ぐらいしか性能は出せていないということになります。

ちなみにかつて、できた当初は世界一だった京のコンピュータの富士通の技術者がスピードの速いコードを書いたらしいです。高速フーリエ変換か何かというものがその性能のベンチマークの演算らしいのですけれども、それをやらせたときに、97%だかの理論性能に対する実際の性能というものを引き出せたという話があります。このあたりはやり出すともうきりがなくて、非常に難解な世界になってしまうので、あまり踏み込むべき所ではないのですけれども。

ただ、われわれ的にもやれることはあって、例えばループです。ループを、例えばａという二次元の変数を作ったとして、これの行のループと列のループを同時にやる。例えば、現価を取るというようなことです。そのようなことをやるときに、「じゃあ、どっちを先にやるのが速いですか」という時に、普通に考えれば、どちらを先にやっても同じなのではないかと思うのですけれども、この辺の事情があって、メモリーには順番に並んでいるので、近いものを一気にまとめて取り出して、ここで計算してしまった方が当然速いです。そうすると、例えばＶＢでやると、２倍まではいかないですけれども、1.6倍ぐらい速くなったり、C++などの本格的なプログラミングのコードでやると、３倍ぐらい違ったりします。



このあたりが簡単にできるかなというところで、もう少し踏み込むと、このような話があります。今どきのＣＰＵは処理速度を速くするために何をやっているかというと、一つにはパイプライン処理といって、演算をいくつかの段階に分けます。インテルのＣＰＵだと 20 ぐらいの段階に分かれているといわれています。要はベルトコンベアーの処理と一緒で、インプットから、まず命令のコードを入れるという作業があって、それをデコードして、実際のコンピュータが行う演算の命令に書き換えて実行して、レジスタに返すというようなことを、20 段階ぐらいでやっているといわれています。

例えばコードを書くときに、何かある計算をして、次の行でその結果を使って何か演算をするという場合には、一番初めの命令がこちらから入ってここに出てくるまでの間、20 クロックぐらい待っていなければいけないのです。その間、ここには何があるかというと、何もなくて、これも遊んでいる状態です。では、二つのコードに関連性がなくて独立の計算になっているという場合には、最初にインプットでどかんと入れて、これがここへ運ばれている間に、すぐさまこちらに次の命令を入れられるので、速さが約２倍になるというようなことが、パイプライン処理というものです。

SIMD 化というのは、僕もそれほど詳しくないのですけれども、レジスタの長さを、普通は 64 ビットの長さのところを倍の 128 ビットにして、そこに二つのデータを入れて処理させてやれば、２倍で速いのではないかというやり方らしいです。詳しくはよく分からないです。

ハイパー・スレッディングも同じような考えで、double の処理と long の処理は別のパイプラインになっているので、そのような処理が二つ続いているのであれば、そこを二つに分けて同時に実行できるから、一つのコアで見た目上は二つのコアに見せかけることができるというような、インテルの技術です。

もう一つは積和演算というものがあります。積和演算というのはこのようなもので、A＋B×C です。これは二つの計算からなっていますけれども、まず B×C をやって、それを A と足すという二つの演算です。このような演算は今どきのＣＰＵは一度でできるような仕組みに、どのような仕組みか分からないですけれども なっているらしくて、このような積和の演算は非常に効率よく計算できるようになっているらしいです。

実際に次のコードで、X に A と B の掛け算をして足し込むというようなものを 5,000×5,000 で実行して

みると、このようなパイプライン処理やシムド化などを使わない状態で、シングルコアで計算をさせると、大体 480 何メガフロップスというスピードが出ます。これは単純に計算のスピードの単位です。これを最適化して、より効率的にコンパイルすると、4 倍ぐらいの速さになる。このあたりの最適化もコンパイラによりけりで、何とも言えないのですけれども、いかに独立の、関連のない計算をたくさん並べるかということが速さのポイントになるそうです。これは参考まででした。

ここまでで何かございましたら。

司会　ご質問は。

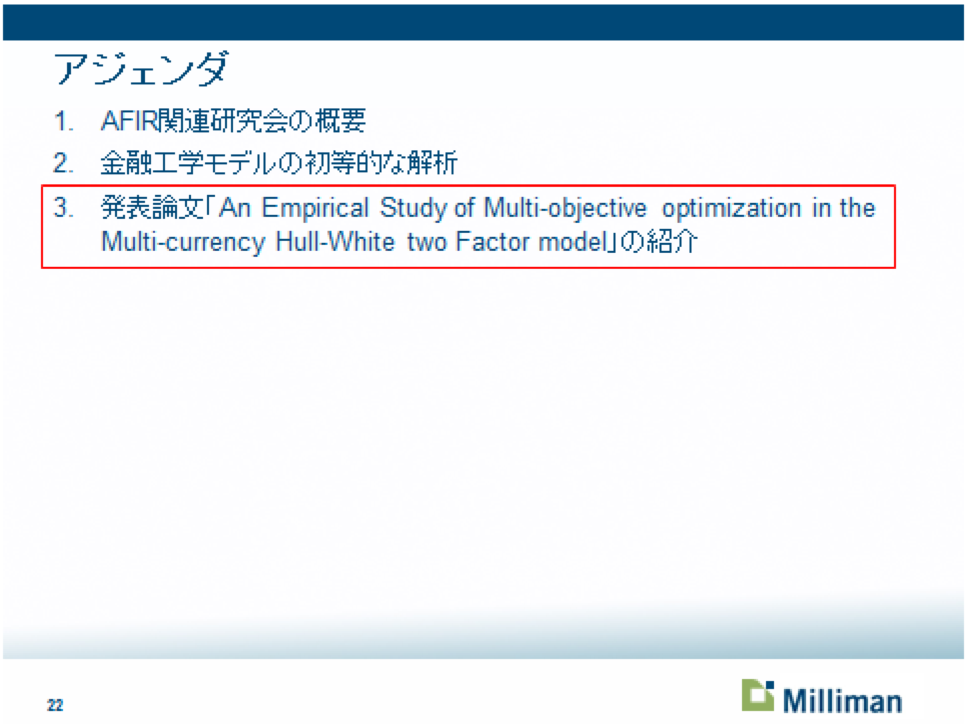
質問者　すみません、ありがとうございます。今ご紹介いただいた最適化の所ですけれども、実際に、例えば保険会社でモデリングする場合などでは、基数の計算などが典型的だと思いますが、そのような基数の計算や、将来キャッシュフロー死亡率などを使った将来キャッシュフロー展開などをするときに、多数の似たような演算のすごい数の繰り返しがたくさん出てくるというケースがあると思います。実際に今、具体的に例で出していただいた aij と bij の積和の計算で 4 倍違ったというのは、どこの技術が、例えばそのケースでは一番……。

大塚　はい。ありがとうございます。このケースでは、まずパイプライン処理です。最適化前は、一度計算を始めて、出てくるまで待って、出終わったら次の計算に行くという繰り返しです。ところが、これは最適化と言っても、コンパイラに任せて最適化したので、コンパイラがきちんと考えてくれてやってくれるかということはちょっと分からないところはあるのですけれども、最適化をすると、一度この計算……。積和の計算だから、これは違いますね。すみません、今言ったのはうそで、パイプライン処理ではなくて、この積和の演算だと思います。この一つの積和の形になっているのですけれども、これを積和の演算独自の処理に直して、一度に計算させるということをやってくれた結果かなと思います。

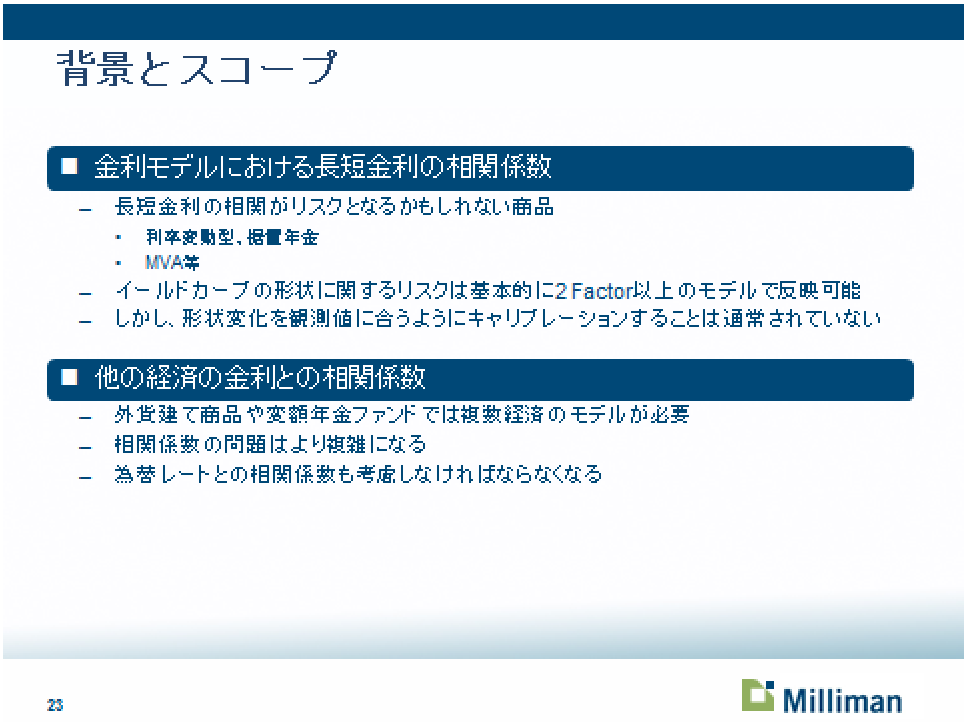
ちょっと、すみません、この中で何がどのように良かったのかということはすぐに分からないです。というのは、この最適化というのは、もちろん本当にやろうとすると、コンパイラを自分で作らなければいけなくなってしまうのですけれども、今どきのコンパイラはそれなりに自分で考えてコードの最適化をしてコンパイルしてくれるので、われわれがやることとしては、コンパイラが「ここは効率化できる計算だよ」ということを分かりやすいようにコードを書いてあげるということぐらいなのです。そういう意味では、コンパイラが何をしてくれたのかということはちょっと分からないことになります。

質問者　ありがとうございます。

司会　他にご質問はございませんか。では、続けますか。



大塚　はい。ありがとうございました。それでは本題の方、研究の発表の紹介に入っていきたいと思います。中身の数式がいろいろと複雑なので、最初に何をやりたかったのかということを口頭で説明させていただきます。



2 Factor のモデルを使いたいということが、まず先にありました。というのは、金利の 1 Factor モデルというのは基本的にはイールドカーブがパラレルに動くようなモデルですから、そこにもう一つ不確実性を加えるというのは、金利のプリンシパル・コンポーネントではないですけれども、レベルのリスクに加えて、イールドカーブの形状が変わることがリスクとなるような商品のプライシングに適している。 2 Factor のモ

デルのうたい文句がそのようなことなので、まずそれを使いたいということがあります。

実際問題、そのような商品が世の中にあるかということを真面目に考えて見ると、大してそのようなものはないのではないかと思ったりはします。つまり１Factor でいいのではないかと、正直なところはそう思います。ここで、それでも少しでも差があるのであれば、それが、では将来会社の何億の利益があるかないかということに関連するのだと思えば、「１を２に増やすぐらい、どうってことないじゃないか」という考え方もできます。そういう意味で、とりあえず２Factor のモデルを使いたいということです。

２Factor のモデルで本質的なことは、金利の形状が変わることをどのようにとらえられるかということです。これは別の言葉で言えば、長期の金利と短期の金利の相関が、１Factor ではほぼ１で動くというのに対して、２Factor では別のある程度の相関を持って動かすことができるというものです。その一つの指標として、長期の金利と短期の金利の相関係数を、例えばマーケットで観測されるような相関係数に設定をして、そうなるようにモデルをキャリブレーションするということができるでしょうかというのが、一つの問題です。

それでいくと、それはそれなりに大変ではあるのですけれども、課題としては少し易しいかなということで、ここでもう一つの経済を増やしてみる。つまり、例えば、日本円だけではなくて、ドルの金利も同時にシュミレーションするような状況を考えたい。これは例えば、外貨建ての定額の年金や、変額年金のファンドのモデリングを真面目にやりたい場合には、そのような複数経済のモデルが必要になる。そうすると、日本円の長期の金利・短期の金利と、米ドルの長期の金利・短期の金利の、４×４の相関係数のマトリックスができます。その４×４の相関係数のマトリックスをきちんと再現するように、金利モデル・パラメータを設定できるでしょうかという問題です。

それプラス、二つの経済があるので、為替レートのモデルも必要です。これは例えば、外貨建ての年金では動的解約で、通常必要になるでしょう。結局は４＋１で、５×５の相関係数のマトリックスを再現するように金利モデルのキャリブレーションができるでしょうか。キャリブレーションの目的はあくまでもスワップションの価格の再現性を上げるといいますか、スワップション価格の誤差を小さくするということが目的なので、その目的にかなう範囲で何か設定したいターゲットの相関係数をきちんと再現できるかという問題です。

フォワードレートの従う過程

The Hull-White two factor model

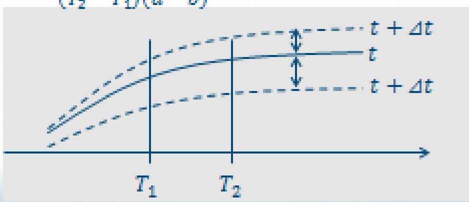
- 短期金利の従う過程

$$\begin{cases} dr = \{\theta(t) + u - ar\}dt + \sigma_1 dW_1 \\ du = -budt + \sigma_2 dW_2 \end{cases} \tag{9}$$

- 時点tにおける時点T_1と時点T_2との間のフォワード・レート(連続複利)の従う過程

$$\begin{aligned} &df(t, T_1, T_2) \\ &= \cdots dt + \frac{B(a,t,T_2) - B(a,t,T_1)}{T_2 - T_1}\sigma_1 dW_1 \\ &+ \frac{B(b,t,T_2) - B(b,t,T_1) - B(a,t,T_2) + B(a,t,T_1)}{(T_2 - T_1)(a-b)}\sigma_2 dW_2 \end{aligned} \tag{10}$$

ただし、$B(z,t,T) := \frac{1-e^{-z(T-t)}}{z}$



24 Milliman

ここからは少し嫌なスライドが続いてまいります。少し、さらさらと流します。

まず、Hull－White ２ Factor モデルはどのようなモデルかというと、(９) のような式です。これを見てもあまり学ぶところはないです。その下にいきまして、ここから出てくるフォワードレートの確率過程がこのように書けまして、「…」の所はドリフトの項で、結局ここは一意に決まるのでどうでもいいような所です。

問題はこちら側で、その将来のフォワードレートというのはｄｗ１とｄｗ２の二つのブラウン運動の足し算で書かれます。これはここだけをお分かりいただければいいです。

長短金利の相関係数

The Hull-White two factor model

- インターバルdt間における共分散
  - 瞬間的な相関係数は市場観測ができない
  - 一定の観測期間における相関係数を市場観測値等にキャリブレーションするのが自然

$$\begin{aligned} &\int_0^{dt} df(s, 0+W, 0+W+R_L)(s)df(s, 0+W, 0+W+R_S)(s) \\ &= \frac{1}{2a^3 R_S R_L}\left(\sigma_1^2 + \frac{\sigma_2^2}{(a-b)^2} - \frac{2\sigma_1\sigma_2\rho}{(a-b)}\right)\{1-e^{-aR_S}\}\{1-e^{-aR_L}\}\{e^{-2a(W-dt)} - e^{-2aW}\} \\ &+ \frac{\sigma_2^2}{2b^3 R_S R_L (a-b)^2}\{1-e^{-bR_S}\}\{1-e^{-bR_L}\}\{e^{-2b(W-dt)} - e^{-2bW}\} \\ &+ \frac{(a-b)\sigma_1\sigma_2\rho - \sigma_2^2}{ab(a+b)R_S R_L (a-b)^2}[\{1-e^{-aR_S}\}\{1-e^{-bR_L}\} + \{1-e^{-aR_L}\}\{1-e^{-bR_S}\}] \\ &\times \{e^{-(a+b)(W-dt)} - e^{-(a+b)W}\} \end{aligned} \tag{11}$$

  - 一見複雑だが計算可能

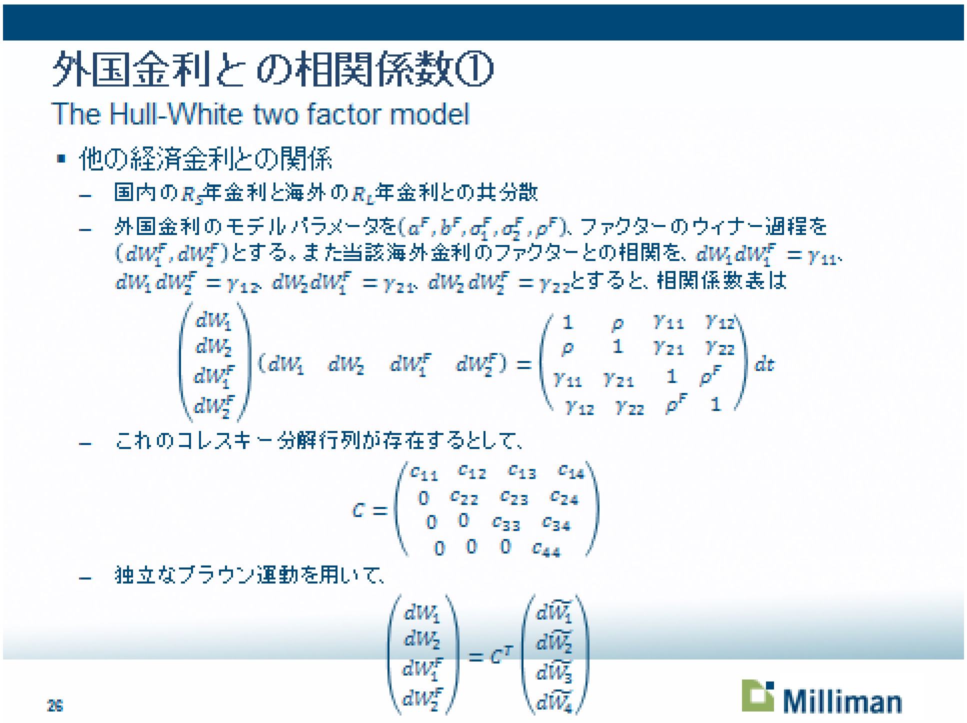
25 Milliman

そうすると、フォワードレートの確率過程が与えられたものですから、長期の金利と短期の金利の共分散が

計算できます。長期の金利・短期の金利を、例えば RL 年金利と RS 年金利と置いています。これはロングの L とショートの S です。

もう一つ言いたいのは、よくこの手の話で、瞬間的な相関係数ということが言われたりします。それは瞬間的な相関係数というのは、瞬間的な相関係数と一定期間観測しただけの相関係数というのはまた少し違うのです。それなりに結構違うので、僕はあまりそちらを使うことは自然ではないといいますか、その関係がよく分からないまま、そちらを設定するのは違うと思っています。だから、ｄｔという一定のタイムラグのある観測の間で観測される共分散をここで計算しています。

ここでは、式自体は重要ではなくて、二つのことですが、まずこの二つの長期の金利・短期の金利の共分散というのは金利モデルのパラメータ、「a, b, σ1, σ2, ρ」が金利モデルのパラメータですけれども、それの関数になっていて、解析的に計算が可能であるということです。これ自体は式は長いですけれども、普通に足し算・引き算・掛け算・割り算と、指数の計算ぐらいですから、普通にエクセルでも計算ができる。



先ほどは国内の長期金利・短期金利の共分散ですけれども、次に外国の経済です。例えば、日本円の長期の金利とＵＳＤの長期の金利の相関係数というようなものの共分散をこれから計算します。これは作法的なもので、ブラウン運動の掛け算をするときに、相関のあるものではなくて、独立のブラウン運動に直してから計算するというような作法的なものがありますので、ちょっとここはおまじない的なスライドです。結論としてはこのようになります。

外国金利との相関係数②

The Hull-White two factor model

- 他の経済金利との関係

$$
\begin{aligned}
&\int_0^{dt} df(s,W,W+R_S)(s)\, df^F(s,W,W+R_L)(s)\\
&= \{(a-b)(a^F-b^F)c_{11}c_{13}\sigma_1\sigma_1^F-(a-b)c_{11}c_{14}\sigma_1\sigma_2^F-(a^F-b^F)(c_{12}c_{13}+c_{22}c_{23})\sigma_2\sigma_1^F\\
&+(c_{12}c_{14}+c_{22})\sigma_2\sigma_2^F\}\{1-e^{-aR_S}\}\left\{1-e^{-a^F R_L}\right\}\left\{e^{-(a+a^F)(W-dt)}-e^{-(a+a^F)W}\right\}\\
&/\{a a^F R_S R_L(a-b)(a^F-b^F)(a+a^F)\}\\
&+\frac{(a-b)c_{11}c_{14}\sigma_1\sigma_2^F-(c_{12}c_{14}+c_{22}c_{24})\sigma_2\sigma_2^F}{a b^F R_S R_L(a-b)(a^F-b^F)(a+b^F)}\{1-e^{-aR_S}\}\left\{1-e^{-b^F R_L}\right\}\left\{e^{-(a+b^F)(W-dt)}\right.\\
&\left.-e^{-(a+b^F)W}\right\}\\
&+\frac{(a^F-b^F)(c_{12}c_{13}+c_{22}c_{23})\sigma_2\sigma_1^F-(c_{12}c_{14}+c_{22}c_{24})\sigma_2\sigma_2^F}{a^F bR_S R_L(a-b)(a^F-b^F)(a^F+b)}\{1-e^{-bR_S}\}\left\{1\right.\\
&\left.-e^{-a^F R_L}\right\}\left\{e^{-(a^F+b)(W-dt)}-e^{-(a^F+b)W}\right\}\\
&+\frac{(c_{12}c_{14}+c_{22}c_{24})\sigma_2\sigma_2^F}{b b^F R_S R_L(a-b)(a^F-b^F)(b+b^F)}\{1-e^{-bR_S}\}\left\{1-e^{-b^F R_L}\right\}\left\{e^{-(b+b^F)(W-dt)}-e^{-(b+b^F)W}\right\}
\end{aligned} \tag{12}
$$

27 Milliman

これは日本円の RS 年金利と、添え字の F を付けているのはフォーリンの F ですが、外国金利の RL 年金利の共分散を、ｄｔの間の時間のラグで計算したものです。ここでもこの式自体は大して学ぶところはなくて、重要なのは、まずこれら二つの金利モデル、すみません、添え字の F と付いている物は全部ＵＳＤの方の金利モデル・パラメータですけれども、これらの「a, b, σ1, σ2, ρ」の関数になっていて、計算ができるということです。

為替レートとの相関係数

The Hull-White two factor model

- 対数正規過程との共分散

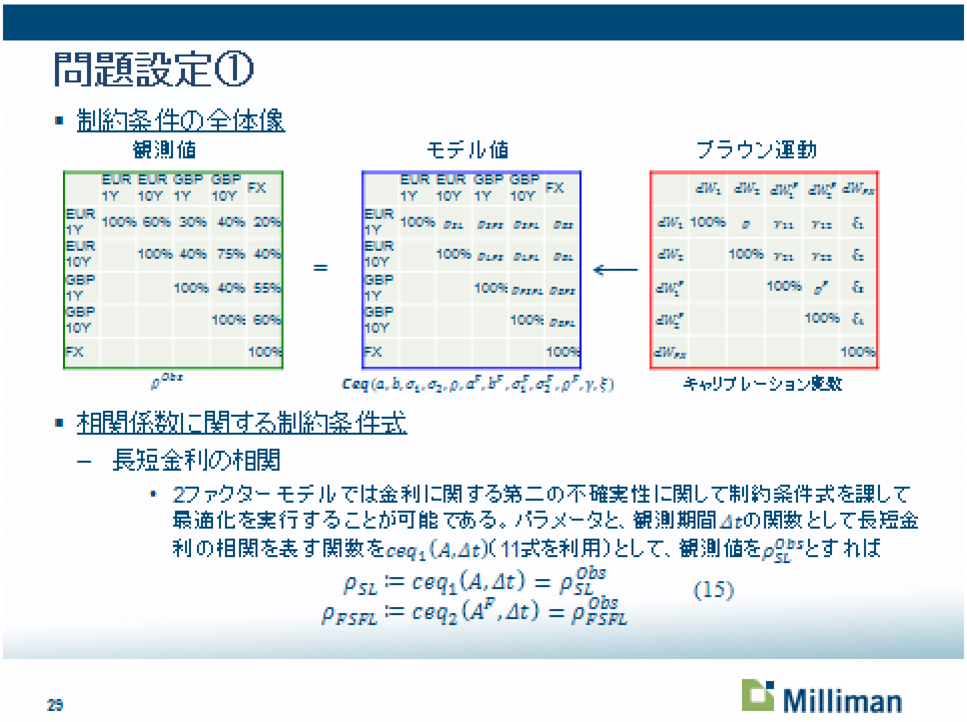
$$dS(t)=\mu S(t)dt+\sigma_E S(t)dW_E \tag{13}$$

  - における対数利回りとR_S年金利の相関は、$dW_E dW_1=\rho_{E1}$、$dW_E dW_2=\rho_{E2}$として、

$$
\begin{aligned}
&\int_0^{dt} df(s,0+W,0+W+R_S)\, d\{\log(S(s))\}\\
&=\frac{(a-b)\rho_{E1}\sigma_1\sigma_E-\rho_{E2}\sigma_2\sigma_E}{a^2R_S(a-b)}\{1-e^{-aR_S}\}\{e^{-a(W-dt)}-e^{-aW}\}\\
&+\frac{\rho_{E2}\sigma_2\sigma_E}{b^2R_S(a-b)}\{1-e^{-bR_S}\}\{e^{-b(W-dt)}-e^{-bW}\}
\end{aligned} \tag{14}
$$

28 Milliman

最後に、為替レートとの相関係数も同様です。ここはさらさらと流させていただきます。



問題の設定です。今からやりたいことは、その目的関数はスワップションの誤差を最小化するということです。長期の金利と短期の金利は1年と10年というように勝手に置いていますけれども、これは何年でも構いません。実際の市場の観測値として、今ユーロとイギリスポンドで考えていますけれども、これらの相関係数がある。こちらのこの真ん中の四角は、モデルで計算される相関係数です。これは今まで見てきたような算式で計算されます。ちょっと書ききれないので、ローで表しています。そのような計算モデルから計算される相関係数があって、左二つがイコールになるように、二つの金利モデルと為替レートのブラウン運動の相関係数を決めなければいけません。基本的には等式の数に対して変数の数が一緒なので、それほど変なものでなければ解が求められるはずです。それを今からやりたい。そのような問題設定をここから算式で表していますけれども、やりたいことはそのようなことです。

問題設定②

- 相関係数に関する制約条件式
  - 外国金利との相関
    - 第二通貨の金利モデルパラメータを$A^F=\{a^F,b^F,\dots\}$とし、第一通貨と第二通貨それぞれの2ファクターのブラウン運動の相関係数が

$$\begin{pmatrix} dW_1 \\ dW_2 \\ dW_1^F \\ dW_2^F \end{pmatrix} (dW_1\ dW_2\ dW_1^F\ dW_2^F) = \begin{pmatrix} 1 & \rho_1 & \gamma_{11} & \gamma_{12} \\ \cdots & 1 & \gamma_{21} & \gamma_{22} \\ & \cdots & 1 & \rho_2 \\ & & \cdots & 1 \end{pmatrix} dt = \mathbf{P}dt \qquad (16)$$

であるとし、観測期間Δtにおける第一通貨のS/L年金利と第二通貨のS/L年金利の相関係数が同様に$ceq_j(A,A^F,P,S,L,\Delta t)$(12式を利用)のように書けるとして、

$$\begin{pmatrix} \rho_{SFS} \\ \rho_{SFL} \\ \rho_{LFS} \\ \rho_{LFL} \end{pmatrix} := \begin{pmatrix} ceq_3(A,A^F,P,S,S,\Delta t) \\ ceq_4(A,A^F,P,S,L,\Delta t) \\ ceq_5(A,A^F,P,L,S,\Delta t) \\ ceq_6(A,A^F,P,L,L,\Delta t) \end{pmatrix} = \begin{pmatrix} \rho_{SFS}^{Obs} \\ \rho_{SFL}^{Obs} \\ \rho_{LFS}^{Obs} \\ \rho_{LFL}^{Obs} \end{pmatrix} \qquad (17)$$

のような観測金利と一致するような制約条件を課して最適化を実行することができる。

30 Milliman

問題設定③

- 相関係数に関する制約条件式
  - 為替リターンとの相関係数
    - 為替過程のブラウン運動と第一第二通貨それぞれのブラウン運動との相関が

$$\begin{pmatrix} dW_1 \\ dW_2 \\ dW_1^F \\ dW_2^F \end{pmatrix} dW_{FX} = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{pmatrix} dt = \xi dt \qquad (18)$$

として、同様に14式を利用して、

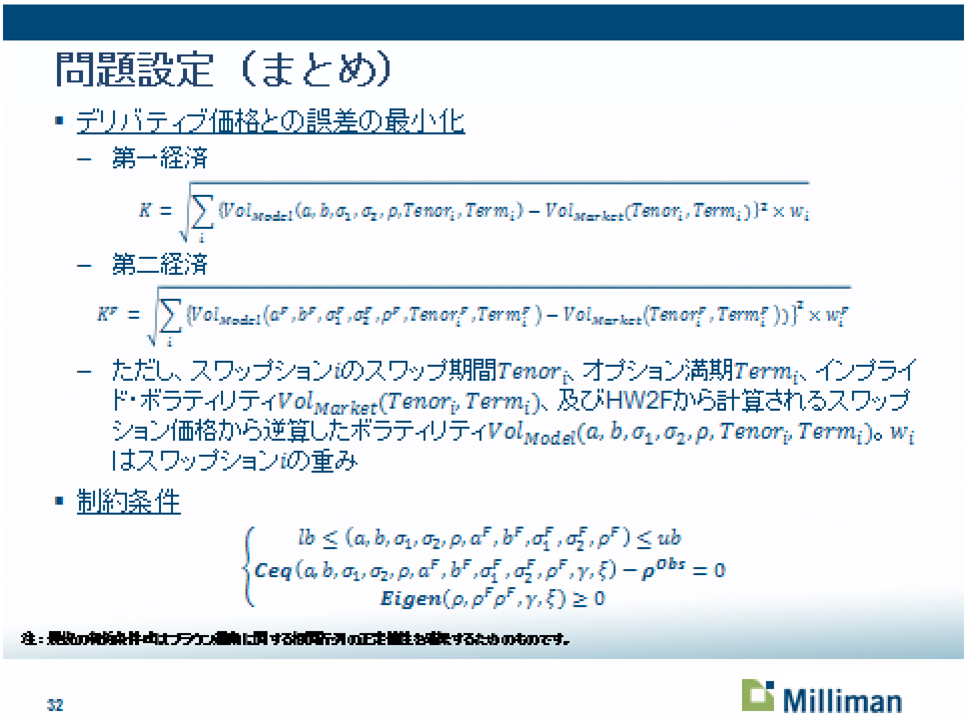
$$\begin{pmatrix} \rho_{ES} \\ \rho_{EL} \\ \rho_{EFS} \\ \rho_{EFL} \end{pmatrix} := \begin{pmatrix} ceq_7(A,\xi_1,\xi_2,S,\Delta t) \\ ceq_8(A,\xi_1,\xi_2,L,\Delta t) \\ ceq_9(A^F,\xi_3,\xi_4,S,\Delta t) \\ ceq_{10}(A^F,\xi_3,\xi_4,L,\Delta t) \end{pmatrix} = \begin{pmatrix} \rho_{ES}^{Obs} \\ \rho_{EL}^{Obs} \\ \rho_{EFS}^{Obs} \\ \rho_{EFL}^{Obs} \end{pmatrix} \qquad (19)$$

のように、S/L年金利と為替リターンとの相関係数の観測値と一致するように制約条件を課して最適化を実行する。

  - 留意点
    - ブラウン運動から連続的に計算される共分散と離散化モデルの正規乱数の共分散は一般に異なり、その影響の処理が必要です

31 Milliman

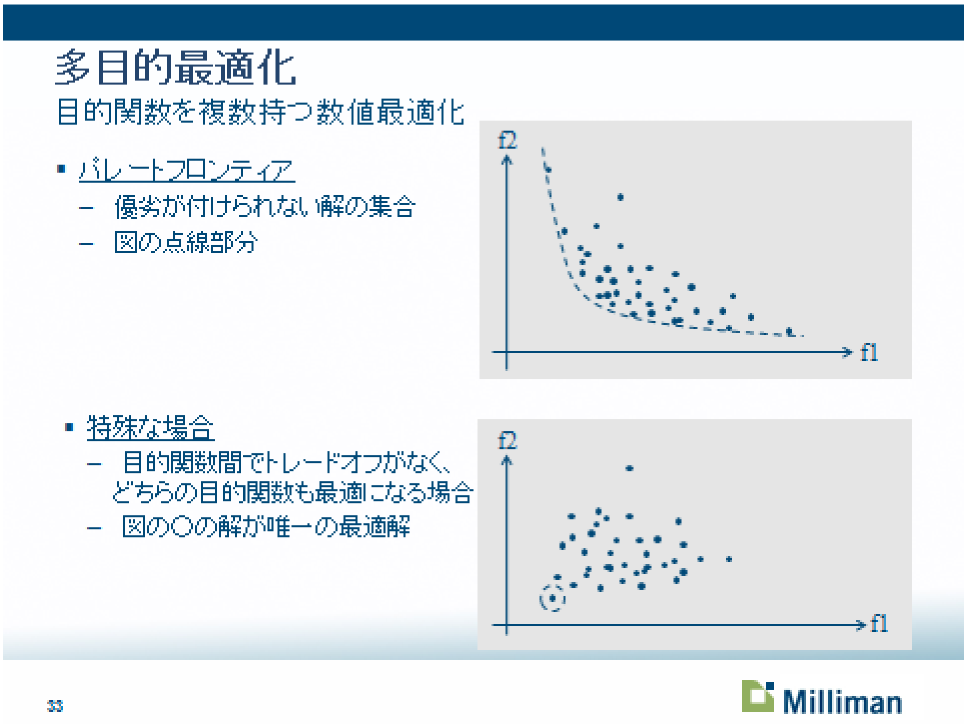
このあたりは、ぱっと飛ばします。



最後のまとめです。問題設定のまとめです。まず目的関数は、第一の経済は今の問題設定ではユーロですけれども、ユーロのスワップションのモデルの価格とマーケットのモデルの価格の２乗の誤差を全部最小化するというような方式です。別の方式も考えられますけれども、とりあえずこのような設定にします。第二経済イギリスポンドの方のスワップション価格についても同じです。このＫとKF、この二つを今から最小化したい。

そして、制約条件としてはどのようなことになるかというと、先ほど申し上げたとおり、相関係数に関する、このｃｅｑというのがモデルの相関係数です。今これはベクトルっぽく表示していますけれども、これは先ほどの行列です。これと実際のマーケットの観測値がイコールになるという制約を置いて、その制約条件付きの数値最適化をしたいということです。

注意点としては、最終的に求められるブラウン運動の相関係数というのは、これは相関係数表でなければいけないので、正定値の行列でなければいけません。だから、固有値が全部正になるという制約条件も必要になります。その他いろいろと技術的な問題はあるのですけれども、省略します。

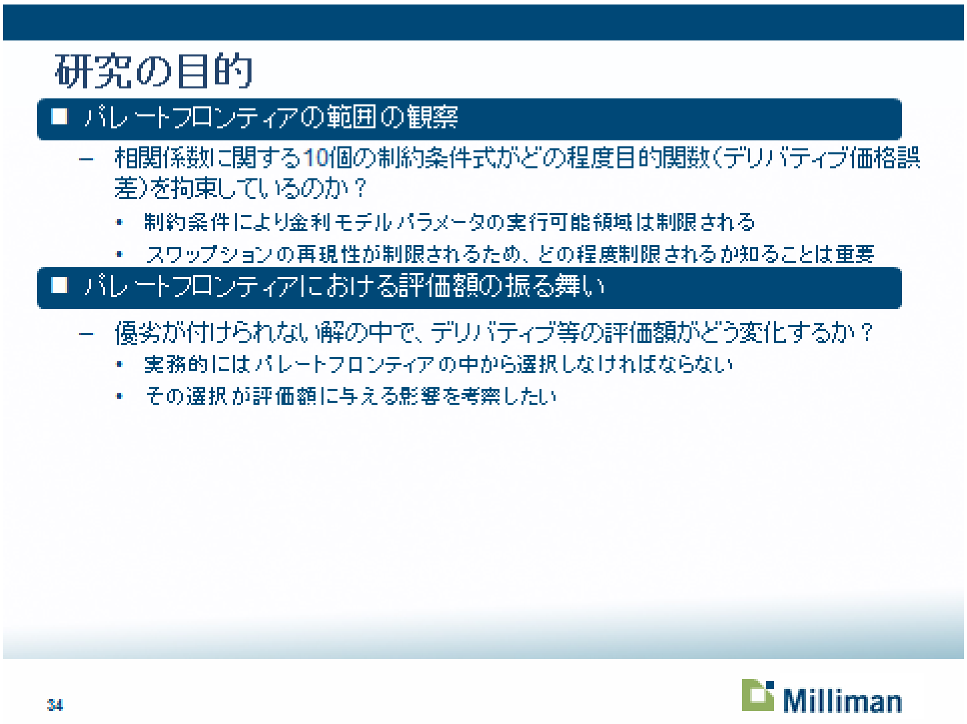


ここで何を言いたいかというと、先ほどの目的関数は二つあるのですけれども二つの目的関数を同時に最適化しなければいけません。これは多目的最適化といいまして、普通であれば日本円の金利モデルのキャリブレーションと、ＵＳＤの金利モデルのキャリブレーションを独立にやるのですけれども、今回は独立にできないです。なぜかというと、相関係数表は全部の金利モデル・パラメータの関数になっているので、一方の最適化をしてそこで変数が動いてしまうと、相関係数表がまたずれてしまうので、結果的に第二の経済の方のスワップションの目的関数が違うことになるというようなことがある。要するにトレードオフの関係です。二つを独立に扱えないということです。この二つを同時に最適化しなければいけないのですけれども、そのような問題を多目的の最適化といいます。

多目的最適化については、それなりに研究が進んでいるのか分からないですけれども、一般的にいわれるのは、パレートフロンティアという優劣がつけられない解の集合が得られるということは知られています。そのパレートフロンティアというのは、要するに、一方の目的関数を犠牲にしない限り、自分の目的関数を最小化できないというような場合です。一方を悪くしなければ、一方が良くならないというような問題です。簡単に言えば、３個のケーキを５人で分けるというような問題の場合に、例えばＡ君・Ｂ君・Ｃ君にあげてしまうとＤ君はもらえない。Ｄ君がもらうためには、例えばＣ君が犠牲になってもらわなければいけない。ではどれが正しい解なのかというと、分からない。そのようなケースです。

これの特殊な場合として、例えば、３個のケーキを２人で分ける。２人とも１個しか食べたくないというような場合では、完全に解は最適な解が見つかる。そのような場合には、この特殊なケースです。どちらも犠牲にせずに最適な解が得られるというのが、特殊な場合としてあります。

今回扱う二経済のモデルのキャリブレーション結果が、果たしてどちらになるでしょうかというのが、まず一つの興味としてあります。きちんときれいなパレートフロンティアが出てくるのかというところです。こちらで説明します。



ここでやっと研究の目的になります。

まず、相関係数に関する 10 個の制約条件式です。制約条件式の個数としては結構多い方だと思います。制約条件式であるからには、パラメータが動ける範囲を制約してしまっているので、その影響がどの程度、目的関数の解の探索範囲を制限しているのかということが一つ気になるところです。パレートフロンティアが出るのであれば、上限・下限という動く範囲が大きいのか小さいのかということが一つ興味のあるところです。

それを知った上で、今度は実際に実務で使う場合にはその中から一つを選ばなければいけないので、どれを選ぶかで評価額がどれぐらい変わるのかということを知らなければいけないです。つまり、どれを選ぶかで結果が全然違うのであれば、そもそも信用することができません。でも、逆にそれほど変わらないのであれば、どれを選んでもいいのだから、適当なものを 1 個選べばいいということで、実務的には扱いが全く違ってきます。この 2 点について調べてみました。

キャリブレーションの実行

- スワップレート
  - EUR & GBP
  - 基準日：2013年12月末
- スワップション

EUR

Option Ter	Swap Tenor 1	5	10	15	20
1	81.40%	44.74%	30.24%	25.51%	23.67%
5	37.88%	27.89%	25.01%	23.59%	24.01%
7	29.28%	25.02%	23.47%	23.12%	22.84%
10	23.71%	22.80%	22.47%	21.57%	21.72%
15	21.44%	23.07%	23.30%	21.51%	20.20%
20	22.80%	23.33%	23.02%	20.29%	18.43%

GBP

Option Ter	Swap Tenor 1	5	10	15	20
1	55.63%	35.95%	26.16%	22.40%	19.97%
5	30.20%	23.12%	19.94%	18.79%	18.08%
7	23.90%	19.73%	17.96%	17.34%	16.75%
10	19.52%	17.28%	16.30%	15.98%	15.52%
15	17.76%	16.52%	15.44%	14.88%	14.21%
20	18.03%	16.29%	15.12%	14.32%	13.49%

35 Milliman

使った前提条件としては、ユーロとイギリスポンドを使いました。なぜ、この二つにしたかというと、来年シドニーで発表するということもあって、あまりＪＰＹ、ＵＳＤというように持ち出さない方がいいのかどうなのかという、それだけのことです。きっとこの二つを持ち出した方が、この手の人たちは喜ぶかなというぐらいです。12 末のスワップレートとスワップションの前提を使ってキャリブレーションしました。

キャリブレーションの実行

- 相関係数

	EUR 1yr	EUR 10yr	GBP 1yr	GBP 10yr	GBPEUR
EUR 1yr	100%	60%	30%	40%	20%
EUR 10yr	60%	100%	40%	75%	40%
GBP 1yr	30%	40%	100%	40%	55%
GBP 10yr	40%	75%	40%	100%	60%
GBPEUR	20%	40%	55%	60%	100%

- 目的関数

$$OF = Weight \times K + (1 - Weight) \times K^F$$

ただし、$K = \sqrt{\sum_i \{Vol_{Model}(a, b, \sigma_1, \sigma_2, \rho, Tenor_i, Term_i) - Vol_{Market}(Tenor_i, Term_i)\}^2 \times w_i}$

$$K^F = \sqrt{\sum_i \{Vol_{Model}(a^F, b^F, \sigma_1^F, \sigma_2^F, \rho^F, Tenor_i^F, Term_i^F) - Vol_{Market}(Tenor_i^F, Term_i^F)\}^2 \times w_i^F}$$

注：ここでは二つの目的関数を加重平均して単目的化を行っています。一般に単目的化によりパレート最適な解が得られることは、各々の目的関数が凸関数の場合にのみ証明されています。

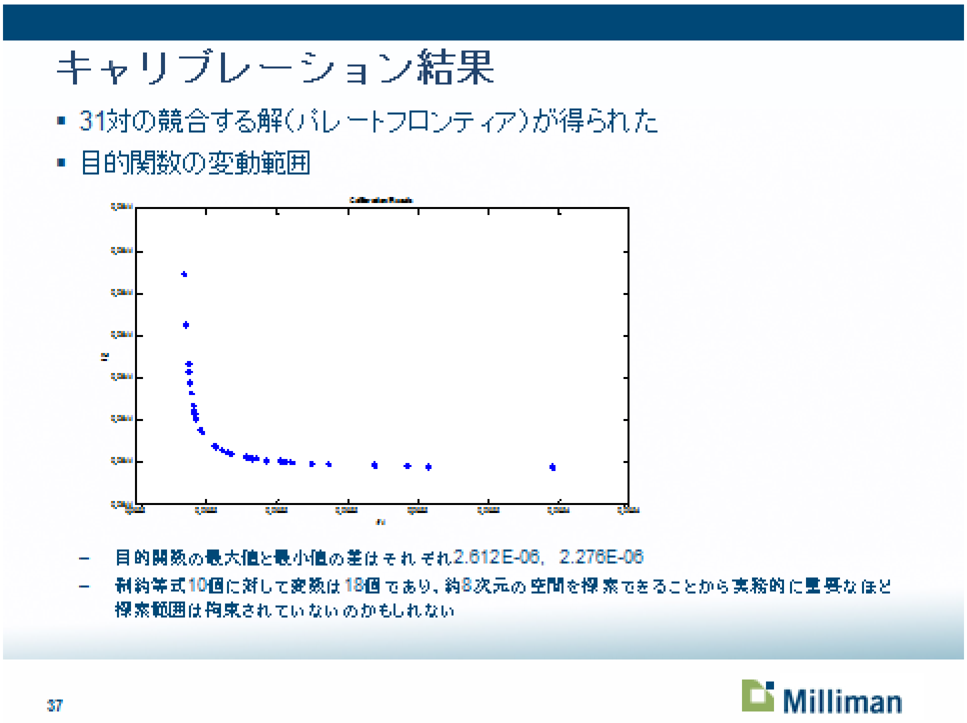
36 Milliman

相関係数はこのように設定しました。これは適当な数です。何でもいいのですけれども、適当な数を設定して、目的関数としては、先ほどのとおり、K と KF を同時に最適化します。ここで OF というのは、Objective Function という意味で使いたかったのですけれども、ここでは二つの目的関数を単目的化といって、二つの

加重平均を取ってこれを最小化するというような方法で解いています。

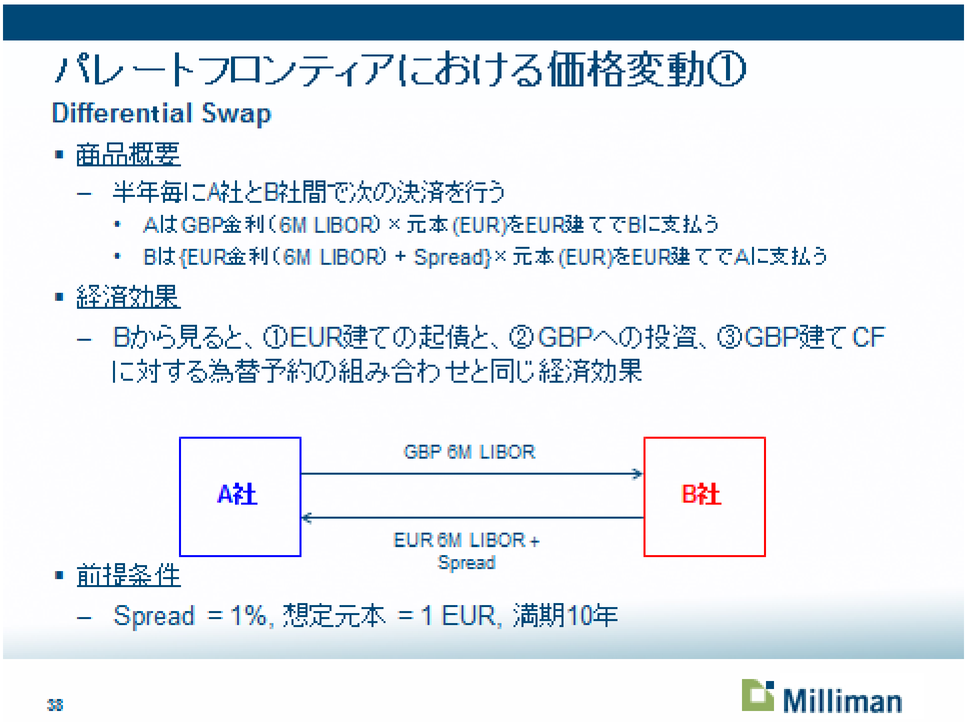
多目的最適化の解法の中で大きく二つの流儀があって、一つはこのように単目的化をして解くという方法です。もう一つの方法は、何か「遺伝的アルゴリズムでああだこうだ」というようなややこしいことになってしまうので、とりあえず単目的化で行ける所まで行ってみるということでやっています。

ちなみに、単目的化でパレートフロンティアが得られるというのは、凸関数という、下が1個しかなくて必ずこのような形になっているという関数の場合のみ証明がされているので、まだ一般的にこのケースできちんとパレートフロンティアが得られるという保証はありません。



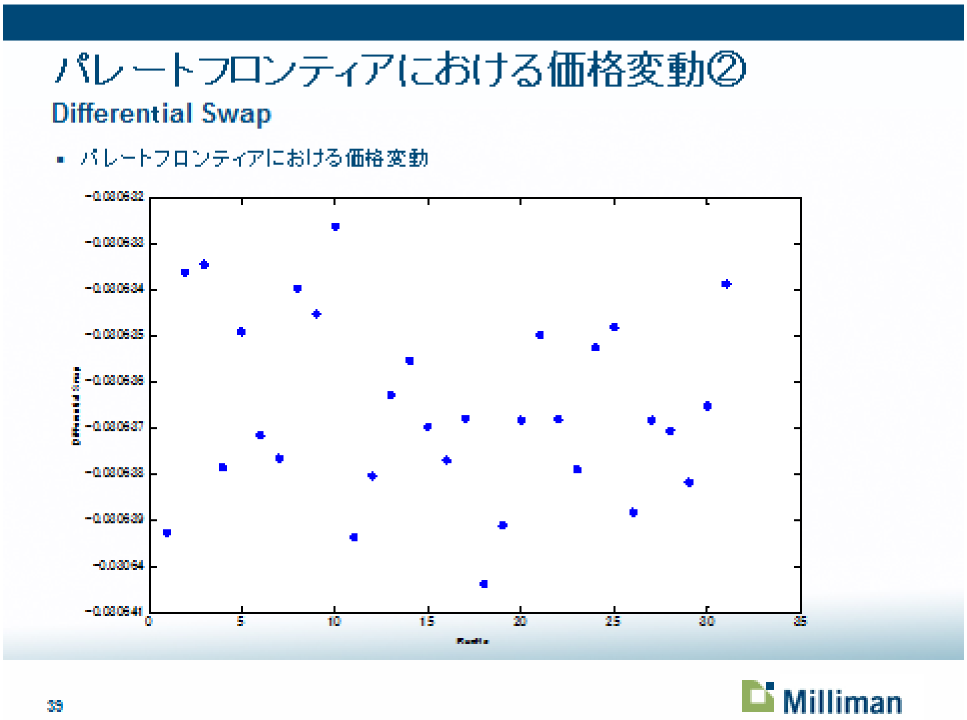
実際にやった結果ですけれども、横軸がユーロの目的関数で、縦軸がイギリスポンドの目的関数です。きれいにパレートフロンティアになっています。軸の目盛が大変見にくいかもしれないですけれども、大体、この縦の振れ幅と横の振れ幅の最大値と最小値の差をとってみると、非常に小さな数です。図にすると大きくは見えますけれども、実際の値としては、その目的関数としてはそれほど大きな変動になっていない。それほど変わらないのではないだろうかということが予想されます。

これはたまたまなのか、いつでもこうなのかということが問題で、それについて考えてみます。例えば、制約等式というのは10個ですけれども、動かせる変数というのは18個あるのです。そうすると、簡単に考えると、約8次元の空間の中を探索できるということになるので、実務的に影響を与えるほどには、探索範囲というのは制限されていないのではないだろうかということがうかがえます。「証拠を見せろ」と言われれば難しいですけれども、恐らく動かせる変数と制約等式の差、今で言えば8という数字ですけれども、その次元の大きさによって、このあたりの状況が変わってくるのではないだろうかと思われます。



これを今度は実際のデリバティブで使ってやってみれば、このパレートフロンティアの中で１個ずつ計算をしてみて、評価額を求めてみて、どのような差が出てくるでしょうかということを、二つのデリバティブで見てみます。

一つはディファレンシャル・スワップというもので、これは要するに、外国の金利と自分の国の金利を交換するというような取引です。これの評価をするときに、二つの経済を同時にモデルした金利モデルが必要になるので、これを取り上げています。取引の仕組みはこのようなことで、計算した結果です。



これで見ますと、元本１の場合の評価額を左から順に、先ほどのパレートフロンティアの左から順番の対

応になっています。見たところ、パレートフロンティアの中から動かしていくに従って、それほど何かのパターンというのは見えなくて、ばらついているように見えるということと、これの上限値と下限値の差が今10の－５乗ぐらいあります。これは元本１のものなので、例えば、100万円の元本で考えれば10円ぐらいしか変動しないということなので、それほど評価額に影響しないのかなということが見られます。

パレートフロンティアにおける価格変動③
一時払外貨建据置年金のTVFOG

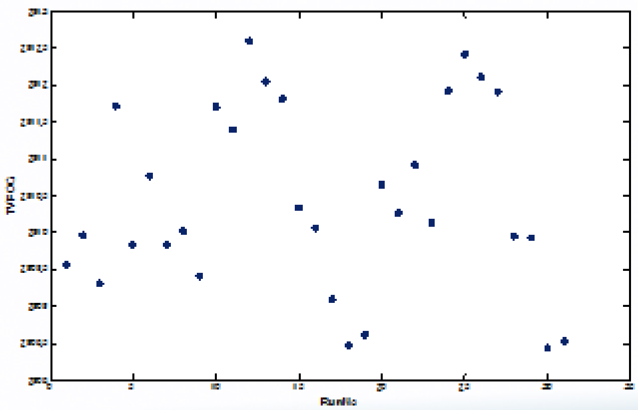
- 前提条件
  - 商品：EUR建て、据置期間10年、基本保険金1,000万円、MVAマージン0.45%、維持費ローディング1.65%
  - 57歳男性
  - 解約率
    - MVA後の邦貨建て解約返戻金>SPなら2.5%
    - それ以外は1.25%
  - 解約控除率 7%, 6%, 5%, 4%, 3%, 2%, 1% 0%,…
  - 件数比例事業費 30,000

40　Milliman

では、別の場合で例えば、外貨建てのディファードアニュイティではどうか。これの動的解約のタイムバリューを計算してみると、どうなるか。前提条件はこのとおりです。MVAがありますけれども、設定としては大変シンプルな設定です。

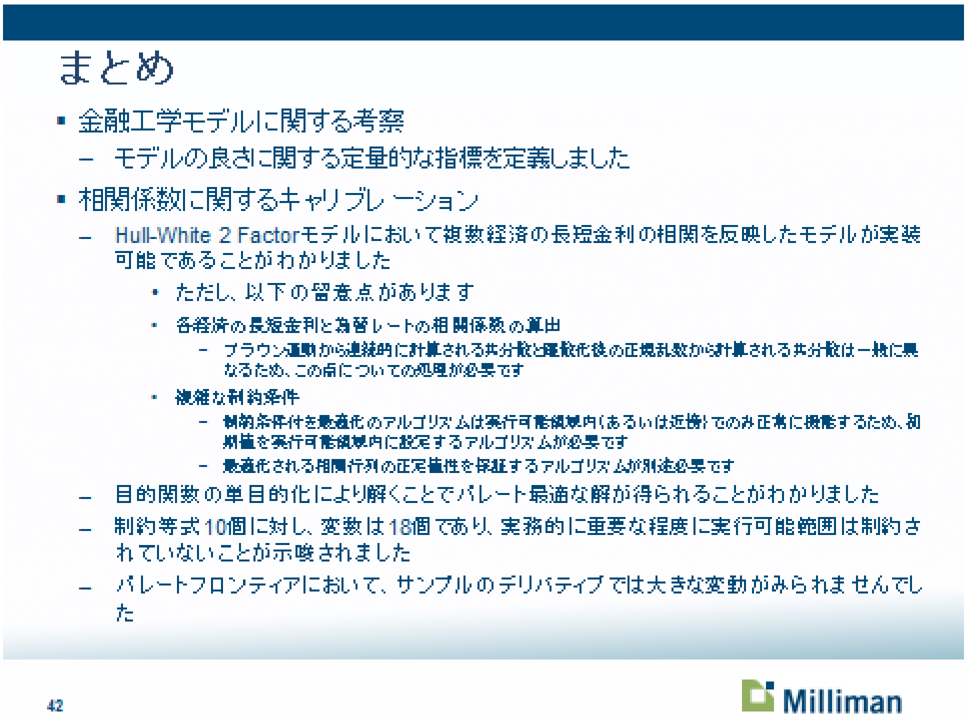
パレートフロンティアにおける価格変動④
一時払外貨建据置年金のTVFOG

- ばらつきはあるが保険金額10,000,000に対して大きな変動はないように見える



41　Milliman

これは保険金額 1,000 万円に対して、タイムバリューだけ計算したものがこちらです。これも大してパターンは見えなくて、何かばらばらとばらついているだけのように見えるということと、これも左側の目盛が見にくいですけれども、上限値と下限値を見ると４円から５円ぐらいしか差がないです。保険金額 1,000 万円でこの程度の差なら、差はないと言ってもいいと思います。



以上、はしりで来ましたけれども、まとめに入らせていただきます。まず初めに、モデルの良さに関する定量的な指標を定義しましたといいますか、提案してみましたということをやりました。相関係数に関するキャリブレーションについては、２Factor の Hull-White を使って二経済の金利モデルを実装するということができますということが分かりました。

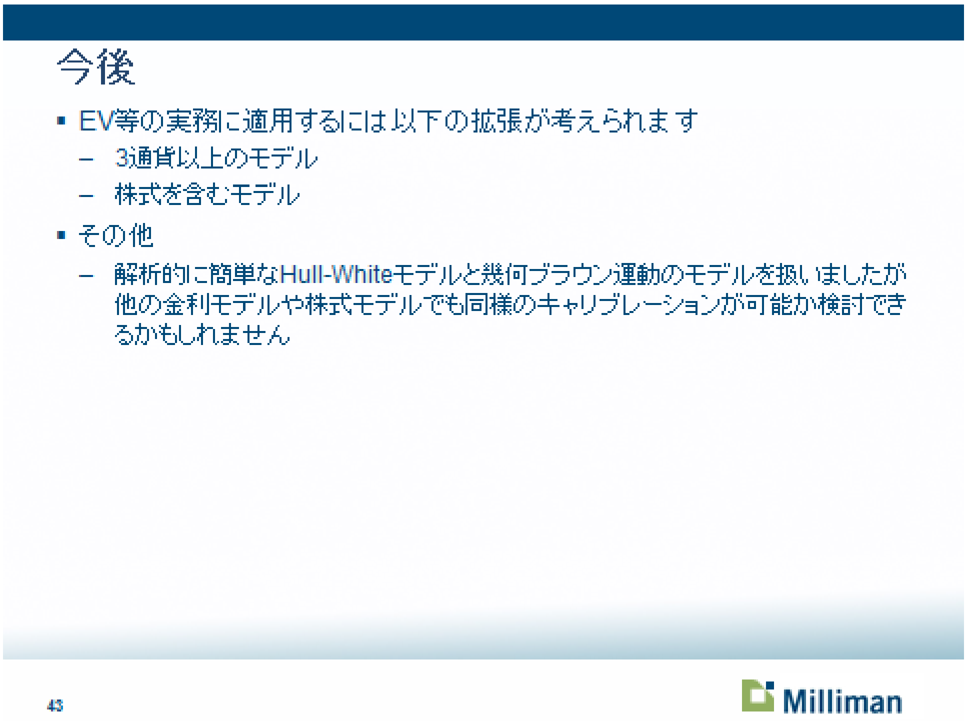
ただし、いろいろと技術的な課題というものはあって、今日は紹介しなかったのですけれども、ブラウン運動から連続的に計算される、要は先ほど数式で示した共分散と、実際に離散化をして例えば月次ベース・年次ベースでシュミレーションするときに出てくる共分散というのは、シナリオの誤差という意味ではなくて、理論的に微妙に違うので、そのあたりの処理をどうするかということが一つ悩ましいところです。その点については今日はやっていません。

あとは、初期値を設定してキャリブレーションをするのですけれども、数値最適化の制約条件付きの最適化になります。制約条件付きの最適化をするときは、初めにその初期値を設定するところが実行可能解といいますか、要は制約条件の範囲の中になければいけないので、そこにまず行き着くためのアルゴリズムというものがまた別に必要になって、なおかつそこから外に何かの拍子にはみ出ないように、うまく処理をするということが必要です。それも制約条件が 10 個あるので、結構そのあたりの技術が大変なところです。

また、先ほども少し言いましたけれども、固有値が全部正でなければいけないということが、これが結構やっかいな制約条件で、これについてもまた別にアルゴリズムが必要になります。

あとは今申し上げたとおりですけれども、目的関数を多目的化して解くというだけでも、パレートフロンティアが出てくるということが分かりましたということ。それほど実行可能範囲というものは制約されていないのではないでしょうかということ。サンプルのデリバティブではありますけれども、そのパレートフロ

ンティアから結局、最終的にどこを選ぶかというのはそれほど結論としての評価額には影響しないのではないでしょうかということが分かりました。



今後としては、二通貨でうまくいったので、今度は例えば変額のモデリング、変額のファンドのモデリングになると二通貨だけでは足りないので、真面目にやろうとすれば三通貨、四通貨以上のモデルが必要ですから、そのようなことができるのかどうか。そうなると、今度は株式を含むモデルが必要ですので、そのようなことができるかということ。

その他としては、解析的に簡単な Hull-White を使って、幾何ブラウン運動で扱いましたけれども、もっと別の金利モデル、例えばヘストンのモデルや株式のモデルについても、そのようなモデルにしたときに同じことがきちんとできるのかどうかということは、大変難しいとは思いますけれども、検討の余地はあるかということです。

長らくお話しさせていただきましたけれども、途中はしりになった所や細かく読んでよく分からないというときは、こちらに直通とメールアドレスが書いてありますので、私宛てにご質問等をいただければと思います。今日は長い間、ありがとうございました。

Ａ　どうも、大塚さん、ありがとうございました。若干、お時間がございますので、フロアよりぜひ、ご質問を頂戴したいと思います。

加藤　三井生命の加藤です。いつもお世話になっています。
23 ページに……。今回の最後の論文の意味や効果というのはどのようなところにあるのかということがよく分からなかったので、教えていただきたいのですけれども、「しかし、形状変更は観測値に合うようにキャリブレーションすることは通常されていない」とあり、それを今回やりましたというお話ですね。

大塚　そうです。はい。

加藤　通常はどのようにしていて、それは今回やったこととどう違うのか。通常と今回やったこととの実務上の効果というのは、どのような所に表れてくるのか。そのようなことを分かる範囲で解説していただけると助かります。

大塚　はい。分かりました。今回、真面目に二つの経済の金利で 2 Factor のモデルでやってみるということをしたのですけれども、実際に普段このようなことをやるかというと、このようなことはやらないです。というのは、大変難しいのです、やはり。今回はきれいな結果だけをお見せしていますけれども、これを得るにはなかなか苦労していて、これを決算時にやれるかというと、まだまだ技術などが足りません。本当はやった方がいいけれども、まだ難しいところがあるという理由で、やれていないというのが現状です。

実際には、では、どのようにしているかというと、これは会社さんによりけりだというところもあると思いますけれども、２Factor のモデルで言えば、第一のファクターがちょうど１Factor の Hull-White のファクターと一緒ですので、例えば短期の金利、１年金利でも何でもいいですが、それと他の指数、為替や外国の短期の金利などを並べてきて、そこから相関係数をとって、その相関係数になるようにファクター同士の相関係数を決めるということが、第一のファクターの相関係数を決めるということが考えられます。

問題は第二のファクターです。そこは僕の知る限り、ごまかすのです。一つのごまかし方としては、第二のファクターというのはここで見ると、第二のファクターのｄｗ２というのはｕを動かすファクターです。ｕというのは上のｄｒの式で、長期的な水準と考えられます。要するに、ｒがこの回帰していくべき長期的な水準の変動と解釈できるので、これは長期の金利そのものだと思い込んで、第二のファクターの動きというのは、例えば市場観測できる長期の金利です。10 年、20 年、30 年なりの動きと一緒になるはずだという説明をして、それでローのファクターのｄｗ１とｄｗ２の相関係数として、例えば１年金利と 10 年金利の相関係数を使う。使うというからには、それはキャリブレーションには含めないことになるのですけれども、そのように扱うという方法も考えられます。そこはよく分からないので、何らの仮定もできないということで、そのｄｗ１とｄｗ２の相関をそもそもなくしてしまう。もう独立の、何か金利に影響を与える、よく分からないものという意味で、残差の扱いと一緒で、何からも独立した拡散させるものだと考えて、何の仮定もしないという方法も考えられます。

それがどのように影響するかというのはなかなか分からなくて、例えば、通常は何か変動要因の分析をするときに、ボラティリティが増えたからそのタイムバリューが増えたのだなどといういろいろな説明がありますけれども、中でも相関係数がどのように変わったからＥＶはこのように動いたという分析というのは、恐らく今後もできないのではないかと思います。相関係数の変化と評価額の変化を結び付けることがなかなか難しい。明らかな場合もあります。例えば、株同士の相関が上がったからその変額のファンドのボランティリティが余計に上がったのだというようなことは言えます。結局、それはボラティリティという言葉に直してから分析をするもので、今やっている実務がどれぐらい正しいのかというようなことは、ちょっと厳密な意味では分からないということです。

ところが、この問題をずっと放ってはおけないので、何らかの解決はしなければいけないということで、今回、二経済というように簡単化していますけれども、取り組んでみたという経緯でございます。

飯沼　ありがとうございます。お世話になっております。SBI の飯沼です。大変貴重で、かつ共有させていただいている問題意識でございまして、なるほどなと感心いたしました。ありがとうございます。

ただ、一つ感じますのが、まさに今、見ていただいた（9）式ですけれども、最初の方のご質問と多分、本質的に重複しているのだと思ったのですが。基本的に第一式は通常の平均回帰過程であって、第二式のｕが長期金利であるというように、もう過程を置いているところが、正直言うと、本当なのかなと。

かつ、このキャリブレーションもフォワードレートの確率微分方程式の相関をとって、２Factor Hull-white ではそこを解析的に出せますので、そうすると、スワップションの式も解析的に作れるので、市場観測値の相関と結び付けて最適化に使われている。それはそうだなと思うのですが、観測値自体も、先ほど１年と 10 年ということで決定されていらっしゃったと思うのです。そこは本当なのかなと思っていて、例えば、本来の２Factor Hull-white の自由度というのは、１年と 10 年と置いてしまった瞬間に拘束されてしまっていて、本当であれば、例えば円は１年と 10 年でいいのかもしれないけれども、ドルであれば１年と７年が正しいのかもしれなくて、それで記述した方が世の中のスワップションを表現できるかもしれない。そ

こをもうすでに恣意的に決めてしまっていらっしゃるので、そうすると、結局、制約条件、まず金利モデル自体がそのようなことを置いてしまっていいのかというところ。

あとは、先ほどの最適化の考え方も、実はちょっと変わっていくのではないかと。さらに、スワップションで多分キャリブレーションされていらっしゃったと思いますが、もうこれもご存知のとおり、１年、10年というところでもう置いているからこそ、スワップションは１年、10年ありますのであれですが、本来もっと長期的なものを見るならば、長期金利とうたっているのであれば、長期的なものを見なければいけない。とするならば、本来は、例えば２Factor Hull-white でもご存じのとおり、アフィンモデルを使えば、債券価格で表現できるはずですので、債券価格自体の相関をキャリブレーションに使う。そうすると、変なスワップションの市場のゆがみなどがなくて、かつ、もしかしたら、これは夢かもしれないですけれども、１年、10年というそこの数字自体も一つのパラメータに置き直して、さらに最適化を行うということも可能なのではないかと思ったのですけれども、いかがでございましょう。

大塚　ありがとうございます。それはご指摘のとおりだと思います。１年、10年と決めるのは、それは一つ仮定するしかないというところです。選べる候補としても、２Factor のモデルでは二つしかないので、結局どれかを選ばなければいけない。どれかをターゲットにすると、では、別のところは犠牲にされてしまうというようなこともあるので、それがどこまで最適化の変数に含めて、よく分からないものとして、結局、最適なものに行き着くようなアルゴリズムの中に入れられるものかという線引きは、確かにおっしゃるとおり、今後検討する余地があると思います。ありがとうございます。

司会　はい。それでは、もうお時間がまいりましたので、以上とさせていただきます。それではもう一度、大塚さんに拍手をお願いします。どうもありがとうございました。どうもお疲れさまです。

大会委員　大会委員から事務連絡をさせていただきます。ここで休憩とさせていただきます。次のパネルディスカッションは16時10分から開始いたします。開始２分前にはご参集いただきますよう、お願いいたします。なお、今年度より、正会員を対象に出席確認を行っております。出席確認がまだお済みでない正会員の方は、出席確認受付にお立ち寄りください。