H.H.Cを用いた園路平面線形の数値解析プログラムの開発

岸 塚 正 昭*
熊 谷 惟 明**
金 井 格*

Development of a Programme for Numerical Analysis of Horizontal
Footpath Alignments by the Aid of Hand-held Computer

Masaaki KISHIZUKA
Koreaki KUMAGAI
Tadashi KANAI

摘要：ハンドヘルド・コンピュータによる解析を目的に、園路線形のモデル解として提案される非線型方程式系に対し、私案を含むいくつかの数値計算手法をプログラム化し、実行時間の短縮について工夫する。刻み幅を小さくすればループ演算が幾何級数的に増大する。本法は刻みを大きくするために、関数のふるまいから部分的に精査するアルゴリズムを創意した。実行結果の精度は、大型計算機の倍精度計算結果との対比により検証している。

1. 緒 言

近年、電算機を用いる解析方法の発達はめざましく、工学上のあらゆる問題について実用的な数値解を求めることが可能になっている。本報は後述の特徴を有するハンドヘルド・コンピュータ（HHC）を用いて、園路線形の測設に要求される精度の数値解を得る目的で、前報で“自然”歩行動線の理論解として提示した実超越方程式系の解析プログラムを開発する。

道路線形の問題は、設計・施工の領域に関わる課題の一つである。古くは自動車の走行力学ないし景観上の視点から追求がなされ、^{1) 2)}既に高速道路では円曲線のみならず、クロソイド曲線をも導入した機能的線形の理論と計算法が確立し実施に移されている。しかし歩く速度が主体である園路線形の場合は、人間的尺度での曲率線形が望ましく、筆者らはこの観点に立脚した研究を行っている。かつて筆者の一人は、自然の大地上に形成された踏み跡道を題材に、人間の歩行軌跡の「曲率式」を回帰した³⁾次いで前報⁴⁾（以下、同文献を指すものとし、肩註を略す）では本式と、2地点間を円曲線で幾何学的に結ぶための「条件式」を連立し、園路線形モデルの解法を大型電算機を用いたFORTRANプログラムで考察しており、今回の新たなアルゴリズムをもって開発されたBASICプログラムは、パソコンレベルの解析手段として実用の域にあると考えたので報告する。

携帯型コンピュータHHCは、個人で購入できる価格であること、コードレスで軽量なため機動性に富み、記憶容量（RAM）も数10KBで、ある程度の技術計算が可能なこと、またはメモリーの不揮発性などの特徴をも

ち、現場技術者による利用事例が多い。本開発研究に使用したハンドヘルドコンピュータ^{*1}は、プリンタ、ディスプレイ、マイクロカセット・ドライブとコンピュータの周辺機器すべてを搭載したオールインワンタイプのものである。前記の簡便性に加えて、HHCを用いた理由は二つある。その一つは、BASICによる数値計算は遅く一般には時間の浪費とされるが、プログラムの開発研究にはむしろアルゴリズムの改善効果が明瞭に現われて、より実行時間の短いソフトの創意に好都合であること。その二は、園路線形の測設に当っては、BASICによる測量計算用の豊富な市販ソフトがあるため、それらとリンクすることで一連の本基礎研究の成果が生かされることである。

数値計算プログラムの総合評価は、①演算時間、②精度、③費用の3項目からなされよう。「良いソフト」さえ作れ（使え）ば、パソコンレベルでもハード的には数値計算をするのに十分な精度を得ることが可能とされる⁵⁾。本プログラムによる計算精度は、最終段階で大型計算機による倍精度計算との比較から検証される。よってここでは演算時間の短縮に的を絞り、2で先づ計算式を提示したあと、3で根の存在区間を効率よく発見する「探索手法」と、根への「収束手法」に対するいくつかの代替案を含むテストプログラムを作成して提示し、4でこれらの実行結果から特にBrentの手法：ZEROINとの比較検討を行い、私案（MK法と呼称）の有効性を示す。

2. 園路線形の計算式

(1) 入力パラメータの検討

* 東京農業大学造園工学研究室 ** 同電算機室

本プログラムの対象となるのは、2点D・A間の園路線形を円弧1+接線+円弧2の順で構成した、複合曲線(C型)ないし反向曲線(S型)に対する線形の解析である。

図-1 点データ取得による入力パラメータ

歩行者の出発点における直角平面座標D(0,0)と身体正面の方向角 θ_1 を基準に、到達点における同座標A(l, m)と方向角 θ_2 が与えられたとき計算できる。現実のデータ入力、ディジタイザ等の図形座標読取装置を利用して、図-1のようにD(x_1, y_1)→D'(x'_1, y'_1), A(x_2, y_2)→A'(x'_2, y'_2)の序列の点データの形で行うことを想定しており、このときは次の式(1), (2), (3)によって入力パラメータ l, m, V をあらかじめ計算する。

$$l = (x_2 - x_1) \cdot \cos \theta_1 - (y_2 - y_1) \cdot \sin \theta_1 \quad \dots(1)$$

$$m = (x_2 - x_1) \cdot \cos \theta_1 + (y_2 - y_1) \cdot \sin \theta_1 \quad \dots(2)$$

$$V = \theta_2 - \theta_1 \quad \dots(3)$$

ここで、Y軸正方向を基準に、右回り θ を正值、左回り θ を負値と定義し、 θ_n は式(4)によって定める。

$$\theta_n = \tan^{-1} \{ (x'_n - x_n) / (y'_n - y_n) \} \quad \dots(4)$$

ただし、 $(y'_n - y_n) < 0$ ならば、 $\theta_n = \theta_n \pm 180^\circ$ とする。

(2) 方程式の吟味

円弧1・2の各曲率半径を $R_1 \cdot R_2$ 、各中心角を $\delta_1 \cdot \delta_2$ において、自然歩行に適う線形の理論解は次の連立方程式系の実根である⁴⁾

$$\begin{cases} V = \delta_1 + \delta_2 \quad \dots(5) \\ R(1 - \cos \delta_1) = R_2(1 - \cos \delta_2) + m \cdot \sin \delta_1 - l \cdot \cos \delta_1 \quad \dots(6) \\ 39 - |\delta n| = 13.5 \log_{10}(Rn - 5.8) \quad \dots(7) \end{cases}$$

ただし、 $\delta n < 0$ ならば、 Rn を負値とする。この式(5)・(6)・(7)の連立系を解くため、代入法・消去法・等置法を適用して、次の変数 δ_1 の一元方程式(8)が与えられる。

$$F(\delta_1) = f(\delta_1) - \{ f(g(\delta_1)) \cdot \text{vers}(g(\delta_1)) + m \cdot \sin \delta_1 - l \cdot \cos \delta_1 \} / \text{vers}(\delta_1) = 0 \quad \dots(8)$$

上記の $f \cdot g \cdot \text{vers}$ の各関数は次のように定義される。

$$g(x) = V - x, \text{vers}(x) = 1 - \cos x$$

$$f(x) = \text{sgn} \left[\text{TEN} \left(\frac{39 - \text{ABS}(x)}{13.5} \right) + 5.8 \right]$$

上記の sgn は、絶対値 $\text{ABS}(x)$ の引数 x が正值のとき+1、負値のとき-1を与え、 $\text{TEN}(x)$ の表記は、指数関数 10^x に同義である。

(3) 園路線形の最適解

前報で式(8)の方程式の解は、複数個出力されることが判明した。それらの理論解のうち園路線形に最適解としては、歩行実験の軌跡図⁶⁾と照合した結果、式(10)の L を最小にする一つの理論解 δ を選別すればよい。この最適解に対する式(9)の接線長 a は⁷⁾、必ず正值であることを絶対条件とするか、もしくは一步の歩幅(≒60cm)以内までは負値を許容して、 $a > -0.6 \text{ m}$ をもって最適解の必要条件とするかは未検討である。

$$a = \{ m \cdot \sin \alpha - l \cdot \cos \alpha + R_1(\cos \alpha - \cos \beta) \} / \sin \beta$$

$$\text{ただし、} \alpha = (V + \delta_1) / 2, \beta = (V - \delta_1) / 2 \quad \dots(9)$$

$$L = R_1 \cdot \delta_1 \cdot \pi / 180 + |a| + R_2 \cdot \delta_2 \cdot \pi / 180 \quad \dots(10)$$

以上みてきたように、自然歩行に適う園路線形の曲半径 R とその中心角 δ を解析する原理は、式(7)を基本解に2点間を2円弧で結ぶ条件式(6)を解く連立法である。

ところで式(8)は非線型な超越関数方程式であり、次章においてこれの効率的な数値解析プログラムを開発する。

3. プログラムの開発

(1) 計算の手順

超越方程式においてはすべての根(関数の零点)が一括して得られるような、大局的な収束性をもつ解法は知られていない。前報のパイロットプログラムでは、 $\delta = \pm 360^\circ$ の変域の中で刻み幅 $H=1^\circ$ の δ ごとに、関数 $F(\delta)$ を $N=720$ 回ループ計算して、この間に関数の符号が変わる区間 $[\delta^{(n-1)} \delta^{(n)}]$ を探索した。中間値の定理から、この区間内には関数 F が連続であれば、少くとも一つの実根が存在する。次いでこの存在区間について、刻み幅を $10^{-1}, 10^{-2}, \dots$ と縮小した10進分

図-2 計算の概略流れ図

表-1 テストプログラム・リスト

10	20	30	40
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****
* ****	* ****	* ****	* ****

法による収束計算を行った。本研究でも同様に2階層に分けた計算手順であるが、のちに計算時間の比較検討を行うために、探索計算が刻み幅 H の入力パラメータ、収束計算が6手法の代替案を含むテストプログラムが作成された。図-2に全体の概略流れ図を示す。

(2) 探索手法の開発

初期の刻み幅を粗にすれば探索時間は短縮するものの、偶数個の実根を区間に挟む場合は、その両端点 $\delta^{(n-1)} \cdot \delta^{(n)}$ の関数値は同符号で変化せず、前項(1)のアルゴリズムのみでは根の存在に気づかず失敗する。いま $\delta^{(n-1)}$ 、 $\delta^{(n)}$ 、 $\delta^{(n+1)}$ の順に式(11)による両隣りの関数値の差異 E_n 、 E_{n+1} が連続的に調べられたとき、 E の符号が正から負へ変化すれば、零点に向かって接近してきた関数グラフが再び離反していく状況を意味する。

$$\left. \begin{aligned} E_n &= F(\delta^{(n)}) - F(\delta^{(n-1)}) \\ E_{n+1} &= F(\delta^{(n+1)}) - F(\delta^{(n)}) \end{aligned} \right\} \dots\dots\dots(11)$$

よってこの挙動を検出した場合は、その前後の変域に対し刻み幅 H をさらに狭小な $H/10$ に変えて、より詳細に符号の変化する区間 $[a, b]$ の有無を探索してみる。この具体的なルーチンの例を、プログラムリスト(表-1)の行番号1350-1570に示す。

(3) 収束手法の開発

前項(2)によって、 $F(a)F(b) \leq 0$ となる2点 a, b を両端点とする区間 $[a, b]$ が求められる。さらに根を常にはさみながら、区間幅を縮小していく手法に2分法(bisection)、按分法(regula falsi)が知られる。また条件つきで収束する手法に、 a (または b)側に2個の初期値 $x^{(1)}$ と $x^{(2)}$ を与えて、按分法の式(12)から $x^{(3)}$ 、 $x^{(4)}$...と次第に根の近似解 $x^{(k)}$ を改善していく、割線法(secant)がある。

$$x^{(k+1)} = x^{(k)} - \frac{F(x^{(k)})(x^{(k-1)} - x^{(k)})}{F(x^{(k-1)}) - F(x^{(k)})} \dots\dots(12)$$

以上の各手法の収束特性を検討して考案された収束プログラムが、表-1の行番号2450以下のサブルーチンである。このアルゴリズムで筆者が主張していることは、多分に形態学的な発想に依拠している。つまり、①最初のステップの按分法(一次線型補間)による計算結果の判定から、根をはさむ区間の関数グラフの形を推定して、②もし直線状に近ければ収束次数の高い割線法に乗り移る。③判定条件から凹または凸状の曲線と推定されたときは、安全確実な2分法を選択して、根から遠い端点 a (b)側を次第に縮小するが、この間に b (a)側へ縮小区間が切り変わった時点で再び最初のステップ①へ戻る。②、③の判定用条件式(リストの行番号2530, 2540)の導出は経験的性格をもつ。按分法は、2分法の最終段階の手順として組込まれている。以下、本開発手法を仮にMK法として記述する。

(4) 停止規則その他

反復計算を有限回で打切るための規則として、次の2つの条件式を設定し、刻み幅 H が式(13)と式(14)を同時に満足したときループ計算を停止して、式(8): $F(\delta_1) \div 0$ となる $\delta_1 = \delta(k)$ を解として出力する。

$$|F(\delta(k))| < E_1 (> 0) \dots\dots\dots(13)$$

$$|\delta(k) - \delta(k-1)| < E_2 (> 0) \dots\dots\dots(14)$$

ここで、収束停止標準である入力パラメータ $E_1 \cdot E_2$ は小さく選ぶほど解の精度の向上が期待できる。しかしまた、 E_2 を計算機イプシロン(EPS: リスト1920-1950)よりあまり小さく与えても、関数計算に付随する誤差のために、修正量:式(14)の左辺が E_2 より小さくなれないことも知られている⁸⁾。

4. 実行結果と考察

本開発手法の効果を比較するためのテストプログラムが表-1である。本報で得られたすべての解の精度と計算時間は、収束停止標準 E_1, E_2 に対して、測量器械の精度:測角1秒・測距1mmを参考に、共に 10^{-5} が与えられた場合の出力結果である。有効数字の最多桁数は、小数点下5桁+上3桁=8桁を要することになり、このため最悪の場合に計算誤差は計算桁数の下位半分に達することを予定し、16桁の倍精度関数ユーティリティプログラム(マシン語)がLOADされた。比較の対象とした入力パラメータ $l \cdot m \cdot V$ は、前報のもの9例をそのまま利用する。以下では、他の手法を含めた実行結果との比較から、本開発手法の有効性について論じる。

(1) 探索手法の効果

初期の刻み幅 1° を 10° にすれば、所与の変域全体に対する計算のループ回数は $1/10$ になり、計算効率の上昇は当然とみる。ここで示すべき効果とは、刻み幅 H を粗くしても前報で得られた解を同様にものも捕捉可能かどうかである。初期刻み幅 $H=10^\circ$ を入力したテスト結果は、いずれも問題なく解を出力した。式(11)の実効について検証するために、その1例として意図的な初期刻み $H=13^\circ$ を入力して試行した結果を、出力フォーマットそのままの図-3に示す。一方、初期刻みを $15^\circ, 20^\circ$ にして検討したが、次の計算ステップの収束時間が逆に増大して効率が悪い。以降は 10° に限定して述べる。

(2) 収束手法の比較の方法と結果

収束手法は従来の解法の5種類と今回の開発手法、即ち10分法・2分法・按分法・按分+10分法⁹⁾・ZEROIN¹⁰⁾・MK法、の6種類を用意した。比較の指標には、ループ回数、演算時間を考えそれぞれに出力するようにフォーマットされている(図-3参照)。9例のテストデータの各収束手法に対する実行結果の比較を表-2に示す。

(3) MK法の結果についての考察

1ループの計算に要する時間は平均4.4秒であり、そのバラツキは平均を中心とした正規分布に近い。9例の

変動幅が4.1～4.9秒と小さいのが特徴で、ほぼ計算時間とループ回数は正比例する。事例(1)～(9)の出力解の個数は、2～4個と内容が異なり、収束時間単独での比較は意味がない。例えば例(4)は2個で、かつ探索区間幅が 10^{-4} のオーダーまで縮小されたため、収束のそれは最短

時間だが、探索時間との合計は約12分と最長になるからである。今回の試行にみる限り、一般的な性格の入力データならば6分前後で最適解が出力されると思われる。

(4) 別手法：ZEROINとの比較

表-2が明らかにしたように、10分法、2分法、按分法のアルゴリズムを単一で使用することは、計算機が演算を速く行う保障がない限り、プログラムの測度が極めて悪く得策でない。ZEROINは、Dekker (1969)のアルゴリズムのBrent (1973)によるFORTRAN プログラムとして知られ¹⁰⁾、そのサブルーチンをリスト行番号1900に示す。ただし、行番号2230-2240の実行は永ループに陥るため、この部分を行番号2250に変更して移植したものとの比較である。

ZEROINの1ループ当りの実行時間は、5.0～7.3秒と変動の幅が大きいことに特徴があること、その平均タイムはMK法より1.5秒ほど遅い。このため9事例の比較において、30秒～3分の開きが生じた。ループ回数がMK法より少ない例(5)～(8)でも計算時間はむしろ遅い。ここで速くなっている分は、BASICの場合には貴重である。このZEROINは、計算機メーカーの提供するソフト²⁾の中にも存在するように、単一方程式の零点を求める最良の計算機アルゴリズムである。如何なる関数にも対応でき、頑強な汎用性ゆえに信頼度も高いそれを、本論は決して拒否するものではない。システム設計の開発環境により使い分けることが必要と考えるからである。

(5) 計算結果の評価

既述してきた比較論は、各手法とも収束停止規準 $E_1 \cdot E_2$ が同条件のもとでの同一計算機による収束解であり、このため出力値が皆同じ誤差水準であったから成立した。一方、計算結果の精度は使用機種にも依存し、他機種との同アルゴリズムによる対比手法は、精度評価のために我々が簡単に行い得る手段の一つである。次に大型計算機³⁾による実行との対比結果を記す。

この精度検定の実行においては、プログラムリスト行番号1050の箇所を1060のように、計算式： $1 - \cos \delta$ の

図-3 出力例

表-2 実行結果の比較

例	デ ス ト デ ー タ			探索時間 (初期刻み H=10°)	収 束				法			
	V	L	M		10分法 time loop	2分法 time loop	按分法 time loop	按+10分法 time loop	ZEROIN time loop	M K 法 time loop	法	法
(1)		5.8 m	5.8 m	6:29	8:35 144	4:13 66	13:29 -	3:25 52	3:37 37	2:22 32		
(2)		5.80016681	5.80016681	7:03	14:26 244	7:22 118	20:00 -	8:25 133	8:29 101	5:37 81		
(3)	+90°	30.225	7.938	3:46	10:07 168	5:24 84	10:54 -	3:26 50	4:23 45	2:38 35		
(4)		17.4	-5.8	11:09	4:55 81	2:22 36	6:53 -	1:06 15	1:20 11	0:49 10		
(5)		17.40050043	-5.80016681	3:46	8:35 141	5:24 84	19:46 -	3:19 48	4:22 43	3:12 44		
(6)	0°	23.2	5.8	3:46	7:33 123	5:23 84	3:26 49	2:48 40	2:46 24	2:16 30		
(7)	+75°	69.285	114.165	3:52	8:46 144	5:37 88	3:47 54	3:54 58	2:53 25	2:14 29		
(8)	-90°	34.0	16.0	3:51	7:38 124	5:24 84	10:16 -	3:51 57	3:07 28	2:33 34		
(9)	+270°	34.0	16.0	3:52	8:08 133	5:24 84	16:05 -	3:55 58	4:07 41	2:33 34		
				計	79:57 1302	46:33 728	※ ※	34:09 511	35:04 355	24:14 329		

※註) はさみうち法と称するように必ず解に収束する手法である。合計欄が未記入なのは、収束がのろく途中で計算を打ち切ったためである。

実行式： $2 \cdot \sin^2(\delta/2)$ による式の変換が行われた。その結果、前報の表-1で提示した停止規準 $E_1=10^{-5}$ の 10^{-8} との対比誤差はすべて消失すると共に、それらと本プログラムの2つの停止規準 $E_1=E_2=10^{-5}$ による精度とは全事例とも同水準であった。これの具体的数値の提示は、前報の表-1との重複を避けて割愛する。

5. 結 語

著者らは、園路線形を数値解析するBASICプログラムの開発において、より少ない反復回数で実根の近傍値を探索するロジックの創意と、「MK法」という新しい収束用アルゴリズムを創案し、これらの効果を中心に検討を行った。その結果は次のように結論できる。

(1) 探索計算で前報のFORTRANプログラムは、刻み幅 1° で720回、 0.1° で7200回と反復回数を幾何級数的に増大させる。本プログラムは 10° 刻み(72回)で済ますことも可能で、この回数の節減による時間短縮効果は、FORTRANに移植した場合でも大きい。

(2) 収束計算でMK法は、従来の手法ZEROINに比較して30秒～3分ほど収束時間が速く、BASICにおける有効性が実証された。このため前項の探索時間と併せて、HHCを用いた実行時間は $\pm 360^\circ$ の探索範囲で6～12分であった。既報⁶⁾の歩行実験によれば探索範囲は $\pm 150^\circ$ と半減すること、さらに上級パソコンの演算速度で換算したとき、このBASICプログラムは十分に実用の域に達したものとみてよい。

最後に、表-1のリスト(プログラムサイズ7442+倍精度関数1280Byte)の中から、今回の使用機種^{*1}に特有のエプソンベーシックを掲げ、他の機種への移植に当たっては注意がいることを付記する。

WIDTH<1行の文字数>,<行数>,<スクロールマージン>……仮想スクリーンの大きさを設定する。
SOUND<音程>,<長さ>……テキスト画面を鳴

らす。

註および参考文献

- * 1. エプソンHC-20
- * 2. 例えばFACOMでは、SSL II(サイエンティフィック・サブルーチン・ライブラリー-II)
- * 3. FACOM M-360
- 1) 鈴木忠義編訳(1966):国土と都市の造形:鹿島出版, 157-276 [原著:C. Tunnard & B. Pushkarev: Man-made America; Chaos or Control?, Yale University Press, New Haven, 1963]
- 2) 中村英夫編訳(1976):道路の線形と環境設計:鹿島出版 [原著:H. Lorenz: Trassierung und gestaltung von strassen und autobahnen, Bauverlag GmbH, Wiesbaden & Berlin, 1971]
- 3) 岸塚正昭(1969):園路の曲率に関する研究-特に曲る方向と曲半径の関係:造園雑誌32(4), 24-30
- 4) 岸塚正昭(1985)ほか:自然歩行による園路線形の解法に関する考察:造園雑誌48(5), 211-216
- 5) 春海佳三郎(1985):パソコン数値計算における誤差と対策:インターフェース11(3), 314-328
- 6) 岸塚正昭ほか(1984):歩行動線の幾何学モデルの検証:造園雑誌47(5), 図-2～6, 221-223
- 7) 本式の誘導は、前掲4), 213-214, 参照
- 8) 伊理正夫(1981):数値計算-理工系基礎の数学12, 朝倉書店, 51
- 9) 竹之内脩(1984):パソコン微積分入門-BASIC数学別冊, 現代数学社, 37
- 10) 森正武訳(1978):計算機のための数値計算法:日本コンピュータ協会, 178 [原著:G. E. Forsythe/M. A. Malcolm/C. B. Moler: Computer Methods for Mathematical Computation, Prentice-Hall, Inc., New Jersey, 1977]

Summary: This study intends to develop a method of numerically analysing footpath alignments by the aid of hand-held computer. For this purpose, several alternative methods of numerical computation, designed to solve a set of non-linear equations proposed as a model solution of horizontal footpath alignments, were programmed, and these programmes were further elaborated in order to reduce the computation time requirement. As is empirically known, the frequency of loop operation increases in geometric progression when the initial interval is made smaller. Hence, in order to reduce the computation time requirement (i.e., to enlarge the interval), an algorithm was invented that allows detailed examination on a partial basis in accordance with the behavior of the function. The accuracy of the hand-held computer operation was verified by way of comparing it with the double precision analytical result derived from the operation of a large-capacity computer.