

文 献 紹 介

A: 数 値 解 析 B: プログラミング C: 計 算 機 方 式
D: 回 路 お よ び 機 器 E: オートマトン F: 応 用 そ の 他

A-11. ラゲール函数を用いるラプラス逆変換の数値解法

William T. Weeks: Numerical Inversion of Laplace Transforms Using Laguerre Functions [JACM, Vol. 13, No. 3, July, 1966, pp. 419~429]

与えられた解析函数 $g(p)$ に対して,

$$g(p) = \int_0^{\infty} e^{-pt} f(t) dt$$

となる未知函数 $f(t)$ を定めること, すなわち Laplace 逆変換の数値解を求めるのに, Laguerre 函数の一次結合で近似することは, Papoulis, Lanczos らによって提唱されている. 本論文では, Papoulis の方法を, 自動計算に適合するように精密化している.

函数 $f(t)$ は, $c \geq c_0$ に対して, $e^{-ct}|f(t)|$ およびその二乗が, 区間 $(0, \infty)$ で可積分であると仮定する. 区間 $(0, \infty)$ で直交系をなす Laguerre 函数

$$\Phi_n(x) = e^{-\frac{x}{T}} L_n(x), \quad n=0, 1, 2, \dots$$

($L_n(x)$ は n 次の Laguerre 多項式) で, $e^{-ct}f(t)$ を展開し, その第 N 部分和を $e^{-ct}f_N(t)$ とおく. すなわち, $T > 0$ をスケール因子として,

$$f_N(t) = e^{ct} \sum_{n=0}^N a_n \Phi_n\left(\frac{t}{T}\right),$$

$$a_n = \frac{1}{T} \int_0^{\infty} e^{-ct} f(t) \Phi_n\left(\frac{t}{T}\right) dt$$

である. $f_N(t)$ の Laplace 像を $g_N(p)$ とおけば, $g_N(c+i\omega)$ は, ω の函数として, $(-\infty, \infty)$ で, $g(c+i\omega)$ に, 二乗平均収束する. すなわち,

$$\begin{aligned} I_N &= \int_0^{\infty} e^{-2ct} |f(t) - f_N(t)|^2 dt \\ &= (2\pi)^{-1} \int_{-\infty}^{\infty} |g(c+i\omega) - g_N(c+i\omega)|^2 d\omega \end{aligned}$$

とおけば, $\lim_{N \rightarrow \infty} I_N = 0$ である.

$$g_N(p) = \sum_{n=0}^N a_n \left(p - c - \frac{1}{2T}\right)^n \cdot \left(p - c + \frac{1}{2T}\right)^{-n-1}$$

を, I_N に代入し, さらに, 積分変数を,

$$\omega = \frac{1}{2T} \cot \frac{\theta}{2}$$

で, 変換すれば,

$$\begin{aligned} I_N &= \frac{T}{2\pi} \int_{-\pi}^{\pi} \left| \left(\frac{1}{2T} + \frac{i}{2T} \cot \frac{\theta}{2} \right) g\left(c + \frac{i}{2T} \cot \frac{\theta}{2}\right) \right. \\ &\quad \left. - \sum_{n=0}^N a_n e^{in\theta} \right|^2 d\theta \end{aligned}$$

である. いま, $f(t)$ を実数値函数とすれば, a_n は全て実数だから, 上の等式は, $\sum_{n=0}^N a_n \cos n\theta$ が,

$$h(\theta) = \frac{1}{2T} \operatorname{Re} g(c+i\omega) - \omega \operatorname{Im} g(c+i\omega) \Big|_{\omega = \frac{1}{2T} \cot \frac{\theta}{2}}$$

に, $(-\pi, \pi)$ で, 平均収束すること
を示している. そこで, $h(\theta)$ を次の公式により, 三角多項式で補間近似する.

$$\theta_j = \left(\frac{2j+1}{N+1} \right) \cdot \frac{\pi}{2}, \quad j=0, 1, \dots, N;$$

$$a_0 = \frac{1}{N+1} \sum_{j=0}^N h(\theta_j);$$

$$a_n = \frac{2}{N+1} \sum_{j=0}^N h(\theta_j) \cos n\theta_j, \quad (n \neq 0).$$

この a_n を用いて, $f_N(t)$ を, 計算すれば, $f(t)$ の近似を得る.

計算にあたっては, $\omega_j = \frac{1}{2T} \cot \frac{\theta_j}{2}$, $\cos \theta_j$, $\sin \theta_j$, $\cos n\theta_j$ および $\Phi_n(x)$ が, 次の漸化式をみたすことに注意するとよい.

$$\omega_j = (\gamma \omega_{j-1} - \delta)(\omega_{j-1} + \gamma)^{-1},$$

$$\left(\gamma = \frac{1}{2T} \cot \theta_0, \quad \delta = \frac{1}{4T^2} \right);$$

$$\cos \theta_j = \alpha \cos \theta_{j-1} - \beta \sin \theta_{j-1},$$

$$\sin \theta_j = \beta \cos \theta_{j-1} + \alpha \sin \theta_{j-1},$$

$$(\alpha = \cos 2\theta_0, \quad \beta = \sin 2\theta_0);$$

$$\cos n\theta_j = 2 \cos \theta_j \cos (n-1)\theta_j - \cos (n-2)\theta_j;$$

$$\Phi_0(x) = e^{-x/2}, \quad \Phi_1(x) = (1-x)\Phi_0(x),$$

$$n\Phi_n(x) = (2n-1-x)\Phi_{n-1}(x) - (n-1)\Phi_{n-2}(x) \quad (n > 1).$$

残された問題は, T と c を如何に選ぶかである. これについては, 経験的な結果として, 次のようなことがいえる.

区間 $[0, t_{\max}]$ で, $f(t)$ の近似値を, 得たい場合には,

$$T = t_{\max}/N,$$

$$c = \begin{cases} c_0 + 1/t_{\max} : c_0 + 1/t_{\max} \geq 0 \\ 0 : c_0 + 1/t_{\max} < 0 \end{cases}$$

で十分である。ただし、 N は十分大きくとるものとする (実験例では 20~50)。

最後に、四つの実験例が報告されているが、これらは、Schmittroth, Norden, Papoulis らの結果と、ほぼ同程度の正確さを示している。 (牛島照夫)

A-12. 代数式操作の展望

J.E. Sammet: Survey of Formula Manipulation [CACM, Vol. 9, No. 8, Aug., 1966, pp. 555~569]

式を記号のまま計算する必要がしばしばあって、それは手間のかかる、退屈な、まちがえやすい、その反面において定型的な手順のくりかえしであるので、ある意味で計算機に向いた仕事である。これは計算機利用のきわめて初期、1953 年ごろからすでに試みられている。最近では LISP を用いて特定の問題を解いたり、ALGOL や FORTRAN の機能に追加をおこなって一般的な問題を解いたりしている。

いちばん試みが多いのは、記号微分と多項式とであるが、記号積分、有理関数なども含め、通常のプログラム言語の有する入出力、ループ、数値計算の能力を利用し包括的なシステムを組む方向にあり、その例は Formula ALGOL, ベル研究所の ALTAC, アイビーエムの FORMAC がある。

Formula ALGOL は機能の追加を最小とし、どんな複雑な問題でも解けるかわりに、あらゆる処理を自分で記述せねばならない。FORMAC はきまった操作が組込んであるのでその範囲内では実用的に使える。

冗長な、あるいはわかりにくい表現の式を単純化することは、印刷形式をととのえる、記憶容量を節約する、等値演算 ($1 \cdot a \rightarrow a$, $0 + a \rightarrow a$, $0 \cdot a \rightarrow 0$ など) の除去などでぜひ必要だが、いったい $a(b+c)$ と $ab+ac$ とどちらをより単純とするのか、一意の基準がないのが悩みである。

結果の出力は、普通の行印刷機では不十分で、2次元の表示装置や、写真植字機が著しい効果をあげる。計算の進行過程が明白に予測できないときや、途中で人間が介入するほうがずっとスマートにできることがあるので直結実時間方式の入出力がのぞましい。

数値計算においてはある数値にいかなる演算を加えても、単一の数値のままになっていることが多いが、記号計算では、演算のたびに占有語数がふえるので、記憶容量は格段に余計必要とされる。また記号式のなかでも数値計算はおこなわれるが、有理関数などでは整数の正確な値が要求され、しかもそれが階乗などでとんで

もなく大きい桁数になる。したがって普通の浮動小数点表示ではまったく無意味になることがある。

記号計算の報告されている応用例には、天体力学、空気力学、洪水、ルジャンドル関数、ベッセル関数、量子力学などがある。約 100 件の参考文献をあげる。

(西村恕彦)

B-13. SNOBOL 3 プログラミング言語

D.J. Farber, R.E. Griswold and I.P. Polonsky: The SNOBOL 3 Programming Language [BSTJ, Vol. 45, No. 6, July, Aug., 1966, pp. 895~944]

SNOBOL 3 は string manipulation 言語 SNOBOL の拡張であり、新たな機能のおもなものは、プログラマによる関数定義ができることである。関数定義は primitive function DEFINE を用いて次のように行なう。

関数の定義ルーチン (サブルーチン) をプログラムの適当な部分に書き、プログラムで用いられる関数の形式、定義ルーチンの entry point および定義ルーチンの中で用いられる local name (もしあれば) のリストをおのおの第 1, 第 2, および第 3 引数として DEFINE を実行する。このあとでは、第 1 引数と同じ形式の関数が参照されるたびに定義ルーチンが実行され、その値が関数の値となる。

例 DEFINE ("FACT(N)", "F")

SYSPOT=FACT ("2")+

FACT ("4")/(END)

F FACT=.EQ (N, "0") "1"/S(RETURN)

FACT=FACT (N-"1")*

N/(RETURN)

このプログラムが実行されると、2 および 4 の factorial の和を計算して値を印字する。ラベル F を持つステートメントとその次のステートメントが、local name を持たない FACT の定義ルーチンを構成している。

本論文は SNOBOL 3 言語の記述、システム (ベル研究所における、IBM 7094 による) のコントロールカード、エラーメッセージ、組み込みの primitive functions のリスト、および例プログラムを網羅している。

(二村良彦)

B-14. 実時間用のプログラミング言語に対する種々の要求

A. Opler: Requirement for Real-Time Language

ages [CACM, Vol. 9, No. 3, Mar., 1966, pp. 196~199]

実時間と呼ばれる計算機応用分野を明確に記述することはむずかしく、オンラインと重なる場合はさらに複雑となる。実時間の領域には次に示すような6個の計算の型がある。

- 1) 訓練用プログラムなどの実時間シミュレーション
- 2) 演算途中で実時間処理を行なうもの
- 3) アナログ電子計算機につながれたシステムの一部としてはたらくもの
- 4) 中央の計算機の周辺装置の一要素としてはたらくもの
- 5) 遠隔通信装置としてはたらくもの
- 6) 1個以上の電子計算機の演算を制御するもの

1), 2) は一部を除いて現在使われている言語で処理される。3) は特定の周辺装置をあつかう特別な言語が必要である。4) はインプットとアウトプットにかなりの変換と規格化が必要である。5) は同時に到着するメッセージ、使用されるコード間の変換、その他をうまく処理する言語が要求される。6) は割込みやI/Oの処理に関するハードウェアの制御その他が要求される。

現在考えられるいくつかの問題には、ハードウェア制御装置相互間の密接なつながりの問題、システムの信頼性の問題、処理時間についての問題、実時間プログラムと外部装置との関連性の問題、待行列および優先順位に対する問題などがある。

実時間の言語に必要な要素には、算術的変数、物理的な周囲との相互関係（入出力変換、応答時間など）、通信関係の要素、優先順位・待行列など競合する要求の処理、経過時間など時刻照合がある。

実時間に用いられる言語の開発が必要であり、それには多くの時間をついやし、作成されたとしても、検査がむずかしく、しばしば信頼性に欠ける。この言語を開発する二つの方法は次の如くである。

- 1) フォートランなどの現存する言語を拡張、修正する方法
- 2) 実時間に用いられる新しい特定の言語を開発する方法

結局、この分野において開発を進めるためには、言語の拡張・修正を行なうのではなく、解決されなければならないシステム問題をコンパイルし、処理を行なうことによりなされなければならない。（岡原正幸）

B-15. オンラインで組まれるプログラムについて

T.I. Schwartz: Online Programming [CACM, Vol. 9, No. 3, Mar., 1966, pp. 199~203]

オンラインのために考えられた言語とオフラインのために考えられた言語とはかなりの相違がある。この相違は言語が用いられるシステムの周辺装置により生ずる。物理的な周辺装置はコンソール、ディスプレイなどであり、計算機を取扱う人により影響される。

一般にオンライン言語はオフラインにも応用されるが、オフライン言語をオンラインに用いる場合は相当言語が不足する。

オンラインでは、ユーザは計算機と対話できる端末装置を持っており、時分割を行なって計算機の使用効率をあげ、ユーザの経費を節減している。

インプットにおいて、オンラインとオフラインとの相違点をオンライン側から考えると、次のようなものである。

- 1) 詳細な計画が不要である
- 2) プログラムの誤りが容易に訂正できる
- 3) アウトプットが少量である
- 4) ユーザは自分自身のインプットをコンソールなどを用いて直接計算機に入れることができる
- 5) ユーザは処理時間を占有する

人間との相互関係をのぞけば、オンラインとオフラインとのプログラムはあまり変らない。しかしオンラインでは小さなルーチンを合わせて大きなプログラムを作成する。小さなルーチンはインプット、デバッグその他が容易で、タイプしやすいなど有利である。また、これに用いられる言語になくってはならない重要な特性はユーザとプログラムとの相互間の要求を簡明に述べる能力である。

プログラムはシステムにインプットされ、修正が行なわれる。次にコンパイルが行なわれ、プログラムが処理される。

オンラインでの誤りは処理中のプログラムのアウトプットから直ちに発見される。すなわち、検出と訂正がすみやかに行なわれる。

オンラインプログラムの作成過程における最も興味ある問題は、その過程のはなはだしい時間短縮である。すなわち、コーディングからデバッグまでの全過程が数時間で何回もくりかえし行なうことができる。また、このようなシステムを作るには、周囲の状況によ

ラムに

CACM,

ラインの

ある。こ

置により

イスプレ

される。

用される

合は相当

きる端末

の使用効

インとの

うなもの

ってニーザに強制される操作方法とその状況を効果的に用いる操作方法とを考えなければならない。これらを考えることにより、ある限られた時間を一層有効に使うことができる。(岡原正幸)

D-16. 磁性薄膜記憶素子へのキーパの効果

C. G. Ravi and G. G. Koerber: Effects of a Keeper on Thin Film Magnetic Bits [IBM J. Res. and Dev., Vol. 10, No. 2, Mar., 1966, pp. 130~134]

電流の帰路を兼ねる厚い金属基板接地面に蒸着された磁性薄膜記憶素子に高導磁率のキーパを設けることによってもたらされる利点についての理論解析と実験結果を示している。非金属キーパの効率率は73%、金属キーパの効率率は56%であった。

キーパの存在によるストリップラインの鏡像電流を導入し、接地面が存在するので鏡像の手法を用いてストリップラインによる接地面上の接線方向磁界(Ht)の理論式を求めた。これによるとキーパを使用すれば磁性薄膜中の磁界は増加し隣接ビットへの漏洩磁界は減少する。特定の構成の記憶アレーにおいて漏洩磁界の減少率は計算値では93.5%、実験値では非金属キーパの場合100%、金属キーパの場合92%となった。

特定の語線が繰り返し選択されれば語電流に直流成分が生じ帰路電流の直流成分は接地面に拡がる。この現象があるので接地面中の鏡像電流として KI ($K \leq 1$, K の値は語電流の衝撃係数に依存)の値を用いて磁性薄膜のスウィッチングに有効なHtの理論式を得た。これによるとキーパを使用すればHtの K に対する依存性は大幅に減少する。実験によると特定の語線が繰り返し選択された場合、スウィッチングに有効な磁界はキーパなしでは15%減少したのに反し、キーパありでは5%しか減少しなかった。

接地面を貫通する磁性薄膜からの磁束はスウィッチング中に過電流の時定数でトラップされるので、“1”(“0”)読取り“0”(“1”)書込みの動作モードでは書込み情報磁界と逆極性のバイアス磁界が発生し記憶特性は悪化する。キーパを使用すれば接地面を貫通する磁束は減少するのでバイアス磁界も減少することが実験で確認され、理論式とよく一致した。同様な理論でキーパを使用した場合の隣接ビットからの磁界も計算できる。

上述したキーパの機能を考慮して総合効率を第1表に示した。キーパの材料としてフェライトが適しているが、キーパの動作に対するキーパ材料の導電率と誘

第1表 Worst case effects and keeper efficiency

Worst case effects	Efficiency as a percentage reduction in worst case effects	
	Metallic keeper	Non-metallic keeper
Trapped flux effect	40	64
Ground plane current spreading effect	55	73
Stray field from adjacent bit lines	67	75
Stray field from adjacent bits	80	100
Overall efficiency	56	73

電率の効果については、今後の問題として残されている。(村上 坦)

E-17. 印刷漢字の認識

R. Casey and G. Nagy: Recognition of Printed Chinese Characters [IEEE. Trans on Elec. Comp. EC-15, No. 1, Feb., 1966, pp. 91~101]

中国における出版物は1962年には年間350万語だったものが、年々25%という非常な速度で増大している。文献を機械で翻訳することは方々で真剣に考慮されているが、大きな問題は中国語の性質よりもむしろ文字をいかに効率よく符号化するかにある。統計によると90%程度までの文書に含まれる漢字の数は約2,000~5,000とされており、漢字読取機はこの程度の数の文字は読む必要がある。そこで1,000字の漢字を20字ずつ印刷し、写真に取ってフライングスポットにより23×23ビットに量子化した。20字のうち半分を用いて認識論理を構成し、残りをテストに使用した。

認識としてはマッチング法を採用したが、一度に分けるよりも、類別分類法(クラスタ)によるのが有利である。ランダムにマスクを選ぶ方法、Bonnerの分類法、扁と旁(つくり)による方法、グループ内距離の最小化法などを比較したところ、扁と旁および最小距離法の組み合わせが最も効果があることがわかった。

次にグループ内で各個の文字を区別するマスクを選ぶ方法として、一つの文字内で確実に白または黒になる点だけを各15個ずつランダムに選び、他の文字と比較して十分な非一致性が得られるまで点を増加した。

以上を7094でシミュレートしたが、マスクはそのままではコアに入り切らないので、白と黒の点の数およびある点から次の点までの距離の形でパックするこ

とにより、1,000 字分のマスクを 20,000 個の番地に格納することができ、毎分 40 字の割合で処理した。もし 360 の 62 で実行すると毎分 800 字に当たるであろう。

誤読率と読取不能率は互いに相反関係があり、7% 不能で 1% 誤読であった。マスク中の点の数を増加すると率が向上するが、43 点と 200 点のマスクでは改善されなかった。誤読は 3 種に分けられる。第 1 は走査部の解像度不足によるもので、明朝体は横線が細いため由と山などは混同しやすい。第 2 は文字線のかけによるものであり、今後の改善の余地がある。第 3 は走査の不完全さにより、白黒のはっきりした点が十分取れなかったために起ったものである。マッチの範囲をもっと拡大すれば改善されるであろう。

(玄地 宏)

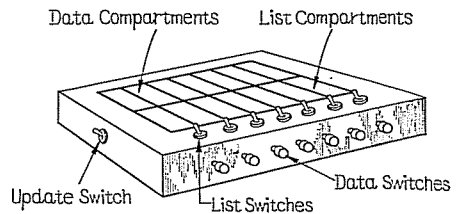
F-18. コンソール上の情報処理における人間の短期記憶能力の利用

B. P. Zeigler and T. B. Sheridan: Human Use of Short-Term Memory in Processing Information on a Console [IEEE. Trans. on Human Factors in Electronics HFE-6, No. 1, Sep., 1965 pp. 74~82]

機械を用いる思考 (machine-aided cognition) 方法が発展するにつれて、コンソール上における人間と機械の情報のやりとりが重要になる。この場合、人間側に何らかの形で記憶能力が要求されるが、その機構については、ほとんど知られていない。この論文は、人間の短期記憶機構に関する実験と、そのモデルの示唆を与える。

次のような実験を行なった。第 1 図のようなコンソールで、データ・スイッチを押すと、データのますの一つに、2 個の数字があらわれ、その積の 1 位の数を、X, Y, Z の三つのカテゴリーに分ける。次に、それま

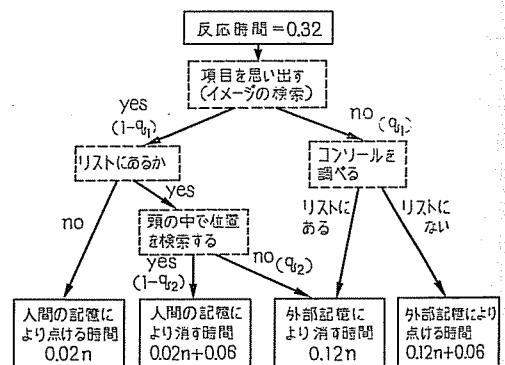
でにリストのますにならんだ X, Y, Z のリストの中に、現在得たカテゴリーが含まれていれば、そのリス



第 1 図 コンソール装置

ト・スイッチを押して、そのリストのますの字を消し、含まれていなければ、現在の位置のリスト・スイッチを押して、リストに新しい字を追加する。これをリスト処理と呼ぶ。以上を 7 回くりかえして、1 回の実験を終わる。

実験条件を種々変え、それらの実験結果から次のような推論を試みた。人間の記憶は連想的であり、その検索も連想的である。検索方法のモデルは、第 2 図のようになる。パラメータを適当にとることによって実験結果と良く一致する。



第 2 図 検索モデル、数は秒である。
 q_1, q_2 は各状態の確率

(田村浩一郎)