

〈展 望〉 計算センターをめぐる

森 口 繁 一*

ま え が き

本日の特別講演を頼まれましたときに、題目をきめるまでの時間が大へん短かかったので、つまり電話を待ってもらっている間にとっさに思いついたのがあの題目であります。あとで考えてみまして、なぜOR学会で計算センターの話をすることにしたのか、ちょっとおかしいという気はいたしましたけれども、さらによく考えてみますと、これは大へんうまい題目の選び方であったという気が次第にして参りました。というのは、一般にびたりと専門に合った話をしますと、とかく間違えたり変なことを言って突っ込まれたりして困ることがあるので、OR学会では計算機関係の話をし、計算機に関係した学会ではORの話をするという方法をとるとその難点から救われるので、とっさの反応としては大へん賢明な反応をしたと思っております。

ところでまた「計算センターをめぐる」という題がなかなか傑作でありまして、それを見ただけではおよそ2つの意味のどちらであるかがわからない。1つの解釈の仕方は、外国を旅行して方々の計算センターを見て歩いた。つまり計算センターをめぐるってきたのでその話をするのであろうという解釈です。これは正しい解釈です。それからもう1つの解釈は計算センターというものをめぐっていろいろな問題点が考えられるので、そういう話をいろいろとするのであろう。これもある程度正しい解釈であります。その両方の解釈、どちらにもとれるようにしたのも大へんうまい表現であったと思っております。

そんなわけで、実は1年ほど前9月2日に羽田を発ってヨーロッパをぐるぐるとかけ歩きましてから、9月の末ごろニューヨークに着いて、それから9カ月間ニューヨークのコロンビア大学におりました。あとアメリカ国内を少しは旅行しましたが、夏の間2カ月をスタンフォード大学ですごしまして、この8月の末に帰って参りましたのですが、その約1年間の旅行の間に計算センターというものをだいたい見ました。

でもどのくらい見たか私も数えてみたことはなかったのですが、昨晚このピラを作りましたらこれだけ見たところがあることがわかりました。もっとよく調べるともう1つ2つ出てくるかもしれませんが、大体これが実際に見た計算センターの表であります。1行が1カ所ですからどのくらいになりますか。イギリスで5つ、オランダで2つ、ドイツで4つ、スイスで1つ、イタリ

* 東京大学工学部 昭和36年11月4日第10回秋季研究発表会特別講演

アで1つ、アメリカで6つということになっております。そういうところをめぐったときの印象といったようなものをお話しすれば、第1の意味の計算センターをめぐっての話になるかと思えます。しかし実をいうと、そういう話というのは行って来た人ほどには聞き手には面白くないものなんです。一般に、行って来た人は何月何日に何というところで何という人と握手をしたなんていう話はそれは面白いに違いないのですけれども、行かなかった人は握手をしたかしたなかったかなんということは大して重要なことでありませんし、あまり本人ばかり面白がっていてもしょうがないことです。それでこういうのにばかり時間をとったのでは、皆さんが退屈なさるのではないかと考えたのです。

そこでもう1つの意味での「めぐって」の話をするために、1つの例題として私がスタンフォード大学で出発の10日ほど前からかかって出発直前に結果を得た計算がございまして、その計算の実際のプログラムについて少し時間をかけてお話ししたら面白いのではないかと思います。1つには、つまり無料で計算機のプログラミングの講習会をすることになります。無料であるのはいい点なんです、あまり役に立たないという欠点を持っております。これは大体においてアルゴルという今国際的に共通の言語として使おうという気運の高まっている言語なんですけれども、その言語の58年版のやつを使っておりますので、60年版でないとこれから役に立たないのです。そんなわけで多少材料が適切でない点がありますが、しかしいって言えば、これがある程度気持だけでもわかっておれば、60年版のアルゴルを理解する上に相当役に立つだろうということ。それからまた話の中ではプログラムだけではなくて、およそそういう大学の持っている計算センターで計算をするということがどんなふうに行進するものかということをお話ししたいと思っておりますので、それが計算センターをめぐっての利用者側から見た話ということで、興味を持っていただけるのではないかと考えております。

* イ ギ リ ス *

まず第1の部分をお話しいたします。イギリスのケンブリッジ、これは比較的早くから計算機並びにそのプログラミングに関する研究をしていたので有名であります。そのEDSACという機械は東大のTACなんかの手本になった機械でありましたが、私が行って見ましたところ、初代のEDSACはもう解体されて価格10ポンドで処分されたそうです。10ポンドというと日本のお金にして約1万円ですから大へん安いことになってしまったわけですが、今はその2代目のEDSAC IIという機械が動いております。これはかなりよく動いております、2000語ぐらいの記憶容量を持っているのですが、そのうちの1000語を入力ルーチンに使ってしましまして、残りの1000語だけを実際の問題に使おうというような使い方をしてる。この点私は意外に思うと同時に大へん愉快に思いました。というのは、日本の機械について記憶容量の半分ぐらいを使う入力システムを作ることを方々にお勧めしていた関係で、そういうようなことを実行しているのを見て大へん愉快に思ったわけです。

入力の部分(例えばこんなような命令を読み込んで行く部分)を念を入れて作っておきますと、プログラムを作る時間が大へんらくになるわけなのです。そのかわり 2000 語の機械を 1000 語の機械として使うというのですから、言葉を 2000 覚える力がある機械を 1000 しか覚えられない機械のようにして使うというのですから、随分もったいない使い方をするとも言えるわけです。私、国産機の場合はそんな勘定にはならないのであって、作業用番地としては残りの半分も使えるのだからちっとももったいないくないという議論をしておりますが、EDSAC II の場合は、そんな議論もしないでとにかく半分だけあんた方使っていいのだ、あとの半分は保留だと言い切っているところが実に徹底していていいと思いました。

その EDSAC の状況などを見てみますと、みんな非常に自由な気分で使っている。最初に行ったところでそういう気分を味わったので実に気持がよかったです。次々と問題をかけにくるのですが、問題をプログラムしてそれをテープにする部分はめいめいがやるのです。それを持って行ってかけるのはそういうテストをする時間がきまっています、そのとき 10 人も 20 人もきてテープを持って待っているのです。それで時間がじゅんぐりに割り当てられますので、そこへ行ってかけてみる。われわれが計算機を使う場合、今までの例は機械にかけて、大抵うまくいきませんから、その場でいろいろ考えるのです。どうだろうか、こうだろうか、ここがああなっているのではなからうか、ここがこうなっているのではなからうかと言っていろいろ調べてみたりして、計算機のボタンをあっちこっち押したり、あるいはプログラムを読んでみたりして、そこで考えてそれで又そこでわかればそれをなおすようなテープをちょこちょこっと作って読み込ませて、30 分も、1 時間もねばってやっと通ったとか言って引きさがってくるのですが、そういうやり方はしていない。そうではなくて、問題を通しますと機械がしばらく考えていまして、(さっきの入力ルーチンでプログラムを読み込むわけですが、読み込むとしばらく考えていて)そうしてやにわにバタバタとテープが穿孔されて出てまいります。いわゆる電信テープですからテレビをごらんの方は海外ニュースの時間などにつなぎにバタバタバタと機械が打っているのをごらんになると思いますが、あれと同じ電信テープを使っているわけです。それに穴をあけたものがサーッと出てくるのです。1 秒間に 30 字の速度のものを現在使っているというのですが、そのうちに 1 秒間に 300 字打つようなやつを使うと言っておりました。とにかく相当な速さでテープが出てくる。ですからプログラムをした人はそのテープを受け取って帰るだけなので、それ以上はそこにはいけないのです。受け取ってきて今度は別のところにそのテープを読んで字を打つタイプライターがありまして、それにかけると初めて機械が何を言ったかがわかるような仕組みになっているわけです。

ですから何が何やらわからない形で機械の答は受け取ってきて、タイプライターにかけてこれを読む。これは遅いですがこれも機械の時間に無関係です。しかも又それはずして自分のところに行って考えるわけです。だから割合落ち着いた気持でプログラムのミスやなんかを考えることができる。そうして又それによってわかったことでおしてかけるのは 2~3 分、せいぜい 5 分

くらいですんでしまう。こんな形でテストが進行するのです。

そうしてもうこれで大丈夫という状態になったら所定の紙に指図を書いて、そのテープを添えて壁にかけておきます。掲示板がありまして、そこに3センチおきぐらいに穴のあいた板があります。今台所などにも使うそうですが、そういうのに鍵がずっとさし込んであります。ほどよい間隔で、そこにぶらさげておくと、テスト以外の時間に専門の操作員がそれをはずしてはかけて、1時間かかろうが2時間かかろうが、その所定の時間だけちゃんと操作して結果を棚に入れておいてくれるという仕組みになっているわけです。

こういうやり方はやはり計算センターの正常の使い方として手本にすべきだなと思いましたが、これは勿論機械の性能にもよるわけですし、それからそういうことのできるようないい検査のプログラムが出来ているということが前提条件になりますので、にわかにそうやってうまく行くというわけではないのです。EDSAC II というのは今日の水準から見てそうすばらしい高性能とは思われないのですけれども、それでもそういう使い方ができるということをよく見て感心して参りました。

なお新しい附属装置としまして、EDSAC II にブラウン管の表示装置がついておりまして、そこに何かプログラムで文字だの数字だの、あるいはある特定のメモリーの内容だのをださせて観察することもできる。私が行きましたらそれにちょっと余興だとか言って、ある物を出してくれました。それは飛行機と潜水艦の戦闘のシミュレイトですね。テレビの受像管みたいなやつのもう少し小さいのに絵が出てくるわけですが、左手から飛行機が現われてきて、右へ向かって飛んで行くのです。それからここに潜水艦が現われて大砲などもっかけておりますが、これが進行していく。勿論飛行機の方が速くて潜水艦の方が遅いのですが、それから、スイッチが操作板にありまして、左側のスイッチをパッと入れると飛行機から爆弾が落ちるようになっております。それから右側のスイッチを上げると潜水艦から大砲の弾が出てくるようになっておりまして、そのスイッチをパッと入れたあとは飛行機からの方はちゃんと計算してある弾道に従って、あるいはその都度計算するのかもしれませんが、ちゃんと放物線を描いて爆弾が落ちていくようになっております。潜水艦の方も、高射砲の弾もやはり放物線で、曲り方は少ないですけども、飛んで行くようになっている。そうしてうまくあたりますとさっと相手方が爆発してなくなってしまうのです。一度きりですとチャンスを失ったらつまりませんから右の端に行ってしまうと次のが左から出てくるようになっておりまして、何度でもくり返して出てくる。なかなか面白いのでして、ちょうど弾道の具合を考えながらいいときパッと入れるとうまく当たってくれるのですが、私がやりましたら飛行機になったときも、潜水艦になったときもうまく当りまして、大へん射撃がうまいことになったのです。私を案内してくれた人がこのプログラムができているのをまだ知らなくて、こんなのができているのかと言って面白がっておりました。作った方ではフルシチョフが来たときはこれはやらないつもりだとか言っておりました。

その次の日に Rothamsted の農事試験場に参りました。これは例の R. A. フィッシャーとい

う近代統計学を始めた人が画期的な仕事をした場所なんです、その Rothamsted 農事試験所に行きましたら今はフィッシャーの後継者のイエイツが統計部門の部長をやっております、そこに計算機が1台あります。それがエリオットの401という計算機で今の日本の国産機と比べても大分低性能で大へん使いにくいような機械なんです、それを7年半ばかり使って、今でも毎日ちゃんと使っているのです。これは性能が低いとか古くさいとか言わないで、ちゃんと使っております。もうちょっと優秀なマーキュリーか何かを入れる計画があるのだそうですけれども、それはまだ2年くらい先になるだろうとイエイツは言っております。そういう状態で私の見たところでは非常にうまく使っているようでした。

農事試験の解析をするのに卓上計算機を使う人はほとんどなくなったということを言っております。なんでもそこへ持ってきて計算機で解析する。その結果大きな実験をいやがらずにするようになったということを言っております。解析が難しいからという理由で、能率のいい計画をやめるとかあきらめるとかということが以前にはあったが今はない。解析だけは安心だということでやっている。その状態にもっていくためには、というのですから、どれが原因で、どれが結果だか知りませんが、とにかくその統計関係のプログラム・ライブラリーの整備状況というのはすばらしいものでありまして、もうなんでもあるのです。

面白いのは一般用データ処理ルーチンというようなものがありまして、どんなデータでも一応それにかける。例えば平方根変換とか逆正弦変換とかあるいは対数変換というデータの変換はそれで行うことができる。それから要因分析計画などでやった場合には水準を、例えば4水準でやるときに、1, 2, 3, 4 と水準の名前を書く人と、0, 1, 2, 3 と書く人とあるのです。これは駅のホームでも0番線から始める大阪駅みたいなのと1番線から始める東京駅とがあるのと同じような原理であります、そのときに0水準から始めた人と1水準から始めた人で解析のプログラムを変えろというのは大へん厄介ですから、これを事前のデータ処理のときにおおして標準化してしまう。何かそんなような一般にやらなければならないことをみんなまとめた一般用データ処理ルーチンというのがあって、それにまずかけて処理をする。それから特定のプログラムに入るので、これがまた多元配置の一般のプログラムもあれば2元配置専用の能率のもっと高いプログラムもある。それから一般の要因分析のプログラムもあるし、2のN乗型専用とか3のN乗型専用の、特定の目的には非常に能率がいいというプログラムもある。そういったふうなあんばいで、各種各様のプログラムが用意してあります。

イエイツさんの言うところでは、こういう研究所の電子計算機というのは事務手続きを面倒にするとなかなか人が使わないものなんだ。だから自由に出入りして勝手にいじれるようにしておくのが一番よろしいということです。実際それを実行して、その結果所員たちは大へん計算機になれ、親んでいるようでありまして、われわれが考えると偉い人で自分でプログラミングなんかやりそうもない人でも、ちゃんと自分で機械にかけています。面白いことですからやり始めるともうやめられないので人になんかやらせられないという気がするのかもしれませんが、大いに楽

しんでいる様子がかがわれました。

イギリスは貧乏な点では日本と似ている。どっちがどっちかよくわかりませんがアメリカなんか比べてともかく貧乏なんです。そういう国では安い機械でも古い機械でも一生懸命使うというのは当然な話なので、日本なんか決してアメリカみたいな金持ではないのですからやはりこんな点を学ぶべきではないかという感じを私は持ちました。もう自分のところよりも速い計算機がよそに現われると、自分のところの計算機は使いたくなくなるというのではとてもいかなんではないかと思いました。

その次はロンドン大学のマーキュリーですが、これはずっとこんなのよりは高性能でありまして、コア・メモリーを持っている。それでやはりそれだけに非常にどんどんと活用されておりまして、ロンドン大学というのはご承知のように大きな組織でありまして、その本部というのは管理組織だけしかもっていない、管理組織であり最後に試験をする組織である。実際の教育は沢山のカレッジで行なわれているわけです。何十とあるロンドン市内方々に散らばっているカレッジで行なわれている。そのロンドン大学計算センターというのは沢山のカレッジ全部に対してサービスをする使命を持った中央の計算センターなんです。

その次のパークベック・カレッジというのはロンドン大学のカレッジの1つであります。ここに又面白い計算機に関係した学科がありまして、それが数値的オートメーション学科というような名前なんです。そこの主任教授が Booth という人でありまして、本など書いていて有名です。奥さんも数学者でありまして、旦那さんと奥さんの共著の本もあります。それから旦那さんだけの本もあります。この旦那さんだけの本は、数値計算の本で宇田川さんが訳した訳本も日本で出ております。非常に面白い本で、その本が出た当時はほかでは見られないような面白い考えが沢山あったので私ども非常に喜んで読みましたが、とにかく閉口したのは非常に間違いが多い。その書評の中に奥さんのイニシャルを間違えていることが指摘してありましたが、その後第2版のときに訂正されまして、今は正しくなっております。

そのミスター Booth に会いまして、向うの計算機マックスとか M2 とかというのを見せてもらいましたが、小型ですけれども特殊な目的に合うように作られておりまして、特に M2 は言語学の方の応用にいいような話でありました。

その次の NPL というのはイギリスの国立の物理学研究所ですが、その中で計算機を持っている部門がある。実はそこへは行かなかったのですが行きたいと思っておりまして、ローマで、その人と一緒に食事をする機会があったのを幸いに少し様子を聞いてみました。そうしたらそこではエースという機械とデュースという機械がある。どっちが先できたとお思いになりますか。エースというのは1に関係した、ナンバーワンというような意味ですね。トランプのマークが1つある札には A と書いてありますがエースの A なんです。それで巨人のエースとかなんとかという使い方もあるわけです。とにかく1番ということで、それからデュースというのは2つというのに関係ある言葉でテニスなんかでも使われております。従って1番、2番という名前ですが、

さて機械はどちらが先にできたでしょうか。こんなふうに分ければ人はおそらく順序は逆でデューズの方が先にできたのかな、と思うでしょう。実はそうなんです。で私は NPL の人に会ったら、デューズは前から使っているのだが、エースはまだ僕達が使う段階になってないというから、逆じゃないかと聞きましたら、話はこうだと教えてくれたのは、エースという大きな機械を作ろうとしたのですが、そのためにはまずパイロット・モデルを作った方がいいというので小さいのを作ったのだそうです。そうしたらその機械が非常によくできたので、とりあえずそれを売り出してお金をかせごうじゃないかということになって、それを作った会社がデューズという名をつけて売り出したのだそうです。これが何十台だか売れたらしいのですけれども、大きい方はなかなかできなくて最近形はできあがって近く実用できる程度になった。こういうのはよくあるのですね。あとから作り始めたのが先にできてしまうというのはよくあります。

これらを通じて一体どんな言語でプログラムしているか興味をもって聞いたり見たりしたのですが、大体において機械語に近い言語、機械の中で命令が記憶されている形に近い——近いというのはつまり 1対1 に対応しているという意味で近い——そういう言語が使われております。実はそれは原始的なやり方なのでありまして、あとでお話するような、より知的水準の高い言語に比べるとプログラムを書き下すのに大へん面倒が多いわけなんです。習うのにも時間が余計かかります。こんなのが 5 時間で習えるものとすれば、今の機械語に 1対1 に対応する言語というのは 50 時間かかるというくらい学習に時間がかかる。そうして実際のプログラムを書くのにもそれに近いくらいの時間の違いがあると私は思っておりますが、それにもかかわらず何故機械語と 1対1 に対応する言語がイギリスで非常に多く使われているか。これは計算機そのものの記憶容量と使用時間に関係するのでありまして、知的水準の高い言語は習いやすく使いやすいという反面すこし能率が悪いという欠点があるわけです。記憶容量でいっても計算時間でいっても、何割かという程度の違いがあります。この違いが 2% とか 3% とかいうものと第 1 の利点の方にカバーされてしまうと思うのですけれども、無視できない程度の割合、2割とか3割という程度の能率の違いがある。そこでイギリスではそういう知的水準のより高い自動プログラミングに対してある程度の研究や開発は行なわれておりますけれども、それが全面的に実用されるというところにはいっていないし、それから専門家の間でも、そういう方へいくことが果して必要かどうかということにかなり疑問があるようでございました。

* オ ラ ン ダ *

さてそれからヨーロッパ大陸の方へ行きますと、オランダのフィリップス、これはヨーロッパでも有数の大きな電気の会社でありまして、名前はフィリップス電球会社ということになっておりますが、電球に限らずラジオでもテレビでもそれから重電機でも作っております、電気の総合メーカーです。そこには今まで計算機と言えば IBM の 650 を使っておりましたが、最近もっと大きな 704 というようなクラスのものを作っております。私が行きました当時はまだ秘密だと

いうので、いろいろと要目など聞いたのですけれども、公表してくれるなどと言われましたが、今年の2月に発表になりまして、もう今ではだれでもいけば見せてもらえますし、その資料などもくわしいのが私のところへ届いております。それはほとんど同じような形のを2つ作って、片方をステビンと名付けて、片方をバスカルと名付けてある。こちらが事務計算用でこちらが科学計算用だそうです。

もっともこれがフィリップスで作った大型の計算機の最初ではないのでありまして、今までにも1台あった。それはどういうのかと言いますと、これはハマカーさんが来たときにOR学会のセミナーでもお話しになったのであるいはお聞きになった方もあるかもしれませんが、このフィリップスという会社は先ほどのようにヨーロッパで有数の電気のメーカーだものですから、ヨーロッパ中の計算機製造会社に部品を納めるのですね。部品の注文がどんどんくる。それに対して設計上の指針を得るため、それから又万一苦情が起ったときその苦情に対する対策を考えるとというような目的のためにテスト用の計算機を作ってもっていたのです。計算機に使う部品なら計算機の中に入れてテストするのが一番よろしいというような考え方で、部品をテストするために計算機を作った。それがかなり大きなものであったのだそうですが、そういう経験もありますので、今度の大型機もそう大して不連続的な冒険というわけではなかったようであります。

それからアムステルダムの Mathematisch Centrum というところへ行きました。これはオランダの数学センターであります。オランダ国政府とアムステルダムの町とそれから民間の産業界とが三者協同して維持している応用数学のセンターです。応用数学と言いましても、非常にすぐれた純粋数学の学者もおるのでありまして、純粋と応用を一丸とした数学の研究所として、日本でも数理科学研究所というのを作りたいという要望が一部にあって実現するかもしれないような情勢になってきましたが、そのいわば手本みたいのがここにある。スカウテンという人が音頭をとって始めた仕事であります。今でも相当優秀な所員がおりまして、たっぷりと予算をもらって大へん楽しく仕事をしております。その計算機は自分のところで作ったアルマックというのと、その後最近試作した X ワンというのがあります。アルマックは小さくて大して面白くないですが、この X ワンというのは非常にうまくできましたので、その後会社を1つ作りまして、お金は保険会社が出したらしいのですが、会社を作って(その数学センターの看板と並べてその計算機会社の看板が出ておりますが、エレクトロ・ロジカという会社を作って)、そこでこのいわば流れ作業生産と言いますか、大量生産を始めて、もうすでに10台ぐらい動いておるということを、1年ぐらい前に言っておりましたから、今ではもっとふえているかもしれません。自分のところへはそれの第6号機を入れておりました。何故1号機を入れないで6号機を入れたんですかと聞きましたら、ほかの要求が強いものだから譲ったのだと言っておりましたが、しかしあなたの方にとっても賢明な政策でしょうと言ったら、そうだそうだといってニヤニヤしておりました。というのは初めの方には故障が多いのですね。6号機ぐらいになると大体安心して使えるようになるので、そういう意味で非常にうまいのではないかと思います。

このセンターでまず非常に感心したのは、ここが例のアルゴルという国際代数言語、アルゴリスミック・ランゲージという意味なんです、これのアルゴとLをとったのですが、そういう言語を受け付けるようなプログラムができ上がっておりまして、そのダイクストラというひげをはやした非常に熱情のあふれる人が作ったので、自分でいろいろ説明してくれました。そういうことをいち早くやって知的水準の高い言語でプログラムすることを積極的に押し進めている。そういうセンターの1つでありました。

* ド イ ツ *

ドイツではダルムシュタットの工科大学とマインツのヨハネス・グーテンベルク・ユニフェルジテート、それからミュンヘンのテーハーそれからミュンヘンにあるマックス・プランク・インスティトゥート、これは実はこの間までゲッティンゲンにあったのでゲッティンゲンのGをとったゲーツワイとかゲードライとかという、そういう機械を持っているのですけれども、現在は所長のハイゼンベルグがミュンヘンの出身なんだそうできて、その関係もあってミュンヘンの郊外にすばらしい建物を建てて引越してきております。そんなわけで私はゲッティンゲンに行きそこなったにもかかわらずこの機械を見ることができたのですが、これだけをドイツでは見ました。

いずれも何かちょっとイギリスに比べると遅れて出発したような感じがはっきりとしたのですけれども、それからもう1つはいかにもドイツ式に堅牢なものを作っているという感じがいたしました。叩いてもこわれないようなものを作っているような気がしましたが、何か一歩遅れているような気もしないでもなかったです。ただアルゴルに関してはいずれも非常に熱心でありまして、ダルムシュタット、マインツ、ミュンヘンいずれも積極的にアルゴルのプログラムをむしろ競争して作っているような感じがいたしました。お互いに連絡する組織もできておるようでした。

* ス イ ス *

それからスイスのETH、これはスイス連邦工業大学といったようなものですが、それがチューリッヒにある。チューリッヒ大学とは別です。これは連邦政府のものなんです、そこにERMETHという機械がある。エルムまでが電子計算機という意味でありまして、ETHがこの学校の名前ですね。これが大体でき上がってございました。ここもミュンヘンや何かと兄弟分のような間柄でありまして、非常に密接に協力しながらアルゴルの仕事を進めているといったところです。

* イ タ リ ー *

イタリアーではローマ大学でフェランティのマークワンというのを見ましたが、これはマーキュリーよりももうちょっと前の機械でありまして、そこに使ってある真空管はイングランド戦闘というのをたたかった真空管なんだそうです。つまりドイツ軍がイギリス本土を攻撃して、あるいは今にも地上軍が進入するかに見えたときにそれを迎え撃ったのがこの真空管だということです。

レーダーなどで戦ったわけですが、そのレーダーが今度の戦争に現われた最初ではなかったでしょう。それを戦った真空管で作られた、といっても払い下げ品を使ったというのではなく、それと同じ型の真空管を使ったという意味だそうです。これは大へん古いのですけれども、なんとか使っておりました。

* ア メ リ カ *

アメリカへ行きまして私はコロンビア大学に9月から6月までいたのですが、それは客員教授ということで教えたり研究をしたりしていたわけです。数理統計学科に所属しておりました。教える方の部分は全然無関係ですから省略しますが、研究の方では折角計算機の使えるところにいるのだからと思ひまして、計算機を使うような研究をいたしました。それは先程有水さんから話しのあったような、ダイナミック・プログラミングのような型の計算でありまして、こんなような、ダイナミック・プログラミング型の方程式を解く問題なんです。

現在のことをうまくきめようと思うと来年のことを考えなければいけない。で来年のことをうまくきめるためには再来年のことがあるというようなことになって、結局ずっと先まで考えないと今年のことはきまらない。それはダイナミック・プログラミングによくあるタイプだと思ひますが、そうなってくる。そこで計算機の問題としても大へん難しくなるのでありまして、こんなことをずっとやっていくときに、果して値が得られるかどうかという疑問が起こりますし、それからそれを得るためにはずっと先の方から逆に戻ってこなければなりません。

どこから出発して戻ればいいのかというようなことが計算の計画を立てる上に非常に難しいわけなんです。この問題は統計学に關係して起こった問題でありますので、統計的な直観がききまして、初め私にこの問題を話したプロフェッサーはここにやっぱりダイナミック・プログラミングでよくやるように添字 N のはといった形で問題を現わしていたのですけれどもこれではとてもやりきれないのです。たくさんものを記憶しておかなければなりませんから現在の計算機では先の方までやれないということになります。

しかし私が少し統計的な直観なども使って吟味しました結果、この添字はなくてすむということに気が付きまして、それで話は非常に簡単になってしまったわけです。しかしこういうダイナミック・プログラミング型の問題というのはいたるところにころがっていますし、そういうものの解き方というのは似たようなものが大概ありますから、ダイナミック・プログラミング的のものを考えるというのは大へんいいことだと思いますね。ちょっと脱線になりますが、こういう考え方を創始したベルマン自身に会って講演を聞く機会がありました。そのときにベルマンはなかなか面白いことを言ったですね。先程の有水さんの話にも似たような式が出てきましたが、このダイナミック・プログラミング型の問題の書き方というのは、今当面の興味を引いているものがあるとそれに関係して同じタイプのものが、ちょっとこういうところが変わっただけで現われてくるのですね。そうしてそういうもの間の函数方程式という形で問題が表わされている。そう

というのがダイナミック・プログラミングの特徴ですが、それがつまり形が変わらないでそういう仲間全体の中に問題がはめ込まれているという特徴をもっているわけです。これをベルマンが *principle of invariant inbedding* というふうに表示しておりました。

それから下の方にソーシャル・セキュリティ、IBM 705, 3 セットと書いてありますが、これはちょっと面白いもので、ボルティモアにあるのです。ボルティモアというのはワシントンに近い別の町ですが、そのボルティモアの郊外に大きなビルディングが新築されまして、そこにソーシャル・セキュリティの本部がある。そこはアメリカ中でおおよそ雇われて月給をもらっている人のデータが全部集まっているのです。しかも月々いくらかもらったということが全部集まってくるのです。そこであなたも番号をもっているだろうと言われましたからコロンビア大学でもらったそのカードを出しましたらこれこれと言うわけです。番号がずっと書いてあります。じゃあ、あなたのデータを今出してみるから待っているというのですね。何しろそういうカテゴリーに属する人が何千万人という程度あるわけですね。アメリカ中で、人口は1億5千ですから、その中ですからとにかく1億にはなりません、何千万という人間の中から私の名前を引っ張り出そうというわけですね。どんなことをするのかと思って見ていましたら、実にすばらしいマイクロフィルムの設備がありまして、私のカードに書いてある10桁ぐらいの番号ですが、それをたよりに3分間ぐらいで私のデータを出してしまいました。とにかく鮮かんです。ちょっとこれ(講義用幻灯機)に似たような感じの道具にマイクロフィルムをポッとはめ込みまして、シャッと回すのですが、それがまた端から一行一行読んでいたのではとても間に合わないのですが、うまいボタンの操作がありまして、しばらくいたりきたりしているうちに、びたりと名前が出てくるのですね。そうしてファイルナンバーがはっきりしますからそこでファイルのところへ行くと何月から月給をもらいはじめて、月給がいくらだということがちゃんと書いてあるのです。それで大へん感心しました。それからその番号がわからなくなった人は、名前からでも出せるようになっておまして、そっちも実演してくれましたが、とにかくそういうおびただしいデータをうまくファイルしているという意味であの辺にいらっしゃったらそこへ寄る値打ちがあると思います。これはカミンズという人がこのデータ処理方面の親分をしておまして、この人は大の日本びいきです。いつも日本語で、かたかなで書かれた名刺を所持しておまして、日本人と見ると私はこういう者ですとか言って日本へ行ったときの楽しかった思い出などをしゃべったりなんかして、実に親切にしてくれますからあの辺にいらっしゃったら寄ってみるといいです。705 が現在3セット入っておりますが、もうそれがフルに動いているような有様です。アメリカの経済はここで見るのが一番よくわかるのだと言っておりました。人手を要しない経済の動きなんというのはないわけですからね。

それからナショナル・ビューロー・オブ・スタンダーズは10年前に行きましたときは SEAC 1台だけだったのですが、これは小型の試作機ですが、今度行ってみましたら、それはもう研究用に使っているだけで実用には IBM の 704 を使っておりました。そのほかにまた特殊な目的

でパイロットとか、エイモスフォアとかいうのを開発しておりました。

それから MIT. これはマサチューセッツ工業大学というので、いわば工大関係では東の横綱といった格ですが、そこに IBM 709 をもった計算センターがありまして、これはニューイングランド地区の大学、現在 36 大学が参加しておりますが、その共同利用センターであります。そこでは運営組織に大へん感心いたしました。よく整ってまして、いいサービスを広く提供しておりました。

それから北カロライナ大学。これは 10 年前の私の母校なんですが、そこへ今度行ってみましたら、私が行ったときにはバラックだったところがとりこわされて、そこに建物が延びてまして、その中に UNIVAC の 1105 という機械が入っていました。これは相当大的な機械でその機械はセンサス・ビューロー(国勢調査局)がお金を出して、それからレミントン・ランドの会社も(スペリーランド・コーポレーションの 1 つの部になっているわけですが、そっちの方からも)経済的な援助があって、大学側は建物と人員を提供するというようなかたちで共同で作られたセンターであります。センサスの仕事のためにこれだけの時間を使うというようなことがちゃんときめてありまして、その時間の利用に関してはワシントンの郊外にあるセンサス・ビューローから直接人もき、プログラムも、データもみんなくるわけです。大学の中におかせておいて、大学にも一部利用させながら自由に自分の計算をするというやり方を官庁がやっているわけですね。これなんかなかなかうまいやり方だと思います。大学側では非常に喜んで、スポンサーのないアカデミックな研究などにも十分時間をさいております。それから会社側からの要求として年に 600 時間以上は純粋の教育及び無料の計算にあてなければならないということになっておりまして、そういう方面の教育と研究を盛んにするようにしてあるわけです。IBM 関係も似たような精神で、大学には無料又は半額といったような条件で提供している例が相当ありました。この MIT の IBM 709 も実はそんなんで、ただで機械を入れているらしいのです。ただし、ただということをおおびらには言わないことになっているらしいです。

さてスタンフォードの大学ですが、これはバローズの 220、これらに比べるとちょっと小さい機械ですがそういうのがありまして、そこでは 90% ぐらいがアルゴリズムで使われているようです。機械語と 1 対 1 に対応する言語も 1 割ぐらいはプログラミングに使われているといった状況です。以上で第一の意味でめぐった話を終ります。

** スタンフォード大学 **

次に第二の種類の話をお願いします。スタンフォードでこの機械を使うのにいろいろお金の出し方がありまして、私の場合は私に月給を払うのが、ある統計学関係の研究費の方から出ておりましたので、計算機の代金の方も同じ枠から出してもらうのが一番自然ですし、その研究費を管理している側でも喜んで出してくれるような状況でしたから一番そういう意味で手続きのしやすい方法だったわけです。

そこへ行きまして手続きを聞きましたら、それは主任者がこの紙にサインしてくれればいいのだというわけで、私の名前と研究の目的を書いて、それからあと2時間ぐらい使いたいのだということを書いて、その研究費を管理しているリーバーマンというプロフェッサーのところへ行って、そのサインをもらいました。それだけでいいのです。それをもって計算センターへ行きますと、受付の女性がにこやかに迎えてくळेまして、即座にSの440という番号をくれました。それで何かカードもくれましたが、そのカードはあなたが忘れたときに思い出すためにあるので、人に見せる必要はありませんと言われました。そのS440というのをその後は書きさえすればなんでもやってもらえるのです。

プログラムを所定の紙に書きまして、パンチ室に持って行ってS440と書き込んでありますのでそれを出すと、そこにいるおばちゃんがオーライというわけで受付番号を記入してちゃんと棚に入れます。そうするとほとんどその日のうちにパンチができ上って棚に乗っているといったようなあんばいです。

それで計算機そのものの室も問題を入れる棚と出す棚がありまして、プログラム全部を例のIBMのパンチカードにするわけですが、そのパンチカードに今のようにしてパンチしてもらったやつをゴムバンドをはめてただだまってそこへおいておきます。その1枚目のところにジョブ・カードと称しまして、こういうことの入ったカードが1枚あるのです。Sの440という私の番号、それからJOBという文字、8月14日、1961年というのが計算を依頼する日付、その次の30MINというのはマキシマム30分の見込みだということを示しているわけです。これのあんまり長いやつは夜中に回されるとか、短いやつだと比較的早くやってもらえるとかというような違いがございまして。そのかわりあんまり短かく見積っておきますと計算が終わらないうちにはずされてしまうという欠点がありますので、これを何分にきめるのがいいかというのは一種のORタイプの問題ですが、大して重要な点ではありません。それからお金の勘定は実際にかかった時間でしますのでこれには無関係です。それからその次のミスター森口、これはミスターかミスかミセスか書いておかないと向うで呼び方に困りますからそうしてあって、X2208というのは電話番号です。ご承知のように電話の内線というのはエクステンションと言いますが、それを省略してEXTと書いてもいいのですけれどももっと簡単に略しようと思ったらX2208とします(エクステンションを音で最初の方をとったわけです)。こんなような乱暴な省略記号が割合広く使われております。一番向うの1というのはカードの1枚目という意味でありまして、あとで脱落とか順序の間違いなどをチェックするのに便利で、そんなのが1枚入っておりさえすればあとどんなカードが続いていてもともかく棚に入れておきさえすればそれを計算機にかけてくれる。それで計算機の方にはちょっと色の黒い、つまり人種の違う女の子でしたが、それがいてボタンを押している。プログラムのことはなんにも知らないのです。だからカードが1枚逆になっているために通らないというのでもなおしたりしてくれない。しかしとにかくこっちの希望通りにさっと入れて、こっちの方さえ正しければいきなり計算をやって正しい答えを出してく

れます。まずいときはどこが違っているというようなことをちゃんと機械が指摘して印刷してくれますから、それを見て考えればいいのでして、それからこれは重症だなと見るとそのときの記憶の内容を全部プリントアウトして添えておいてくれますから外で考えるのには大へん便利にできております。

ところで私のやりました問題は水晶の振動に関する問題で、古賀逸策先生から頼まれた仕事の続きなんです。なんでも水晶の振動が放送局の電波の振動数の基準になっているわけですね。そういう関係で水晶の振動数というのを非常に精密に測定したデータが一杯あるのですが、どうもおかしな傾向が見えるという話なんです。

くねくねと曲がったグラフが実際実験上得られるのだそう。それを理論でなんとかかきけるかというのを考えているわけです。で古賀さんの説によりますと曲げ振動とずれの振動とが連成しているのだ、というのですが、そういう関係で方程式を立てると大体こんなようなグラフになる。こっちは幅と厚さの比みたいなものです。でこっちは振動数です。そんなような理論を古賀さんが立てたのです。その数値計算をしてくれないかと言われて、前に東京にいたころ TAC でやったことがあるのですが、プリントをたくさんやらせると何 10 時間もかかるような計算だったのです。

それをここでやりましたら、ここはまず第一にプログラムがアルゴリズムですから割合やさしいですね。普通の数式みたいに書いていけばいいのです。K というのは 2.72423 \$, このダラーマークはセミコロンのかわりでありまして、セミコロンという字がないものですからドルで代用しているのです。コメントというのはどこへでも挿入できるので、これはテストのための数値だということを表わすためにコメント、テスト、本番のときは数値が変わるので、このカード 1 枚さしかえれば本番になるわけなんです。それからギリシャ字がありませんから PI で π を表わして、3.1415927 とここまでかけます。それから HFPI(ハーフ π) というのは π を 2 で割ったものだというふうにして勝手に記号をきめて、たいくつしてくると自分の恋人の名前を使ったりするのだそうですが、かまいません。何でもいいのです。それから A1 というのは π 二乗分の 8 であると、A2 は 3 倍の A1 かける 1 マイナス A1 であるというふうにして、なんでも解きたいものをずっと数式に書く、B1 は A6 と A8 をかけたものから A7 の二乗を引いて、それから A1, A8 の積マイナス A2, A7 の積を引いて、こんなものを足したものと、普通の数式の通りに書けばそれでプログラムになるのです。それからこういうふうにきまして、ずっとプログラムが続くわけですが、INTEGER NIT, これは整数はこうやって指定しておくことになっていまして、整数と実数では扱い方が違うものですから外からこうして教えてやります。NIT というのは反復回数をかぞえるために使っただけです。それから ARRAY. これはデクレアレーションと呼ばれますが、以下のものはベクトルであると、NURTYR というものは 4 つの要素からなるベクトルであるということを表わしているわけです。

それからその次の FOR M=30, 2, 60 というのは現在のアルゴリズム 60 では、FOR M: =30

STEP 2 UNTIL 60 DO と書くのですけれども、要するに 30 から始めて 2 ずつ増して行って 60 まで計算してほしいのだと、そういうことです。そういう M に対してこれから先この BEGIN からこれに対応する END までやってくれということなんです。それでこのビギンとエンドというのはこの頃交通標識にもありますね。速度制限 30 KM/H BEGIN と書いてある。それは「ここから」、それから END のところには「ここまで」と書いてございますが、それと同じやつをここで使っている。これはアルゴル 60 でも同じです。BEGIN, END はプログラムの括弧みたいなものです。そうしてまた、NU というものは最初 M とおく、それから NIT は最初 0 とおいて、LOOP3 という、ここからループが始まるわけですが、こういうものを始める。その外 I というものを 2 から 1 ずつ変えて 4 までの間こんなことを行ない、それからまた Y(2) がゼロよりも大きければ分母の第 1 項という意味で DEN1 を、こんな式で計算する。さもなければ DEN1 はこの式で計算するというふうに変えて条件によって式をかえることもできます。それで DEN2 はこれ、DEN3 はこれ、NUPRIME、この式が実は肝心のかなめになる超越方程式なんです、NUPRIME というのが次のニューの近似値です。こういうような式で反復するわけです。それでニューを入れるとこれが計算されて新しいニュープライムが出てくる。で反復が一つ進みます。そうして、もし $ABS(NU - NUPRIME)$ が 1 かける 10 のマイナス 5 乗より小さいならば CONV に行く。CONV というのは converge したときに行くところ。それから 100 回かけてもまだ収束しないようだったら OSC というところへとんで処理しろと、こうやっているわけです。さもなければニューをニュープライムにおいて LOOP3 へ戻れというわけで、これをくり返します。こんなことで超越方程式の反復解法ができるわけですね。収束すれば CONV というところへとぶし、収束しないで 100 回反復したらば OSC というところへとんでそれに相当する処理をする、OSC というのは何をするかというと、NUOSC というのを F401 という様式に従って書いて出せという意味ですが、その OUTPUT NUOSC というのは NU と NUPRIME を出すのだと、そしてその様式は、まず NU=、それをそのままプリントして、そのあとへこの NU を 9 桁とって小数点以下 6 桁という書き方で書きなさい。それからブランクを 5 字おいて、それから NUPRIME= というのをこの形でプリントして、このあとへニュープライムの数を 9 桁とって小数点以下 6 桁印刷しなさい、そのあとへ AFTER 100 ITERATIONS というのをこの通りに印刷しなさい。そういうようなことを行をかえて実行しなさいということが書いてあるわけです。だから大体こんなプリントを出したいなと思ったらそれをこんなような形で書けばそのままプリントが出てくるようになっております。ここまで計算機が読みますと、COMPILED PROGRAM ENDS AT 1529、これは記憶装置の番地です。それから PROGRAM VARIABLES BEGIN AT 8857。1 万語の記憶容量をもった機械ですが、私のプログラムは 1529 番地のアドレスまでです、それから変数は 8857 からあとに入っていますから、これからこれまでの間はまだあいているわけで、まだ一杯容量は残っているわけです。こんなことを打ち出してからもう実行に移りまして、ずっと結果がプリントされて出てまいります、そ

の最後のところを抜き出してここに書いたのでこの部分は要するに結果が私が希望したように出てきているというわけです。NU は 60.192792 AFTER 5 ITERATIONS と反復の回数を出させたのですね、大体 5~6 回で収束しています。しかし TAC でやったとき振動が起こっていますから、どうして今度は振動が起こらなかったかをもっと調べたいとは思っております。

ここに RESULT OUT OF RANGE IN SINH と出てきましたが、この計算の途中に SINH が使っている。それが実は変数の変域がそのプログラムに許されている範囲を超えていたという警告がここに出ているのです。しかるにもかかわらず私の結果には悪影響はないのでして、結果がちゃんと出ております。しかしこういうような警告がプリントされて出てくるのはいいことです。何かそれが悪くきくような場合はこれで対策をこうずることが出来ますから。そんなようなことが細かいところまで行き届いているのが、この種のコンパイラーの非常にいい点だと思います。

最後のところに NO CARD NOT READ とありますが、この問題にはデータ・カードがついていなかったで、従って読まれなかったと、それでいいのですが、カードがありますとカードが何枚あってそのうち何枚読んだというようなことを書いてチェックにするわけです。最後にここに \$\$\$ TIME OFF 0311 とあり、前のところに \$\$\$ TIME ON 0305 とあるのは夜中の 3時5分にこの問題が機械にかかった。これが3時 11分に終わったということでこれだけの長いプログラム——カード枚数にして128枚のカードに書かれたプログラム——、それだけのものを読んで機械が理解できるプログラムになおして、その間あらゆる種類のチェックをやって数 10 ページになるプリントを出しているのです。それだけのものをやるのに6分しかかかっていないわけです。コンパイルの時間が短いというのはこのアルゴリズムのシステムの大きな特長でありまして、709 の FORTRAN よりもコンパイルの時間は少し速いようです。機械の性能は低いのですがコンパイルの時間は速い。(それは一つには 709 の FORTRAN は非常によくばっていろいろなことをコンパイルのときにやるために時間がかかっているらしいのですけれども。)ともかくそんなような程度でこれが出てくるのです。

本番はこれから類推して1時間半ぐらいかかるのではないかと思ったのですが(プリントからいってもこれの何十倍というものが出てまいりますので)、90分と書いて出しましたら実際は69分か何かできておりました。これもやはり明方やったようでありました。こういうものは夜は学生アルバイトがやっております、さっきの色の黒いお嬢さんは昼間だけやっています。なおプログラムの最後が FINISH で終るのですが、その前へ STOP というのを入れておきましたら、bad coding と書き入れて(恐らく学生が書き入れたのだと思うのですが)、あとで行ったらそのカードを訂正してストップカードが消えるようにしてからやっておりました。実はストップさせないのが原則なんです。FINISH とあればそこまでくれば用がすんだわけですから、その次のプログラムをいきなり読み始めるようにして運営しているのです、1秒のすき間もなく機械を使おうという精神の表われですね。

これはつまり相当性能の高い機械を使って、相当知的水準の高い言語で、プログラムして使っている1例だと思えます。アメリカの計算センターは IBM の方は勿論 FORTRAN が圧倒的に多く使われておりますし、そのほかの機械でも例えばこのように FORTRAN とか ALGOL とかいうクラスのもので使われる部分が多いわけです。その運営は実に能率的にできております。しかも使う人があまり気をもんだり手続きでハンコをもらうのに行列を作ったりする部分がないように非常に自由に運営されている。しかも各種の統計は機械的にきれいに出てくるような処置がしてあるというようなことをいろいろ見てまいりました。

これで大体予定した時間がまいりましたから一応話しを終えることにいたします。(拍手)

*

*

*

国	計算センター	計 算 機
英	Cambridge	EDSAC II
	Rothamsted	Eliott 401
	Univ. of London	Mercury
	Birkbeck Col.	MAC, M-2
	N. P. L.	ACE, DEUCE
蘭	Philips	STEVIN, PASCAL
	Math. Centrum	ARMAC, XI
独	Darmstadt T. H.	DERA
	Mainz (J. Gutenberg-U.)	Z22, Siemens 2002
	München T. H.	PERM
	München Max Planck I.	G2, G3
ス	ETH (Zürich)	ERMETH
伊	U. di Roma	Ferranti Mk1
	Columbia U.	IBM 650, 709, 7090
米	Social Security	IBM 705(3 sets)
	Nat. Bur. Stds.	SEAC, 704, PILOT, AMOS IV
	MIT	IBM 709
	U. of North Carolina	UNIVAC 1105
	Stanford U.	Burroughs 220

\$\$\$ TIME ON 0305

```

S440 JOB 08/14/61, 30MIN MR MORIGUTI X2208 1
  BURROUGHS ALGEBRAIC COMPILER-STANFORD VERSION 8/8/61$
  COMMENT QUARTZ PHASE1 PREPARATION$ 3
  K=2.72423$COMMENT TEST$ 4T
  PI=3.1415927$HFPI=PI/2$ 5
  A1=8/PI*2$A2=3A1(1-A1)$ 6C
  A6=2.0/3$ A7=2.0/5$ A8=2.0/7$ 7
  A3=A1-A6$ A4=A1-A7$ A5=9.0A1/5-A8$ 8
  B1=(A6.A8-A7*2)-(A1.A8-A2.A7)+(A1.A7-A2.A6)$ 9
  B2=(A5.A6-A4.A7)+A1(A4-A5)+(A3.A8-A4.A7)-A2(A3-A4)$ 10
  B3=A3.A5-A4*2$ 11
  B4=(A1.A8-A2.A7)-(A8-A2*2)+(A7-A1.A5)$ 12
  B5=(A4-A5)+(A1.A5-A2.A4)$ 13
  B6=(A1.A7-A2.A6)-(A7-A1.A2)+(A6-A1*2)$ 14
  B7=(A3-A4)+(A1.A4-A2.A3)$ 15
  B8=(A6.A8-A7*2)-A1(A1.A8-A2.A7)+A2(A1.A7-A2.A6)$ 16
  B9=(A5.A6-A4.A7)-A1(A1.A5-A2.A4)+(A3.A8-A4.A7)+A2(A1.A4-A2.A3)$ 17
  B10=A3.A5-A4*2$ 18
  C1=A1(B1-B4+B6)$ C2=A1(B2-B5+B7)$ C3=A1.B3$ 19
  C4=-B8$ C5=C1-B9$ C6=C2-B10$ C7=C-C3$ 20
  D0=C4$ D1=C5-C4$ D2=C6-C5$ D3=C7-C6$ D4=-C7$ 21
  E1=C4+K.B8$ E2=C5+K.B9$ E3=C6+K.B10$ E4=C7$ 22
  WRITE($$COEF,F101)$ OUTPUT COEF(D0,D1,D2,D3,D4,E1,E2,E3,E4)$ 23
  FORMAT F101(5F14.8,* =D0-4*,W0,4F14.8,* =E1-4*,W0)$ 24
  COMMENT PHASE2$ 25
  INTEGER 1$ 26
  ARRAY UV(4),Y(4),AA(4),BB(4),CC(4),EE(4),CHECK(4),FF(4),GG(4),ROOTYR(4) 27
  $ 28
  FOR ROOTX=(1.035,0.005,1.045)$ COMMENT TEST$ 29T
  BEGIN X=ROOTX*2$ 30
    WRITE($$XRESULT,F201)$ OUTPUT XRESULT(ROOTX,X)$ 31
    FORMAT F201(*ROOTX=*,X6.4,B5,*X=*,X10.8,W3)$ 32
    G1=D1+E1.X$ G2=D2+E2.X$ G3=D3+E3.X$ G4=D4+E4.X$ 33
  BIQ..U=0$ 34
  LOOP1..F1=G1+D0.U$ F2=G2+F1.U$ F3=G3+F2.U$ F4=G4+F3.U$ 35
    FP1=F1+D0.U$ FP2=FP1.U$ FP3=F3+FP2.U$ 36
    Q=F4/FP3$ 37
    IF ABS(Q) LEQ 1*-6$ GO TO CUBIC$ 38
    U=U-Q$ GO TO LOOP1$ 39
  CUBIC..U1=U$ U=0$ 40
  LOOP2..H1=F1+D0.U$ H2=F2+H1.U$ H3=F3+H2.U$ 41
    HP1=H1+D0.U$ HP2=H2+HP1.U$ 42
    Q=H3/HP2$ 43
    IF ABS(Q) LEQ 1*-6$ GO TO QUAD$ 44
    U=U-Q$ GO TO LOOP2$ 45
  QUAD..U2=U$ 46
    ROOT=SQRT(H1*2-4.D0.H2)$ 47
    U3=(-H1-ROOT)/2D0$ U4=(-H1+ROOT)/2D0$ 48

```

```

ORDER 2. U=U2$ 49
  EITHER IF U GEQ U3$ GO TO ORDER1$
  OR IF U GEQ U4$ BEGIN U2=U3$ U3=U END$ 51
  OTHERWISE$BEGIN U2=U3$ U3=U4$ U4=U END$ 52
ORDER1. IF U1 GEQ U2$ GO TO P1$ 53
  UV(1)=U2$ IF U1 GEQ U3$ BEGIN UV(2)=U1$ GO TO P3 END$ 54
  UV(2)=U3$ IF U1 GEQ U4$ BEGIN UV(3)=U1$ GO TO P4 END$ 55
  UV(3)=U4$ UV(4)=U1$ GO TO PHASE3$ 56
  P1. UV(1)=U1$ UV(2)=U2$ P3. UV(3)=3$ P4. UV(4)=U4$ 57
PHASE3. FOR I=(1, 1, 4)$ 58
  BEGIN Y(I)=(UV(I)+X-1)/K$ 59
  AA(I)=B1, UV(I)*2+B2. UV(I)+B3$ 60
  BB(I)=- (B4. UV(I)*2+B5. UV(I))/AA(I)$ 61
  CC(I)=(B6. UV(I)*2+B7. UV(I))/AA(I)$ 62
  EE(I)=- (B8. UV(I)*3+B9. UV(I)*2+B10. UV(I))/Y(I). AA(I)$ 63
  CHECK(I)=A1(1+BB(I)+CC(I))+(Y(I)-X)EE(I)$ 64
END$ IF Y(1) LEQ 0$ STOP 1$ 65
  IF Y(2) EQL 0$ STOP 2$ 66
  IF Y(3) GEQ 0$ STOP 3$ 67
  IF Y(4) GEQ 0$ STOP 4$ 68
BC12=BB(1)CC(2)-BB(2)CC(1)$ BC13=BB(1)CC(3)-BB(3)CC(1)$ 69
BC14=BB(1)CC(4)-BB(4)CC(1)$ BC23=BB(2)CC(3)-BB(3)CC(2)$ 70
BC24=BB(2)CC(4)-BB(4)CC(2)$ BC34=BB(3)CC(4)-BB(4)CC(3)$ 71
GG(1)=BC34-BC24+BC23$ 72
GG(2)=BC34-BC14+BC13$ 73
GG(3)=-BC24+BC14-BC12$ 74
GG(4)=BC23-BC13+BC12$ 75
ROOTYR(1)=1. 0$ 76
FOR I=(2, 1, 4)$ ROOTYR(I)=SQRT(ABS(Y(I))/Y(1))$ 77
FOR I=(1, 1, 4)$ FF(I)=EE(I). GG(I)/ROOTYR(I)$ 78
WRITE($$YRST, F301)$ OUTPUT YRST(FOR I=(1, 1, 4)$ Y(I))$ 79
FORMAT F301(4F20. 8, *=Y*, W0)$ 80
WRITE($$BRST, F302)$ OUTPUT BRST(FOR I=(1, 1, 4)$ BB(I))$ 81
FORMAT F302(4F20. 8, *=B/A*, W0)$ 82
WRITE($$CRST, F303)$ OUTPUT CRST(FOR I=(1, 1, 4)$ CC(I))$ 83
FOPMAT F303(4F20. 8, *=C/A*, W0)$ 84
WRITE($$ERST, F304)$ OUTPUT ERST(FOR I=(1, 1, 4)$ EE(I))$ 85
FORMAT F304(4F20. 8, *=E/A*, W0)$ 86
WRITE($$CHRST, F305)$ OUTPUT CHRST(FOR I=(1, 1, 4)$ CHECK(I))$ 87
FORMAT F305(4F20. 8, *=CHECK*, W0)$ 88
WRITE($$RTYRST, F306)$ OUTPUT RTYRST(FOR I=(1, 1, 4)$ ROOTYR(I))$ 89
FORMAT F306(4F20. 8, *=ROOT(ABS(Y)/Y1)*, W0)$ 90
WRITE($$GRST, F307)$ OUTPUT GRST(FOR I=(1, 1, 4)$ GG(I))$ 91
FORMAT F307(4F20. 8, *=G*, W0)$ 92
COMMENT PHASE4$ 93
  INTEGER M, NIT$ 94
  ARRAY(NURTYR(4), AR(4))$ 95
  FOR M=(30, 2, 60)$ COMMENT TEST$ 96T
  BEGIN NU=M$ NIT=0$ 97

```

```

LOOP3. FOR I=(2, 1, 4)$ NURTYR(I)=NU.ROOTYR(I)$           98
  EITHER IF(Y(2) GTR 0)$DEN1=FF(2)/TAN(HFPI.NUTYR(2))$     99
  OTHERWISE$ DEN1=-FF(2)/TANH(HFPI.NURTYR(2))$           100
  DEN2=-FF(3)/TANH(HFPI.NURTYR(3))$                       101
  DEN3=-FF(4)/TANH(HFPI.NURTYR(4))$                       102
  NUPRIME=M+ARCTAN(FF(1)/(DEN1+DEN2+DEN3))/HFPI$         103
  NIT=NIT+1$                                               104
  IF ABS(NU+NUPRIME) LSS 1* *-5$ GO TO CONV$             105
  IF NIT GEQ 100$ GO TO OSC$                               106
  NU=NUPRIME$ GO TO LOOP3$                                 107
OSC. WRITE($$NUOSC, F401)$ OUTPUT NUOSC(NU, NUPRIME)$     108
  FORMAT F401(*NU=*,X9.6,B5,*NUPRIME=*,X9.6,*AFTER 100 ITERATIONS
*, W0)$                                                    110
  GO TO CONTINUE$                                         111
CONV.. NURTYR(1)=NU/SQRT(Y(1))$                            112
  WRITE($$NUCONV, F402)$ OUTPUT NUCONV(NU, NIT)$         113
  FORMAT F402(*NU=*,X9.6,B3,*AFTER*, I4,*ITERATIONS*, W0)$ 114
  WRITE($$NULIST, F403)$OUTPUT NULIST(FOR I=(1, 1, 4)$NURTYR(I))$ 115
  FORMAT F403(4F20.8,* =NU ROOT(YI/Y1)*, W0)$           116
  AR(1)=1.0$                                               117
  INTEGER II$ II=3$                                        118
  EITHER IF Y(2) GTR 0$                                     119
    AR(2)=-GG(2).SIN(HFPI.NU)/GG(1).POOTYR(2).SIN(HFPI.NURTYR(2))$ 120
  OTHERWISE$ II=2$                                         121
  FOR I=(II, 1, 4)$ AR(I)=GG(I).SIN(HFPI.NU)/GG(1).ROOTYR(I).SINH(HFP
I.NURTYR(I))$                                             122
  WRITE($$ARST, F404)$ OUTPUT ARST(FOR I=(1, 1, 4)$ AR(I))$ 124
  FORMAT F404(4F20.8,* =AI/A1*, W0)$                       125
  CONTINUE.                                               126
  END                                                       127
  END$ BAD CODING FINISH$                                  128
COMPILED PROGRAM ENDS AT 1529
PROGRAM VARIABLES BEGIN AT 8857

```

```

-.47960000,-06-.33066716,-04-.15564532,-03.41243866,-03-.22324703,-03=DO-4
.82694070,-06.68639550,-04.56111579,-03,22324703,-03=E1-4

```

```

ROOTX=1.0350      X=1.07122500
.16337587, 01 .20393328,-01 -.23430997, 02 -.27491715, 01=Y
-.14827023, 01 -.45384334,-01 -.15568377, 01 -.15222238, 01=B/A
.74823790, 00 .89049830,-02 .56842873, 00 .50986884, 00=C/A
-.38261691, 00 .74322079, 00 .38344517,-03 -.26213520,-02=E/A
.14000000,-06 .48000000,-06 .51800000,-07 .19000000,-07=CHFCK
.10000000, 01 .11172493, 00 .37870532, 01 .12972000, 01=ROOT(ABS(Y)/Y1)
NU=30.312389 AFTER 6 ITERATIONS
.23715168, 02 .33866495, 01 .11479462, 03 .39321231, 02=NU ROOT(YI/X1)
RESULT OUT OF RANGE IN SINH

```

```

.10000000, 01 -.78486758, 00 .17522279, 00 .00000000, 00=AI/A1
NU=32.171805 AFTER 5 ITERATIONS
.25169899, 02 .35943926, 01 .12183633, 03 .41733265, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 .61293901, 00 .99132646, -01 .00000000, 00=AI/A1
NU=34.074417 AFTER 4 ITERATIONS
.26658424, 02 .38069618, 01 .12904162, 03 .44201333, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 -.53418984, 00 .43365787, -01 .00000000, 00=AI/A1
NU=35.992013 AFTER 4 ITERATIONS
.28158672, 02 .40212051, 01 .13630366, 03 .46688839, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 .51527585, 00 .46653258, -02 .00000000, 00=AI/A1
NU=37.907514 AFTER 5 ITERATIONS
.29657282, 02 .42352143, 01 .14355777, 03 .49173627, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 -.54827145, 00 .53829051, -01 .00000000, 00=AI/A1
NU=39.802305 AFTER 5 ITERATIONS
.31139688, 02 .44469097, 01 .15073344, 03 .51631550, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 .64719096, 00 .11362142, 00 .00000000, 00=AI/A1
NU=41.642298 AFTER 6 ITERATIONS
.32579223, 02 .46524828, 01 .15770159, 03 .54018388, 02=NU POOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 -.85256658, 00 .19810315, 00 .00000000, 00=AI/A1
NU=43.360982 AFTER 9 ITERATIONS
.33923851, 02 .48445026, 01 .16421034, 03 .56247865, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 .11889233, 01 .31364022, 00 .00000000, 00=AI/A1
NU=46.538359 AFTER 8 ITERATIONS
.36409700, 02 .51994949, 01 .17624324, 03 .60369559, 02=ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 .10759699, 01 .27828013, 00 .00000000, 00=AI/A1
NU=48.303701 AFTER 6 ITERATIONS
.37790831, 02 .53967276, 01 .18292868, 03 .62659560, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 -.77335181, 00 .17073081, 00 .00000000, 00=AI/A1
NU=50.166291 AFTER 5 ITERATIONS
.39248045, 02 .56048253, 01 .18998241, 03 .65075712, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 .60725100, 00 .96024903, -01 .00000000, 00=AI/A1
NU=52.070150 AFTER 4 ITERATIONS
.40737546, 02 .58175338, 01 .19719242, 03 .67545398, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 -.53199226, 00 .40889258, -01 .00000000, 00=AI/A1
NU=53.988013 AFTER 4 ITERATIONS
.42238004, 02 .60318069, 01 .20445547, 03 .70033250, 02=NU ROOT(YI/Y1)
RESULT OUT OF RANGE IN SINH
.10000000, 01 .51563476, 00 .70010881, -02 .00000000, 00=AI/A1

```

```

NU=55.902987  AFTER  5 ITERATIONS
.43736201, 02 .62457573, 01 .21170758, 03 .72517354, 02=NU ROOT (YI/Y1)
                RESULT OUT OF RANGE IN SINH
.10000000, 01 -.55139945, 00 .56443734, -01 .00000000, 00=AI/A1
NU=57.796078  AFTER  5 ITERATIONS
.45217277, 02 .64572627, 01 .21887682, 03 .74973072, 02=NU ROOT (YI/Y1)
                RESULT OUT OF RANGE IN SINH
.10000000, 01 .65438968, 00 .11707904, 00 .00000000, 00=AI/A1
NU=59.631932  AFTER  7 ITERATIONS
.46653574, 02 .66623734, 01 .22582929, 03 .77354542, 02=NU ROOT (YI/Y1)
                RESULT OUT OF RANGE IN SINH
.10000000, 01 -.86643061, 00 .20320016, 00 .00000000, 00=AI/A1

```

〔以下略〕

▶ ニュース ◀

1961年10月31日午後、日仏工業技術会・日本OR学会・情報処理学会・自動制御研究会の共催でGrenoble工業大のArnold Kaufmann教授の「大型電子計算機による経営のシミュレーション」と題する講演が行われた。この講演内容を、当日の録音テープに基き、電力中央研究所山岡春夫氏が抄録した日仏工業技術会会報1961年12月号の記事の概要を同会の承認を得てお知らせする。

スマール人はすでに6,000年前にその灌漑用水系の研究にシミュレーションを利用したといわれているがとにかく、シミュレーションとは実験的研究方法の1つであって、われわれがその特性を知っているある過程をくり返し再現することだといえる。その目的、性格、速さなどによって分類できるが、多くの場合は確率的要素を含むので、信頼限界と結びついて試行回数を考えるという基本的問題があり、これはフランスでも注意されている。

フランスでもシミュレーションによる研究は広く行われ計算機としてはIBMの650, 705等のほかBullのGamma E. T. や60などが使われており、大型機が約150台、中型機が約1,000台ある。(このあとシミュレーションの使われている分野の説明があったが、配置および割当て問題、在庫問題、待ち行列問題、ビジネス・ゲーム等、その対象となったものはわが国でもポピュラーなものが多かった)

数学的側面としては上述のサンプル・サイズの問題、convergenceの問題(これはergodicという観点から研究している人がフランスに多くいる)、time horizonの後のcontinuationの問題(5年計画としては最適でも6年目に悪くなってしまうことなどの起らないように)、DPなどが主に数学者の注目をひいている。また数論のような純粋数学においても、その仮定をチェックするためなどに用いられる。

シミュレーションは解析の方法との相互共生において発展せねばならない。解析の方法でとける問題をシミュレーションで解く例がままあるが、シミュレーションは、一般的に、補助的な実験的方法にすぎないことに注意し、問題の構造を解析の方法で把握すべきであることを強調したい。