Универзитет у Београду

Математички факултет

Мастер рад

Генерисање случајних величина методом стохастичке колокације

Аутор:
Благоје Ивановић

Ментор:
Др Бојана Милошевић

22. septembar 2018.

Садржај

1 Увод **2**

1.1 Мотивација и кратак опис методе 2

1.2 Интерполација 5

1.3 Гаусова квадратурна формула 12

1.4 Тестови сагласности 15

2 Генерисање случајних величина **21**

2.1 Генерисање једнодимензионих случајних величина 21

2.2 Одабир чворова интерполације 22

2.3 Генерисање вишедимензионих случајних величина 33

2.4 Генератори униформне и нормалне расподеле 37

3 Детаљи имплементације и резултати **43**

3.1 Опште напомене 43

3.2 Растезање мреже 43

3.3 Нумерички резултати 45

4 Закључак **58**

Литература **60**

Поглавље 1

Увод

1.1 Мотивација и кратак опис методе

Циљ је генерисати узорке из расподеле случајне величине Y која има функцију расподеле $F_Y(y) = P\{Y \leq y\}, \ y \in \mathbb{R}$. У наставку се претпоставља да је Y апсолутно непрекидна случајна величина, са носачем $[a, b]$, који може бити и неограничен ($a, b \in \overline{\mathbb{R}}$). Претпоставља се и да је $F_Y(y)$ строго растућа функција, односно, да постоји њен инверз $F_Y^{-1}(p) = \inf\{y \in \mathbb{R} \mid F_Y(y) \geq p\}, \ p \in [0, 1]$.

1.1.1 Метода инверзне трансформације

Најједноставнија метода генерисања узорка из расподеле Y је метода инверзне трансформације. Она полази од тврђења да за случајну величину $U = F_Y(Y)$ важи

$$P\{U \leq u\} = P\left\{F_Y(Y) \leq u\right\} = P\left\{Y \leq F_Y^{-1}(u)\right\} = F_Y\left(F_Y^{-1}(u)\right) = u,$$

тј. да U има униформну расподелу $\mathcal{U}[0, 1]$.

Аналогно, за неку случајну величину V из униформне расподеле, важи

$$P\left\{F_Y^{-1}(V) \leq y\right\} = P\left\{V \leq F_Y(y)\right\} = F_Y(y),$$

односно, $F_Y^{-1}(V)$ има исту расподелу као и Y.

Стога, како би се генерисао узорак $y_1, \ldots, y_n$ из расподеле Y, довољно је генерисати узорак $u_1, \ldots, u_n$ из униформне расподеле и узети $y_i = F_Y^{-1}(u_i), \ i = 1, \ldots, n$.

Велика предност овог приступа је његова једноставност, али и општост, јер описује поступак за генерисање узорака из било које апсолутно непрекидне расподеле, за коју је познат инверз. Потребно је само ефикасно генерисати случајне бројеве из интервала [0, 1], за шта постоје разни алгоритми, и применити инверзну трансформацију.

Међутим, често не постоји аналитички израз за $F_Y^{-1}(p)$, па чак ни за функцију расподеле $F_Y(y)$, будући да се она обично представља као интеграл функције густине, који се не може аналитички изразити, па се прибегава нумеричким апроксимацијама и специјализованим алгоритмима развијеним за генерисање посебних фамилија расподела.

То је већ случај и са једном од најпознатијих и најпримењенијих расподела - нормалном, чија је густина $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$, а функција расподеле просто интеграл $\Phi(x) = \int_{-\infty}^{x} \phi(t)\ dt$. И за нормалну расподелу су развијени посебни алгоритми који су у стању да брзо генеришу велики број узорака, а један од њих је описан у наставку, у делу 2.4.2.

Непостојање аналитичког израза за инверз функције расподеле је управо и највећи проблем методе инверзне трансформације. Наиме, нумеричко израчунавање вредности инверза може бити веома временски захтевно, што значајно утиче на перформансе методе, па при генерисању великих узорака време извршавања може постати превелико.

У овом раду је, уз коришћење принципа нумеричке математике, приказан начин да се превазиђу наведени проблеми методе инверзне трансформације за широку класу расподела и омогући изузетно брзо генерисање случајних величина на основу познавања инверза функције расподеле.

1.1.2 Основна идеја

Основна идеја генерисања случајних величина методом стохастичке колокације је да се инверз $F_Y^{-1}(p)$, који је „скуп“ за израчунавање, одреди само у неколико тачака, па се интерполира неким интерполационим полиномом $g(p) \approx F_Y^{-1}(p)$ на основу вредности у тим тачкама - чворовима интерполације.

Сада се узорак $y_1, \ldots, y_n$ из расподеле Y генерише као $g(u_1), \ldots, g(u_n)$, где је $u_1, \ldots, u_n$ узорак из униформне расподеле.

Тиме избегавамо велики број скупих израчунавања инверза и мењамо их брзим израчунавањем интерполационог полинома. Једно од главних питања је како повољно одабрати чворове у циљу добијања добре апроксимације са што мање тачака и то је једна од битних дискусија у овом раду.

Међутим, јавља се још један проблем. Претпоставимо да је Y случајна величина са расподелом чији је носач цео скуп $\mathbb{R}$, као што је, на пример, логистичка расподела. Тада је

$$0 < F_Y(y) < 1,\ \forall y \in \mathbb{R},$$

што повлачи да $F_Y^{-1}(p) \to -\infty$, када $p \to 0$, као и да $F_Y^{-1}(p) \to +\infty$, када $p \to 1$. Дакле имамо велике нестабилности при крајевима интервала $[0,1]$, што узрокује јако лошу апроксимацију.

Да бисмо решили тај проблем, уводимо још једну случајну величину X, са функцијом расподеле $F_X(x)$, коју умемо брзо да генеришемо. Како случајна величина $F_X(X)$ има униформну расподелу $\mathcal{U}[0,1]$, аналогно методи инверзне трансформације, показује се да случајна величина $F_Y^{-1}\big(F_X(X)\big)$ има исту расподелу као и Y.

Одатле закључујемо да узорак $y_1, \ldots, y_n$ из расподеле Y може да се генерише као $F_Y^{-1}\big(F_X(\xi_1)\big), \ldots, F_Y^{-1}\big(F_X(\xi_n)\big)$, где је $\xi_1, \ldots, \xi_n$ узорак из расподеле X.

Сада можемо да интерполирамо функцију $F_Y^{-1}\big(F_X(x)\big)$ интерполационим полиномом $g(x) \approx F_Y^{-1}\big(F_X(x)\big)$ и да узорак из Y генеришемо са $y_1 = g(\xi_1), \ldots, y_n = g(\xi_n)$.

Шта се овиме добија? Претпоставимо да X има нормалну расподелу. Тада је функција коју интерполирамо $F_Y^{-1} \circ \Phi : \mathbb{R} \to \mathbb{R}$, тј. прешли смо са интерполације на јединичном интервалу на интерполацију на целој реалној правој, што значајно повећава квалитет апроксимације и ублажава нестабилност, будући да више немамо екстремне грешке које смо имали близу крајева интервала.

Такође, једноставно се показује да се, у случају нормално расподељеног X, овим поступком смањују први изводи функције која се интерполира. Наиме, за свако $p \in (0,1)$, важи[1]

$$\frac{d}{dx} F_Y^{-1}\big(\Phi(x)\big)\Big|_{x=\Phi^{-1}(p)} = \frac{\phi(\Phi^{-1}(p))}{f_Y(F_Y^{-1}(p))} < \frac{1}{f_Y(F_Y^{-1}(p))} = \frac{d}{ds} F_Y^{-1}(s)\Big|_{s=p}.$$

[1] Φ и ϕ су, редом, густина и функција расподеле стандардне нормалне расподеле

1.2 Интерполација

Циљ интерполације је да неку функцију $f(x)$, задату на скупу од $n+1$ чворних тачака $x_0, \dots, x_n$ са $f(x_i) = y_i$, апроксимира полиномом $g_n(x) = a_0 + a_1 x + \dots + a_n x^n$ степена највише n, који се поклапа са функцијом у чворовима, односно важи $g_n(x_i) = y_i$. Полином $g_n(x)$ називамо интерполационим полиномом и кажемо да он интерполира функцију $f(x)$.

Једноставно се показује да је интерполациони полином јединствен. Наиме, нека је $p_n(x) = b_0 + b_1 x + \dots + b_n x^n$ такође полином који интерполира $f(x)$. Тада је и разлика $R(x) = p_n(x) - g_n(x)$ полином степена највише n. За њу важи да је $R(x_i) = 0$, $\forall i = 0, \dots, n$. Дакле, R има бар $n+1$ (различиту) нулу, а како полином степена највише n може имати највише n нула, закључујемо да је $R(x) \equiv 0$, што значи да је $g_n(x) \equiv p_n(x)$.

Питање је такође и да ли интерполациони полином уопште постоји за одређене функције и одговор на то је увек потврдан, што се може показати простом конструкцијом, а три корисне форме интерполационог полинома су представљене у наставку.

Интерполациони полином $g_n(x)$ је елемент векторског простора $\mathbb{R}_n[x]$ једнодимензионих полинома степена највише n и, као такав, може се представити у облику $g_n(x) = c_0 p_0(x) + c_1 p_1(x) + \dots + c_n p_n(x)$, где је $\{p_0(x), p_1(x), \dots, p_n(x)\}$ једна база простора $R_n[x]$. Форме интерполационог полинома које су представљене разликују се управо у бази која је одабрана за репрезентацију.

1.2.1 Алгебарски полиноми

Узмимо стандардну базу $\{1, x, \dots, x^n\}$. Интерполациони полином је облика

$$g_n(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n.$$

Проблем који нас занима је како одредити коефицијенте a_i овог развоја. Како важи $g_n(x_i) = y_i$, $i = 0, \dots, n$, потребно је да решимо следећи квадратни систем $n+1$ једначина:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix},$$

где прву матрицу називамо Вандермондовом матрицом и њена детерминанта је једнака $\prod_{0\leq i<j\leq n}(x_i - x_j) \neq 0$. Будући да је детерминанта различита од нуле, систем је сагласан и има јединствено решење. Приметимо да ово доказује уједно и јединственост, али и постојање интерполационог полинома.

Велики проблем код овог начина одређивања интерполационог полинома је нестабилност решавања задатог система једначина, узрокована веома лошом условљеношћу Вандермондове матрице. За условљеност ове матрице, дефинисану са $\operatorname{cond}(V) = \|V\|\|V^{-1}\|$, где је $\|V\| = \sup_{\|x\|_2=1} \|Vx\|_2$, показано је да експоненцијално расте са бројем чворова. Корисне доње границе које ово показују су дате у [21].

Интуитивно, проблем је у томе што, повећавањем броја чворова, тежимо да смањимо удаљеност између њих, односно разлике $x_i - x_j$ су мале, па је горе наведена детерминанта блиска нули. Ово чини нумеричко решавање система нестабилним и није лако доћи до правих коефицијената репрезентације. Због тога, ова форма интерполационог проблема се врло ретко користи.

Са друге стране, предност ове форме је њена једноставност и могућност да се изузетно брзо евалуира вредност полинома у некој тачки. Велика брзина се постиже зато што ова форма допушта примену Хорнеровог правила за израчунавање полинома, које подразумева представљање полинома у форми:

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = ((a_n x + a_{n-1})x + \cdots + a_1)x + a_0,$$

чиме се значајно смањује број рачунских операција потребних за евалуацију полинома.

Ова репрезентација интерполационог полинома је најбржа за евалуацију од свих које ћемо навести.

1.2.2 Лагранжови полиноми

Најчешће коришћена база полинома за интерполацију је база Лагранжових полинома $\{l_0(x), \ldots, l_n(x)\}$, дефинисаних са:

$$l_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j},\ i = 0, \ldots, n. \tag{1.1}$$

Ови полиноми су конструисани тако да важи

$$l_i(x_j) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases}, \tag{1.2}$$

односно $l_i(x_j) = \delta_{ij}$, где је δ_{ij} Кронекеров делта симбол. Ово својство је врло важно и чини Лагранжове полиноме посебно погодним за интерполацију, али и теоријска разматрања.

Желимо да представимо интерполациони полином у облику

$$g_n(x) = b_0 l_0(x) + b_1 l_1(x) + \cdots + b_n l_n(x)$$

и неопходно је да одредимо коефицијенте b_i. Користећи својство (1.2) и чињеницу да је $g_n(x_i) = y_i$, имамо

$$\forall i,\ y_i = g_n(x_i) = b_i,$$

односно, коефицијенти репрезентације су управо вредности функције коју интерполирамо у чворовима интерполације.

Одатле добијамо познати облик интерполационог полинома:

$$g_n(x) = \sum_{i=0}^{n} y_i l_i(x) = \sum_{i=0}^{n} \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} y_i.$$

Израчунавање на основу овог израза је неефикасно, јер захтева $O(n^2)$ операција, и нумерички је нестабилан за велики број чворова због тога што разлике $x_i - x_j$ могу постати блиске нули. Срећом, једноставним модификацијама могуће га је довести у нумерички стабилан облик који захтева само $O(n)$ операција.

Ако погледамо изразе за Лагранжове полиноме (1.1), видимо да је бројилац $\prod_{j \neq i}(x - x_j)$ једнак $\frac{l(x)}{(x - x_i)}$, где је $l(x) = \prod\limits_{j=0}^{n}(x - x_j)$.

С друге стране, видимо да је именилац у (1.1) $\prod_{j \neq i}(x_i - x_j)$ константан у односу на x, па те вредности можемо израчунати пре евалуације. Стога дефинишемо барицентричне тежине $\omega_i = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{1}{x_i - x_j} = \frac{1}{l'(x_i)}$, па полином $l_i(x)$ постаје

$$l_i(x) = l(x) \frac{w_i}{x - x_i}.$$

Одатле сумацијом добијамо први облик барицентричне форме Лагранжовог интерполационог полинома

$$g_n(x) = l(x) \sum_{i=0}^{n} \frac{w_i}{x - x_i} y_i. \tag{1.3}$$

Сада је за рачунање вредности Лагранжовог интерполационог полинома потребно само $O(n)$ операција, а доказано је да је овај облик интерполационог полинома нумерички стабилан и за веома велики број чворова (видети [13]).

Још један елегантан запис Лагранжовог полинома, који се чешће користи, добијамо следећим разматрањем. Приметимо да, ако интерполирамо функцију $f(x) \equiv 1$, добијамо интерполациони полином $q_n(x) = l(x) \sum_{i=0}^{n} \frac{w_i}{x-x_i}$, који такође мора бити увек 1, па се интерполациони полином $g_n(x)$ може добити и дељењем $g_n(x) = g_n(x)/q_n(x)$, што представља барицентрични форму Лагранжовог интерполационог полинома

$$g_n(x) = \frac{\sum_{i=0}^{n} \frac{w_i}{x-x_i} y_i}{\sum_{i=0}^{n} \frac{w_i}{x-x_i}}. \tag{1.4}$$

И за ову форму је показано да је нумерички стабилна, под условом да су чворови одабрани на погодан начин (видети [13]).

1.2.3 Њутнови полиноми

Последња репрезентација коју приказујемо је у Њутновој бази полинома $\{n_0(x), n_1(x), \ldots, n_n(x)\}$ дефинисаних са

$$n_i(x) = \prod_{j=0}^{i-1} (x - x_j),\ i = 1, \ldots, n,$$

и $n_0(x) = 1$. Интерполациони полином постаје

$$g_n(x) = c_0 + c_1 n_1(x) + \cdots + c_n n_n(x),$$

и неопходно је одредити коефицијенте c_i.

Приметимо да је $n_i(x_j) = 0$ за све $j < i$, па је систем једначина $g_n(x_i) = y_i,\ i = 0, \ldots, n$ потребних за одређивање коефицијената облика

$\sum\limits_{j=0}^{i} c_j n_j(x_i) = y_i,\ i = 0, \ldots, n$, односно имамо троугаони систем

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & n_1(x_1) & 0 & \ldots & 0 \\ 1 & n_1(x_2) & n_2(x_2) & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & n_1(x_n) & n_2(x_n) & \ldots & n_n(x_n) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

Очигледно је $c_0 = y_0$ и $c_1 = \frac{y_1 - y_0}{n_1(x_1)} = \frac{y_1 - y_0}{x_1 - x_0}$.

Дефинишимо Њутнове подељене разлике рекурзивно са

$$\begin{aligned} f[x_i] &= y_i,\ i = 0, \ldots, n \\ f[x_i, \ldots, x_{i+k}] &= \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i},\ k = 1, \ldots, n - i. \end{aligned}$$

Из ове дефиниције видимо да је $c_0 = f[x_0]$ и $c_1 = f[x_0, x_1]$. Индуктивно се показује да су коефицијенти Њутнове форме интерполационог полинома управо подељене разлике, тачније, важи $c_k = f[x_0, \ldots, x_k]$.

Нека је $k \in \{1, \ldots, n\}$ и нека су $p(x)$ i $q(x)$ полиноми који интерполирају функцију $f(x)$, редом, у тачкама $x_0, \ldots, x_{k-1}$ и $x_1, \ldots, x_k$. Они су (по индуктивној хипотези) облика

$$\begin{aligned} p(x) =& f[x_0] + f[x_0, x_1](x - x_0) + \ldots \\ &+ f[x_0, \ldots, x_{k-1}](x - x_0) \cdot \ldots \cdot (x - x_{k-2}) \\ q(x) =& f[x_1] + f[x_1, x_2](x - x_1) + \ldots \\ &+ f[x_1, \ldots, x_k](x - x_1) \cdot \ldots \cdot (x - x_{k-1}) \end{aligned}$$

Постматрајмо полином $r(x) = \frac{(x - x_0)q(x) - (x - x_k)p(x)}{x_k - x_0}$. Једноставно се проверава да је $r(x_i) = y_i$ за све $i = 0, \ldots, k$, тј. $r(x)$ је интерполациони полином над чворовима $x_0, \ldots, x_k$ и може се записати као

$$r(x) = p(x) + c_k(x - x_0) \cdot \ldots \cdot (x - x_{k-1}).$$

Коефицијент c_k је једнак коефицијенту уз x^k у полиному $r(x)$, што је

$$c_k = \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{x_k - x_0} = f[x_0, \ldots, x_k].$$

Одавде добијамо да је Њутнова форма интерполационог полинома

$$g_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, \ldots, x_n](x - x_0) \cdot \ldots \cdot (x - x_{n-1}).$$

Ова форма интерполационог полинома је нумерички много стабилнија од алгебарске форме, док се у нашим експериментима показала мање стабилном од барицентричне Лагранжове формуле.

Битно је приметити да и ова форма, као и алгебарска, допушта коришћење Хорнерове методе за евалуацију полинома. Наиме, захваљујући угњежђености Њутнових полинома, Њутнов облик интерполационог полинома можемо записати на следећи начин:

$$\begin{aligned} g_n(x) &= c_0 + c_1(x - x_0) + \cdots + c_n(x - x_0) \cdot \ldots \cdot (x - x_{n-1}) \\ &= ((c_n(x - x_{n-1}) + c_{n-1})(x - x_{n-2}) + \cdots + c_1)(x - x_0) + c_0, \end{aligned}$$

где су c_k подељене разлике $f[x_0, \ldots, x_k]$. Овиме значајно умањујемо број рачунских операција неопходних за израчунавање вредности полинома, будући да не рачунамо производе $\prod_{j=0}^{i-1}(x - x_i)$ у целости, већ се сваки елемент $x - x_i$ појављује само једном у формули. Поменимо само да је алгебарска форма ипак за нијансу бржа за евалуацију од Њутнове, док је Лагранжова спорија од обе.

1.2.4 Грешка иннтерполације

Интерполациони полином $g_n(x)$ се са функцијом $f(x)$ коју интерполира поклапа у чворовима интерполације, али то нам ништа не говори о томе колико грешимо у апроксимацији функције ван чворова. У теореми 1.2.2 (видети [22]) је дата грешка интерполације у некој тачки x, односно $f(x) - g_n(x)$. Пре тога, као подсетник наводимо Ролову теорему, која има кључно место у извођењу.

Теорема 1.2.1 (Ролова тероема)**.** *Нека је $f : [a, b] \to \mathbb{R}$ непрекидна функција, диференцијабилна на (a, b). Ако важи $f(a) = f(b)$, онда постоји бар једна тачка $c \in (a, b)$, за коју је $f'(c) = 0$.* ■

Теорема 1.2.2 (Грешка интерполације)**.** *Нека је $f(x)$, $(n+1)$ пута диференцијабилна функција и нека је $g_n(x)$ полином степена највише n који интерполира дату функцију у тачкама $x_0, \ldots, x_n$. Тада, за сваку тачку*

x', постоји тачка ξ која припада минималном интервалу који садржи све тачке $x_0, \ldots, x_n, x'$, таква да је

$$f(x') - g_n(x') = \frac{f^{(n+1)}(\xi)}{(n+1)!} l(x'),$$

где је

$$l(x) = \prod_{i=0}^{n} (x - x_i).$$

Доказ. У случају да је x једна од чворних тачака, грешка је 0 и тврђење важи тривијално. Зато претпоставимо да $x \neq x_i,\ i = 0, \ldots, n$.

Дефинишимо функцију

$$F(x) = f(x) - g_n(x) - Kl(x),$$

где је K константа одређена тако да је $F(x') = 0$, односно,

$$K = \frac{f(x') - g_n(x')}{l(x')}.$$

Како је, за све $i = 0, \ldots, n$, $l(x_i) = 0$, а $g_n(x_i) = f(x_i)$, биће и $F(x_i) = 0$. Дакле, функција $F(x)$ има бар $(n+2)$ нуле $x_0, \ldots, x_n, x'$.

Одатле, узастопном применом Ролове теореме, закључујемо да $F'(x)$ има бар $(n+1)$ нула, $F''(x)$ има бар n нула, ..., а $F^{(n+1)}$ има бар једну нулу ξ на интервалу одређеном тачкама $x_0, \ldots, x_n, x'$.

Полином $g_n(x)$ је степена највише n, па је $g_n^{(n+1)}(x) = 0$ за све x, док је $l^{(n+1)}(x) = (n+1)!$ за све x. Стога је

$$0 = F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - K(n+1)!,$$

одакле је

$$K = \frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

Како смо на почетку рекли да је $K = \frac{f(x') - g_n(x')}{l(x')}$, изједначавањем два израза за K добијамо грешку инетрполације

$$f(x') - g_n(x') = \frac{f^{(n+1)}(\xi)}{(n+1)!} l(x').$$

$\blacksquare$

1.3 Гаусова квадратурна формула

Интеграле је често немогуће изразити у аналитичком облику, па их је тешко израчунавати. Стога су развијене нумеричке методе интеграције које углавном апроксимирају интеграл $\int_a^b f(x)\omega(x)dx$ формулом облика $\sum_{i=1}^n w_i f(x_i)$, где је $\omega \in C(a,b)$, $\omega(x) > 0$ тежинска функција, тачке x_i, $i = 1, \ldots, n$ чворови, а w_i коефицијенти квадратурне формуле. Ми ћемо се бавити само Гаусовом квадратурном формулом, у којој се одређују и чворови и коефицијенти квадратурне формуле тако да формула буде тачна уколико је подинтегрална функција $f(x)$ полином степена не већег од $2n-1$. С тим циљем у виду, прво морамо да се осврнемо на ортогоналне полиноме, који играју пресудну улогу у Гаусовој квадратурној формули.

1.3.1 Ортогонални полиноми

Нека је $\omega(x)$ непрекидна тежинска функција на интервалу $[a,b]$ (који може бити и неограничен), за коју важи: $\omega(x) \geq 0$ за све $x \in [a,b]$, $\int_a^b \omega(x)\,dx > 0$, постоје моменти $\int_a^b x^k\omega(x)\,dx < \infty$, $k = 0, 1, \ldots$, као и да је $\omega(x) = 0$ само на скупу мере нула. За наше сврхе је битно поменути да $\omega(x)$ може бити густина неке апсолутно непрекидне расподеле.

На простору $\mathcal{L}^2_\omega(a,b) = \{f : [a,b] \to \mathbb{R} \mid \int_a^b f^2(x)\omega(x)\,dx < \infty\}$ је овом функцијом дефинисан скаларни производ

$$(f,g) = \int_a^b f(x)g(x)\omega(x)\,dx.$$

Кажемо да су две функције $f(x)$ и $g(x)$ ортогоналне на интервалу $[a,b]$ у односу на тежинску функцију $\omega(x)$ ако је испуњено

$$(f,g) = \int_a^b f(x)g(x)\omega(x)\,dx = 0.$$

Применом Грам-Шмитовог поступка, ортогонализацијом базе полинома $1, x, x^2, \ldots$, може се конструисати систем ортогоналних полинома $Q_n(x)$, $n = 0, 1, \ldots$ у односу на овај скаларни производ, таквих да је $(Q_i, Q_j) = 0$, $i \neq j$ и $\deg Q_n = n$.

Битно својство полинома $Q_n(x)$ из система ортогоналних полинома је дато следећом теоремом (видети [22]).

Теорема 1.3.1. *Корени* $x_i, i = 1, \ldots, n$ *полинома* Q_n *су реални, једноструки и сви припадају интервалу* (a, b).

Доказ. Издвојимо све корене полинома $Q_n(x)$ из интервала (a, b) који су непарне вишеструкости и означимо их са $a < \xi_1 < \cdots < \xi_j < b$, $j \leq n$.

Тада полином $Q_n(x)P(x)$, где је $P(x) = \prod_{i=1}^{j}(x - x_i)$, има све нуле парне вишеструкости, па не мења знак на (a, b). Одатле је, пошто је $\omega(x) > 0$ скоро свуда на $[a, b]$,

$$(Q_n, P) = \int_a^b Q_n(x)P(x)\omega(x)\,dx \neq 0.$$

Стога, полином $P(x)$ мора бити степена n, јер би у супротном, због ортогоналности полинома $Q_n(x)$ са свим полиномима нижег степена (који су линеарне комбинације полинома $Q_k(x),\ k < n$), морало бити $(Q_n, P) = 0$.

Дакле, имамо n различитих корена полинома $Q_n(x)$ који су непарне вишеструкости и припадају (a, b), па, пошто је $\deg Q_n = n$, морају бити и једноструки. ∎

Ако желимо да полиноми $Q_k(x)$ из ортогоналног система полинома буду монични, односно да им је коефицијент уз највиши степен једнак 1, показује се да је могуће конструисати ортогоналне полиноме такве да важи рекурентна веза

$$Q_{k+1}(x) = (x - \alpha_k)Q_k(x) - \beta_k Q_{k-1}(x), \quad k = 0, 1, \ldots \tag{1.5}$$

где су $Q_{-1}(x) \equiv 0$ и $Q_0(x) \equiv 1$, а коефицијенти дати са

$$\alpha_k = \frac{(xQ_k, Q_k)}{(Q_k, Q_k)}, \quad \beta_k = \frac{(Q_k, Q_k)}{(Q_{k-1}, Q_{k-1})}.$$

Поред оваквих рекурентних веза, за ортогоналне полиноме важе и многи други идентитети. Један посебно значајан, који е нама од користи је Кристофел–Дарбуов идентитет, који гласи:

$$\sum_{k=0}^{n} \frac{Q_k(x)Q_k(y)}{h_k} = \frac{k_n}{k_{n+1}h_n} \frac{Q_{n+1}(x)Q_n(y) - Q_n(x)Q_{n+1}(y)}{x - y}, \tag{1.6}$$

где су $x, y \in (a, b)$, $h_k = (Q_k, Q_k)$ квадрат норме полинома $Q_k(x)$, а k_n коефицијент уз елемент највишег степена у полиному $Q_n(x)$ (видети [5]).

1.3.2 Квадратурна формула

Вратимо се сада на горепоменуту Гаусову квадратурну формулу и покажимо да је, за погодно одабране чворове, она тачна за све полиноме степена не вишег од $2n-1$, где је n број чворова квадратурне формуле. У теореми 1.3.2 дат је конструктиван доказ, који има додатну предност да показује да је Гаусова квадратурна формула једнака интеграцији Лагранжовог интерполационог полинома у чворним тачкама (видети [27]).

Теорема 1.3.2. *Нека је $\omega(x)$ тежинска функција дефинисана на $[a,b]$ и нека је $Q_0(x), Q_1(x), \ldots$ њој одговарајући систем ортогоналних полинома. Нека су*

$$a < x_1 < x_2 < \cdots < x_n < b$$

нуле n-тог полинома у систему, $Q_n(x)$. Тада је могуће наћи константе $w_1, \ldots, w_n$ такве да је квадратурна формула

$$\int_a^b f(x)\omega(x)\,dx = \sum_{i=1}^{n} w_i f(x_i)$$

тачна кад год је $f(x)$ полином степена не вишег од $2n-1$.

Доказ. Нека је $f(x)$ полином степена не већег од $2n-1$. Нека је $L(x)$ Лагранжов интерполациони полином за $f(x)$ у чворовима $x_i,\ i = 1, \ldots, n$. $L(x)$ је полином степена $n-1$. Тада је $L(x) = \sum_{i=1}^{n} f(x_i) l_i(x)$, где су $l_i(x)$ Лагранжови базни полиноми.

Интеграцијом интерполационог полинома добија се следећи облик квадратурне формуле:

$$\int_a^b f(x)\omega(x)\,dx \approx \int_a^b L(x)\omega(x)\,dx = \sum_{i=1}^{n} f(x_i) w_i,$$

где су $w_i = \int_a^b l_i(x)\omega(x)\,dx$.

Како је $L(x) \equiv g(x)$ за полиноме $g(x)$ степена не више од $n-1$, и квадратурна формула ће бити тачна за полиноме до степена $n-1$.

Нека је $\deg f \geq n$. Полином $f(x) - L(x)$ има корене у чворним тачкама x_i, па се може представити као $f(x) - L(x) = Q_n(x) r(x)$, за неки полином $r(x)$ степена не већег од $n-1$. Будући да је квадратурна формула тачна за полиноме степена не више од $n-1$, а полином $Q_n(x)$ је ортогоналан на све полиноме мањег степена од n, имамо низ једнакости

$$\begin{aligned}\int_a^b f(x)\omega(x)\,dx &= \int_a^b L(x)\omega(x)\,dx + \int_a^b Q(x)r(x)\omega(x)\,dx \\ &= \int_a^b L(x)\omega(x)\,dx = \sum_{i=1}^n f(x_i)\int_a^b l_i(x)\omega(x)\,dx,\end{aligned}$$

одакле су коефицијенти квадратурне формуле $w_i = \int_a^b l_i(x)\omega(x)\,dx$. ■

За коефицијенте Гаусове квадратурне формуле може се доказати да су позитивни, а општи облик коефицијената је

$$w_k = \frac{k_n h_{n-1}}{k_{n-1}Q_n'(x_k)Q_{n-1}(x_k)}, \tag{1.7}$$

где је $h_k = (Q_k, Q_k)$ квадрат норме полинома $Q_k(x)$, а k_n коефицијент уз елемент највишег степена у полиному $Q_n(x)$ (видети [5]).

1.4 Тестови сагласности

Како нам је задатак да генеришемо узорак из неке расподеле, неопходно је да имамо начин да тестирамо резултате нашег узорковања. То ћемо радити користећи тестове сагласности са расподелом. Тестови које ћемо користити су засновани на рачунању неког вида растојања између емпиријске функције расподеле узорка и функције расподеле из које претпостављамо да је узорак проистекао.

За тестирање наше методе користимо три позната теста сагласности: Колмогоров–Смирновљев тест, Андерсон–Дарлингов тест и Крамер–фон Мизесов тест сагласности.

Установимо прво општи задатак тестирања сагласности, којим се баве сви поменути тестови, као и ознаке које користимо.

Нека је $x_1, \ldots, x_n$ узорак за који претпостављамо да представља скуп опажања неке случајне величине X са функцијом расподеле $F(x)$. Да бисмо тестирали да ли је претпоставка тачна, рачунамо растојање функције расподеле $F(x)$ и емпиријске функције расподеле узорка $F_n(x) = \frac{1}{n}\sum_{i=1}^n I\{x_i \le x\}$, где је $I\{\}$ индикаторска функција, па на основу тога доносимо закључке о истинитости хипотезе да је узорак потекао из расподеле $F(x)$.

Растојање које користимо за мерење колико емпиријска функција расподеле одступа од претпостављене расподеле зависи од теста и представља основну разлику међу наведеним тестовима.

1.4.1 Колмогоров–Смирновљев тест

У овом тесту удаљеност емпиријске и претпостављене функције расподеле меримо Колмогоров-Смирновљевом статистиком

$$D = \sup_{x\in\mathbb{R}} \left| F(x) - F_n(x) \right| .$$

Због непребројивости скупа реалних бројева, овај израз је тешко израчунати у изворном облику, зато ћемо извести једноставнији облик ове статистике.

Прво треба приметити да је емпиријска функција расподеле $F_n(x)$ степенаста, као и да се скокови дешавају само у тачкама x_i, где је вредност $F_n(x_i) = i/n, i = 1, \dots, n$. Такође, постоји лева гранична вредност $\lim\limits_{x\to x_i-} F_n(x) = \frac{i-1}{n}$.

С друге стране, функција расподеле $F(x)$ је растућа, а на интервалу $[x_{i-1}, x_i)$ емпиријска функција расподеле $F_n(x)$ је константна и једнака $\frac{i-1}{n}$. Отуд је, због непрекидности $F(x)$, $\sup_{x\in[x_{i-1},x_i)} \left| F(x) - F_n(x) \right|$ или једнак $\left| F(x_{i-1}) - \frac{i-1}{n} \right|$ или $\left| F(x_i) - \frac{i-1}{n} \right|$.

Одатле, статистику D можемо рачунати као

$$D = \max\{D^+, D^-\},$$

где су $D^+ = \max\limits_{1\le i\le n} \left| F(x_i) - \frac{i}{n} \right|$ и $D^- = \max\limits_{1\le i\le n} \left| F(x_i) - \frac{i-1}{n} \right|$. Овај облик је значајно лакши за рачунање.

Битно својство овог теста сагласности је независност расподеле статистике D од расподеле $F(x)$, описана следећом теоремом.

Теорема 1.4.1 (Независност од расподеле)**.** *Ако је $F(x)$ непрекидна и строго растућа функција, тада расподела статистике*

$$D = \sup_{x\in\mathbb{R}} \left| F(x) - F_n(x) \right|$$

не зависи од $F(x)$.

Доказ. Нека је $F^{-1}(p)$ инверз функције расподеле $F(x)$. За расподелу статистике D, након увођења смене $p = F(x)$, важи

$$\begin{aligned} P\{D \leq t\} &= P\left\{\sup_{x\in\mathbb{R}}\left|F(x) - F_n(x)\right| \leq t\right\} \\ &= P\left\{\sup_{p\in[0,1]}\left|F(F^{-1}(p)) - F_n(F^{-1}(p))\right| \leq t\right\} \\ &= P\left\{\sup_{p\in[0,1]}\left|p - F_n(F^{-1}(p))\right| \leq t\right\}. \end{aligned}$$

Даље, за $F_n(F^{-1}(p))$ имамо

$$F_n(F^{-1}(p)) = \frac{1}{n}\sum_{i=1}^{n} I\{X_i \leq F^{-1}(p)\} = \frac{1}{n}\sum_{i=1}^{n} I\{F(X_i) \leq p\},$$

где је $X_1, \ldots, X_n$ узорак са расподелом $F(x)$, па је $U = F(X_i)$ униформно расподељена случајна величина на $[0, 1]$. Одатле закључујемо да је

$$P\{D \leq t\} = P\left\{\sup_{p\in[0,1]}\left|p - \frac{1}{n}\sum_{i=1}^{n} I\{U \leq p\}\right| \leq t\right\},$$

што очигледно не зависи од $F(x)$. ■

Показано је да расподела тест статисике D асимтотски тежи Колмогоровљевој расподели, када обим узорка n тежи бесконачности (видети [15]). Тачније, важи

$$\lim_{n\to\infty} P\left\{\sqrt{n}D \leq x\right\} = 1 - 2\sum_{k=1}^{\infty}(-1)^{k-1}e^{-2k^2x^2}.$$

1.4.2 Крамер–фон Мизесов тест

Крамер–фон Мизесов тест се ослања на средњеквадратно одступање емпиријске и претпостављене функције расподеле

$$W^2 = \int_{-\infty}^{\infty}\left(F(x) - F_n(x)\right)^2\, dF(x).$$

Извешћемо згоднији облик ове статистике за израчунавање. Претпоставимо да је узорак $x_1, \dots, x_n$ сортиран у растућем поретку и нека је $z_i = F(x_i)$. Сменом $u = F(x)$ добијамо

$$\begin{aligned} W^2 &= \int_0^{z_1} u^2\,du + \sum_{k=1}^{n-1} \int_{z_k}^{z_{k+1}} \left(u - \frac{k}{n}\right)^2 du + \int_{z_n}^{1} (u-1)^2\,du \\ &= \frac{z_1^3}{3} + \frac{1}{3}\sum_{k=1}^{n-1}\left[\left(z_{k+1} - \frac{k}{n}\right)^3 - \left(z_k - \frac{k}{n}\right)^3\right] - \frac{(z_n-1)^3}{3}. \end{aligned}$$

Након развијања кубова бинома и свођења израза, коришћењем идентитета $\sum_{k=1}^{n-1} k^2(z_{k+1} - z_k) = n^2 z_n - \sum_{k=1}^{n}(2k-1)z_i$ и $\sum_{k=1}^{n-1} k(z_{k+1}^2 - z_k^2) = nz_n^2 - \sum_{k=1}^{n} z_i^2$, израз се своди на

$$\begin{aligned} W^2 &= \frac{1}{3} + \frac{1}{n}\sum_{k=1}^{n}\left(z_k^2 - \frac{2k-1}{n} z_k\right) \\ &= \frac{1}{3} + \frac{1}{n}\sum_{k=1}^{n}\left(z_k - \frac{2k-1}{2n}\right)^2 + \frac{1}{n}\sum_{k=1}^{n}\left(\frac{2k-1}{2n}\right)^2, \end{aligned}$$

одакле, након примене идентитета $\sum_{k=1}^{n}(2k-1)^2 = \frac{1}{3}n(4n^2-1)$, добијамо коначни облик

$$W^2 = \frac{1}{12n^2} + \frac{1}{n}\sum_{k=1}^{n}\left(z_k - \frac{2k-1}{2n}\right)^2.$$

На сличан начин као и код Колмогоров–Смирновљевог теста се показује да је и ова статистика независна од расподеле $F(x)$.

Гранична расподела тест статистике nW^2, када обим узорка n тежи бесконачности је једнака расподели случајне величине која се може представити као бесконачна сума

$$\sum_{j=1}^{\infty} \frac{Z_j^2}{j^2\pi^2},$$

где су $Z_1, Z_2, \dots$ независне случајне величине са стандардном нормалном расподелом (видети [15]).

1.4.3 Андерсон–Дарлингов тест

Андерсон–Дарлингов тест је базиран на растојању

$$A^2 = \int_{-\infty}^{\infty} \frac{\big(F(x) - F_n(x)\big)^2}{F(x)\big(1 - F(x)\big)}\, dF(x).$$

Лако израчунљив израз за ову статистику се добија на врло сличан начин као и за Крамер–вон Мизесову тест статистику. Претпоставимо да је узорак $x_1, \dots, x_n$ сортиран у растућем поретку и означимо $z_i = F(x_i)$. Након смене $u = F(x)$ добијамо

$$A^2 = \int_0^{z_1} \frac{u}{1-u}\, du + \sum_{k=1}^{n-1} \int_{z_k}^{z_{k+1}} \frac{1}{u(1-u)} \left(u - \frac{k}{n}\right)^2 du + \int_{z_n}^{1} \frac{1-u}{u}\, du.$$

После краћег рачуна сводимо овај израз на облик

$$A^2 = -\log z_n(1-z_n) - 1 + \sum_{k=1}^{n-1} \frac{k^2}{n^2} \log \frac{z_{k+1}(1-z_k)}{z_k(1-z_{k+1})} + 2\sum_{k=1}^{n-1} \frac{k}{n} \log \frac{1-z_{k+1}}{1-z_k}.$$

Коришћењем идентитета

$$\sum_{k=1}^{n-1} k^2 \log \frac{z_{k+1}(1-z_k)}{z_k(1-z_{k+1})} = n^2 \log \frac{z_n}{1-z_n} - \sum_{k=1}^{n} (2k-1) \log \frac{z_k}{1-z_k}$$

и

$$\sum_{k=1}^{n-1} k \log \frac{1-z_{k+1}}{1-z_k} = n \log(1-z_n) - \sum_{k=1}^{n} \log(1-z_k)$$

добијамо

$$A^2 = -1 - \sum_{k=1}^{n} \frac{2k-1}{n^2} \log z_k - \sum_{k=1}^{n} \frac{2n-2k+1}{n^2} \log(1-z_k),$$

што, обртањем смера индексирања у другој суми, тј. сменом $k = n-j+1$, доводи до коначног облика тест статистике

$$A^2 = -1 - \sum_{k=1}^{n} \frac{2k-1}{n^2} \big(\log z_k + \log(1-z_{n-k+1})\big).$$

И за Андерсон–Дарлингову тест статистику важи да је независна од расподеле, као и за претходне две описане тест статистике.

Слично као код Крамер–фон Мизесовог теста, гранична расподела тест статистике nA^2 кад $n \to \infty$ је једнака расподели случајне величине представљене бесконачном сумом

$$\sum_{j=1}^{\infty} \frac{Z_j^2}{j(j+1)},$$

где су $Z_1, Z_2, \dots$ независне случајне величине са стандардном нормалном расподелом (видети [15]).

Предност ове тест статистике је то што, због дељења изразом $F(x)(1-F(x))$ у интегралу, даје већу тежину грешкама при крајевима интервала, што је чини осетљивијом на одступања у реповима расподеле од Крамер–фон Мизесове статистике.

Поглавље 2

Генерисање случајних величина

Ово поглавље бави се теоријским разматрањем методе стохастичке колокације за генерисање случајних величина [11].

2.1 Генерисање једнодимензионих случајних величина

Претпоставимо да је потребно да на ефикасан начин генеришемо узорак из расподеле неке случајне величине Y са функцијом расподеле $F_Y(y)$, чији је инверз $F_Y^{-1}(p)$. Нека је X случајна величина из чије расподеле узорке можемо да генеришемо на ефикасан начин. Означимо са $F_X(x)$ функцију расподеле случајне величине X. Једноставно се доказује да Y има исту расподелу као и случајна величина $F_Y^{-1}\big(F_X(X)\big)$, јер су и $F_Y(Y)$ и $F_X(X)$ обе униформно расподељене на интервалу $[0,1]$. Стога, ако интерполирамо функцију $F_Y^{-1}\big(F_X(x)\big)$ полиномом $g_N(x) \approx F_Y^{-1}\big(F_X(x)\big)$ степена $N-1$, случајна величина $g_N(X)$ ће имати приближно исту расподелу као и случајна величина Y.

Одавде, да би се генерисао узорак $y_1, \ldots, y_n$ из расподеле Y, могу се предузети следећи кораци:

1. Конструише се интерполациони полином $g_N(x) \approx F_Y^{-1}\big(F_X(x)\big)$ одређен чворовима интерполације $x_1, \ldots, x_N$,

2. Генерише се узорак $\xi_1, \ldots, \xi_n$ из расподеле случајне величине X,

3. Израчуна се тражени узорак као $y_1 = g_N(\xi_1), \ldots, y_n = g_N(\xi_n)$.

Овиме је описан поступак генерисања случајних величина методом стохастичке колокације [11].

Проблеми које треба имати у виду се крију у прва два корака. Наиме, како је ово поступак који генерише приближну, а не тачну расподелу случајне величине Y, од пресудног значаја је да апроксимација полиномом $g_N(x)$ буде што боља. На то у великој мери утиче број чворова интерполације N, односно степен интерполационог полинома, али још значајнији је распоред чворова интерполације, који осигурава стабилну интерполацију. О томе ће бити више речи у наредном одељку.

На квалитет апроксимације утиче и одабир случајне величине X. Приметимо да, за X које је из униформне $\mathcal{U}[0,1]$ расподеле, заправо имамо апроксимацију методе инверзне трансформације. Користан избор расподеле за X је и стандардна нормална расподела, за коју се показало (видети [11]) да у већини случајева доводи до добрих резултата.

Са становишта имплементације, на резултате има утицај и одабир генератора расподеле случајне величине X. Својства генератора попут оних која се тичу случајности генерисаних бројева и периода генератора се директно преносе на коначни узорак, будући да је он просто полиномна трансформација генерисаног узорка из расподеле X. Неколико генератора униформне и нормалне расподеле је описано у делу 2.4.

2.2 Одабир чворова интерполације

Описаћемо две методе за одабир чворова интерполације. Једна има за циљ да минимизује средњеквадратну грешку интерполације, док друга користи погодна својства Чебишевљевих полинома за интерполацију.

2.2.1 Оптималан избор у средњеквадратном смислу

Један критеријум при одабиру оптималних чворова за интерполацију је минимизација средњеквадратне грешке апроксимације

$$E\left((g(X) - g_N(X))^2\right) = \int_a^b (g(x) - g_N(x))^2 f_X(x)\,dx,$$

где је $f_X(x)$ густина случајне величине X.

Видимо да је средњеквадратна грешка у овом случају просто интеграл функције $(g(x) - g_N(x))^2$ у односу на тежинску функцију $f_X(x)$.

Стога можемо применити ранија разматрања о ортогоналним полиномима и Гаусовој квадратурној формули.

Са циљем налажења оптималних чворова интерполације $x_1, \ldots, x_N$ у односу на средњеквадратну грешку, међутим, морамо се осврнути на ортогоналне пројекције функција на просторе полинома.

Пројекција функције на простор полинома

Нека је $\omega(x)$ позитивна тежинска функција на интервалу $[a, b]$ (који може бити неограничен). Дефинишемо $\mathcal{L}^2$ простор у односу на ову тежинску функцију

$$\mathcal{L}^2_\omega = \{f : [a, b] \to \mathbb{R} \mid \int_a^b f^2(x)\omega(x)\,dx < \infty\},$$

са скаларним производом

$$(f, g) = \int_a^b f(x)g(x)\omega(x)\,dx \tag{2.1}$$

и њиме индукованом нормом

$$\|f\| = \sqrt{(f, f)} = \sqrt{\int_a^b f^2(x)\omega(x)\,dx}. \tag{2.2}$$

Нека је $Q_0, Q_1, \ldots, Q_N \in \mathbb{R}_N[x]$ ортогонални систем полинома у односу на тежинску функцију $\omega(x)$, где је $\mathbb{R}_N[x]$ простор полинома степена не више од N.

Дефинишемо оператор пројекције $P_N : \mathcal{L}^2_\omega \to \mathbb{R}_N[x]$, такав да је, за свако $f \in \mathcal{L}^2_\omega$,

$$P_N f(x) = \sum_{k=0}^{N} \hat{f}_k Q_k(x),$$

где су коефицијенти пројекције

$$\hat{f}_k = \frac{1}{\|Q_k\|^2}(f, Q_k), 0 \leq k \leq N.$$

Функција $P_N f(x)$ је очигледно полином степена не већег од N и назива се ортогоналном пројекцијом функције $f(x)$ на простор полинома $\mathbb{R}_N[x]$, у односу на претходно дефинисани скаларни производ (2.1).

Наредна теорема је врло битна за одабир оптималних чворова интерполације, а каже да је ортогонална пројекција најбоља апроксимација (у средњеквадратном смислу) функција полиномима степена не вишег од N (видети [12]).

Теорема 2.2.1. *За свако $f \in \mathcal{L}^2_\omega$ и свако $N \in \mathbb{N}_0$, $P_N f$ је најбоља апроксимација у $\mathcal{L}^2$ норми* (2.2)*, односно важи*

$$\|f - P_N f\| = \inf_{q \in \mathbb{R}_N[x]} \|f - q\| .$$

Доказ. Сваки полином $q \in \mathbb{R}_N[x]$ се може представити помоћу ортогоналних полинома $Q_0(x), \ldots, Q_N(x)$ као $q(x) = \sum_{k=0}^{N} c_k Q_k(x)$, а како је минимизација $\|f - q\|$ еквивалентна минимизацији $\|f - q\|^2$, најбољу апроксимацију добићемо тражењем инфимума $\|f - q\|^2$ по коефицијентима c_k. Овај израз можемо развити као

$$\begin{aligned} \|f - q\|^2 = (f - q, f - q) &= \|f\|^2 - 2(f, q) + \|q\|^2 \\ &= \|f\|^2 - 2\sum_{k=0}^{N} c_k (f, Q_k) + \sum_{k=0}^{N} c_k^2 \|Q_k\|^2 , \end{aligned}$$

због ортогоналности система полинома $Q_k,\ k = 0, \ldots, N$. Одатле, рачунањем парцијалних извода добијамо

$$\frac{\partial}{\partial c_j} \|f - q\|^2 = -2(f, Q_j) + 2c_j \left\|Q_j\right\|^2 , \quad 0 \le j \le N.$$

Постављањем парцијалних извода на 0, добијамо да се минимум достиже за вредности коефицијената $c_j = \frac{1}{\left\|Q_j\right\|^2}(f, Q_j) = \hat{f}_j$. ■

Дискретна пројекција

Иако се најбоља апроксимација добија пројекцијом $P_N f(x)$, коефицијенте $\hat{f}_k = \frac{1}{\|Q_k\|^2}(f, Q_k)$ је тешко израчунати, будући да интеграл (f, Q_k) не мора бити аналитички израчунљив. Зато прибегавамо апроксимацији тог интеграла Гаусовом квадратурном формулом и добијамо дискретну пројекцију

$$I_N f(x) = \sum_{k=0}^{N} \tilde{f}_k Q_k(x), \tag{2.3}$$

где су коефицијенти пројекције сада одређени квадратурном формулом са $N+1$ чворова

$$\tilde{f}_k = \frac{1}{\|Q_k\|^2} \sum_{i=1}^{N+1} f(x_i) Q_k(x_i) w_i,$$

где су $x_1, \dots, x_{N+1}$ нуле полинома $Q_{N+1}(x)$, а w_i коефицијенти Гаусове квадратурне формуле.

Напоменимо да, будући да апроксимирамо коефицијенте $\hat{f}$ коефицијентима $\tilde{f}$, у општем случају не важи $P_N f = I_N f$. Међутим, постоје резултати, у које нећемо дубље залазити, који показују да је грешка $P_N f - I_N f$ истог реда као и грешка пројекције $P_N f$. Стога, за повољне довољно глатке функције, понашање дискретне пројекције $I_N f$ и пројекције $P_N f$ је врло слично у пракси. Више детаља има у [12].

Дискретна пројекција и интерполациони полином

Резултат који је нама значајан и даје везу између интерполације и апроксимације пројекцијом дат је следећом теоремом (видети [12]).

Теорема 2.2.2. *Нека је $f \in \mathcal{L}^2_\omega$ и $N \in \mathbb{N}_0$. Нека је $I_N f$ дискретна пројекција дефинисана као у (2.3). Тада $I_N f$ интерполира f у чворовима Гаусове квадратурне формуле $x_1, \dots, x_{N+1}$, који представљају корене полинома $Q_{N+1}(x)$, односно, важи*

$$I_N f(x_i) = f(x_i), \quad i = 1, \dots, N+1.$$

Доказ. Развијмо прво $I_N f(x)$:

$$\begin{aligned}
I_N f(x) &= \sum_{k=0}^{N} \tilde{f}_k Q_k(x) \\
&= \sum_{k=0}^{N} \left(\frac{1}{\|Q_k\|^2} \sum_{i=1}^{N+1} f(x_i) Q_k(x_i) w_i \right) Q_k(x) \\
&= \sum_{i=1}^{N+1} \sum_{k=0}^{N} \frac{1}{\|Q_k\|^2} f(x_i) Q_k(x_i) Q_k(x) w_i \\
&= \sum_{i=1}^{N+1} f(x_i) \left(w_i \sum_{k=0}^{N} \frac{1}{\|Q_k\|^2} Q_k(x_i) Q_k(x) \right)
\end{aligned}$$

$$= \sum_{i=1}^{N+1} f(x_i) l_i(x),$$

где смо означили

$$l_i(x) = w_i \sum_{k=0}^{N} \frac{1}{\|Q_k\|^2} Q_k(x_i) Q_k(x).$$

Доказаћемо да су полиноми $l_i(x)$ Лагранжови базни полиноми, односно показаћемо да важи $l_i(x_j) = \delta_{ij},\ i, j = 1, \ldots, N+1$.

Прво, w_i су овде коефицијенти Гаусове квадратурне формуле, који су дати у (1.7) и једнаки

$$w_i = \frac{k_{N+1} \|Q_N\|^2}{k_N Q'_{N+1}(x_i) Q_N(x_i)},$$

где је k_n коефицијент уз елемент највишег степена у полиному $Q_n(x)$.

Према Кристофел–Дарбуовој формули (1.6), важиће

$$l_i(x) = w_i \frac{k_N}{k_{N+1} \|Q_N\|^2} \frac{Q_{N+1}(x_i) Q_N(x) - Q_N(x_i) Q_{N+1}(x)}{x_i - x},$$

што, скраћивањем одговарајућих вредности у w_i и коришћењем чињенице да је, по дефиницији, $Q_{N+1}(x_i) = 0$ даје

$$l_i(x) = \frac{1}{Q'_{N+1}(x_i) Q_N(x_i)} \frac{Q_N(x_i) Q_{N+1}(x)}{x - x_i}.$$

У случају да је $x = x_j$, за $j \neq i$, биће $Q_{N+1}(x_j) = 0$, а $x - x_i \neq 0$, па је испуњено $l_i(x_j) = 0,\ j \neq i$.

Како је $l_i(x)$ полином, он је и непрекидан, а коришћењем Лопиталовог правила имамо да је

$$\lim_{x \to x_i} \frac{Q_N(x_i) Q_{N+1}(x)}{x - x_i} = Q_N(x_i) Q'_{N+1}(x_i),$$

па је одатле и $l_i(x_i) = \frac{1}{Q'_{N+1}(x_i) Q_N(x_i)} Q_N(x_i) Q'_{N+1}(x_i) = 1$.

Дакле, важи $l_i(x_j) = \delta_{ij}$, па је $I_N f(x_k) = \sum_{i=1}^{N+1} f(x_i) \delta_{ik} = f(x_k)$, што је требало доказати. ∎

Значи, дискретна пројекција $I_N f(x)$ интерполира $f(x)$ у чворовима Гаусове квадратурне формуле, односно нулама полинома $Q_{N+1}(x)$. Такође, $I_N f(x)$ је линеарна комбинација полинома $Q_i(x)$, $i = 0, \dots, N$ који су степена не више од N, па је и сама полином из $\mathbb{R}_N[x]$. Зато, због јединствености интерполационог полинома, закључујемо да је $I_N f(x)$ управо једнака интерполационом полиному функције $f(x)$ у чворовима одређеним нулама полинома $Q_{N+1}(x)$.

Овај резултат нам, дакле, каже да, уколико желимо да минимизујемо средњеквадратну грешку

$$E\left((g(X) - g_N(X))^2\right) = \int_a^b (g(x) - g_N(x))^2 f_X(x)\, dx,$$

где је $f_X(x)$ густина случајне величине X, оптималне чворове интерполације $x_1, \dots, x_N$ налазимо на следећи начин. Одредимо ортогоналан систем полинома $Q_0, Q_1 \dots$ у односу на тежинску функцију $f_X(x)$ и скаларни производ $(u, v) = E\big(u(X)v(X)\big)$ и за чворове интерполације узмемо корене полинома $Q_N(x)$, тј. чворове Гаусове квадратурне формуле.

2.2.2 Нуле Чебишевљевих полинома

Претпоставимо да случајна величина Y коју генеришемо има ограничен носач $[a, b]$. Тада, због ограничености $F_Y^{-1}(p)$, проблем нумеричке нестабилности при крајевима интервала је много блажи него у неограниченом случају, па се може разматрати и директна интерполација функције $F_Y^{-1}(p)$. У контексту методе стохастичке колокације, интерполација $F_Y^{-1}(p)$ одговара интерполацији композиције $F_Y^{-1}(F_X(x))$, где је помоћна случајна величина X униформно расподељена.

У овом случају, према претходно описаном поступку тражења оптималних чворова интерполације у средњеквадратном смислу, најбоље је за чворове интерполације узети нуле Лежандрових полинома, којима се нећемо бавити, а који представљају ортогоналан систем полинома у односу на тежинску функцију $\omega(x) \equiv 1$, што одговара густини униформне расподеле $f_X(x)$.

Међутим постоји и други избор чворова интерполације који је веома распрострањен у пракси због својих повољних особина за апроксимацију, а то су корени Чебишевљевих полинома.

Чебишевљеви полиноми су заправо ортогонални полиноми у односу на тежинску функцију $\frac{1}{\sqrt{1-x^2}}$, на интервалу $[-1,1]$. Дефинишу се као

$$T_n(x) = \cos(n\cos^{-1} x),$$

где је n-ти Чебишевљев полином $T_n(x)$ степена n. Није тешко показати да је овај израз заиста полином на $[-1,1]$. Првих неколико чланова су

$$\begin{aligned}
&T_0(x) = \cos(0\cos^{-1} x) = \cos(0) = 1,\\
&T_1(x) = \cos(\cos^{-1} x) = x,\\
&T_2(x) = \cos(2\cos^{-1} x) = 2\cos^2(\cos^{-1} x) - 1 = 2x^2 - 1.
\end{aligned}$$

Користећи формулу

$$\cos(\alpha) + \cos(\beta) = 2\cos\left(\frac{\alpha+\beta}{2}\right)\cos\left(\frac{\alpha-\beta}{2}\right),$$

добијамо идентитет

$$\cos((n+1)\theta) = 2\cos(\theta)\cos(n\theta) - \cos((n-1)\theta),$$

одакле добијамо рекурентну формулу за Чебишевљеве полиноме

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),$$

из које се види да су $T_n(x)$ заиста полиноми n-тог степена на $[-1,1]$.

Лако се види да за све $x \in [-1,1]$ важи $\left|T_n(x)\right| \leq 1$, јер је $\cos x$ функција која узима вредности из интервала $[-1,1]$.

Нађимо нуле Чебишевљевих полинома, које се користе за интерполацију. Потребно је наћи $x \in [-1,1]$ такво да је

$$T_n(x) = \cos(n\cos^{-1} x) = 0.$$

Одатле мора бити

$$\cos^{-1} x = \frac{(2k-1)\pi}{2n},$$

где је $k = 1,\ldots,n$ јер је $\cos^{-1} x \in [0,\pi]$. Одатле добијамо свих n корена Чебишевљевог полинома $T_n(x)$:

$$\eta_k = \cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k = 1,\ldots,n.$$

Ове тачке су се показале као изузетно повољан избор чворних тачака интерполационог полинома, што је показано у наставку. Велика предност је и то што оне имају аналитички израз, па је тривијално израчунати чворове интерполације за било који жељени степен полинома.

Поменимо прво да се, за $n \geq 1$, и тачке $x_j = \cos\left(\frac{j\pi}{n}\right),\ j = 0, \ldots, n$, директним рачуном показује да важи

$$T_n(x_j) = (-1)^j,$$

односно $\left|T_n(x)\right|$ достиже максимум у тим тачкама.

Из рекурентне везе $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$ лако се види да је коефицијент уз највиши степен у полиному $T_n(x)$ једнак 2^{n-1}, за $n \geq 1$, па је стога $\hat{T}_n(x) = 2^{1-n}T_n(x)$ моничан, при чему су му нуле исто $\eta_k,\ k = 1, \ldots, n$. Дакле, важиће

$$\hat{T}_n(x) = \prod_{k=1}^{n}(x - \eta_k),$$

као и

$$\max_{x\in[-1,1]}\left|\hat{T}_n(x)\right| = 2^{1-n}.$$

Погледајмо сада у ком смислу су корени Чебишовљевих полинома оптимални за интерполацију. Грешка интерполације полинома $g_{n-1}(x)$ који интерполира функцију $f(x)$ у тачкама $x_1, \ldots, x_n$, дата у Теореми 1.2.2, гласи

$$f(x) - g_{n-1}(x) = \frac{f^{(n)}(\xi)}{n!}\prod_{k=1}^{n}(x - x_k),$$

за неко $\xi \in [-1, 1]$. У овом изразу, на први део $\frac{f^{(n)}(\xi)}{n!}$ не можемо утицати, јер зависи од саме функције коју интерполирамо, док на преостали производ можемо, будући да зависи од чворних тачака. Зато нас занима којим избором чворних тачака минимизујемо максимум

$$\max_{x\in[-1,1]}\left|\prod_{k=1}^{n}(x - x_k)\right|. \tag{2.4}$$

До сада смо показали да за нормиране Чебишевљеве полиноме $\hat{T}_n(x)$ важи да је овај максимум једнак 2^{1-n}. Следећом теоремом показано је да је ово и доња граница израза (2.4), односно да се минимум достиже управо ако узмемо корене Чебишевљевих полинома као чворне тачке интерполације (видети [9]).

Теорема 2.2.3. *За сваки моничан полином* $q(x)$ *степена* $n \in \mathbb{N}$ *важи*

$$\max_{x\in[-1,1]}\big|q(x)\big| \geq 2^{1-n}.$$

Доказ. Претпоставимо да постоји моничан полином $q(x)$ степена n за који је $\max_{x\in[-1,1]}\big|q(x)\big| < 2^{1-n}$.

Тада је $q(x)-\hat{T}_n(x)$ полином степена $n-1$, будући да су и $q(x)$ и $\hat{T}_n(x)$ монични. Како у тачкама $x_j = \cos\left(\frac{j\pi}{n}\right)$, $j = 0,\ldots,n$ важи $\hat{T}_n(x_j) = (-1)^j 2^{1-n}$, а $\max_{x\in[-1,1]}\big|q(x)\big| < 2^{1-n}$, то значи да је

$$\begin{aligned} q(x_j) - \hat{T}_n(x_j) &> 0, \quad \text{за непарно } j,\\ q(x_j) - \hat{T}_n(x_j) &< 0, \quad \text{за парно } j, \end{aligned}$$

односно $q(x) - \hat{T}_n(x)$ мења знак у n интервала $[x_j, x_{j-1}], j = 1,\ldots,n$. То значи да $q(x) - \hat{T}_n(x)$ има бар n нула, а како знамо да је степена $n-1$, имамо да је $q(x)-\hat{T}_n(x) \equiv 0$, односно $q(x) = \hat{T}_n(x)$, што је контрадикција. ∎

Дакле нуле Чебишевљевих полинома су најбоље чворне тачке интерполације, у смислу минимизације (2.4) и у том случају је грешка интерполације ограничена са

$$\big|f(x) - g_n(x)\big| \leq \frac{1}{2^{n-1}n!}\max_{\xi\in[-1,1]}\Big|f^{(n)}(\xi)\Big|.$$

Приметимо да смо до сада разматрали интерполацију на интервалу $[-1,1]$, јер су Чебишевљеви полиноми дефинисани на том интервалу. Лако се може прећи на произвољан ограничен интервал $[a,b]$ линеарном сменом $y = \frac{a+b}{2} + \frac{b-a}{2}x$, одакле оптимални чворови интерполације постају

$$\nu_k = \frac{a+b}{2} + \frac{b-a}{2}cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k = 1,\ldots,n.$$

Испоставља се да се овом сменом задржавају повољна својства интерполације у нулама Чебишевљевих полинома и овај одабир чворова је веома погодан за интерполацију на произвољном интервалу $[a,b]$.

Зато при генерисању случајних величина Y на ограниченом носачу често користимо управо чворове одређене нулама Чебишевљевих полинома за интерполацију $g_n(x) \approx F_Y^{-1}(x)$, јер имају врло повољна својства за интерполацију, а, захваљујући постојању њиховог аналитичког израза, једноставни су за имплементацију.

2.2.3 Голуб–Велшов алгоритам

Закључили смо већ да су оптимални чворови интерполације у средње-квадратном смислу корени ортогоналних полинома у односу на тежинску функцију одређену густином $f_X(x)$. Остаје питање како се ти чворови проналазе. Овде је описан Голуб–Велшов алгоритам [10] за налажење чворова Гаусове квадратурне формуле (који одговарају чворовима интерполације у нашем случају). Овај алгоритам је користан јер је универзалан, односно може се применити за све тежинске функције, премда се показао као нумерички недовољно стабилан за велики број чворова.

Присетимо се рекурентне везе (1.5) која важи за ортогоналне полиноме:

$$Q_{k+1}(x) = (x-\alpha_k)Q_k(x) - \beta_k Q_{k-1}(x), \quad k = 0, 1, \dots \tag{2.5}$$

где су $Q_{-1}(x) \equiv 0$ и $Q_0(x) \equiv 1$. Ако узмемо да је $e_N = (0, \dots, 0, 1)^\top$ и $P(x) = \big(Q_0(x), Q_1(x), \dots, Q_{N-1}(x)\big)^\top$, лако се проверава да важи релација

$$JP(x) = xP(x) - Q_N(x)e_N, \tag{2.6}$$

где је

$$J = \begin{pmatrix} \alpha_0 & 1 & & & & \\ \beta_1 & \alpha_1 & 1 & & \text{\Large 0} & \\ & \beta_2 & \alpha_2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & \text{\Large 0} & & \beta_{N-2} & \alpha_{N-2} & 1 \\ & & & & \beta_{N-1} & \alpha_{N-1} \end{pmatrix}.$$

Нека је λ један од N различитих корена полинома $Q_N(x)$. Тада, за $x = \lambda$, из (2.6) добијамо

$$JP(\lambda) = \lambda P(\lambda) - Q_N(\lambda)e_N = \lambda P(\lambda),$$

односно, λ је сопствена вредност матрице J. Дакле, корени полинома $Q_N(x)$, који представљају чворове Гаусове квадратурне формуле, су сопствене вредности матрице J.

Рачунањем детерминанти тродијагоналних матрица лако се изводи да су сопствене вредности матрице J једнаке сопственим вредностима матрице

$$S = \begin{pmatrix} \alpha_0 & \sqrt{\beta_1} & & & & \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & 0 & \\ & \sqrt{\beta_2} & \alpha_2 & \sqrt{\beta_3} & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \sqrt{\beta_{N-2}} & \alpha_{N-2} & \sqrt{\beta_{N-1}} \\ & & & & \sqrt{\beta_{N-1}} & \alpha_{N-1} \end{pmatrix}. \tag{2.7}$$

Остаје још само питање како одредити коефицијенте α_k и β_k рекурентне везе (2.5). Претпоставимо да постоји првих $2N$ момената случајне величине X и нека је матрица $M = (m_{ij})_{i,j=0}^{N}$ одређена елементима

$$m_{ij} = E\left(X^i X^j\right) = \int_{-\infty}^{\infty} x^i x^j f_X(x)\,dx = E\left(X^{i+j}\right), \quad i, j = 0, \dots, N.$$

У [10] је показано да, уколико је $R^\top R = M$ Шолески декомпозиција матрице M, где је $R = (r_{i,j})_{i,j=1}^{N+1}$ горње-троугаона матрица, коефицијенти α_k и β_k могу да се представе као

$$\alpha_{k-1} = \frac{r_{k,k+1}}{r_{k,k}} - \frac{r_{k-1,k}}{r_{k-1,k-1}}, \quad k = 1 \dots, N, \tag{2.8}$$

$$\beta_k = \frac{r_{k+1,k+1}}{r_{k,k}}, \quad k = 1 \dots, N-1, \tag{2.9}$$

где дефинишемо $r_{0,0} = 1$ и $r_{0,1} = 0$.

Дакле, да резимирамо, алгоритам за налажење оптималних чворова интерполације $g_N(x) \approx F_Y^{-1}\left(F_X(x)\right)$ је следећи:

1. Одреди се матрица момената $M = \left(E\left(X^{i+j}\right)\right)_{i,j=0}^{N}$,
2. Израчуна се Шолески декомпозиција $M = R^\top R$,
3. Одреде се коефицијенти α_k и β_k релацијама (2.8) и (2.9),
4. Чворови интерполације су сопствене вредности матрице S у (2.7).

2.3 Генерисање вишедимензионих случајних величина

Битна особина методе стохастичке колокације за генерисање случајних величина је то што се, поред једнодимензионих, може користити и за генерисање вишедимензионих случајних величина, односно случајних вектора. Да бисмо дошли до тога, неопходно је да се осврнемо на вишедимензиону интерполацију.

2.3.1 Вишедимензиона интерполација

Проблем вишедимензионе интерполације је знатно компликованији од једнодимензионе, јер се у више димензија јављају проблеми који не постоје у једнодимензионом случају, попут тога да у општем случају немамо јединственост интерполационог полинома за произвољан избор чворних тачака. Овде нећемо залазити у његово детаљно разматрање, већ ћемо изложити само поступак којим можемо доћи до интерполационог полинома на правоугаоној мрежи, што нам је од користи за генерисање вишедимензионих случајних величина.

Поставимо прво уопштени проблем интерполације. Ради једноставности нотације, ограничавамо се на две димензије, док се поступак даље лако уопштава.

Претпоставимо да је $\mathcal{N}$ скуп n чворова у равни:

$$\mathcal{N} = \{(u_1, v_1), \ldots, (u_n, v_n)\} \subset \mathbb{R}^2$$

и нека сваком пару (u_i, v_i) одговара вредност $c_i,\ i = 1, \ldots, n$. Циљ нам је да одредимо функцију (обично полином) $F : \mathbb{R}^2 \to \mathbb{R}$ за коју је

$$F(u_i, v_i) = c_i.$$

Нас тренутно занима случај када је $\mathcal{N}$ Декартов производ скупова $\{x_1, \ldots, x_p\}$ и $\{y_1, \ldots, y_q\}$, односно

$$\mathcal{N} = \{(x_i, y_j) \mid 1 \leq i \leq p, 1 \leq j \leq q\}.$$

Нека је G оператор над простором функција $f : \mathbb{R} \to \mathbb{R}$ дефинисан са

$$(Gf)(x) = \sum_{i=1}^{p} f(x_i) g_i(x),$$

где за функције $g_i(x)$ важи $g_i(x_k) = \delta_{ij}$, тј. $(Gf)(x)$ интерполира $f(x)$. Приметимо да, у случају да су

$$g_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{p} \frac{x - x_j}{x_i - x_j},\ i = 1, \ldots, p$$

Лагранжови базни полиноми, $(Gf)(x)$ представља управо Лагранжову форму интерполационог полинома функције $f(x)$ у чворовима $x_1, \ldots, x_p$. Овај оператор можемо проширити на функције $f : \mathbb{R}^2 \to \mathbb{R}$ са

$$(\bar{G}f)(x, y) = \sum_{i=1}^{p} f(x_i, y) g_i(x).$$

Очигледно, за свако фиксно y, $(\bar{G}f)(x, y)$ итерполира $f(x, y)$ у тачкама $(x_i, y),\ i = 1, \ldots, p$.

Исти поступак можемо да применимо и за $y_1, \ldots, y_q$. Нека је

$$(Hf)(y) = \sum_{i=1}^{q} f(y_i) h_i(y)$$

и одговарајуће продужење

$$(\bar{H}f)(x, y) = \sum_{i=1}^{q} f(x, y_i) h_i(y).$$

Дефинишемо производ ова два оператора

$$\begin{aligned}(G \otimes Hf)(x, y) &= (\bar{G}(\bar{H}f))(x, y) \\ &= \sum_{i=1}^{p} (\bar{H}f)(x_i, y) g_i(x) \\ &= \sum_{i=1}^{p} \sum_{j=1}^{q} f(x_i, y_j) h_j(y) g_i(x).\end{aligned}$$

Како је $g_i(x_j)h_k(y_l) = \delta_{ij}\delta_{kl}$, лако закључујемо да је $(G \otimes Hf)(x_i, y_j) = f(x_i, y_j)$. Дакле $(G \otimes Hf)(x, y)$ интерполира $f(x, y)$ на мрежи $\mathcal{N}$.

Користећи други израз за Лагранжове базне полиноме из првог поглавља:

$$g_i(x) = g(x) \frac{w_i}{x - x_i},$$

где је $g(x) = \prod_{j=1}^{p}(x - x_j)$, а $w_i = \prod_{\substack{j=1\\ j\neq i}}^{p} \frac{1}{x_i - x_j}$, добијамо дводимензиони Лагранжов интерполациони полином, у барицентричној форми:

$$L(x,y) = g(x)h(y)\sum_{i=1}^{p}\sum_{j=1}^{q}\frac{w_i s_i f(x_i, y_j)}{(x - x_i)(y - y_j)},$$

где су w_i и $g(x)$ претходно дефинисани, док су, аналогно, $s_i = \prod_{\substack{j=1\\ j\neq i}}^{q} \frac{1}{y_i - y_j}$ и $h(y) = \prod_{j=1}^{q}(y - y_j)$. У граничном случају, када је $y = y_j$ за неко $j = 1, \ldots, q$, горњи израз није дефинисан, али је сингуларитет отклоњив, јер постоји $\lim_{y\to y_j} L(x, y)$, па $L(x, y_j)$ одређујемо спуштајући се за једну димензију ниже, односно дефинишемо $L(x, y_j) = (\bar{G}f)(x, y_j)$, што је једнодимензиони интерполациони полином на правој $y = y_j$. Аналоган поступак примењујемо и ако је $x = x_i$ за неко $i = 1, \ldots, p$.

2.3.2 Генерисање вишедимензионих расподела

Ограничавамо се за почетак на две димензије. Желимо да генеришемо случајни вектор (U, V) са заједничком расподелом $F_{U,V}(u, v) = P\{U \le v, V \le v\}$. Претпоставимо да знамо маргиналну расподелу $F_U(u)$ случајне величине U, али и да за сваку допустиву вредност u случајне величине U знамо условну расподелу $F_{V|U=u}(v) = P\left\{V \le v \mid U = u\right\}$. Тада генерисању узорка $(u_1, v_1), \ldots, (u_n, v_n)$ из расподеле случајног вектора (U, V) можемо приступити на следећи начин:

1. Генеришемо узорак $u_1, \ldots, u_n$ из расподеле $F_U(u)$.

2. За свако $k = 1, \ldots, n$ генеришемо v_k из расподеле $F_{V|U=u_k}(v)$.

У оба корака можемо применити методу инверзне трансформације за генерисање, па самим тим и методу стохастичке колокације.

Међутим, долазимо до проблема. Претпоставимо да имамо узорак $u_1, \ldots, u_n$ из првог корака и да нам остаје да генеришемо v_k, $k = 1, \ldots, n$. Применом једнодимензионе варијанте генерисања методом стохастичке колокације, за свако појединачно k, конструисали бисмо интерполациони полином $g^{(k)}(x) \approx F^{-1}_{V|U=u_k}(F_X(x))$, где је X случајна величина са расподелом коју знамо брзо да генеришемо, и вредност v_k из расподеле $F_{V|U=u_k}(v)$ бисмо добили са $v_k = g^{(k)}(\xi_k)$, где је ξ_k из расподеле $F_X(x)$.

Овакав приступ је врло неефикасан, јер за свако u_k конструишемо посебан полином и генеришемо само један број из одговарајуће расподеле.

Са друге стране, ако посматрамо функцију две промењиве $\nu(x,y) = F^{-1}_{V|U=y}(F_X(x))$, очигледно је да се узорак $v_1,\dots,v_n$ може добити и као $v_k = \nu(\xi_k, u_k),\ k = 1,\dots,n$, где су ξ_k из расподеле $F_X(x)$. Стога, аналогно једнодимензионом случају, поступком наведеним у претходном одељку, може се конструисати дводимензиони интерполациони полином $g_{N_X,N_U}(x,y) \approx \nu(x,y)$, где се интерполација врши над правоугаоном мрежом $\{x_1,\dots,x_{N_X}\} \times \{y_1,\dots,y_{N_U}\}$, па се тражени узорак може одредити као $v_k = g_{N_X,N_U}(\xi_k,u_k),\ k = 1,\dots,n$.

Овако избегавамо n конструкција различитих једнодимензионих интерполационих полинома и замењујемо их једном конструкцијом дводимензионог интерполационог полинома и n евалуација истог, чиме остварујемо значајну уштеду у времену израчунавања, јер не конструишемо онолико интерполационих полинома колики је обим узорка.

Пратећи [11], одабир чворова интерполационе мреже вршимо на сличан начин као у једнодимензионом случају, описаном у одељку 2.2.1. Наиме, чворове $x_1,\dots,x_{N_X}$ бирамо као чворове Гаусове квадратурне формуле у односу на тежинску функцију одређену густином $f_X(x)$, док чворове $y_1,\dots,y_{N_U}$ одређујемо као чворове Гаусове квадратурне формуле у односу на тежинску функцију одређену густином $f_U(u)$. Уколико није једноставно одредити чворове у односу на $f_U(u)$, одабир се може олакшати ако приметимо да је услов $U = y$ еквивалентан услову $F^{-1}_X(F_U(U)) = F^{-1}_X(F_U(y))$, па се уместо $\nu(x,y)$ може интерполирати функција

$$\bar{\nu}(x,y) = F^{-1}_{V|F^{-1}_X(F_U(U))=F^{-1}_X(F_U(y))}(F_X(x))$$

и одабир чворова опет свести на чворове Гаусове квадратурне формуле у односу на $f_X(x)$.

Описана метода генерисања дводимензионих расподела се лако уопштава и на више димензија. Треба генерисати узорак $(y_1^{(k)},\dots,y_d^{(k)}),\ k = 1,\dots,n$ из расподеле случајног вектора $(Y_1,\dots,Y_d)$. Нека је позната маргинална расподела случајне величине Y_1 и све условне расподеле $Y_{i+1}|Y_1 = y_1,\dots,Y_i = y_i,\ i = 1,\dots,d-1$, за допустиве вредности y_j. Нека је X случајна величина коју знамо брзо да генеришемо. Генерисање траженог узорка методом стохастичке колокације се врши у d корака[1]:

[1] Ради једноставности нотације, у сваком кораку интерполациони полином означавамо са $g(\dots)$.

1. Генерише се $\xi_1^{(1)}, \ldots, \xi_1^{(n)}$ из расподеле X, па се одатле, коришћењем интерполационог полинома $g(x) \approx F_{Y_1}^{-1}(F_X(x))$, одреди прва димензија узорка $y_1^{(1)} = g(\xi_1^{(1)}), \ldots, y_1^{(n)} = g(\xi_1^{(n)})$,
2. Генерише се $\xi_2^{(1)}, \ldots, \xi_2^{(n)}$ из расподеле X, па се коришћењем дводимензионог интерполационог полинома $g(x, y) \approx F_{Y_2|Y_1=y}^{-1}(F_X(x))$ одреди $y_2^{(1)} = g(\xi_2^{(1)}, y_1^{(1)}), \ldots, y_2^{(n)} = g(\xi_2^{(n)}, y_1^{(n)})$,

   $\vdots$

d. Генерише се $\xi_d^{(1)}, \ldots, \xi_d^{(n)}$ из расподеле X, па се коришћењем интерполационог полинома $g(x, y_1, \ldots, y_{d-1}) \approx F_{Y_d|Y_1=y_1,\ldots,Y_{d-1}=y_{d-1}}^{-1}(F_X(x))$ одреди $y_d^{(1)} = g(\xi_d^{(1)}, y_1^{(1)}, \ldots, y_{d-1}^{(1)}), \ldots, y_d^{(n)} = g(\xi_d^{(n)}, y_1^{(n)}, \ldots, y_{d-1}^{(n)})$.

Видимо да је у сваком кораку k неопходно конструисати k-димензиони интерполациони полином и евалуирати га n пута, за сваки елемент жељеног узорка. Интерполација се и у овом случају врши над правоугаоном мрежом, где чворове интерполације бирамо на аналоган начин оном описаном у дводимензионом случају.

2.4 Генератори униформне и нормалне расподеле

Најчешће коришћени генератори случајних бројева су генератори униформне и нормалне расподеле, па стога вреди размотрити методе за генерисање ових расподела. У овом поглављу дат је кратак преглед основних метода за генерисање (псеудо) случајних бројева из униформне расподеле и Зигурат алгоритам за генерисање стандардне нормалне расподеле, који је један од најефикаснијих алгоритама те врсте.

2.4.1 Генератори униформне расподеле

У Монте Карло методама у најчешћој су употреби генератори псеудослучајних бројева. Они имају за циљ да генеришу детерминистички низ бројева чија статистичка својства опонашају својства случајних бројева из униформне расподеле. Предности псеудослучајних генератора се огледају у њиховој брзини, малој меморијској захтевности, као и могућности генерисања истог низа бројева неограничен број пута, ресетовањем стања генератора, што је врло важно за поновљивост резултата симулација.

Чест начин генерисања бројева из униформне $\mathcal{U}[0,1]$ расподеле је да се генерише случајан природан број k из дискретне униформне расподеле на неком скупу $\{0,\ldots,L-1\}$, па се тражени број из $\mathcal{U}[0,1]$ расподеле добија као k/L.

Већина генератора случајних бројева су засновани на коришћењу рекурентних формула и модуларне аритметике. Овде су описани неки од основних генератора, док је шири преглед могуће наћи у [20].

Један од најједноставнијих генератора је линеарни конгруентни генератор (енгл. *linear congruential generator*, LCG) код кога се случајан број x_i генерише на основу претходног броја у низу, x_{i-1}, као

$$x_i = a_0 + a_1 x_{i-1} \mod M, \tag{2.10}$$

где су $M > 0, a_1 > 0$ и a_0 цели бројеви. Овај генератор генерише бројеве из скупа $\{0,\ldots,M-1\}$, па се број из интервала $[0,1]$ добија као x_i/M.

Узимајући $a_0 = 0$ у (2.10), добија се мултипликативни конгруентни генератор (енгл. *multiplicative congruential generator*, MCG)

$$x_i = a_1 x_{i-1} \mod M. \tag{2.11}$$

Овај генератор је чешће у употреби јер има слична својства као LCG, а рачунски је ефикаснији.

Уопштење овог генератора је вишеструко рекурзивни генератор (енгл. *multiple recursive generator*, MRG)

$$x_i = a_1 x_{i-1} + a_2 x_{i-2} + \cdots + a_k x_{i-k} \mod M, \tag{2.12}$$

где је $k \geq 1$ и $a_k \neq 0$.

Битно својство генератора случајних бројева је њихов период, који означава после колико генерисаних бројева се низ понавља. Генератор (2.10) може дати највише M различитих бројева, па му је период не већи од M. У случају (2.11), не сме се добити $x_i = 0$, па је период највише $M-1$. Како постоји M^k различитих k-торки $(x_{i-1},\ldots,x_{i-k})$, период генератора (2.12) је највише M^k-1, јер није допустива k-торка сачињена само од нула.

За вредности M се обично бирају прости бројеви, међу којима су популарни Мерсенови прости бројеви, облика $2^r - 1$, за неко природно r. Чест је и избор $M = 2$. У том случају, MRG генерише случајне битове $x_i \in \{0, 1\}$, па се низ, на пример, 32 таква бита може искористити као репрезентација случајног броја $u_i \in [0, 1]$.

Један од најпопуларнијих генератора униформне расподеле је Мерсенов Твистер (енгл. *Mersenne Twister*), који се користи као подразумевани генератор случајних бројева у многим програмским језицима, међу којима је и програмски језик R, који користимо у примерима у делу 3.3. Велика предност овог генератора је што има период од чак $2^{19937} - 1$. Овај генератор је доста компликованији од претходно описаних, а детаљан опис је дат у [19].

2.4.2 Зигурат алгоритам

Један од најбржих алгоритама за генерисање стандардне нормалне расподеле је Зигурат алгоритам [18].

Овај алгоритам спада у класу алгоритама генерисања случајних бројева „одбацивањем“ (енгл. *rejection sampling).* У основи, идеја је следећа. Нека је $f(x)$ густина расподеле случајне величине X и нека је $A = \{(x, y) \mid 0 < y < f(x)\}$ скуп тачака испод графика функције $f(x)$, а B неки надскуп тог скупа ($A \subseteq B$), коначне мере. Да бисмо генерисали број ξ из расподеле X, генеришемо пар $(x, y) \in B$ из униформне расподеле над B, па ако је $(x, y) \in A$, узимамо $\xi = x$, док у супротном одбацујемо x и поново бирамо нови пар $(x, y) \in B$ док се не испуни услов $y < f(x)$, након чега прихватамо $\xi = x$.

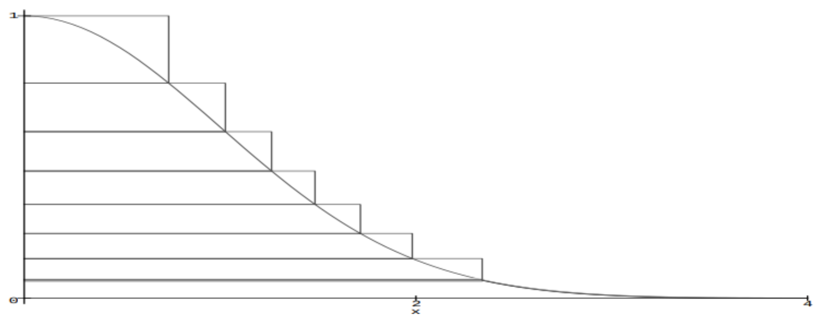
Видимо три пута ка постизању што веће брзине оваквог приступа:

- максимизовање брзине генерисања почетних парова $(x, y) \in B$ (најлакше је генерисати унифомно над правоугаоницима),
- максимизовање брзине провере услова $(x, y) \in A$ (израчунавање густине $f(x)$ може бити неефикасно),
- минимизовање разлике у површини скупова A и B, да би било што мање одбацивања (идеално би било $A = B$).

Зигурат алгоритам тежи да у што већој мери испуни сваки од захтева и показало се да постиже изузетне резултате.

Опис алгоритма

Надскуп B који користи овај алгоритам је илустрован на слици 2.1. Састоји се од уније $n-1$ правоугаоника $R_k, k = 1, \ldots, n-1$ наслаганих



Слика 2.1: Илустрација области узорковања Зигурат алгоритма.

један на други (на слици је $n = 8$, док се у пракси обично узима $n \in \{64, 128, 256\}$) и области R_n на дну која садржи реп расподеле. Скупови су одабрани тако да сви имају једнаку површину v.

Нека су $x_1 < \cdots < x_{n-1}$ апсцисе десних ивица, а $y_i = f(x_i)$ ординате доњих ивица правоугоника R_k, и означимо $x_0 = 0$. Тада је, за неко $(x, y) \in R_i$ у великом броју случајева изузетно једноставно одредити да ли је $y < f(x)$, јер је то сигурно испуњено уколико је $x < x_{i-1}$, па нам y уопште није неопходно. У супротном, мора се извршити провера која укључује израчунавање $f(x)$.

Случајан пар $(x, y) \in B$ се може униформно генерисати тако што се случајно изабере индекс $i \in \{1, \ldots, n\}$, па униформно генерише пар (x, y) из подобласти R_i. Будући да су све сем једне подобласти правоугаоници, у већини случајева генерисање случајног пара из R_i своди се на бирање два случајна броја U_1, U_2 из униформне $\mathcal{U}[0,1]$ расподеле, и узимање $(x, y) = (x_i U_1, y_i + (y_{i-1} - y_i) U_2)$ (уколико је потребно генерисати и y након провере $x < x_{i-1}$).

У случају области R_n која укључује реп расподеле, генерисање олакшава чињеница да се област састоји од правоугаоника Π коме је припојен реп расподеле $x > x_{n-1}$. Означимо краће $x_{n-1} = r$. Како је површина области R_n једнака v, случајно одабрана тачка ће припасти правоугаонику Π са вероватноћом $\frac{rf(r)}{v}$ и у том случају за одабрану тачку сигурно важи $y < f(x)$. Тада је y координата непотребна, па се исти ефекат добија и ако се генерише $x = \frac{v}{f(r)} U$, за U из униформне расподеле $\mathcal{U}[0,1]$

и провери да ли је $x < r$, што се дешава управо са вероватноћом $\frac{rf(r)}{v}$. Уколико је $x > r$, случајан број x из репа нормалне расподеле генерише се према методи датој у [17]: генерише се $x = -\frac{\log U_1}{r}$, $y = -\log U_2$ све док не буде испуњено $y + y > x^2$, након чега се узима $r + x$ за одабрани случајан број.

Поменимо да је овде описан алгоритам за позитивне расподеле, док је у случају нормалне расподеле потребно у око 50% случајева променити знак генерисаног броја. То можемо урадити тако што узмемо $x = x_i U_1$, где је U_1 из $\mathcal{U}[-1,1]$ расподеле, док провера $x < x_{i-1}$ постаје $|x| < x_{i-1}$.

Резимирајмо корак по корак Зигурат алгоритам за генерисање случајних бројева из стандардне нормалне расподеле. Означимо прво $x_n = \frac{v}{f(x_{n-1})}$. Тада алгоритам изгледа овако:

1. Одабере се случајан индекс $i \in \{1, \ldots, n\}$,
2. Генерише се униформно $U_0 \in [-1, 1]$ и узме $x = x_i U_0$.
3. Ако је $|x| < x_{i-1}$, врати се x.
4. Ако је $i = n$, генерише се $x = -\frac{\log U_1}{x_{n-1}}$, $y = -\log U_2$, где су $U_1, U_2 \in \mathcal{U}[0, 1]$, све док не буде испуњено $y + y > x^2$, након чега се врати $sgn(U_0)(x_{n-1} + x)$.
5. Генерише се униформно $U_3 \in [0, 1]$ и узме $y = y_i + (y_{i-1} - y_i)U_3$.
6. Ако је $y < f(x)$, врати се x.
7. Иначе, враћа се на корак 1.

Битна предност овог алгоритма је та што, за често коришћену вредност $n = 128$, скуп B постаје добра апроксимација скупа A тачака испод графика густине $f(x)$. Тачније, однос површина скупова A и B је чак 98.78%. Ово значи да у мање од 2% случајева одбацујемо генерисано x и бирамо поново. Такође, последица тога је и да је, у више од 95% случајева, за генерисање потребно извршити само прва три корака алгоритма, који укључују само генерисање случајног индекса, броја из униформне расподеле и једноставно поређење реалних бројева, што је главни разлог за изузетну брзину Зигурат алгоритма.

Напомена (о имплементацији алгоритма). Оригинална имплементација Зигурат алгоритма у [18] користи SH3 генератор униформне расподеле, што доводи до лошег понашања алгоритма, као што је показано у [16], где је демонстрирано и да се квалитет резултата у великој мери побољшава коришћењем KISS генератора случајних бројева. Управо то побољшање имплементира и функција `zrnorm` из пакета `RcppZiggurat` (детаљи у [6]), коју у примерима на крају рада користимо за генерисање бројева из нормалне расподеле.

Такође, у имплементацији у [18] предложено је да се само једним позивом генератора униформне расподеле одреде и случајан индекс у кораку 1 и случајан број из корака 2 тако што се генерише један 32-битни случајан број за корак 2, а последњих 7 битова тог броја се искористе за случајан индекс у првом кораку (који је из скупа $\{0, \ldots, 127\}$). Овај поступак доводи до додатног убрзања, јер је потребан један позив мање ка генератору случајних бројева.

Поглавље 3

Детаљи имплементације и резултати

У овом поглављу су прво приказани неки детаљи имплементације методе стохастичке колокације и пар приступа решавању неких практичних проблема који се јављају. Након тога дајемо нумеричке резултате које показују ефикасност методе на различитим расподелама.

3.1 Опште напомене

У свим нашим експериментима користимо нормалну или униформу расподелу за помоћну случајну величину X из методе, будући да су се показале довољно добре за већину расподела. Углавном за ограничене расподеле користимо униформно X у комбинацији са чворовима интерполације одређеним нулама Чебишевљевих полинома, док за неограничене користимо нормално расподељено X, а чворове бирамо као оптималне у средњеквадратном смислу, Голуб–Велшовим алгоритмом.

3.2 Растезање мреже

Претпоставимо да користимо нормално расподељену помоћну случајну величину X. Чест проблем на који се наилази, посебно код расподела случајних величина Y са тешким реповима, су нестабилности које се јављају када $\Phi(x) \to 0$ и када $\Phi(x) \to 1$ што узрокује да $F_Y^{-1}\big(\Phi(x)\big)$ веома брзо расте. У табели 3.1 приказан је пример који ово илуструје са 9 чворних тачака и дате су вредности функције расподеле стандардне нормалне расподеле у тим тачкама. Такође, приказане су и вредности функције расподеле нормалне расподеле са стандардном девијацијом $\sigma = 1.3714$.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x_i	-4.5127	-3.2054	-2.0768	-1.0233	0.0	1.0233	2.0768	3.2054	4.5127
$\Phi(x_i)$	0.0000	0.0007	0.0189	0.1531	0.5	0.8469	0.9811	0.9993	1.0000
$\Phi_\sigma(x_i)$	0.0005	0.0097	0.0650	0.2278	0.5	0.7722	0.9350	0.9903	0.9995

Табела 3.1: Чворови интерполације и одговарајуће вредности $\Phi(x)$ и $\Phi_\sigma(x)$, где је $\Phi_\sigma(x)$ функција расподеле нормалне $\mathcal{N}(0, \sigma^2)$ расподеле.

Вредности $\Phi(x)$ у почетним и крајњим тачкама се нагомилавају око 0 и 1, што доводи до поменуте нестабилности. Ово је још израженије за већи број чворних тачака.

Један начин да се реши овај проблем је предложен у [11] и подразумева увођење нове случајне величине $\hat{X}$ која има нормалну $\mathcal{N}(0, \sigma^2)$ расподелу, при чему вредност σ одређујемо тако да је $F_{\hat{X}}(x_N) = \Phi_\sigma(x_N)$ једнако неком унапред одређеном квантилу $p_{\max}$, односно $\Phi_\sigma(x_N) = p_{\max}$. Како за свако x важи $\Phi_\sigma(x) = \Phi(x/\sigma)$, лако израчунавамо

$$\sigma = \frac{x_N}{\Phi^{-1}(p_{\max})}.$$

Вредност $\sigma = 1.3714$ коришћену у табели 3.1 смо добили управо овом формулом за $N = 9$ и $p_{\max} = 0.9995$.

Приликом генерисања узорка $y_1, \ldots, y_n$ из расподеле Y, сада се интерполира функција $F_Y^{-1}\left(\Phi_\sigma(x)\right)$ у чворним тачкама одређеним на основу стандардне нормалне расподеле (која одговара случајној величини X) интерполационим полиномом $g_N(x)$, па се жељени узорак генерише као $y_i = g_N(\xi_i)$, где је $\xi_1, \ldots, \xi_n$ узорак из нормалне $\mathcal{N}(0, \sigma^2)$ расподеле. Ову методу називамо растезањем мреже (енгл. *grid stretching*).

Овим поступком се губи теоретска оптималност чворних тачака x_i будући да смо извршили линеарну прерасподелу оптималних чворова, али се зато чува стабилност интерполације и умањује ефекат нестабилности инверзија $F_Y^{-1}(p)$ при крајевима интервала $[0, 1]$ (видети [11]).

3.3 Нумерички резултати

Сада ћемо кроз неколико примера расподела приказати нумеричке резултате који илуструју понашање методе стохастичке колокације за генерисање случајних величина. За све експерименте користимо програмски језик R, а за имплементацију метода стохастичке колокације смо направили пакет `scmc`, који је у развојној фази и тренутно се налази на адреси `https://github.com/blaza/scmc`. Сви примери прате исту структуру, коју описујемо у наставку.

3.3.1 Структура примера

За сваку расподелу су табеларно приказани резултати већ описаних статистичких тестова: Колмогоров–Смирновљевог ("KS"), Крамер–вон Мизесовог ("CVM") и Андерсон–Дарлинговог ("AD"). Приказани су вредност тест статистике која представља неку врсту растојања између емпиријске функције расподеле генерисаног узорка и теоретске функције расподеле случајне величине коју генеришемо, као и p-вредност тог теста. Све вредности представљају просечне вредности добијене након 1024 понављања експеримента генерисања узорка обима 100,000. У истој табели дато је време генерисања једног узорка (истог обима), у милисекундама.

У првој врсти табеле су дати резултати добијени коришћењем уграђених функција у програмском језику R, уколико постоје, или, у супротном, коришћењем функција из неких од пакета за програмски језик R. У осталим врстама дати су резултати коришћењем методе стохастичке колокације, са различитим бројем чворова. Детаљи о одабиру других параметара описани су у оквиру примера.

Уз сваки пример показана су и три пара графика, за три различита одабира чворова интерполације, а сваки пар се састоји из:

1. графика који приказује интерполациони полином $g_N(x)$, функцију коју интерполира, $F_Y^{-1}\big(F_X(x)\big)$, као и чворове интерполације,

2. графика који приказује функцију расподеле $F_Y(x)$ циљне случајне величине и емпиријску функцију расподеле генерисаног узорка.

Ради уштеде простора у раду, у примерима су приказани резултати само за погодне конфигурације методе (број чворова интерполације, расподела помоћне случајне величине, погодна трансформација узорка, итд.) за одговарајуће циљне расподеле.

Значајно већи број резултата за много различитих расподела, као и различитих конфигурација за исте, могу се видети на интерактивној веб апликацији коју смо развили за експерименталне сврхе коришћењем R пакета `shiny`, а која се налази на адреси `https://blaza.shinyapps.io/shiny_scmc_distros/`.

3.3.2 Примери

Пример 1 (Логистичка расподела)**.** Нека циљна случајна величина Y има логистичку расподелу са функцијом расподеле

$$F_Y(x) = \frac{e^x}{1+e^x}, \quad x \in \mathbb{R}.$$

За примену методе стохастичке колокације, користимо помоћну случајну величину X са стандардном нормалном расподелом, а чворове интерполације бирамо као оптималне у средњеквадратном смислу.

У табели 3.2 су приказани резултати тестова, као и времена генерисања узорка обима 100,000 за 5, 7 и 9 чворова. Уграђена функција у програмском језику R која је коришћена за поређење је `rlogis`.

	KS.stat	KS.pval	CVM.stat	CVM.pval	AD.stat	AD.pval	време
R f-ja	0.0026	0.5827	0.1546	0.5887	0.9377	0.5790	4 ms
N = 5	0.0031	0.3895	0.2056	0.4084	1.2471	0.3876	1.03 ms
N = 7	0.0028	0.5208	0.1788	0.5218	1.0635	0.5200	1.17 ms
N = 9	0.0027	0.5288	0.1566	0.5256	0.9664	0.5209	1.39 ms

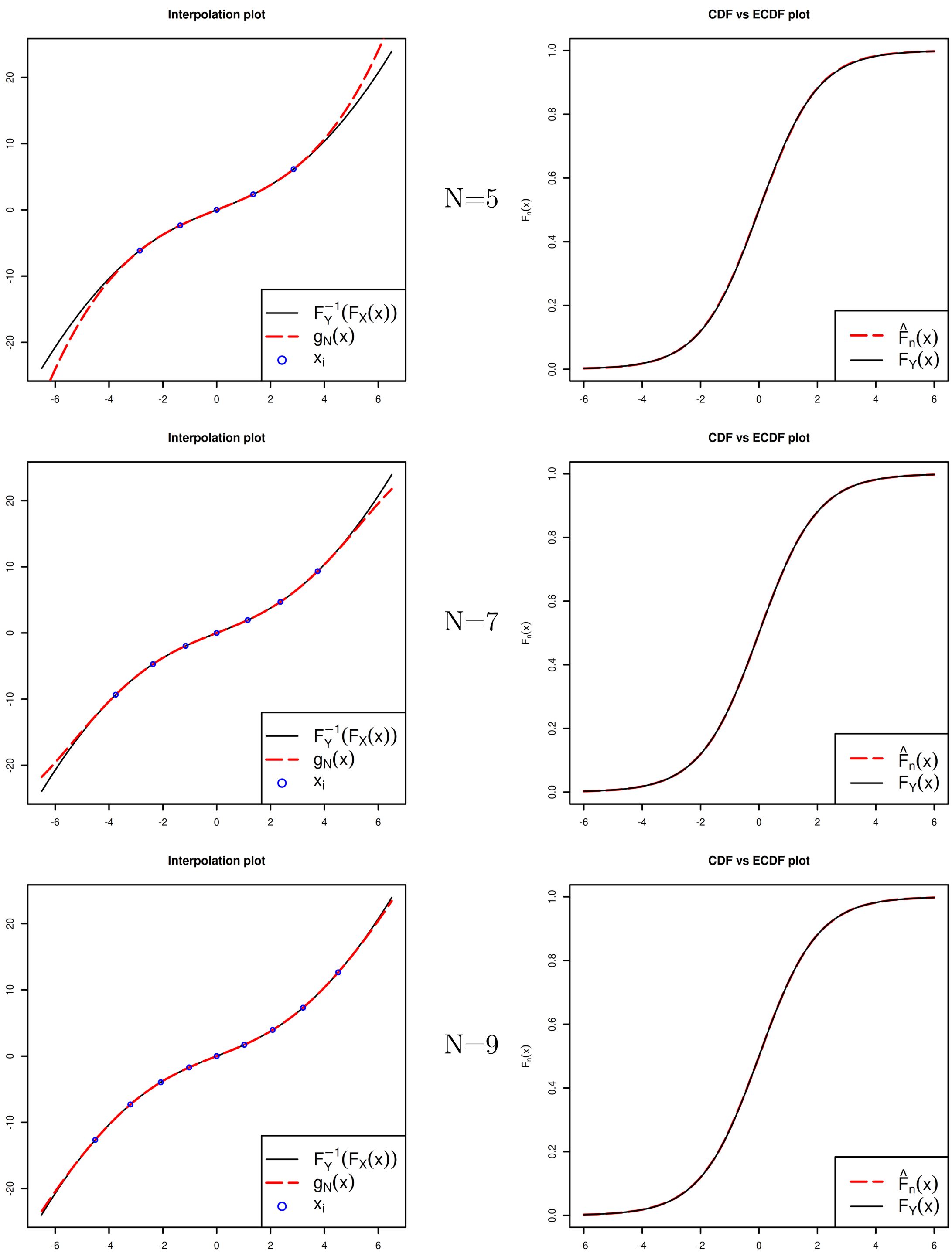
Табела 3.2: Резултати тест статистика и времена генерисања за логистичку расподелу.

Видимо да за 7 и 9 чворова имамо упоредиве резулате са методом коришћеном у програмском језику R, док је брзина скоро четвороструко већа. Треба поменути да је коришћен Зигурат алгоритам као генератор нормалне расподеле, што даје велики допринос брзини.

Резултати квалитета апроксимације су јасније видљиви на графицима на слици 3.1.

#

Слика 3.1: Графици квалитета интерполације $g_N(x) \approx F_Y^{-1}\left(F_X(x)\right)$ и апроксимације функције расподеле емпиријском функцијом расподеле, за логистичку расподелу. Црвена испрекидана линија представља апроксимацију, а пуна црна линија теоретску вредност.



Пример 2 (χ^2 расподела)**.** У овом примеру симулирамо случајну величину Y са χ^2 расподелом. Она је дефинисана као збир квадрата k стандардно нормално расподељених случајних величина и њена густина је дата изразом

$$f_Y(x) = \frac{x^{k/2-1}e^{-x/2}}{2^{k/2}\Gamma(k/2)}, \quad x \in [0, \infty),$$

где је k број степени слободе.

Као пример користимо χ^2 расподелу са 3 степена слободе.

За примену методе стохастичке колокације, користимо помоћну случајну величину X са стандардном нормалном расподелом, а чворове интерполације бирамо као оптималне у средњеквадратном смислу. Будући да је ово позитивна случајна величина, показало се да трансформација $Y' = \log Y$ доводи до боље апроксимације интерполационим полиномом. Приликом генерисања се трансформација природно инвертује у $Y = e^{Y'}$.

У табели 3.3 су приказани резултати тестова, као и времена генерисања узорка обима 100,000 за 7, 9 и 11 чворова. Уграђена функција у програмском језику R која је коришћена за поређење је `rchisq`.

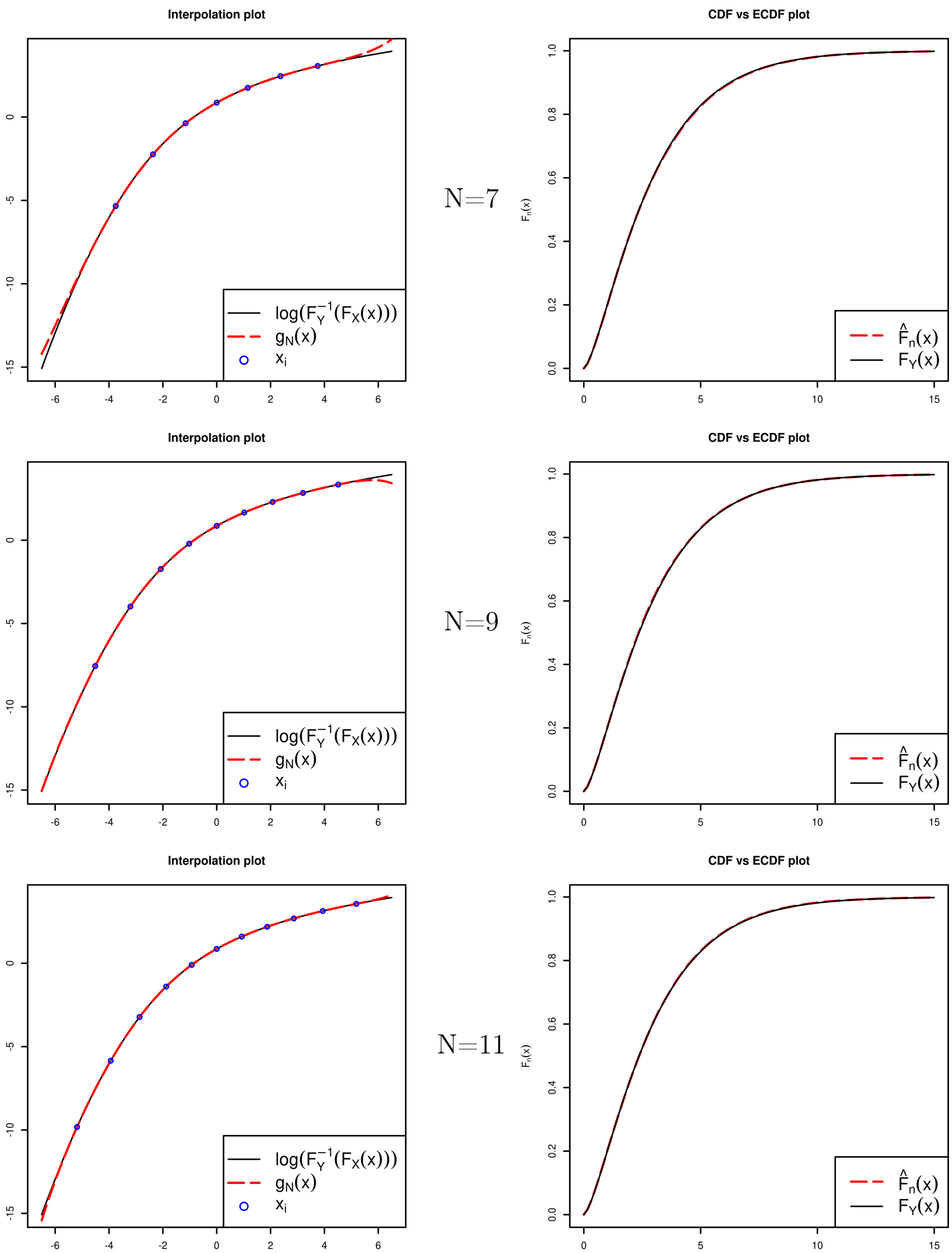
	KS.stat	KS.pval	CVM.stat	CVM.pval	AD.stat	AD.pval	време
R f-ja	0.0027	0.4865	0.1607	0.4596	0.9695	0.4533	10.17 ms
N = 7	0.0026	0.5501	0.1462	0.5528	0.8799	0.5648	2.71 ms
N = 9	0.0027	0.5208	0.1714	0.5119	1.0617	0.4944	2.79 ms
N = 11	0.0027	0.5383	0.1646	0.5406	0.9755	0.5400	3.11 ms

Табела 3.3: Резултати тест статистика и времена генерисања за χ^2_3 расподелу.

Добијамо задовољавајућу апроксимацију расподеле, уз значајно убрзање. И овде је у употреби Зигурат генератор нормалне расподеле. Резултати квалитета апроксимације су приказани и на слици 3.2.

#

Слика 3.2: Графици квалитета интерполације $g_N(x) \approx \log F_Y^{-1}\left(F_X(x)\right)$ и апроксимације функције расподеле емпиријском функцијом расподеле, за χ^2_3 расподелу. Црвена испрекидана линија представља апроксимацију, а пуна црна линија теоретску вредност.



Пример 3 (Кошијева расподела)**.** Нека је сада Y случајна величина са стандардном Кошијевом расподелом, чија је функција расподеле дата изразом

$$F_Y(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(x), \quad x \in \mathbb{R}.$$

За примену методе стохастичке колокације, користимо помоћну случајну величину X са стандардном нормалном расподелом, а чворове интерполације бирамо као оптималне у средњеквадратном смислу. Кошијева расподела је расподела са тешким реповима, па су изражене нестабилности поменуте у делу 3.2. Зато користимо поступак описан у том одељку приликом генерисања, чиме добијамо стабилну интерполацију.

У табели 3.4 су приказани резултати тестова, као и времена генерисања узорка обима 100,000 за 5, 11 и 15 чворова. Уграђена функција у програмском језику R која је коришћена за поређење је `rcauchy`. У овом случају потребан је већи број чворова него раније за добру апроксимацију, а наведени бројеви чворова су одабрани ради илустрације побољшања добијеног повећањем броја чворова интерполације.

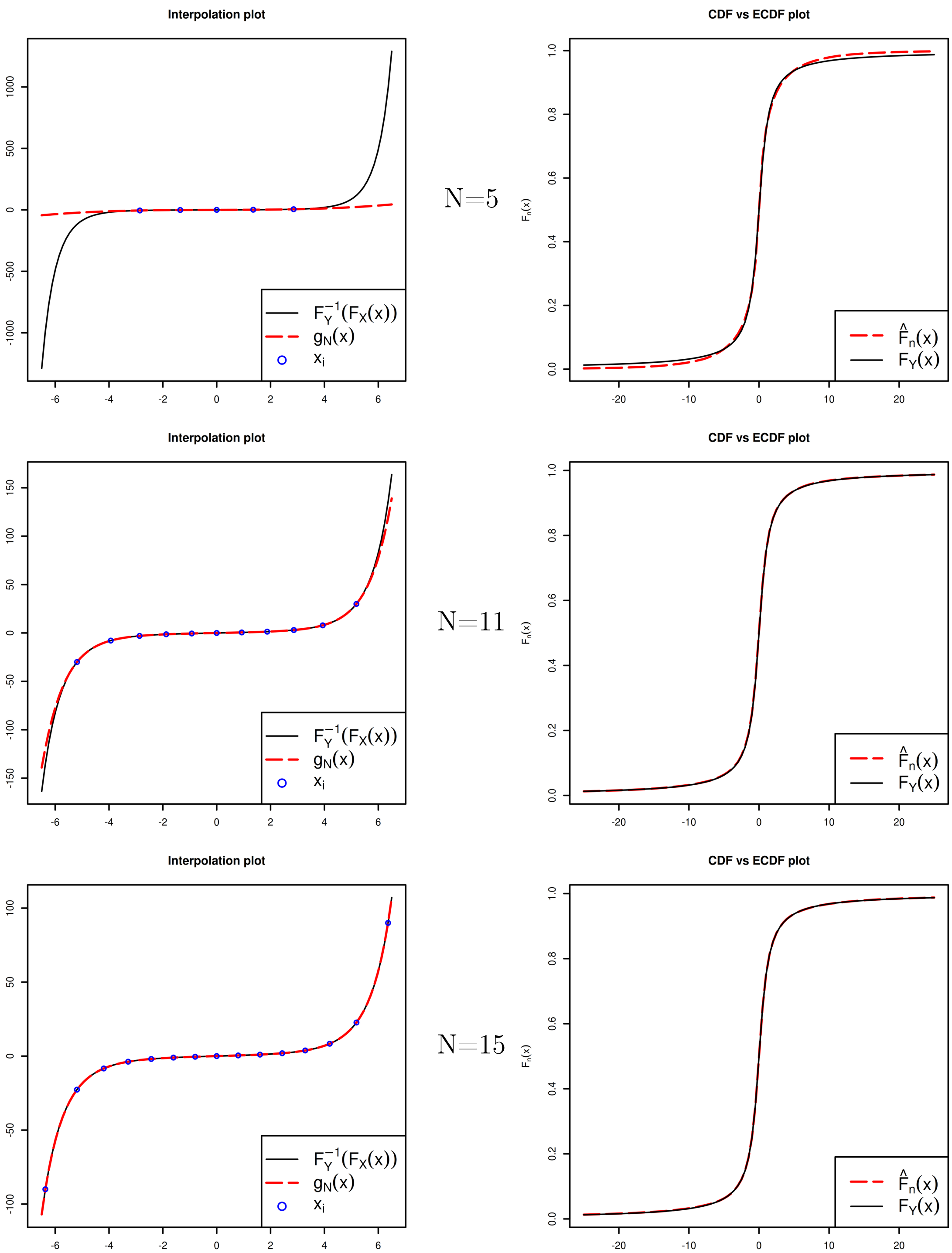
	KS.stat	KS.pval	CVM.stat	CVM.pval	AD.stat	AD.pval	време
R f-ja	0.0026	0.5359	0.1479	0.5499	0.8652	0.5584	5.23 ms
N = 5	0.0176	0.0000	9.8352	0.0000	91.4675	0.0000	1.12 ms
N = 11	0.0027	0.5069	0.1609	0.4978	1.0066	0.4721	1.66 ms
N = 15	0.0027	0.4977	0.1610	0.4854	0.9694	0.4860	2.33 ms

Табела 3.4: Резултати тест статистика и времена генерисања за Кошијеву расподелу.

Брзина генерисања је видно већа од уграђене функције у R-у. За мали број тачака имамо лошу апроксимацију, али она постаје прихватљива за N=11 и N=15. Коришћен је Зигурат генератор нормалне расподеле. Резултати квалитета апроксимације су приказани и на слици 3.3.

#

Слика 3.3: Графици квалитета интерполације $g_N(x) \approx F_Y^{-1}\left(F_X(x)\right)$ и апроксимације функције расподеле емпиријском функцијом расподеле, за Кошијеву расподелу. Црвена испрекидана линија представља апроксимацију, а пуна црна линија теоретску вредност.



Пример 4 (Вејбулова расподела)**.** Генеришемо случајну величину Y са двопараметарском Вејбуловом расподелом, чија је функција густине расподеле у општем случају дата изразом

$$f_Y(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, \quad x \in [0, \infty)$$

где је λ параметар размере, а k параметар облика. Ми за пример користимо вредности параметара $\lambda = 1$ и $k = 1/2$, јер је за $k < 1$ ово расподела са тешким реповима.

За примену методе стохастичке колокације, користимо помоћну случајну величину X са стандардном нормалном расподелом, а чворове интерполације бирамо као оптималне у средњеквадратном смислу. Вејбулова расподела са $k = 1/2$ је расподела са тешким реповима, па користимо поступак растезања мреже описан у одељку 3.2 приликом генерисања, чиме добијамо стабилну интерполацију. Такође, ова расподела је и позитивна, па прво вршимо логаритамску трансформацију, као у случају χ^2 расподеле.

У табели 3.5 су приказани резултати тестова, као и времена генерисања узорка обима 100,000 за 5, 9 и 15 чворова. Уграђена функција у програмском језику R која је коришћена за поређење је `rweibull`.

	KS.stat	KS.pval	CVM.stat	CVM.pval	AD.stat	AD.pval	време
R f-ja	0.0026	0.5476	0.1421	0.5585	0.8587	0.5614	4.69 ms
N = 5	0.0030	0.4165	0.2004	0.4185	1.1884	0.4104	2.47 ms
N = 9	0.0027	0.5116	0.1579	0.5078	0.9501	0.5103	2.74 ms
N = 15	0.0027	0.5073	0.1468	0.5137	0.9046	0.5092	3.68 ms

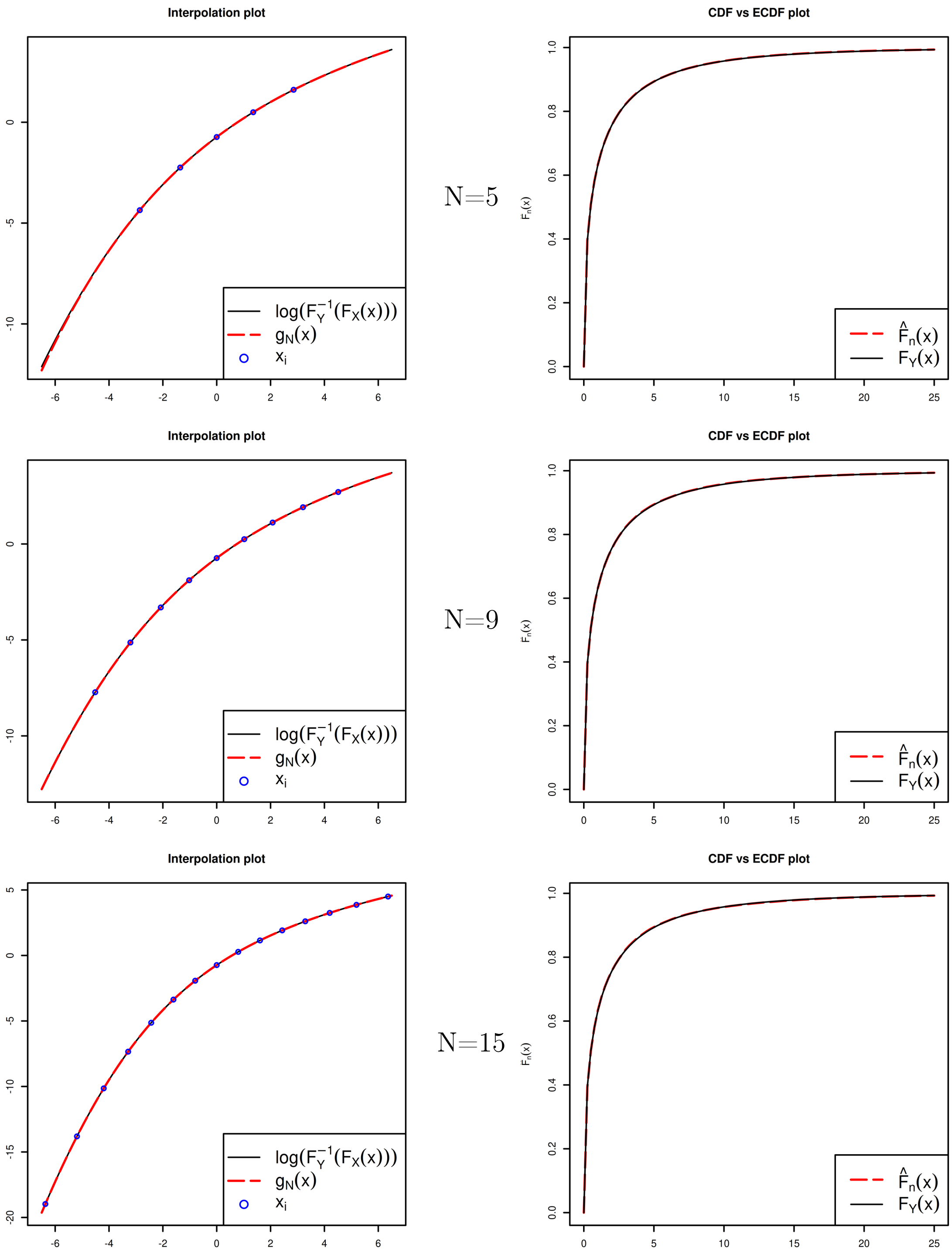
Табела 3.5: Резултати тест статистика и времена генерисања за Вејбулову расподелу.

Брзина генерисања је видно већа од уграђене функције у R-у. И у овом случају је потребан већи број чворова интерполације да би се добила добра апроксимација. Коришћен је Зигурат генератор нормалне расподеле.

Резултати квалитета апроксимације су приказани на слици 3.4.

#

Слика 3.4: Графици квалитета интерполације $g_N(x) \approx \log F_Y^{-1}\big(F_X(x)\big)$ и апроксимације функције расподеле емпиријском функцијом расподеле, за Вејбулову расподелу. Црвена испрекидана линија представља апроксимацију, а пуна црна линија теоретску вредност.



Пример 5 (Бета расподела)**.** У овом примеру посматрамо једну ограничену расподелу. Нека је Y случајна величина са бета расподелом, чија је густина:

$$f_Y(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}, \quad x \in (0,1),$$

где је $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$, док су α и β параметри који одређују облик густине. Ми као пример користимо вредности параметара $\alpha = \beta = 1/2$.

За примену методе стохастичке колокације, овај пут користимо помоћну случајну величину X са униформном $\mathcal{U}[0,1]$ расподелом, док за чворове интерполације бирамо корене Чебишевљевих полинома.

Како је ова расподела ограничена, односно $0 < Y < 1$, при крајевима интервала се не јавља нагли раст функције $F_Y^{-1}(p)$ ка бесконачности, па је у таквим случајевима често могуће извршити директну интерполацију, без увођења функције $F_X(x)$. Ова расподела је пример такве расподеле. Користимо нуле Чебишевљевих полинома за чворове интерполације, јер имају аналитички облик за израчунавање, а познато је да имају добра својства за интерполацију.

У табели 3.6 су приказани резултати тестова, као и времена генерисања узорка обима 100,000 за 13, 17 и 21 чворова. Уграђена функција у програмском језику R која је коришћена за поређење је `rbeta`.

	KS.stat	KS.pval	CVM.stat	CVM.pval	AD.stat	AD.pval	време
R f-ja	0.0026	0.5536	0.1526	0.5227	0.9586	0.5091	15.98 ms
N = 13	0.0029	0.4321	0.2265	0.4114	1.2781	0.4242	2.63 ms
N = 17	0.0028	0.4779	0.1858	0.4759	1.0796	0.4812	3.29 ms
N = 21	0.0028	0.4826	0.1806	0.4793	1.0580	0.4865	3.93 ms

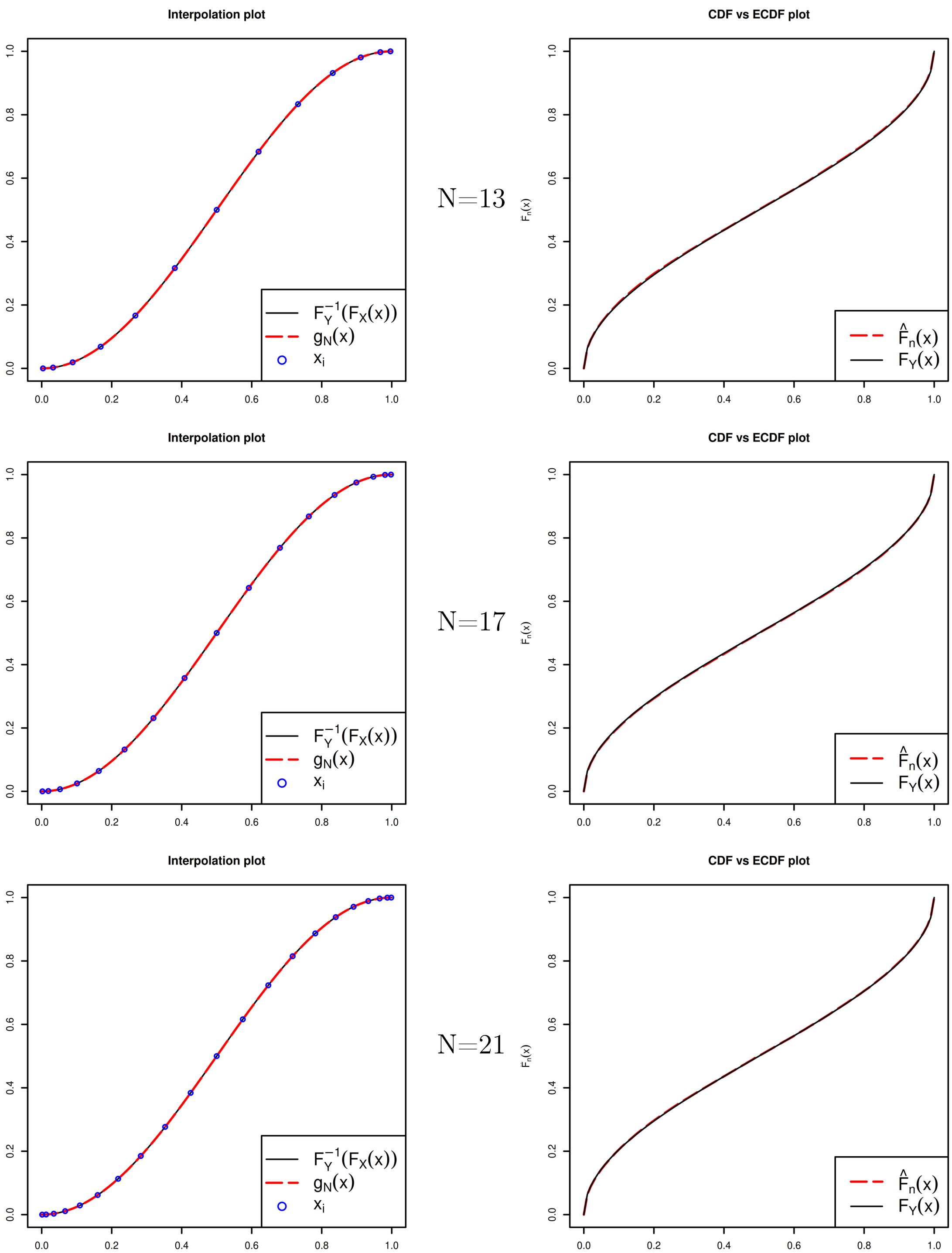
Табела 3.6: Резултати тест статистика и времена генерисања за бета расподелу.

Брзина генерисања је значајно већа од уграђене функције у R-у. Као генератор униформне расподеле користимо уграђену функцију `runif`.

Резултати квалитета апроксимације су приказани на слици 3.5.

#

Слика 3.5: Графици квалитета интерполације $g_N(x) \approx F_Y^{-1}\left(F_X(x)\right)$ и апроксимације функције расподеле емпиријском функцијом расподеле, за бета расподелу. Црвена испрекидана линија представља апроксимацију, а пуна црна линија теоретску вредност.



Пример 6 (Дводимензиона нормална расподела)**.** У овом примеру демонстрираћемо генерисање случајних вектора. Зарад једноставности, као пример узимамо дводимензиону нормалну расподелу.

Нека је (Y_1, Y_2) случајан вектор са дводимензионом нормалном расподелом:

$$(Y_1, Y_2)^\top \sim \mathcal{N}\left(\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}, \begin{pmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_1^2\end{pmatrix}\right).$$

Маргиналне расподеле случајних величина Y_1 и Y_2 су $Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, а ρ је коефицијент корелације између Y_1 и Y_2.

Познато је да је тада и условна расподела $Y_2|Y_1 = y$ такође нормална, односно

$$Y_2|Y_1 = y \sim \mathcal{N}\left(\mu_2 + \frac{\sigma_2}{\sigma_1}\rho(y - \mu_1), (1-\rho^2)\sigma_2^2\right).$$

За наш пример узимамо параметре $\mu_1 = 1, \mu_2 = 2, \sigma_1 = \sigma_2 = 1, \rho = 0.3$. Користећи претходне формуле, добијамо расподеле

$$\begin{aligned} Y_1 &\sim \mathcal{N}(1, 1), \\ Y_2|Y_1 = y &\sim \mathcal{N}\left(1.7 + 0.3y, 0.91\right). \end{aligned}$$

За генерисање пратимо поступак описан у делу 2.3. Прво генеришемо узорак $y_1^{(1)}, \dots, y_1^{(n)}$ из маргиналне расподеле Y_1 методом стохастичке колокације, па затим генеришемо вредности за Y_2, уз помоћ дводимензионог интерполационог полинома $g_{3,3}(x, y_1) \approx F^{-1}_{\mathcal{N}(1.7+0.3y_1, 0.91)}(F_X(x))$. Користимо 3×3 мрежу за интерполацију.

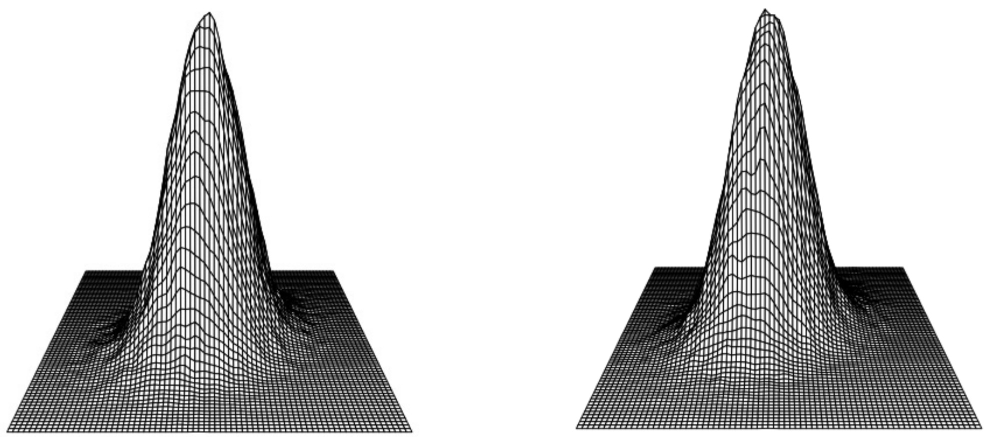
У табели 3.7 поредимо брзине генерисања узорка обима 100,000 наше имплементације (у табели "SCMC"), генератора `rmvnorm` из R пакета `mvtnorm` (у табели "mvtnorm") и генератора `rmvn` из пакета `mvnfast` (у табели "mvnfast"). Показаћемо медијану и просек времена извршавања 100 понављања генерисања узорка.

генератор	просечно време	медијално време	број понављања
mvtnorm	13.55 ms	12.08 ms	100
mvnfast	5.51 ms	5.19 ms	100
SCMC	7.18 ms	6.89 ms	100

Табела 3.7: Поређење времена генерисања за дводимензиону нормалну расподелу расподелу.

Наша имплементација је бржа од имплементације из `mvtnorm` пакета. С друге стране, спорија је од имплементације `mvnfast` пакета, али не заостаје много. Међутим, треба имати у виду да смо овде поредили методу стохастичке колокације са имплементацијама које су специјализоване за вишедимензиону нормалну расподелу и сходно су оптимизоване, док је циљ методе стохастичке колокације да буде што општија.

На слици 3.6 показане су оцене густина узорка генерисаног генератором из `mvtnorm` пакета и оног генерисаног методом стохастичке колокације. Слике су врло сличне што је индикација да је расподела добијена методом стохастичке колокације коректна. То потврђује и тест `kde.test` из пакета `ks`.



Слика 3.6: Графици оцене густине расподеле узорка генерисаних генератором из `mvtnorm` пакета (лево) и методом стохастичке колокације (десно). Оцене густина су добијене функцијом `kde2d` из пакета `MASS`.

#

Поглавље 4

Закључак

У овом раду представљена је метода стохастичке колокације за генерисање случајних величина (енгл. *Stochastic Collocation Monte Carlo*), која је заснована на интерполацији функције $F_Y^{-1}\big(F_X(x)\big)$, где је Y случајна величина коју треба генерисати, док је X нека помоћна случајна величина за коју је познат брз алгоритам генерисања.

Описана су два приступа избору погодних чворова за интерполацију. Прва метода се заснива на избору чворова оптималних у средњеквадратном смислу (у односу на тежинску функцију $f_X(x)$). Показано је да се оптимални чворови у средњеквадратном смислу поклапају са чворовима Гаусове квадратурне формуле. Друга метода избора која је представљена подразумева одабир нула Чебишевљевих полинома за чворове интерполације, чија су погодна својства за интерполацију већ позната, али и показана у раду.

Фокус је у највећем делу рада био на генерисању једнодимензионих случајних величина, али је приказана и метода генерисања вишедимензионих случајних величина, односно случајних вектора, уколико су познате зависности између елемената вектора, односно условне расподеле једног елемента у односу на остале. Описана метода се ослања на вишедимензиону интерполацију на правоугаоној мрежи.

Приказан је Зигурат алгоритам за генерисање случајних бројева из нормалне расподеле, који је један од најбржих алгоритама такве врсте и који је заслужан за велика убрзања у извршеним експериментима.

Коначно, дато је неколико примера који демонстрирају ефикасност методе за генерисање разних типова расподела. Резултати методе су поређени са уграђеним генераторима у програмском језику R. Квалитет генерисаних узорака, у смислу минимизације тест статистика Колмогоров–Смирновљевог, Крамер–фон Мизесовог и Андерсон–Дарлинговог теста, је био близак уграђеним генераторима. Поред тога, брзина генерисања је за све расподеле била већа него код уграђених генератора, а у већини примера је смањење у времену генерисања било вишеструко.

Дат је и пример генерисања дводимензионе нормалне расподеле као илустрација примене методе при генерисању вишедимензионих расподела. Један смер будућег истраживања је управо даље развијање методе за генерисање случајних вектора. Могуће побољшање се добија преласком са интерполације на правоугаоној мрежи, која захтева велики број чворова, на интерполацију на проређеним мрежама (енгл. *sparse grids*), чиме се значајно смањује број потребних чворова, а чува квалитет апроксимације.

Литература

[1] T. W. Anderson and D. A. Darling. A test of goodness of fit. *Journal of the American Statistical Association*, 49(268):765–769, dec 1954.

[2] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, 12(4):273–288, 2000.

[3] J.-P. Berrut and L. N. Trefethen. Barycentric lagrange interpolation. *SIAM Review*, 46(3):501–517, jan 2004.

[4] D. A. Darling. The Kolmogorov-Smirnov, Cramer-von Mises tests. *The Annals of Mathematical Statistics*, 28(4):823–838, 1957.

[5] P. Davis. *Methods of numerical integration.* Academic Press, Orlando, 1984.

[6] D. Eddelbuettel. *Ziggurat Revisited*, 2018. Доступно на `https://cran.rstudio.com/web/packages/RcppZiggurat/vignettes/RcppZiggurat.pdf`.

[7] M. Embree. Lecture 15: Chebyshev polynomials for optimal interpolation, Spring 2016.

[8] W. Gander. Change of basis in polynomial interpolation. *Numerical Linear Algebra with Applications*, 12(8):769–778, 2005.

[9] W. Gautschi. *Numerical analysis*. Birkhäuser, Boston, 2012.

[10] G. H. Golub and J. H. Welsch. Calculation of gauss quadrature rules. *Mathematics of Computation*, 23(106):221–221, may 1969.

[11] L. A. Grzelak, J. Witteveen, M. Suarez-Taboada, and C. W. Oosterlee. The stochastic collocation monte carlo sampler: Highly efficient sampling from 'expensive' distributions. *SSRN Electronic Journal*, 2014.

[12] J. Hesthaven. *Spectral methods for time-dependent problems*. Cambridge University Press, Cambridge, 2007.

[13] N. J. Higham. The numerical stability of barycentric lagrange interpolation. *IMA Journal of Numerical Analysis*, 24(4):547–556, Oct 2004.

[14] D. Kincaid and W. Cheney. *Numerical Analysis: Mathematics of Scientific Computing.* Brooks/Cole Publishing Co., 1991.

[15] K. Knight. *Mathematical statistics.* Chapman & Hall/CRC Press, 2000.

[16] P. Leong, G. Zhang, D.-U. Lee, W. Luk, and J. Villasenor. A comment on the implementation of the Ziggurat method. *Journal of Statistical Software*, 12(7), 2005.

[17] G. Marsaglia. Generating a variable from the tail of the normal distribution. *Technometrics*, 6(1):101–102, feb 1964.

[18] G. Marsaglia and W. W. Tsang. The Ziggurat method for generating random variables. *Journal of Statistical Software*, 5(8), 2000.

[19] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 1998.

[20] A. B. Owen. *Monte Carlo theory, methods and examples.* 2013.

[21] V. Y. Pan. How bad are Vandermonde matrices? *SIAM Journal on Matrix Analysis and Applications*, 37(2):676–694, 2016.

[22] D. Radunović. *Numeričke metode.* Akademska misao, Beograd, 2004.

[23] M. A. Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, sep 1974.

[24] G. W. Stewart. *Afternotes on numerical analysis.* Society for Industrial and Applied Mathematics, Philadelphia, Pa, 1996.

[25] E. W. Weisstein. Horner's rule. From MathWorld—A Wolfram Web Resource.

[26] E. W. Weisstein. Rolle's theorem. From MathWorld—A Wolfram Web Resource.

[27] D. Xiu. *Numerical methods for stochastic computations : a spectral method approach.* Princeton University Press, Princeton, N.J, 2010.

Биографија

Благоје Ивановић је рођен 3. августа 1994. године у Београду. Похађао је основну школу „Светозар Милетић“ у Земуну. Завршио је средњу електротехничку школу „Никола Тесла“ у Београду 2013. године, када уписује Математички факултет Универзитета у Београду, на смеру статистика, актуарска и финансијска математика. Основне академске студије завршава 2017. године са просечном оценом 9.81. Још од средње школе се интересује за програмирање и рачунарство, док су му главне области интересовања у математици вероватноћа и статистика, а од скора и математичка оптимизација.