

## A TÖBBDIMENZIÓS NORMÁLIS ELOSZLÁSFÜGGVÉNY MONTE CARLO INTEGRÁLÁSSAL TÖRTÉNŐ KISZÁMITÁSÁNAK SZÁMITÓGÉPES TAPASZTALATAI

Deák István

### 1. Bevezetés

A többdimenziós normális eloszlásfüggvény konkrét értékeinek kiszámítása több gyakorlati példában szükséges lehet. Például Prékopa [3] STABIL modelljének megoldása folyamán, amely a következő alakú:

$$(1.1) \quad P\{g_i(\underline{x}) \geq \beta_i, \quad i = 1, \dots, n\} \geq P, \\ \underline{a}_i' \underline{x} \geq b_i, \quad i = 1, \dots, m, \\ \min f(\underline{x});$$

ahol a  $\beta_1, \beta_2, \dots, \beta_n$  valószínűségi változók együttes eloszlása  $n$  dimenziós normális eloszlás. A cikkben leírunk egy szubrutinrendszert, amely egy Monte Carlo integrálási technikával számítja ki a  $\Phi(\underline{h})$  konkrét értékeit, ahol  $\Phi(\underline{h})$  a zérus várható értékű, 1 szórású és  $R$  korrelációs mátrixú normális eloszlásfüggvény, vagyis

$$(1.2) \quad \Phi(\underline{h}) = \frac{1}{(2\pi)^{n/2} |R|^{1/2}} \int_{-\infty}^h \dots \int_{-\infty}^h \exp\left\{-\frac{1}{2} \underline{x} R^{-1} \underline{x}\right\} d\underline{x}.$$

A felhasznált technikát úgy választottuk, hogy egy konkrét  $\Phi(\underline{h})$  értéket a lehető legrövidebb időn belül számítsa ki a számítógép és a szubrutinok tetszőleges  $R$  korrelációs mátrix és  $\underline{h}$  vektor esetén működjenek az  $n = 12$  dimenziós esetben is. Az itt leírt technikát alkalmaztuk a magyar villamosenergiaiparra illesztett STABIL modell számítógépes kiszámítása során [3].

A 2. szakaszban az alkalmazott Monte Carlo technika elvi leírását adjuk meg, a 3. szakaszban a számítógépes program szerkezetét tárgyaljuk – különös tekintettel a minél gyorsabb lefutásra. A 4. szakaszban a számítógépes tapasztalatokat, valamint a szubrutinok sebességére vonatkozó időeredményeket írjuk le.

### 2. A többdimenziós normális eloszlásfüggvény konkrét értékeinek kiszámítása Monte Carlo integrálással

Legyen feladatunk az

$$(2.1) \quad I_1 = \Phi(\underline{h}) = \int_{-\infty}^h \dots \int_{-\infty}^h \varphi(\underline{x}) d\underline{x}$$

integrál meghatározása, ahol  $\varphi(\underline{x})$  a 0 várható értékű, 1 szórású,  $R$  korrelációs mátrixú  $n$  dimenziós normális sűrűségfüggvény, vagyis

$$\varphi(\underline{x}) = \frac{1}{(2\pi)^{n/2} |R|^{1/2}} \exp \left\{ -\frac{1}{2} \underline{x} R^{-1} \underline{x} \right\},$$

ahol  $R$  pozitív definit szimmetrikus mátrix, melynek a főátlójában 1-esek állnak. Legyen  $D$  a következő halmaz:

$$(2.2) \quad D = \{ \underline{x} | \underline{x} \leq \underline{h} \}.$$

Az  $I_1$  érték meghatározása ekvivalens a  $\varphi(\underline{x})$  függvény  $D$  tartomány feletti integráljának kiszámításával. A gyakorlati kiszámítás folyamán a  $D$  halmaz helyett a

$$(2.3) \quad D^* = \{ \underline{x} | -\underline{b} \leq \underline{x} \leq \underline{h} \}$$

tartomány felett integrálunk, ahol a  $\underline{b}$  vektor mindegyik komponensének az értéke néggyel egyenlő. Az integrálási tartomány ilyen csonkolása által elkövetett hiba elhanyagolhatóan kicsi az alkalmazott Monte Carlo módszer hibáihoz képest, viszont egyszerűbbé és gyorsabbá teszi a számításokat.

Az  $I_1$  érték becslésére használjuk a

$$(2.4) \quad \Theta_1 = \frac{1}{N} \prod_{j=1}^n (h_j + b) \cdot \sum_{i=1}^N \varphi(\underline{v}^{(i)})$$

valószínűségi változót, ahol  $\underline{v}^{(i)}$ ;  $i = 1, \dots, N$  olyan valószínűségi vektorváltozók, amelyek egyenletes eloszlásúak a  $D^*$  tartományban. Azért kellett a  $D$  tartomány helyett a  $D^*$  korlátozott tartományt bevezetni, hogy egyenletes eloszlású pszeudovéletlen vektorokat tudjunk az integrálási tartományban előállítani. A (2.4) képlet lényegében a  $D^*$  tartomány területével szorozza az  $e$  terület feletti függvényértékek számtani átlagát, ezt a  $\Theta_1$  becslést integrálközépnek nevezzük. A számítás gyakorlatilag úgy történik, hogy a  $\underline{v}^{(i)}$ ,  $i = 1, \dots, N$  valószínűségi vektorváltozónak  $N$  darab  $\underline{r}^{(i)}$ ,  $i = 1, \dots, N$  realizációját vesszük és ezek segítségével a  $\Theta_1$  valószínűségi változónak a várható értékére egy becslést kapunk.

Vizsgáljuk meg a (2.4) képletben alapuló eljárás hibáját. Vezessük be a

$$(2.5) \quad \xi_i = \prod_{j=1}^n (h_j + b) \varphi(\underline{v}^{(i)})$$

jelölést;  $\Theta_1$  a  $\xi_i$  valószínűségi változók számtani közepe, vagyis

$$\Theta_1 = \frac{1}{N} \sum_{i=1}^N \xi_i.$$

Erre pedig a Csebisev egyenlőtlenség segítségével kapjuk, hogy

$$P \{ |\Theta_1 - M(\Theta_1)| \leq x_p D(\Theta_1) \} \geq p,$$

valamilyen  $p$  valószínűségi szinttel. Az  $x_p$  változó értéke a  $p$  döntési szinttől függ,

például  $p = 0,997$  szinten  $x_p = 3,0$  lesz. Ha figyelembe vesszük még, hogy

$$M(\Theta_1) = I_1, \quad D^2(\Theta_1) = \frac{D^2(\xi_i)}{N},$$

akkor valamilyen,  $p$ -hez közeli valószínűséggel igaz lesz, hogy

$$(2.6) \quad |\Theta_1 - I_1| \leq x_p \frac{D(\xi_i)}{\sqrt{N}}.$$

Tehát az integrálás pontossága fordítottan arányos a mintapontok számának négyzetgyökével és egyenesen arányos a  $\xi_i$  valószínűségi változók szórásával, vagyis a hiba nagysága nagymértékben a  $D(\varphi(\underline{v}^{(i)}))$  szórástól függ. Kézenfekvőnek látszik a  $\xi_i$  valószínűségi változók szórásának csökkentésével javítani a kiszámítási eljárást. Az irodalomban javasolt többféle szóráscsökkentő eljárást (importance sampling, control variates) kipróbálva nem lehetett a gépidőt – azonos pontosságú eredmény elérését megkívánva – csökkenteni; bár kevesebb mintapontot kellett felvenni, de a kiszámítás bonyolultsága miatt megnőtt a futtatási idő.

A felhasznált  $\xi_i$  valószínűségi változók szórását a következő ötlet segítségével lehet azonban csökkenteni. Legyen  $K$  a következő  $n$  dimenziós kocka:

$$K = \{ \underline{x} \mid -\underline{b} \leq \underline{x} \leq \underline{b} \}.$$

Tekintettel arra, hogy  $\varphi(x)$  értéke a  $K$  kockán kívül már elég kicsi, ha  $\underline{b} \geq 4$ , egy elhanyagolható  $\delta$  hibával fennáll az

$$1 = \int_{R^n} \dots \int \varphi(\underline{x}) d\underline{x} = \int_K \dots \int \varphi(\underline{x}) d\underline{x} + \delta$$

egyenlőség. Ezért a  $\varphi(\underline{x})$  függvénynek a  $K - D^*$  tartomány feletti integrálját  $I_2$ -vel jelölve

$$(2.7) \quad \begin{aligned} I_1 &= \int_{D^*} \dots \int \varphi(\underline{x}) d\underline{x} = 1 - \int_{R^n} \dots \int \varphi(\underline{x}) d\underline{x} + \int_{D^*} \dots \int \varphi(\underline{x}) d\underline{x} = \\ &= 1 - \int_{K-D^*} \dots \int \varphi(\underline{x}) d\underline{x} - \delta = 1 - I_2 - \delta \approx 1 - I_2, \end{aligned}$$

ahol a  $\approx$  jel azt mutatja, hogy a gyakorlatban elhanyagolható hibával egyenlőség áll fenn. Az  $I_2$  integrált a következő valószínűségi változóval becsüljük:

$$(2.8) \quad \Theta_2 = \frac{1}{N} [(2b)^n - \prod_{j=1}^n (h_j + b)] \cdot \sum_{i=1}^N \varphi(\underline{v}^{(i)});$$

ahol a  $\underline{v}^{(i)}$ ,  $i = 1, \dots, N$  valószínűségi változók egyenletes eloszlásúak a  $K - D^*$  tartományban. Tehát (2.8) szerint a  $\Theta_2$  becslés a  $K - D^*$  tartományban a függvényértékek átlagának szorzataként áll elő.

A gyakorlati alkalmazásokban az  $I_1$  integrál értéke 0.80 – 0.95 körül szokott lenni, így az  $I_2$  valószínűség 0.05 – 0.20 között változik. Ha  $I_1$  helyett az  $I_2$  értéket becsüljük meg, vagyis a  $\Theta_1$  összeg helyett a  $\Theta_2$  összeget számítjuk ki, akkor a  $\xi_i = \varphi(\underline{v}^{(i)})$

valószínűségi változók szórását azáltal csökkentjük, hogy a  $\varphi(v^{(i)})$  értékek kisebbek a  $K - D^*$  tartományban, mint a  $D^*$  tartományban.

Tehát érdemesebb úgy meghatározni az  $I_1$  integrál értékét, hogy kiszámítjuk az  $I_2$  egy becslését a  $\Theta_2$  összeg segítségével egy adott  $\underline{h}$  vektor és rögzített  $N$  esetén, majd az  $I_1 \approx 1 - I_2$  közelítő egyenlőség felhasználásával számítjuk ki az  $I_1$  értéket. Ha az  $I_1$  érték közel van 1-hez, akkor ez a kiszámítási eljárás a (2.6) hibabecslés szerint kisebb hibájú lesz, mintha az adott  $\underline{h}$  és  $N$  esetén az  $I_1$  értéket közvetlenül a  $\Theta_1$  összeg meghatározásával számítanánk ki. Mivel egy adott eloszlás és  $\underline{h}$  vektor esetén előre nem tudhatjuk, hogy  $I_1$  értéke 1-hez közeli lesz vagy sem, a következő kritériumot használtuk a szubrutinokban annak eldöntésére, hogy a  $\Theta_1$  összeg vagy a  $\Theta_2$  összeg meghatározásával számítsuk ki  $I_1$  értékét. Jelölje  $T_{D^*}$ ,  $T_K$ ,  $T_{K-D^*}$  rendre a  $D^*$ ,  $K$ ,  $K-D^*$  tartományok területét. Ha a

$$(2.9) \quad \frac{T_{D^*}}{T_K} < (0,6)^n$$

egyenlőtlenség teljesül, ahol  $n$  az eloszlás dimenziószáma, akkor az adott  $\underline{h}$  vektor esetében a  $\Theta_1$  összeget számítjuk, egyébként a  $\Theta_2$  átlag segítségével becsljük az  $I_2$  értéket és ebből számítjuk ki az  $I_1$  értékét. Szemléletesen nézve a (2.9) feltétel azt vizsgálja, hogy a  $\varphi(\underline{x})$  sűrűségfüggvénynek az origóban felvett csúcsa és a körülötte levő nagyobb értékek hová esnek és azt az eljárást választjuk, amelynek alkalmazásával ezeket a nagyobb  $\varphi(\underline{x})$  értékeket elkerülhetjük. Ezzel a választással a  $\Theta_i$  összegben felhasznált valószínűségi változók szórása kisebb lesz.

A számítógépes program további gyorsítását lehet elérni a következő gondolat segítségével. A  $K - D^*$  tartományban egyes helyeken olyan kicsi lesz a  $\varphi(v^{(i)})$  értéke, hogy még a  $T_{K-D^*}$  területtel megszorozva is elhanyagolhatóan kis értéket kapunk. A  $\Theta_2$  átlag kiszámítását így a következő módon végezzük. Ha

$$(2.10) \quad T_{K-D^*} \cdot \varphi(v^{(i)}) < 0,001$$

akkor ez a tag a  $\Theta_2$  összegben csak  $\frac{1}{N} \cdot 0,001$  értéket ad, így ez a tag a  $\Theta_2$  átlagból elhagyható. A  $\Theta_2$  átlag meghatározása a következő számítógépes lépésekből tevődik össze:

1.) Generálunk egy  $v^{(i)}$  egyenletes eloszlású pontot a  $K$  kockában, megvizsgáljuk, hogy a  $K - D^*$  tartományba esik-e, ha nem, akkor új véletlen pontot generálunk, egyébként a 2.) lépés következik.

2.) Kiszámítjuk a  $\varphi(\underline{x})$  sűrűségfüggvényben szereplő

$$S_i = -\frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n R_{jk} v_k^{(i)} v_j^{(i)}$$

kvadrátikus alakot, ahol  $R_{jk}$  az  $R$  korrelációs mátrix inverzének  $(j,k)$ -adik eleme.

3.) Meghatározzuk az  $\exp(S_i)$  értéket és hozzáadjuk a  $\Theta_2$  átlag már eddig kiszámított részéhez.

4.) Ha már  $N$  darab  $\underline{v}^{(i)}$  vektort generáltunk a  $K - D^*$  tartományban, akkor a  $\Theta_2$  összeget beszorozzuk a még hiányzó állandókkal.

Lényeges az, hogy a (2.10) kritérium teljesülését már a 2.) lépés után ellenőrizhetjük, ugyanis a

$$T_{K-D} \frac{1}{(2\pi)^{\frac{n}{2}} |R|^{\frac{1}{2}}} \cdot \exp\{S_i\} \leq 0,001$$

feltételből átalakításokkal kapjuk, hogy ha az

$$(2.11) \quad S_i \leq \ln \left\{ \frac{1}{T_{K-D^*}} 0,001 \cdot (2\pi)^{\frac{n}{2}} \cdot |R|^{\frac{1}{2}} \right\}$$

egyenlőtlenség fennáll (ami ekvivalens a (2.10) feltétellel), akkor a 3. lépést ennek a  $\underline{v}^{(i)}$  vektornak az esetében már nem kell elvégezni, a  $\Theta_2$  összegben a megfelelő tagot zérusnak tekintjük.

A fenti gondolatmenet segítségével további módosítást végzünk még el. A  $\underline{b} = \underline{4}$  csonkoló vektor helyett egy olyan  $\underline{c}$  vektort vehetünk, amelyre teljesül az, hogy ha  $|\underline{x}| \geq \underline{c}$  akkor  $T_{K-D^*} \cdot \varphi(\underline{x}) \leq 0,001$  lesz. Vagyis újabb csonkolást hajtunk végre; a  $D^*$  vagy a  $K - D^*$  tartomány helyett az integrálást a  $D^{**}$  vagy a  $K^* - D^{**}$  tartomány felett végezzük, ahol

$$(2.12) \quad \begin{aligned} D^{**} &= \{ \underline{x} \mid -\underline{c} \leq \underline{x} \leq \underline{h} \}, \\ K^* &= \{ \underline{x} \mid -\underline{c} \leq \underline{x} \leq \underline{c} \}. \end{aligned}$$

Ennek a lépésnek abban áll a jelentősége, hogy  $\underline{c} \leq \underline{h}$  és ezért  $T_{D^{**}} \leq T_{D^*}$  és  $T_{K^*} \leq T_K$ . Így kisebb területű tartományban véve fel az  $N$  számú egyenletes eloszlású  $\underline{v}^{(i)}$  pontokat, jobb, pontosabb eredményt kapunk. A gyakorlatban a  $\underline{c}$  vektor komponenseire 2,5 – 4,0 közötti értékeket kapunk az  $R$  korrelációs mátrixtól függően.

### 3. A számítógépes program szerkezete

A szubrutinrendszer az MTA CDC 3300 számítógépére készült FORTRAN nyelven és 18 szubrutinból áll. Ezek hosszabb fejlesztési munka eredményeként készültek el, az itt leírt rendszer a negyedik változat. Az előző változatokról lásd [1].

A programok megírásánál a minél kisebb helyfoglalást és a lehető legrövidebb futási idő elérését tűztük ki célul. Az első szempont azért fontos, mert ezeket a szubrutinokat más programokon belül használják fel, a második pedig azért, hogy egy elfogadható gépi időn belül lefussanak a programok. A két szempont némileg ellentétes; az előző változat lényegesen több utasításból állt, de valamivel gyorsabb volt. Két szubrutin közül azt lehet gyorsabbnak tekinteni, amely azonos  $p$  megbízhatósági szinten azonos várható hibával rövidebb idő alatt számítja ki egy adott eloszlás esetén ugyanazt a  $\Phi(\underline{h})$  függvényértéket. (Ugyanis különböző  $\underline{h}$

vektorok esetén a  $\Phi(\underline{h})$  érték kiszámítása különböző gyorsaságú lesz).

A szubrutinok egy része előkészítő műveleteket végez. Meghatározzuk az adott korrelációs mátrix inverzét, kiszámítják az előző szakaszban említett  $D^{**}$  tartomány előállításához szükséges  $\underline{c}$  csonkoló vektor komponenseit, a (2.11) egyenlőtlenség jobb oldalán szereplő kifejezés értékét és egyéb állandókat.

Egy adott  $\underline{h}$  vektor esetén a (2.6) alapján meg tudjuk becsülni, hogy adott  $|\Theta_i - I_i|$ ,  $i = 1, 2$  hiba esetén és adott  $p$  biztonsági szinthez mekkora  $N$  mintaszámot kell választani a Monte-Carlo integráláshoz.

A  $D(\xi_i)$  szórását a

$$(3.1) \quad D^*(\xi_i) \approx \frac{1}{N} \sum_{i=1}^N \xi_i^2 - \left( \frac{1}{N} \sum_{i=1}^N \xi_i \right)^2$$

empirikus szórásnégyzettel becsüljük. Az egyes szubrutinok a következő módon működnek:

- ELJAV2 megadott  $N$  esetén a  $D^{**}$  tartomány felett integrál.
- ELJAV3 megadott  $N$  esetén a  $K^* - D^{**}$  tartomány felett integrál és a (3.1) összefüggés segítségével kiszámítja, hogy különböző hibák és megbízhatósági szintek esetén mekkora  $N$  mintaszám szükséges.
- ELJAV4 megadott  $N$  esetén a  $K^* - D^{**}$  tartomány felett integrál.
- ELJAV5 megadott megbízhatósági szintű megadott hibához a (3.1) összefüggés felhasználásával számítja ki a valószínűséget a  $D^{**}$  tartomány feletti integrálással.
- ELJAV6 ugyanugy működik mint ELJAV5, csak a  $D^{**}$  tartomány helyett a  $K^* - D^{**}$  tartomány felett integrál.

A minél rövidebb futási idő elérése érdekében szükségessé vált a véletlenszám generátor ügyes megválasztása és a sűrűségfüggvény kitevőjének egyszerűsített kiszámítása. Az

$$S_i = -\frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n R_{jk} v_j^{(i)} v_k^{(i)}$$

jelölést használva a

$$(3.2) \quad \Theta_2 = \frac{1}{N} \cdot T_{K^* - D^{**}} \cdot \sum_{i=1}^N \frac{1}{(2\pi)^2 |R|^{\frac{1}{2}}} \exp(S_i)$$

kifejezés kiszámításának során, az összegképzés minden lépésben egy darab, a  $K^* - D^{**}$  tartományban egyenletes eloszlású  $\underline{v}^{(i)}$  valószínűségi vektorváltozóra van szükségünk. A következő igen egyszerű és emiatt gyors módszert használtuk a  $\underline{v}^{(i)}$  vektorok előállítására. Jelöljük  $\underline{v}_j^{(i)}$ -vel a  $\underline{v}^{(i)}$  vektor  $j$ -edik komponensét, a  $\underline{v}^{(i+1)}$  vektort a következő rekurzív összefüggésekkel állítottuk elő:

$$v_j^{(i+1)} = v_{j+1}^{(i)}, \quad j = 1, \dots, n-1,$$

(3.3)

$$v_n^{(i+1)} = v_1^{(i)} + v_n^{(i)}; \quad (\text{mod } b);$$

ahol  $b$  azonos a  $D^*$  tartomány meghatározásában szereplő konstanssal. Tekintettel arra, hogy ilyen kongruenciás véletlenszám generátorok felhasználásával óvatosan kell eljárunk (lásd [2]), ezért minden 300-ik lépésben teljesen újrageneráljuk a  $\underline{v}^{(i)}$  vektort egy másik, karakterkeverésen alapuló véletlenszám generátorral. A (3.3) összefüggések segítségével előállított  $\underline{v}^{(i)}$  vektort megvizsgáljuk, hogy a  $K^* - D^{**}$  tartományba esik-e, ha igen, akkor kiszámítjuk az  $S_i$  kifejezés értékét ezzel a  $\underline{v}^{(i)}$  vektorral, egyébként előállítjuk a  $\underline{v}^{(i+1)}$  vektort.

A másik gyorsítási lehetőség az  $S_i$  érték kiszámításánál adódott. Az  $R^{-1} = \{R_{jk}\}_{j,k}$  mátrix szimmetrikus és így ezt a  $\sum$  jel előtti  $-\frac{1}{2}$  tényezővel egybeolvasztva az  $S_i$  kifejezés felírható egy  $Q$  háromszögmátrix segítségével a következő alakban:

$$(3.4) \quad S_i = -\frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n R_{jk} v_j^{(i)} v_k^{(i)} = \sum_{j=1}^n \sum_{k=1}^j Q_{jk} v_j^{(i)} v_k^{(i)}$$

$$\text{ahol} \quad Q_{jj} = -\frac{1}{2} R_{jj} \quad \text{és} \quad Q_{jk} = -R_{jk}, \quad \begin{array}{l} j = 1, \dots, n, \\ k = 1, \dots, j. \end{array}$$

Ezzel a felirással az  $S_i$  kiszámításánál szereplő tagok számát  $n^2$ -ről  $\frac{n(n+1)}{2}$ -re csökkentettük. További gyorsítást ad az a módosítás, ha az  $S_i$  kétszeres összegét nem egy kétszeres  $D\Phi$  ciklus segítségével számítjuk ki, – ahogyan a legkézenfekvőbb lenne – hanem részletesen kiírjuk az összeget. Ezzel a kissé hosszadalmas felirással elkerüljük a szimbolikus indexezés miatt szükséges időt; a futási idő a harmadára csökken ezáltal (bővebbet erről a következő szakaszban írunk).

#### 4. Számítógépes tapasztalatok és futási idők

A szubrutinok FORTRAN nyelven készültek, ellenőrzésüket több különböző  $\underline{h}$  vektor és korrelációs mátrix esetén végeztük el. A különböző korrelációs mátrixokat véletlenszám generátor segítségével állítottuk elő a következő módon. Generáltuk  $\underline{u}^{(1)}, \underline{u}^{(2)}, \dots, \underline{u}^{(n)}$   $n$  darab  $n$  dimenziós független vektort, amelyek komponensei a  $(0,1)$  intervallumban egyenletes eloszlású és független pszeudovéletlen számok voltak. Ezekre alkalmaztuk az

$$(4.1) \quad \underline{u}^{(i)*} = \frac{\bar{u}^{(i)}}{\sqrt{|u^{(i)}|}}$$

transzformációt, ahol

$$|\underline{u}^{(i)}| = \sum_{j=1}^n u_j^{(i)2}$$

Legyen az  $R$  mátrix  $(i,j)$ -edik eleme az  $\underline{u}^{(i)*} \underline{u}^{(j)*}$  skalárszorzat. Az ilyen módon előállított  $R = \{ u^{(i)*} u^{(j)*} \}_{i,j}$  mátrix szimmetrikus, pozitív definit lesz, a főátlóban mindenütt 1 áll, így tekinthető a standardizált többdimenziós normális eloszlás korrelációs mátrixának. A  $\Phi(\underline{h})$  érték kiszámításánál szereplő  $\underline{h}$  vektort szintén véletlenszám generátor segítségével állítottuk elő; a  $\underline{h}$  vektor komponensei egymástól független, a  $(0,5)$  intervallumban egyenletes eloszlású pszeudóvéletlen számok voltak.

Több ilyen módon előállított  $R$  mátrix és  $\underline{h}$  vektor esetén futtattuk le a szubrutinokat és ebben a szakaszban az így szerzett tapasztalatokat közöljük. Először a  $\varphi(\underline{v}^{(i)})$  értékek kiszámításának részletes időeredményeivel foglalkozunk, majd az egyes szubrutinok futási idejével és pontosságával kapcsolatos eredményeket adjuk meg és végül a többdimenziós normális eloszlásfüggvény Monte Carlo integrálással történő kiszámítására vonatkozó következtetéseinket adjuk meg.

A  $\xi_i = \varphi(\underline{v}^{(i)})$  értékek kiszámítása a 2. szakaszban leírt 1.), 2), 3), lépések szerint történik. Az  $S_i$  exponensek kétszeres  $D\Phi$  cikluson belüli (3.4) összefüggés szerinti kiszámítása és az  $S_i$  értékeknek a (3.4) szerinti, de részletesen kiírt alakban történő kiszámítása lényeges időbeli különbséget jelent. Az exponens kiszámítása három dimenzióesetén kiírt alakban a következő:

$$S = Q(1,1) * P(1) * P(1) + P(2) * (Q(1,2) * P(2) + Q(2,2) * P(2)) + P(3) * (Q(1,2) * P(1) + Q(2,3) * P(2) + Q(3,3) * P(3)),$$

ahol  $P(I)$  jelöli a véletlen vektor komponenseit.

Ezzel a felírási móddal további szorzásokat hagyhatunk el a (3.4) képlethez képest. Az 1), 2) és 3) lépéseket külön-külön ezerszer lefuttatva a következő időeredményeket kaptuk

1) Véletlen vektor előállítás	0,6 sec	1,3 sec
2) $S_i$ kétszeres $D\phi$ ciklussal való számítása	2,1 sec	11,7 sec
3) $\exp(S_i)$ kiszámítása	0,6 sec	0,4 sec
2)* $S_i$ (4.2) szerinti számítása	0,7 sec	3,6 sec
Igy $N = 1000$ esetén a $\Theta_2$ kiszámításához szükséges idő	1,9 sec	5,3 sec

1. Táblázat

Az  $\exp(S_i)$  érték kiszámításának ideje azért különböző a 4 és a 12 dimenziós esetben, mert a (2.11) szerinti konstans miatt az 1000 eset közül kevesebben kellett ténylegesen kiszámítani az  $\exp(S_i)$  értéket. Az 1. táblázatban szereplő idők átlagértékek, más kezdőértékekkel indítva a véletlenszám generátorokat, kissé eltérő időket kaphatunk.

A szubrutinok helyes működésének ellenőrzését úgy végeztük el, hogy az egységmátrixot vettük korrelációs mátrixnak, ez a független esetnek felel meg. Ilyenkor a keresett  $\Phi(\underline{h})$  érték a  $\Phi(\underline{h}_i)$  egydimenziós valószínűségek szorzataként adódnak. Közöljük néhány ilyen eset futási eredményeit. 4 dimenzióban, ha a  $\underline{h}$  vektor (4,00; 3,25; 4,00; 3,51), akkor a pontos értéke a valószínűségnek 1.00;

	A valószínűség	A hiba 95%-os biztonsággal	N a mintaszám	Idő
ELJAV2 $D^{**}$ felett integrál	1,000	2,02	2000	2,68 sec
ELJAV3 $K^* - D^{**}$ felett integrál	0,999	0,0002	4000	4,48 sec
ELJAV4 $K^* - D^{**}$ felett integrál	1,000	0,00001	182079	146,7 sec
ELJAV6 $K^* - D^{**}$ felett integrál	0,999	0,0002	4000	3,4 sec

2. Táblázat

5 dimenzióban, ha a  $\underline{h}$  vektor (4,00; 4,00; 1,22; 0,10; 3,59) akkor a valószínűség pontos értéke 0,4797, a számított értékek

	A valószínűség	A hiba 95%-os biztonsággal	N a mintaszám	Idő
ELJAV2	1,000	1,69	2500	3,2 sec
ELJAV3	0,562	0,117	5000	13,8 sec
ELJAV4	0,492	0,104	10751	27,5 sec
ELJAV6	0,483	0,036	78829	130,0 sec

3. Táblázat

A 6 dimenziós független esetben, ha a  $\underline{h}$  vektor a következő (4,00; 1,29; 0,55; 2,70; 3,41; 0,57,) akkor a valószínűség pontos értéke 0,455, a szubrutinok által adott eredmények pedig a következők:

	A valószínűség	A hiba 95%-os biztonsággal	N a mintaszám	Idő
ELJAV2	1,000	0,679	3000	4,73 sec
ELJAV3	0,426	0,188	6000	22,6 sec
ELJAV4	0,417	0,173	7207	21,5 sec
ELJAV6	0,484	0,030	172854	388,6 sec

4. Táblázat

12 dimenziós független esetben, ha a  $h$  vektor a következő (1,33; 4,00; 8,57; 0,30; 0,74; 4,00; 0,26; 0,25; 1,38; 1,56; 2,51; 4,00) akkor a valószínűség pontos értéke 0,172 a számítottértékek pedig

	A valószínűség	A hiba 95%-os szinten	N a mintaszám	Idő
ELJAV2	0,048	0,05	6000	26,1 sec
ELJAV3	0,489	0,52	12000	152,7 sec
ELJAV4	0,004	0,62	12530	77 sec

5. Táblázat

A teljesség kedvéért közöljük egy 4 dimenziós és egy 12 dimenziós korrelált esetet is. 4 dimenzióban:

	A valószínűség	A hiba 95%-os szinten	N a mintaszám	Idő
ELJAV2	1,000	2,40	2000	5,4 sec
ELJAV4	0,922	0,04	6000	13,6 sec
ELJAV6	0,921	0,03	14060	30,9 sec

6. Táblázat

12 dimenzióban:

	A valószínűség	A hiba	N mintaszám	Idő
ELJAV2	0,0001	0,0001	6000	25,2 sec
ELJAV3	0,971	0,038	6000	70,3 sec
ELJAV4	0,545	0,656	13000	68 sec

7. Táblázat

A fenti eredményekből is látható, hogy minél nagyobb a keresett valószínűség, annál könnyebben tudjuk a valószínűség értékét a  $K^* - D^{**}$  tartomány feletti integrálással kiszámítani. A szubrutinok magasabb dimenzióban (lásd 7. Táblázat első sora) egyes esetekben megbízhatatlan eredményt adnak a kis mintaszám miatt; ebben az esetben a 95%-os biztonsági szinten  $\pm 0.05$  nagyságú hibához (az ELJAV3 által kiszámított) szükséges mintaszám  $N = 2\,000\,000$  lenne, ehhez körülbelül 110 perc gépidőre lenne szükség. Más korrelációs mátrix és  $\underline{h}$  vektor esetén is a  $\pm 0.05$  nagyságú hibával terhelt eredmény kiszámításához a 12 dimenziós esetben 60-120 sec. idő szükséges (kedvező esetben). Így végkövetkeztetésképpen kimondhatjuk, hogy 8-12 dimenzió esetén Monte Carlo integrálással csak 0.90-nél nagyobb valószínűség esetén tudjuk a valószínűséget meghatározni 2 percnél rövidebb idő alatt, 8 dimenzióig pedig gyakorlatilag minden esetet ki tudunk számítani. Egyes esetekben – attól a problémától függően, amelyben a többdimenziós normális valószínűségekre szükségünk van – az egyes szubrutinok jó felhasználásával gyorsan kaphatunk eredményt. Például Prékopa András STABIL modelljének [3] a futtatása során a négydimenziós valószínűségeket körülbelül 80 alkalommal számította ki a gép, összesen mintegy 10 percre volt szükség ehhez.

A várakozásnak megfelelően egy konkrét esetben a mintaszám négyzetgyökével arányosan csökkent a hiba, viszont az irodalom fellelhető utasításokkal ellentétben a kiszámításhoz szükséges idő nem lineárisan, hanem gyorsabban növekszik a dimenziószám emelkedésével.

## I r o d a l o m

- [1] Deák István: Egy sztochasztikus programozási modell számítógépes kiértékelése, MTA Számítástechnikai Központ Közlemények, (1972) 9. p. 33-49.
- [2] G. Marsaglia: Random numbers fall mainly in the planes, Proc. Nat. Acad. Sci. USA G 1 (1968) p. 25-28.
- [3] Prékopa A., Ganzer S., Deák I., Patyi K.: A STABIL sztochasztikus programozási modell kísérleti alkalmazása a magyar villamosenergiaiparra. Alkalmazott Matematikai Lapok 1(1975) 3-22.

## Р Е З Ю М Е

Значения многомерной функции нормального распределения с помощью интегральной техники Монте Карло, полученных в результате расчетов на ЭВМ.

Иштван Деак

В статье дается одна модифицированная Монте Карло интегральная техника для получения конкретных значений многомерной функции нормального распределения с любой коррекционной матрицей, самое большое в 12-и измерениях.

Подпрограммы разработанные автором основаны на базе описанной техники. Результаты машинного прогона изложены в статье.

## S U M M A R Y

Computer experiences of the evaluation of the multidimensional normal distribution function by a Monte Carlo integration technique

István Deák

The paper discusses a modified Monte Carlo integration technique to determine the concrete values of the multidimensional normal distribution function with arbitrary correlation matrix at least of 12 dimensions. The subroutines made by the author and based on the described technique were run and their results are presented.