

3.5 網格點的生成方法(Grid Generation Method)

求區域內之結構性網格點，較常用的兩種方法是代數格點生成法(Algebraic Grid Generation)與解偏微分法(Partial Differential Equations Method)。

3.5.1 代數格點生成法(Algebraic Grid Generation):

代數格點生成法，架構在內插法理論上，一般來說有下列幾種的內插方式：

- (1) Linear Interpolation: P_0, P_1 間直接以直線連接。



圖 3-11 Linear Interpolation

- (2) Lagrange Interpolation: P_0, P_1, P_2, P_3 為控制點。



圖 3-12 Lagrange Interpolation

- (3) Hermit Interpolation: P_0, P_1 與斜率 P_1', P_0' 為控制點。

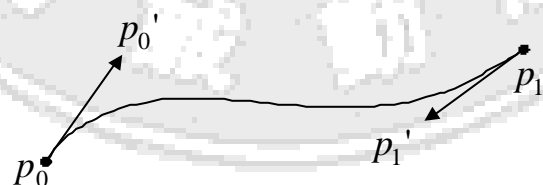


圖 3-13 Hermit Interpolation

- (4) Cubic Spline Interpolation:

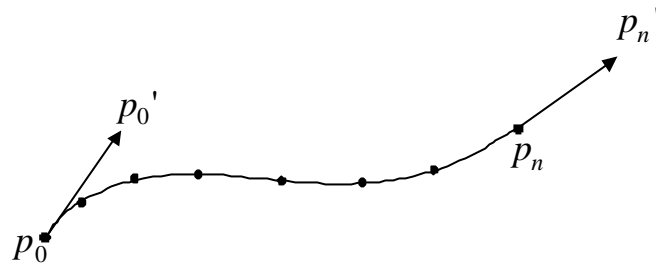


圖 3-14 Cubic Spline Interpolation

(5) Bazier Interpolation: 以控制多邊形(Control Polygon)控制曲線，其中 P_1 、 P_2 分別在斜率 P_0' 、 P_3' 的 $\frac{1}{3}$ 處。

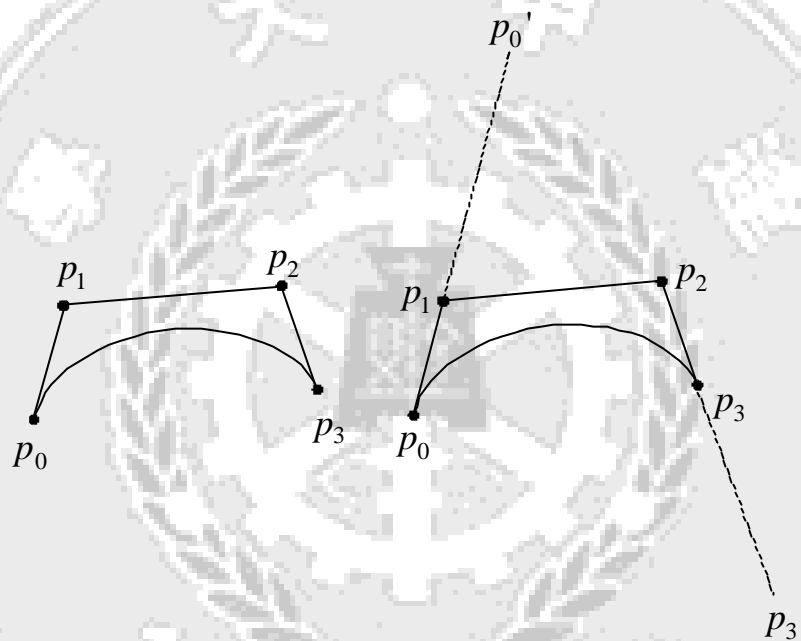


圖 3-15 Bazier Interpolation

(6) B-spline Interpolation: 結合 Cubic Spline 曲線與 Bazier 曲線的形式。

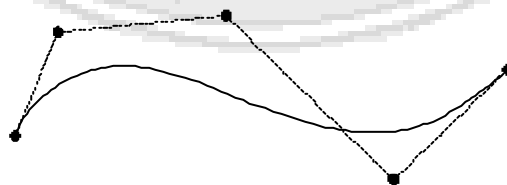


圖 3-16 B-splines Interpolation

代數格點生成法又可分為單方向內插法(Unidirectional Interpolation)與多方向內插法(Multidirectional Interpolation)兩種

1. 單方向內插法(Unidirectional Interpolation):

若只考慮一個方向上邊界格點的對應關係，於兩個對邊上各對應的端點間以曲線連接，於此曲線上決定格點的分佈。然後再於另一組對應方向上以曲線連接已經定義的格點，而產生完整的網格線，則稱為單方向內插法。以 Lagrange Interpolation 為例：

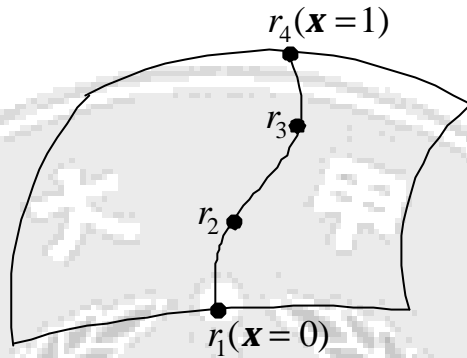


圖 3-17 Lagrange Interpolation

$$\tilde{r}(\mathbf{x}) = \sum_{n=1}^4 \tilde{f}_n(\mathbf{x}) \tilde{r}(\mathbf{x}_n) \quad (3-15)$$

$$\tilde{f}_n(\mathbf{x}) = \prod_{\substack{l=1 \\ l \neq n}}^4 \frac{(\mathbf{x} - \mathbf{x}_l)}{(\mathbf{x}_n - \mathbf{x}_l)}$$

Let $\tilde{r}(\mathbf{x}) = a\mathbf{x}^3 + b\mathbf{x}^2 + c\mathbf{x} + d$

$$\tilde{r}(0) = d = \tilde{r}_{\sim 1}$$

$$\tilde{r}(\mathbf{x}_2) = a\mathbf{x}_2^3 + b\mathbf{x}_2^2 + c\mathbf{x}_2 + d = \tilde{r}_{\sim 2}$$

$$\tilde{r}(\mathbf{x}_3) = a\mathbf{x}_3^3 + b\mathbf{x}_3^2 + c\mathbf{x}_3 + d = \tilde{r}_{\sim 3}$$

$$\tilde{r}(1) = a + b + c + d = \tilde{r}_{\sim 4}$$

(3-16)

即
$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ \mathbf{x}_2^3 & \mathbf{x}_2^2 & \mathbf{x}_2 & 1 \\ \mathbf{x}_3^3 & \mathbf{x}_3^2 & \mathbf{x}_3 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \tilde{r}_{\sim 1} \\ \tilde{r}_{\sim 2} \\ \tilde{r}_{\sim 3} \\ \tilde{r}_{\sim 4} \end{bmatrix}$$

因此可求得

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ \mathbf{x}_2^3 & \mathbf{x}_2^2 & \mathbf{x}_2 & 1 \\ \mathbf{x}_3^3 & \mathbf{x}_3^2 & \mathbf{x}_3 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} r_{\sim 1} \\ r_{\sim 2} \\ r_{\sim 3} \\ r_{\sim 4} \end{bmatrix} \quad (3-17)$$

2. 多方向內插法(Multidirectional Interpolation):

多方向內插法 (Multidirectional Interpolation, 又稱 Transfinite Interpolation), 則利用張量乘積的觀念, 同時產生二或三個方向的格線。其基本觀念如下:

(1) 投影因子(Projector) P_x , P_h , P_z :

表示邊界上的三組相對應面上分別對內作單方向內插, 此情況中, r 與這兩個面吻合(Match)。

$$\begin{aligned} P_x(r) &= \sum_{n=1}^N f_n(\mathbf{x}) r(\mathbf{x}_n, \mathbf{h}, \mathbf{z}) \\ P_h(r) &= \sum_{m=1}^M y_m(\mathbf{h}) r(\mathbf{x}, \mathbf{h}_m, \mathbf{z}) \\ P_z(r) &= \sum_{l=1}^L q_l(\mathbf{z}) r(\mathbf{x}, \mathbf{h}, \mathbf{z}_l) \end{aligned} \quad (3-18)$$

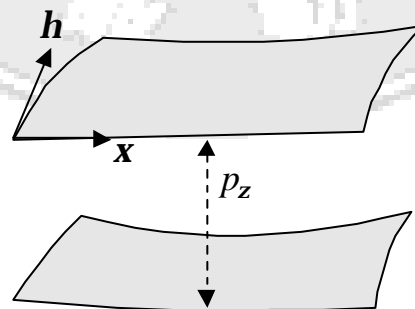


圖 3-18 投影因子 P_x

(2) 雙投影乘積因子(Double Product Projector) $P_x P_h$

沿著 \mathbf{x} 和 \mathbf{h} 為常數時的四條邊線作雙方向的內插，其與四條邊線吻合。

$$\begin{aligned}
 & P_x P_h(r) \\
 &= P_x(P_h(r)) \\
 &= \sum_{n=1}^N f_n(\mathbf{x}) \left[\sum_{m=1}^M y_m(\mathbf{h}) r(\mathbf{x}_n, \mathbf{h}_m, \mathbf{z}) \right] \\
 &= \sum_{n=1}^N \sum_{m=1}^M f_n(\mathbf{x}) y_m(\mathbf{h}) r(\mathbf{x}_n, \mathbf{h}_m, \mathbf{z})
 \end{aligned} \tag{3-19}$$

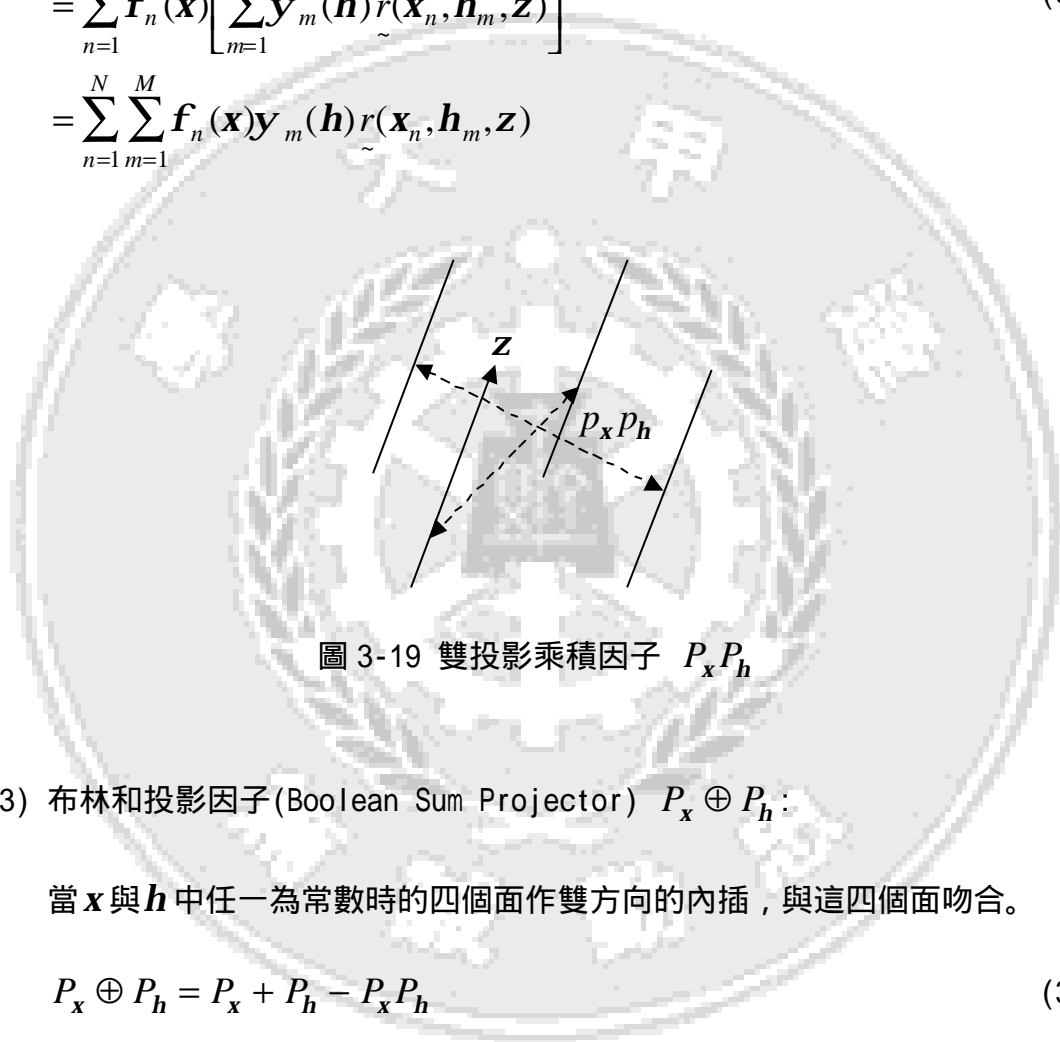


圖 3-19 雙投影乘積因子 $P_x P_h$

(3) 布林和投影因子(Boolean Sum Projector) $P_x \oplus P_h$:

當 \mathbf{x} 與 \mathbf{h} 中任一為常數時的四個面作雙方向的內插，與這四個面吻合。

$$P_x \oplus P_h = P_x + P_h - P_x P_h \tag{3-20}$$

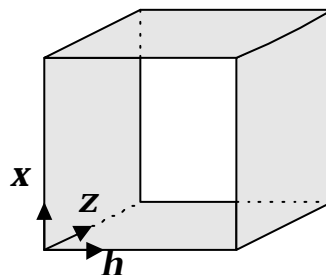


圖 3-20 布林和投影因子 $P_x \oplus P_h$

(4) 布林和投影因子(Boolean Sum Projector) $P_x P_h \oplus P_z$:

與 x 與 h 常數的四條邊線及 z 為常數的兩個面吻合，作三方向的內插。

$$P_x P_h \oplus P_z = P_x P_h + P_z - P_x P_h P_z \quad (3-21)$$

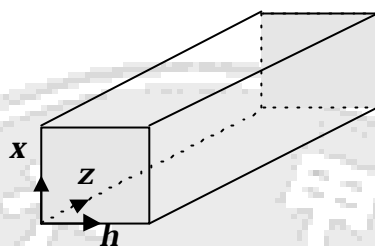


圖 3-21 布林和投影因子 $P_x P_h \oplus P_z$

(5) 布林和投影因子(Boolean Sum Projector) $P_x P_h \oplus P_h P_z \oplus P_z P_x$:

作三方向內插， r 與 12 條邊線吻合。

$$P_x P_h \oplus P_h P_z \oplus P_z P_x = P_x P_h + P_h P_z + P_z P_x - 2P_x P_h P_z \quad (3-22)$$

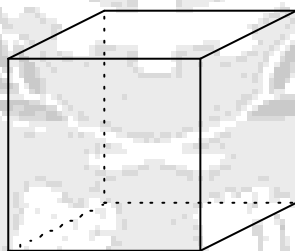


圖 3-22 布林和投影因子 $P_x P_h \oplus P_h P_z \oplus P_z P_x$

(6) 參乘積投影因子(Triple Product Projector) $P_x P_h P_z$:

從 8 個頂點向內作內插。

$$\begin{aligned}
& P_x P_h P_z(r) \\
&= P_x(P_h P_z(r)) \\
&= \sum_{n=1}^N \mathbf{f}_n(\mathbf{x}) \left[\sum_{m=1}^M \sum_{l=1}^L \mathbf{y}_m(\mathbf{h}) \mathbf{q}_l(\mathbf{z}) r(\mathbf{x}_n, \mathbf{h}_m, \mathbf{z}_l) \right] \\
&= \sum_{n=1}^N \sum_{m=1}^M \sum_{l=1}^L \mathbf{f}_n(\mathbf{x}) \mathbf{y}_m(\mathbf{h}) \mathbf{q}_l(\mathbf{z}) r(\mathbf{x}_n, \mathbf{h}_m, \mathbf{z}_l)
\end{aligned} \tag{3-23}$$

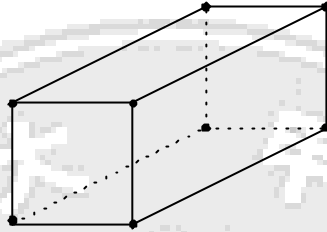


圖 3-23 參乘積投影因子 $P_x P_h P_z$

(7) 布林和投影因子(Boolean Sum Projector) $P_x \oplus P_h \oplus P_z$:

與全部的邊界吻合。

$$P_x \oplus P_h \oplus P_z = P_x + P_h + P_z - P_x P_h - P_h P_z - P_z P_x + P_x P_h P_z \tag{3-24}$$

以 Transfinite Lagrange Interpolation 為例:

$$\begin{aligned}
& p_x(r(\mathbf{x}, \mathbf{h}, \mathbf{z})) \\
&= \mathbf{f}_1(\mathbf{x}) r(0, \mathbf{h}, \mathbf{z}) + \mathbf{f}_2(\mathbf{x}) r(\mathbf{x}_2, \mathbf{h}, \mathbf{z}) + \mathbf{f}_3(\mathbf{x}) r(\mathbf{x}_3, \mathbf{h}, \mathbf{z}) + \mathbf{f}_4(\mathbf{x}) r(1, \mathbf{h}, \mathbf{z})
\end{aligned}$$

$$\begin{aligned}
& p_h(r(\mathbf{x}, \mathbf{h}, \mathbf{z})) \\
&= \mathbf{y}_1(\mathbf{h}) r(\mathbf{x}, 0, \mathbf{z}) + \mathbf{y}_2(\mathbf{h}) r(\mathbf{x}, \mathbf{h}_2, \mathbf{z}) + \mathbf{y}_3(\mathbf{h}) r(\mathbf{x}, \mathbf{h}_3, \mathbf{z}) + \mathbf{y}_4(\mathbf{h}) r(\mathbf{x}, 1, \mathbf{z})
\end{aligned}$$

$$\begin{aligned}
& p_z(r(\mathbf{x}, \mathbf{h}, \mathbf{z})) \\
&= \mathbf{q}_1(\mathbf{z}) r(\mathbf{x}, \mathbf{h}, 0) + \mathbf{q}_2(\mathbf{z}) r(\mathbf{x}, \mathbf{h}, \mathbf{z}_2) + \mathbf{q}_3(\mathbf{z}) r(\mathbf{x}, \mathbf{h}, \mathbf{z}_3) + \mathbf{q}_4(\mathbf{z}) r(\mathbf{x}, \mathbf{h}, 1)
\end{aligned}$$

$$\begin{aligned}
& p_x p_h(r(\mathbf{x}, \mathbf{h}, \mathbf{z})) \\
&= p_x(p_h(r(\mathbf{x}, \mathbf{h}, \mathbf{z}))) \\
&= f_1(\mathbf{x})[y_1(\mathbf{h})r(0,0,\mathbf{z}) + y_2(\mathbf{h})r(0,\mathbf{h}_2,\mathbf{z}) + y_3(\mathbf{h})r(0,\mathbf{h}_3,\mathbf{z}) + y_4(\mathbf{h})r(0,1,\mathbf{z})] \\
&+ f_2(\mathbf{x})[y_1(\mathbf{h})r(\mathbf{x}_2,0,\mathbf{z}) + y_2(\mathbf{h})r(\mathbf{x}_2,\mathbf{h}_2,\mathbf{z}) + y_3(\mathbf{h})r(\mathbf{x}_2,\mathbf{h}_3,\mathbf{z}) + y_4(\mathbf{h})r(\mathbf{x}_2,1,\mathbf{z})] \\
&+ f_3(\mathbf{x})[y_1(\mathbf{h})r(\mathbf{x}_3,0,\mathbf{z}) + y_2(\mathbf{h})r(\mathbf{x}_3,\mathbf{h}_2,\mathbf{z}) + y_3(\mathbf{h})r(\mathbf{x}_3,\mathbf{h}_3,\mathbf{z}) + y_4(\mathbf{h})r(\mathbf{x}_3,1,\mathbf{z})] \\
&+ f_4(\mathbf{x})[y_1(\mathbf{h})r(1,0,\mathbf{z}) + y_2(\mathbf{h})r(1,\mathbf{h}_2,\mathbf{z}) + y_3(\mathbf{h})r(1,\mathbf{h}_3,\mathbf{z}) + y_4(\mathbf{h})r(1,1,\mathbf{z})]
\end{aligned}$$

同理 $p_h p_z(r(\mathbf{x}, \mathbf{h}, \mathbf{z}))$ 及 $p_z p_x(r(\mathbf{x}, \mathbf{h}, \mathbf{z}))$ 亦可求得。

$$\begin{aligned}
& p_x p_h p_z(r(\mathbf{x}, \mathbf{h}, \mathbf{z})) \\
&= p_x(p_h p_z(r(\mathbf{x}, \mathbf{h}, \mathbf{z}))) \\
&= f_1(\mathbf{x})\{y_1(\mathbf{h})[q_1 r(0,0,0) + q_2(\mathbf{h})r(0,0,z_2) + q_3(\mathbf{h})r(0,0,z_3) + q_4(\mathbf{h})r(0,0,1)] \\
&+ y_2(\mathbf{h})[q_1 r(0,\mathbf{h}_2,\mathbf{z}) + q_2(\mathbf{h})r(0,\mathbf{h}_2,\mathbf{z}) + q_3(\mathbf{h})r(0,\mathbf{h}_2,\mathbf{z}) + q_4(\mathbf{h})r(0,\mathbf{h}_2,1)] \\
&+ y_3(\mathbf{h})[q_1 r(0,\mathbf{h}_3,\mathbf{z}) + q_2(\mathbf{h})r(0,\mathbf{h}_3,\mathbf{z}) + q_3(\mathbf{h})r(0,\mathbf{h}_3,\mathbf{z}) + q_4(\mathbf{h})r(0,\mathbf{h}_3,1)] \\
&+ y_4(\mathbf{h})[q_1 r(0,1,\mathbf{z}) + q_2(\mathbf{h})r(0,1,\mathbf{z}) + q_3(\mathbf{h})r(0,1,\mathbf{z}) + q_4(\mathbf{h})r(0,1,1)]\} \\
&+ f_2(\mathbf{x})\{y_1(\mathbf{h})[q_1 r(\mathbf{x}_2,0,0) + q_2(\mathbf{h})r(\mathbf{x}_2,0,z_2) + q_3(\mathbf{h})r(\mathbf{x}_2,0,z_3) + q_4(\mathbf{h})r(\mathbf{x}_2,0,1)] \\
&+ y_2(\mathbf{h})[q_1 r(\mathbf{x}_2,\mathbf{h}_2,\mathbf{z}) + q_2(\mathbf{h})r(\mathbf{x}_2,\mathbf{h}_2,\mathbf{z}) + q_3(\mathbf{h})r(\mathbf{x}_2,\mathbf{h}_2,\mathbf{z}) + q_4(\mathbf{h})r(\mathbf{x}_2,\mathbf{h}_2,1)] \\
&+ y_3(\mathbf{h})[q_1 r(\mathbf{x}_2,\mathbf{h}_3,\mathbf{z}) + q_2(\mathbf{h})r(\mathbf{x}_2,\mathbf{h}_3,\mathbf{z}) + q_3(\mathbf{h})r(\mathbf{x}_2,\mathbf{h}_3,\mathbf{z}) + q_4(\mathbf{h})r(\mathbf{x}_2,\mathbf{h}_3,1)] \\
&+ y_4(\mathbf{h})[q_1 r(\mathbf{x}_2,1,\mathbf{z}) + q_2(\mathbf{h})r(\mathbf{x}_2,1,\mathbf{z}) + q_3(\mathbf{h})r(\mathbf{x}_2,1,\mathbf{z}) + q_4(\mathbf{h})r(\mathbf{x}_2,1,1)]\} \\
&+ f_3(\mathbf{x})\{y_1(\mathbf{h})[q_1 r(\mathbf{x}_3,0,0) + q_2(\mathbf{h})r(\mathbf{x}_3,0,z_2) + q_3(\mathbf{h})r(\mathbf{x}_3,0,z_3) + q_4(\mathbf{h})r(\mathbf{x}_3,0,1)] \\
&+ y_2(\mathbf{h})[q_1 r(\mathbf{x}_3,\mathbf{h}_2,\mathbf{z}) + q_2(\mathbf{h})r(\mathbf{x}_3,\mathbf{h}_2,\mathbf{z}) + q_3(\mathbf{h})r(\mathbf{x}_3,\mathbf{h}_2,\mathbf{z}) + q_4(\mathbf{h})r(\mathbf{x}_3,\mathbf{h}_2,1)] \\
&+ y_3(\mathbf{h})[q_1 r(\mathbf{x}_3,\mathbf{h}_3,\mathbf{z}) + q_2(\mathbf{h})r(\mathbf{x}_3,\mathbf{h}_3,\mathbf{z}) + q_3(\mathbf{h})r(\mathbf{x}_3,\mathbf{h}_3,\mathbf{z}) + q_4(\mathbf{h})r(\mathbf{x}_3,\mathbf{h}_3,1)] \\
&+ y_4(\mathbf{h})[q_1 r(\mathbf{x}_3,1,\mathbf{z}) + q_2(\mathbf{h})r(\mathbf{x}_3,1,\mathbf{z}) + q_3(\mathbf{h})r(\mathbf{x}_3,1,\mathbf{z}) + q_4(\mathbf{h})r(\mathbf{x}_3,1,1)]\} \\
&+ f_4(\mathbf{x})\{y_1(\mathbf{h})[q_1 r(1,0,0) + q_2(\mathbf{h})r(1,0,z_2) + q_3(\mathbf{h})r(1,0,z_3) + q_4(\mathbf{h})r(1,0,1)] \\
&+ y_2(\mathbf{h})[q_1 r(1,\mathbf{h}_2,\mathbf{z}) + q_2(\mathbf{h})r(1,\mathbf{h}_2,\mathbf{z}) + q_3(\mathbf{h})r(1,\mathbf{h}_2,\mathbf{z}) + q_4(\mathbf{h})r(1,\mathbf{h}_2,1)] \\
&+ y_3(\mathbf{h})[q_1 r(1,\mathbf{h}_3,\mathbf{z}) + q_2(\mathbf{h})r(1,\mathbf{h}_3,\mathbf{z}) + q_3(\mathbf{h})r(1,\mathbf{h}_3,\mathbf{z}) + q_4(\mathbf{h})r(1,\mathbf{h}_3,1)] \\
&+ y_4(\mathbf{h})[q_1 r(1,1,\mathbf{z}) + q_2(\mathbf{h})r(1,1,\mathbf{z}) + q_3(\mathbf{h})r(1,1,\mathbf{z}) + q_4(\mathbf{h})r(1,1,1)]\}
\end{aligned}$$

代數法是最簡單的格點生成法，其有速度快、亦於控制格點分佈的優點。但是網格線不平滑、不正交、且可能重疊等是此方法的缺點。