

储量计算 | 对克立格法计算储量现用程序的理解和讨论

首钢勘探公司地质研究所 黄淼云

在冶金地质系统，用克立格法计算储量，现用的一套计算机程序，是武汉地质学院於崇文、蒋耀松和首钢勘探公司侯景儒，于1978年6月引进和改编的。使用121机，ALGOL-60语言。1979年初，我所王书惠由121机改至108—乙机。

这套程序分为两大部分。第一部分是计算变异函数；第二部分是建立克立格方程、计算克立格估值。前者引自美国Journal著《矿产地质统计学》。

1979年底，冶金部地质司在桂林举办“克立格法”学习班，参加学习班的同志们，用这套程序计算了九个不同矿种的矿体储量。现结合这次教学实践，对这套程序的设计思想及使用，写几点认识。

半变异函数的计算程序

区域化变量 $Z(x_i)$ ，在内蕴假设条件下，计算实验半变异函数的数学模型为：

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [Z(x_i) - Z(x_i+h)]^2 \quad (1)$$

其中， $\gamma(h)$ 为半变异函数； $N(h)$ 为对应于某个距离 h 的数据对的对数（图1）。

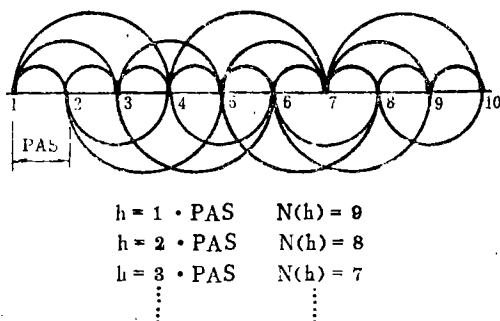


图1

$Z(x_i)$ 半变异函数的计算程序共有四个。对一维域，如果 $Z(x_i)$ 的分布是规则的，用GAMA₁程序计算；不规则的用GAMA₄程序计算。对二维域，如果 $Z(x_i)$ 的分布是规则的，用GAMA₂程序计算；不规则的用GAMA₃程序计算。

此四个程序，在程序设计上，有一些共同点，最突出的是在每一个计算点上，要求计算的所有滞后距上（或滞后数上）的两点间增量平方是“一次算”的，其值分别存入不同的存储单元。例如：滞后数为3，在点₁计算时，要算点₁与点₂、点₁与点₃、点₁与点₄三个增量平方，然后再换至点₂（图2）。

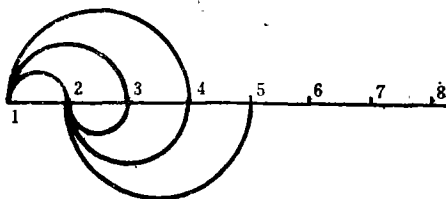


图2

1. 对GAMA₂程序的理解 如果在二维域中, $Z(x)$ 的分布满足下列图形中的任一种, 即可使用GAMA₂程序计算其半变异函数值。

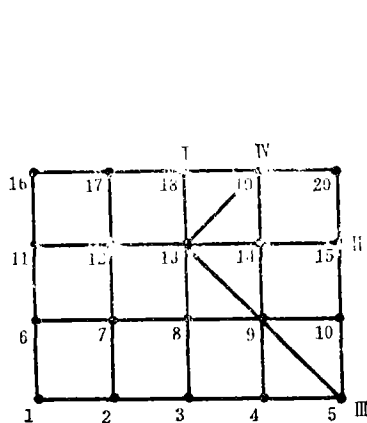


图 3

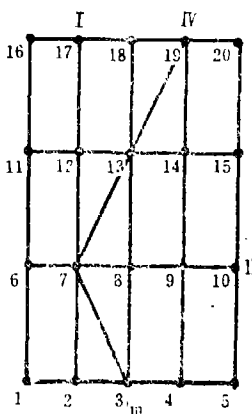


图 4

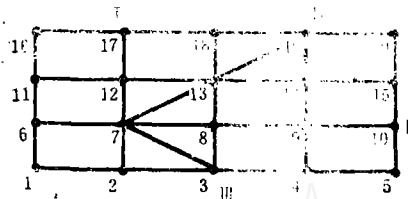


图 5

认为GAMA₂程序在设计上, 是分方向计算其半变异函数的(图3、图4、图5), 认为只要每个方向上, 区域化变量间的距离相等, 即为规则分布, 就可使用本程序。因为在每个方向上, 两点之间的距离, 均称为基本滞后距, 我们是用基本滞后距的个数, 算至最大滞后数的。在每个方向上, 参加计算的是滞后数, 而不是滞后距。因此, 只要求每个方向上点距相等, 而并不要求各方向上点距一定要相等。为此, 在作图时, 要用滞后数乘以基本滞后距, 才是所要求的滞后距。

我们印发的108—乙计算机程序中, GAMA₂程序的第80行、83行, 作如下修正(以下凡称原有程序, 均指该印发程序):

```
'CH' 0, 80, 80,
'BEGIN' SPACE(12), PUTR(IK),
...
'CH' 0, 83, 83,
PUTR(IK), PUTI(NC(IK+KD*KMAX)),
...
```

GAMA₂程序算例中“IK*PAS”栏的结果是错误的。

2. 对GAMA₃程序的讨论 原GAMA₃程序是用于计算二维不规则分布数据任意个方向上的、相同滞后距的半变异函数。即对任意个方向上基本滞后距PAS都应相同。

如果各方向上的PAS不等, 也可用此程序计算, 但要原GAMA₃程序作如下改源:

```
'CH' 1, 2, 2,
'REAL' TEST, DP, DA,
...
'CH' 1, 5, 5,
READR(TEST, DP, DA),
...
'CH' 1, 10, 10,
'ARRAY' VR(1:ND), X, Y(1:ND), ALP, PAS(1:NDI),
...
'CH' 1, 13, 13,
INPUT(VR, X, Y, ALP, PAS),
...
'DE' 1, 39, 43,
...
'AD' 1, 38,
'FOR' KD:=-1'STEP'1'UNTIL 'NDI' DO'
```

```

'BEGIN' K:=ENTIER(H/PAS(NDI)+0.5)+1,
'IF' K 'GR' KMAX'OR'ABS(H-(K-1)*PAS(NDI))
'GR'DP'THEN' 'GOTO'L2,
COSD:=(DX*CAN(KD)+DY*SAN(KD))/H,
...
'CH' 2, 35, 35,
PUTSR(' (DISTANCE-TOL='>DP),
///

```

KRI—3 D程序

前面说过，KRI—3 D程序是用于建立克立格方程，求取克立格估值的。

1. 数学方法简述 克立格法是以最小估计方差，给出对于块段平均品位的无偏线性估计量（也称克立格估计量）。

设有一组品位值 Z_α ， $\alpha = 1, 2, \dots, n$ ，则克立格估计量 Z_k 即为：

$$Z_k = \sum_{\alpha=1}^n \lambda_\alpha Z_\alpha \quad (2)$$

为使估计方差为极小，权系数 λ_α 应满足下面的克立格方程组：

$$\begin{cases} \sum_{\beta=1}^n \lambda_\beta \bar{C}(V_\alpha, V_\beta) - \mu = \bar{C}(V_\alpha, V) & \beta = 1, 2, \dots, n \\ \sum_{\alpha=1}^n \lambda_\alpha = 1 \end{cases} \quad (3)$$

μ 是拉格朗日乘数， $\bar{C}(V_\alpha, V_\beta)$ 是信息域与信息域之间的协方差平均值， $\bar{C}(V_\alpha, V)$ 是信息域与待估域之间的协方差平均值。

①克立格方程组的最终形式 整个需要估算储量的矿区，根据具体情况，划分成许多平行六面体（以下简称为块段）。设每个块段中，有一个垂直钻孔通过，其位置是随机的，取其岩芯样的品位平均值，为块段的平均品位值。即每个块段均有品位平均值，每个块段均可视作信息域。当估算每个块段的克立格估值时，要用它周围的26个块段作为信息域，来算它的估值。此时，该块段是待估域（图6）。

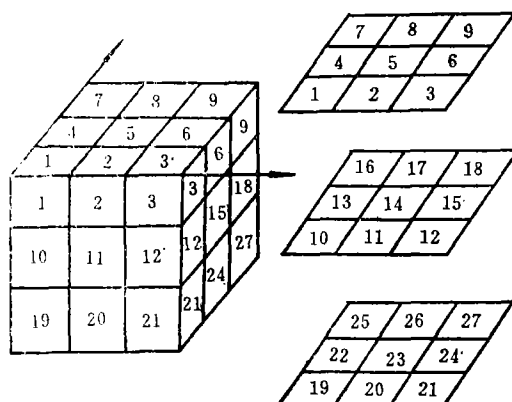


图6

图6中块段号14（简写成 V_{14} ），即为待估段B。显然，待估块段数要比信息块段数少。

设：

NX, NY, NZ 为估算矿区在坐标轴 X, Y, Z 方向上的列、行、层数，

G 为待估块段数组，

GK 为信息块段数组,
 则G与GK 数组的上、下界,可写成以下形式: $G[1:NZ, 1:NY, 1:NX]$,
 $GK[0:NZ+1, 0:NY+1, 0:NX+1]$ 。
 在这套程序中,把27个块段又分成五个域(图7)。

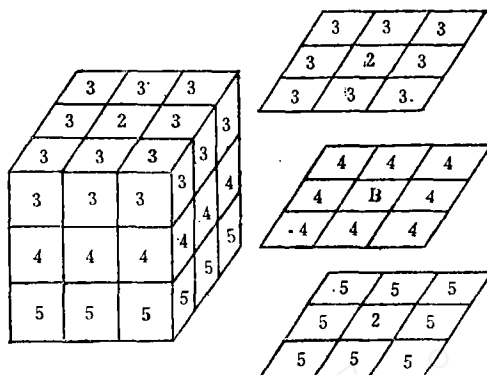


图7

域₁(D_1)为待估块段本身;
 域₂(D_2):由 V_6, V_{23} 组成;
 域₃(D_3):由 $V_1, V_2, V_3, V_4, V_6, V_7, V_8, V_9$ 组成;
 域₄(D_4):由 $V_{10}, V_{11}, V_{12}, V_{13}, V_{15}, V_{16}, V_{17}, V_{18}$ 组成;
 域₅(D_5):由 $V_{19}, V_{20}, V_{21}, V_{22}, V_{24}, V_{25}, V_{26}, V_{27}$ 组成;
 因此,克立格方程组的最终形式为:

$$\left. \begin{aligned} \bar{C}(D_1, D_1)\lambda_1 + \bar{C}(D_1, D_2)\lambda_2 + \bar{C}(D_1, D_3)\lambda_3 + \bar{C}(D_1, D_4)\lambda_4 + \bar{C}(D_1, D_5)\lambda_5 + 1 &= \bar{C}(D_1, B) \\ \bar{C}(D_2, D_1)\lambda_1 + \bar{C}(D_2, D_2)\lambda_2 + \bar{C}(D_2, D_3)\lambda_3 + \bar{C}(D_2, D_4)\lambda_4 + \bar{C}(D_2, D_5)\lambda_5 + 1 &= \bar{C}(D_2, B) \\ \bar{C}(D_3, D_1)\lambda_1 + \bar{C}(D_3, D_2)\lambda_2 + \bar{C}(D_3, D_3)\lambda_3 + \bar{C}(D_3, D_4)\lambda_4 + \bar{C}(D_3, D_5)\lambda_5 + 1 &= \bar{C}(D_3, B) \\ \bar{C}(D_4, D_1)\lambda_1 + \bar{C}(D_4, D_2)\lambda_2 + \bar{C}(D_4, D_3)\lambda_3 + \bar{C}(D_4, D_4)\lambda_4 + \bar{C}(D_4, D_5)\lambda_5 + 1 &= \bar{C}(D_4, B) \\ \bar{C}(D_5, D_1)\lambda_1 + \bar{C}(D_5, D_2)\lambda_2 + \bar{C}(D_5, D_3)\lambda_3 + \bar{C}(D_5, D_4)\lambda_4 + \bar{C}(D_5, D_5)\lambda_5 + 1 &= \bar{C}(D_5, B) \end{aligned} \right\} (4)$$

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 - \mu = 1$$

也可以写成矩阵形式:

$$\begin{bmatrix} \bar{C}(D_1, D_1) & \cdots & \bar{C}(D_1, D_5) & 1 \\ \vdots & & \vdots & \\ \bar{C}(D_5, D_1) & \cdots & \bar{C}(D_5, D_5) & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_5 \\ -\mu \end{bmatrix} = \begin{bmatrix} \bar{C}(D_1, B) \\ \vdots \\ \bar{C}(D_5, B) \\ 1 \end{bmatrix} \quad (5)$$

简写成: $[K] \cdot [\lambda] = [M_2] \quad (6)$

显然,域由块段组成,块段又认为是由离散点的集合来替代。块段由64个离散点来替代;一维的岩芯段用10个离散点来替代。

②克立格方程组等号左边系数项的计算 不包括1和0的系数阵 $[K]$,是个对称阵,求出上三角矩阵15项系数,即可完成 $[K]$ 的计算。

I. 非对角线项 $\bar{C}(D_1, D_2)$ 的计算:

此项系数是算 D_1, D_2 间协方差平均值。有垂直钻孔通过,用岩芯段来计算。即把岩芯段长分成10等分,每等分的中点,为离散点的点位,计算10对点间的协方差平均值。即:

$$\bar{C}(D_1, D_2) = \bar{C}(S_1, S_2) = \frac{1}{10 \cdot 10} \sum_{i=1}^{10} \sum_{j=1}^{10} C(x_i, x_j)$$

II. 非对角线项 $\bar{C}(D_1, D_3)$ 、 $\bar{C}(D_1, D_4)$ 、 $\bar{C}(D_1, D_5)$ 、 $\bar{C}(D_2, D_3)$ 、 $\bar{C}(D_2, D_5)$ 的计算:

此五项系数是计算两域块段间的协方差平均值的算术平均值。如:

$$\bar{C}(D_1, D_4) = \bar{C}(V_{14}, V_k) / 8; K = 10, 11, 12, 13, 15, 16, 17, 18.$$

块段用64个离散点来替代。即每个块段要分割成64个等分小块段, 其小块段的中心, 就是离散点的点位。两块段间的协方差平均值的计算公式, 即为:

$$\bar{C}(V_{14}, V_{10}) = \frac{1}{64 \cdot 64} \sum_{i=1}^{64} \sum_{j=1}^{64} C(x_i, x_j)$$

III. 非对角线项 $\bar{C}(D_3, D_4)$ 、 $\bar{C}(D_3, D_5)$ 、 $\bar{C}(D_4, D_5)$ 的计算:

此三项系数因有同一垂直钻孔通过, 故既要算岩芯段间协方差平均值, 又要算块段间协方差平均值, 如:

$$\bar{C}(D_3, D_5) = \bar{C}(S_1, S_3) / 8 + \sum_{K=1}^8 \sum_{K'=1}^8 \bar{C}(V_k, V_{k'}) / 64,$$

其中, $K = 1, 2, 3, 4, 6, 7, 8, 9$;

$K' = 19, 20, 21, 22, 24, 25, 26, 27$ 。

IV. 对角线项系数 $\bar{C}(D_1, D_1)$ 、 $\bar{C}(D_2, D_2)$ 、 $\bar{C}(D_3, D_3)$ 、 $\bar{C}(D_4, D_4)$ 、 $\bar{C}(D_5, D_5)$ 的计算:

$\bar{C}(D_1, D_1) = \bar{C}(S_1, S_1)$ 。即计算岩芯段自己对自己的协方差平均值。这是一维域重合的情况, 程序中用产生随机数的方法, 使其两两重合点间稍移动一些距离, 以避免协方差平均值系统偏高。三维域重合时, 也这样处理。

$$\bar{C}(D_2, D_2) = (\bar{C}(S_1, S_1) + \bar{C}(S_1, S_3)) / 2,$$

$$\bar{C}(D_3, D_3) = \bar{C}(D_4, D_4) = \bar{C}(D_5, D_5)$$

$$= \bar{C}(S_1, S_1) / 8 + \sum_{K=1}^8 \sum_{K'=1}^8 \bar{C}(V_k, V_{k'}) / 64,$$

其中, $K = K' = 1, 2, 3, 4, 6, 7, 8, 9$ 。

③ $[M_2]$ 的计算 此阵除去1外, 还有五项系数:

$$\bar{C}(D_1, B) = \bar{C}(B, B),$$

$$\bar{C}(D_2, B) = \bar{C}(V_1, V_{10}),$$

$$\bar{C}(D_3, B) = \bar{C}(D_1, D_3),$$

$$\bar{C}(D_4, B) = \bar{C}(D_1, D_4),$$

$$\bar{C}(D_5, B) = \bar{C}(D_1, D_5)$$

2. 程序设计思想简述 用了四个过程。函数过程RANDOM用于产生随机数, 当计算二个重合域(一维岩芯段或三维块段)时调用。函数过程COVA用于计算两点间的协方差函数值。函数过程CBAR用于计算域间(一维岩芯段或三维块段)协方差函数的平均值, 在CBAR过程体内要调用COVA过程。由离散点的点数, 决定对COVA套用的次数。对以上三个过程的调用, 就可以求得 $[K]$ 及 $[M_2]$, 建立克立格方程。LA DST过程用于解克立格方程, 求 λ_i 及 μ 。解得 λ_i , 就可以算各个块段的克立格估值 Z_k 了。

3. 对函数过程COVA的讨论 上面说过, 此函数过程是求两点间的协方差值。它是先求两点间的半变异函数值, 然后由半变异函数与协方差函数之间的关系, 求其协方差值的。

众所周知, 如果半变异函数在原点处是连续的, 则半变异函数与协方差函数之间的关系, 可由图8来表示。

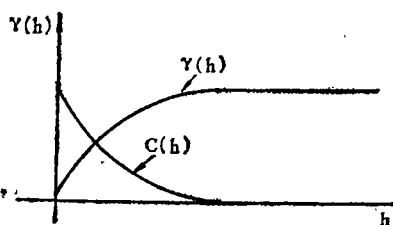


图 8

图 8 中, $\gamma(h)$ 表示半变异函数曲线,

$C(h)$ 表示协方差函数曲线。

如果半变异函数在原点处不连续, 则有块金效应。那么, 两函数之间的关系, 是否可用图 9 来表示?

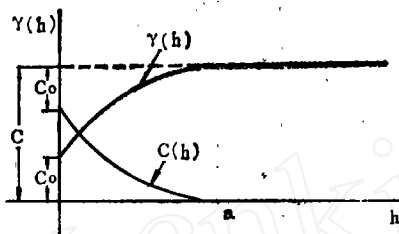


图 9

在图 9 中 C_0 为块金常数, C 为基台值, a 为变程。

因此, 在不同距离上, 其半变异函数的计算公式应为:

$$\gamma(h) = \begin{cases} C_0 + C & h > a \\ \text{选用不同模型的公式算} & 0 < h \leq a \\ 0 & h = 0 \end{cases} \quad (7)$$

同理, 在不同距离上协方差函数的计算公式应为:

$$C(h) = \begin{cases} 0 & h > a \\ \text{选用不同模型的公式算} & 0 < h \leq a \\ C_0 + C & h = 0 \end{cases} \quad (8)$$

为此, 对 KR I—3 D 程序中, 'PAGE' 1, 第 21 行至第 41 行的函数过程 COVA, 应作如下修正:

```
'REAL' 'PROCEDURE' COVA(NST, HX, HY, HZ, CO, C, AA, CA),
'VALUE' NST, HX, HY, HZ, CO, 'INTEGER' NST,
'REAL' HX, HY, HZ, CO, 'ARRAY' C, AA, CA,
'BEGIN' 'INTECER' IS, 'REAL' COV, H, DX, DY, DZ,
COV: = 0.0,
H: = SQRT(HX * HX + HY * HY + HZ * HZ),
'IF' H / LQ / ZZZ 'THEN'
'BEGIN' COV: = CO, NN: = NN + 1,
'END', MM: = MM + 1,
'FOR' IS: = 1 'STEP' 1 'UNTIL' NST 'DO'
'BEGIN'
DX: = HX * CA(IS, 1, 1) + HY * CA(IS, 1, 2) + HZ * CA(IS, 1, 3),
DY: = HX * CA(IS, 2, 1) + HY * CA(IS, 2, 2) + HZ * CA(IS, 2, 3),
DZ: = HX * CA(IS, 3, 1) + HY * CA(IS, 3, 2) + HZ * CA(IS, 3, 3),
H: = SQRT(DX * DX + DY * DY + DZ * DZ),
'IF' AA(IS) / IS / 0 'THEN' COV: = COV + C(IS) * EXP(H / AA(IS))
'ELSE' 'IF' AA(IS) / GR / 0 'AND' H / LS / AA(IS)
'THEN' CoV: = COV + C(IS) * (1 - 1.5 * H / AA(IS)
+ 0.5 * H * H * H / (AA(IS) * AA(IS) * AA(IS))),
'END',
COVA: = COV
'END'
```