

## 1. GİRİŞ

Jeoistatistiksel benzetimin amacı, mevcut veriler gibi aynı değerlere, aynı histograma ve aynı variograma sahip olacak şekilde veri üretmektir. Eğer yalnızca histogram ve variogram üretiliyorsa yöntem koşulsuz, bunlara ek olarak ölçüm yapılan noktalardaki veriler de üretiliyorsa koşullu benzetim adını alır. Geoistatistiksel benzetim, uzaklığa bağlı değişkenliğin ve belirsizliğin ön plana çıktığı durumlarda sıklıkla kullanılır. Bu gibi durumlar madencilikte, tenör-tonaj eğrilerinin kestiriminden üretim planlamasına, paçal yapmadan cevher hazırlama tesislerine sabit kalitede cevher beslemesine kadar uzanan geniş bir alanda ortaya çıkar.

Tenör, kalınlık gibi değişkenlerin geoistatistiksel benzetimlerini gerçekleştirmek amacıyla literatürde çeşitli yöntemler geliştirilmiştir. Bunlar arasında dönen bantlar yöntemi (Matheron, 1973), matris çarpımı yöntemi (Davis, 1987a) ve ardışık (sequential) yöntemler (Deutsch ve Journel, 1998) sayılabilir. Bu yöntemlerden LU matris ayrıştırma tekniği, tez çalışmasının ana konusunu oluşturmaktadır.

LU matris ayrıştırma tekniği, kovaryans matrisi (**C**) nin alt ve üst üçgen matrisler şeklinde parçalanmasına (**C=LU**) dayanmakta, benzetim değerleri alt üçgen matrisin bağımsız rastlantı sayılardan oluşan bir vektörle çarpılmasından üretilmektedir. Benzetim ve koşullandırmanın aynı anda gerçekleştirilmesi, düzenli ya da düzensiz her türlü grid sistemine uygulanabilmesi, kesin (exact) benzetimler üretmesi yöntemin en önemli üstünlüklerini oluşturmaktadır. Bununla birlikte çok sayıda matematiksel işlem gerektirdiğinden yavaş sonuç vermesi ve ayrıca 1000 den daha düşük noktada benzetime olanak tanınması yöntemin zayıf yanlarını oluşturmaktadır. Benzetim yapılacak nokta sayısının sınırlı olması, kuramsal nedenlerden çok bilgisayarların yetersiz kapasitelerinden kaynaklanmaktadır.

Literatürde LU tekniğinin iyileştirilmesi ile ilgili çok sayıda kuramsal çalışma yapılmıştır. Örneğin Davis (1987b), **L** alt üçgen matrisini, **C** kovaryans matrisinin karekökü (**B**) ile değiştirmeyi ve bu **B** matrisini minimax polinom yaklaşımı ile hesaplamayı önermiştir. Alabert (1987), LU yönteminde birbiri üzerine geçen pencerelerin kullanılmasını teklif etmiştir. Dietrich ve Newsam (1995), **B** matrisini Chebyshev matris polinom yaklaşımı ile hesaplayan bir yöntem geliştirmiştir. Dowd

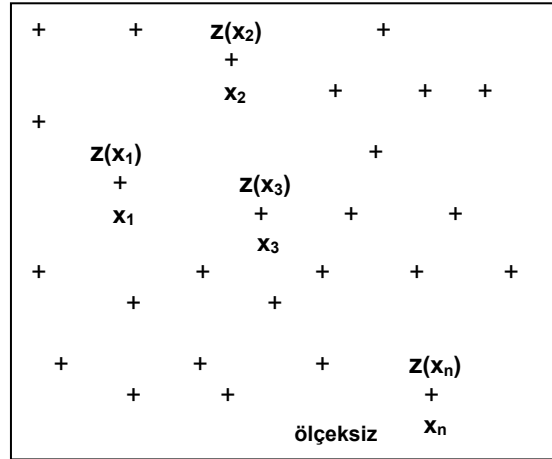
ve Saraç (1993), LU için üst sınırı artıran ring ayrıştırma tekniğini kullanmışlardır. Vargas-Guzman ve Dimitrakopoulos (2002),  $L$  matrisini koşullu kovaryans matrisler kullanarak kolonlar şeklinde kısımlara ayıran bir yaklaşım geliştirmişlerdir.  $L$  matrisine yaklaşmak amacıyla yapılan bu çalışmalar yanında bağımsız rastlantı sayıların üretimine yönelik çalışmalarda yapılmıştır. Örneğin Bourgault (1997), LU tekniği ile koşullu benzetimleri üretmede yaygın olarak kullanılan Gauss (normal) dağılımı yerine tekdüze (uniform) ve iki kutuplu (dipole) dağılımları gözönüne almış ve bunların benzetimler üzerindeki etkisini incelemiştir. De Laco ve Palma (2002) ise rastlantı sayıların üretiminde Acorn, Marsaglia ve optimize edilmiş Marsaglia gibi farklı üreteç (generator) lerin benzetimleri ne ölçüde etkilediğini araştırmışlardır.

Kovaryans matrisinin ayrıştırılmasında klasik olarak kullanılan teknik, Cholesky ayrıştırma tekniğidir. Bununla birlikte bakışlı (simetrik) ve izgesel (spectral) ayrıştırma teknikleri de vardır ve bu tezin amacı, Cholesky, bakışlı ve izgesel ayrıştırma tekniklerinin LU ile üretilen benzetimler üzerindeki performanslarını incelemektir. Bu amaçla bir benzetim deneyi düzenlenmiş ve dört farklı yapısal uzaklık ( $a=600, 1000, 1600$  ve  $2000$ ) gözönüne alınarak her bir ayrıştırma tekniği için 100'er adet koşullu ve koşulsuz benzetim yapılmıştır. Daha sonra bu benzetimler, variogram üretimi, deneysel variogramların model variogramlara yaklaşma hızı ve bağımsızlık testi gibi ölçütler kullanılarak değerlendirilmiştir.

Çalışmanın ikinci bölümünde koşullu ve koşulsuz LU benzetim yönteminin kuramsal temelleri açıklanmıştır. Ayrıca her bir ayrıştırma yöntemi için nasıl benzetim yapıldığına ilişkin örnekler verilmiştir. Üçüncü bölümde farklı ayrıştırma yöntemlerinin benzetim sonuçları üzerindeki etkilerini görmek amacıyla hazırlanan benzetim deneyi ve performans ölçütleri açıklanmıştır. Dördüncü bölümde benzetim deneyi ve sonuçları değerlendirilmiştir. Beşinci ve son bölümde sonuçlar verilmiştir.

## 2. KURAMSAL ÇERÇEVE

Jeoistatistikte tenör ve kalınlık gibi bölgesel değişkenlerin aldığı değerler, durağan rastlantı fonksiyonu  $Z^{(1)}$  den çekilmiş bir örneklem olarak modellenir. Bu model altında bölgesel değişkenin  $x_\alpha, \alpha = 1, \dots, n$  noktada örneklendiği ve bu noktalardaki değerlerin  $z(x_\alpha), \alpha = 1, \dots, n$  olduğu varsayılır (Şekil 2.1).



Şekil 2.1. Bölgesel değişkenin örneklendiği lokasyonlar

Genelde  $z(x_\alpha), \alpha = 1, \dots, n$  verileri, arama ve geliştirme amaçlı yapılan sondajlardan elde edilip daha sonra fizibilite ve üretim planlaması amacıyla kullanılır. Bu aşamada yatağın ilgili değişken için her bir noktadaki değerinin bilindiği bir modelini çıkarmak gerekir.

Kriging, bu amaçla kullanılacak lokal bir kestirim tekniğidir. Bununla birlikte kriging, verilerin ağırlıklı ortalamasını alarak kestirim yaptığı için yumuşatılmış (smoothed) yani düşük değerlerin yüksek, yüksek değerlerin ise düşük kestirildiği değerler üretir. Bu tür yumuşatılmış değerlerin üretim planlaması amacıyla kullanılması sakıncalıdır.

<sup>1</sup> Bu çalışmada rastlantı fonksiyonları ya da rastlantı değişkenleri büyük harflerle , bunların aldığı değerler ise küçük harflerle gösterilecektir.

Çünkü gerçekte seçimli madencilik yapmaya elverişli olmayan bir yatak, krigingle kestirilmiş değerler dikkate alındığında elverişli bir hale gelebilir. Bu bakımdan yatakların modellenmesi sırasında kriging tekniğinin bu yumuşatıcı etkisini düzeltmek gerekir.

Yumuşatılmış değerler daha dar bir aralıkta değişim gösterdiğinden kestirim değerlerinin varyansı, gerçek değerlerin varyansından oldukça düşüktür. Bu iki varyans arasındaki fark, kestirim yapılan noktadaki hata varyansına eşittir:

$$\text{Var}[Z(x)] - \text{Var}[Z^*(x)] = \text{Var}[H(x)] \quad (2.1)$$

(2.1) eşitliğinde  $\text{Var}[Z^*(x)]$ ;  $x$  noktasındaki  $Z^*(x)$  kestiricisinin varyansını,  $\text{Var}[Z(x)]$ ; bu noktadaki değişkenin (gerçek değerinin) varyansını göstermektedir.  $\text{Var}[E(x)]$  ise  $H(x) = Z(x) - Z^*(x)$  hatasının varyansı olup kestirim yöntemleri tarafından üretilmeyen varyanstır. Bu noktada (2.1) eşitliğinin yalnızca basit kriging sistemi kullanıldığında geçerli olduğunu ifade etmek gerekir.

Dönen bantlar yöntemi, ardışık normal benzetim ve matris ayrıştırma teknikleri gibi bir çok benzetim yordamı (algoritması), kestirim tekniklerinin yol açtığı hatanın benzetimine ve dolayısıyla hata varyansının, kestiricinin varyansına eklenmesine dayanır:

$$Z_S(x) = Z^*(x) + H_S(x) \quad (2.2)$$

(2.2) eşitliğinde  $Z_S(x)$ ;  $x$  noktasındaki benzetimi,  $H_S(x)$  ise bu noktadaki benzetilen hatayı göstermektedir. Bu yordamlarda benzetim yapılan bileşen (hata), aynı dağılıma sahip, bağımsız standart normal değerlerin doğrusal bir bileşimi olarak ifade edilir:

$$H_S(x) = \sum_{\alpha} \omega_{\alpha} Y(x_{\alpha}) \quad (2.3)$$

(2.3) eşitliğinde  $Y(x_{\alpha})$ ; standart normal dağılımdan çekilmiş değerleri göstermektedir. Doğrusal bileşimdeki  $\omega_{\alpha}$  ağırlıkları; benzetim değerleri, gerçek değerlerle aynı varyansa ve aynı variograma sahip olacak şekilde belirlenir. Bu

ağırlıklar, matris ayrıştırma tekniğinde kovaryans matrisi  $\mathbf{C}$  nin bir matrisle bu matrisin evriğinin (transpozu) çarpımı şeklinde ayrıştırmasıyla elde edilir:

$$\mathbf{C} = \mathbf{P}\mathbf{P}^T \quad (2.4)$$

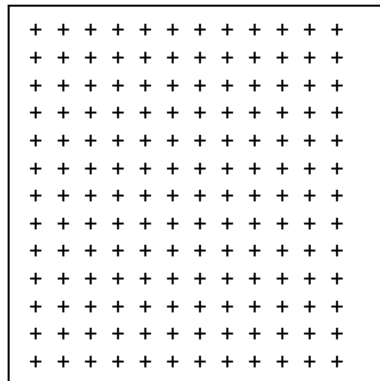
(2.4) eşitliğinde  $\mathbf{P}$ ; gerçel bir matristir.  $\mathbf{P}^T$  ise  $\mathbf{P}$  nin evriğini göstermektedir. Benzetim değerleri ise  $\mathbf{P}$  nin standart bağımsız normal dağılım vektörü  $\mathbf{y}$  ile çarpımından üretilir.

$$\mathbf{Z}_S = \mathbf{P}\mathbf{y} \quad (2.5)$$

$\mathbf{y}$ , tanım gereği standart bağımsız olduğu için benzetim değerlerinin kovaryansı,  $\mathbf{Z}$  nin kovaryansına eşittir.

$$\text{Cov}(\mathbf{Z}_S, \mathbf{Z}_S^T) = \text{Cov}(\mathbf{P}\mathbf{y}, \mathbf{y}^T \mathbf{P}^T) = \mathbf{P} \text{Cov}(\mathbf{y}, \mathbf{y}^T) \mathbf{P}^T = \mathbf{P}\mathbf{P}^T = \text{Cov}(\mathbf{Z}, \mathbf{Z}^T) \quad (2.6)$$

Klasik bir benzetim problemi olarak  $s_\beta, \beta = 1, \dots, n$ , benzetim yapılacak  $n$  adet noktayı gösterdiğini varsayalım (Şekil 2.2).



Şekil 2.2. Benzetim yapılacak noktalar

Bölgesel değişkene ilişkin kovaryans (ya da variogram) fonksiyonu bilindiğinde benzetim yapılacak noktalara ilişkin kovaryans matrisi

$$\mathbf{C} = \begin{bmatrix} C(s_1 - s_1) & C(s_1 - s_2) & \dots & C(s_1 - s_n) \\ C(s_2 - s_1) & C(s_2 - s_2) & \dots & C(s_2 - s_n) \\ \dots & \dots & \dots & \dots \\ C(s_n - s_1) & \dots & \dots & C(s_n - s_n) \end{bmatrix} \quad (2.7)$$

şeklinde yazılır. (2.7) eşitliğinde  $C(s_i - s_j)$ ;  $s_i$  ve  $s_j$  noktaları arasındaki uzaklığa bağlı kovaryans değerini göstermektedir.  $\mathbf{C}$  nin  $\mathbf{PP}^T$  ayrıştırması çeşitli şekillerde yapılabilir. Bu çalışmada klasik olan Cholesky yanında izgesel ve bakışimli ayrıştırma teknikleri gözönüne alınmıştır.

## 2.1. Cholesky Ayrıştırma Yöntemi

$C(s_i - s_j) = C(s_j - s_i)$  olduğundan  $\mathbf{C}$  matrisi bakışimli ve pozitif tanımlı bir matristir. Bu nedenle  $\mathbf{C} = \mathbf{LU}$  şeklinde ayrıştırıldığında  $\mathbf{U} = \mathbf{L}^T$  olup ( Golub and Van Loan, 1991, p.141) bu ayrıştırma Cholesky ayrıştırma yöntemi olarak bilinir:

$$\mathbf{C} = \mathbf{L} \cdot \mathbf{L}^T \quad (2.8)$$

(2.8 ) eşitliğine  $\mathbf{L}$ ; alt üçgen matrisi,  $\mathbf{L}^T$  ise  $\mathbf{L}$  nin evriğini göstermektedir. Daha açık bir şekilde

$$\mathbf{C} = \mathbf{LL}^T = \begin{bmatrix} L_{11} & 0 & \dots & 0 \\ L_{21} & L_{22} & 0 & 0 \\ \dots & \dots & \dots & \dots \\ L_{n1} & \dots & \dots & L_{nn} \end{bmatrix} \cdot \begin{bmatrix} L_{11} & 0 & \dots & 0 \\ L_{21} & L_{22} & 0 & 0 \\ \dots & \dots & \dots & \dots \\ L_{n1} & \dots & \dots & L_{nn} \end{bmatrix}^T \quad (2.9)$$

ile ifade edilebilir.

Cholesky ayrıştırma yöntemi ile üretilecek koşulsuz bir benzetimin ifadesi

$$\mathbf{y} = \mathbf{L} \cdot \mathbf{w} \quad (2.10)$$

eşitliği ile gösterilir. (2.10) eşitliğinde  $y$ , Cholesky ayrıştırma yöntemi ile üretilen koşulsuz benzetimleri içeren vektör,  $w$  ise, normal dağılımdan alınmış bir vektördür. (2.10) eşitliği daha açık bir şekilde

$$\begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & \cdot & \cdot & 0 \\ L_{21} & L_{22} & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ L_{n1} & \cdot & \cdot & \cdot & L_{nn} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{bmatrix} \quad (2.11)$$

ifade edilebilir.

Cholesky ayrıştırma yöntemi ile üretilecek koşullu bir benzetim ise aşağıdaki gibi tanımlanabilir :

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{L}_{dd} & 0 \\ \mathbf{L}_{nd} & \mathbf{L}_{nn} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{w}_d \\ \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{dd} \cdot \mathbf{w}_d \\ \mathbf{L}_{nd} \cdot \mathbf{w}_d + \mathbf{L}_{nn} \cdot \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{y}_s \end{bmatrix} \quad (2.12)$$

$\mathbf{L}_{dd}$ , veri noktaları arasındaki kovaryans matrisin Cholesky ayrıştırma yöntemi ile elde edilen alt üçgen matrisi,  $\mathbf{L}_{nd}$ , veri noktaları ile benzetim yapılacak noktalar arasındaki kovaryans matrisin Cholesky ayrıştırma yöntemi ile elde edilen alt üçgen matrisi,  $\mathbf{L}_{nn}$ , benzetim yapılacak noktalar arasındaki kovaryans matrisin Cholesky ayrıştırma yöntemi ile elde edilen alt üçgen matrisidir.  $\mathbf{w}_d$ , örneklenmiş noktalardaki değerleri üretecek şekilde, veri noktalarındaki normal dağılımdan alınmış vektör,  $\mathbf{w}_n$ , benzetim yapılacak noktalardaki normal dağılımdan alınmış değerlerin oluşturduğu vektördür.

$$\mathbf{y}_g = \mathbf{L}_{dd} \cdot \mathbf{w}_d \quad (2.13)$$

(2.13) eşitliğinden  $\mathbf{w}_d$  vektörünü çekip ( $\mathbf{w}_d = \mathbf{L}_{dd}^{-1} \cdot \mathbf{y}_{gerçek}$ ) yerine koyarsak aşağıdaki ifadeyi elde ederiz.

$$\mathbf{y}_s = \mathbf{L}_{nd} \cdot \mathbf{L}_{dd}^{-1} \cdot \mathbf{y}_g + \mathbf{L}_{nn} \cdot \mathbf{w}_n \quad (2.14)$$

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{L}_{dd} & 0 \\ \mathbf{L}_{nd} & \mathbf{L}_{nn} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{L}_{dd}^{-1} \cdot \mathbf{y}_g \\ \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{L}_{nd} \cdot \mathbf{L}_{dd}^{-1} \cdot \mathbf{y}_g + \mathbf{L}_{nn} \cdot \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{y}_s \end{bmatrix} \quad (2.15)$$

(2.14) eşitliğindeki  $\mathbf{y}_k$ , koşullu benzetimleri içeren vektördür.

## 2.2. Bakışimli Ayırıştırma Yöntemi

Bakışimli ayırıştırma yöntemi ile kovaryans matris

$$\mathbf{C} = \mathbf{B} \cdot \mathbf{B}^T \quad (2.15)$$

şeklinde ayrıştırılmaktadır. Bakışimli ayırıştırma yöntemi ile elde edilen  $\mathbf{B}$  matrisi her zaman bakışimli, pozitif tanımlı bir matristir. Dolayısıyla  $\mathbf{B}$  matrisi  $\mathbf{B}^T$  matrisine (evriğine) eşittir. Böylece kovaryans matris

$$\mathbf{C} = \mathbf{B} \cdot \mathbf{B} = \mathbf{B}^2 \quad (2.16)$$

şeklinde ifade edilebilir. (2.16) eşitliği daha açık bir şekilde

$$\mathbf{C} = \mathbf{B} \cdot \mathbf{B}^T = \begin{bmatrix} B_{11} & \cdot & \cdot & \cdot & B_{1n} \\ B_{21} & \cdot & \cdot & \cdot & B_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ B_{1n} & \cdot & \cdot & \cdot & B_{nn} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & \cdot & \cdot & \cdot & B_{1n} \\ B_{21} & \cdot & \cdot & \cdot & B_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ B_{1n} & \cdot & \cdot & \cdot & B_{nn} \end{bmatrix}^T \quad (2.17)$$

ifade edilebilir. Kovaryans matrisin Cholesky ayırıştırma yöntemi ile elde edilen alt üçgen ve üst üçgen matrisleri 0 elemanını içermesine rağmen, Bakışimli ayırıştırma yöntemi ile elde edilen  $\mathbf{B}$  matrisinin herhangi bir elemanı 0 değildir ve bütün elemanları pozitifdir.

$\mathbf{B}$  matrisi dik (orthogonal) ve köşegen (diagonal) matrisler cinsinden aşağıdaki gibi tanımlanmaktadır.

$$\mathbf{B} = \mathbf{Q} \cdot \mathbf{\Lambda}^{1/2} \cdot \mathbf{Q}^T \quad (2.18)$$

(2.18) eşitliğinde  $\mathbf{Q}$ , kovaryans matris  $\mathbf{C}$ 'nin öz vektör (eigenvector) lerinden oluşan dik matris,  $\mathbf{Q}^T$ ,  $\mathbf{Q}$  dik matrisinin evriği,  $\mathbf{\Lambda}$  ise öz değer (eigenvalue) lerden oluşan köşegen matristir.

Kovaryans matris bakışimli ayrıştırma yöntemi ile (2.15) eşitliğinde olduğu gibi tanımlandığı için, dik ve köşegen matrisler cinsinden aşağıdaki gibi ifade edilebilir.

$$\mathbf{C} = \mathbf{B}\mathbf{B}^T = \mathbf{Q}\mathbf{\Lambda}^{1/2}\mathbf{Q}^T\mathbf{Q}\mathbf{\Lambda}^{1/2}\mathbf{Q}^T = \mathbf{Q}\mathbf{\Lambda}^{1/2}\mathbf{I}\mathbf{\Lambda}^{1/2}\mathbf{Q}^T = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \quad (2.19)$$

Bakışimli ayrıştırma yöntemi ile üretilecek koşulsuz bir benzetimin ifadesi

$$\mathbf{y} = \mathbf{B}\mathbf{w} \quad (2.20)$$

eşitliği ile gösterilir. (2.20) eşitliğinde  $\mathbf{y}$ , bakışimli ayrıştırma yöntemi ile üretilen koşulsuz benzetimleri içeren vektör,  $\mathbf{w}$  ise, normal dağılımdan alınmış bir vektördür. (2.20) eşitliği daha açık bir şekilde

$$\begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} = \begin{bmatrix} B_{11} & \cdot & \cdot & \cdot & B_{1n} \\ B_{21} & \cdot & \cdot & \cdot & B_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ B_{1n} & \cdot & \cdot & \cdot & B_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{bmatrix} \quad (2.21)$$

ifade edilebilir.

Bakışimli ayrıştırma yöntemi ile üretilecek koşullu bir benzetim ise aşağıdaki gibi tanımlanabilir :

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{B}_{dd} & \mathbf{B}_{dn} \\ \mathbf{B}_{nd} & \mathbf{B}_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{w}_d \\ \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{dd}\mathbf{w}_d + \mathbf{B}_{dn}\mathbf{w}_n \\ \mathbf{B}_{nd}\mathbf{w}_d + \mathbf{B}_{nn}\mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{y}_s \end{bmatrix} \quad (2.22)$$

$\mathbf{B}_{dd}$ , veri noktaları arasındaki kovaryans matrisin bakışimli ayrıştırma yöntemi ile elde edilen bakışimli matris,  $\mathbf{B}_{nd}$  ve  $\mathbf{B}_{dn}$ , veri noktaları ile benzetim yapılacak noktalar arasındaki kovaryans matrisin bakışimli ayrıştırma yöntemi ile elde edilen bakışimli matris,  $\mathbf{B}_{nn}$ , benzetim yapılacak noktalar arasındaki kovaryans matrisin bakışimli ayrıştırma yöntemi ile elde edilen bakışimli matristir.

$$\mathbf{y}_g = \mathbf{B}_{dd} \cdot \mathbf{w}_d + \mathbf{B}_{dn} \cdot \mathbf{w}_n \quad (2.23)$$

Cholesky ayrıştırma yönteminde olduğu gibi (2.23) eşitliğindeki  $\mathbf{w}_d$  vektörünü çekersek aşağıdaki eşitliği elde ederiz.

$$\mathbf{w}_d = \mathbf{B}_{dd}^{-1} \cdot \mathbf{y}_g - \mathbf{B}_{dd}^{-1} \cdot \mathbf{B}_{dn} \cdot \mathbf{w}_n \quad (2.24)$$

$$\mathbf{y}_s = \mathbf{B}_{nd} \cdot \mathbf{B}_{dd}^{-1} \cdot \mathbf{y}_g + [\mathbf{B}_{nn} - \mathbf{B}_{nd} \cdot \mathbf{B}_{dd}^{-1} \cdot \mathbf{B}_{dn}] \mathbf{w}_n \quad (2.25)$$

(2.24) eşitliğindeki  $\mathbf{w}_d$  değerini (2.22) 'deki koşullu benzetim ifadesinde yerine koyarsak (2.25) eşitliğindeki koşullu benzetimleri içeren vektörü elde ederiz.

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{B}_{nd} \cdot \mathbf{B}_{dd}^{-1} \cdot \mathbf{y}_g + [\mathbf{B}_{nn} - \mathbf{B}_{nd} \cdot \mathbf{B}_{dd}^{-1} \cdot \mathbf{B}_{dn}] \mathbf{w}_n \end{bmatrix} \quad (2.25)$$

## 2.2. İzgesel Ayrıştırma Yöntemi

İzgesel ayrıştırma yöntemi ile kovaryans matris

$$\mathbf{C} = \mathbf{R} \cdot \mathbf{R}^T \quad (2.26)$$

şeklinde ayrıştırılmaktadır. (2.26) eşitliği daha açık bir şekilde

$$\mathbf{C} = \mathbf{R} \cdot \mathbf{R}^T = \begin{bmatrix} R_{11} & \cdot & \cdot & \cdot & R_{1n} \\ R_{21} & \cdot & \cdot & \cdot & R_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{1n} & \cdot & \cdot & \cdot & R_{nn} \end{bmatrix} \cdot \begin{bmatrix} R_{11} & \cdot & \cdot & \cdot & R_{1n} \\ R_{21} & \cdot & \cdot & \cdot & R_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{1n} & \cdot & \cdot & \cdot & R_{nn} \end{bmatrix}^T \quad (2.27)$$

ifade edilebilir. Bakışlı ayrıştırma yöntemi ile şekil olarak benzemesine rağmen teoride aralarındaki en önemli fark,  $\mathbf{R}$  matrisi keyfi bir matristir. Dolayısıyla  $\mathbf{R}$  matrisi her zaman pozitif tanımlı olmayabilir.  $\mathbf{R}$  matrisi  $\mathbf{R}^T$  matrisine (evriğine) eşit değildir.

Bunun yanısıra  $\mathbf{R}$  matrisi dik ve köşegen matrisler cinsinden aşağıdaki gibi tanımlanmaktadır.

$$\mathbf{R} = \mathbf{Q} \cdot \mathbf{\Lambda}^{1/2} \quad (2.28)$$

(2.28) eşitliğinde  $\mathbf{Q}$ , kovaryans matris  $\mathbf{C}$ 'nin öz vektör (eigenvector) lerinden oluşan dik matris,  $\mathbf{\Lambda}$  ise öz değer (eigenvalue) lerden oluşan köşegen matrisdir.

Kovaryans matris, izgesel ayrıştırma yöntemi ile (2.26) eşitliğinde olduğu gibi tanımlandığı için dik ve köşegen matrisler cinsinden aşağıdaki gibi ifade edilebilir.

$$\mathbf{C} = \mathbf{R} \cdot \mathbf{R}^T = \mathbf{Q} \cdot \mathbf{\Lambda}^{1/2} \cdot \mathbf{\Lambda}^{1/2} \cdot \mathbf{Q}^T = \mathbf{Q} \cdot \mathbf{\Lambda} \cdot \mathbf{Q}^T \quad (2.29)$$

İzgesel ayrıştırma yöntemi ile üretilecek koşulsuz bir benzetimin ifadesi

$$\mathbf{y} = \mathbf{R} \cdot \mathbf{w} \quad (2.30)$$

eşitliği ile gösterilir. (2.30) eşitliğinde  $\mathbf{y}$ , izgesel ayrıştırma yöntemi ile üretilen koşulsuz benzetimleri içeren vektör,  $\mathbf{w}$  ise, normal dağılımdan alınmış bir vektördür. (2.30) eşitliği daha açık bir şekilde

$$\begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix} = \begin{bmatrix} R_{11} & \cdot & \cdot & \cdot & R_{1n} \\ R_{21} & \cdot & \cdot & \cdot & R_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ R_{1n} & \cdot & \cdot & \cdot & R_{nn} \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{bmatrix} \quad (2.31)$$

ifade edilebilir.

İzgesel ayrıştırma yöntemi ile üretilecek koşullu bir benzetim ise aşağıdaki gibi tanımlanabilir :

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{R}_{dd} & \mathbf{R}_{dn} \\ \mathbf{R}_{nd} & \mathbf{R}_{nn} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{w}_d \\ \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{dd} \cdot \mathbf{w}_d + \mathbf{R}_{dn} \cdot \mathbf{w}_n \\ \mathbf{R}_{nd} \cdot \mathbf{w}_d + \mathbf{R}_{nn} \cdot \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{y}_s \end{bmatrix} \quad (2.32)$$

$\mathbf{R}_{dd}$ , veri noktaları arasındaki kovaryans matrisin izgesel ayrıştırma yöntemi ile elde edilen keyfi matris,  $\mathbf{R}_{nd}$  ve  $\mathbf{R}_{nd}$ , veri noktaları ile benzetim yapılacak noktalar

arasındaki kovaryans matrisin izgesel ayrıştırma yöntemi ile elde edilen keyfi matris,  $\mathbf{R}_{nn}$ , benzetim yapılacak noktalar arasındaki kovaryans matrisin izgesel ayrıştırma yöntemi ile elde edilen keyfi matrisdir.

$$\mathbf{y}_g = \mathbf{R}_{dd} \cdot \mathbf{w}_d + \mathbf{R}_{dn} \cdot \mathbf{w}_n \quad (2.33)$$

Cholesky ve bakışimli ayrıştırma yöntemlerinde olduğu gibi (2.33) eşitliğindeki  $\mathbf{w}_d$  vektörünü çekersek aşağıdaki eşitliği elde ederiz.

$$\mathbf{w}_d = \mathbf{R}_{dd}^{-1} \cdot \mathbf{y}_g - \mathbf{R}_{dd}^{-1} \cdot \mathbf{R}_{dn} \cdot \mathbf{w}_n \quad (2.34)$$

$$\mathbf{y}_s = \mathbf{R}_{nd} \cdot \mathbf{R}_{dd}^{-1} \cdot \mathbf{y}_g + [\mathbf{R}_{nn} - \mathbf{R}_{nd} \cdot \mathbf{R}_{dd}^{-1} \cdot \mathbf{R}_{dn}] \mathbf{w}_n \quad (2.35)$$

(2.34) eşitliğindeki  $\mathbf{w}_d$  değerini (2.32) 'deki koşullu benzetim ifadesinde yerine koyarsak aşağıdaki koşullu benzetimleri içeren matrisi elde ederiz.

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{R}_{nd} \cdot \mathbf{R}_{dd}^{-1} \cdot \mathbf{y}_g + [\mathbf{R}_{nn} - \mathbf{R}_{nd} \cdot \mathbf{R}_{dd}^{-1} \cdot \mathbf{R}_{dn}] \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{y}_s \end{bmatrix} \quad (2.36)$$

$\mathbf{R}_{dd}$  matrisi, keyfi bir matris olduğundan ters (inverse) matrisi her zaman tanımlı değildir. Dolayısıyla kovaryans matris  $\mathbf{C}$ , izgesel ayrıştırma yöntemi ile ayrıştırıldığında her zaman koşullu benzetim üretilemez.

Cholesky, bakışimli ve izgesel ayrıştırma yöntemleri ile koşullu benzetimler üretirken örneklenmiş lokasyonlardaki veriler kullanılır. Verilerin sayısı arttıkça, üretilen koşullu benzetim değerleri merkezi sınır teoreminden (central limit theorem) dolayı normal dağılıma yaklaşır. Dolayısıyla benzetim değerlerinin histogramı, örneklenmiş verilerin histogramı ile aynı olmaz. Bu yüzden örneklenmiş değerlerin önce normal dağılıma dönüştürülmesi gerekir. Bu şekilde üretilen koşullu benzetimlerin histogramı normal dağılım göstereceğinden, daha sonra üretilen benzetimlere geri dönüşüm uygulanır. Böylece örneklenmiş verilerin histogramı benzetimlerin histogramına eşit olur.

## 2.4. Merkezi Sınır Teoremi

$X_i$ ,  $i = 1, 2, \dots, n$  ortalaması  $\mu_i$ , varyansı  $\sigma_i$  olan her hangi bir olasılık dağılımına,  $P(x_1, x_2, \dots, x_n)$ , sahip  $n$  tane bağımsız rastlantı değişkeni olsun.  $X_i$  değişkeninin normal formu (2.37) eşitliğindeki gibi normal dağılıma yaklaşan, sınırlı kümülatif dağılım fonksiyonuna sahiptir.

$$X_{\text{norm}} = \frac{\sum_{i=1}^n x_i - \sum_{i=1}^n \mu_i}{\sqrt{\sum_{i=1}^n \sigma_i^2}} \quad (2.37)$$

Merkezi sınır teoremine göre,  $n$  değeri arttıkça dağılımın ortalaması  $\mu = 0$ , varyansı  $\sigma^2 = 1$  'e yaklaşarak dağılım normalleşmektedir. Bu durumda dağılımın normal formu (2.38) eşitliğindeki gibi olmaktadır.

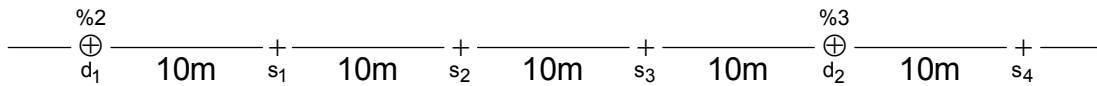
( <http://mathworld.wolfram.com/CentralLimitTheorem.html> )

$$X \equiv \frac{1}{n} \sum_{i=1}^n x_i \quad (2.38)$$

Choleky, bakışimli ve izgesel ayrıştırma teknikleri ile koşullu ve koşulsuz benzetimlerin üretilmesini daha açık bir şekilde anlatabilmek amacıyla aşağıda sayısal bir örnek verilmiştir.

## 2.5. Benzetime İlişkin Sayısal Örnekler

Cholesky, bakışimli ve izgesel ayrıştırma teknikleri ile koşullu ve koşulsuz benzetimleri üretmek amacıyla 50 m uzunluğunda bir mesafe seçilmiştir. (Şekil2.3)



Şekil 2.3. Veri ve benzetim lokasyonları

Şekil 2.3. deki  $d_1$  ve  $d_2$  mevcut veri noktalarını,  $s_1, s_2, s_3, s_4$  ise benzetim yapılacak noktaları ifade etmektedir.  $d_1$  noktasındaki gerçek değer %2 iken  $d_2$  noktasında

bu deęer %3 dür. Noktalar arasındaki uzaklık ise 10m dir. Amaç  $s_i$ ,  $i=1, \dots, 4$  noktalarında koşullu ve koşulsuz benzetim deęerleri üretmektir.

Normal dağılımdan alınan rastlantı deęerleri, -1.1, -0.3, 2.1, 1.7, 0.5, 1.3 olarak verilmektedir. Bunlar sırasıyla  $d_1, s_1, s_2, s_3, d_2$  ve  $s_4$  noktalarındaki rastlantı sayılarıdır.

Variogram modeli, küresel model seçilmiştir. Küresel model,

$$\gamma(h) = C_0 + C \left[ 1.5 \left( \frac{h}{a} \right) - 0.5 \left( \frac{h}{a} \right)^3 \right], \quad h \leq a \quad (2.39)$$

$$\gamma(h) = C_0 + C, \quad h > a \quad (2.40)$$

$$\gamma(h) = 0, \quad h=0 \quad (2.41)$$

şeklinde tanımlanmaktadır. Küresel model ifadesinde  $a$ , yapısal uzaklık,  $h$ , variogramın hesaplandığı mesafe,  $C$ , variogramın eşik deęeri,  $C_0$ , variogramın külçe etkisini göstermektedir. Eęer yapısal uzaklık variogramın hesaplanacağı mesafeden büyük ya da eşitse variogram (2.39) veya küçükse (2.40) eşitliği kullanılarak hesaplanır. Variogram fonksiyonu, bölgesel deęişkenin deęerleri arasındaki farkın uzaklığa baęlı deęişimlerini gösterdiği için teorik olarak  $h=0$  olduğunda variogram deęeri (2.41) eşitliğindeki gibi sifıra eşittir.

Örnek olarak seçilen modelde  $C=0.8$ ,  $C_0=0.2$  ve  $a=70$  alınmıştır. Bu veriler kullanılarak kovaryans matrisin elemanları

$$C(h) = \sigma^2 - \gamma(h) \quad (2.42)$$

eşitliği kullanılarak hesaplanmıştır. Deęerler normal dağılımdan alındığı için  $\sigma^2=1$  dir. Bu durumda kovaryans matrisin elemanları

$$C(h) = 1 - \gamma(h) \quad (2.43)$$

ile hesaplanır. Veri noktaları ile birlikte toplam 6 nokta olduğu için 6\*6 büyüklüğünde kovaryans matris elde edilir.

$$\mathbf{C} = \begin{bmatrix} \begin{bmatrix} 1-\gamma(d_1-d_1) & 1-\gamma(d_1-d_2) \\ 1-\gamma(d_2-d_1) & 1-\gamma(d_2-d_2) \end{bmatrix} & \begin{bmatrix} 1-\gamma(d_1-s_1) & \dots & 1-\gamma(d_1-s_4) \\ 1-\gamma(d_2-s_1) & \dots & 1-\gamma(d_2-s_4) \end{bmatrix} \\ \begin{bmatrix} 1-\gamma(s_1-d_1) & 1-\gamma(s_1-d_2) \\ \vdots & \vdots \\ 1-\gamma(s_4-d_1) & 1-\gamma(s_4-d_2) \end{bmatrix} & \begin{bmatrix} 1-\gamma(s_1-s_1) & \dots & 1-\gamma(s_1-s_4) \\ \vdots & \ddots & \vdots \\ 1-\gamma(s_4-s_1) & \dots & 1-\gamma(s_4-s_4) \end{bmatrix} \end{bmatrix} \quad (2.44)$$

Örnek olarak  $d_1$  ve  $d_2$  arasındaki uzaklığa karşılık gelen variogram (2.39) eşitliği kullanılarak hesaplanırsa (2.45) eşitliğindeki sonuç elde edilir.

$$\gamma(d_1-d_2) = \gamma(40) = 0.2 + 0.8 \left[ 1.5 \left( \frac{40}{70} \right) - 0.5 \left( \frac{40}{70} \right)^3 \right] = 0.811 \quad (2.45)$$

$d_1$  ve  $d_2$  arasındaki kovaryans ise (2.43) eşitliğinde olduğu gibi hesaplandığından  $C(40) = 1 - \gamma(40) = 1 - 0.811 = 0.189$  elde edilir. Bu şekilde belirlenen kovaryans matris (2.46) da verilmiştir.

$$\mathbf{C} = \begin{bmatrix} 1 & 0.189 & 0.630 & 0.466 & 0.317 & 0.089 \\ 0.189 & 1 & 0.317 & 0.466 & 0.630 & 0.630 \\ 0.630 & 0.317 & 1 & 0.630 & 0.466 & 0.189 \\ 0.466 & 0.466 & 0.630 & 1 & 0.630 & 0.317 \\ 0.317 & 0.630 & 0.466 & 0.630 & 1 & 0.466 \\ 0.089 & 0.630 & 0.189 & 0.317 & 0.466 & 1 \end{bmatrix} \quad (2.46)$$

İlk olarak Cholesky ayrıştırma yöntemi ile (2.8) eşitliğini kullanarak, kovaryans matrisi ayrıştırdığımızda elde edilen alt üçgen matris (2.47) de verilmiştir.

$$\mathbf{L} = \begin{bmatrix} 1 & & & & & \\ 0.189 & 0.982 & & & & \\ 0.630 & 0.202 & 0.750 & & & \\ 0.466 & 0.385 & 0.345 & 0.718 & & \\ 0.317 & 0.581 & 0.199 & 0.265 & 0.673 & \\ 0.089 & 0.624 & 0.009 & 0.044 & 0.092 & 0.769 \end{bmatrix} \quad (2.47)$$

(2.10) eşitliği kullanılarak, Cholesky ayrıştırma yöntemi ile üretilen koşulsuz benzetimler (2.48) eşitliğindeki gibidir.

$$\mathbf{y} = \begin{bmatrix} 1 & & & & & & \\ 0.189 & 0.982 & & & & & \\ 0.630 & 0.202 & 0.750 & & & & \\ 0.466 & 0.385 & 0.345 & 0.718 & & & \\ 0.317 & 0.581 & 0.199 & 0.265 & 0.673 & & \\ 0.089 & 0.624 & 0.009 & 0.044 & 0.092 & 0.769 & \end{bmatrix} \cdot \begin{bmatrix} -1.1 \\ 0.5 \\ -0.3 \\ 2.1 \\ 1.7 \\ 1.3 \end{bmatrix} = \begin{bmatrix} -1.1 \\ 0.28 \\ -0.82 \\ 1.08 \\ 1.58 \\ 1.46 \end{bmatrix} \quad (2.48)$$

Koşullu benzetimler ise (2.13) ve (2.14) eşitlikleri kullanılarak elde edilebilir.

$$\mathbf{y}_g = \mathbf{L}_{dd} \cdot \mathbf{w}_d$$

$$\mathbf{y}_g = \begin{bmatrix} 1 & \\ 0.189 & 0.982 \end{bmatrix} \cdot \begin{bmatrix} 2.0 \\ 3.0 \end{bmatrix} = \begin{bmatrix} 2 \\ 3.32 \end{bmatrix}$$

$$\mathbf{y}_s = \mathbf{L}_{nd} \cdot \mathbf{L}_{dd}^{-1} \cdot \mathbf{y}_g + \mathbf{L}_{nn} \cdot \mathbf{w}_n$$

$$\mathbf{y}_s = \begin{bmatrix} 0.630 & 0.202 \\ 0.466 & 0.385 \\ 0.317 & 0.581 \\ 0.089 & 0.624 \end{bmatrix} \cdot \begin{bmatrix} 1 & \\ -0.192 & 1.018 \end{bmatrix} \cdot \begin{bmatrix} 2.00 \\ 3.32 \end{bmatrix} + \begin{bmatrix} 0.750 & & & \\ 0.345 & 0.718 & & \\ 0.199 & 0.265 & 0.673 & \\ 0.009 & 0.044 & 0.092 & 0.769 \end{bmatrix} \cdot \begin{bmatrix} -0.3 \\ 2.1 \\ 1.7 \\ 1.3 \end{bmatrix}$$

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_g \\ \mathbf{y}_s \end{bmatrix} = \begin{bmatrix} 2.00 \\ 3.32 \\ 1.64 \\ 3.49 \\ 4.01 \\ 2.98 \end{bmatrix}$$

Bakışimli ayrıştırma yöntemi ile (2.13) eşitliğini kullanarak, kovaryans matrisi ayrıştırdığımızda elde edilen **B** matrisi aşağıdaki gibidir.

$$\mathbf{B} = \begin{bmatrix} 0.926 & 0.050 & 0.305 & 0.189 & 0.108 & 0.010 \\ 0.050 & 0.884 & 0.105 & 0.179 & 0.284 & 0.305 \\ 0.305 & 0.105 & 0.884 & 0.284 & 0.179 & 0.050 \\ 0.189 & 0.179 & 0.284 & 0.873 & 0.279 & 0.108 \\ 0.108 & 0.284 & 0.179 & 0.279 & 0.873 & 0.189 \\ 0.010 & 0.305 & 0.050 & 0.108 & 0.189 & 0.926 \end{bmatrix} \quad (2.49)$$

Öz vektörlerden oluşan dik matris ile öz değerlerden oluşan köşegen matris ise

$$\mathbf{Q} = \begin{bmatrix} 0.335 & 0.540 & 0.542 & -0.427 & 0.306 & -0.159 \\ 0.417 & -0.427 & 0.063 & -0.386 & -0.567 & -0.411 \\ 0.417 & 0.427 & 0.063 & 0.386 & -0.567 & 0.411 \\ 0.462 & 0.161 & -0.450 & 0.410 & 0.290 & -0.553 \\ 0.335 & -0.540 & 0.542 & 0.427 & 0.306 & 0.159 \end{bmatrix} \quad (2.50)$$

$$\mathbf{\Lambda} = \begin{bmatrix} 3.188 & & & & & \\ & 1.304 & & & & \\ & & 0.534 & & & \\ & 0 & & 0.369 & & \\ & & & & 0.314 & \\ & & & & & 0.291 \end{bmatrix} \quad (2.51)$$

(2.50) ve (2.51) deki gibidir.

Bakışimli ayrıştırma yöntemi ile (2.20) eşitliği kullanılarak üretilen koşulsuz benzetimler

$$\mathbf{y} = \begin{bmatrix} 0.926 & 0.050 & 0.305 & 0.189 & 0.108 & 0.010 \\ 0.050 & 0.884 & 0.105 & 0.179 & 0.284 & 0.305 \\ 0.305 & 0.105 & 0.884 & 0.284 & 0.179 & 0.050 \\ 0.189 & 0.179 & 0.284 & 0.873 & 0.279 & 0.108 \\ 0.108 & 0.284 & 0.179 & 0.279 & 0.873 & 0.189 \\ 0.010 & 0.305 & 0.050 & 0.108 & 0.189 & 0.926 \end{bmatrix} \cdot \begin{bmatrix} -1.1 \\ 0.5 \\ -0.3 \\ 2.1 \\ 1.7 \\ 1.3 \end{bmatrix} = \begin{bmatrix} -0.49 \\ 1.61 \\ 0.42 \\ 2.24 \\ 2.28 \\ 1.88 \end{bmatrix} \quad (2.52)$$

şeklinde bulunur.

Koşullu benzetimler ise (2.23) ve (2.25) eşitlikleri kullanılarak elde edilebilir.

$$\mathbf{y}_g = \mathbf{B}_{dd} \cdot \mathbf{w}_d + \mathbf{B}_{dn} \cdot \mathbf{w}_n$$

$$\mathbf{y}_g = \begin{bmatrix} 0.926 & 0.050 \\ 0.050 & 0.884 \end{bmatrix} \cdot \begin{bmatrix} 2.0 \\ 3.0 \end{bmatrix} + \begin{bmatrix} 0.305 & 0.189 & 0.108 & 0.010 \\ 0.105 & 0.179 & 0.284 & 0.305 \end{bmatrix} \cdot \begin{bmatrix} -0.3 \\ 2.1 \\ 1.7 \\ 1.3 \end{bmatrix} = \begin{bmatrix} 2.50 \\ 3.98 \end{bmatrix}$$

$$\mathbf{y}_s = \mathbf{B}_{nd} \cdot \mathbf{B}_{dd}^{-1} \cdot \mathbf{y}_g + [\mathbf{B}_{nn} - \mathbf{B}_{nd} \cdot \mathbf{B}_{dd}^{-1} \cdot \mathbf{B}_{dn}] \cdot \mathbf{w}_n$$

$$\mathbf{B}_{nd} \cdot \mathbf{B}_{dd}^{-1} \cdot \mathbf{y}_g = \begin{bmatrix} 0.305 & 0.105 \\ 0.189 & 0.179 \\ 0.108 & 0.284 \\ 0.010 & 0.305 \end{bmatrix} \cdot \begin{bmatrix} 1.078 & -0.061 \\ -0.061 & 1.135 \end{bmatrix} \cdot \begin{bmatrix} 2.50 \\ 3.98 \end{bmatrix} = \begin{bmatrix} 1.206 \\ 1.245 \\ 1.504 \\ 1.356 \end{bmatrix}$$

$$[\mathbf{B}_{nn} - \mathbf{B}_{nd} \cdot \mathbf{B}_{dd}^{-1} \cdot \mathbf{B}_{dn}] \cdot \mathbf{w}_n = \begin{bmatrix} 0.775 & 0.205 & 0.116 & 0.016 \\ 0.205 & 0.802 & 0.204 & 0.048 \\ 0.116 & 0.204 & 0.773 & 0.092 \\ 0.016 & 0.048 & 0.092 & 0.821 \end{bmatrix} \cdot \begin{bmatrix} -0.3 \\ 2.1 \\ 1.7 \\ 1.3 \end{bmatrix} = \begin{bmatrix} 0.416 \\ 2.031 \\ 1.826 \\ 1.318 \end{bmatrix}$$

$$\mathbf{y}_s = \begin{bmatrix} 1.206 \\ 1.245 \\ 1.504 \\ 1.356 \end{bmatrix} + \begin{bmatrix} 0.416 \\ 2.031 \\ 1.826 \\ 1.318 \end{bmatrix} = \begin{bmatrix} 1.622 \\ 3.276 \\ 3.330 \\ 2.674 \end{bmatrix}$$

$$\mathbf{y}_k = \begin{bmatrix} 2.50 \\ 3.98 \\ 1.62 \\ 3.28 \\ 3.33 \\ 2.67 \end{bmatrix}$$

İzgesel ayrıştırma yöntemi ile (2.26) eşitliğini kullanarak, kovaryans matrisi ayrıştırdığımızda elde edilen **R** matrisi (2.54) eşitliğindeki gibidir.

$$\mathbf{R} = \begin{bmatrix} 0.598 & 0.617 & 0.396 & -0.260 & 0.172 & -0.086 \\ 0.745 & -0.487 & 0.046 & -0.235 & -0.318 & -0.222 \\ 0.745 & 0.487 & 0.046 & 0.235 & -0.318 & 0.222 \\ 0.825 & 0.184 & -0.329 & 0.249 & 0.163 & -0.298 \\ 0.825 & -0.184 & -0.329 & -0.249 & 0.163 & 0.298 \\ 0.598 & -0.617 & 0.396 & 0.260 & 0.172 & 0.086 \end{bmatrix} \quad (2.54)$$

Öz vektörlerden oluşan dik matris ile öz değerlerden oluşan köşegen matris ise

$$\mathbf{Q} = \begin{bmatrix} 0.335 & 0.540 & 0.542 & -0.427 & 0.306 & -0.159 \\ 0.417 & -0.427 & 0.063 & -0.386 & -0.567 & -0.411 \\ 0.417 & 0.427 & 0.063 & 0.386 & -0.567 & 0.411 \\ 0.462 & 0.161 & -0.450 & 0.410 & 0.290 & -0.553 \\ 0.335 & -0.540 & 0.542 & 0.427 & 0.306 & 0.159 \end{bmatrix} \quad (2.55)$$

$$\mathbf{\Lambda} = \begin{bmatrix} 3.188 & & & & & \\ & 1.304 & & & & \\ & & 0.534 & & & \\ & 0 & & 0.369 & & \\ & & & & 0.314 & \\ & & & & & 0.291 \end{bmatrix} \quad (2.56)$$

(2.55) ve (2.56) daki gibidir.

(2.30) eşitliği kullanılarak, İzgesel ayrıştırma yöntemi ile üretilen koşulsuz benzetimler

$$\mathbf{y} = \begin{bmatrix} 0.598 & 0.617 & 0.396 & -0.260 & 0.172 & -0.086 \\ 0.745 & -0.487 & 0.046 & -0.235 & -0.318 & -0.222 \\ 0.745 & 0.487 & 0.046 & 0.235 & -0.318 & 0.222 \\ 0.825 & 0.184 & -0.329 & 0.249 & 0.163 & -0.298 \\ 0.825 & -0.184 & -0.329 & -0.249 & 0.163 & 0.298 \\ 0.598 & -0.617 & 0.396 & 0.260 & 0.172 & 0.086 \end{bmatrix} \cdot \begin{bmatrix} -1.1 \\ 0.5 \\ -0.3 \\ 2.1 \\ 1.7 \\ 1.3 \end{bmatrix} = \begin{bmatrix} -0.83 \\ -2.40 \\ -0.35 \\ -0.30 \\ -0.76 \\ -0.13 \end{bmatrix} \quad (2.57)$$

Koşullu benzetimler ise (2.33) ve (2.35) eşitlikleri kullanılarak elde edilebilir.

$$\mathbf{y}_g = \mathbf{R}_{dd} \cdot \mathbf{w}_d + \mathbf{R}_{dn} \cdot \mathbf{w}_n$$

$$\mathbf{y}_g = \begin{bmatrix} 0.598 & 0.617 \\ 0.745 & -0.487 \end{bmatrix} \cdot \begin{bmatrix} 2.0 \\ 3.0 \end{bmatrix} + \begin{bmatrix} 0.396 & -0.260 & 0.172 & -0.086 \\ 0.046 & -0.235 & -0.318 & -0.222 \end{bmatrix} \cdot \begin{bmatrix} -0.3 \\ 2.1 \\ 1.7 \\ 1.3 \end{bmatrix} = \begin{bmatrix} 2.56 \\ -1.31 \end{bmatrix}$$

$$\mathbf{y}_s = \mathbf{R}_{nd} \cdot \mathbf{R}_{dd}^{-1} \cdot \mathbf{y}_g + [\mathbf{R}_{nn} - \mathbf{R}_{nd} \cdot \mathbf{R}_{dd}^{-1} \cdot \mathbf{R}_{dn}] \cdot \mathbf{w}_n$$

$$\mathbf{R}_{nd} \cdot \mathbf{R}_{dd}^{-1} \cdot \mathbf{y}_g = \begin{bmatrix} 0.745 & 0.487 \\ 0.825 & 0.184 \\ 0.825 & -0.184 \\ 0.598 & -0.617 \end{bmatrix} \cdot \begin{bmatrix} 0.649 & 0.821 \\ 0.992 & -0.796 \end{bmatrix} \cdot \begin{bmatrix} 2.56 \\ -1.31 \end{bmatrix} = \begin{bmatrix} 2.184 \\ 1.146 \\ -0.173 \\ -1.858 \end{bmatrix}$$

$$[\mathbf{R}_{nn} - \mathbf{R}_{nd} \cdot \mathbf{R}_{dd}^{-1} \cdot \mathbf{R}_{dn}] \cdot \mathbf{w}_n = \begin{bmatrix} -0.347 & 0.539 & -0.413 & 0.355 \\ -0.638 & 0.560 & 0.208 & -0.118 \\ -0.507 & 0.036 & 0.364 & 0.511 \\ 0 & 0.520 & 0 & 0.172 \end{bmatrix} \cdot \begin{bmatrix} -0.3 \\ 2.1 \\ 1.7 \\ 1.3 \end{bmatrix} = \begin{bmatrix} 0.995 \\ 1.568 \\ 1.512 \\ 1.315 \end{bmatrix}$$

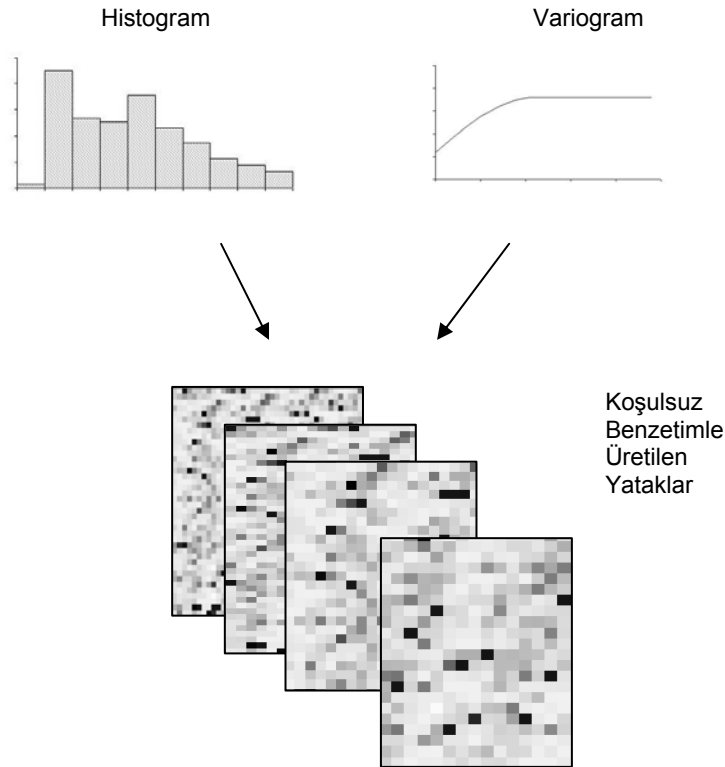
$$\mathbf{y}_s = \begin{bmatrix} 2.184 \\ 1.146 \\ -0.173 \\ -1.858 \end{bmatrix} + \begin{bmatrix} 0.995 \\ 1.568 \\ 1.512 \\ 1.315 \end{bmatrix} = \begin{bmatrix} 3.179 \\ 2.714 \\ 1.339 \\ -0.543 \end{bmatrix}$$

$$\mathbf{y}_k = \begin{bmatrix} 2.56 \\ -1.36 \\ 3.18 \\ 2.71 \\ 1.34 \\ -0.54 \end{bmatrix}$$

### 3. BENZETİM DENEYİNDE KULLANILACAK MODEL YATAKLARIN ÜRETİMİ VE PERFORMANS ÖLÇÜTLERİ

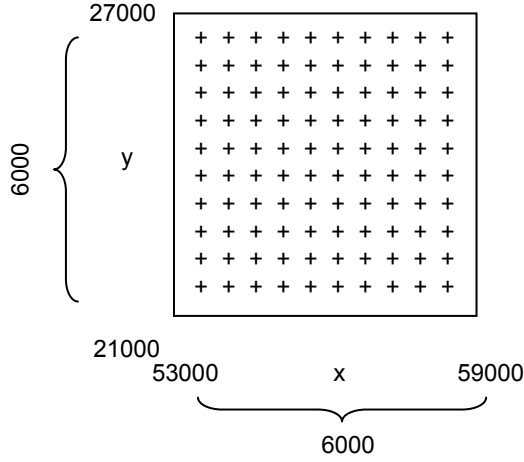
#### 3.1. Koşulsuz Benzetim

Cholesky, bakışimli ve izgesel ayrıştırma yöntemleri kullanılarak gerçek veriler gibi aynı histograma ve variograma sahip koşulsuz benzetimler üretilmiştir. Bu amaçla GSLIB (Deutsch ve Journel,1998) den LUSIM programı temel alınmıştır. LUSIM yalnızca Cholesky ayrıştırmasını dikkate alarak benzetim yapmaktadır. Bu program, bakışimli ve izgesel ayrıştırma yöntemlerini dikkate alacak şekilde yeniden yazılmış ve FORTRAN kodu Ek -1 de verilmiştir. Koşulsuz benzetimin akım şeması Şekil 3.1. de verilmiştir.



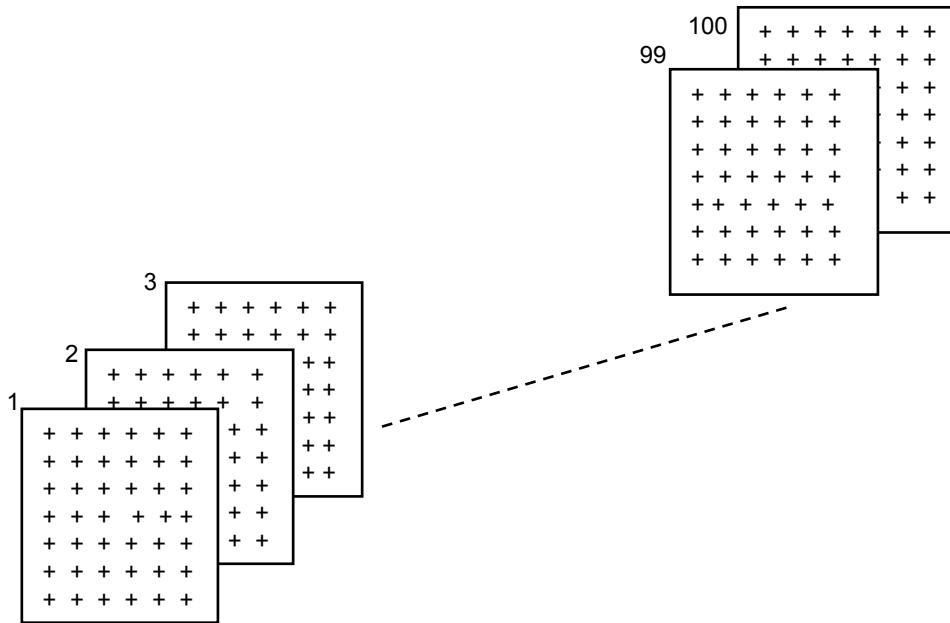
Şekil 3.1. Koşulsuz benzetim akım şeması

Koşulsuz benzetimler üretmek için model olarak seçilen alan ise Şekil 3.2. de verilmiştir.



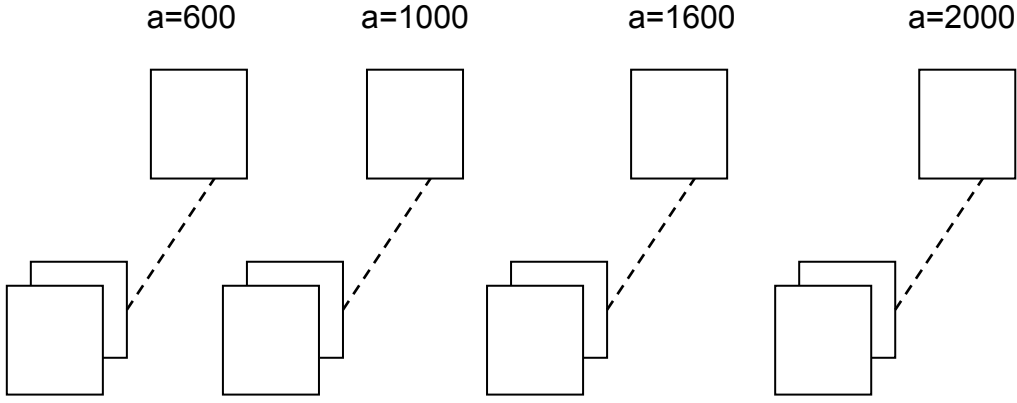
Şekil 3.2. Koşulsuz benzetim için seçilen model alan

Belirlenen model alanın bir kenarı 6000 birim olup bu kenar üzerinde 30 nokta bulunmaktadır. Toplam  $30 \times 30 = 900$  nokta da koordinatları belirli kare bir alanda koşulsuz benzetimler üretilmiştir. Üretilen her bir nokta arasındaki mesafe 200 birimdir. 900 nokta içeren model yatağın üretilmesi her bir yöntem için 100 'er defa tekrarlanmıştır. (Şekil 3.3)



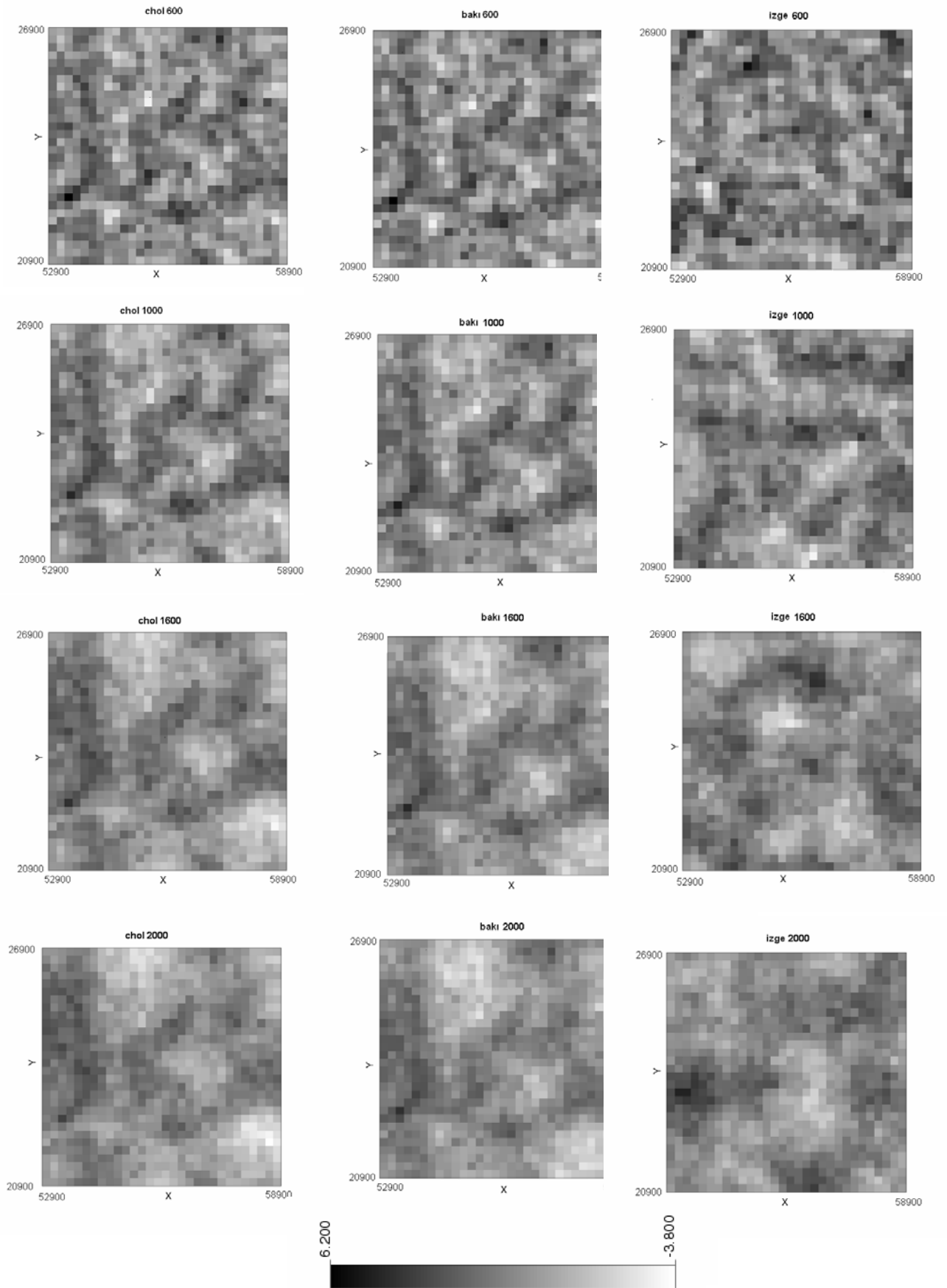
Şekil 3.3. Koşulsuz benzetimle üretilen 100 adet yatak

Koşulsuz benzetimler üretirken külçe etkisi,  $C_0 = 0$  ve variogramın eşik değeri,  $C = 1$  alınmıştır. Bununla birlikte  $a = 600$ ,  $a = 1000$ ,  $a = 1600$  ve  $a = 2000$  olmak üzere dört farklı yapısal uzaklık değeri kullanılarak, bu parametrenin benzetimler üzerindeki etkisi incelenmiştir. Dolayısıyla üç yöntemle toplam 1200 tane koşulsuz benzetim içeren yatak üretilmiştir. (Şekil 3.4.)



Şekil 3.4. Değişik yapısal uzaklıklarda üretilen yataklar

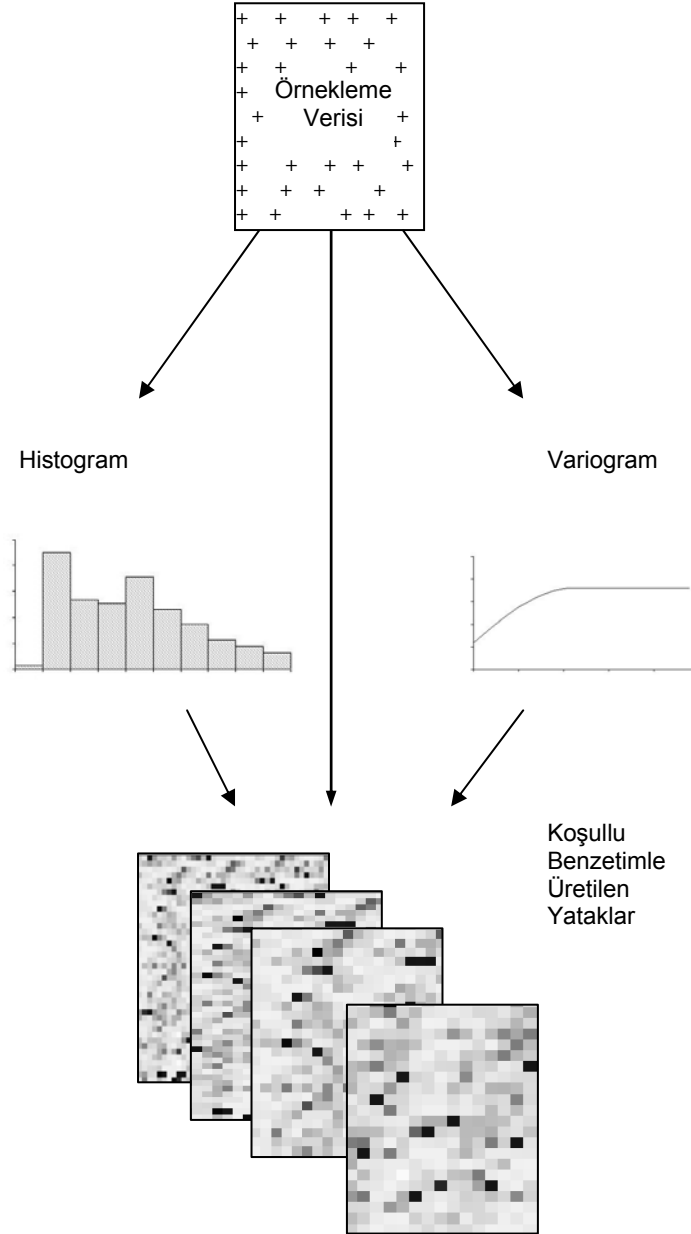
Her bir yöntemle üretilen ilk yatağın imge (image) haritaları Şekil 3.5 de gösterilmiştir.



Şekil 3.5.Cholesky, bakışimli ve izgesel ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen birinci koşulsuz benzetimlerin imge haritaları

### 3.2. Koşullu Benzetim

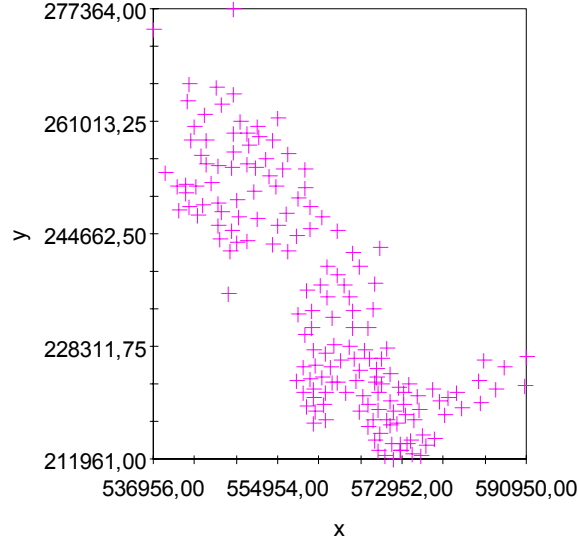
Cholesky ve bakışimli ayrıştırma yöntemleri kullanılarak mevcut veriler gibi aynı değerlere, aynı histograma ve aynı variograma sahip koşullu benzetimler üretilmiştir. Koşullu benzetimin akım şeması Şekil 3.6 da verilmiştir.



Şekil 3.6. Koşullu benzetim akım şeması

Koşulsuz benzetimden farklı olarak koşullu benzetim de mevcut veriler gibi aynı değerler üretilmektedir. Bu çalışmada kullanılan koşullandırıcı verilerin

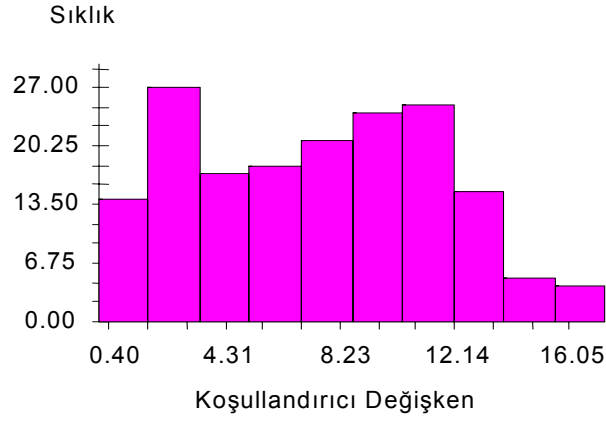
lokasyonları Şekil 3.7, tanımlayıcı istatistikleri Çizelge 3.1. ve histogramı Şekil 3.8. de gösterilmiştir.



Şekil 3.7. Mevcut veri lokasyonları

Çizelge 3.1. Verilerin tanımlayıcı istatistikleri

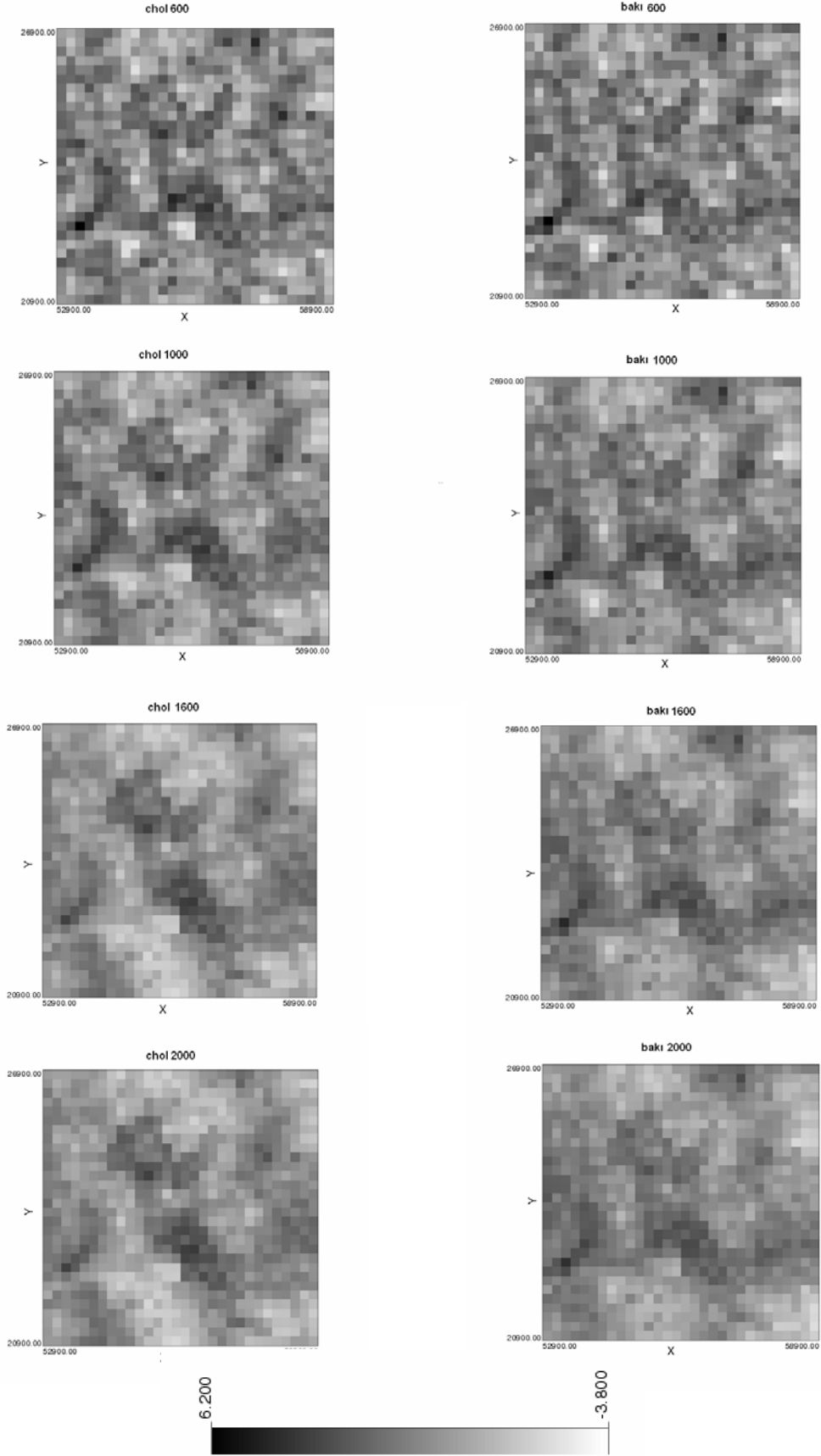
Ortalama	7.05
Standart Hata	0.32
Ortanca	7.38
Kip	10.40
Standart Sapma	4.15
Örnek Varyans	17.23
Basıklık	-1.05
Çarpıklık	0.09
Aralık	15.65
En Büyük	0.40
En Küçük	16.05
Toplam	1197.74
Say	170
En Büyük	16.05
En Küçük	0.40



Şekil 3.8. Mevcut verilerin histogramı

Koşulsuz benzetimler üretirken dikkate alınan parametreler koşullu benzetimler için de kullanılmıştır. Koşulsuz benzetim de olduğu gibi külçe etkisi,  $C_0 = 0$  ve variogramın eşik değeri,  $C = 1$  alınmıştır.  $a = 600$ ,  $a = 1000$ ,  $a = 1600$  ve  $a = 2000$  olmak üzere dört farklı yapısal uzaklık kullanılarak, yapısal uzaklığın benzetimler üzerindeki etkisi incelenmiştir. Bu işlem, Cholesky ve bakışimli ayrıştırma yöntemleri için 100 'er defa tekrarlanmıştır. (Şekil 3.3) İzgesel yöntem koşullu benzetimlerin üretimini garanti etmediğinden dikkate alınmamış bu nedenle Cholesky ve bakışimli ayrıştırma yöntemleri kullanılarak toplam 800 tane koşullu benzetim içeren yatak üretilmiştir.

Bu ayrıştırma yöntemleri ile üretilen birinci koşullu benzetimlerin imge haritaları Şekil 3.9. da verilmiştir.



Şekil 3.9.Cholesky ve bakımlı ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen birinci koşulsuz benzetimlerin imge haritaları

### 3.3. Performans Ölçütleri

Ayrıştırma yöntemlerinin performansları üretilen koşullu ve koşulsuz benzetimlerin ortalama ve varyansları, variogram üretimi, deneysel variogramların model variogramlara yaklaşma hızı, Kolmogorov – Simirnov (KS) testi ve bağımsızlık testi gibi ölçütler kullanılarak değerlendirilmiştir.

#### 3.3.1. Ortalama ve Varyans

Performans değerlendirmesinde ilk ele alınan ölçüt, aynı bir noktada 100 'er adet üretilen benzetim değerlerinin aritmetik ortalamasıdır. Ortalama (3.1) eşitliği kullanılarak hesaplanmıştır.

$$\mu = \frac{1}{n} \sum_{i=1}^n s_i \quad (3.1)$$

(3.1) eşitliğindeki n; benzetim sayısı,  $s_i$  ise benzetim değerini ifade etmektedir.

Benzetimlerin varyansı ise (3.2) eşitliği kullanılarak hesaplanmıştır.

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (s_i - \mu)^2 \quad (3.2)$$

#### 3.3.2. Variogram

Performans değerlendirmesindeki diğer bir ölçüt variogram üretimidir. Bu amaçla aynı bir yatağın 100 benzetim modeline ilişkin deneysel variogramları model variogram ile karşılaştırılmıştır. Cholesky, bakışimli ve izgesel ayrıştırma yöntemleri ile üretilen benzetimlerin deneysel variogramları ile model variogramlar arasındaki dalgalanmalara bakılarak üretilen variogramların model variogramlara yakınlığı incelenmiştir. Ayrıca artan yapısal uzaklık değerlerinde üretilen variogramların model variogramlardan ne kadar saptığına ve yapısal uzaklığın variogramlar üzerindeki etkisine bakılmıştır. Model variogramlar (2.39), (2.40), (2.41) eşitliklerinde tanımlanan küresel modellerdir.

### 3.3.3. Yakınsama Testi

Üretilen benzetimlerin variogramlarının model variogramlara ne kadar hızda yaklaştığı ve hedeflenen variograma kaçınıcı benzetimde ulaştığını belirlemek amacıyla yakınsama (convergence) testi yapılmıştır. İlk olarak deneysel variogramların modellere doğru yaklaşması kontrol edilmiştir. Bu amaçla (3.3) eşitliği ile verilen indeks dikkate alınmıştır.

$$\psi_{\ell} = \frac{\sqrt{\sum_{i=1}^n (\gamma_{\ell}(h_i) - \gamma(h_i))^2}}{\sum_{i=1}^n \gamma(h_i)^2} \quad \ell = 1, \dots, 100 \quad (3.3)$$

(3.3) eşitliğindeki  $\psi_{\ell}$ ; model ve ortalama deneysel variogram arasındaki yakınlığı hesaplayan indeks, n; adım uzaklığı sayısı,  $h_i$ , i' ninci adım uzaklığı için ortalama mesafe,  $\gamma$ , model variogram ve  $\gamma_{\ell}$ , deneysel variogramı göstermektedir. Bu testi gerçekleştirmek amacıyla bir FORTRAN programı yazılmış ve Ek-2 de verilmiştir.

Her bir benzetim için deneysel variogramlar hesaplandıktan sonra , ortalama variogramlar benzetimlerin her bir serisi için hesaplanmıştır. Örneğin birinci ortalama variogram , ilk benzetimin deneysel variogramına eşittir. İkinci ortalama variogram , ilk iki benzetimden kestirilen deneysel variogramların ortalamasına eşittir. 100. ortalama variogram ise , 100 tane benzetimin deneysel variogramlarının ortalamasına eşittir.

### 3.3.4. Bağımsızlık Testi

Jeostatistiksel benzetimde üretilen model yatakların birbirinden bağımsız olması arzu edilir. Bu amaçla, 100 'er defa üretilen benzetimlerin birbirinden bağımsız olup olmadığını, hangi yöntemin ya da yöntemlerin bağımsızlığa daha yakın sonuçlar verdiğini görmek amacıyla bağımsızlık testi yapılmıştır. Bu amaçla (3.4) ile verilen indeks dikkate alınmıştır.

$$\bar{z}(x) = \frac{\sum_{i=1}^n z_{\ell}(x)}{\sqrt{n}} \quad (3.4)$$

(3.4) eşitliğindeki  $z_{\ell}(x)$ ; x noktasındaki  $\ell$ ' ninci benzetim değerini, n ise bir model yatak içerisinde üretilen benzetim sayısını ifade etmektedir. Bu çalışmada n değeri 100 ' e eşittir. Bu durumda (3.4) eşitliği, (3.5) eşitliğine indirgenir.

$$\bar{z}(x) = \frac{\sum_{i=1}^{100} z_{\ell}(x)}{\sqrt{100}} = \frac{1}{10} \sum_{i=1}^{100} z_{\ell}(x) \quad (3.5)$$

Her bir yöntem ile farklı yapısal uzaklıklarda üretilen benzetim değerlerini toplayıp 10 'a böldüğünde üç yöntem için ortalama bir  $z(x)$  değeri bulunur. Bu değerler kullanılarak, normlanmış toplamların deneysel variogramları hesaplanır. Normlanmış toplamlardan hesaplanan deneysel variogramlar, model variogram ile karşılaştırılarak hangi yöntem ile üretilen deneysel variogramın model variograma daha yakın dolayısıyla bağımsız olduğu incelenebilir.

### 3.3.5. KS Testi

Yöntemlerin performanslarını karşılaştırmak amacıyla son olarak dikkate alınan ölçüt KS testidir. Üretilen koşullu ve koşulsuz benzetimlerin normal dağılıma sahip olup olmadığını, bir model yatak içerisinde hangi noktalarda ve kaç noktada normal dağılımdan sapıldığını belirlemek amacıyla KS testi uygulanmıştır. KS testinin uygulama aşamaları aşağıda verilen adımlardan oluşmaktadır.

İlk olarak 100 benzetim değeri küçükten büyüğe doğru sıralanır.

- 100 benzetim değerinin ortalaması ( $\mu$ ) ve standart sapması ( $\sigma$ ) hesaplanır.
- Benzetim değerleri (3.6) eşitliği ile standartlaştırılır.

$$z_i = \frac{s_i - \mu}{\sigma} \quad (3.6)$$

(3.6) eşitliğindeki  $s_i$ , i ' ninci benzetim değeridir.

- $z_i$  ye karşılık gelen kuramsal birikimli dağılım fonksiyonu hesaplanır.

$$F_{0_i} = F(z_i) = \begin{cases} 0,5 * (1 + \sqrt{(1 - \text{üs}(-2 * z_i * z_i / 3,1416))}) & , z_i > 0 \\ 1 - F_{0_i} & , z_i < 0 \end{cases} \quad (3.7)$$

- Teorik dağılım belirlendikten sonra benzetimin  $F_m$  deneysel dağılımı (3.8) eşitliğinde verilen ifade ile hesaplanır.

$$F_{m_i} - 1 = \frac{i-1}{n} \quad i = 1, \dots, 100 \quad (3.8)$$

(3.8) eşitliğindeki  $n$ ; bir model yatak içerisindeki benzetim sayısını göstermektedir.

- KS testi bir uzaklık testi olduğu için  $|F_0 - F_m|$ , teorik ve deneysel dağılım arasındaki maksimum uzaklık belirlenir. Bu amaçla (3.9) ve (3.10) eşitliklerinde verilen  $D+$  ve  $D-$  ifadeleri hesaplanır.

$$D+ = F_{m_i} - F_{0_i} \quad (3.9)$$

$$D- = F_{0_i} - (F_{m_i} - 1) \quad (3.10)$$

Maksimum uzaklık ise (3.11) eşitliği kullanılarak hesaplanır.

$$D_{\max} = \max (D+, D-) \quad (3.11)$$

- En son aşamada dağılımın normal olup olmadığını belirlemek için  $D_{\max}$  değeri (3.12) eşitliğinde verilen değişim katsayısı (D.K) ifadesi ile karşılaştırılır.

$$D.K (0.20) = \frac{1.07}{\sqrt{n}} = \frac{1.07}{\sqrt{100}} = 0.107 \quad (3.12)$$

(3.12) eşitliğindeki D.K; 0.20 düzeyindeki KS istatistik değeridir.

Eğer  $D_{\max} < D.K$  ise dağılım normal, eğer  $D_{\max} > D.K$  ise dağılım normalden farklıdır. KS testi için bir FORTRAN programı yazılmış ve EK-3 de verilmiştir.

#### 4. BENZETİM DENEYİ VE SONUÇLARI

Bir önceki bölümde verilen performans ölçütleri dikkate alınarak ayrıştırma yöntemlerinin karşılaştırması yapılmıştır. Bu amaçla ilk olarak, her bir noktada hesaplanan ortalama ve varyanslar gözönüne alınmıştır. Farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin ortalamalarına ilişkin karşılaştırmalar Şekil 4.1 de, varyanslar ise Şekil 4.2 de verilmiştir. Karşılaştırmalar 900 noktada belirlenen istatistiklerin (ortalama ve varyans) saçılma grafikleri şeklinde yapılmıştır. Şekil 4.1 de verilen, Cholesky ve bakışimli ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin ortalamaları arasında doğrusal bir ilişkinin olduğu görülmektedir. Fakat Cholesky - izgesel ve bakışimli - izgesel ayrıştırma yöntemleri ile üretilen koşulsuz benzetimlerin ortalamaları arasında herhangi bir ilişki gözlenmemiştir. Bunun yanısıra yapısal uzaklık arttıkça Cholesky ve bakışimli ayrıştırma yöntemleri arasındaki doğrusal ilişkinin derecesi azalmaktadır. Aynı şekilde Şekil 4.2 de verilen varyansların karşılaştırmasına bakıldığında, Cholesky ve bakışimli ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin varyansları arasında doğrusal bir ilişkinin olduğu görülmektedir. Fakat Cholesky - izgesel ve bakışimli - izgesel ayrıştırma yöntemleri ile üretilen koşulsuz benzetimlerin varyansları arasında herhangi bir ilişki gözlenmemiştir. Bunun yanısıra varyanslar arasındaki ilişki ortalamalar arasındaki ilişkiye oranla daha azdır. Şekil 4.3 de Cholesky ve bakışimli ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin ortalamaları ve varyansları karşılaştırılmıştır. Karşılaştırmalar koşulsuz benzetimlerde olduğu gibi 900 noktada belirlenen istatistiklerin (ortalama ve varyans) saçılma grafikleri şeklinde yapılmıştır. Koşullu benzetimlerin ortalamaları ve varyansları arasında doğrusal bir ilişki gözlenmiştir. Fakat bu ilişkinin derecesi yapısal uzaklık arttıkça azalmaktadır.

Şekil 4.4 de Cholesky, Şekil 4.5 de bakışimli ve Şekil 4.6 da izgesel ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin deneysel variogramlarının model variogramlar ile karşılaştırması verilmiştir. Karşılaştırmalar aynı bir yatağın 100 benzetim modeline ilişkin deneysel variogramları ile model variogramın aynı dağılım grafikleri üzerinde gösterilerek yapılmıştır. Şekil 4.4, 4.5 ve 4.6, deneysel variogramların model variogram çevresindeki dalgalanmaları açısından ayrıştırma yöntemleri arasında pratik bir farkın olmadığını

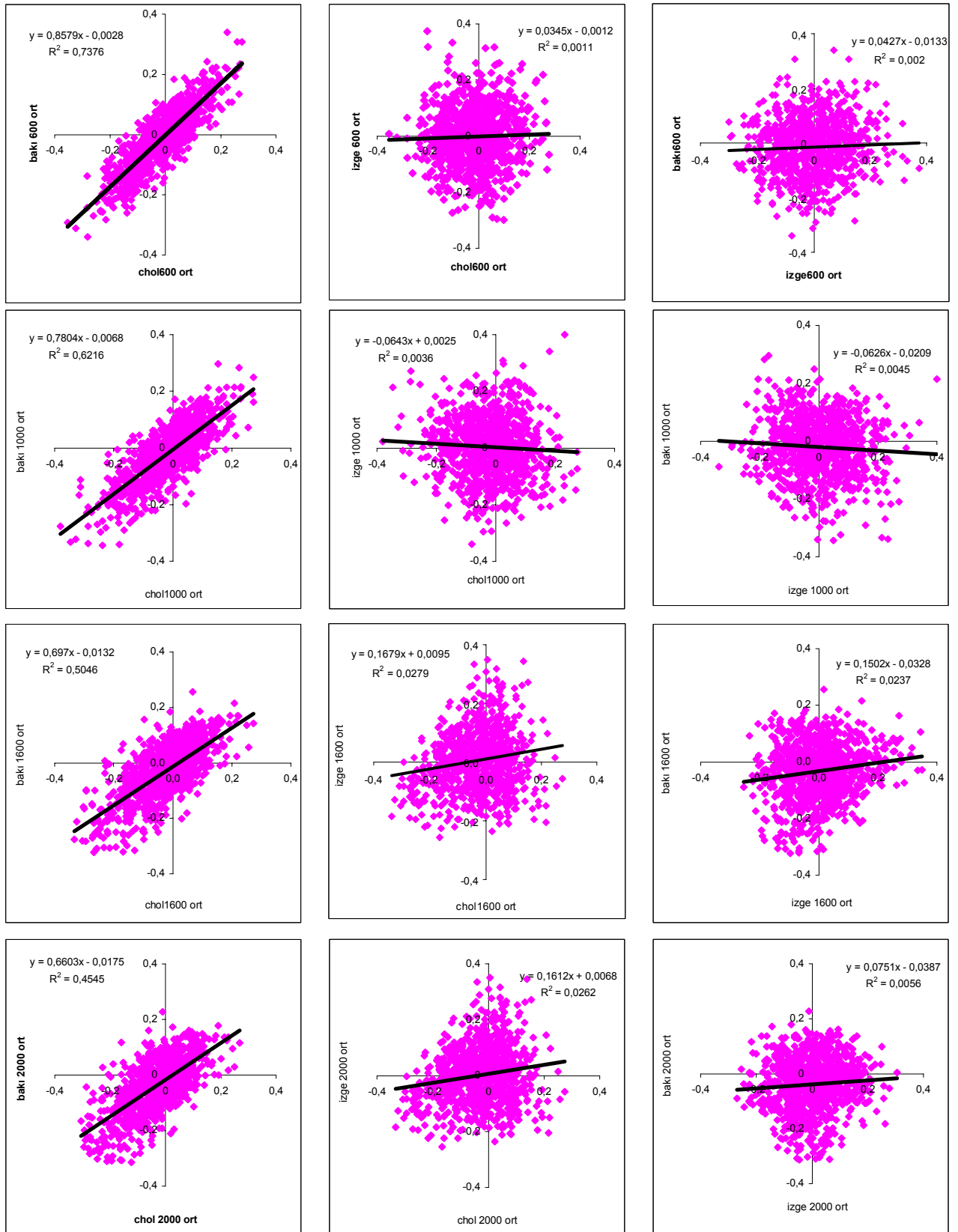
göstermektedir. Bununla birlikte yapısal uzaklık arttıkça deneysel variogramların model variogramdan sapma miktarı artmaktadır. Şekil 4.7 de Cholesky ve Şekil 4.8 de bakışimli ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin deneysel variogramlarının model variogramlar ile karşılaştırması verilmiştir. Koşulsuz benzetimlerde olduğu gibi karşılaştırmalar aynı bir yatağın 100 benzetim modeline ilişkin deneysel variogramları ile model variogramın aynı dağılım grafikleri üzerinde gösterilerek yapılmıştır. Şekil 4.7 ve Şekil 4.8 de verilen grafiklerde düşük yapısal uzaklıklarda üretilen deneysel variogramların modellerden sapma oranının birbirine yakın olduğu gözlenmiştir. Ancak yapısal uzaklık arttıkça Cholesky ayrıştırma yöntemi ile üretilen deneysel variogramın modelden daha fazla saptığı ve bakışimli ayrıştırma yöntemine göre daha yüksek değerlerde variogram ürettiği gözlenmiştir.

Şekil 4.9 da farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin yakınsama testi sonuçları verilmiştir. Üç ayrıştırma yöntemi ile üretilen benzetimlerin variogramlarının model variograma yaklaşma hızları arasında önemli bir fark yoktur. Şekil 4.10 da ise ayrıştırma yöntemleri yapısal uzaklığa göre kendi içlerinde karşılaştırılmıştır. Yapısal uzaklık arttıkça yakınsama hataları da artmaktadır. Şekil 4.11 de farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin yakınsama testi sonuçları verilmiştir. Cholesky ayrıştırma yöntemi ile üretilen koşullu benzetimlerin variogramlarının model variograma yaklaşması, bakışimli ayrıştırma yöntemine göre daha hızlı olduğu gözlenmiştir. Yapısal uzaklık arttıkça yakınsama hataları da artmaktadır. Şekil 4.12 de ayrıştırma yöntemleri yapısal uzaklığa göre kendi içlerinde karşılaştırılmıştır. Şekil 4.13 de farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin ortalama mesafe ve benzetim sayısı dikkate alındığında yakınsama testi sonuçları verilmiştir. Üç ayrıştırma yöntemi ile üretilen benzetimlerin yakınsama testi sonuçları arasında önemli bir fark yoktur. Şekil 4.14 de ise farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin ortalama mesafe ve benzetim sayısı dikkate alındığında yakınsama testi sonuçları verilmiştir. Cholesky, bakışimli ayrıştırma yöntemine göre model variograma daha hızlı yaklaşmaktadır.

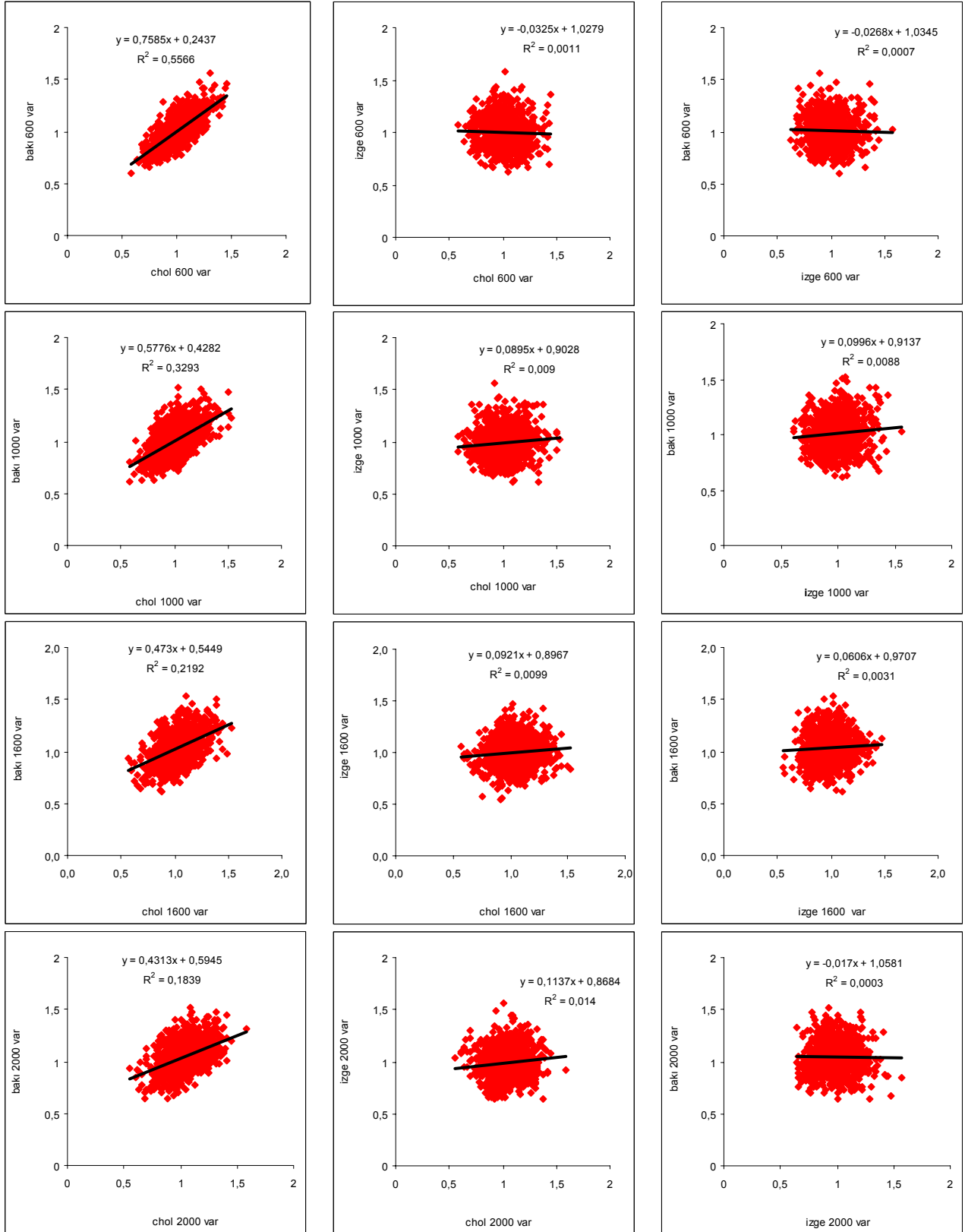
Şekil 4.15 de farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin her bir yöntem için bağımsızlık test sonuçları verilmiştir. Yapısal uzaklık arttıkça

yöntemlerin bağımlı benzetimler ürettiği gözlenmiştir. Bunun yanısıra izgesel ayrıştırma yöntemi ile üretilen benzetimlerin bağımlılığı yüksektir. Şekil 4.16 da farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin her bir yöntem için bağımsızlık test sonuçları verilmiştir. Koşullandırıcı veri kullanıldığında yöntemlerin oldukça bağımlı benzetimler ürettiği gözlenmiştir.

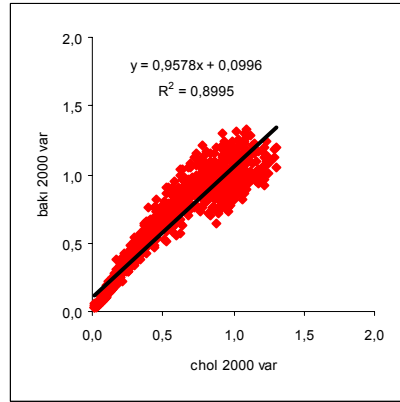
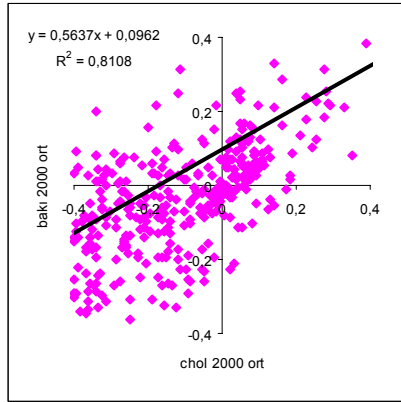
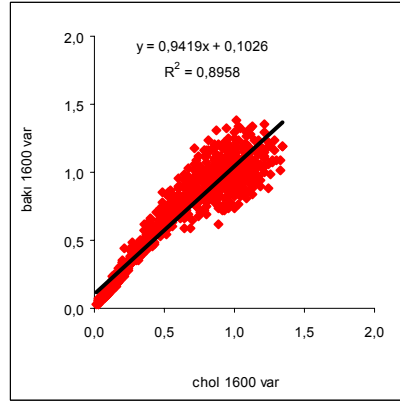
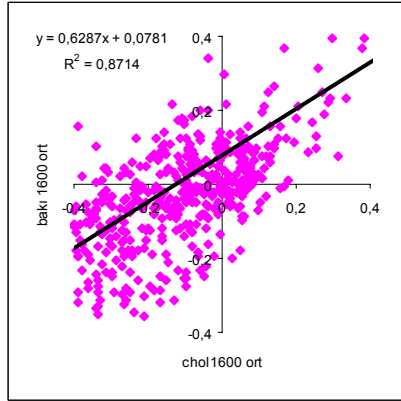
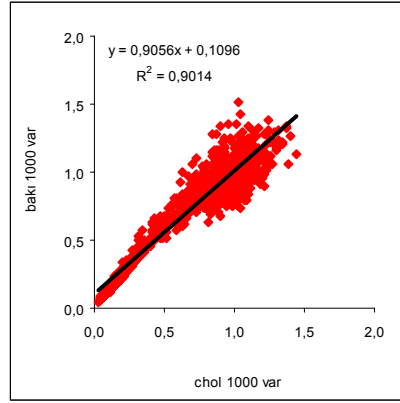
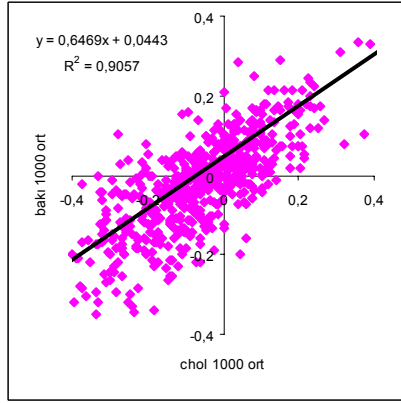
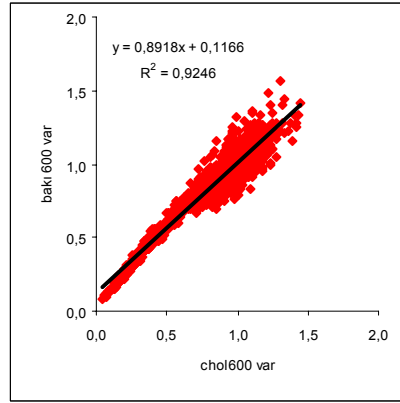
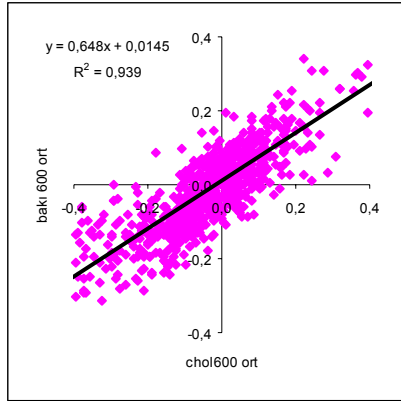
Şekil 4.17 de farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin KS testi sonuçları verilmiştir. İzgesel ayrıştırma yönteminde, yapısal uzaklık değeri arttıkça normal dağılımdan sapan nokta sayısının azaldığı gözlenmektedir. Şekil 4.18, farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin KS testi sonuçlarını vermektedir. İki yöntem arasında pratik açıdan bir fark olmadığı gözlenmiştir.



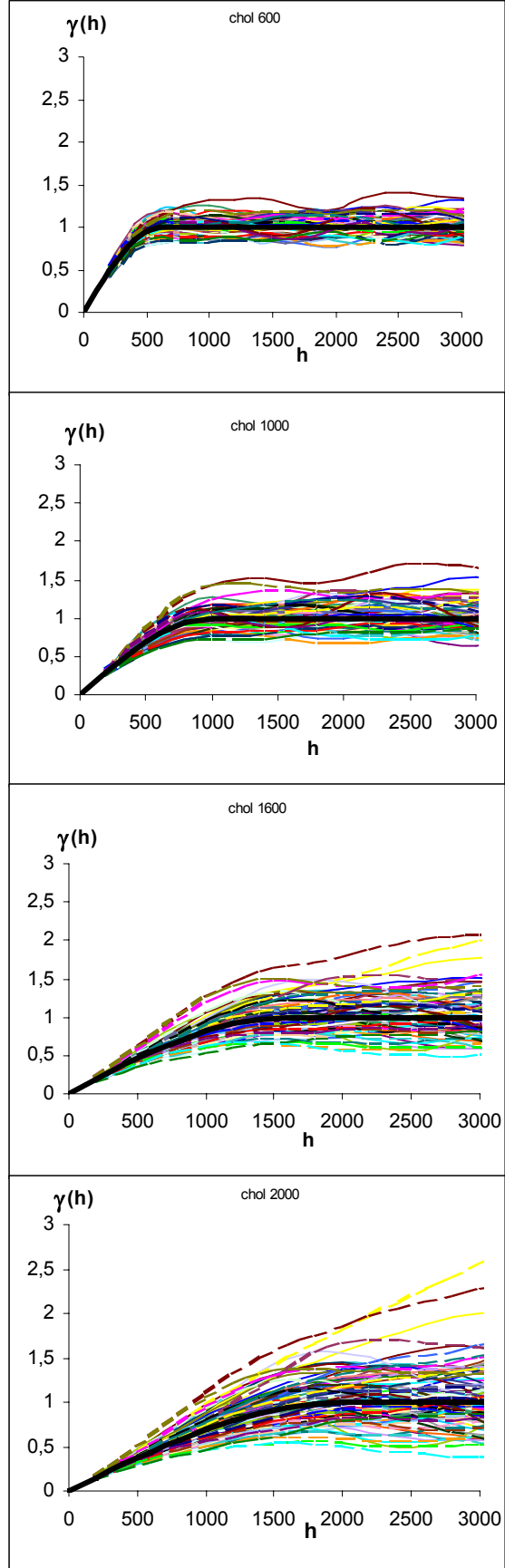
Şekil 4.1. Cholesky, bakımlı ve izgesel ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin ortalamaları arasındaki ilişki



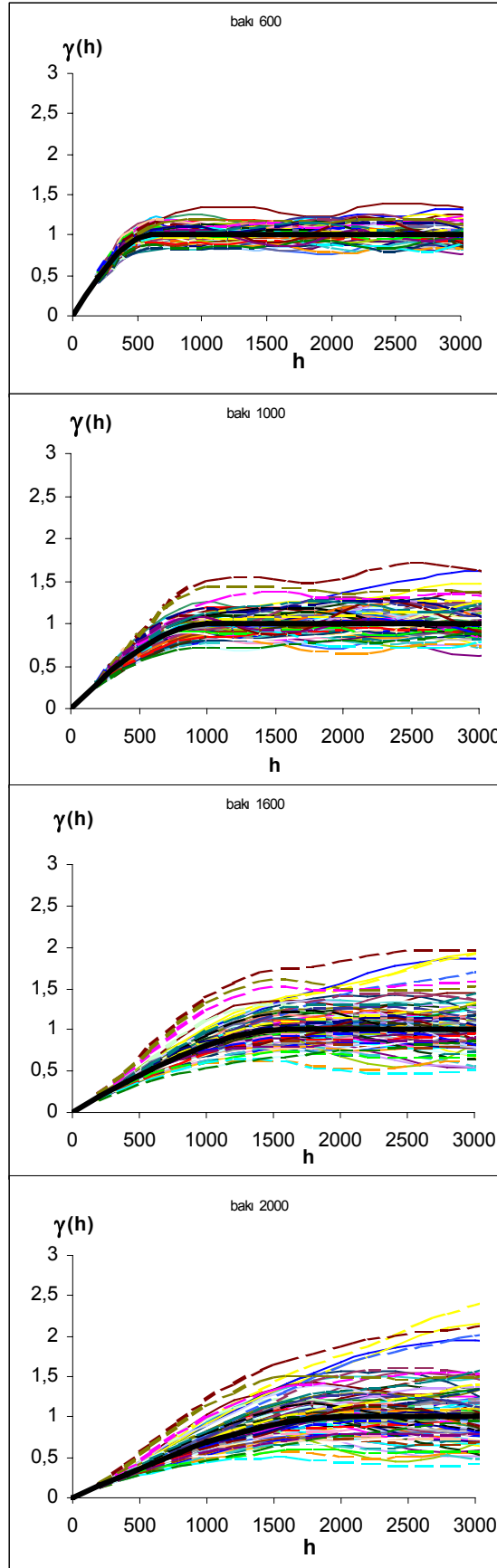
Şekil 4.2. Cholesky, bakışimli ve izgesel ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin varyansları arasındaki ilişki



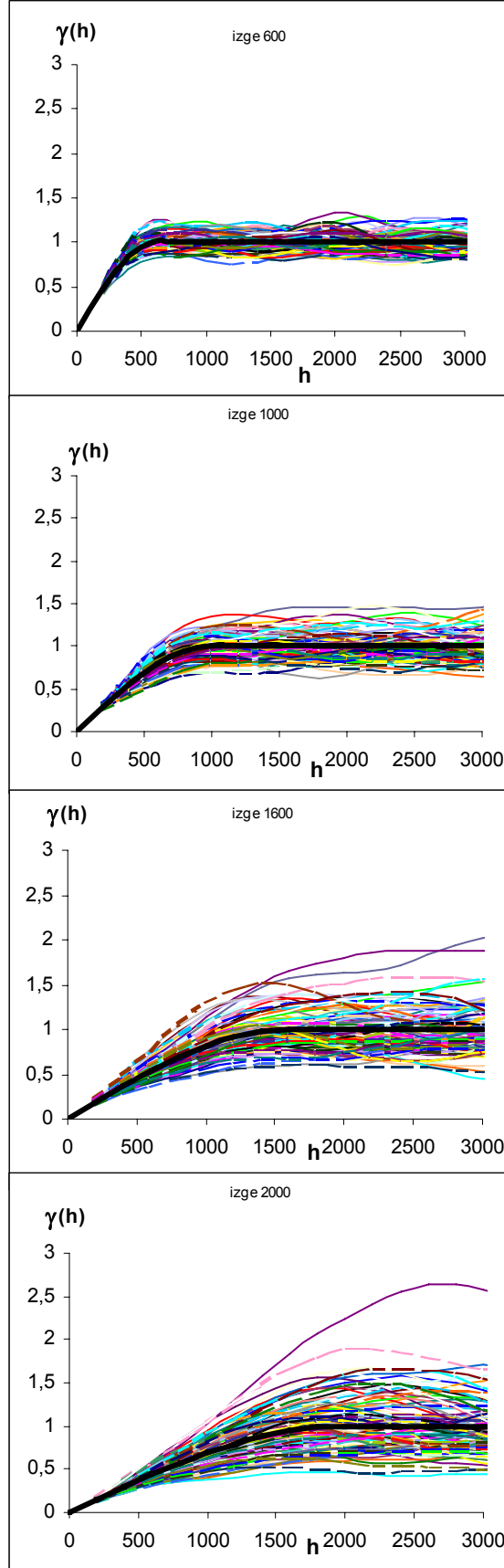
Şekil 4.3. Cholesky ve bakışlı ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin ortalamaları ve varyansları arasındaki ilişki



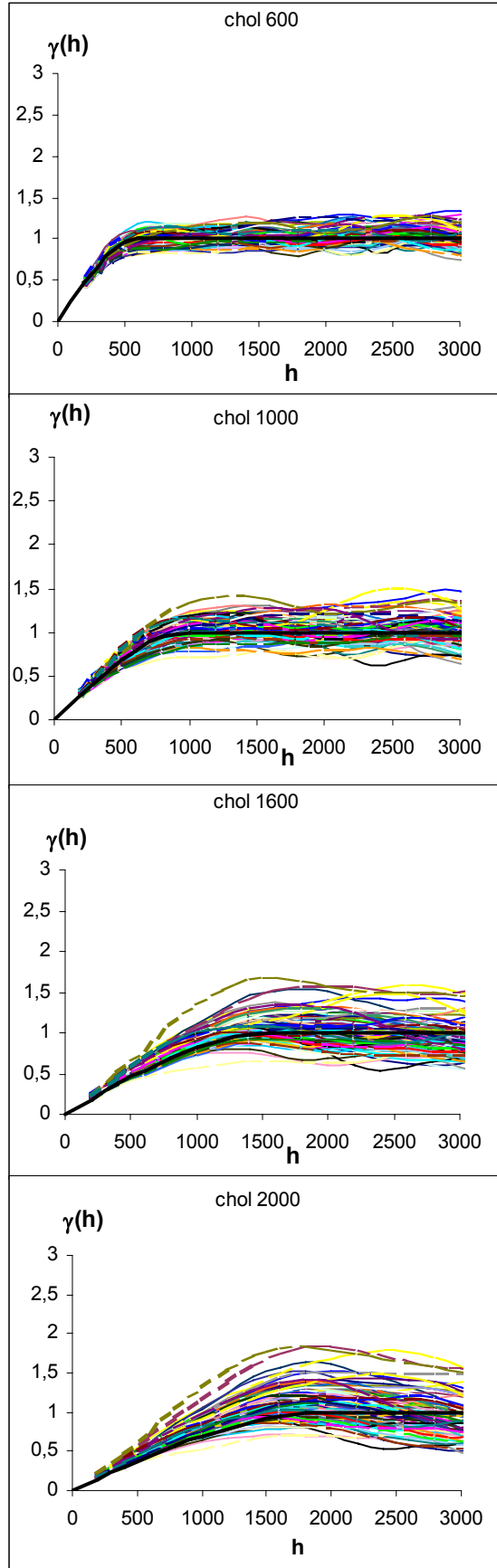
Şekil 4.4. Cholesky ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin variogramlarının model variogram ile karşılaştırması



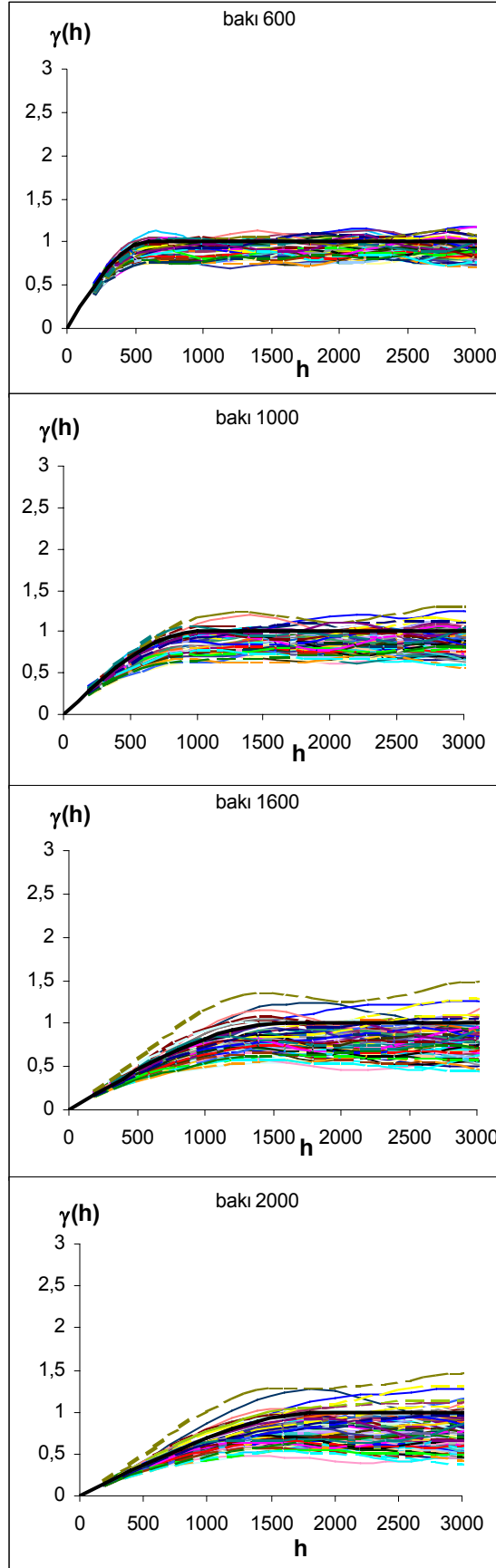
Şekil 4.5. Bakışımı ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin variogramlarının model variogram ile karşılaştırması



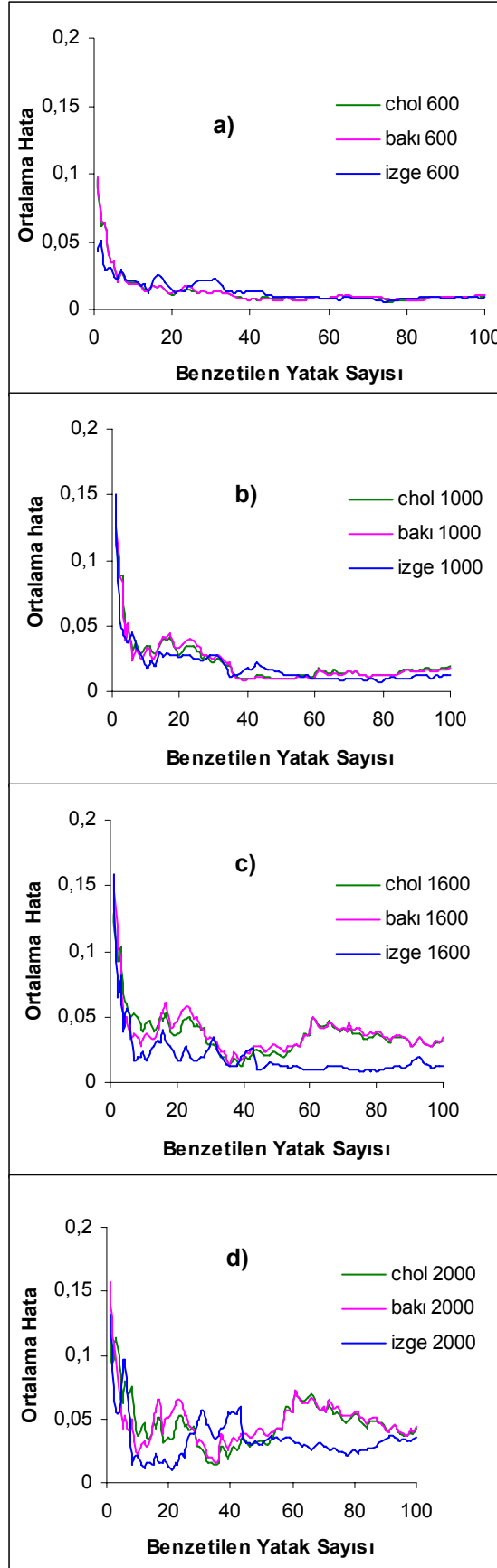
Şekil 4.6. İzgesel ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin variogramlarının model variogram ile karşılaştırması



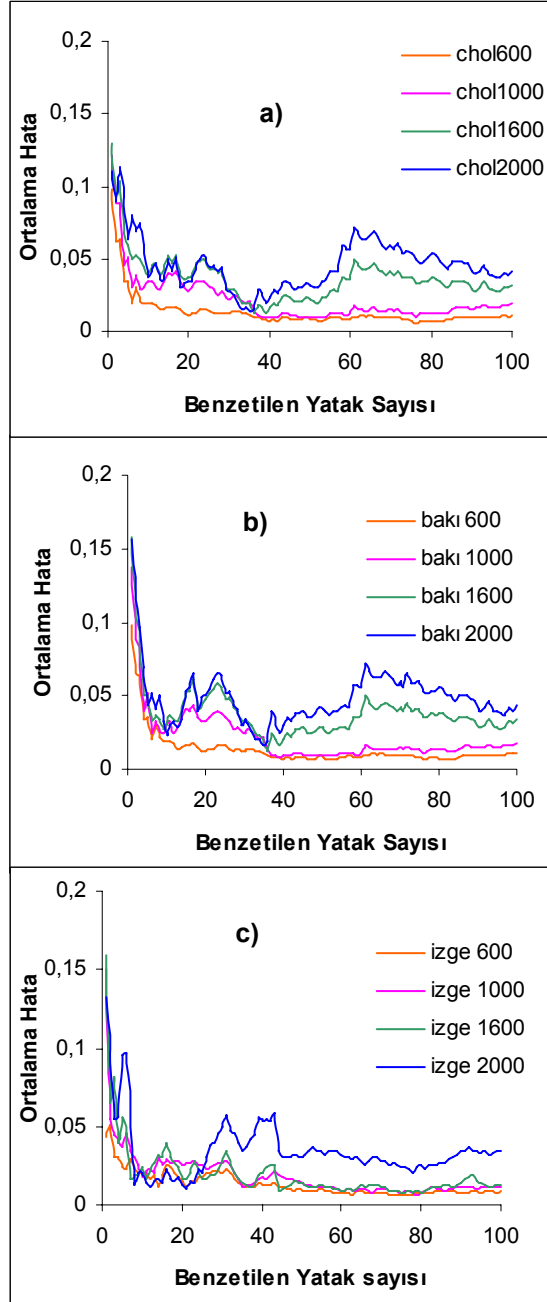
Şekil 4.7. Cholesky ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin variogramlarının model variogram ile karşılaştırması



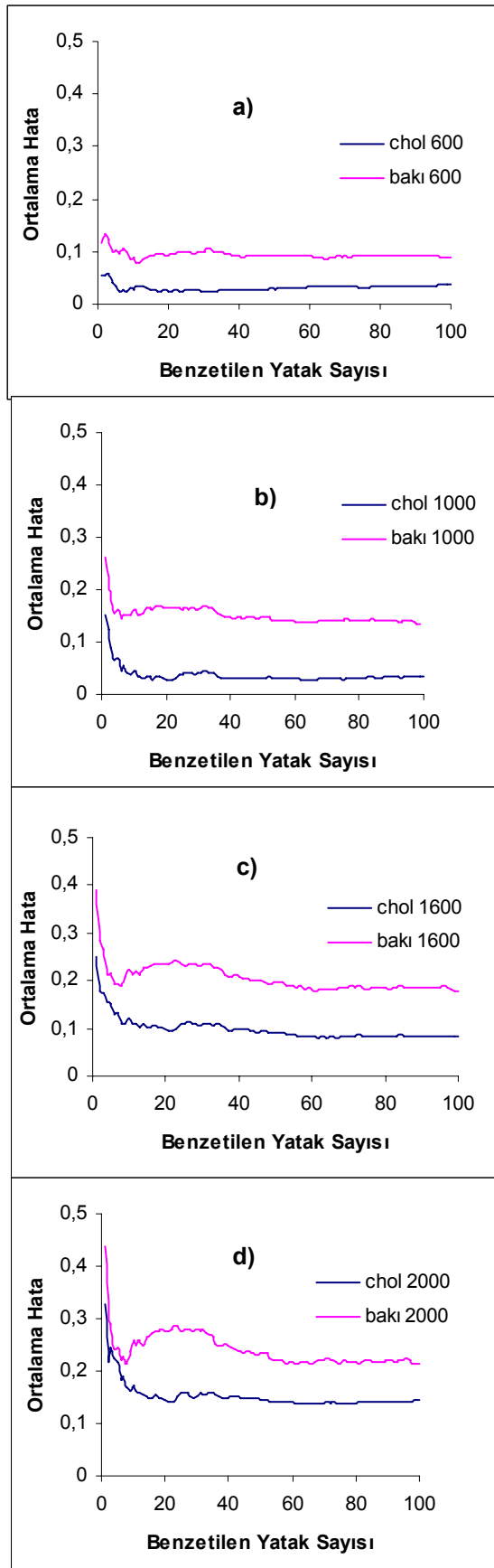
Şekil 4.8. Bakışimli ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin variogramlarının model variogram ile karşılaştırması



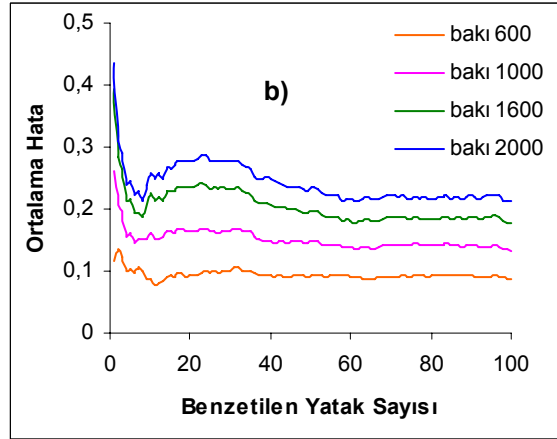
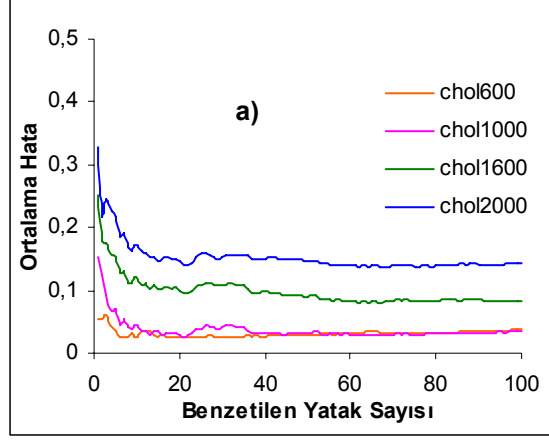
Şekil 4.9. Farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin yakınsama testi a) a=600, b) a=1000, c) a=1600, d) a=2000



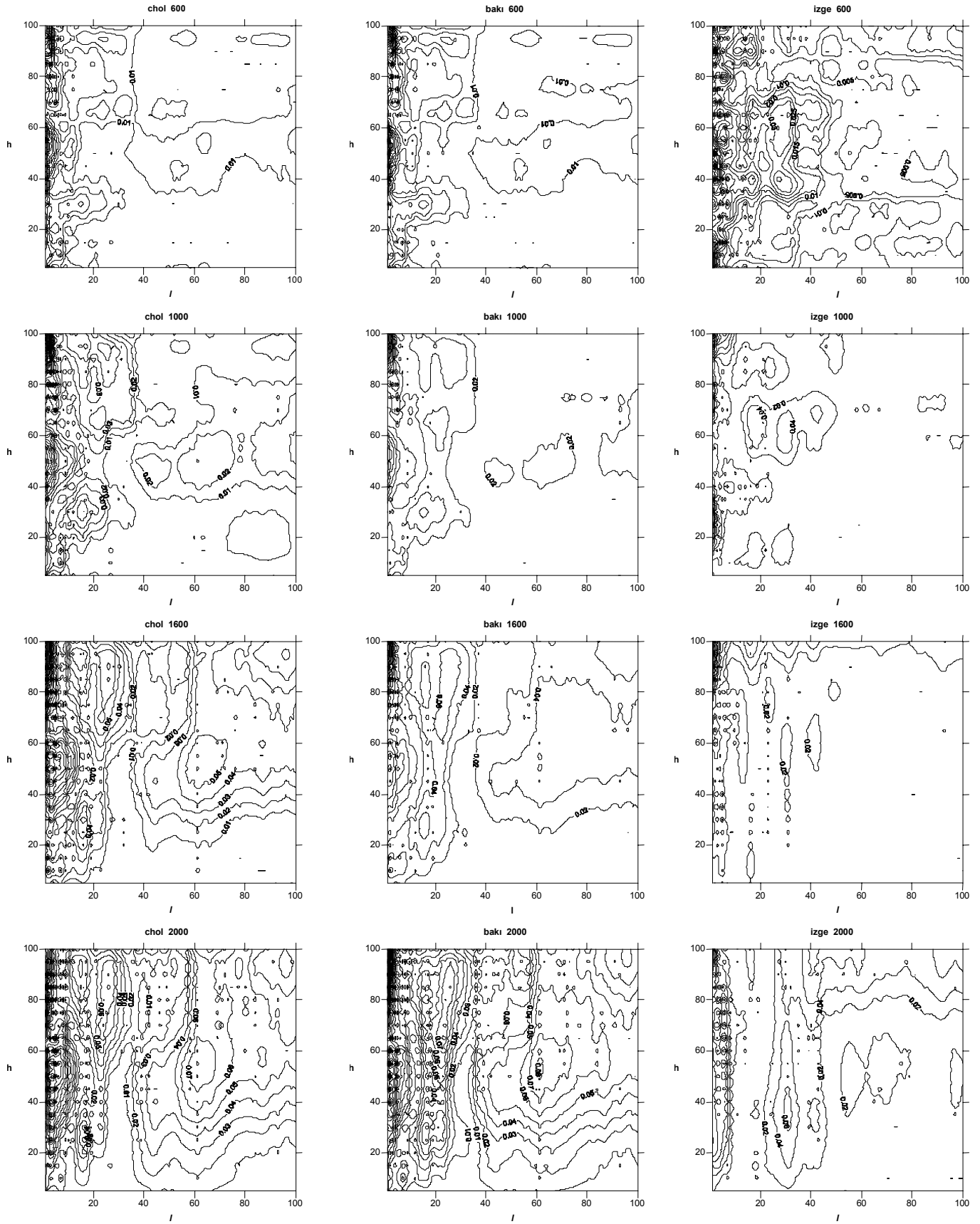
Şekil 4.10. Ayrıştırma yöntemlerinin yakınsama testine göre karşılaştırması  
a) Cholesky b) Bakışlımlı c) İzgesel



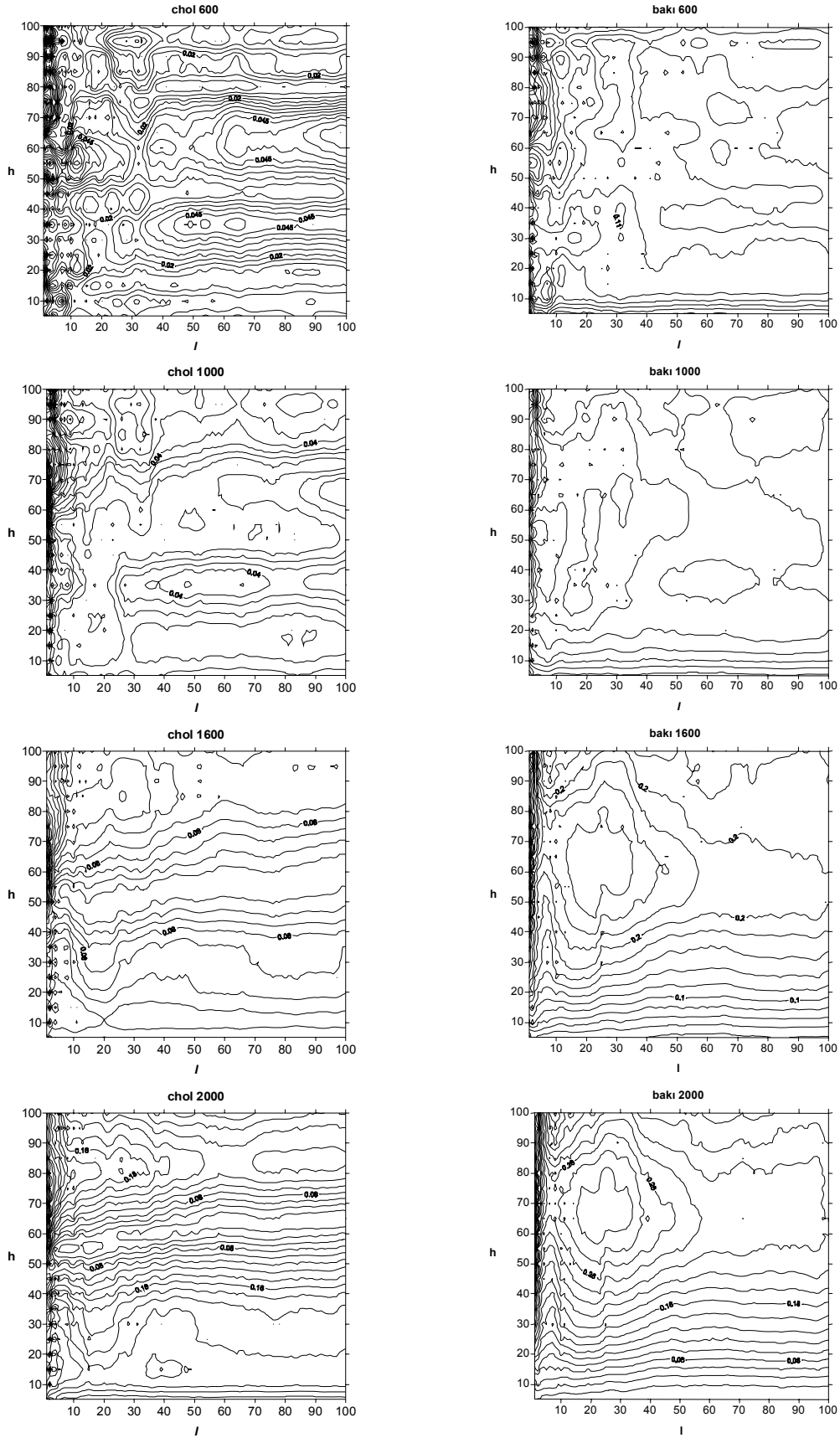
Şekil 4.11. Farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin yakınsama testi  
a) a=600, b) a=1000, c) a=1600, d) a=2000



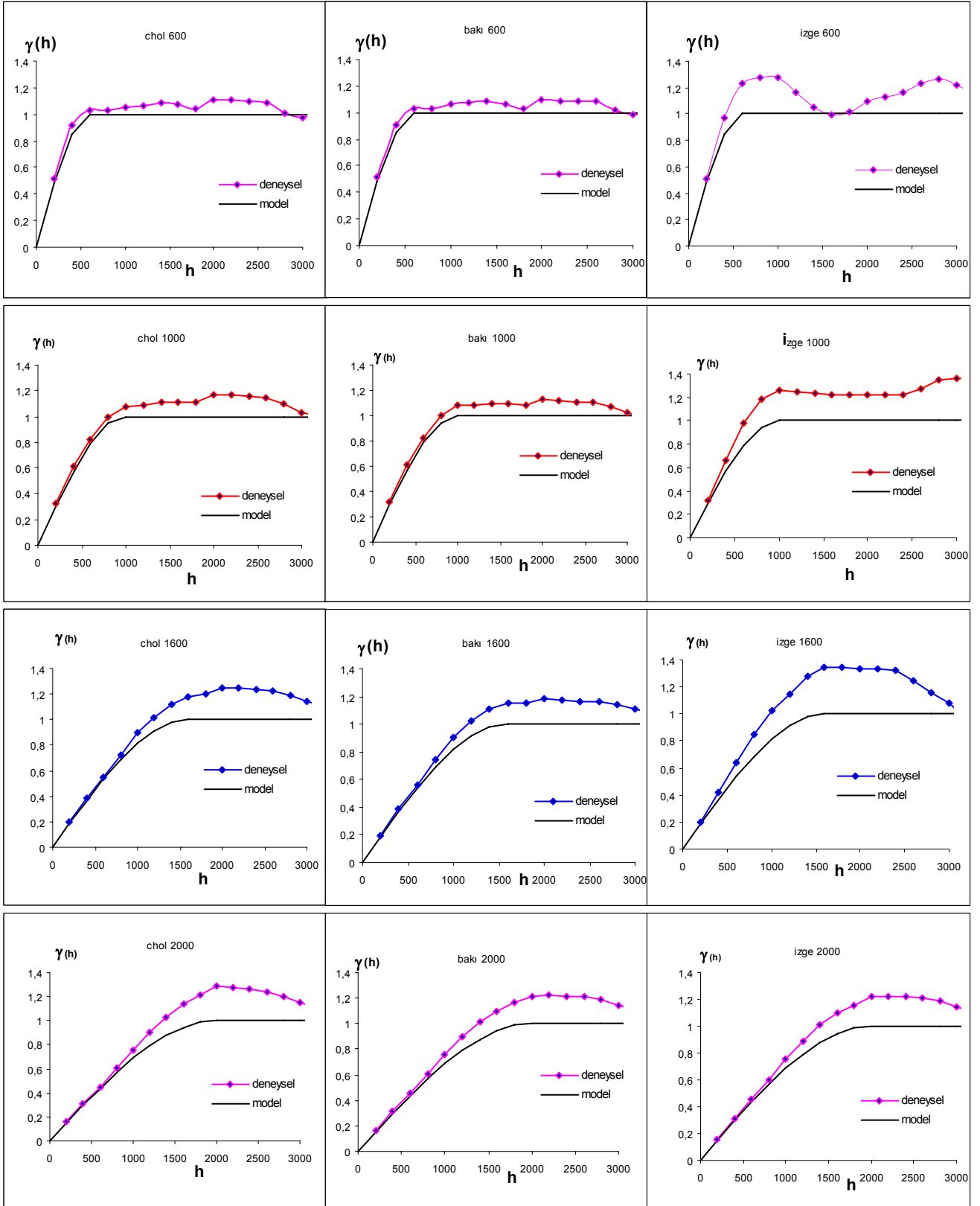
Şekil 4.12. Yapısal uzaklığın yakınsama testi üzerindeki etkisi  
a) Cholesky b) Bakışimli



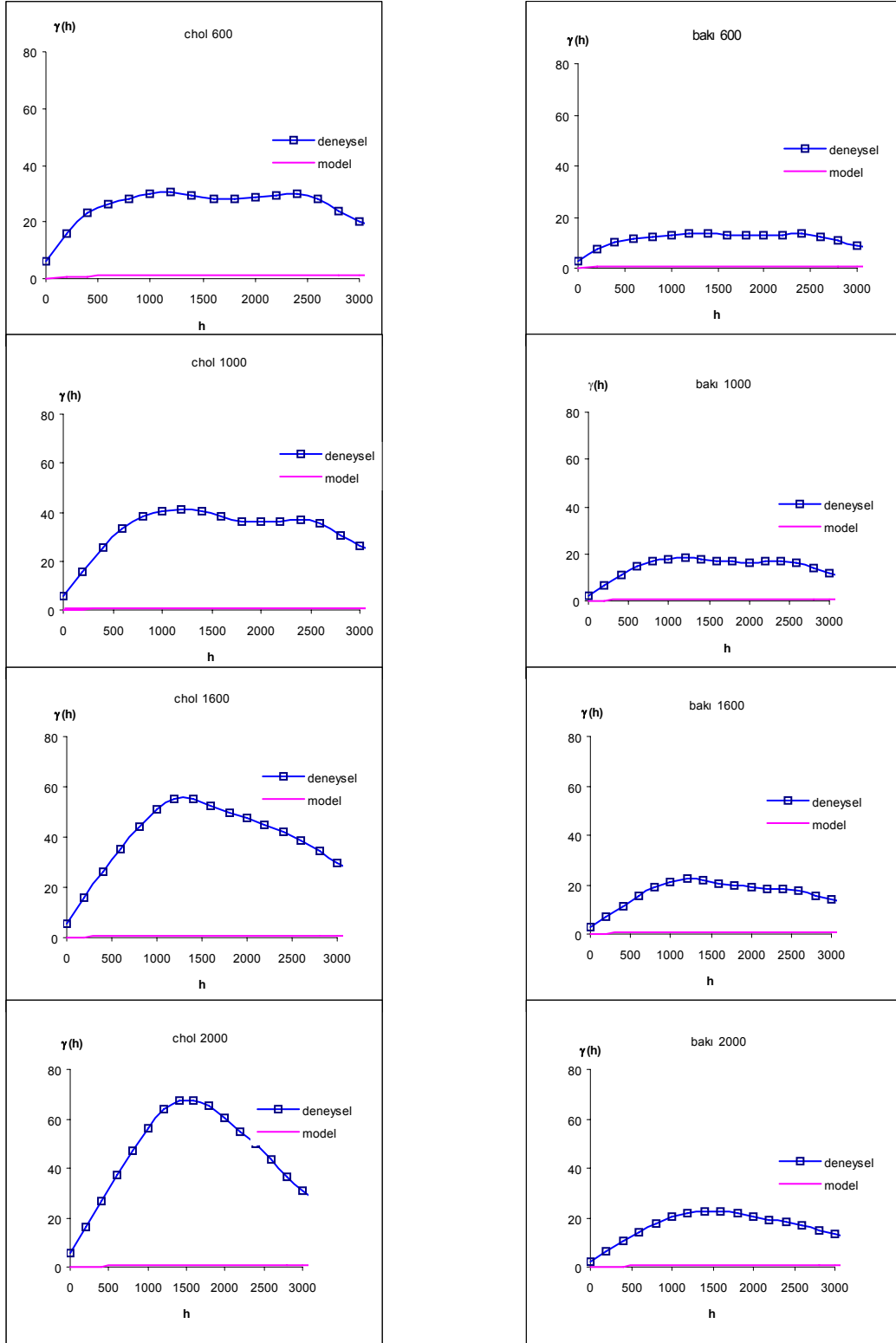
Şekil 4.13. Farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin, ortalama mesafe ve benzetim sayısı dikkate alınarak, yakınsama testi sonuçlarının karşılaştırması



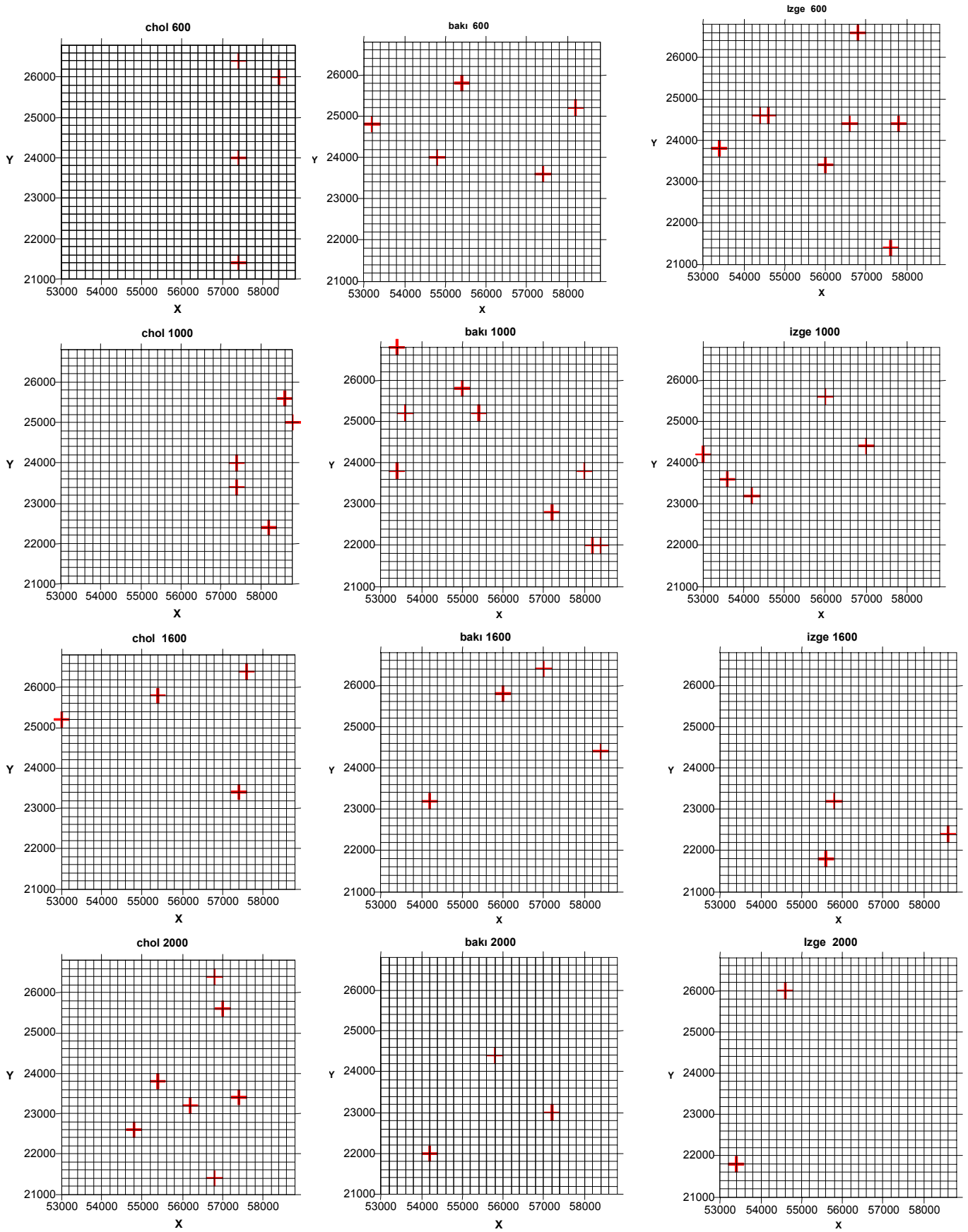
Şekil 4.14. Farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin, ortalama mesafe ve benzetim sayısı dikkate alınarak, yakınsama testi sonuçlarının karşılaştırması



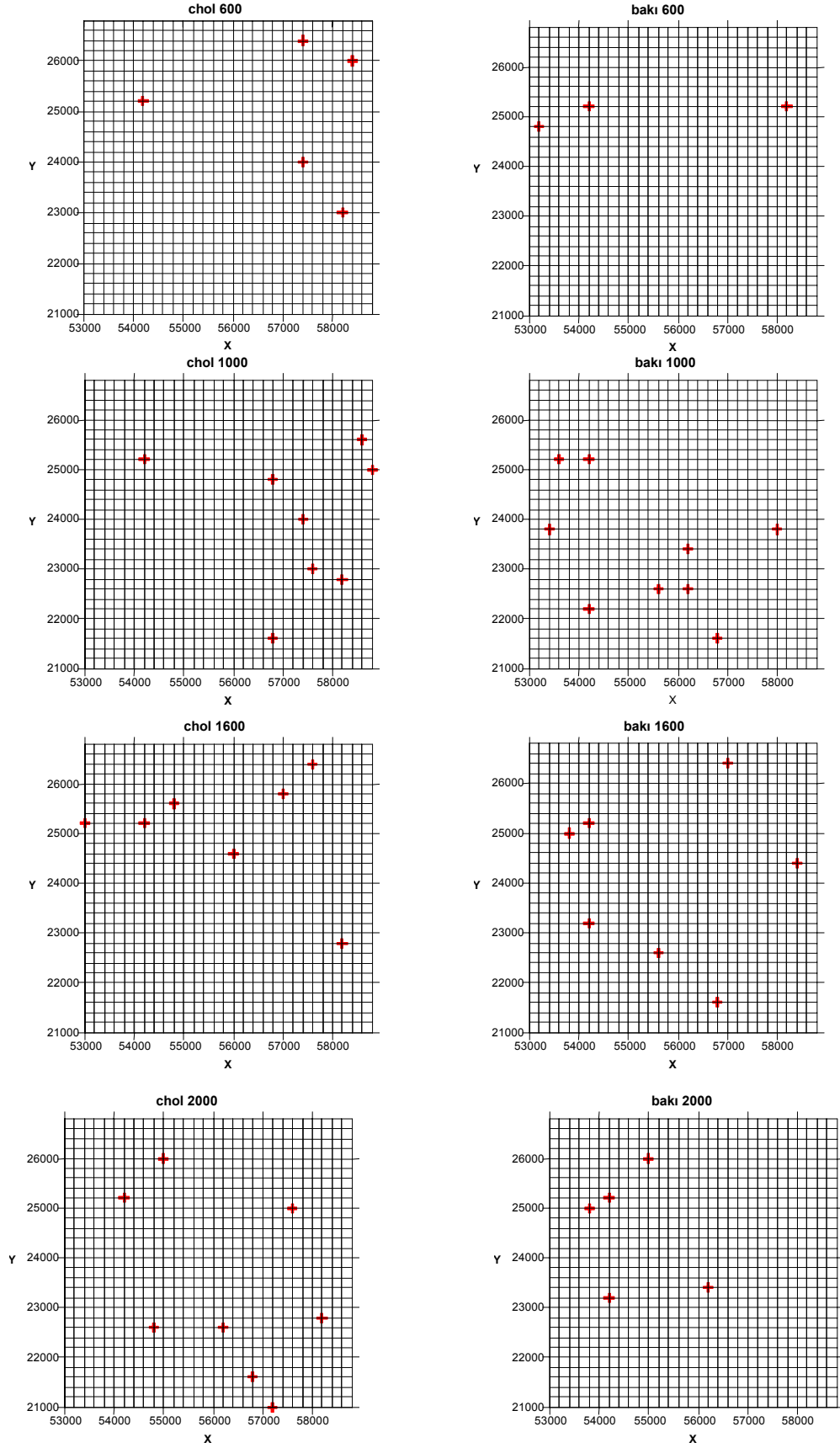
Şekil 4.15 Farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin bağımsızlık testi sonuçları



Şekil 4.16. Farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin bağımsızlık testi sonuçları



Şekil 4.17 Cholesky ve bakışimli ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin KS testi sonuçlarının karşılaştırması



Şekil 4.18. Cholesky, bakışimli ve izgesel ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin KS testi sonuçlarının karşılaştırması

## 5. SONUÇLAR

Bu çalışmada Cholesky, bakışimli ve izgesel ayrıştırma tekniklerinin LU ile üretilen benzetimler üzerindeki performansları incelenmiştir. Bu amaçla bir benzetim deneyi düzenlenmiş ve dört farklı yapısal uzaklık ( $a=600, 1000, 1600$  ve  $2000$ ) gözönüne alınarak her bir ayrıştırma tekniği için 100'er adet koşullu ve koşulsuz benzetim yapılmıştır. Daha sonra bu benzetimler, variogram üretimi, deneysel variogramların model variogramlara yaklaşma hızı ve bağımsızlık testi gibi ölçütler kullanılarak değerlendirilmiştir.

Cholesky ve bakışimli ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin ortalamaları arasında doğrusal bir ilişkinin olduğu görülmektedir. Fakat Cholesky - izgesel ve bakışimli - izgesel ayrıştırma yöntemleri ile üretilen koşulsuz benzetimlerin ortalamaları arasında herhangi bir ilişki gözlenmemiştir. Bunun yanısıra yapısal uzaklık arttıkça Cholesky ve bakışimli ayrıştırma yöntemleri arasındaki doğrusal ilişkinin derecesi azalmaktadır. Aynı şekilde varyansların karşılaştırmasına bakıldığında, Cholesky ve bakışimli ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin varyansları arasında doğrusal bir ilişkinin olduğu görülmektedir. Fakat Cholesky - izgesel ve bakışimli - izgesel ayrıştırma yöntemleri ile üretilen koşulsuz benzetimlerin varyansları arasında herhangi bir ilişki gözlenmemiştir. Bunun yanısıra varyanslar arasındaki ilişki ortalamalar arasındaki ilişkiye oranla daha azdır. Cholesky ve bakışimli ayrıştırma yöntemleri ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin ortalamaları ve varyansları arasında doğrusal bir ilişki gözlenmiştir. Fakat bu ilişkinin derecesi yapısal uzaklık arttıkça azalmaktadır.

Üç ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin deneysel variogramları model variogramlar ile karşılaştırıldığında, deneysel variogramların model variogram çevresindeki dalgalanmaları açısından ayrıştırma yöntemleri arasında pratik bir fark yoktur. Bununla birlikte yapısal uzaklık arttıkça deneysel variogramların model variogramdan sapma miktarı artmaktadır. Cholesky ve bakışimli ayrıştırma yöntemi ile farklı yapısal uzaklıklarda üretilen koşullu benzetimlerin deneysel variogramları model variogramlar ile karşılaştırıldığında, düşük yapısal uzaklıklarda üretilen deneysel variogramların modellerden sapma oranının birbirine yakın olduğu gözlenmiştir. Ancak yapısal

uzaklık arttıkça Cholesky ayrıştırma yöntemi ile üretilen deneysel variogramın modelden daha fazla saptığı ve bakışimli ayrıştırma yöntemine göre daha yüksek değerlerde variogram ürettiği gözlenmiştir.

Yakınsama testine göre üç ayrıştırma yöntemi ile üretilen benzetimlerin variogramlarının model variograma yaklaşma hızları arasında önemli bir fark yoktur. Yapısal uzaklık arttıkça yakınsama hataları da artmaktadır. Koşulsuz benzetimlerin ortalama mesafe ve benzetim sayısı dikkate alındığında, üç ayrıştırma yöntemi ile üretilen benzetimlerin yakınsama testi sonuçları arasında önemli bir fark yoktur. Cholesky ayrıştırma yöntemi ile üretilen koşullu benzetimlerin variogramlarının model variograma yaklaşması, bakışimli ayrıştırma yöntemine göre daha hızlıdır. Koşullu benzetimlerde de yapısal uzaklık arttıkça yakınsama hataları da artmaktadır. Koşullu benzetimlerin ortalama mesafe ve benzetim sayısı dikkate alındığında Cholesky, bakışimli ayrıştırma yöntemine göre model variograma daha hızlı yaklaşmaktadır.

Farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin her bir yöntem için uygulanan bağımsızlık testine göre, yapısal uzaklık arttıkça yöntemlerin bağımlı benzetimler ürettiği gözlenmiştir. Bunun yanısıra izgesel ayrıştırma yöntemi diğerlerine göre daha bağımlı benzetimler üretmektedir. Koşullu benzetimler için uygulanan bağımsızlık testine göre, koşullandırıcı veri kullanıldığında yöntemlerin oldukça bağımlı benzetimler üretmektedir. Bu koşullu benzetimin en önemli dezavantajıdır. Çünkü jeostatistiksel benzetimlerde üretilen benzetimlerin birbirinden bağımsız olması arzu edilir.

Farklı yapısal uzaklıklarda üretilen koşulsuz benzetimlerin KS testi sonuçlarına göre izgesel ayrıştırma yönteminde, yapısal uzaklık değeri arttıkça normal dağılımdan sapan nokta sayısının azalmaktadır. Koşullu benzetimlerin KS testi sonuçlarında ise iki yöntem arasında pratik açıdan bir fark yoktur.

Üretilen benzetimlere dayanarak üç yöntemin performansı karşılaştırıldığında, izgesel ayrıştırma yönteminin yalnızca normalliğin önemli olduğu durumlarda kullanılabileceği görülmüştür. Çünkü yapısal uzaklık değeri arttıkça normal dağılımdan sapan nokta sayısının azalması sadece izgesel ayrıştırma yöntemi ile üretilen koşulsuz benzetimlerde gözlenmiştir. Cholesky ya da bakışimli ayrıştırma

yöntemlerinden hangisinin tercih edileceđi dikkate alınacak ölçüte bađlıdır. Örneđin koşulsuz benzetimle variogramın yeniden üretimi dikkate alındığında Cholesky ve bakışimli ayrıştırma yöntemleri arasında pratik açıdan bir fark yoktur. Fakat benzetimler koşullu olduğunda Cholesky ayrıştırma yöntemi ile üretilen benzetimlerin variogramları yapısal uzaklık arttıkça modelden daha çok sapmaktadır. Dolayısıyla koşullu benzetimler de variogram bir ölçüt olarak dikkate alınıyorsa bakışimli ayrıştırma yöntemini kullanmak daha uygundur. Koşullu benzetimler üretirken yakınsaklık önemli ise, Cholesky ayrıştırma yöntemini kullanmak gerekir. Çünkü Cholesky ayrıştırma yöntemi ile üretilen koşullu benzetimlerin variogramlarının model variograma yaklaşması, bakışimli ayrıştırma yöntemine göre daha hızlıdır.

## KAYNAKLAR

- Alabert, F., 1987, The Practice of Fast Conditional Simulations Through the LU Decomposition of the Covariance Matrix, *Mathematical Geology*, Vol. 19, No. 5, 369-386
- Bourgault, G., 1997, Using Non-Gaussian Distributions in Geostatistical Simulations, *Mathematical Geology*, Vol. 29, No. 3, 315-334
- Davis, M. W., 1987a, Production of Conditional Simulations via the LU Triangular Decomposition of the Covariance Matrix, *Mathematical Geology*, Vol. 19, No.2, 91-98
- Davis, M. W., 1987b, Generating Large Stochastic Simulations- The Matrix Polynomial Approximation Method, *Mathematical Geology*, Vol. 19, No.2, 99-107
- De iaco, S. and Palma, M., 2002, Convergence of Realization-Based Statistics to Model-Based Statistics for the LU Unconditional Simulation Algorithm: Some Numerical Tests, *Stochastic Environmental Research and Risk Assessment*, Lecce, Italy, 333-341
- Deutsch, C. V., and Journel, A. G., 1998, *GSLIB: Geostatistical Software Library and User's Guide*, Oxford University Press, New York, 370
- Dietrich, C. R., and Newsam, G. N., 1995, Efficient Generation of Conditional Simulations by Chebyshev Matrix Polynomial Approximations to the Symmetric Square Root of the Covariance Matrix, *Mathematical Geology*, Vol.27, No. 2, 207-228
- Dowd, P. A. and Saraç, C., 1993, An Extension Of The LU Decompositon Method Of Simulation, *Geostatistical Simulations*, Kluwer Academic Publ., Netherlands, 23-36
- Golub, G. H. And Van Loan, C. F., 1991, *Matrix Computations*, The Johns Hopkins University Press, London, 642 p.
- Matheron, G., 1973, The Intrinsic Random Functions and Their Applications, *Advances in Applied Probability*, 5, 439-468
- Vargas-Guzmán, J. A. and Dimitrakopoulos, R., 2002, Conditional Simulation of Random Fields by Successive Residuals, *Mathematical Geology*, Vol.34, No. 5, 597-611
- Probability and Statistics ▶ Statistical Distributions ▶ Limit Theorems ▼  
<http://mathworld.wolfram.com/CentralLimitTheorem.html>

**EK – 1**

**LUSIM ve ALT PROGRAMLARININ FORTRAN KODLARI**

```

    program main
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
C Copyright (C) 1996, The Board of Trustees of the Leland Stanford
%
C Junior University. All rights reserved.
%
C The programs in GSLIB are distributed in the hope that they will be
%
C useful, but WITHOUT ANY WARRANTY. No author or distributor accepts
%
C responsibility to anyone for the consequences of using them or for
%
C whether they serve any particular purpose or work at all, unless he
%
C says so in writing. Everyone is granted permission to copy, modify
%
C and redistribute the programs in GSLIB, but only under the condition
%
C that this notice and the above copyright notice remain intact.
%
C Original: Deutsch C.V. and Journel A.G., 1998 revised: Tercan A.E., 2004
%
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
C-----
C
C      Conditional Simulation of a 3-D Rectangular Grid
C      *****
C
c This is a template driver program for GSLIB's "lusim" subroutine.
c 2-D realizations of a Gaussian process with a given autocovariance
c model and conditional to input Gaussian data are created. The
c conditional simulation is achieved by one matrix inversion using
c the Cholesky decomposition technique described in M. Davis, Math.
c Geol. 19(2), 1987 , and in F. Alabert , Math. Geol. 19(5) 1987
c
c The program is executed with no command line arguments. The user
c will be prompted for the name of a parameter file. The parameter
c file is described in the documentation (see the example lusim.par)
c and should contain the following information:
c
c - Name of the data file (GEOEAS format)
c - column numbers for x, y, wt, and variable (if no wt set to 0)
c - Data trimming limits
c - An output file (may be overwritten)
c - debugging level
c - debugging output file
c - Random Number Seed
c - The number of simulations

```

- c - X grid definition (number, minimum, size): nx,xmn,xsiz
- c - Y grid definition (number, minimum, size): ny,ymn,ysiz
- c - Z grid definition (number, minimum, size): nz,zmn,zsiz
- c - Variogram Definition: number of structures, and nugget effect
- c - The next "nst" lines requires the following (see manual):
  - c a) an integer code specifying the variogram type:
  - c b) the "a" parameter for the structure.
  - c b) the "c" parameter for the structure.
  - c c) three angles
  - c d) two anisotropy ratios
- c
- c
- c The output file will be a GEOEAS file containing the simulated values
- c The file is ordered by x and then y then simulation (i.e., x cycles
- c fastest, then y, then simulation number).

```

c
c-----
      include 'lusim.inc'
c
c Read the Parameter File and the Data:
c
      call readparm
c
c Call lusim for the simulation:
c
      call lusim
c
c Finished:
c
      close(lout)
      close(ldbg)
      write(*,9998) VERSION
9998 format(/' LUSIM Version: ',f5.3, ' Finished'/)
      stop
      end

```

```

      subroutine readparm
c-----
c
c          Initialization and Read Parameters
c          *****
c
c The input parameters and data are read in from their files. Some quick
c error checking is performed and the statistics of all the variables
c being considered are written to standard output.
c
c-----
      include 'lusim.inc'
      character str*40

```

```

    real    var(20)
    real*8  acorni
    logical testfl
c
c Input/Output Units:
c
    lin = 1
    ldbg = 3
    lout = 4
c
c Note VERSION number:
c
    write(*,9999) VERSION
    9999 format(/' LUSIM Version: ',f5.3/)
c
c Get the name of the parameter file - try the default name if no input:
c
    write(*,*) 'Which parameter file do you want to use?'
    read (*,'(a40)') str
    if(str(1:1).eq.' ')str='lusim.par'
    inquire(file=str,exist=testfl)
    if(.not.testfl) then
        write(*,*) 'ERROR - the parameter file does not exist,'
        write(*,*) '      check for the file and try again '
        write(*,*)
        if(str(1:20).eq.'lusim.par' ) then
            write(*,*) '      creating a blank parameter file'
            call makepar
            write(*,*)
        end if
        stop
    endif
    open(lin,file=str,status='OLD')
c
c Find Start of Parameters:
c
1   read(lin,'(a4)',end=98) str(1:4)
    if(str(1:4).ne.'STAR') go to 1
c
c Read Input Parameters:
c
    read(lin,'(a40)',err=98) datafl
    call chknam(datafl,40)
    write(*,*) ' data file = ',datafl

    read(lin,*,err=98) ix,iy,iz,ivr
    write(*,*) ' columns = ',ix,iy,iz,ivr

    read(lin,*,err=98) tmin,tmax
    write(*,*) ' trimming limits = ',tmin,tmax

```

```

read(lin,*,err=98) idbg
write(*,*) ' debugging option = ',idbg

read(lin,'(a40)',err=98) dbgfl
call chknam(dbgfl,40)
write(*,*) ' debug file = ',dbgfl

read(lin,'(a40)',err=98) outfl
call chknam(outfl,40)
write(*,*) ' output file = ',outfl

read(lin,*,err=98) nsim
write(*,*) ' number of simulations = ',nsim

read(lin,*,err=98) nx,xmn,xsiz
write(*,*) ' nx, xmn, xsiz = ',nx,xmn,xsiz

read(lin,*,err=98) ny,ymn,ysiz
write(*,*) ' ny, ymn, ysiz = ',ny,ymn,ysiz

read(lin,*,err=98) nz,zmn,zsiz
write(*,*) ' nz, zmn, zsiz = ',nz,zmn,zsiz

read(lin,*,err=98) ixv(1)
write(*,*) ' random number seed = ',ixv(1)
do i=1,1000
    p = real(acorni(idum))
end do

read(lin,*,err=98) nst(1),c0(1)
write(*,*) ' nst, c0 = ',nst(1),c0(1)

if(nst(1).le.0) then
    write(*,9997) nst(1)
9997    format(' nst must be at least 1, it has been set to ',i4,/,
+         ' The c or a values can be set to zero')
    stop
endif
do i=1,nst(1)
    read(lin,*,err=98) it(i),cc(i),ang1(i),ang2(i),ang3(i)
    read(lin,*,err=98) aa(i),aa1,aa2
    anis1(i) = aa1 / max(aa(i),EPSLON)
    anis2(i) = aa2 / max(aa(i),EPSLON)
    write(*,*) ' it, aa, cc = ',it(i),aa(i),cc(i)
    write(*,*) ' ang1,2,3, anis1,2 = ',ang1(i),ang2(i),
+         ang3(i),anis1(i),anis2(i)
    if(it(i).eq.4) stop 'Power model not allowed'
end do
close(lin)

```

```

c
c Perform some quick error checking:
c
  if(nx.gt.MAXX) stop 'nx is too big - modify .inc file'
  if(ny.gt.MAXY) stop 'ny is too big - modify .inc file'
  if(nz.gt.MAXZ) stop 'nz is too big - modify .inc file'
  open(ldbg,file=dbgfl,status='UNKNOWN')
  write(ldbg,100)
100 format(/,'LUSIM Debugging file',/)
  open(lout,file=outfl,status='UNKNOWN')
  write(lout,101)
101 format('LUSIM Output',/, '1',/, 'simulated values')
c
c Check to make sure the data file exists, then either read in the
c data or write an error message and stop:
c
  inquire(file=datafl,exist=testfl)
  nd = 0
  if(.not.testfl) then
    itrans = 1
    write(*,*) 'WARNING data file ',datafl,' does not exist!'
    write(*,*) '      creating unconditional simulations.'
    return
  endif
c
c The data file exists so open the file and read in the header
c information. Initialize the storage that will be used to summarize
c the data found in the file:
c
  open(lin,file=datafl,status='OLD')
  read(lin,'(a40)',err=99) str
  read(lin,*,err=99)    nvari
  av = 0.0
  ss = 0.0
  do i=1,nvari
    read(lin,*,err=99)
  end do
c
c Read all the data until the end of the file:
c
2  read(lin,*,end=3,err=99) (var(j),j=1,nvari)
  vrt = var(ivr)
  if(vrt.lt.tmin.or.vrt.ge.tmax) go to 2
  nd = nd + 1
  if(nd.gt.MAXDAT) then
    write(*,*) ' ERROR: number of data exceed MAXDAT',MAXDAT
    stop
  end if
  if(ix.ge.1) then
    x(nd) = var(ix)

```

```

else
    x(nd) = xmn
end if
if(iy.ge.1) then
    y(nd) = var(iy)
else
    y(nd) = ymn
end if
if(iz.ge.1) then
    z(nd) = var(iz)
else
    z(nd) = zmn
end if
vr(nd) = vrt
c
c Check for co-located:
c
do i=1,nd-1
    test = abs(x(nd)-x(i)) + abs(y(nd)-y(i)) + abs(z(nd)-z(i))
    if(test.le.EPSLON) then
        write(*,*) ' ERROR: co-located data ',i,nd
        stop
    end if
end do
av = av + vrt
ss = ss + vrt*vrt
go to 2
3 close(lin)
c
c Compute the averages and variances as an error check for the user:
c
av = av / max(real(nd),1.0)
ss =(ss / max(real(nd),1.0)) - av * av
write(*,102) ivr,nd,av,ss
write(lDBG,102) ivr,nd,av,ss
102 format(/,'Data for LUSIM: Variable number ',i2,/,
+      ' Number = ',i4,/,
+      ' Average = ',f12.4,/,
+      ' Variance = ',f12.4)
return
c
c Error in an Input File Somewhere:
c
98 stop 'ERROR in parameter file!'
99 stop 'ERROR in data file!'
end

subroutine lusim
c-----

```

```

c
c      Conditional Simulation of a 3-D Rectangular Grid
c      *****
c
c This subroutine generates 3-D realizations of a Gaussian process with
c a given autocovariance model, and conditional to input Gaussian data.
c The conditional simulation is achieved by a single matrix inversion
c using the Cholesky decomposition technique described in M. Davis, Math
c Geol. 19(2), 1987 , and in F. Alabert , Math. Geol. 19(5) 1987
c
c NOTE: This algorithm should only be used for small simulations
c      (nd+nx*ny<200) unless CPU time is not an issue and a check is
c      made for machine precision problems.
c
c
c INPUT VARIABLES:
c
c nd          Number of data (no missing values)
c x(nd)       X coordinates of the data
c y(nd)       Y coordinates of the data
c z(nd)       Z coordinates of the data
c vr(nd)      Data values
c nx          Number of blocks in X
c ny          Number of blocks in Y
c nz          Number of blocks in Z
c xmn         X Coordinate of the first node
c ymn         Y Coordinate of the first node
c zmn         Z Coordinate of the first node
c xsiz        X spacing of the grid nodes
c ysiz        Y spacing of the grid nodes
c zsiz        Z spacing of the grid nodes
c nst         Number of variogram structures
c c0          Nugget effect
c it(nst)     Variogram type (1=sph,2=exp,3=gaus,4=powr)
c aa(nst)     Range except for the power model where "aa" is
c             the power
c cc(nst)     Contribution of each nested structure except for
c             the power model where "cc" is the slope
c ang1,2,3(nst) Azimuth angle for anisotropy
c anis1,2(nst) Anisotropy ratio
c seed        Random number seed
c lout        Fortran output unit for the simulations
c nsim        number of simulations
c
c
c OUTPUT VARIABLES: Simulated Values are written to "lout"
c
c
c WORKING VARIABLES:

```

```

c
c xg,yg,zg(nxyz) Location of grid nodes
c c11(nd,nd) Covariance matrix of the conditioning data
c c22i(nxy,nxyz) Covariance matrix of the grid area
c c12(nd,nxyz) Covariance matrix between conditioning data
c and simulated points
c l11(nd,nd) Lower triangular matrix resulting from the LU
c decomposition of C11
c l22(nxyz,nxyz) Lower triangular matrix obtained by LU decomposition
c of C22i - L21.U12 (see M.Davis)
c w2(nxyz) Vector of gaussian random numbers
c y2(nxyz) Vector containing the simulated values of the grid
c
c

```

```

c Original: F.G. Alabert Nov. 1985 revised: Tercan A.E., 2004
c-----

```

```

include 'lusim.inc'
real xg(MAXXY),yg(MAXXY),zg(MAXXY),y2(MAXXY),w2(MAXXY),
+ c11(MAXDAT,MAXDAT),c12(MAXDAT,MAXXY),c22(MAXXY,MAXXY),
+ u12(MAXDAT,MAXXY),c22i(MAXXY,MAXXY),l11(MAXDAT,MAXDAT),
+ l11inv(MAXDAT,MAXDAT),l21v1(MAXXY),l22(MAXXY,MAXXY),
+ le(MAXDAT),slsk(MAXXY),v1(MAXDAT),
+ vect(MAXXY),r22(MAXXY,MAXXY),
+ xc(MAXXY),yc(MAXXY),sim(MAXXY,101)
real*8 p,rotmat(MAXNST,3,3),acorni
logical coloc

c
write(*,*) 'which decomposition method do you want to use?'
write(*,*) 'enter 1 for spectral decomposition'
write(*,*) ' 2 for symmetric " "'
write(*,*) ' 3 for Cholesky " "'
read(*,*) id

c
c Set up the rotation/anisotropy matrices that are needed for the
c variogram:
c
do is=1,nst(1)
call setrot(ang1(is),ang2(is),ang3(is),anis1(is),anis2(is),
+ is,MAXNST,rotmat)
end do
nxy = nx*ny
nxyz = nx*ny*nz

c
c Establish positions of grid points (may be fewer than nx*ny*nz):
c
c nxyzu = number actually used
c
nxyzu = 0
do iz = 1,nz
do iy = 1,ny

```

```

do ix = 1,nx
  zz = zmn + (iz-1)*zsiz
  yy = ymn + (iy-1)*ysiz
  xx = xmn + (ix-1)*xsiz
  coloc = .false.
  do i=1,nd
    test = abs(xx-x(i)) + abs(yy-y(i)) + abs(zz-z(i))
    if(test.le.EPSLON) coloc = .true.
  end do
c
c Only simulate this grid node if not colocated:
c
  if(.not.coloc) then
    nxyzu = nxyzu + 1
    xg(nxyzu) = xx
    yg(nxyzu) = yy
    zg(nxyzu) = zz
    if(idbg.ge.3) write(ldbg,101) nxyzu,xg(nxyzu),
+
+      yg(nxyzu),zg(nxyzu)
101      format(' Node ',i4,': x = ',f9.3,' y = ',f9.3,
+
+      ' z = ',f9.3)
    endif
  end do
end do
end do
c
c Compute C22i: first get all the covariances:
c
  call cova3(xg(1),yg(1),zg(1),xg(1),yg(1),zg(1),1,nst,MAXNST,
+
+    c0,it,cc,aa,1,MAXNST,rotmat,cmax,sill)
  do i = 1,nxyzu
  do j = i,nxyzu
    call cova3(xg(i),yg(i),zg(i),xg(j),yg(j),zg(j),1,nst,MAXNST,
+
+      c0,it,cc,aa,1,MAXNST,rotmat,cmax,cov)
    c22i(i,j) = cov
    c22i(j,i) = cov
    if(idbg.ge.3) write(ldbg,102) i,j,cov
102    format(' Covariance between Node ',i4,' and ',i4,' = ',f9.6)
  end do
  end do
c
c Compute C11, L11, inv(L11) AND inv(L11).Z1: First compute c11:
c
  do i=1,nd
    c11(i,i) = sill
    do j=i+1,nd
      call cova3(x(i),y(i),z(i),x(j),y(j),z(j),1,nst,MAXNST,
+
+        c0,it,cc,aa,1,MAXNST,rotmat,cmax,cov)
      c11(i,j) = cov
      c11(j,i) = cov

```

```

        if(idbg.ge.3) write(ldbg,103) i,j,cov
    end do
end do
103 format(' Data-Data Covariance: ',i4,' and ',i4,' = ',f9.6)
c
c Compute C12 :
c
    do i = 1,nd
        do j = 1,nxyzu
            call cova3(x(i),y(i),z(i),xg(j),yg(j),zg(j),1,nst,
+                MAXNST,c0,it,cc,aa,1,MAXNST,rotmat,cmax,cov)
            c12(i,j) = cov
            if(idbg.ge.3) write(ldbg,104) i,j,cov
        end do
    end do
104 format(' Covariance between data',i4,'and node',i4,' = ',f9.6)
c
c
    if(id.eq.1.or.id.eq.2) then
        call schur(c11,c22i,c12,nd,nxyzu,vr,vect,r22,MAXDAT,MAXXY,
&                MAXDAT+MAXXY,idbg,ldbg,id)
        go to 88
    end if
c
c Compute l11:
c
    call chol(c11,l11,nd,MAXDAT,ierr)
c
c Compute inv(l11):
c
    call linv(l11,l11inv,nd,MAXDAT)
c
c Compute l11inv.z1:
c
    do i=1,nd
        v1(i) = 0
        do k=1,i
            v1(i) = v1(i) + l11inv(i,k)*vr(k)
        end do
    end do
c
c Computation of LAMBDAe:
c
    sle = 0.
    do i = 1,nd
        le(i) = 0.
        do k = 1,nd
            do j = 1,nd
                le(i) = le(i) + l11inv(j,i)*l11inv(j,k)
            end do
        end do
    end do

```

```

        end do
        sle = sle + le(i)
    end do
c
c Compute U12 = L11INV.C12:
c
    do i = 1,nd
        do j = 1,nxyzu
            u12(i,j) = 0
            do k = 1,i
                u12(i,j) = u12(i,j) + l11inv(i,k)*c12(k,j)
            end do
        end do
    end do
c
c Compute C22 - L21.U12 = C22 - U12'.U12 ---> C22:
c
    do i = 1,nxyzu
        do j = 1,nxyzu
            c22(i,j) = c22i(i,j)
            do k = 1,nd
                c22(i,j) = c22(i,j) - u12(k,i)*u12(k,j)
            end do
        end do
    end do
c
c Compute L22:
c
    call chol(c22,l22,nxyzu,MAXXY,ierr)
c
c Compute L21.L11INV.Z1 = U12'.V1 ---> L21V1:
c
    do i = 1,nxyzu
        l21v1(i) = 0
        do k = 1,nd
            l21v1(i) = l21v1(i) + u12(k,i)*v1(k)
        end do
    end do
c
c Compute LAMBDAsk:
c
    do i = 1,nxyzu
        slsk(i) = 0.
        do k = 1,nd
            do j = k,nd
                slsk(i) = slsk(i) + u12(j,i)*l11inv(j,k)
            end do
        end do
    end do
c

```

```

c Local bias correction:
c
  do i = 1,nxyzu
    do j = 1,nd
c      l21v1(i) = l21v1(i) + (1-slsk(i))*le(j)*vr(j)/sle
    end do
  end do
c
c LOOP OVER THE NUMBER OF SIMULATIONS
c
c idum=-1      ! added by me
88 do 100 isim=1,nsim
c
c Generate (nxyzu) Gaussian Random Numbers:
c
  do i=1,nxyzu
    w2(i) = real(acorni(idum))
c    w2(i) = ran2(idum)
  end do
  do i = 1,nxyzu
1    p = dble(w2(i))
    call gauinv(p,xp,ierr)
    w2(i) = xp
    if(w2(i).lt.(-6.).or.w2(i).gt.(6.)) then
c      w2(i) = real(acorni(idum))
    w2(i) = ran2(idum)
c    go to 1
    endif
  end do
c
c Compute the simulation at each grid point:
c
  if(id.eq.1.or.id.eq.2) then
    do i=1,nxyzu
      sum2=0.0
      do k=1,nxyzu
        sum2=sum2+r22(i,k)*w2(k)
      end do
      y2(i)=vect(i)-sum2
    end do
    go to 89
  end if
  do i=1,nxyzu
    y2(i) = l21v1(i)
    do k=1,i
      y2(i) = y2(i) + l22(i,k)*w2(k)
    end do
  end do
c
c Write the simulation out to file:

```

```

c
89      ic = 0
      do iz=1,nz
        do iy=1,ny
          do ix=1,nx
            zz = zmn + (iz-1)*zsiz
            yy = ymn + (iy-1)*ysiz
            xx = xmn + (ix-1)*xsiz
            ic = ic + 1
            test = abs(xx-xg(ic)) +
+              abs(yy-yg(ic)) +
+              abs(zz-zg(ic))
            xc(ic)=xx
            yc(ic)=yy
c
c Was this grid node simulated or was it co-located with a datum:
c
            if(test.le.EPSLON) then
              simval = y2(ic)
              sim(ic,ism)=simval
            else
              ic = ic - 1
              do i=1,nd
                test = abs(xx-x(i)) +
+                  abs(yy-y(i)) +
+                  abs(zz-z(i))
                if(test.le.EPSLON) then
                  simval = vr(i)
                  sim(ic,ism)=simval
                  go to 2
                end if
              end do
            continue
          2
        end if
      write(lout,'(2(f8.1,1x),f8.2)') xx,yy,simval
    end do
  end do
end do

c
c END LOOP OVER SIMULATIONS:
c
100 continue
   do i=1,nxyzu
     write(lout,110) xg(i),yg(i),(sim(i,j),j=1,nsim)
   end do
   return
110 format(2(f8.1,1x),100(f5.2,1x))
   end
   subroutine chol(a,t,n,ndim,ierr)
c-----

```

```

C
C           Cholesky Decomposition
C           *****
C
C This subroutine calculates the lower triangular matrix T which, when
C multiplied by its own transpose, gives the symmetric matrix A. (from
C "Numerical Analysis of Symmetric Matrices," H.R. Schwarz et al.,
C p. 254)
C
C
C
C INPUT VARIABLES:
C
C a(n,n)      Symmetric positive definite matrix to be
C              decomposed (destroyed in the calculation of t)
C t(n,n)      Lower triangular matrix solution
C n           Dimension of the system you're decomposing
C ndim        Dimension of matrix a (Note: In the main program,
C              matrix a may have been dimensioned larger than
C              necessary, i.e. n, the size of the system you're
C              decomposing, may be smaller than ndim.)
C ierr        Error code: ierr=0 - no errors; ierr=1 - matrix a
C              is not positive definite
C
C
C
C NO EXTERNAL REFERENCES:
C-----
C      dimension a(ndim,ndim),t(ndim,ndim)
C      ierr = 0
C
C Check for positive definiteness:
C
C   do ip=1,n
C     if(a(ip,ip).le.0.0) then
C       write(*,'(a)') 'WARNING: chol - not positive definite'
C       ierr = 1
C       go to 1
C     endif
C     t(ip,ip) = sqrt (a(ip,ip))
C     if(ip.ge.n) return
C     do k = ip+1,n
C       t(k,ip) = a(ip,k)/t(ip,ip)
C     end do
C     do i = ip+1,n
C       do k = i,n
C         a(i,k) = a(i,k) - t(i,ip) * t(k,ip)
C       end do
C     end do
C   continue
1

```

```

    end do
c
c Finished:
c
    return
    end

    subroutine linv(a,b,n,ndim)
c-----
c
c      Inverse of a Lower Triangular Matrix
c      *****
c
c This subroutine finds the inverse of a lower triangular matrix A and
c stores the answer in B. (from "Numerical Analysis of Symmetric
c Matrices," H.R. Schwarz et al.,)
c
c
c
c INPUT VARIABLES:
c
c a(n,n)      Lower triangular matrix to be inverted
c b(n,n)      the inverse
c n           Dimension of the matrix you're inverting
c ndim        Dimension of matrix a (Note: In the main program,
c             matrix a may have been dimensioned larger than
c             necessary, i.e. n, the size of the system you're
c             decomposing, may be smaller than ndim.)
c
c-----
    dimension a(ndim,ndim),b(ndim,ndim)
    do i = 1,n
        if(i.gt.1) then
            do k = 1,i-1
                sum=0.
                do j = k,i-1
                    sum = sum + a(i,j)*b(j,k)
                end do
                b(i,k) = -sum/a(i,i)
            end do
        end if
        b(i,i) = 1./a(i,i)
    end do
c
c Finished:
c
    return
    end

```

subroutine makepar

```
C-----  
C           Write a Parameter File  
C           *****  
C-----  
      lun = 99  
      open(lun,file='lusim.par',status='UNKNOWN')  
      write(lun,10)  
10  format('           Parameters for LUSIM',/,  
+         '           *****',/,/,  
+         'START OF PARAMETERS:')  
  
      write(lun,11)  
11  format('parta.dat           ',  
+         '-file with data')  
      write(lun,12)  
12  format('1  2  0  3           ',  
+         '- columns for X,Y,Z, normal scores')  
      write(lun,13)  
13  format('-1.0e21  1.0e21           ',  
+         '- trimming limits')  
      write(lun,14)  
14  format('3           ',  
+         '-debugging level: 0,1,2,3')  
      write(lun,15)  
15  format('lusim.dbg           ',  
+         '-file for debugging output')  
      write(lun,16)  
16  format('lusim.out           ',  
+         '-file for realization(s)')  
      write(lun,17)  
17  format('100           ',  
+         '-number of realizations')  
      write(lun,18)  
18  format('4  40.25  0.5           ',  
+         '-nx,xmn,xsiz')  
      write(lun,19)  
19  format('4  28.25  0.5           ',  
+         '-ny,ymn,ysiz')  
      write(lun,20)  
20  format('1  0.00  1.0           ',  
+         '-nz,zmn,zsiz')  
      write(lun,21)  
21  format('112063           ',  
+         '-random number seed')  
      write(lun,22)  
22  format('1  0.2           ',  
+         '-nst, nugget effect')
```

```
write(lun,23)
23 format('1 0.8 0.0 0.0 0.0 ',
+ '-it,cc,ang1,ang2,ang3')
write(lun,24)
24 format(' 10.0 10.0 10.0 ',
+ '-a_hmax, a_hmin, a_vert')

close(lun)
return
end
```

```

double precision function acorni(idum)
C-----
c Fortran implementation of ACORN random number generator of order less
c than or equal to 12 (higher orders can be obtained by increasing the
c parameter value MAXORD).
c
c
c NOTES: 1. The variable idum is a dummy variable. The common block
c       IACO is used to transfer data into the function.
c
c       2. Before the first call to ACORN the common block IACO must
c       be initialised by the user, as follows. The values of
c       variables in the common block must not subsequently be
c       changed by the user.
c
c       KORDEI - order of generator required ( must be =< MAXORD)
c
c       MAXINT - modulus for generator, must be chosen small
c               enough that 2*MAXINT does not overflow
c
c       ixv(1) - seed for random number generator
c               require 0 < ixv(1) < MAXINT
c
c       (ixv(l+1),l=1,KORDEI)
c       - KORDEI initial values for generator
c       require 0 =< ixv(l+1) < MAXINT
c
c       3. After initialisation, each call to ACORN generates a single
c       random number between 0 and 1.
c
c       4. An example of suitable values for parameters is
c
c       KORDEI = 10
c       MAXINT = 2**30
c       ixv(1) = an odd integer in the (approximate) range
c               (0.001 * MAXINT) to (0.999 * MAXINT)
c       ixv(l+1) = 0, l=1,KORDEI
c
c Author: R.S.Wikramaratna,                               Date: October 1990
C-----
implicit double precision (a-h,o-z)
parameter (KORDEI=12,MAXOP1=KORDEI+1,MAXINT=2**30)
common/iaco/ ixv(MAXOP1)
do i=1,KORDEI
  ixv(i+1)=(ixv(i+1)+ixv(i))
  if(ixv(i+1).ge.MAXINT) ixv(i+1)=ixv(i+1)-MAXINT
end do
acorni=dble(ixv(KORDEI+1))/MAXINT
return
end

```

```

subroutine chknam(str,len)
C-----
C          Check for a Valid File Name
C          *****
C
C This subroutine takes the character string "str" of length "len" and
C removes all leading blanks and blanks out all characters after the
C first blank found in the string (leading blanks are removed first).
C
C-----
      parameter (MAXLEN=132)
      character str(MAXLEN)*1
C
C Remove leading blanks:
C
      do i=1,len-1
         if(str(i).ne.' ') then
            if(i.eq.1) go to 1
            do j=1,len-i+1
               k = j + i - 1
               str(j) = str(k)
            end do
            do j=len,len-i+2,-1
               str(j) = ' '
            end do
            go to 1
         end if
      end do
1  continue
C
C Find first blank and blank out the remaining characters:
C
      do i=1,len-1
         if(str(i).eq.' ') then
            do j=i+1,len
               str(j) = ' '
            end do
            go to 2
         end if
      end do
2  continue
C
C Return with modified file name:
C
      return
      end

```

```

subroutine covar3(x1,y1,z1,x2,y2,z2,ivarg,nst,MAXNST,c0,it,cc,aa,
+             irot,MAXROT,rotmat,cmax,cova)
c-----
c
c             Covariance Between Two Points
c             *****
c
c This subroutine calculated the covariance associated with a variogram
c model specified by a nugget effect and nested variogram structures.
c The anisotropy definition can be different for each nested structure.
c
c
c INPUT VARIABLES:
c
c x1,y1,z1      coordinates of first point
c x2,y2,z2      coordinates of second point
c nst(ivarg)    number of nested structures (maximum of 4)
c ivarg         variogram number (set to 1 unless doing cokriging
c              or indicator kriging)
c MAXNST       size of variogram parameter arrays
c c0(ivarg)    isotropic nugget constant
c it(i)        type of each nested structure:
c              1. spherical model of range a;
c              2. exponential model of parameter a;
c                i.e. practical range is 3a
c              3. gaussian model of parameter a;
c                i.e. practical range is a*sqrt(3)
c              4. power model of power a (a must be gt. 0 and
c                lt. 2). if linear model, a=1,c=slope.
c              5. hole effect model
c cc(i)        multiplicative factor of each nested structure.
c              (sill-c0) for spherical, exponential, and gaussian
c              slope for linear model.
c aa(i)        parameter "a" of each nested structure.
c irot         index of the rotation matrix for the first nested
c              structure (the second nested structure will use
c              irot+1, the third irot+2, and so on)
c MAXROT       size of rotation matrix arrays
c rotmat       rotation matrices
c
c
c OUTPUT VARIABLES:
c
c cmax         maximum covariance
c cova        covariance between (x1,y1,z1) and (x2,y2,z2)
c
c
c EXTERNAL REFERENCES: sqdist  computes anisotropic squared distance
c rotmat       computes rotation matrix for distance

```

```

C-----
parameter(PI=3.14159265,PMX=999.,EPSLON=1.e-10)
integer nst(*),it(*)
real c0(*),cc(*),aa(*)
real*8 rotmat(MAXROT,3,3),hsqd,sqdist

C
c Calculate the maximum covariance value (used for zero distances and
c for power model covariance):
C
  1start = 1 + (ivarg-1)*MAXNST
  1cmax = c0(ivarg)
  1do is=1,nst(ivarg)
    1ist = 1start + is - 1
    1if(it(ist).eq.4) then
      1cmax = cmax + PMX
    1else
      1cmax = cmax + cc(ist)
    1endif
  1end do

C
c Check for "zero" distance, return with cmax if so:
C
  1hsqd = sqdist(x1,y1,z1,x2,y2,z2,1rot,MAXROT,rotmat)
  1if(real(hsqd).lt.EPSLON) then
    1cova = cmax
    1return
  1endif

C
c Loop over all the structures:
C
  1cova = 0.0
  1do is=1,nst(ivarg)
    1ist = 1start + is - 1

C
c Compute the appropriate distance:
C
    1if(ist.ne.1) then
      1ir = min((1rot+is-1),MAXROT)
      1hsqd=sqdist(x1,y1,z1,x2,y2,z2,1ir,MAXROT,rotmat)
    1end if
    1h = real(dsqrt(hsqd))

C
c Spherical Variogram Model?
C
    1if(it(ist).eq.1) then
      1hr = h/aa(ist)
      1if(hr.lt.1.) cova=cova+cc(ist)*(1.-hr*(1.5-.5*hr*hr))

C
c Exponential Variogram Model?
C

```

```

        else if(it(ist).eq.2) then
            cova = cova + cc(ist)*exp(-3.0*h/aa(ist))
c
c Gaussian Variogram Model?
c
            else if(it(ist).eq.3) then
                cova = cova + cc(ist)*exp(-(3.0*h/aa(ist))
+
                *(3.0*h/aa(ist)))
c
c Power Variogram Model?
c
            else if(it(ist).eq.4) then
                cova = cova + cmax - cc(ist)*(h**aa(ist))
c
c Hole Effect Model?
c
            else if(it(ist).eq.5) then
c
                d = 10.0 * aa(ist)
c
                cova = cova + cc(ist)*exp(-3.0*h/d)*cos(h/aa(ist)*PI)
                cova = cova + cc(ist)*cos(h/aa(ist)*PI)
            endif
        end do
c
c Finished:
c
        return
        end

```

```

subroutine eigsrt(d,v,n,np)
c  implicit real*8 (a-h,o-z)
c  parameter (np=500)
dimension d(np),v(np,np)
do 13 i=1,n-1
  k=i
  p=d(i)
  do 11 j=i+1,n
    if(d(j).ge.p)then
      k=j
      p=d(j)
    endif
11  continue
  if(k.ne.i)then
    d(k)=d(i)
    d(i)=p
    do 12 j=1,n
      p=v(j,i)
      v(j,i)=v(j,k)
      v(j,k)=p
12  continue
  endif
13  continue
return
end

```

```

subroutine gauinv(p,xp,ierr)
C-----
C
C Computes the inverse of the standard normal cumulative distribution
C function with a numerical approximation from : Statistical Computing,
C by W.J. Kennedy, Jr. and James E. Gentle, 1980, p. 95.
C
C
C INPUT/OUTPUT:
C
C p = double precision cumulative probability value: dble(psingle)
C xp = G^-1 (p) in single precision
C ierr = 1 - then error situation (p out of range), 0 - OK
C
C-----
      real*8 p0,p1,p2,p3,p4,q0,q1,q2,q3,q4,y,pp,lim,p
      save p0,p1,p2,p3,p4,q0,q1,q2,q3,q4,lim
C
C Coefficients of approximation:
C
      data lim/1.0e-10/
      data p0/-0.322232431088/,p1/-1.0/,p2/-0.342242088547/,
+ p3/-0.0204231210245/,p4/-0.0000453642210148/
      data q0/0.0993484626060/,q1/0.588581570495/,q2/0.531103462366/,
+ q3/0.103537752850/,q4/0.0038560700634/
C
C Check for an error situation:
C
      ierr = 1
      if(p.lt.lim) then
          xp = -1.0e10
          return
      end if
      if(p.gt.(1.0-lim)) then
          xp = 1.0e10
          return
      end if
      ierr = 0
C
C Get k for an error situation:
C
      pp = p
      if(p.gt.0.5) pp = 1 - pp
      xp = 0.0
      if(p.eq.0.5) return
C
C Approximate the function:
C
      y = dsqrt(dlog(1.0/(pp*pp)))
      xp = real( y + (((y*p4+p3)*y+p2)*y+p1)*y+p0) /

```

```
+      (((y*q4+q3)*y+q2)*y+q1)*y+q0 )  
if(real(p).eq.real(pp)) xp = -xp  
c  
c Return with G^-1(p):  
c  
  return  
end
```

```

subroutine jacobi(a,n,d,v,nrot,np)
c  implicit real*8 (a-h,o-z)
c  parameter (nmax=500,np=500)
c  dimension a(np,np),d(np),v(np,np),b(np),z(np)
dimension a(np,np),d(np),v(np,np),b(np),z(np)

do 12 ip=1,n
  do 11 iq=1,n
    v(ip,iq)=0.
11  continue
    v(ip,ip)=1.
12  continue
  do 13 ip=1,n
    b(ip)=a(ip,ip)
    d(ip)=b(ip)
    z(ip)=0.
13  continue
nrot=0
do 24 i=1,50
  sm=0.
  do 15 ip=1,n-1
    do 14 iq=ip+1,n
      sm=sm+abs(a(ip,iq))
14  continue
15  continue
  if(sm.eq.0.)return
  if(i.lt.4)then
    tresh=0.2*sm/n**2
  else
    tresh=0.
  endif
  do 22 ip=1,n-1
    do 21 iq=ip+1,n
      g=100.*abs(a(ip,iq))
      if((i.gt.4).and.(abs(d(ip))+g.eq.abs(d(ip)))
*      .and.(abs(d(iq))+g.eq.abs(d(iq))))then
        a(ip,iq)=0.
      else if(abs(a(ip,iq)).gt.tresh)then
        h=d(iq)-d(ip)
        if(abs(h)+g.eq.abs(h))then
          t=a(ip,iq)/h
        else
          theta=0.5*h/a(ip,iq)
          t=1./(abs(theta)+sqrt(1.+theta**2))
          if(theta.lt.0.)t=-t
        endif
        c=1./sqrt(1+t**2)
        s=t*c
        tau=s/(1.+c)
        h=t*a(ip,iq)

```

```

z(ip)=z(ip)-h
z(iq)=z(iq)+h
d(ip)=d(ip)-h
d(iq)=d(iq)+h
a(ip,iq)=0.
do 16 j=1,ip-1
  g=a(j,ip)
  h=a(j,iq)
  a(j,ip)=g-s*(h+g*tau)
  a(j,iq)=h+s*(g-h*tau)
16  continue
do 17 j=ip+1,iq-1
  g=a(ip,j)
  h=a(j,iq)
  a(ip,j)=g-s*(h+g*tau)
  a(j,iq)=h+s*(g-h*tau)
17  continue
do 18 j=iq+1,n
  g=a(ip,j)
  h=a(iq,j)
  a(ip,j)=g-s*(h+g*tau)
  a(iq,j)=h+s*(g-h*tau)
18  continue
do 19 j=1,n
  g=v(j,ip)
  h=v(j,iq)
  v(j,ip)=g-s*(h+g*tau)
  v(j,iq)=h+s*(g-h*tau)
19  continue
  nrot=nrot+1
endif
21  continue
22  continue
do 23 ip=1,n
  b(ip)=b(ip)+z(ip)
  d(ip)=b(ip)
  z(ip)=0.
23  continue
24  continue
pause '50 iterations should never happen'
return
end

```

```

subroutine matmul(a,b,c,m,l,n,maxa,maxab,maxb)
c    implicit real*8 (a-h,o-z)
c    parameter (max=500)
dimension a(maxa,maxab),b(maxab,maxb),c(maxa,maxb)
c
do 10 j=1,m
do 20 k=1,n
c(j,k)=0.0
do 30 i=1,l
c(j,k)=c(j,k)+a(j,i)*b(i,k)
30    continue
20    continue
10    continue
return
end

```

```

FUNCTION ran2(idum)
c Press W.H. et. all., Numerical Recipes in Fortran 77, 1996
INTEGER idum,IM1,IM2,IMM1,IA1,IA2,IQ1,IQ2,IR1,IR2,NTAB,NDIV
REAL ran2,AM,EPS,RNMX
PARAMETER (IM1=2147483563,IM2=2147483399,AM=1./IM1,IMM1=IM1-1,
*IA1=40014,IA2=40692,IQ1=53668,IQ2=52774,IR1=12211,IR2=3791,
*NTAB=32,NDIV=1+IMM1/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
INTEGER idum2,j,k,iv(NTAB),iy
SAVE iv,iy,idum2
DATA idum2/123456789/, iv/NTAB*0/, iy/0/
if (idum.le.0) then
  idum=max(-idum,1)
  idum2=idum
  do 11 j=NTAB+8,1,-1
    k=idum/IQ1
    idum=IA1*(idum-k*IQ1)-k*IR1
    if (idum.lt.0) idum=idum+IM1
    if (j.le.NTAB) iv(j)=idum
11  continue
  iy=iv(1)
endif
k=idum/IQ1
idum=IA1*(idum-k*IQ1)-k*IR1
if (idum.lt.0) idum=idum+IM1
k=idum2/IQ2
idum2=IA2*(idum2-k*IQ2)-k*IR2
if (idum2.lt.0) idum2=idum2+IM2
j=1+iy/NDIV
iy=iv(j)-idum2
iv(j)=idum
if(iy.lt.1)iy=iy+IMM1
ran2=min(AM*iy,RNMX)
return
END

```

```

subroutine schur(c11,c22,c12,nd,nxyzu,vr,vect,r22,
+           MAXDAT,MAXXY,max,idbg,ldbg,id)
c  Press W.H. et. all., Numerical Recipes in Fortran 77, 1996
c  revised: Tercan, A.E.,2004
c   $R=Q*D^{1/2}$ ,  $R_t=D^{1/2}*Q_t$ ,  $R_{_1t}=Q*D_{_1}(1/2)$ ,  $R_{_1}=D_{_1}(1/2)*Q_t$ 
c  implicit real*8 (a-h,o-z)
c  parameter (max=500)
dimension c11(MAXDAT,MAXDAT),c22(MAXXY,MAXXY)
dimension c12(MAXDAT,MAXXY),vr(MAXDAT)
dimension r11_1(MAXDAT,MAXDAT)
dimension vect(MAXXY),
+       r22(MAXXY,MAXXY),r21(MAXXY,MAXDAT),
+       r211_1(MAXXY,MAXDAT),rr(MAXXY,MAXXY)
dimension evect(MAX,MAX),d(MAX,MAX),temp(MAX,MAX),
+       css(MAXDAT,MAXDAT)
c
dimension a(max,max),r(max,max)
dimension eval(max),evec(max,max)
dimension eva(maxdat),eve(maxdat,maxdat),di(maxdat,maxdat),
+       tem(maxdat,maxdat),evet(maxdat,maxdat)
c
n=nd+nxyzu
do i=1,nd
  do j=1,nd
    a(i,j)=c11(i,j)
  end do
end do
do i=1,nd
  do j=1,nxyzu
    a(i,j+nd)=c12(i,j)
  end do
end do
do i=1,nxyzu
  do j=1,nd
    a(i+nd,j)=c12(j,i)
  end do
end do
do i=1,nxyzu
  do j=1,nxyzu
    a(i+nd,j+nd)=c22(i,j)
  end do
end do
if(idbg.eq.5) then
  write(ldbg,*) 'covariance matrix'
  do i=1,n
    write(ldbg,44) (a(i,j),j=1,n)
  end do
end if
c

```

```

call jacobi(a,n,eval,vec,nrot,max)
call eigprt(eval,vec,n,max)
do i=1,n
  do j=1,n
    d(i,j)=0.0
    evec(i,j)=vec(j,i)
  end do
  d(i,i)=eval(i)
end do
if(idbg.eq.5) then
  write(ldbg,*) 'eigen values'
  write(ldbg,44) (eval(i),i=1,n)
  write(ldbg,*) 'eigen vectors'
  do i=1,n
    write(ldbg,44) (vec(i,j),j=1,n)
  end do
  write(ldbg,*) 'Covariance matrix=EVEC.D.EVECT'
  call matmul(vec,d,temp,n,n,n,max,max,max)
  call matmul(temp,evec,a,n,n,n,max,max,max)
  do i=1,n
    write(ldbg,44) (a(i,j),j=1,n)
  end do
end if
write(*,*) '*** PASSED Eigen solution ***'
do i=1,n
  do j=1,n
    if(i.eq.j) then
      d(i,j)=sqrt(eval(j))
    else
      d(i,j)=0.0
    end if
  end do
end do
c) d=sqrt(eval)
if(id.eq.1) then ! spectral decomposition
  call matmul(vec,d,r,n,n,n,max,max,max)
else if(id.eq.2) then ! symmetric decomposition
  call matmul(vec,d,temp,n,n,n,max,max,max)
  call matmul(temp,evec,r,n,n,n,max,max,max)
end if
if(idbg.eq.5) then
  write(ldbg,*) 'R Matrix:'
  do i=1,n
    write(ldbg,44) (r(i,j),j=1,n)
    do j=1,n
      evec(i,j)=r(j,i)
    end do
  end do
  write(ldbg,*) 'C=R.Rt'
  call matmul(r,evec,temp,n,n,n,max,max,max)

```

```

        do i=1,n
            write(ldbg,44) (temp(i,j),j=1,n)
        end do
    end if
c
c   construct sub-matrices r11, r22, r12 and r21
c
    do i=1,nd
        do j=1,nd
            c11(i,j)=r(i,j)
        end do
    end do
    do i=1,nxyzu
        do j=1,nxyzu
            r22(i,j)=r(nd+i,nd+j)
        end do
    end do
    do i=1,nd
        do j=1,nxyzu
            c12(i,j)=r(i,nd+j)
        end do
    end do
    do i=1,nxyzu
        do j=1,nd
            r21(i,j)=r(nd+i,j)
        end do
    end do
    if(idbg.eq.5) then
        write(ldbg,*) 'R11 matrix:'
        do i=1,nd
            write(ldbg,44) (c11(i,j),j=1,nd)
            do j=1,nd
                css(i,j)=c11(i,j)
            end do
        end do
    end if
c
c   inverse r11=c11 using Schur decomposition
c
    call jacobi(c11,nd,eva,eve,nrot,maxdat)
    call eigprt(eva,eve,nd,maxdat)
    ieig=0
    do i=1,nd
        if(eva(i).lt.0.0) ieig=ieig+1
        do j=1,nd
            evet(i,j)=eve(j,i)
            if(i.eq.j) then
                di(i,j)=1.0/eva(i)
            else
                di(i,j)=0.0
            end if
        end do
    end do

```

```

        end if
    end do
end do
write(*,*) 'number of negative eigen values:',ieig
if(ieig.lt.1) then
    call matmul(eve,di,tem,nd,nd,nd,maxdat,maxdat,maxdat)
    call matmul(tem,evet,r11_1,nd,nd,nd,maxdat,maxdat,maxdat)
else
    write(*,*) 'the inverse of the matrix R11 does not exist'
    stop
end if
if(idbg.eq.5) then
    write(ldbg,*) 'inverse matrix r11_1:'
    do i=1,nd
        write(ldbg,44) (r11_1(i,j),j=1,nd)
    end do
    write(ldbg,*) 'r11.r11_1='
    call matmul(r11_1,css,tem,nd,nd,nd,maxdat,maxdat,maxdat)
    do i=1,nd
        write(ldbg,44) (tem(i,j),j=1,nd)
    end do
end if
c
c  compute vector vect and matrix rr
c
call matmul(r21,r11_1,r2111_1,nxyzu,nd,nd,MAXXY,MAXDAT,MAXDAT)
do i=1,nxyzu
    sum=0.0
    do j=1,nd
        sum=sum+r2111_1(i,j)*vr(j)
    end do
    vect(i)=sum
end do
call matmul(r2111_1,c12,rr,nxyzu,nd,nxyzu,MAXXY,MAXDAT,MAXXY)
do i=1,nxyzu
    do j=1,nxyzu
        r22(i,j)=rr(i,j)-r22(i,j)
    end do
end do
if(ldbg.eq.5) then
    write(ldbg,*) 'r22 matrix:'
    do i=1,nxyzu
        write(ldbg,44) (r22(i,j),j=1,nxyzu)
    end do
end if
return
44  format(10(f8.4,1x))
end

```

```

subroutine setrot(ang1,ang2,ang3,anis1,anis2,ind,MAXROT,rotmat)

```

```

C-----
C
C      Sets up an Anisotropic Rotation Matrix
C      *****
C
C Sets up the matrix to transform cartesian coordinates to coordinates
C accounting for angles and anisotropy (see manual for a detailed
C definition):
C
C
C INPUT PARAMETERS:
C
C ang1      Azimuth angle for principal direction
C ang2      Dip angle for principal direction
C ang3      Third rotation angle
C anis1     First anisotropy ratio
C anis2     Second anisotropy ratio
C ind       matrix indicator to initialize
C MAXROT    maximum number of rotation matrices dimensioned
C rotmat    rotation matrices
C
C
C NO EXTERNAL REFERENCES
C
C-----
C      parameter(DEG2RAD=3.141592654/180.0,EPSLON=1.e-20)
C      real*8  rotmat(MAXROT,3,3),afac1,afac2,sina,sinb,sint,
C      +      cosa,cosb,cost
C
C Converts the input angles to three angles which make more
C mathematical sense:
C
C      alpha  angle between the major axis of anisotropy and the
C             E-W axis. Note: Counter clockwise is positive.
C      beta   angle between major axis and the horizontal plane.
C             (The dip of the ellipsoid measured positive down)
C      theta  Angle of rotation of minor axis about the major axis
C             of the ellipsoid.
C
C      if(ang1.ge.0.0.and.ang1.lt.270.0) then
C          alpha = (90.0 - ang1) * DEG2RAD
C      else
C          alpha = (450.0 - ang1) * DEG2RAD
C      endif
C      beta = -1.0 * ang2 * DEG2RAD
C      theta =   ang3 * DEG2RAD
C
C Get the required sines and cosines:
C

```

```
sina = dble(sin(alpha))
sinb = dble(sin(beta))
sint = dble(sin(theta))
cosa = dble(cos(alpha))
cosb = dble(cos(beta))
cost = dble(cos(theta))
```

c

c Construct the rotation matrix in the required memory:

c

```
afac1 = 1.0 / dble(max(anis1,EPSLON))
afac2 = 1.0 / dble(max(anis2,EPSLON))
rotmat(ind,1,1) = (cosb * cosa)
rotmat(ind,1,2) = (cosb * sina)
rotmat(ind,1,3) = (-sinb)
rotmat(ind,2,1) = afac1*(-cost*sina + sint*sinb*cosa)
rotmat(ind,2,2) = afac1*(cost*cosa + sint*sinb*sina)
rotmat(ind,2,3) = afac1*(sint * cosb)
rotmat(ind,3,1) = afac2*(sint*sina + cost*sinb*cosa)
rotmat(ind,3,2) = afac2*(-sint*cosa + cost*sinb*sina)
rotmat(ind,3,3) = afac2*(cost * cosb)
```

c

c Return to calling program:

c

```
return
end
```

```

real*8 function sqdist(x1,y1,z1,x2,y2,z2,ind,MAXROT,rotmat)
C-----
C
C   Squared Anisotropic Distance Calculation Given Matrix Indicator
C   *****
C
C This routine calculates the anisotropic distance between two points
C given the coordinates of each point and a definition of the
C anisotropy.
C
C
C INPUT VARIABLES:
C
C x1,y1,z1      Coordinates of first point
C x2,y2,z2      Coordinates of second point
C ind           The rotation matrix to use
C MAXROT        The maximum number of rotation matrices dimensioned
C rotmat        The rotation matrices
C
C
C
C OUTPUT VARIABLES:
C
C sqdist        The squared distance accounting for the anisotropy
C                and the rotation of coordinates (if any).
C
C
C NO EXTERNAL REFERENCES
C
C
C-----
      real*8 rotmat(MAXROT,3,3),cont,dx,dy,dz
C
C Compute component distance vectors and the squared distance:
C
      dx = dble(x1 - x2)
      dy = dble(y1 - y2)
      dz = dble(z1 - z2)
      sqdist = 0.0
      do i=1,3
          cont = rotmat(ind,i,1) * dx
+           + rotmat(ind,i,2) * dy
+           + rotmat(ind,i,3) * dz
          sqdist = sqdist + cont * cont
      end do
      return
      end

```

```

C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
C
%
C Copyright (C) 1996, The Board of Trustees of the Leland Stanford
%
C Junior University. All rights reserved.
%
C The programs in GSLIB are distributed in the hope that they will be
%
C useful, but WITHOUT ANY WARRANTY. No author or distributor accepts
%
C responsibility to anyone for the consequences of using them or for
%
C whether they serve any particular purpose or work at all, unless he
%
C says so in writing. Everyone is granted permission to copy, modify
%
C and redistribute the programs in GSLIB, but only under the condition
%
C that this notice and the above copyright notice remain intact.
%
C
%
C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
C-----
C
C
C          LU Simulation
C          *****
C
C The following Parameters control static dimensioning within lusim:
C
C MAXX      maximum nodes in X
C MAXY      maximum nodes in Y
C MAXZ      maximum nodes in Z
C MAXDAT    maximum number of data
C MAXNST    maximum number of nested structures for variogram
C
C-----
C
C User Adjustable Parameters:
C
C   parameter(MAXX = 30, MAXY = 30, MAXZ = 4,
C   +         MAXDAT = 200, MAXNST = 4)
C
C Fixed parameters:
C
C   parameter(MAXXY=MAXX*MAXY*MAXZ,PMX=999.,UNEST=-1.0e21,
C   +         EPSLON=1.0e-6,VERSION=2.000)
C

```

```

c ACORN parameters:
c
  parameter(KORDEI=12,MAXOP1=KORDEI+1,MAXINT=2**30)
c
c Variable Declaration:
c
  character datafl*40,outfl*40,dbgfl*40
  real    c0(1),cc(MAXNST),aa(MAXNST),ang1(MAXNST),
+        ang2(MAXNST),ang3(MAXNST),anis1(MAXNST),anis2(MAXNST),
+        x(MAXDAT),y(MAXDAT),z(MAXDAT),vr(MAXDAT)
  integer nst(1),it(MAXNST)
c
c Common Blocks:
c
  common /iaco/  ixv(MAXOP1)

  common/genral/ idbg,ldbg,lout,nsim,nx,xmn,xsiz,ny,ymn,ysiz,
+              nz,zmn,zsiz

  common/vargar/ nst,it,aa,c0,cc,ang1,ang2,ang3,anis1,anis2

  common/charar/ datafl,outfl,dbgfl

  common/dataar/ nd,x,y,z,vr,tmin,tmax

```

**EK – 2**

**VARCON PROGRAMININ FORTRAN KODU**

```

program varcon
c  Tercan, A.E., 2004
implicit real*8 (a-h,o-z)
parameter (max=50,mg=101)
dimension dis(max),varexp(max,mg),varmod(max),varort(max,mg)
character datafl*40, outfl*40, outfl2*40, outfl3*40
data lpar/10/, ldat/20/, lout/30/, lout2/40/, lout3/50/
c
open(lpar,file='c:\msdev\projects\varcon\varcon.par',
&      status='old')
read(lpar,'(a40)') datafl
read(lpar,'(a40)') outfl
read(lpar,'(a40)') outfl2
read(lpar,'(a40)') outfl3
read(lpar, * ) nlag,nsim
open(ldat,file=datafl,status='old')
open(lout,file=outfl)
open(lout2,file=outfl2)
open(lout3,file=outfl3)
c
do i=1,nlag
  read(ldat,*) dis(i),(varexp(i,j),j=1,nsim),varmod(i)
  varort(i,1)=varexp(i,1)
end do
c
do i=1,nlag
  do j=1,nsim-1
    sum=0.0
    do k=1,j+1
      sum=sum+varexp(i,k)
    end do
    varort(i,j+1)=sum/float(j+1)
  end do
  write(lout2,110) dis(i),(varort(i,j),j=1,nsim),varmod(i)
end do
c
do j=1,nsim
  sum=0.0
  sum2=0.0
  inum=0
  do i=1,nlag
    err=varort(i,j)-varmod(i)
    err2=err*err
    sum=sum+err2
    varmod2=varmod(i)*varmod(i)
    sum2=sum2+varmod2
    varphi=sqrt(err2/varmod2)
    inum=inum+5
    write(lout3,120) j,inum,varphi
  end do

```

```
    phi=sqrt(sum/sum2)
    write(lout,100) j,phi
end do
stop
100 format(i4,1x,f8.3)
110 format(f5.0,1x,100(f5.3,1x),f5.3)
120 format(i3,1x,i3,1x,f6.3)
end
```

**EK – 3**

**KS TEST PROGRAMININ FORTRAN KODU**

```

program kstest
c  Tercan, A.E., 2004
implicit real*8 (a-h,o-z)
parameter (maxsim=101)
dimension sim(maxsim)
character datafl*40, outfl*40
data lpar/10/, ldat/20/, lout/30/, pi/3.1416/, cons/1.07/
c
open(lpar,file='c:\msdev\projects\kstest\kstest.par',
+      status='old')
read(lpar,'(a40)')datafl
read(lpar,'(a40)')outfl
read(lpar, * )nsim
read(lpar, * )ndat
open(ldat,file=datafl,status='old')
open(lout,file=outfl)
c
fns=float(nsim)
cv=cons/sqrt(fns)
do i=1,ndat
  read(ldat,*)x,y,(sim(j),j=1,nsim)
c
  n1=nsim-1
  do j=1,n1
    do k=1,n1
      if(sim(k).gt.sim(k+1))then
        t1=sim(k)
        sim(k)=sim(k+1)
        sim(k+1)=t1
      end if
    end do
  end do
c
  sum=0.0
  sum2=0.0
  do j=1,nsim
    sum=sum+sim(j)
    sum2=sum2+sim(j)*sim(j)
  end do
  ave=sum/fns
  var=(sum2/fns)-(ave*ave)
  sd=sqrt(var)
c
c  Calculate f0
eb=-99999.9
do j=1,nsim
  svar=(sim(j)-ave)/sd
  ff=0.5*(1.0+sqrt(1.0-exp(-2.0*svar*svar/pi)))
  if (svar.gt.0.0)then
    f0=ff

```

```

else
  f0=1.0-ff
end if
fn=float(j)/fns
fn_1=float(j-1)/fns
dplus=fn-f0
dminus=f0-fn_1
dmax=amax1(dplus,dminus)
if (dmax.gt.eb) then
  eb=dmax
end if
if (eb.lt.cv) then
  indi=1
else
  indi=0
end if
end do
write (lout,100) x,y,ave,var,indi
end do
stop
100 format (2(f8.0,1x),2(f8.2,1x),i1)
end

```