

I. C. I. P*. 論文紹介

35. 丸めの誤差の推定

Ch. Blanc: Sur L'estimation des erreurs d'arrondi [ICIP/A.L.1 (1959)]

電子計算機を用いる複雑な計算には丸めの誤差がいつも伴う。丸めの誤差の問題には例えば次のようなものがある。

a) 与えられた問題について、計算法が決まっている場合の最終結果にあらわれてくる丸めにもとづく誤差の評価。

b) 与えられた問題に対していろいろな計算法を丸めの誤差の立場から比較すること。

c) ある与えられた形式の問題(例えば逆行列の計算)について、そのままの形式で計算機にかけることが適当かどうかを丸めの誤差の見地から判断すること。これらの問題を分析するために次の各項について検討している。

1. 丸めの基本的な誤差
2. 丸めの誤差と情報
3. 丸めの誤差をその積率により調べる
4. 線形形式の分解に伴う丸めの誤差

そしてさらに丸めの誤差に関する基本的な仮定の妥当性や、問題を簡単にする可能性についても述べている。

(伊藤文子)

36. 行列演算での丸めの誤差

J.H. Wilkinson: Rounding Errors in Algebraic Process [ICIP/A.1.8 (1959)]

連立一次方程式を解く最も初等的なガウスの消去法とは次のようなものである。

$$A^{(1)} \vec{x} = \vec{b}^{(1)} \dots \dots \dots (1)$$

より出発して対角線下の要素を逐次0とすることにより

$$(A^{(n)} \vec{b}^{(n)}) = \begin{pmatrix} a & \vdots & b \\ 0 & \ddots & \vdots \end{pmatrix}$$

とし、これより下の x_n から逐次 x_1 までを定める。

計算途中の四捨五入による誤差を考える場合、次の

二つがある。

(i) 実際に得られた解が正確な解とどのくらい違うか。

(ii) 実際に得られた解は最初の式とどのくらい違う式を解いたことになるか。

この論文が議論するのは(ii)の方である。ガウスの消去法については、各段階で小数点以下 t 桁まで算出すればその解は

$$A\vec{x} = \vec{b}$$

に対して

$$(A+E)\vec{x} = \vec{b} + \delta\vec{b}$$

を解いたことになる。ここに $E, \delta\vec{b}$ は次の行列の各要素でおさえられる。

$$\frac{1}{2}10^{-t} \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & \dots & 1 \\ 0 & 1 & 2 & \dots & 2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 2 & \dots & n-1 \end{pmatrix} \quad \frac{1}{2}10^{-t} \begin{pmatrix} 1 \\ 3 \\ \vdots \\ 2n-1 \end{pmatrix}$$

これに対して Gauss-Banachiewicz の方法を少し改良して、精度のよい解法が示されている。連立一次方程式(1)を

$$LA^{(n)} = A^{(1)} (\equiv A) \quad L = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ m_{ij} & & 1 \end{pmatrix}$$

なる二つの三角行列の積に分けて $A^{(n)}$ を求め、ガウスの方法を使うと解のうち最大のものを小数点より、上に k 桁の数として

$$(A+E)\vec{x} = \vec{b} + \delta\vec{b}$$

$$|E_{ij}| < \frac{1}{2}10^{-t}$$

$$|\delta b_r| < \frac{1}{2}10^{-t} + r \frac{1}{2}10^{k-t}$$

となる。原論文にはこれらの実際上のやり方および数値例が示してある。また、この方法を応用して、行列の固有ベクトルを求めるときの誤差の評価も行っている。

(杉浦成昭)

37. 行列の特性多項式の正確な求め方

D.B. Gillies: The Exact Determination of the Characteristic Polynomial of a Matrix [ICIP/A.1.10 (1959)]

* International Conference on Information Processing (1959年6月パリで第1回が開催された)

有理数を要素にもつ行列の固有値，特性多項式を求める方法を考える．各要素の分母の最小公倍数を行列に掛けることにより，整数を要素にもつ行列について考えれば充分である． A の特性多項式 $p(\lambda) = |A - \lambda I|$ を $p(\lambda) = (-1)^n(\lambda^n + c_1\lambda^{n-1} + \dots + c_n)$ によって計算しては手数が大変なので，従来は $p(\lambda) = 0$ の重根を単根に直した多項式 $p_2(\lambda)$ について近似的に根を求め，それより特性多項式の係数を定めていた．これでは固有値のわずかな誤差が，特性多項式の係数に大きく影響することがあるので，正確な特性多項式を求める方法が二つ発表されている．

(1) 行列式の計算により $p(\lambda)$ を求める方法．

$A_k = A - kI$ ($k=0, 1, \dots, n$) とし，ある列の定数倍を他の列に加えても，行列式は不変であること．

$$\begin{vmatrix} 0 & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{j-1,2} & \dots & a_{j-1,n} \\ a_{j1} & a_{j2} & \dots & a_{jn} \\ 0 & a_{j+1,2} & \dots & a_{j+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & \dots & a_{nn} \end{vmatrix} = (-1)^{j+1} a_{j1} \begin{vmatrix} a_{12} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n2} & \dots & a_{nn} \end{vmatrix}$$

であることの性質を使って $|A_k| = p(k)$ ($k=0, 1, \dots, n$) を求める．補間法により容易にこれから $p(\lambda)$ が求まる．

(2) 随伴行列により $p(\lambda)$ を求める方法．

行列 A に適当な初等変換 U_i ($i=1, 2, \dots, m$) を行って

$$\bar{A} = U_m^{-1} U_{m-1}^{-1} \dots U_1^{-1} A U_1 \dots U_{m-1} U_m$$

$$\bar{A} = \begin{pmatrix} C_1 & D^1 \\ 0 & A^1 \end{pmatrix} \quad C_1 = \begin{pmatrix} 0 & 0 & \dots & 0 & -C_k \\ 1 & 0 & \dots & 0 & -C_{k-1} \\ 0 & 1 & \dots & & \\ \vdots & \vdots & \ddots & & \\ & & & 0 & -C_2 \\ 0 & 0 & \dots & 1 & -C_1 \end{pmatrix}$$

とできる． C_1 が随伴行列である．ここに U_i は次の三つの行列のいずれかである．

$$\begin{vmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \ddots & & & & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{vmatrix} \quad \begin{vmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \ddots & & & & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{vmatrix}$$

$$\begin{vmatrix} 1 & & & & & & & & \\ & \frac{1}{\alpha} & & & & & & & \\ & & \frac{1}{\alpha} & & & & & & \\ & & & \ddots & & & & & \\ & & & & \frac{1}{\alpha} & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{vmatrix}$$

C_1 の特性多項式は $\lambda^k + C_1\lambda^{k-1} + \dots + C_k = p_1(\lambda)$ であるから

$$p(\lambda) = p_1(\lambda) |A^1 - \lambda I|$$

A^1 についても同様な変形を行えば結局

$$p(\lambda) = p_1(\lambda) p_2(\lambda) \dots p_s(\lambda)$$

とできて $p(\lambda)$ が求まる． (杉浦成昭)

38. 共通語の提案

J. Poyen, et B. Vauquois: A Propos D'un Langage Universel [ICIP/C.3.2 (1959)]

電子計算機の性能は急速に高度化されつつある．すなわち，演算速度は極めて速くなり，記憶容量は龐大なものとなってきている．したがって非常に複雑な問題でも実際に取扱いうる可能性が出てきた．しかしそれに伴ってプログラミングはますますこみ入ってきており，そのために新しい大形計算機の便利さ，有用性はそこなわれることが多い．

そこで新しい計算機を作ることに払われる努力と共に，人間と機械の間の“通信”を容易にすることを目的とする“super-programmation”に努力しなければならない．つまりプログラム用語を人間の言葉に近づける必要がある．このような見地からこの論文では計算機に使用する総合的用語の統一について問題を分析し検討した結果，次のような考え方にもとづく用語の提案をしている．

(1) 数式を表現するには Zürich の会議で定められた数式系に Σ (和) と Π (積) の定義を附加し，さらに時間的な指数の概念を補って“周期”を表現できるようにすること．

(2) 数式系以外に必要な最小限の言語として“計算”，“条件”など 34 語を用いること．

(伊藤文子)

39. 特に科学計算のための共通語について

F.L. Baner and K. Samelson: The problem of a common language, especially for scientific numerical work (motives, aims, restrictions and

results of the zürich conference on ALGOL) [ICIP/C.3.3 (1959)]

計算機の言葉として、わかりやすく、しかも適確な共通語の必要性が痛感されている。

科学計算を中心とした共通語が1958年5月、ACM特別委員会とGAMMの代表者によるチューリッヒにおける会議で作られ、算法言語であることから、ALGOLと命名された。この会議では、問題を実際のもの（科学計算）に限り、

- (1) 数学記号を基として説明なしでもわかる。
- (2) 計算の過程を表現できる。
- (3) 機械的に機械語に翻訳できる。

の三つの目的を中心として、ALGOLを作り出した。

共通語である以上、世界の各分野で、広く受け入れられることを希望している。

またソビエトにおけるグループも、ALGOLと同様な目的で、自動プログラミングを計画しているので、でき上るものは、ALGOLと非常に似たものとなるであろう。

(喜多実子)

40. 対数計画法とその特殊線形 凸計画法への応用

G.R. Parisot: Les Programmes Logarithmiques Application aux Calculs des Programmes Convexes Specialement Lineaires [ICIP/D.1.5 (1959)]

この論文は非常に多次元の線形計画法の計算法について述べている。しかも、この計算法の原理は非線形の場合にも適用できる。これはよく知られているFrischの対数的ポテンシャル法の一般化である。

$$\text{最初 } \sum_k a_{jk} x_k + x_j = p_j$$

なる形の制約式が与えられているとき $\sum_k C_k x_k$ の極値を与えるような x_k の値を求めるという問題を、この新しい方法では

$$\sum_k C_k x_k + \sum_k p_k \log x_k + \sum_j p_j \log (b_j - \sum_k a_{jk} x_k)$$

の極値を与える x_k を求めるという問題に変えて解くのである。この方法には次のような利点が考えられる。

- (1) プログラムを、極値からとは限らず任意の点から始めることができる。
- (2) 計算の過程で行列を不変な形で記録しておいてよい。
- (3) 十分な近似に到達できたときにいつでも計算を止めることができる。

このような可能性があることは特に電子計算機を使つての計算には便利である。

(伊藤文子)

41. ハーバード計算研究所における 自動翻訳の研究

V.E. Giuliano and A.G. Oettinger: Research on Automatic Translation at the Havard Computation Laboratory [ICIP/F.4.1 (1959)]

この論文はハーバードの露英自動翻訳辞書の操作とそれを用いて得られた結果、その応用、および現在行われているより進んだ自動翻訳用の装置について述べている。自動翻訳辞書とは逐語訳ができるような機構であり、自動翻訳には必ず使用されるはずのものである。この辞書の操作プログラムはハーバードにあるUnivac-I計算機用に作られている。辞書用の記憶装置としては、この計算機の約1000語の高速度記憶装置ではとうてい間に合わないので、磁気テープを附加して用いた。このために操作上のいくつかの重要な問題が生じてくる。この対策について述べた後、操作プログラムをブロックダイアグラムで示し、各段階のルーチンの説明をしている。

自動翻訳辞書によって逐語訳ができた後、その編集を人間が行う。これによって文章としての形態が整えられる。物理学の論文についてこれを実際に行った例がのせられている。このような実験を数十の論文について行った結果、初歩のロシア語の知識をもつある分野の専門家ならば、このような逐語訳されたその分野の論文を容易に理解し、翻訳できることがわかった。またロシア語を全然知らない専門家でも、その論文内容に十分に興味を持っていれば編集し理解することは可能であるといっている。またこの辞書の操作プログラムを基本として現在進められている自動翻訳の研究について述べ、実験用プログラムをブロックダイアグラムで示している。

(伊藤文子)

42. 機械翻訳のための COMIT システム

V.H. Yngre: The Comet System for Mechanical Translation [ICIP/F.4.3 (1959)]

言語翻訳のためのプログラミングは非常に複雑で、それを作るのには大変な時間がかかる。これを容易にするためにプログラミングに馴れていない言語学者が使えるようなIBM 704用の“プログラム語”を作った。これを使って書かれた翻訳用プログラムは

IBM 704 のコンパイラと解釈ルーチンにより普通のプログラムに変換される。このようなプログラム語ができたために、言語学者が専門のプログラマーの協力がなくても、独力で機械翻訳の問題に取り組むことができるようになった。

このプログラムを作るにあたって目標としたのは有用性 (utility), 使いやすさ (convenience), 簡単さ (simplicity) の三つである。有用性については万能計算機としての能力が十分生かせること、つまり今後予想されるどんな操作にも適用できるようにすること、また使い易さという点では言語学者が容易に使いこなし得ることを第一とし、そのために変換された後のプログラムによる操作速度とか、記憶装置の有効な使い方とかはある程度犠牲にした。また覚えやすいというためには簡単にしなければならない。そのためには特殊な操作をなるべく少なくし、いくつかの基本的な操作の組合せでプログラムが作られるようにした。

この論文ではこれらのことを言語翻訳の実例について説明しているが、このプログラム語は非数値記号の操作を含むプログラミング例えば論理数学、定理の証明、機械による学習などのプログラムにも応用できると述べている。(伊藤文子)

43. 機械による文法組立と構文分析のプログラム

Kanneth E. Harper and David G. Harp: The Use of Machines in the Construction of a Grammar and Computer programme for structural analysis [ICIP/F.4.4 (1959)]

Rand 社がロシア語から英語への機械翻訳に用いた方法論の解説である。従来多く試みられた原語の既成文法を、そのままプログラムに採入れる方法が行詰まるのは、文法そのものの不完全性が機械翻訳には適せぬためと考え、構文を決定することが機械翻訳における文法上、意味上の問題処理に役立つとの立場に立って、原語の構文分析の観点からその文法を組立てて之を機械翻訳用の構文決定プログラムに使う方法をとっている。

資料 text (延べ約 25 万語) の処理 (原文標本作成・使用語の逐語的語彙作成、試験翻訳・語彙と翻訳プログラム改良のための分析) を通じて、構文をあたかも 1 本の木と枝葉のように、一語 (若しくは節) を中心に各語はそれぞれ列の一語と統語法上並びに意味

上の支配従属関係にあるものとしてとらえ、各品詞の結合関係、構文上の繋りの約束・配語順序はそれぞれ従属関係にある対形として表にする。

text 原語の翻訳には翻訳語のシンタックスを正確にコード化するという風に、龐大な資料を経済的、循環的に機械操作して、個々の場合を統一化、標準化していわばその資料の範囲内での完全な文法・語解辞典を機械化する訳である。

本研究は物理学文献に限られ、その資料は延 25 万語であるが、少なくともその応用範囲内における翻訳としては完全を期し得た。構文決定プログラムそのものは簡単であるから、この方法論は他国語間の機械翻訳に応用して有益と考えられる。(伊藤文子)

44. 英和機械翻訳

S. Takahashi, H. Wada, R. Tadenuma, S. Watanabe: English-Japanese Machine Translation [ICIP/F.4.5]

日本語は外国語と文字および構造が全く異なるので、一般の日本人は外国語を学ぶのに非常に苦勞をしている。機械翻訳の必要性は、外国より日本において非常に大きいのであるが、同時に機械翻訳のプログラミングがむずかしいという事がいえる。機械翻訳を行うには、まず記憶容量が問題となるが、日本には十分な記憶容量を持った計算機がないので、翻訳用の計算機を作り、それに“やまと”となづけた。やまとは、十分な記憶容量の磁気ドラムを持ち、言葉を扱うのに便利な可変な長さの語を記憶することができ、その他、翻訳に便利な命令がある。

機械翻訳は、中学の英語の教科書を対象とし、約 2000 語の英単語と、一年生の英文を訳すプログラムを用意した。

翻訳の原理は大体、次のようなものである。

(1) 英文 1 文を読み込み、それを 1 語ずつ翻訳に必要な情報、品詞、日本語の位置、別訳の位置などが書き込まれている機械語におきかえる。

(2) 修飾語を取除いて、基本文形にする。

(3) 日本語の順序に直し、助詞 (主語には“ガ”直接目的語には“ヲ”, 間接目的語には“ニ”) を挿入する。

(4) 修飾語を日本語の順序に直し、しかるべき位置に附ける。

(5) 活用するものは活用させて、日本語を出す。

この原理に従った、翻訳プログラムのフローチャー

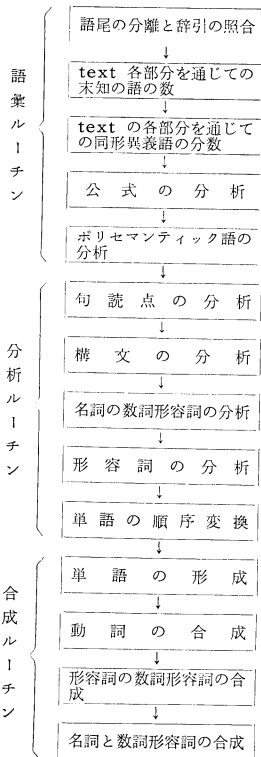
トおよび辞書，表，および機械の構造と命令について詳述している。

(喜多実子)

45. 機械翻訳の方法と英・露翻訳への応用

I.K. Belskya: Machine Translation Method and their Application to Anglo-Russian Scheme [ICIP/F.4.7 (1959)]

この論文は英語からロシア語への機械翻訳の過程について述べたものである。この方法によれば，また英語から日本語，ロシア語，中国語，ドイツ語への翻訳にも応用できる。本研究過程を完成するために 3000 の文章が含まれている 100 個の数学の英文例題を選び出し MT 辞書と MT ルーチンによって翻訳した。MT 辞書は英単語および熟語とも 2000 語と，それに対応するそれ以上のロシア語が収められている。MT ルーチンすなわち翻訳ルーチンは語彙ルーチン，分析ルーチン，合成ルーチンの三つに分けられ，それぞれのルーチンはまた右図のような操作過程に分けられる。各段階における処理について例をあげて詳しく説明されている。



しかし英語とロシア語の構成は非常によく似ているので，英語の構文変更は意味がわかる程度の最小限にとどめたと述べている。

(伊藤文子)

46. 数学的でないデータの処理方法と辞典への応用

A.F. Parker Phodes and R.M. Needham: A reduction method for non-arithmetic data and its application to thesauric translation [ICIP/J.5.6 (1959)]

機械翻訳を実用化するに当たって，根本的な問題はいくつかあるが，われわれが言葉の意味と呼んでいるものを如何に表現するかという問題がある。

良い翻訳というのは単語対単語のおきかえでなく，文章全体の文脈の中で，その単語の意味を表現することである。

そのためには，単語の意味の集合を考え，ある演算をほどこせば，文脈により，種々の意味を生ずる方式が必要とされる。その最も簡単なものは表にすることであるが，言葉の意味は，その場その場で作り出されるものであるから，完全な表にするということは不可能である。そこで東論を応用して

(1) 単語の意味の辞典は，距離空間であって，最も近い意味を持つ単語を探すことができるように，配列されている。すなわち，意味の近いものは，近くに配列されている。

(2) ある文脈の中の単語の持っている意味がその単語自身とそれに近い単語の意味から計算されて出てくる。

上の二つの条件を充す空間および演算について述べている。

(喜多実子)