



Title	建築構造解析と電子計算機
Author(s)	橘, 英三郎
Citation	大阪大学大型計算機センターニュース. 1973, 10, p. 71-79
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/65192">https://hdl.handle.net/11094/65192</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

## 建築構造解析と電子計算機

大阪大学建築工学教室 橘

英三郎

最近は何にによらず、`超、という字があたまにつかないと人はびつくりしなくなった。ちよつと耳にする言葉からさがしても、超特急、超大型コンピューター、超音速飛行機、超高層ビル、ウルトラマン(?)等がすぐに浮かんでくる。ところでこれらの言葉の中でも、超音速飛行機と超高層ビルとでは`超、の意味が少し異なるようだ。超音速飛行機の方は文字通り音速を超える飛行機の意となるが、一方超高層ビルの方は、高層を超えているビルとしたのではよく分からなくなる。それでは高層ではないのか(?)とひらきなおりたいくなる。これは、音速の方はほぼ一定の値をもっているのに対し、高層の方は基準のとり方でどのようにも考えられるということから生じているのであろう。普通このような場合言葉を補って、`〔在来の)高層(の概念)を超えているビル”とすると分かりよくなる。つい最近にいたるまで、建築基準法では、建物の高さは100尺(約30m)をこえてはならないという規定があつたので、それまでの高層ビルといえは100尺までのせいぜい7~8階止まりの建物であつた。それが電子計算機の出現や、解析理論の発達により、以前からかなり問題とされていた100尺という高さの制限がそれほど意味のないことが分かつてきて、現在では50階程度の建物まで建てるのが技術的にも可能となつてきたのである。

ところで一般に建築の分野での研究の対象となるものには`超、というような種類の文字とは実は縁がうすい。どちらかといえは`適正な”、`適度”、`快適な”といったような、極大と極小との間のどのへんに線をひくかというような仕事が多いような気がする。快適な都市計画、窓からの適正な採光、構造の最適設計等である。そしてこの`適当”という言葉ほどやつかいなものはなく、いろいろな立場からみたいろいろな`適当”がそこにあるわけである。

それはさておき、建築工学の分野は大きく分けると、建築計画、施行、構造、材料、環境、等に分けられるが、電子計算機の影響が最つとも大きかつたのは、構造のなかの構造解析の分野であろう。もともと他の分野よりは、数式と関係が深かつたことも理由の1つであるが、それよりも、航空工学の分野で開発されたMatrix法、有限要素法(Finite Element Method)による影響が大きい。

Boeing社の技師により1956年に発案された有限要素法は、この17年間で力学における構造解析のあらゆる分野において広く用いられるようになってきた。弾性論の近似解法の1つにRitz法があり古くから知られているが、この考え方に似ており、たゞ電算機を用いることを前提としたことと、又汎用性に富むという特徴がある。計算内容はほとんど行列の演算からなつ

ている。そしてこの方法を用いれば連続体の応力解析は全て連立1次方程式の問題又は行列の固有値の問題を解くことになってしまう。静的な釣合問題を解く場合実際の計算手順はつぎの部分からなる。

1) 連続体をいくつかの小さい構成要素に分割し、各要素の特性を計算する。つまり各要素の特定の点での力と変形の関係式を Matrix の形で表現する。

2) 全ての要素の特性を組合せることにより系全体の特性をあらわす剛性係数  $[K]$  をつくる。

3) 剛性係数  $[K]$  の逆行列  $[K]^{-1}$  を求めることにより、外力が加った際の変形を求める。この方法を建築構造物の解析に応用することにより、それまで煩雑な手間と高度な計算技術が要求されていたシェル構造等のうちの単純な形をしたものは簡単に解けるようになった。

ところでこの有限要素法を用いて、複雑な形をした建築構造物全体を一度に解こうとすると、一つの問題が生じる。それは前記の計算手順の②のところで全体の剛性係数  $[K]$  がかなりの大次元行列となり、その逆行列を求めるために数値解析上の種々のトラブルが生じてくるという問題である。一般にこの行列  $[K]$  は0要素の多い対称行列であるので、その特性を利用した種々の数値解法（ブロック消去法、ユニット分割法）が既に考えられているのであるが、それでも丸め誤差（round off error）の問題や、又周辺記憶装置を用いる必要等が生じて、なかなかやつかいなこととなる。例として、Fig 2(a)に示すのは1枚の開口部のある壁である。これだけを有限要素法で解析しようとする、100元程度の剛性行列となる。さらにこのような壁が100枚程度からなる中規模の建物の全体の剛性行列となると、結局、10000元程度の行列となる。材料が弾性域内での解析であれば、この逆行列を1回だけ求めれば良いのであるが、さらに弾塑性解析となると、10000元の行列の要素を序々に修正しながら、1000回程度解かねばならず、いかに電子計算機といえども、誤差の集積から計算自体あまり意味のないものになってしまう可能性が生じる

では電子計算機のない時代の建物はどのように解析されていたのか？という誤問が生じるかも知れないので、それについて一言いっておく。建築構造力学の分野には種々のエレガントな略算法があり、それを用いると、慣れた人なら大程の建物を2～3日で解いてしまう。しかも電子計算機のある今日でさえも、ほとんどの建物がこの略算法により計算され、建てられているのである。機械や精密の関係の方が聞かれたらおそらくお尻がこそばかることであろう。こうした略算法は、せいぜい手廻し計算機しかない時代に、建築にたずさわる多くの研究者の経験と工夫とにより考案されたものであり、多小の問題点はあっても、安全率などによりカバーされた、これなりにまとまったものなのである。では計算尺の2～3日からなぜ10000元の1000回にとんでしまうのだろうか。それは、航空工学の分野で開発された解法を、解析の要求精度の建築の分野にそのまま用いているために生じるものと考えられる。したがって、建築構造物の解析に適した、中程度の大きさの電算機でとり扱うことのできるような解析理論の開発及び研究がこれからの重要な課題となってくる。

以下では、我々が壁のついた骨組構造の解析に用いている有限要素法の略算法について簡単

に紹介する。この略算法の主な目的は、できるだけ周辺記憶装置を用いずに全て in core で処理することにある。又そこに用いられている力学的な面での主な特徴は、未知数となる変形量の間に、ある簡単な変形仮定を新たに導入し、それらの仮定の系全体に及ぼす影響が小さくなるように、エネルギー的な補正が行なわれている点である。我々が取扱う連立1次方程式の未知数は、構造物に外力が加った場合の有限の点での変位量に対応しているのであるが、ごく近傍のいくつかの点の変位量は連続体の性質から、それほど大きな差のないことを利用し、これらの未知数の間にある簡単な函数を仮定する。そして同時にそれらの仮定を用いても、系全体のエネルギーとしては仮定を用いない場合の値と自動的に同じものとなるように係数を補正しておくのである。この操作により系全体の特性をあらわす行列を求めると、仮定する関数にもよるが、大体 たて、よこそれぞれ、 $1/10$  程度の行列となる。又、その時の行列は、もとの対称性を失っていないので、その性質をさらに利用することもできる。

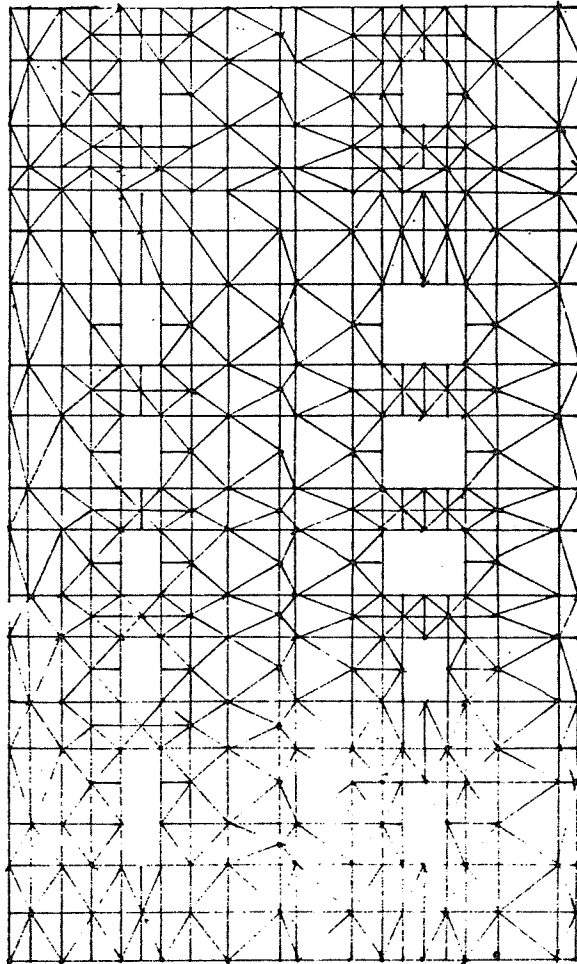


Fig. 1

次に実際の計算手順を図をおって簡単に説明する。今、Fig. 1 に示すのが、開口部のある建物を有限要素法の手法により 3 角形要素の集りに分割した図である。各線分の交点がそれぞれの 2 つの未知数を表わすことにすると、Fig. 1 はそのまま最終的な連立 1 次方程式の係数関係を表わすリニヤグラフとなる。

Fig. 2 は Fig. 1 のリニヤグラフを略算法で求める場合の計算手順を示したものである。

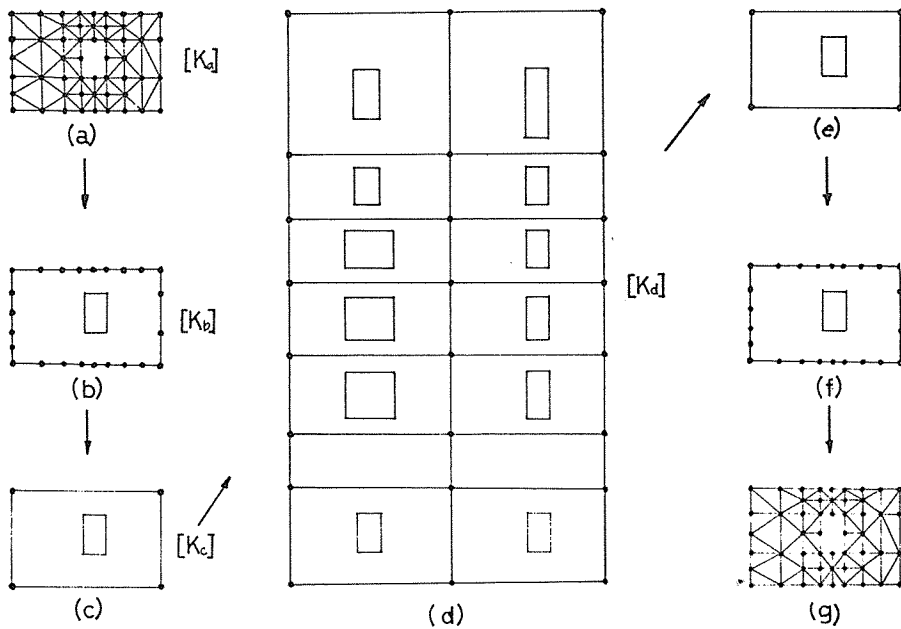


Fig. 2

Fig. 1 を適当な大きさのいくつかの部分グラフに分割しその 1 つをとりだしたものが、Fig. 2 の (a) にあたる。このうちで他の部分グラフと接続しない内部の点に相当する未知数はこの段階で消去可能となるので、それを消去したものが (b) である。(図が複雑となるので省略されているが、実際にはどの点も他の全ての点と線分でつながっている。)(a) から (b) への操作は行列のブロック消去法として良く知られているものである。次に残った周辺の未知数の間に簡単な変形関数を仮定し、エネルギー的に等価となるように係数を補正することにより得られたものが (c) である。残った他の部分グラフについても (a) ~ (c) までの操作をくり返して行う。その結果得られた (c) のような簡略化されたグラフを全て組み合わせることにより、(d) となる。このグラフがあらわす行列の逆行列を求めることにより、外力が加わった際の (d) の各点に相当する未知数が求まる。さらに、それまでの逆をたどり (e) → (f) → (a) の操作を部分グラフの数だけ繰り返すことにより、全ての未知数を求めることができる。(b) の段階からすぐに全体の剛性係数を組み立てることもでき、この場合はブロック消去法と同じものとなるがそれを Fig. 3 に示す。ブロック消去法では仮定を用いていないので理論上での誤差は生じない。しかし、Fig. 3 と Fig. 2 の (d) と比較すると分かるように未知数はまだ

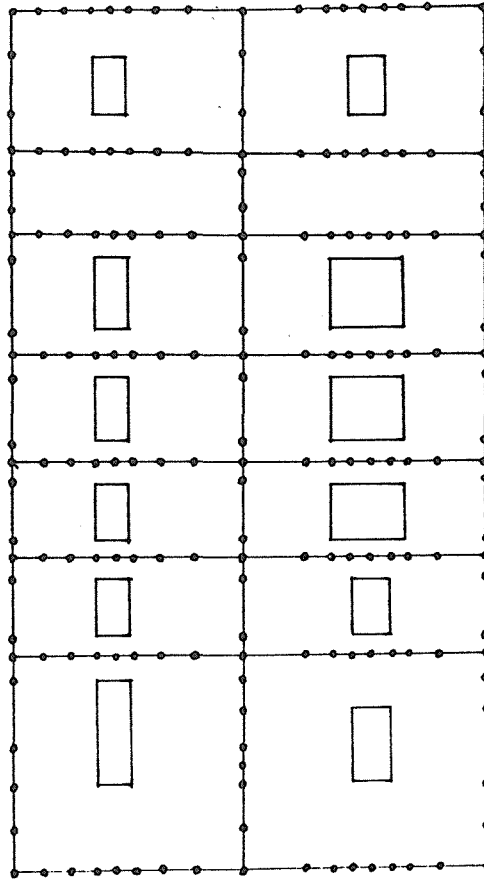


Fig. 3

なり多い。略算法を用いた場合、当然仮定による誤差は生じ、実際に計算をした例では、最大誤差が3%位であった。一方、CPUタイム、逆行列の元数は共に1/10程度で済んだ。Fig. 4ではcore memoryの内容の変化について示したものである。

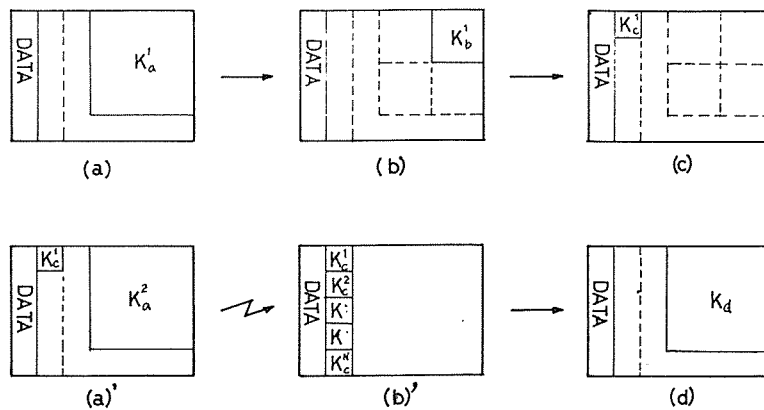


Fig. 4

(a)(b)(c)(d)の番号はそれぞれ Fig.2 の各ステップの番号に対応している。(a)は基本データにより、初めの部分グラフに相当する剛性行列  $[K_a]$  を作ったところを示す。次に内部の点に相当する未知数が消去され、剛性行列  $[K_a]$  は  $[K_b]$  に縮小される。そしてさらに、変形仮定と係数補正とにより  $[K_b]$  に縮小されたところを (c) に示す。次々にこれを繰り返すことにより、他の全ての部分グラフに相当する剛性行列も縮小されて (c)' となる。そしてこれらを組み合わせることにより簡略化された系全体の剛性行列  $[K_G]$  を作成したのが (d) である。以上が略算法の概略である。結局、連立1次方程式の未知数を仮定を用いることにより1桁少なくすることができたかわりに、誤差が3%程度生じることになった"ということになる。話しがもとにもどるが、我々が実際に数値解析をしようとするとき、次の様なジレンマに落ちることがある。理論上での精度を高めようと、未知数の増加等、数式が複雑となり計算上での誤差の増加のため結局精度を高めることができない"といったことである。しかし実際の現象をある1つの理論にのせるとき用いられている多くの仮定が電算機による数値解析の精度にこたえるだけの信頼性があるかという点と疑わしい点も多くある。特に建築の施工現場での工事を見ていると10000元の連立1次方程式がどこまで意味があるのかを考えざるを得ないのである。初めにのべたように、なにかそこに"適度な"線がひけないものだろうか考えるのである。

# グラフィック・ディスプレイの利用形態

京都大学大型計算機センター

西 原 清 一

グラフィック・ディスプレイ装置（以後GDと略記する）は、日常的な道具としてますますその有効性が認識されつつあるが、その反面、それが固有に含んでいる種々の問題点も明らかにされてきた。

従来から指摘されているように、GDの特徴はつぎの点にあると考えられる。

- a 1) 図形（文字も含めて）の即時的な入出力
- a 2) 人と計算機とのインタラクション。

したがって、GDが適用される問題はつぎのような性質を持っていると考えられる。

- b 1) 即時的な情報の入出力
- b 2) グラフ・図形の処理。
- b 3) 試行錯誤的な決定・設計問題

また、GDを利用する立場から見ると、ユーザの作るプログラムには、大別すると、つぎの2種類がある。

- (1) 問題向きの特殊システムにおいて、全面的にそれを応用する場合（GDを問題解決の手段として用いる）。
- (2) 基本的なソフトの上に、問題を解くためのアプリケーション・プログラムを作ったり、GDを効果的に使うようなシステムを開発したりする（GDそのものを研究の対象とする）。

このうちC1)はC2)の構成に大きく依存する。したがってC2)の設計が旨く行われることが結局GDシステムの質を決定することになる。そこで、C1)の含んでいる問題を分析する必要があるが、これは大きく分ければつぎのようなものが考えられる。実際はこれらのうちからさらに具体的な問題が対象となり、それに応じてC2)の設計が方向づけられることになるであろう。

- d 1) 文字・記号列を媒体とした場合。
- d 2) 表示された図形そのものを対象とする場合。
- d 3) 表示された図形を有機的な結合をもった構造体とみなす場合。

このうち、d 1)は先きのb 1)に対応するもので、例えば情報検索（IR）や、（Dを丁度TSSにおける端末（タイプライタ）と同じような使い方をする場合で、特に文字ディスプレイ装置はその特殊な例である。つぎに、d 2)はb 2)の性質に対応するもので、例えば実験データをグラフ表示してその特徴を把握したり、種々の物理現象のシミュレーションを行ない教育的・視覚的效果を図ったり、三面図から透視図を作成して視覚に訴えるというような例がある。また、d 3)はb 3)の性質に対応するもので、例えば回路設計や、フロー・ダイアグラムを用いた



種々の計画など、一般に相互に関連をもった要素からなる図形を取扱うものである。

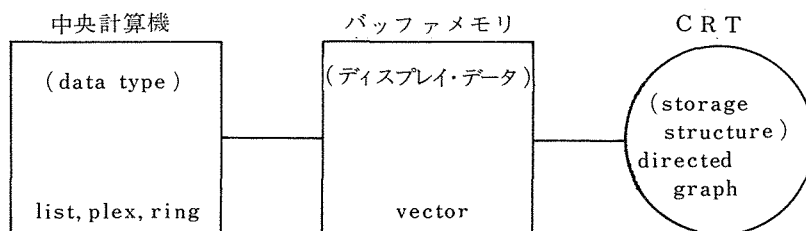
G Dシステムの満たすべき条件は、b 1)～b 3)から明らかにされると思われる。その問題点としてつぎのようなことが考えられる。

- e 1) システム設計上の問題。
- e 2) G Dのハードウェアの問題。
- e 3) データ構造の問題。

まずe 1)は、例えばG DがT S Sの環境にあるときのモニターの設計、とくにバッファ・メモリおよび小型計算機が付置されたG Dは、中央計算機とは独立にある程度まとまった仕事をする機能をもっており、intelligent terminal と呼ばれることもある。e 1)にはG D用言語の開発・作製も含まれる。つぎにe 2)は、b 2)、d 2)に対応するもので、表示された図形を絵として見る場合に重要になってくる。精度、図形発生速度と応答速度、各種図形発生器さらにタブレットなどの図形入力装置、撮影装置（プロッタ、カメラ、ビデオ装置、映画撮影装置）の併置などが問題となってくる。つぎにe 3)は、b 3) d 3)に対応するもので、どちらかといえばソフトウェアの問題である。すなわちデータ構造は、相互に関連をもったいくつかの要素によって構成されている対象を計算機で取り扱える形に表現し保持する機能である。したがってこれに要求されることは、少ない記憶容量に充分な情報が納められること、任意のデータに対する検索・修正が速いこと、汎用であることなどがある。いいかえれば、問題の対象のもっている構造をいかに忠実に計算機の内部に実現するかという問題である。データ構造の例としてつぎのようなものがある。

- f 1) リスト構造（トリー型）
- f 2) リング構造（グラフ型）
- f 3) ハッシュ構造（集合型）

これらは、既成の概念として明確なものではない。またカッコ内はいわば外からみたデータ構造である。個々の詳細については省略するが、各構造の特徴はつぎのように考えられる。f 1)は順序や演算を表現でき、処理言語にはL I S P, I P L, L <sup>6</sup> などがある。f 2)は結合や構造を表現でき、処理言語にはA S P, C O R A L, A P Lなどがある。f 3)は集合や関係を表現でき、処理言語にはT R A M P, L E A Pなどがある。図形処理用言語(Graphic Subroutine Package など)のほかにもこのようなデータ構造処理言語も要求されるであろう。



data type	storage structu	処 理 言 語	表現できる性質	用 途
tree type	list plex	LISP, IPL, L <sup>6</sup>	順 序, 演 算	入力データ列の解析 (言語処理, 定理の証 明, ソーティング)
graph type	ring	ASP, CORAL, APL	結 合, 構 造	構造をもつ図式の処理 (図形処理, PERT, 回 路 網 解 析)
set type	hash tab	TRAMP, LEAP	集 合, 関 係	性質に関する対象の処理 (図形処理, 情報検索)

以上, 大変漠然とした話になってしまったが, このほかにも, 計算センターなどにおいて共同利用する場合には, 必然的にオープン利用という運用形態になるであろうし, これにもまた種々の問題があると思われる。