二次元補間　不等間隔の関数データ

ドナルドシエパード
ハーバード大学
マサチユーセツツ州ケンブリツジ

前書き

経験的面積データを使用する多くの分野において、連続的な表面を生成するために不規則な間隔のデータから補間する必要性が生じる。したがつて、「データポイント」と呼ばれる、これらの不規則な間隔の場所は、さまざまな意味を持つ可能性があります。地理学では、調査場所。都市および地域計画において、データ収集ゾーンの中心。生物学では、観察場所。各データポイントには固有の数（気象学の降雨量、または地理の高さなど）が関連付けられていると想定されます。

これらのデータをある種の等高線図や透視図で表示したり、他のデータポイントに基づいて同じ地域のデータと比較したり、極端な値、勾配などの目的で分析したりするには、非常に便利です。必須ではない場合は、与えられた値に正確に適合する連続関数を定義します。細かいグリツド上の補間値が次に評価されてもよい。このような関数を使用する際には、元のデータにエラーがないか、またはエラーの補正が補間の後に行われることが想定されています。

本質的には、不規則に離間したデータ点からの二次元補間の問題に対する操作上の解決策が望まれる。有限数Nのトリプレット（x；　y；　z；）が与えられると仮定する。はデータ点D;、zの位置座標です。対応するデータ値です。データポイントの位置は一致していない可能性があります。平面内の任意の位置Ｐ（x、y）に値を割り当てるための補間関数ｚ＝ｆ（x、y）が求められる。この二次元内挿関数は、「滑らか」（連続的かつ一度微分可能）であり、指定された点を通過し（すなわち、ｆ（x；、y；）＝ｚ；）、そしてユーザの直観的期待に応えるものである。調査中の現象について。さらに、この機能は、妥当なコストでコンピユータアプリケーションに適している必要があります。

既存のアプローチ

二次元補間における関連問題に対する多くの解決策が長い間使用されてきたが、不規則な間隔のデータに完全に適合するポレーション関数はまれです。データ点がすでに正則格子を形成している場合は、多くの解決策が考えられます。長方形グリツドの最も重要な解決策の中には、ニユートンの分割差分form11aを使用して多項式を周囲の4、9、16、または25点にフイツトする二重線形補間 [1]，によつて各4つのデータ点に双曲線放物面を近似する方法があります。[2]と [3]で説明されています。または、バイキユービツクスプライン補間を使用します。三角格子の場合、平面を各3点に合わせることは簡単で効果的です。Downingは、正方格子から介在点を補間し、三角格子上で平面内挿を可能にするコンピユータ輪郭処理プログラムを開発しました。[5]不規則なデータ点の位置が許され、連続性が必要でない場合、Toblerの二重などのアルゴリズム二次曲面法 [6]が可能です。最後に、補間関数が与えられた値に正確にフイツトする必要がない場合、トレンドサーフエスフイツテイング [7]は起こり得るデータエラーから歪みを修正するのに適切かもしれません。

連続性と厳密な適合の規定内であつても、規則的なデータ位置のための前述の方法のいくつかは不規則な間隔のデータに拡張されるかもしれませんが、それらは厄介であるか過度に恣意的です。1つの解決策は、データポイントを含む平面のセクションが三角形、または四辺形と三角形の混合に分割されるまで、データポイントのペアを接続することです。NordbeckとBengtssonは三角形を使い、それぞれの三角形を定義する3つのトリプレツトに平面を当てはめた。[8]フイツシヤーは領域の内側に四辺形を使い、境界に三角形を使つた。 [9]彼は、四辺形上の二重線形補間と三角上の平面補間を採用しました。これら２つの解決法は、ドメインの必要な細分化が実行されると、どちらも計算が簡単であるため魅力的である。それらは、ローカルの、簡単に評価される関数の集まりを元のデータに適合させることを含みます。深刻な欠点は、補間のためのドメインがどのように分割されるべきかを決定するためにポイント間でネツトワークを形成する必要があることである。結果として生じる補間はこのパーテイシヨンに敏感ですが、それを確立するための適切な手段はありません。3辺と4辺のサブドメインの周囲の合計を最小にするようにネツトワークを選択するという提案

（[8]に記載）は、従うには複雑すぎます。

実際には。たとえそのような細分が見つかつたとしても、それはユニークではないかもしれません。この分割方法を使用する唯一のコンピユータプログラムであるSYMAPバージョン111[9]では、ユーザーは多少恣意的な規則に従つてパーテイシヨンを指定する必要がありました。これらの区分的補間関数によつて生成された表面は連続的であるが、その導関数はサブドメインの境界では不連続であつた。

ユーザにとつてより簡単でより洗練された代替的なアプローチは、それが正確にすべてのデータ値を仮定するように十分な係数を有する2つの変数に多項式または三角関数を適合させることであろう。Berezinは、xとyの次数N-1の多項式をN個のデータ点sに近似する一般式を提案しています[3]。それはすべての基準を満たしていますが、計算は非常に多くのデータポイントで非常に長くなりました。たとえば、1000個のデータポイントを指定して補間値を求めるには、2次元ベクトルの999個のスカラー積を評価して乗算する必要があります。

二次元の正確な補間の既存の方法は、2つのタイプのものである。単一の大域関数、しばしば管理不可能な複雑さのもの。または境界でほぼ一致する単純なローカル関数の適切に定義されたコレクシヨン。本論文で開発した関数は後者のタイプであり、各局所関数のサブドメインが自動的に定義され、局所関数の接合部でも関数は連続的に微分可能であるように構築されている。

加重平均を使つたアプローチ

重み付けがそれらの点までの距離の関数である、データ点における値の加重平均に基づく面が基準を満たすことが明らかになつた。初期の逆距離関数をテストしました。その中で、平面内の任意の点Pにおける値は、データ点Dにおける値の加重平均であつた。

Zとしよう。データ点Dでの値、a’[P、D;]はPとDの間のデカルト距離です。基準点Pが理解される場合、a’[P、D;]はd;と短縮される。この最初の内挿関数を使用したPでの内挿値は、次のとおりです。

= f

f (p) (d) z 1 r uとf (d r dの場合- :: / - すべてのDに対して0。
u 1
I I=1 I=}
Z; (u>o)
dの場合いくつかのDに対して= 0。

Pがデータ点D;、d; - + 0に近づくと、分子と分母のt ″項がすべての範囲を超え、他の項は制限されたままになることに注意してください。したがつて
lim f 1 (P) = z; 必要に応じて
P-+D;

指数の選択

座標P (x、y) とD; (x;、y;) を使うと、偏微分は次のようになります。

$$f_{1x}(x,y) = \frac{\sum_{i}^{N} \; - \; (d府^{2} \;(dj) - u \;(xx)\; z\; (z-z) \quad i\text{-::/,j}}{\sum_{i=1}^{N} (d;t''J2}$$

上記の式で (xx;) を (yy;) に置き換えると、対応する式fm \、ix (x、y) が得られます。f I の偏導関数はすべての点に存在します。隣人のP D;、fの境界は (x T x;) (d; t・ 2、 または ± (d;) u- /•したがつて、u> 1の場合、fixはP - としてxx; (またはd;) を使用して0を近似します。D 1。u = Iの場合、左側と右側の両方の偏導関数が存在します。これらは一般にゼロではなく、符号が反対です。u <Iの場合、導関数は存在しません。したがつて、内挿関数が微分可能であるという要件は、指数がを超えることを必要とします。

I. 経験的試験は、より高い指数 (u > 2) は、全てのデータ点の近くで表面を比較的平坦にする傾向があり、データ点間の小さな間隔にわたつて非常に急勾配を有することを示した。指数が低いほど、データポイントで適切な値を達成するための短い表示で、比較的平らな表面になります。u = 2の指数は、一般的な表面マツピングおよび記述の目的のために一見満足できる経験的結果を与えるだけでなく、最も簡単な計算も提示する。デカルト座標では、 (d; r 2= 1 / [(x - x;) 2+ (y - y;) 2]) である。

純粋な逆距離重み付けの欠点

上記の方法は非常に魅力的であるために十分に単純で一般的でしたが、いくつかの欠点がありました。

I) データ点数が多い場合、z = f I (P) の計算は比例して長くなります。最終的には、この方法は効率的または非実用的になります。
2) データ点D;からPまでの距離だけ

方向ではなく、考慮されます。だから

たとえば、次の2つの共線点の構成では、Pで同じ補間値が得られます。

関数f1 (P) は連続です。

(a) x- - - - X - - - - + ------------------- X

D1 P

(b) x---------+----x ---- x

D_1 P D_2 D_3

方法が直感的に合理的であることであるならば、

構成（a）のPの値は、構成よりもDの値$_3$に近いはずです。

（b）Dの値$_i$とは逆に、

介在するデータポイントは、より遠いポイントの影響をふるい落とすために期待されるべきです。

3) 各データ点Dで得られたゼロ方向導関数。内挿されたサーフエスに対する任意の望ましくない制約を表します。

4) 支配的な項は2つのほぼ等しい数の差から生じるため、計算誤差は点Dの近傍で大きくなります。

重み付け関数を改善する

解決策として、前述の基本的な重み付け関数が保持され、以下の修正および修正項が適用される。

近くのポイントを選ぶ

上記の重み付け関数は、近傍のデータ点のみが任意の計算に重要であることを意味するので

補間された値、計算のかなりの節約は、離れたデータポイントでの計算を排除することによつてもたらされるかもしれません。それらの包含がより高次の表面を作る傾向があるので、余分な変曲点もまた近くの点のみから内挿することによつて除去することができた。近くの点を選択するために、（Ｉ）任意の距離基準（点Ｐのある半径ｒ内の全てのデータ点）、または（２）任意の数基準（最も近いｎ個のデータ点）のいずれかを用いることができる。

前者の選択は、計算が簡単ですが、

データ点が存在しない、または管理できないほど多数のデータ点が半径rの範囲内で検出される可能性があることを可能にしました。後者の選択はデータ点のためのより詳細な探索およびランク付け手順を必要とし、そして点の相対的な位置および間隔に関係なく、単一の数の補間点が最良であると仮定した。の組み合わせ

2つの基準はそれらの利点を組み合わせました。データ点が格子状である場合に補間が合理的に機能するように、最低４つのデータ点が選択された。必要とされる計算の複雑さおよび量を制限するために最大１０個が設定された。さらに、初期サーチ半径ｒは、データポイントの全体的な密度に従つて設定される。Nがデータ点の総数で、Aがデータ点で囲まれた最大の多角形の面積である場合、r

7つのデータポイントが含まれるように定義されます。

平均して、 7,). 半径 r 。あれは：

rrr 2

補間データポイントを選択して重み付けするには、P付近のデータポイントのコレクションC ’と最終検索半径

ｒ ’は以下に定義される。はじめに

Cp = {D、Jd; $。r}　　andn (Cp) =要素数

Cpで。次に、thの順序を考えます。D; Pからの距離を大きくすることによつて。は、05、d;、s; d;$_2$-5 . . ··d; N

・のように定義されます。

今定義する

$$C_P^n = \{D_{i_1}, D_{i_2}, \ldots, D_{i_n}\} \quad (n \$.N),$$

そして

ｒ ’（Ｃ；、）＝ ｍｉｎ ｛ｄｉ；私はデイ。私はca=ｄｉ

最後にしましょう

C:,　(cp) におけるu 4ドル

C = Cp　4 <n (Cp) $の場合10

C^0　I0 <nの場合 (Cp)

- [rr ’ (C :,) n (C) $の場合4

そして r　４ ＜ｎ（Ｃ）＄の場合１０。

r'(C ^{0})　I0 <nの場合 (C)

他に指示がない限り、基準点Ｐは理解されるべきであり、表記から省略されるであろう。実際には、連続する点間の距離関数が評価されるPは、平均データ点間隔と比較して比較的小さいです。大部分の対Ｃ ’およびｒ’は、前の点についての対に関して操作上容易に定義することができる。

基準点Ｐに対して、新しい重み付け係数ｓ。= ｓ（d;）は次の関数から定義できます。

1　　0 <d $の場合

(1s(d) -[r;.cf 2

r' ならば、ｄｒ

3 -

ｒ ’<dの場合

この関数は、ｄ ’r’に対してｓ（d）= 0となるように、すべてのd> 0にわたつて連続微分可能であると定義されています。Ｃ ’はすべてのD; f.C’に対してd; $。ｒ ’が次のように定義されているので、C’の外側を指す重みはゼロであり、効果なしに除外することができます。したがつて、元の関数に似た動作をしますが、多数のデータポイントに使用するのがはるかに簡単な補間関数は、次のとおりです。

$$f_2(P) = \begin{cases} \left[\sum_{D_i \in C'} (s_i)^2 z_i \right] & \mathrm{I}_{dc}I, (s;) 2I\, ii/d_r 01 \text{またはすべてのD} \\ z_i & d; \text{があるDに対して}= 0\text{の場合。} \end{cases}$$

方向を含む

補間を改善するために、距離係数に加えて方向係数が重み付けを定義するのに必要であることは明らかであつた。直感的には、これは、Pからのデータ点の影響が同じ方向にあることによつて「シヤドーイング」していることを表している。各データポイントDに対する方向性重み付け項。near Pは、によつて定義された。

$$t_i = \left[\sum_{D_j \in C'} s_j[1-cos(D_iPD_j)]\right] \Big/ \left[\sum_{D_j \in C'} s_j\right].$$

角度DのコサインPDJは、内積によつて次のように評価できます。

1 (x − x;) (x − x − J) + (y − y;) (y − y − J) 1 / d。−

すべての角度0に対して、 − I − : : cos (0) $ Iであるので、O $ t; ＜2となる。他のデータポンピングDJがPからDとほぼ同じ方向にある場合、 (I-cos) 係数はゼロに近くなります。ゼロに近いです。一方、他のデータ点がD iとPの反対側にある場合、 (I-cos) 係数、したがつてt;2の近くにあります。

余弦関数は、適切さと計算の容易さの両方のために、方向の尺度として使用された。Pの近くの点は遠くの点よりもシヤドーイングにおいて重要であるはずなので、距離重み係数sJは分子と分母に含まれます。計数方向、新しい重み付け関数
w;= (s;} 2 x (I + t;) が定義されているかもしれません。
内挿関数の

I J W; Z;] I [: E W;]もしd; = /: − すべてのDに対して0;
jj (p) d;C' d . ec」
[
z; *d;があるDに対して*
= 0の場合。

勾配を決定する

重み付け関数に対する前述の修正にもかかわらず、Pについてはあるデータ点Dに十分に近いことを思い出されたい。 (d;小さい) 、s;d; _、w;と等しくなります。元の補間と同様に、d ;⁻²のように変化します。関数。前述のように、補間された表面は依然として各D iにおいてゼロ勾配を有する。この望ましくない特性を補正するために、補間された表面がD iにおいて所望の偏導関数を達成するように、近くのデータ点における関数値に増分が加えられた。まず定数Aです。そしてB。各データ点Dについて決定される。xの希望する勾配を表す

と wj sz−z) (x−x)

A; = *djec: (d [DJ、D;]) 2*
E
d −ec wj

そして

E *(zrz (y) 年)*
B; = *djec」 (d (DJ、D;]) 2*
E *W*
djec 1」

次に、距離の次元を持つパラメータvがzの全範囲に基づいて定義されます。そして望ましい斜面:

v=大井マツクスz f [maxf (A / + B /)} f。

このパラメータは、勾配項がzの最終的な内挿値に与える可能性がある最大の効果を制限します。係数0.1は、zの範囲の5分の1の等高線間隔を持つ等高線マツピングに適用する場合の任意の選択を表します。勾配項の影響は、輪郭間隔の半分に制限されます。増分11zで.P (x、y) の値を補間する際に勾配の影響を含めるため。Dごとに計算されます。e CをPの関数として

11z; = *[A;(x-x;) +* *B; (y-y;)][* $\frac{v}{v+d;J}$

PをDから直線に沿つて移動させます。そして

11zへの影響を考慮してください。−dの係数 $\frac{v}{v+;}$ だつた

それは私から単調に減少するので挿入される
dとして0。大きいdの場合、d1 ’のように動作し、0から00に増加します。したがつて、d iに関係なく、任意のデータ点Dについて、

|'1z;| OI [maxfz;} − 最小{z; 1J

望んだ通りに。また、Dで偏微分を評価します。与える

$\frac{a}{ax}$ (/1z;) = *A;そして2 ...* *(Az;)* *あい* = *B;.*
x=x; *y=y;* *x=x;* *y=y;*

D方向のy方向A;そしてＢ。Dについてzの差の分割された差の加重平均です。Cとしよう; '' = CD;-f D;}。それから

したがつて、以下で定義される内挿関数は、所望の偏導関数Ａ; そしてＢ。Dで。それ以外は/のように振る舞います $_3$•

$$f(P) = \left[\sum_{D_i \in C'} w_i(z_i + z_i) \right] / \left[\sum_{D_i \in C'} w_i \right]$$

d; = /: の場合 – すべてのDに対して0; f.C'

$$z_i$$

あるDに対してd; = 0 の場合; f.C'

計算誤差を減らす

前述の補間アルゴリズムがデジタルコンピュータで使用されるとき、丸めおよび切捨ては、PがDに非常に近いときにかなりの不正確さを引き起こす可能性がある。lim f (P) = z;なので、計算問題は
P-D;
コンピュータの精度に応じてfを設定し、f (P) = zと定義することで回避できます。f以内Dの近所。いくつかのデータ点がc近傍内にある場合 *P*のNf。(P)、それらの値は平均化されます。

補間関数は技術的に不連続ですが、不連続性は機械誤差に勾配の大きさを掛けたものと同程度です。このfによって作成されたエラー。– 近隣は、機械の不正確さによって他の場所で引き起こされたものより大きくありません。関数は離散点でのみ評価されるため、実際には問題は重要ではありません。したがって、最終的な内挿関数は次のようになります。

| *d* w> (z、+ z、)] l l w、)] d、> f.forall の場合

$$f(P) = \left[\sum_{f.C'} z_i \right] / \sum_{f.C'} - 1) \; ifd、 < f.for$$

D、f.Nf。 (P) D、f.Nf。 (P) いくらかD、

バリア効果のシミュレーション

データ点がデータ収集区域(国勢調査区域など)の中心を表す都市および地域計画において前述の補間アルゴリズムが使用される場合、2つの区域が隣接するときはいつでもそれらの間に何らかの論理的関係があると仮定される。川、鉄道、高速道路などの何らかの物理的な障壁が地域を隔てる場合、論理的な関係は弱まる可能性があります。障壁を含めることによって、ユーザは距離空間内の不連続性を特定することができる。はこの減衰をシミュレートするために計算されます。距離依存補間のため、この変更は特に簡単です。

t b [P、D;) はバリアの強度と有効距離d! と見なされます。によつて与えられます

$$d_i' = \left((d[P,D_i])^2 + (b[P,D_i])^2 \right)^{1/2}$$

この定義は一般的なものなので、障壁がない場合は *D;* P、b [P、D;] = 0およびd! = d;。ときに障壁

含まれていますd! dを置き換える必要があります。上記のすべての式で一般に、障壁を含めることは、予想されるように、補間のための「近傍」データポイントの異なるセットの選択をもたらし、そして異なる重み付けおよび勾配が確立されることになる。Pが境界を横切るときの有効距離の不連続性のために、補間された表面は障壁において不連続になる。

f (P)の性質

上で定義された補間関数 f (P)は最初に仮定された全ての明示的な要求を満たす。f (P)で定義される曲面は、不連続性が意図的に指定されている障壁を除いて、どこでも連続的に微分可能です。曲面はすべてのデータ点で必要な値をとります。デジタルコンピュータでは、内挿値の評価はかなり簡単です。データポイントにおける部分的な奪い合いは、それらが「あるべき」であるという概念、近くの既知の値からの最初に分けられた違いの平均に基づいています。最終的な内挿関数は、データ点だけでなく、どこにでも極値を持つことがあります。内挿関数の範囲は、範囲o(初期値z;)よりも両端で最大10%大きくなります。
平面functi0nは平面内のすべての点で定義されます。したがって、クロージヤがすべてのデータポイントを含む最小のポリゴンの外側に自然に外挿することができます。この多角形からの距離が大きいと、内挿値はほぼ平均になります。

PとDの間の線に垂直な長さb [P、D;]の「迂回路」を考えます。2点間の障壁の周りを移動する必要がありました。クアン

最も近い4つのデータポイントの値の。

アプリケーション

　前述の補間関数の今日までの主な用途は、そのプログラミングであつた。
FORTRAN IVとそのマッピングプログラム
SYMAPバージョンIVとVへの組み込み[1]° 以下は
SYMAPバージョンVを使用して0.2分でIBM 7094上で作成された等高線図の例。以下の座標と値を使用して4つのデータ点を指定しました。元のデータポイント値の範囲は5つの等しいレベルに分割され、最低レベルと最高レベルはすべての補間値を含むように拡張されました。特殊記号は、標準の英数字を重ね合わせることによつて高速プリンタ上で生成された。

li: .V 1　　1　2　3　4　5
==-=----=---
• • • • • • • • • • +++++++++ 000000000 eeea eet: 1 u
あなたがより多くを見つけることができます。
同期ファ　また、あなたは新しい情報源を見つけることができます。S エア88
あなたがより多くの情報を見つけることができます。
—、—、—、—、—、—、—、——型の2×2×2 8tie8888be示す。
=====================---.----------------------.--
イル

•----+---1----+----2----+----3----+----4----+----s----+----6----+----1----*
!++++++++++++++++++++++++++++++++++ •••I
!+++++++++++++++++++++++++++++++++++++ •••1
!+++++++++++++++++++++++++++++++++++++++ •••1
7- I+++++++++++++++++++++++++++++++++ ••I
++ ••+
!++ ••••••••••••••••••••••••••••••••••••1••••••••••••1
l++++++++++++++++++++++++++++++++++++++ ••!
!++++++++++++++++++++++++++++++++++++++ ••I
!++ ••••••••••••••••••••••••••••••••••••••I
!++ •••1
!++ ••1
6- 1+++++++++++++++++++2++++++++++++++++++++++ •••••••••••••••••••••••••••••••••••••••1
!++ •••••••••••••••••••••••••••••••••• I
!+++ •••••••••••••••••••••••••••••• +I
++ ••••••••••••••••••••• +++++
!+++ ••••••• +++++++I
I++ +++++++++++++++!
1+++I
I++I
5- i++ 2
!++ 01
!+++ 0000001
!++ 000000000000001
I+++ どうどう!
+++ どうどう!
!+++++++++++++++++++++++++++++++++++++++ どうどう、どうどう! I
4- I ++++++++++++++++++++++++++++ ojoijooutjoojoojoooot.1000 B81
1000 ++++++++++++++ アウチ! aee00eey01
luuuuuuuou おおうおううおうううううううう eeeeeeeeeeeeee
3UOOCUUU0uUUU0UUUU000U00UUUUUUUUUUULU000000000UU000000 8 888 808888880SB83
IOGOLUuGOULUUUuUUU000UUuu0u00000Uu00000000C0000C0 888 88880888t! 888888808I
! OOUCUUUUUGOJOUOUOOGUu0000UUOU0000000UOOL000000U e888tl 888880130e @ 88988o8801
lUuOUUUUJOuU0000UUOL10u000GUUu0UU00000000u000C0 89B 9 888€LE8 8S 1
IOOUUUUOUuG000uUOOuOUL1UUL1G0000U0000000J000000 8 088トン! 8888888886
3- 1JGOUL 、、LUuUUUU1JUOJOGuUUJUOUUOJOUOCUGOOOUJOOOQUQQUOOOU eiit! 88 88888t:;
lUCIOQ; jUUUUJUuOUOOOUJOOOCUO0000UUOtJG (JOUO 8タイトル: it 18e 3&688BI3 - &111 I
luuUUOUUUUUUUODOOOOU 1UOOOOJOUJuOUOOOOO 8ef880860
! JUUUUGuOOuUuuuOuuOuOOOOUDUUUUOUOUOO 8888e&
61 i e e a e s dt: 私は 4
IOUOUOUUuOUdUUuuOJJUuUOJOG00u0000U0U0 eaee e8aeet: li38 - S-0 1
2 - ! IOULJGUOOUOOUUUOuUJUOuOuUuUuUuU 88ee&! 18i; 88 1
whoo!
0u0000000U000000813tie eeeeeeae 808ti€80888 @ 66 -
IOGU00uu0UUUUUUJOOUL1UuUUJGuU000000U eseseeeeeeeet1ti
louooooouououooooouuoouoouoooouou e ee e ee 1
ee8eb @ e08B888
5000UUUUUOOOOOOOOOOOOOOOO eet 88 5
ioouoo33oooooooo ujuoouuguuu uuouou おお Om: 肯定! 肯定! e ee 1
1-1 uGOOUuuuuuuuuuuUJUUUUUUOUJ TF488) OUUUOOO e s e e e 10 -1
8818te すなわち
1OUULuu0JOu0J0000UOUOUO0JuUUOUUO eeでのeeeoesoooo N -1
888Ie、3 ee 388 38e n uo u - u 1 u o o
) 1

aeese 80dt: leed8eeB uo11 1

88i: l eSeeeee eee·····················1

*- --+----1----+----2----+----3----+----4----+---s----+----6-- -+----1----•

はい。

4つの不適切なスペースのuataポイントからのLFMONST ATION UFインターポレーション。 NUMdi Sは、位置とレベルを示すデータを表示します。C.UNTOUK MAP生産されたSYMAP、バージョン5、ハーバード大学で。

4つの不等間隔データポイントからの内挿のデモンストレーション

データポイントの位置と価値

データポイント番号	ロケ地タ座標 x_i	Y_i	データ値 z_i	レベル
	6.00	6.75	0.0	
2	6.80	2.25	5.0	5
3	0.80	1.13	2.5	3
4	1.90	6.00	1.5	2

記号レベルの各レベルに対応する値の範囲 範囲

	(-0.5, 1.0)
2	(l.0, 2.0)
3	(2.0, 3.0)
4	(3.0, 4.0)
5	(4.0, 5.5]

（地図上の数字はデータポイントのレベルと場所を示します。）

SYMAPバージョンVプログラムの下での地図作成のための計算時間はおおよそ地図の面積に比例し、データ点の数が増えるにつれて徐々に増加します。ＩＢＭ７０９４上でＩＯ四方のマツプＩＯを準備することは、４つのデータポイントで約０．３５分、２００のデータポイントで０．７０分、６００のデータポイントで１．３分を要する。バージョンVはIBM 360/50でも動作するようになつています。[10]

SYMAPバージョンIVおよびVの場合と同様に、図1のマツプを作成するために、実際には2つのレベルの補間が使用されました。最初に、横方向の値が3つおきに、そして下の値が2つおきに、前述のアルゴリズムによつて計算されました。それから二重線形補間が介在位置で使用された。地図の外観をわずかに変更するだけで、この2段階の手順で計算時間が70%短縮されます。バージョンIVとVはバージョンIIIよりまだ少し遅いですが、それらの柔軟性と便利さは利点を補います。

結論

前述のアルゴリズムは、わずかな修正を加えて、３次元以上の空間内のデータ点からの補間に一般化することができる。高次元空間では、デカルト計量が2次アル距離。また、スカラ積または勾配増分を含む式では、各追加次元に対して別の項が必要です。最後に、半径ｒは、全てのデータ点を含む最小多角形の面積の代わりに、最小多面体の体積から評価される。

一般に、補間法は関数が通過する点を除いて正確に正確ではありません。

装着されています。前述の機能は
そのような変数を特に考慮して開発した
人口密度、住宅状況などとして
計画および地理学の分野から。そのために
仕事、少しの正確な特性についてはほとんど知られていない
関係する変数したがつて、研究者の直感的な判断
客観的な対策がない場合は、
元空間を置き換えます。

使用される補間関数の適切性の許容可能な評価を表します。前述の補間関数は任意の選択を含むが、それは少なくともコンピユータマツピングによる面積データの分析を助ける実用的なツールを提供した。この方法がより洗練された二次元およびより高次元の補間関数の導出に役立つことが期待される。

参照

私はpスウイツツアーcmモール 再ハイトマン
海洋地形の統計解析と等高線図作成手順
プロジエクトトライデントレポート
1440464アーサーDリトルインク1964年4月

2 JFステフエンセン
補間
ウイリアムズとウイルキンズボルチモア1927第19章

3 はベレジンです N P ZHIDKOV
計算方法
アデイソン - ウエズリー出版社 レデイングミサ1965第1章第2章

4 C deBOOR
バイキユービツクスプライン補間
J 数学と物理学41 212-218 1962

5 ダウ
数値解析研究への応用を伴う等高線図の自動構築
国防ドキユメンテーシヨンセンターレポートAD 649811 1966年1月

6 wrトブラー。
地理データのための補間アルゴリズム
ミシガン大学 時代遅れ

7 **クルムベイン**
不規則な制御点間隔を持つ等高線図のトレンドサーフェス解析
jの脆弱性に関して64 823 834年1959

8 Bengtsson **ノルドベツク**
コンピユータによる等値線と等値線図の構築
スウエーデンルンド大学BIT 4 87-105 1964

9 **ああシユミツト**
SYMAP Aユーザマニユアル
トライ郡地域計画委員会 **ランシングミシガン** 1966年5月

10 *コンピユータグラフイツクマツピング'SYMAP'バージヨンV用のユーザーマニユアル*
コンピユータグラフイツクス研究室**ハーバード**大学1968年6月