MIRAI Archive





International Research Unit of Advanced Future Studies(IRU-AFS)

No.2015-011

<講演録>

京都大学基礎物理学研究所 研究会 京都大学未来創成学国際研究ユニット設置記念シンポジウム

日時 2015年8月7日(金)13:40-14:20

場所 コープイン京都 201

司会 三輪敬之(早稲田大学創造理工学研究科 教授)

講演 小林泰三(帝京大学福岡医療技術学部 准教授)

動的過程の計算論

三輪●続きまして、小林泰三さん。お願いいたします。

小林●ご紹介にあずかりました、小林と申します。私の題は「動的過程の計算論」です。 かたい題をつけてしまって失敗したなあと思うのですが。

私の所属は、今年からできた帝京大学の医療技術学部医療技術学科というところです。 ここではおもに授業をしておりまして、研究はあいかわらず九州大学の情報基盤研究開 発センターで進めております。

私以外の講演者の方はみなさん名前の知られた大御所ばかりでして、私がいちばん若 いと思います。ちょっと自己紹介させてください。

◇スライド2「自己紹介」

私は立命館大学の池田研介先生の研究室の出身です。池田研究室にいるときには、金属微粒子のモレキュラー・ダイナミクス(Molecular Dynamics)をしていました。そこから九州大学に移りまして、九州工業大学の高橋公也さんといっしょに、なにか楽器を研究しましょうというと、フルートだったら動く部品がないので、ぜんぶ流体音の範囲で話ができるのではないか

自己紹介

• 出身:池田研介研究室

流体音: フルートの音はどうして出るか?

• 計算機関連

VineLinux PPC kernel 担当、"MacOS X WorkShop" 主宰、NAREGI, HPCI-AE

その他

「小さな大学」「NPO あいんしゅたいん」 「JST-RISTEX 法と科学」

ということで、フルートの音がどうして出るのかという研究をしていました。

情報基盤研究開発センターにいるので、いろいろ計算機のこともしてきました。ご存じの 方もいるかもしれませんが、ヴァイン・リヌックス(Vine Linux)という、大学でよくつ



かわれているリヌックスがあります。むかしは、これの PPC カーネル (Power PC Kernel) を担当していました。あるいは、Mac で eMacs か TeX (テフ) をつかって研究する人のためのディストリビューション (distribution) を 10 年以上、主としてやっています。

スライドには、計算関連で NAREGI や HPCI-AE と書いてあります。NAREGI (National Research Grid Initiative) は 10 年くらい前に行なわれた国家プロジェクトで、グリッド・コンピューティング (Grid Computing) というものがありました。日本の各大学にある大きなスーパーコンピュータを連携して使えたら、もっと研究効果があがるだろうと。では、それをつなぎましょうということでした。

グリッドということばは、パワー・グリッド(電力送電網)に由来します。ふつうの電源のことです。どこでもプラグがあるので、電化製品はそこに差し込めばつかえる。同じように、計算機もどこでもつかえるようにしましょうという触れ込みでした。しかし、いまそうなっていないことからわかるように、うまくいっていないのです。この講演は収録されているので、このへんの裏話も用意してきたのですが、割愛いたします。

これを引き継いだ HPCI というものがあります。ハイ・パフォーマンス・コンピューティング・インフラストラクチャ(High Performance Computing Infrastructure)という名前です。これは神戸の理化学研究所にある京(けい)コンピュータを中心に、旧七帝大、筑波大学、東京工業大学をつないで有機的に使いましょうというものです。いまはやりのクラウドを使おうという話が出て、その担当をやれといわれて、ちょっと関わりました。今日お話しするのは、スライド1の青字のところにかかわることです。

そのほか、学部教育関連のことがらにも携わりました。あとで池田さんが話されるようなお話です。また、坂東昌子さんや佐藤文隆さんが運営している NPO 法人、知的人材ネットワーク・あいんしゅたいんにも所属しています。あとは、ちょっと前のことですが、科学的なことについて法廷の現場に証人としてよばれたりすると、とんでもないことを証言させられそうになるという話がありました。弁護士、科学技術社会論の研究者、東北大学の本堂毅さんといっしょに、法と科学に関係する研究プロジェクトにも在籍していました(JST-RISTEX プロジェクト 法と科学)。

◇スライド3「『動的な』話題の対象」

2日間、大御所たちのお話で内容の濃い議論をしてきてお疲れだろうと思いますので、素朴な話題になるようにスライドをつくりなおしてみました。

最初は計算科学の話です。流体音、フルート からなぜ音が出るのかということです。楽器な

「動的な」話題の対象 流体音:フルートの発音 計算科学 移動境界問題:フィンガリング 連成計算:ソルバ・中間処理管理 計算機科学 資源管理:プロセス・ストレージ・ ネットワーク・ユーザ・… 資源管理:資源提供拠点状況の 情報インフラ 把握と管理 連続運転性:インフラは止められない 「クラウド」 「ビッグデータ」 不確定性への対応 Cyber Physical System (CPS)



ので音を変えますね。吹きながらフィンガリングをする。これは、まだきちんと計算で きていない問題なのです。こんなこともまだできていないのが、われわれです。

計算は、計算機がきちんとつくられているからできるのであって、計算機を対象とした科学があります。フィンガリングなどをしようとすると、フルートのパッドを閉じているときと開いているときとでは、トポロジー(topology)が変わるのです。流体の計算をするときは、空間をこまかく切って計算する(メッシュ図)のですが、閉じたときは管と指のパッドがあるところがつながっているわけです。開けた瞬間に、前になかったものがメッシュに入りこむ。そういうところは微妙な問題がたくさんあって、そう簡単にできるものではない。

次に、計算機科学です。われわれがユーザとしてきちんと計算機を使えるようにするために、資源管理をします。プロセスは、動いているプログラムのことです。ストレージは、ハードディスクや USB メモリのようにデータを溜めるところです。あとはネットワーク。ユーザも管理対象になっています。

ちょっと前まではこのへんで話を閉じていましたが、昨今はビッグデータ (Big Data) やクラウドが情報インフラとして出てきて、計算機科学だけでは話が収まらないところ にきています。

資源管理をするのですが、どちらかというとメタな感じです。たとえば、HPCIでした ら、各大学が計算機をたくさんもっています。それらがどういう状態になっているのか ということの把握と管理と、スライドに書いています。

あとは、ビッグデータなどのサイバー・フィジカル・システム (Cyber Physical System、CPS) というものがあります。どういうものかというと、遠隔地にいろいろ置いてあるセンサからどんどんデータが集まってくる。これらを利用して、リアルタイムをまたほかのところで使いましょうと。

いま研究プロジェクトとしてあがっているのは、やはり医療系が多いです。患者さんに、いろいろと状態をモニタリングする機器をつけています。ふだんはふつうに生活していただく。なにか変動が起こったら、移動クラウドにデータがあがって、主治医のところに情報がゆく。そういうシステムをつくろうというプロジェクトがいくつかあります。交通量に関するものも多いです。このへんが盛んになって、予算がとりやすいこともあって、雨後の竹の子のようにボコボコと出てきています。

「これ、大丈夫かなあ」とぼくがいつも思うのは、こういうインフラは止められないということです。インフラとして深く日常生活に入りこんだら、止められないのです。 しかし、いまのコンピュータのシステムは、止めずにずっと連続運転してゆけますかと。

いちばん困るのは、OS のアップデートなどをするときです。ソフトウェアは、どんどんバグを修正して、機能を強化して、どんどん更新してゆく。そのときはプログラムを



止めますね。OS を入れ替えるときは、パソコンを落とします。そういうことができない 状況のことを考えないといけないのに、不思議と議論されていない。大丈夫かなと思っ ています。

具体的で素朴な話をしていますが、けっきょくどうなのかというと、動的な話――ユーザがどのように計算機を使うのかというのを予測するのは不可能ですね。不確定なものに、どのように対応してゆくのか。そういうことをお話しし、議論するということです。では、具体的な話に入ります。

◇スライド4「計算科学」

フルートの音がどうして出るのかという話です。ここからしばらくは、気軽に楽しんでいただけるかと思います。フルートそのものは唇が入るためにむずかしいので、リコーダーをモデルにして計算します。リコーダーは、たんにフッと吹けば音が出るので、そうむずかしいことは起こっていないとは思うのですが。

計算科学

◇スライド 5「エアリード楽器の発音」

最初は、フッと吹きます。吹くと、歌口にジェット(jet)が出ます。ジェットが出て、反対側のエッジのところにぶつかる。すると、それだけでパタパタと振動するようになります。そこから出てくる音をエッジトーン(Edge Tone)とよびます。

ここで、最初に音らしきものが出るわけです。 いったん音が出たら、楽器には管体がついてい

エアリード楽器の発音 1. Jet の生成 2. エッジトーン 3. 管体共鳴 (周波数選択) 4. 音→流体(Jet)へのフィードバック 5. 楽音 動的な状態の積み重ね (動的過程の一例)

るので、そこで共鳴して、周波数選択が行なわれる。肝になるのはここです。音が戻ってきて、なにかしらジェットに影響を与えないと安定した楽器にはならないのです。なにかしらのフィードバックがある。

フィードバックがあったら、また 1、2、3 と戻るのです。図のようにグルグル回っている。これのおこぼれのようなもので楽音が出る。だいたい五つの動的状態の組みあわせ・重なりあいでリコーダーから音が出る。

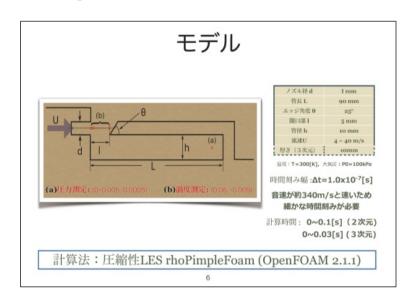
ひとつ言っておきたいのは、ふだんわれわれが聞いている楽音は、音の強さとしてはだいたい 60dB くらいの音です。フルのオーケストラで 120dB くらいです。共鳴管体内の音圧線は 140dB くらいなので、ここに耳をおいたらすぐに鼓膜が破れるくらいの音で



す。

こういう状況でも、ジェットがもっている流体としてのエネルギーと音場のエネルギーをくらべてみると、5 ケタくらいの差があるのです。流体の計算をしている人だったらわかると思うのですが、5 ケタも小さいものは、ゴミみたいなものでふつうはまともに相手にしないのですが、そういうものが効いてくる。こういう単純な話でも、動的状態の積み重ねであり動的過程として捉えないと、なんで音が出ているのかということは「たしかにそうだ」と納得できるところまではいかない。

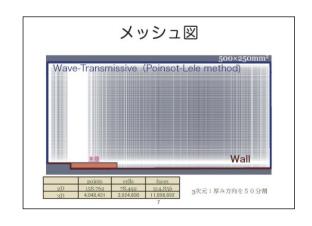
◇ スライド 6「モデル」



ここから先は、このように計算できたというものを楽しんでいただこうと思います。 このようにモデルを切って、もともとこのあたりの構造が微細なことと、音波をあつか わなくてはいけないので、メッシュの幅は 0.1mm です。すると、音が 340m/s でゆくの で、クーラン条件から、時間刻みも 10 の 7 乗、8 乗で進めないといけない。たとえば、 ピーという音を 1 秒間出そうとすると、すごいステップスになる。

◇スライド 7「メッシュ図」

このようにメッシュを切ります。まわりが 透過壁で、下が壁で、左下に楽器を置くとい うようなものでします。なぜこんなにまわり の空間が広いのかというと、かなり音がでる のですが、10の8乗セットくらいの長い計算 をすると、こういうところでどうしても反射 するのです。積もり積もって、いろいろとお

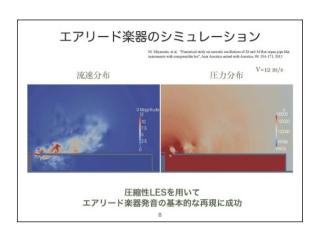




かしなことを起こすので、ここをある程度広くとって計算せざるをえない。やってみる と、こんな感じです。三次元の計算をまん中でスパッと切ったかたちです。

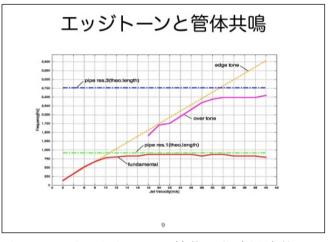
◇スライド8「エアリード楽器のシミュレーション」

左側が流速です。このようにジェットが出ている。右のほうが圧力分布で、赤くなったり青くなったりしているところから、きちんと発振していることがわかると思います。いちおう、ここまでは基本的な条件をそろえれば出てくるのです。ここから研究開始ということです。



◇スライド9「エッジトーンと管体共鳴」

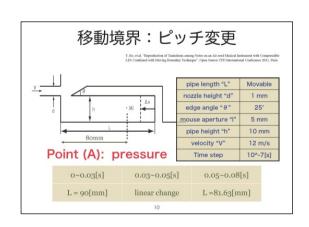
エッジトーンと管体から戻ってきた 音との相互作用で楽音が決まるという 話をしましたが、それを出したのがこ の図です。横軸はジェットの速度で、 縦軸が周波数です。管体をつけずにエ ッジトーンだけにすると、ずっと線形 に上がってゆきます。ですが、楽器で 管体がついていると、管体の共鳴周波 数がどんど来ます。



楽器でジェットを入れてゆくと、きちんとこれに沿ったところに管体の共鳴周波数にロックしてゆく。ロックするということは、たんにエッジトーンのまま振動しているのではなく、なにかしらエッジトーンに制御がフィードバックされているということです。

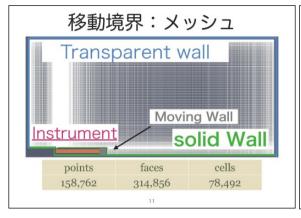
◇スライド 10「移動境界:ピッチ変更」

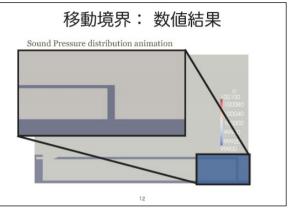
楽器なので、ピッチを変えたいと。端のところをちょっと短くして、ピッチが上がるかどうかが計算できるかを試してみます。同じようにメッシュを組んで、この部分をギュッと動かすということをしています。





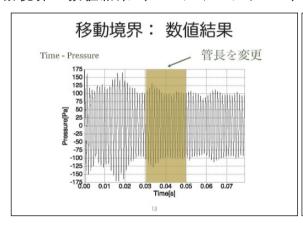
◇ スライド 11「移動境界:メッシュ」、スライド 12「移動境界:数値結果(図)」

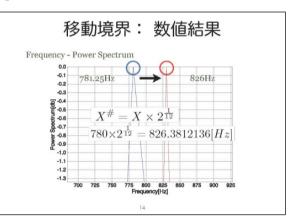




計算しながら、途中でだんだん短くなっている。この壁がぶあつくなっているのをご覧いただいていますが、このようにして計算してみました。これは時間発展で、縦軸部分も圧力をとっています。このように音が出ている。管長を変えた部分がこの部分です。

◇スライド 13「移動境界:数値結果(スペクトルグラフ 1)」、スライド 14「移動境界:数値結果(スペクトルグラフ 2)」

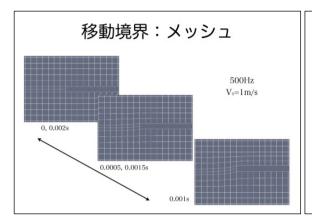


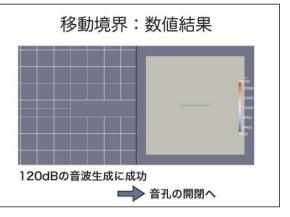


管の長さを変える前と後で周波数スペクトルをとってみると、きちんと半音ぶん変わっていた。これは、理論予測どおりに条件を変えてみて、ほんとうにそうなるかどうかの確認なので、そのとおりになっています。最後に、ちゃんと音孔の開け閉めまでしなければいけない。どのようにメッシュを動かすのかをいろいろとやっています。



◇スライド 15「移動境界:メッシュ」、スライド 16「移動境界:数値結果(メッシュの一部)」

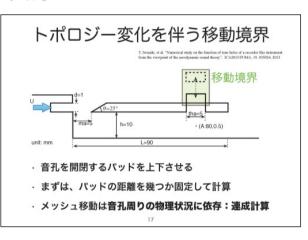




音場の強いところでもちゃんと計算できるスキームかどうかということのチェックが、いま終わったところです。120dBの音波を出すというメッシュの切り方、移動のしかたが目についた。

◇スライド 17「トポロジー変化を伴う移動境界」

図のようなかたちになっていて、薄緑色の部分を上げ下げします。まずは、これを連続的に上げ下げするのではなくて、ある決まった位置に置いて、それぞれ計算をしてみて、きちんと発振するかどうか。メッシュ移動は計算しないときはよいのですが、計算するとこのあたりにいろいろと流体の運動が出ます。渦も出ます。そういうものをきちんと考慮してメッシュを移動しないと、不自然なことが簡単に起こります。

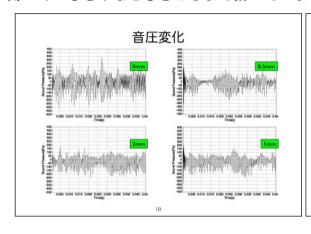


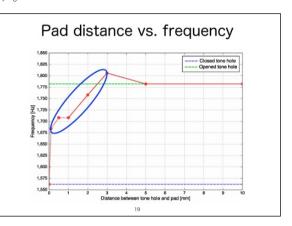
とくに、閉じたり開けたりする瞬間は、どんなにがんばっても衝撃波が待っています。 そういうところをきちんと処理しようとすると、一つのこういうソルバ(solver)だけで はなくて、そういうものを処理する別のプログラムを同時に走らせて連携させることを しなければいけません。連成計算というのですが、まだ標準的な方法というか、「こう すればいい」というようなものがない対象です。



◇スライド 18「音圧変化」、スライド 19「Pad distance vs. frequency」

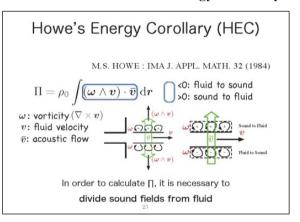
音圧変化を出してみると、横軸が楽器とパッドのあいだの距離です。閉じている場合、1mm、2mm、3mm、4mm、5mmとなっている。だいたい3mmくらい開ければ、音孔は開いているとみなせるところまで動いています。





◇スライド 20「フルートの音源はどこ?」、スライド 21「Howe's Energy Corollary」

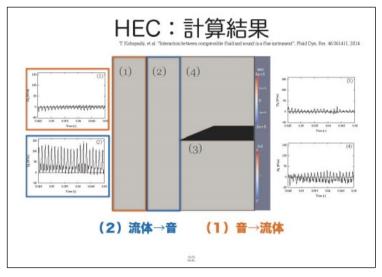
フルートの音源はどこ?



ご覧いただくものの最後として、フルートの音源はどこかというところまできました。これはハウズ・エナジー・コロラリー(Howe's Energy Corollary、HEC)というものです。どういうものかというと、式はここだけでよいです。渦度と流速との外積をとって、音響的流出速度(音波)の内積をとると、これが負のときは流体から音にエネルギーが移っています。逆に正の場合は、音から流体にエネルギーが移っています。そういうことを、マイケル・ハウ(Michael Howe)という人が1984年に唱えました。これはまだ確かめていないので、ほんとうかどうか検証も兼ねてじっさいにやってみました。



◇スライド 22「HEC: 計算結果」



これはリコーダーの歌口の部分を大きくとったものです。領域を 1、2、3、4 の部分に分けて、おのおの HEC の積分をしたものです。

グラフをみるとおわかりいただけるのですが、流体から音のほうにエネルギーが移っているのは2の部分でした。逆に、1の部分で音から流体にエネルギーが流れてゆく。音のエネルギーは2のところから注入されていて、管体で共鳴した周波数がジェットにフィードバックするところが1です。それ以外の場所があるかといわれたらないので、それはそうだという結果なのです。

◇スライド 23「まとめ:動的状態の階層構造」

まとめです。リコーダーから音を出すというだけの簡単なことでも、最初にジェットをつくって、それからエッジトーンが出て、その音が管体共鳴して、周波数選択されて、それがまたエッジトーンを制御する。そのおこぼれで楽音が出る。これだけの動的状態の重ねあわせで音が出ているということがおわかりいただければよろしいかと思います。

まとめ:動的状態の階層構造

(1)ジェット

(2)エッジに当たりエッジトーンを発生

(3)管体共鳴による周波数選択

(4)エッジトーンとの相互作用

(5)楽音の形成 流体と音の連成計算で探る

Lighthill 音源:不確定要素

各不確定要素の制御=演奏者の技量

これだけいろいろな動的なものがあると、不確定要素が各所にいろいろと入りこんでくるのです。これをどう制御するのかというのが、けっきょくは演奏者の技量である。 素人が吹いた音とプロが吹いた音で、なぜあんなに音が違うのか。こういうことを知ろうと思ったら、動的なものがどういうものかをきちんと押さえないと議論ができないでしょう。



いま研究しているのがエッジトーンとの相互作用です。音は流体にとってのゴミみたいなものですが、音がどのように流体に影響をおよぼしているのか。これはまだわかっていません。これをいまずっとしています。

◇スライド 24「不確定性をどう扱うか」

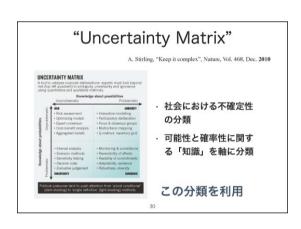
いちばん最初にお話したとおり、計算科学があって、 計算機科学があって、情報インフラがある。三つくら い階層があるのですが、これらにずっと携わっていま す。なにが問題になるかというと、不定性がどうして も出てくるのです。動的なものは不定性のかたまりで すが、これをどうあつかうのかという問題がつねにあ

不確定性をどう扱う?

ります。これに対して仕事をするのですが、ほとんど対症療法ばかりのような気がする のです。そもそも論をしてみようと思います。

◇スライド 30「Uncertainty Matrix」

不定性というものをどのように捉えたらよいのでしょうか。科学技術社会論という分野で、アンドリュー・スターリング(Andrew Stirling)という人がいます。もともとは物理学者です。この人が、不確定性をどう分類するのかという考えを2010年に出してくれています。これであれこれといじっていたのですが、この枠組みで使ってみてみましょう。



◇スライド 31「不確定性の分類」

先ほどのスライドは英語で書いてあったので、 わかりやすく捕捉します。二つの軸があります が、横軸が結果の可能性についての知識がある かないか、行(縦軸)が発生確率についての知 識で、マトリクスをつくっています。どちらも ない場合を無知としております。

次の段です。多義性というのは、なかなかよ い例がないのですが、たとえば、みなさんも計



算機をお使いだと思います。ある設定ファイルがどこか一箇所壊れているとしましょう。



どこか設定ファイルが壊れたということはわかるが、この結果どのように計算機が変な 動きをするのかは、どうなるのかわからない。こういうものを多義性というわけです。

不確定性というのは、これが起こればこういう結果になるということはわかるが、こ れがどのように起こるのかがわからない。動的過程は、こういう不確定性の遷移である ということもできるだろうと思っています。

◇スライド 32「不確定性と研究の営み」

じっさいに自分自身をふり返ってみると、研究自身も最初はなんだかわからないから

始めるのですね。最初は方向性もない。右下か ら始めます。たとえば、流体の計算をするのだ ったら、シミュレーションの設計を最初にしま す。流体だからナビエ-ストークス方程式を解け ばよいでしょうというところまではよいのです が、それを解いてなにが出てくるのかというこ とが、この段階ではわからないのです。



計算を始めると、いろいろと現象が出てきま

す。出てきても、どういうものなのかという分類はできないうちは左下です。ローカル なモデルをつくったりして理解が進んでゆくと、ぜんぶわかりますよという左上にゆく。 こんな一筋でゆくわけではなく、結果が出始めると、思ってもいなかったようなもの がゾロゾロと出てくるのです。「なんじゃこりゃ」というように出てきます。すると、 これはなんなのだろうということをみるために、振り出しに戻ったりして、グルグルと 回る。だいたい研究を進めてゆくと、わからないことが次つぎと出てくるので、ずっと こういう環のなかにいるという見方もできそうです。

◇スライド 33「トランスサイエンス 1」

ここまでは、個々人の研究者の話で済むのでよいのですが、情報インフラなどに話がゆ くと、ちょっと困るのです。プロジェクトが大規 模になってくると――これも科学技術社会論の分 野の話ですが、トランスサイエンス (Trans-Science) ということばがあります。1972年にアルヴィン・ ワインバーグ (Alvin Weinberg) が述べた話です。 そのなかから抜粋すると、「巨大なもの、たとえ

ばアスワンハイダムや100MWのプルトニウム増

殖炉、大きな橋などでは、フルスケールの試作品

「トランスサイエンス」 A.M. Weinberg, "Science and Trans-Science", Minerva, vol.10, 209-222 (1972) ...巨大なもの、たとえばアスワンハイダムや100 MW のプルトニウム増殖炉、大きな橋などでは、フルス ケールの試作品を作るようなことは問題外である。 さらに、そのような巨大装置が稼動するのは100年 のスケールであり、例え試作品が作られたとしても、 試作品に弱点が無いかが確かめられるまで実際の建 設を待つということはナンセンスである。 つまり…



を作るようなことは問題外である。さらに、そのような巨大装置が稼動するのは100年のスケールであり、たとえ試作品がつくられたとしても、試作品に弱点が無いかが確かめられるまで実際の建設を待つことはナンセンスである」と。

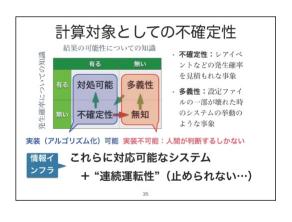
◇スライド34「トランスサイエンス2」

そのとおりなのですが、じっさいに大規模なプロジェクトで起こることは、基本的に一発勝負なのです。失敗するかどうかは、やってみるまでわからない。さまざまな不確定要素が相互作用して、状況を切り分けることが困難なことがずっと起こります。「こういうことをするのだったらこういう専門家を集めればよいだろう」と最初に想定して始めても、これだけではいけない問題がなかから出てくる。

「トランスサイエンス」 A.M. Weinberg, "Science and Trans-Science", Minerva, vol. 10, 209-222 (1972) 大規模プロジェクトで起こることは… * 基本的に一発勝負 * 様々な不確定要素が相互作用する * その分野だけでは答えられない問題がある NAREGIで実感

◇スライド35「計算対象としての不確定性」

これは私自身がじっさいに計算機グリッドのプロジェクトに関わって実感したことです。計算のほうに戻ると、どういうことがいえるか。マトリクスの左側はアルゴリズム化できるので、プログラムにすることができる。しかし、右側はできないのです。しかし、プロジェクトが始まるときは右側から始まりますね。こういうところで、大規模プロジェクトはシステム設計をしなくてはならないのです。では、どうするか。



これらに対応可能なシステムを考えないといけないわけです。けれども、いまなされているのは――NAREGIのときもそうでしたが、「こういう機能が要りますね、ああいう機能も要りますね、こういうのも要りますね」と機能をダーッと並べてゆく。そして「それをつくりましょう」と。これで機能は列挙されるでしょうが、機能をどう機能させるのか。集めた機能をどのようにしたらきちんと組みあわさって、サービスとして機能するのか。どう機能させるのかという話は、まったく出ないのです。それだけでもたいへんなのに、日常に入りこんできている情報インフラの根幹は連続運転性を求められている。こういうことで大問題が起こっているわけです。



◇スライド36「動的過程の計算法」

では、どうすればよいのか。じっさいにプログラムで計算機を動かすしかないのです。プログラムがあって、生データがあって、動かすためのアルゴリズムを考えます。これをどうするかといったら、プログラムのなかに埋めこんで書くのです。

書くとどうなるか。プログラムを動かすと、い ろいろと予期しない状況が出てくる。これに対応 動的過程の計算法
プログラム データ
アルゴリズム データ
アルゴリズム ● まき換える ● 実行する
連続運転性は??
アルゴリズムを
ハードコードしてはいけない!!

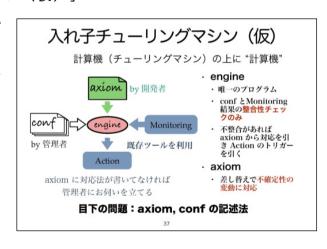
するために、またアルゴリズムを修正する。修正したアルゴリズムを動かすためになにを しなければいけないかというと、プログラムを止めるしかしようがないのです。止めて、 書き換えて、もう一回動かします。

いまは、ほとんどがこのやり方です。だから、じっさいの現場のSE(システム・エンジニア)さんはひじょうな忙しさで過労死寸前の人が数多くいます。これに、さらに連続運転性という問題が被さろうとしています。現場のSEさんは、ほんとうにたいへんな顔をしています。

結局ここからいえることは、インフラになるような連続運転性を担保しなければならないものというのは、アルゴリズムをハードコード(hard-code)してはいけないということです。プログラムのなかにアルゴリズムを書いてはいけない。

◇スライド37「入れ子チューリングマシン(仮)」

それで、どうするか。私が3年くらい前から言っていることなのですが、計算機のうえに、もう1回計算機をのせようという話です。仮想化技術があるではないかということにもなりますが、これは計算機を仮想化するというよりも、計算環境そのものをもってくるものなので、ぜんぜん考えが違います。計算環境そのものというのは、要らないものまでいっしょにきてしまう。むだが多い。



どうするかというと、まずエンジンがあって、これになにを食わせるかというと、アキソム (axiom) というたいそうな名前をつけましたが、ルールを書いたものだと思ってください。ルールを規定したものを用意しておく。じっさいに動かすときには、こういう条件で動かしましょうというコンフィグレーション (configuration、conf) を指定しておく。



あとは、対象となる系をモニタリングします。もしも、想定している状態と違ったら、なにかアクションを起こしましょう。こういう構造にしておけば、エンジンは軽く済みます。チューリングマシン(Turing Machine)そのもののようなもので充分です。エンジンがするものは、このコンフィグレーションとモニタリングの結果です。系の状態と、そうあってほしいというように書かれたコンフィグレーション・ファイルとのあいだの整合性をチェックするだけです。不整合があれば、ルールからこうしましょうというアクションを引いてくる。そうしてくれるツールは、プログラムにいくつかあるので、そういうもののトリガーを引く。

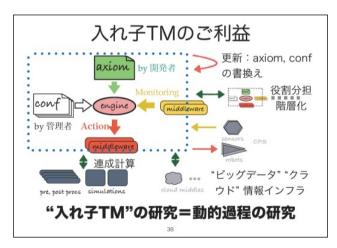
コンフィグレーションは、これまでも動かしながら書いてこられたのですが、アキソム も動かし続けながら書き換えることができるのです。ルールに書いていないことに関して は、書いていないというアクションを起せばよい。そうすれば、人があらためて対応する。

これでいろいろとおもちゃのようなものをつくってやっています。おもちゃのようなものと言いましたが、次の5年間で七帝大のあいだを継ぐインタークラウド (intercloud) の管理ツールとしてつくらなくてはならないものなのです。こういうものに向けて、いましています。

目下の問題はなにかというと、アキソムと書いたはよいのですが、まだ絵に描いた餅なのです。どのように書けばきちんとシステムが回るかというところが、むずかしいです。こういうものをつくっておくとなにがよいか。自己言及的なことができるようになる。動かしながら、自分自身を変えることができるようになる。

◇スライド38「入れ子TMのご利益」

自分自身の分身のようなものを役割分担して置いておき、階層化することもできるし、ほかのところに置いておき、移すこともできる。あとはセンサとロボット、あるいはクラウドなどで、ビッグデータやクラウドなどのアセンブリもつかいます。連成計算のエンジンにもこういうものがつかえます。状況が刻一刻と変わるところで、対応する。



なぜ、このような研究を数年来していたかというと、こういう構造やエンジンを考える ことが動的過程の研究とオーバーラップしているところが多いからだろうという気がしま す。



◇スライド39「最後に:研究と歴史と社会」

最後です。ぼくは理学の出身ですが、情報インフラなどと関わると、また見方が変わってきました。蒸気機関と電磁力学の関係は、われわれが昨日きょうと議論しているところと実社会のインフラに入りこもうとしている技術とのあいだの関係に似ているように感じます。

プログラムのあたまにありますが、「想定

最後に:研究と歴史と社会

(東社会) 研究

(情報インフラ) 動的過程

(素気機関) 熱力学

(加速定外」の問題が次々と出現することこそ、創発の本質と捉え、問題を創り出す創発現象を逆説的に利用することによって、問題解決がはかられる可能性についても議論する。…

「情報インフラ」開発では、待ったなし!!

外の問題が次つぎと出現することこそ、創発の本質と捉え、問題を創り出す創発現象を逆説的に利用することによって、問題解決がはかられる可能性についても議論する」と。これが、じっさいに待ったなしの状況になっているように思います。このままゆくと、現場の人間が何人過労死したらよいのだろうというような、ぜんぶ現場に問題を押しつけるようなかたちでドーッと進んでいます。こういう状態で、いろいろなプロジェクトがどんどん進んでいます。

昨日、佐々木節さんが「ゆっくりやって失敗してもいいよ」とおっしゃっていましたが、 こういう研究会の成果が切実に待ち望まれている分野があるのだと。これがみなさんの日 常生活に深く関わっているのだと。こういう世界があるのだということをご承知いただけ れば幸いです。以上です。

三輪●ありがとうございました。

A氏●すごく賛成するのですが。ちょっと気になるのは、このようなことをしているのは グーグル (Google Inc.) だから、言ってもしようがないのではないかというのが正直な話 なのですが。

ぼくは、何年か前にグーグルのヘッドクオーターに行きました。まさに、24時間365日止まらないようにしていました。言うのは勝手だけれども、グーグルがこれをガンガンとしているのです。だから、これは問題にならないということが問題なのではないですか。

小林●なるほど。一つ、だいじな話があって。グーグルならグーグル――どこでもよいのですが、ひと企業でしているということは、つまりイントラなのですね。

A氏●それを問題にするのではなくて、ここで力学的な問題があるとするのならば、ページランク(PageRank)にしてもGSS(Google Site Search)にしても、そういうアプリを開発して、なかを走っているソフトウェアと関係なく、ハードウェアのうえでソフトウェアを24時間止まらずに走らせることをグーグルがしている。

だから、そういうことはグーグルが解決してまわしてしまっている。そういう意味では、



プラクティカルな問題として考えなくてはいけない。もうなされている問題なので、これをいま問題とするのは、なんとなくアウト・オブ・デートな気がしてしまうのです。そのへんはどうですか。

小林●逆にこういうものが情報のアカデミアのところでまったく出てこない。

A氏●情報のアカデミアというところが気になっているのです。

小林●そうです。私が言ったのはそういうことです。

A氏●そんなものがあるのかということが問題です。もう一つは、最後のお話はバーチャル・マシン(Virtual Machine、仮想機械)でガンガンつかっているではないですか。

基本的に情報のアカデミアという問題は、いま実行されている問題だと思うのです。机上の空論的な問題は、もう…。だからこそグーグルは偉いのですが。ディープラーニング (Deep Learning) もそうです。アカデミアで考える前に走って、論文にする前にブログに あげてしまっている。24時間ノンストップでと考えているあいだにシステムをものすごい 量で構築して世界をつくっているというのが現状だと思うのですが、そうは思われませんか。

小林●もちろん、技術はあるなと思います。バーチャル・マシンを止めずにする技術があるということはわかるのですが。同じものを、ほかのところにライブ・マイグレーション (Live Migration) して、動かしながら変えることができるということはわかります。しかし、ウェブでなにかサービスをつかっているとして、サービスを止めずになかをバージョンアップすることは無理ですね。

A氏●たとえば、Gmailなども日常のものではないですか。鉛筆や水道のようなものになっているのです。小林さんのお話は「もっとよい水を」みたいなセカンド・オーダーの話のように聞こえてしまったのです。ファースト・オーダーでつくったグーグルが偉くて、それでもう既に世界が統一されてしまっています。「もうちょっとおいしい水を飲みたいよね」というような話に聞こえるのですよ。

基本的な部分は構築されてしまったのです。情報科学などで、大学で研究するものはなにが残っているのか。あるいはアカデミアとして研究するものは、ほんとうにあるのかということが問題になっているのではないですか。

たとえば、計算パワーが追いつかない。でも、それとは別にグーグルがつくったページランクやグーグルボットから始まって、いまおっしゃったことを実現して、走らせて、つくっているのです。だから、いまお話を聞いた結果は「でも、それはグーグルがやりましたよね」としか答えようがないような気がしてしまうのですが。

小林●いや、そうではないですね。

A氏●そうではない部分というのが、現実がわからないのですよ。

小林●なんと言ったらよいのでしょうね。24時間連続運転などの話はできているのですが、



そのしかたはこれまでの計算機のつかい方の延長線上ですよね。プログラムをどのように 書いているのか、アルゴリズムをどのように計算機にのせるのか。みんなプログラムにコ ードを書いてしますよね。

しかし、これをしているかぎり、アルゴリズムが変わったらプログラム自体を変えなくてはいけない。それでもいいじゃないか、サービスがつかえたらいいじゃないかという話だったら、話はそれで終わります。けれども、こういう計算のしかたではなくて、動的なもの、不確定なものが出てきたときに、これらをどのように除くか。こういうことをするときに、プログラムさせておくのは判定するものだけにして、アルゴリズムもプログラムの外に出せばよい。

A氏●そもそも、ぼくがどうしてこの質問をしたかというと、前半の話はすごくおもしろくて、基本的な問題とも絡んでくる。ジェットがあって、どうやって求めるのかと。グーグルは、まさにこれと同じように、プラクティカルにGmailを送るという問題を、後半のお話のとおりに解いているわけです。

前半の問題のときに、すごい流体力学があって、境界線モードを書かなくてはいけない ねといったら、抽象的でもそういう問題もあると思うが、フルートの問題にしたとたんに、 ワーッと立ちあがるではないですか。

同じことをしたのがグーグルです。いまお話された感じは、なんとなくぼんやりしているではないですか。しかし、グーグルがまさにその問題に関して10年前に走らせて構築した構造が既にあるとぼくは思うのです。

小林●ご質問の意図を捉えきれているかどうか自信がないのですが、きょう私がしたような話をなぞったようなところがじっさいにできているかどうかというと、グーグルはかなりのことをしています。これは議論の余地がないです。

たとえば、軽くて高速に動くようなものがないと連成計算ができません。ns(ナノ秒) くらいのところでバーッとやりとりして、計算しないといけないものは、バーチャル・マ シンのような重たいものではもちろんできない。軽くて小さなものがどうしても要る。

しかも、nsのオーダーで不確定要素に対するものを処理しようと思ったら、自己言及的なものをなにかしら入れないと無理でしょう。旧来のようにアルゴリズムをプログラムにダーッと書くというのはやめ、そろそろパラダイムを変えませんかということが言いたいところです。

三輪●まだ続きそうですが、終わります。どうもありがとうございます。

(了)