KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

TAIKOMOSIOS MATEMATIKOS KATEDRA

Artūras Katvickis

MAKSIMUMŲ SANDAUGOS ANALIZĖ

Magistro darbas

Vadovas
prof. dr. A. Aksomaitis

KAUNAS, 2005

KAUNO TECHNOLOGIJOS UNIVERSITETAS

FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS

TAIKOMOSIOS MATEMATIKOS KATEDRA

TVIRTINU
Katedros vedėjas
prof. dr. J.Rimas
2005 06 03

MAKSIMUMŲ SANDAUGOS ANALIZĖ

Taikomosios matematikos magistro baigiamasis darbas

Kalbos konsultantas
dr. J. Džežulskienė
2005 05 31

Vadovas
prof. dr. A. Aksomaitis
2005 06 03

Recenzentas
doc.dr. K. Padvelskis
2005 05 31

Atliko
FMMM-3 gr. stud.
A. Katvickis
2005 05 27

KAUNAS, 2005

KVALIFIKACINĖ KOMISIJA

Pirmininkas:	Leonas Saulis, profesorius (VGTU)
Sekretorius:	Eimutis Valakevičius, docentas (KTU)
Nariai:	Jonas Algimantas Aksomaitis, profesorius (KTU)
	Vytautas Janilionis, docentas (KTU)
	Vidmantas Povilas Pekarskas, profesorius (KTU)
	Rimantas Rudzkis, banko NORD/LB vyriausiasis analitikas
	Zenonas Navickas, profesorius (KTU)
	Arūnas Barauskas, UAB „Elsis“ generalinio direktoriaus pavaduotojas

Katvickis A. Analysis of maximum's multiplication. : Master's work in applied mathematics / supervisor dr. prof. A. Aksomaitis; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2005.

SUMMARY

We consider structure (k maximum's multiplication)

$$V_n^{(k)} = Z_n^{(1)} \cdot Z_n^{(2)} \cdot \ldots \cdot Z_n^{(k)};$$

where $Z_n^{(j)} = \max\left(X_1^{(j)}, X_2^{(j)}, \ldots, X_n^{(j)}\right), j = \overline{1,k}$ and $(X_1^{(j)}, X_2^{(j)}, \ldots, X_n^{(j)})$ - is independent identically distributed random values from a uniform distribution $(X \sim T(0,1))$. We find distribution function, limiting distribution function and estimate rate of convergence if n is constant and $n = N_n$ is a discrete random value.

TURINYS

PAVEIKSLŲ SĄRAŠAS

ĮVADAS

Potvyniai, liūtys, pernelyg aukšta arba žema oro temperatūra, uraganai ir kitos gaivalinės nelaimės gali nusinešti daug žmonių aukų bei padaryti didelių materialinių nuostolių, jei tik bendruomenė nėra pasirengusi su jomis kovoti. Tokių nelaimių išvengti neįmanoma, tačiau galima imtis priemonių, sumažinančių jų padarinius. Pavyzdžiui, statant užtvanką reikėtų atsižvelgti į galimų nelaimių mastus bei parinkti atitinkamas medžiagas.

Įstaigos, kontroliuojančios oro taršą, stebi ir fiksuoja nuodingų medžiagų koncentraciją. Jų darbo tikslas – užtikrinti, kad maksimali nuodingų medžiagų koncentracija neviršytų leistinos ribos.

Metalas yra veikiamas korozijos. Pažeidimų dydis apibūdinamas maksimaliu korozijos gyliu. Svarbių visuomenei objektų projektuotojai bei statytojai suinteresuoti, kad korozijos poveikis būtų kuo mažesnis.

Sprendžiant šias ir daugelį kitų technikoje atsirandančių problemų susiduriama su stebėjimų maksimalia reikšme. Dažnai reikalingi metodai turimai informacijai apibūdinti bei išanalizuoti.

Kai atsitiktinių poveikių skaičius didelis, naudotinas maksimumų ir minimumų (ekstremumų) ribinės teoremos. Jos teikia galimybes sudėtingus ekstremumų skirstinius aproksimuoti paprastesniais (ribiniais) skirstiniais. Iškyla aproksimavimo paklaidų problemos.

Šias asimptotines problemas sprendžiu teikiamajame magistro darbe. Tyrimo objektas – stochastinių maksimumų sandauga.

Dviejų maksimumų sandaugos asimptotiniai tyrimai buvo atliekami [2], [3] darbuose. Darbe pateikta maksimumų sandaugos analizė, kai jų skaičius didesnis už 2.

Šia tematika skaitytas pranešimas mokslinėje konferencijoje. Publikacija pateikta priede 1.

1. TEORINĖ DALIS

1.1. MAKSIMUMŲ RIBINIAI SKIRSTINIAI

Tarkime, kad $(X_1, X_2, X_3, ..., X_n)$ – nepriklausomi, vienodai pasiskirstę atsitiktiniai dydžiai su pasiskirstymo funkcija $F(x) = P(X_j < x),\ j = 1,2,...,n$.

Pažymėkime:

$$Z_n = \max(X_1, X_2, X_3, ..., X_n).$$

Maksimumo skirstinio funkcija:

$$H_n(x) = P(Z_n < x) = F^n(x).$$

Ieškoma neišsigimusi pasiskirstymo funkcija $H(x)$ tokia, kad

$$\lim_{n\to\infty} H_n(a_n + b_n x) = H(x);$$

čia a_n ir b_n tam tikros centravimo ir normavimo konstantos.

Monografijoje [1] įrodyta, kad funkcija $H(x)$ gali būti trijų rūšių.

Pažymėkime:

$$\begin{aligned} \alpha(F) &= \inf\{x : F(x) > 0\} \\ \omega(F) &= \sup\{x : F(x) < 1\} \end{aligned}.$$

Pateiksime teoremas, apibūdinančias ribines skirstinio funkcijas $H(x)$ ir nurodysime normalizavimo konstantų a_n ir b_n parinkimo algoritmą.

1.1 teorema. Tarkim $\omega(F) = \infty$ ir egzistuoja toks $\gamma > 0$, kad bet kuriam $x > 0$ yra tenkinama lygybė

$$\lim_{t\to\infty} \frac{1 - F(tx)}{1 - F(t)} = x^{-\gamma}.$$

Tuomet egzistuoja centravimo ir normavimo konstantos a_n ir b_n su kuriomis

$$\lim_{n\to\infty} H_n(a_n + b_n x) = H_{1,\gamma}(x),$$

čia

$$H_{1,\gamma}(x) = \begin{cases} \exp(-x^{-\gamma}), & x > 0 \\ 0, & x \le 0 \end{cases}.$$

Centravimo ir normavimo konstantos gali būti parinktos tokiu būdu:

$$a_n = 0,\ b_n = \inf\left\{x : 1 - F(x) \le \frac{1}{n}\right\}.$$

1.2 teorema. Tarkim $\omega(F) < \infty$ ir egzistuoja toks $\gamma > 0$, kad bet kuriam $x > 0$ yra tenkinama lygybė

$$\lim_{t\to\infty}\frac{1-F^*(t\,x)}{1-F^*(t)} = x^{-\gamma};$$

čia skirstinio funkcija $F^*(x) = F\left(\omega(F) - \frac{1}{x}\right)$, $x < 0$.

Tuomet egzistuoja centravimo ir normavimo konstantos a_n ir b_n, su kuriomis

$$\lim_{n\to\infty} H_n(a_n + b_n x) = H_{2,\gamma}(x),$$

čia

$$H_{2,\gamma}(x) = \begin{cases} 1, & x \ge 0, \\ \exp\left(-(-x)^\gamma\right), & x < 0. \end{cases}$$

Centravimo ir normavimo konstantos gali būti parinktos tokiu būdu:

$$a_n = \omega(F),\ b_n = \omega(F) - \inf\left\{x : 1 - F(x) \le \frac{1}{n}\right\}.$$

1.3 teorema. Tarkim, kokiam nors fiksuotam a

$$\int_a^{\omega(F)} (1 - F(y))dy < \infty,$$

o bet kuriam t $\alpha(F) < t < \omega(F)$ apibrėžta funkcija

$$R(t) = (1 - F(t))^{-1}\int_t^{\varpi(F)} (1 - F(y))dy.$$

Jeigu su $\forall x \in R$ $\lim\limits_{t\to\omega(F)}\frac{1-F(t+xR(t))}{1-F(t)} = e^{-x}$, tai egzistuoja centravimo ir normavimo konstantos a_n ir b_n su kuriomis

$$\lim_{n\to\infty} H_n(a_n + b_n x) = H_{3,0}(x);$$

čia

$$H_{2,\gamma}(x) = \exp\left(-e^{-x}\right),\ -\infty < x < \infty.$$

Centravimo ir normavimo konstantos gali būti parinktos tokiu būdu:

$$a_n = \inf\left\{x : 1 - F(x) \le \frac{1}{n}\right\},\ b_n = R(a_n).$$

Visuose atvejuose centravimo ir normavimo konstantos gali būti parinktos ne vieninteliu būdu.

Tarkim yra žinomos konstantų sekos $\{a_n\}$ ir $\{b_n\}$ su kuriomis $\lim\limits_{n\to\infty} H_n\left(a_n + b_n x\right) = H(x)$ bei sekos $\{A_n\}$ ir $\{B_n\}$, tenkinančios lygybes $\lim\limits_{n\to\infty} \dfrac{a_n - A_n}{b_n} = 0$ ir $\lim\limits_{n\to\infty} \dfrac{b_n}{B_n} = 1$. Tuomet bus teisinga lygybė $\lim\limits_{n\to\infty} H_n\left(A_n + B_n x\right) = H(x)$.

1.2. NAGRINĖJAMAS MODELIS BEI SPRENDŽIAMI UŽDAVINIAI

Maksimumų sandauga apibūdinama tokia struktūra:

$$V_n^{(k)} = Z_n^{(1)} \cdot Z_n^{(2)} \cdot \ldots \cdot Z_n^{(k)}, \tag{1.1}$$

čia $Z_n^{(j)} = \max\left(X_1^{(j)}, X_2^{(j)}, \ldots, X_n^{(j)}\right), j = \overline{1,k}$, t. y. $Z_n^{(j)}$ yra j-tosios atsitiktinės imties didumu n maksimali reikšmė. Imtys $(X_1^{(j)}, X_2^{(j)}, \ldots, X_n^{(j)})$ yra paprastosios atsitiktinės iš tolygiosios generalinės aibės $(X \sim T(0,1))$.

Priklausomai nuo imčių didumo turime du atvejus:

1) imčių didumai yra determinuoti dydžiai n;
2) imčių didumai yra atsitiktiniai dydžiai $N = N_n$, nepriklausantys nuo $X_1^{(j)},\ j = \overline{1,k}$

Darbe sprendžiami šie uždaviniai:

- rasti maksimumų sandaugos pasiskirstymo dėsnį, kai imčių didumai yra determinuotas dydis;
- nustatyti ar egzistuoja neišsigimęs maksimumų sandaugos ribinis skirstinys, kai imčių didumai neaprėžtai didėja ir rasti jį;
- įvertinti konvergavimo į ribinį skirstinį greitį;
- rasti maksimumų sandaugos pasiskirstymo dėsnį, kai imčių didumai yra atsitiktinis dydis;
- nustatyti ar egzistuoja neišsigimęs maksimumų sandaugos ribinis skirstinys, kai imčių didumai yra geometrinis atsitiktinis dydis ir rasti ribinį skirstinį;
- įvertinti konvergavimo į ribinį skirstinį greitį.

2. TIRIAMOJI DALIS

2.1. FIKSUOTO DIDUMO IMČIŲ MAKSIMUMŲ SANDAUGA

2.1.1. SKIRSTINIO FUNKCIJA IR RIBINIS SKIRSTINYS

Bendru atveju, bet kurios atsitiktinio argumento funkcijos $Y = f(X_1, X_2, ..., X_n)$ skirstinys nusakomas integralu:

$$F_Y(y) = \int \cdots_D \int p(x_1, x_2, ..., x_n) dx_1 dx_2 \cdots dx_n \ ;$$

čia sritis D apibrėžiama tokiu būdu:

$$D = \{(x_1, x_2, ..., x_n) : f(x_1, x_2, ..., x_n) < y\}.$$

Tačiau tiesiogiai integruojant rasti ieškoma pasiskirstymo funkcija yra ganėtinai sudėtinga.

2.1 teorema. Jeigu $X_1^{(j)} \sim T(0,1)$, tai

$$P\left(V_n^{(k)} < x\right) = x^n \sum_{j=0}^{k-1} \frac{(-n \ln x)^j}{j!};$$

čia $0 < x < 1$.

Įrodymas. Šią teorema įrodoma pasinaudojant charakteristinių funkcijų savybėmis.

Iš (1.1), logaritmuodami gauname:

$$\ln V_n^{(k)} = \ln\left(Z_n^{(1)} \cdot Z_n^{(2)} \cdot \ldots \cdot Z_n^{(k)}\right) = \sum_{j=1}^{k} \ln Z_n^{(j)} .$$

Pažymėjus $W_n^{(k)} = \ln V_n^{(k)}, T_n^{(j)} = \ln Z_n^{(j)}, j = \overline{1,k}$, gauta lygybę galima užrašyti taip:

$$W_n^{(k)} = \sum_{j=1}^{k} T_n^{(j)} .$$

Žinoma, kad $P\left(Z_n^{(j)} < x\right) = x^n$, $0 < x < 1$. O dydžių $T_n^{(j)}$ pasiskirstymo funkcijos

$$P\left(T_n^{(j)} < u\right) = P\left(\ln Z_n^{(j)} < u\right) = P\left(Z_n^{(j)} < e^u\right) = e^{u \cdot n}, u < 0$$

Kadangi dydžiai $Z_n^{(j)}$ yra nepriklausomi vienodai pasiskirstę, tai ir dydžiai $T_n^{(j)}$ turės tas pačias savybes. Tuomet galima pasinaudoti savybe, jog nepriklausomų atsitiktinių dydžių sumos charakteristinė funkcija yra dedamųjų narių charakteristinių funkcijų sandauga, t. y.

$$f_{W_n^{(k)}}(t) = \prod_{j=1}^{k} f_{T_n^{(j)}}(t) .$$

$$f_{T_n^{(j)}}(t) = Me^{itT_n^{(j)}} = \int_{-\infty}^{0} e^{itu} dP\left(T_n^{(j)} < u\right) = \frac{n}{it+n}, \quad j = \overline{1,k}$$

$$f_{W_n^{(k)}}(t) = \left(\frac{n}{it+n}\right)^k$$

Pasinaudoję atgręžimo formule [4] $p(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itx} f(t)dt$, gauname dydžio $W_n^{(k)}$ tankį:

$$p_{W_n^{(k)}}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itx}\left(\frac{n}{it+n}\right)^k dt = \frac{n^k}{2\pi} 2\pi i \operatorname{Res}_{t=in} \frac{e^{-itx}}{(it+n)^k} = i n^k \frac{1}{(k-1)!}\lim_{t\to in}\frac{d^{k-1}}{dt^{k-1}}\left((t-in)^k e^{-itx}\frac{1}{(it+n)^k}\right) =$$

$$= i n^k \frac{1}{(k-1)!}\frac{1}{i^k}\lim_{t\to in}\frac{d^{k-1}}{dt^{k-1}} e^{-itx} = \frac{n^k}{i^{k-1}(k-1)!} i^{k-1}(-x)^{k-1}\lim_{t\to in} e^{-itx} = \frac{n^k}{(k-1)!} e^{nx}(-x)^{k-1}$$

Suintegravę gautą tankio funkciją gausime:

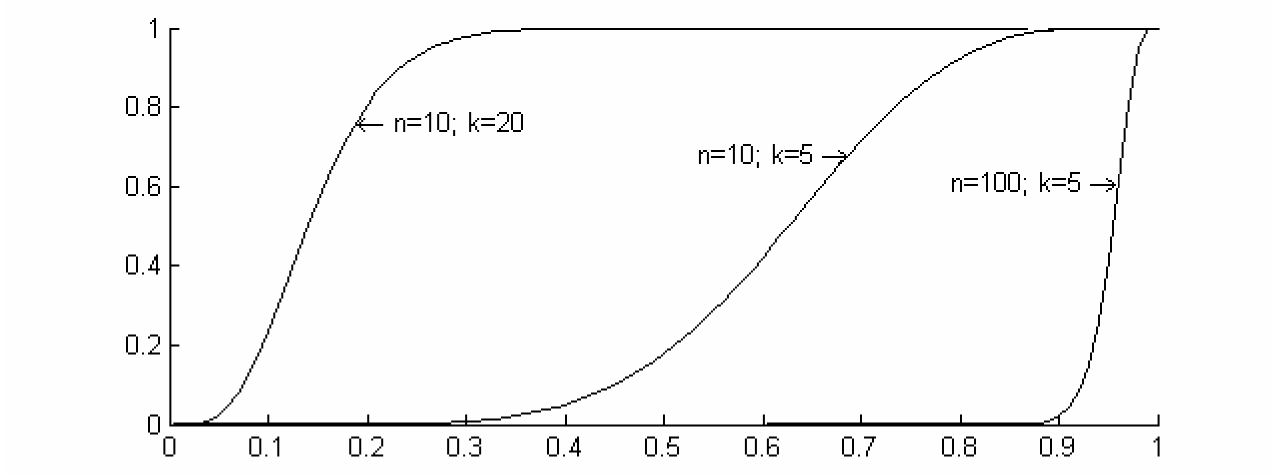
$$F_{W_n^{(k)}}(x) = \int_{-\infty}^{x} p_{W_n^{(k)}}(x)dx = e^{nx}\sum_{j=0}^{k-1}\frac{(-nx)^j}{j!}, \quad x < 0$$

Kadangi $W_n^{(k)} = \ln V_n^{(k)}$, tai $F_{V_n^{(k)}}(x) = F_{W_n^{(k)}}(\ln x)$, t.y.

$$P\left(V_n^{(m)} < x\right) = x^n \cdot \sum_{i=0}^{m-1}\frac{(-n\cdot \ln x)^i}{i!}, \quad 0 < x < 1.$$

Teorema įrodyta.

Šio skirstinio grafikas, esant skirtingoms parametrų n ir k reikšmėms pateiktas 2.1 paveiksle.



2.1 pav. Maksimumų sandaugos pasiskirstymo funkcija

Rasime ribinę maksimumų sandaugos pasiskirstymo funkciją

$$H(x,k) = \lim_{n\to\infty} F_{V_n^{(k)}}(a_n + b_n x);$$

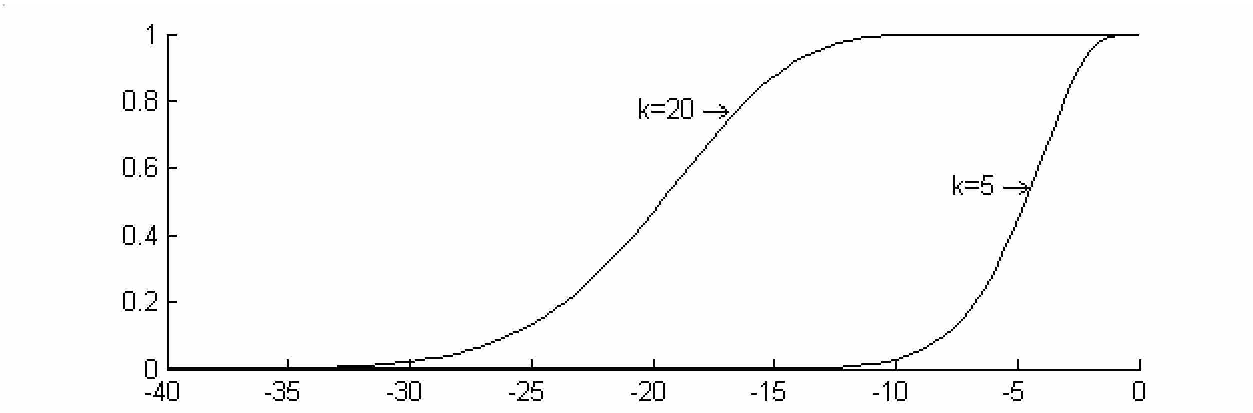
čia a_n ir b_n - centravimo ir normavimo konstantos.

Pasirenkame $a_n = 1$ ir $b_n = \frac{1}{n}$. Tuomet :

$$H(x,k)=\lim_{n\to\infty}F_{V_n^{(k)}}\left(1+\frac{x}{n}\right)=\lim_{n\to\infty}\left(1+\frac{x}{n}\right)^n\sum_{j=0}^{k-1}\frac{\left(-n\ln\left(1+\frac{x}{n}\right)\right)^j}{j!},$$

$$H(x,k)=e^x\cdot\sum_{j=0}^{k-1}\frac{(-x)^j}{j!},\ x<0.$$

Šio skirstinio grafikas, esant skirtingoms parametrų *n* ir *k* reikšmėms pateiktas 2.2 paveiksle.



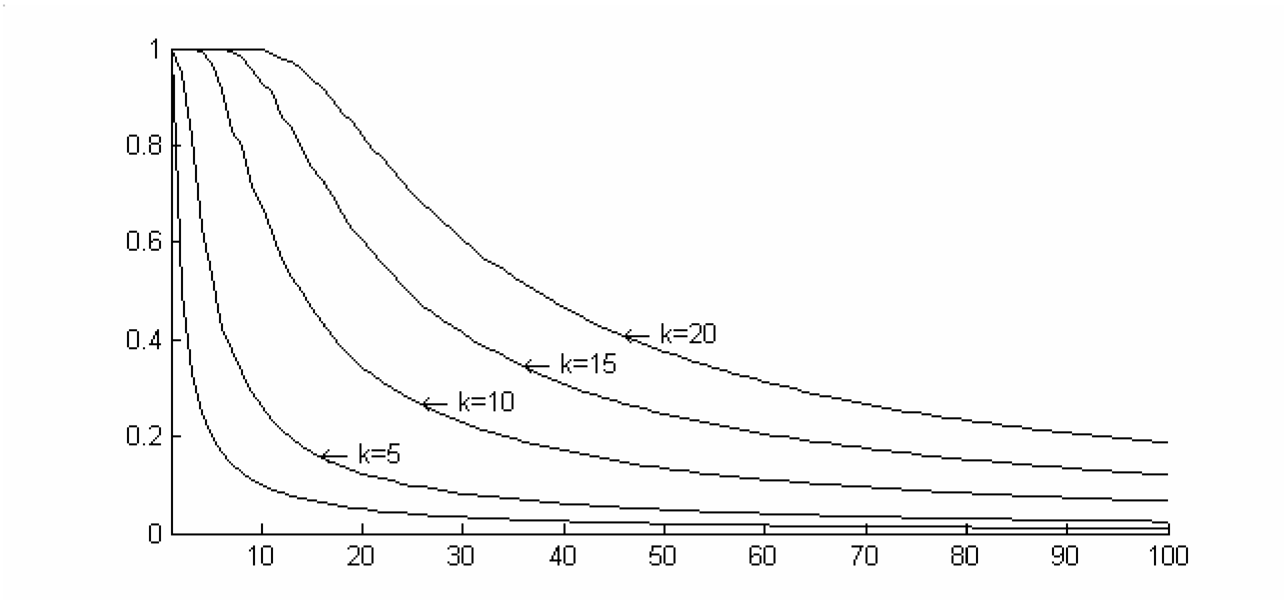
2.2 pav. Maksimumų sandaugos ribinė pasiskirstymo funkcija

2.1.2. KONVERGAVIMO GREITIS

Kad žinotume, kiek ribinė pasiskirstymo funkcija atitinka maksimumų sandaugos skirstinį, įvertintiname paklaidą:

$$\Delta_1(x,n,k)=\left|F_{V_n^{(k)}}(a_n+b_nx)-H(x,k)\right|.$$

Paklaidą vertiname eksperimentiškai, t.y. skaičiuodami maksimalų atstumą tarp ribinio skirstinio ir maksimumų sandaugos pasiskirstymo funkcijos, esant skirtingiems imties tūriams n ir dauginamųjų maksimumų skaičiams k. Grafikai gauti keičiant imties didumą nuo 1 iki 100 bei skaičiuojant 5, 10, 15 ir 20 maksimumų sandaugas.



2.3 pav. Eksperimentinis paklaidos vertinimas

Iš grafikų pavidalo spėjama, kad konvergavimo greičio eilė n atžvilgiu bus $\frac{1}{n}$, todėl palyginimui yra nubrėžta ir ši funkcija.

Įvertiname konvergavimo greitį analiziškai.

2.2 teorema. $\Delta_1(x,n,k) \le \frac{x^2 e^x}{2n} \sum_{j=0}^{k-1} \frac{(-x)^j}{j!}, \ -n < x < 0$.

Įrodymas.

$$\Delta_1(x,n,k) = \left| H(x,k) - F_{V_n^{(k)}}(a_n + b_n x) \right| = \left| e^x \sum_{j=0}^{k-1} \frac{(-x)^j}{j!} - \left(1+\frac{x}{n}\right)^n \sum_{j=0}^{k-1} \frac{\left(-n\ln\left(1+\frac{x}{n}\right)\right)^j}{j!} \right|, \ -n < x < 0$$

Pastebėsime, kad nelygybė

$$e^x \sum_{j=0}^{k-1} \frac{(-x)^j}{j!} \ge \left(1+\frac{x}{n}\right)^n \sum_{j=0}^{k-1} \frac{\left(-n\ln\left(1+\frac{x}{n}\right)\right)^j}{j!}$$

teisinga bet kuriems $n,k \in N$, $x \in (-n;0)$. Todėl modulio ženklas nėra būtinas. Pasinaudoję lygybe

$$ab - cd = a(b-d) + d(a-c),$$

gauname:

$$\Delta_1(x,n,k) = \sum_{j=0}^{k-1} \frac{(-x)^j}{j!} \left(e^x - \left(1+\frac{x}{n}\right)^n \right) + \left(1+\frac{x}{n}\right)^n \sum_{j=0}^{k-1} \frac{(-x)^j - \left(-n\ln\left(1+\frac{x}{n}\right)\right)^j}{j!}.$$

Kadangi $e^x \geq \left(1+\frac{x}{n}\right)^n$, tai $-x \leq -n\ln\left(1+\frac{x}{n}\right)$, t.y. antrasis sumos dėmuo yra neteigiamas ir jį atmetus suma nesumažės. Todėl

$$\Delta_1(x,n,k) \leq \sum_{j=0}^{k-1} \frac{(-x)^j}{j!}\left(e^x - \left(1+\frac{x}{n}\right)^n\right) \tag{2.1}$$

Išskleidę e^x ir $\left(1+\frac{x}{n}\right)^n$ laipsninėmis eilutėmis ir atėmę vieną iš kitos gauname:

$$e^x - \left(1+\frac{x}{n}\right)^n = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots - 1 - n\frac{x}{n} - \frac{n(n-1)}{2!}\frac{x^2}{n^2} - \frac{n(n-1)(n-2)}{3!}\frac{x^3}{n^3} -$$

$$- \frac{n(n-1)(n-2)(n-3)}{4!}\frac{x^4}{n^4} - \ldots =$$

$$= \frac{x^2}{2!}\left(1 - \frac{n-1}{n}\right) + \frac{x^3}{3!}\left(1 - \frac{(n-1)(n-2)}{n^2}\right) + \frac{x^4}{4!}\left(1 - \frac{(n-1)(n-2)(n-3)}{n^3}\right) + \ldots =$$

$$= \frac{x^2}{2!}\left(1 - 1 + \frac{1}{n}\right) + \frac{x^3}{3!}\left(1 - 1 + \frac{3}{n} - \frac{2}{n^2}\right) + \frac{x^4}{4!}\left(1 - 1 + \frac{6}{n} - \frac{11}{n^2} + \frac{6}{n^3}\right) + \ldots =$$

$$= \frac{x^2}{2!n} + \frac{x^3}{3!n}\left(3 - \frac{2}{n}\right) + \frac{x^4}{4!n}\left(6 - \frac{11}{n} + \frac{6}{n^2}\right) + \ldots \leq \frac{x^2}{2!n} + \frac{x^3}{3!n}\cdot 3 + \frac{x^4}{4!n}\cdot 6 + \ldots =$$

$$= \sum_{j=2}^{\infty} \frac{x^j}{j!n}\left(1 + 2 + 3 + \ldots + (j-1)\right) = \sum_{j=2}^{\infty} \frac{x^j}{j!n}\frac{j(j-1)}{2} = \frac{x^2}{2n}\sum_{j=2}^{\infty}\frac{x^{j-2}}{(j-2)!} = \frac{x^2 e^x}{2n} \tag{2.2}$$

Iš (2.1) ir (2.2) gauname, kad:

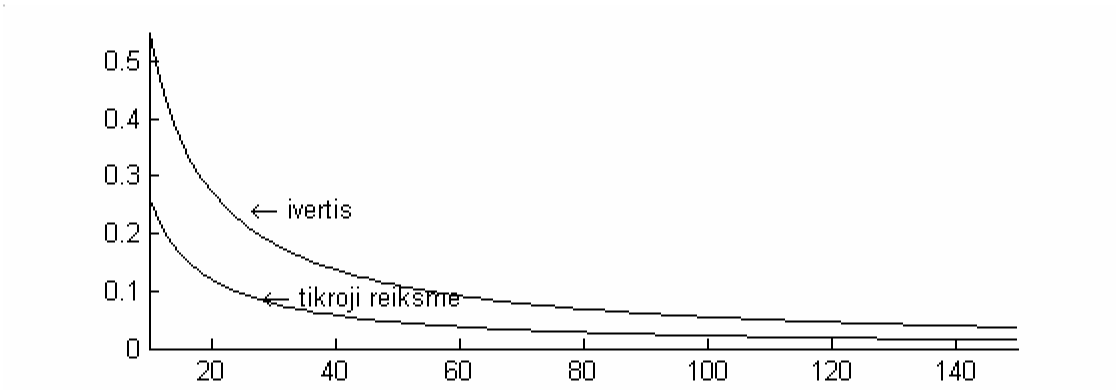
$$\Delta_1(x,n,k) \leq \frac{x^2 e^x}{2n}\sum_{j=0}^{k-1}\frac{(-x)^j}{j!}, \quad -n < x < 0\,.$$

Teorema įrodyta.

Matome kad konvergavimo greitis n atžvilgiu yra $\frac{1}{n}$ eilės. Šį rezultatą patvirtina ir aukščiau pateikti modeliavimo rezultatai.

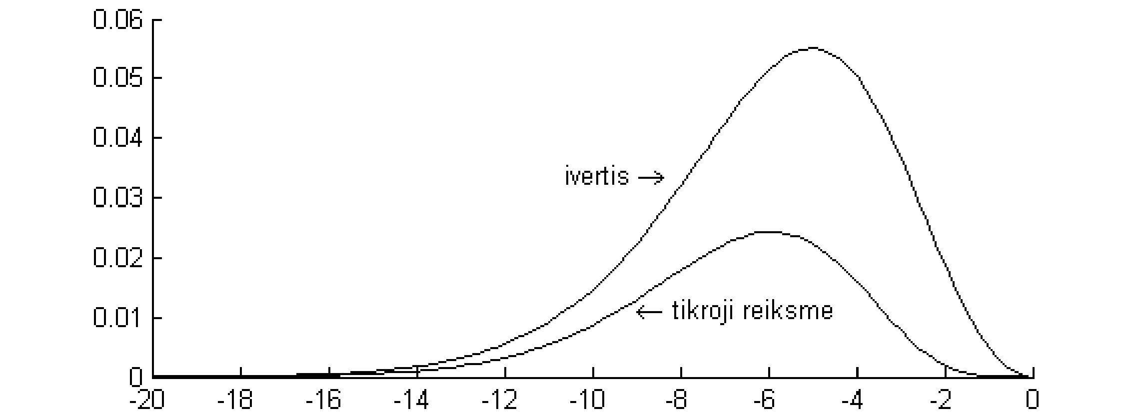
Gautas įvertis yra trijų kintamųjų funkcija ir todėl nagrinėsime šios funkcijos projekcijas į tam tikras plokštumas.

Fiksuodami konkrečią x reikšmę bei keisdami imčių didumus n matome, kad didėjant n skirtumas tarp paklaidos tikrosios reikšmės ir įverčio nyksta, kaip to ir reikėjo tikėtis (2.4 paveikslas).



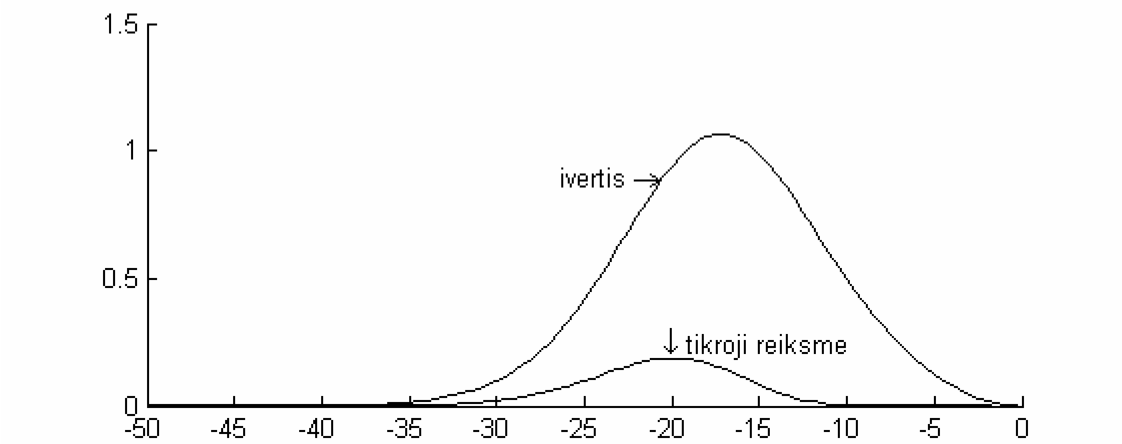
2.4 pav. Paklaida kai *k* = 5, *x* = -5

Fiksuodami imties didumus n galime pastebėti, kad tiek įvertis tiek tikroji paklaida turi vieną ekstremumą pagal x. Taip pat šio ekstremumo aplinkoje didžiausias yra ir absoliutus skirtumas tarp įverčio ir tikrosios paklaidos (2.5 paveikslas).



2.5 pav. Paklaida kai *k* = 5; *n* = 100

Keisdami k reikšmes fiksuotam n matome, kad gautasis įvertis tam tikrame x kitimo intervale nutolsta nuo tikrosios paklaidos reikšmės (2.6 paveikslas). Todėl galime teigti, kad gautas įvertis yra tinkamas kai $k \ll n$.



2.6 pav. Paklaida kai *k* = 20, *n* = 100

2.2. ATSITIKTINIO DIDUMO IMČIŲ MAKSIMUMŲ SANDAUGA

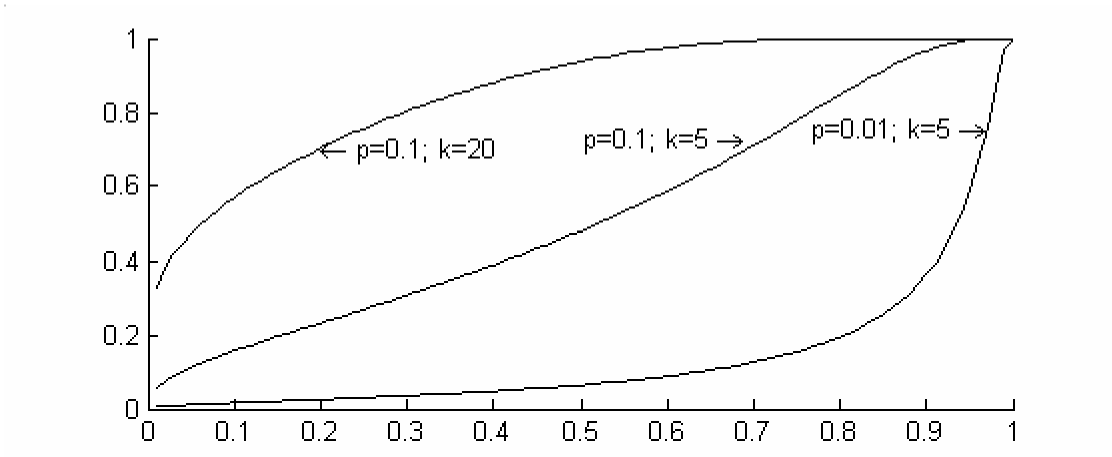
2.2.1. SKIRSTINIO FUNKCIJA IR RIBINIS SKIRSTINYS

Nagrinėsime atvejį, kai imčių didumai yra nepriklausomas nuo visų $X_1^{(j)}$ atsitiktinis dydis N_n, su pasiskirstymo dėsniu $P(N_n = m) = p_{nm}$, čia $p_n \to 0$, kai $n \to \infty$ ir $\sum_m p_{nm} = 1$.

Rasime atsitiktinio dydžio $V_{N_n}^{(k)}$ pasiskirstymo funkciją:

$$P\left(V_{N_n}^{(k)} < v\right) = \sum_{m=1}^{\infty} P(N_n = m) \cdot P\left(V_{N_n}^{(k)} < v \mid N_n = m\right) = \sum_{m=1}^{\infty} P(N_n = m) \cdot P\left(V_m^{(k)} < v\right).$$

Pasirinkę N_n pasiskirsčiusi pagal geometrinį dėsnį, t.y. $P(N_n = m) = p \cdot (1-p)^{m-1}$, čia $p = \frac{1}{n}$, gauname $P\left(V_{N_n}^{(k)} < x\right) = \frac{x}{n} \sum_{j=0}^{k-1} \frac{(-\ln x)^j}{j!} \sum_{m=1}^{\infty} m^j \left(x\left(\frac{n-1}{n}\right)\right)^{m-1}$. Parametro n praktinė reikšmė geometrinio skirstinio atveju yra vidutinis imties dydis. Šios funkcijos grafikas, esant skirtingoms parametrų reikšmėms, pateiktas 2.7 paveiksle.



2.7 pav. Maksimumų sandaugos skirstinio funkcija

Suformuluosime ir įrodysime perkėlimo teoremą.

2.3 teorema. Jeigu

$$\lim_{n\to\infty} P\left(V_n^{(k)} < a_n + b_n x\right) = H(x,k)$$

$$\lim_{n\to\infty} P\left(\frac{N_n}{n} < z\right) = A(z),$$

tai

$$\lim_{n\to\infty} P\left(V_{N_n}^{(k)} < a_n + b_n x\right) = \int_0^{\infty} H(xz,k)\,dA(z).$$

Normavimo ir centravimo konstantos gali būti parinktos $a_n = 1$ ir $b_n = \frac{1}{n}$.

Įrodymas.

$$P\left(V_{N_n}^{(k)} < 1 + \frac{x}{n}\right) = \sum_{m=1}^{\infty} P(N_n = m) \cdot P\left(V_m^{(k)} < 1 + \frac{x}{n}\right) =$$

$$= \sum_{m=1}^{\infty} P(N_n = m) \cdot \left(1 + \frac{x}{n}\right)^m \cdot \sum_{j=0}^{k-1} \frac{\left(-m \cdot \ln\left(1 + \frac{x}{n}\right)\right)^j}{j!} =$$

$$= \sum_{m=1}^{\infty} P\left(\frac{N_n}{n} = \frac{m}{n}\right) \cdot \left(1 + \frac{x}{n}\right)^{n \cdot \frac{m}{n}} \cdot \sum_{j=0}^{k-1} \frac{\left(-n \cdot \frac{m}{n} \cdot \ln\left(1 + \frac{x}{n}\right)\right)^j}{j!}$$

Naudojame keitinį $\frac{m}{n} = z$ ir skaičiuojame šio reiškinio ribą, kai $n \to \infty$. Gauname:

$$P\left(V_{N_n}^{(k)} < 1 + \frac{x}{n}\right) \xrightarrow[n \to \infty]{} \int_0^{\infty} e^{x \cdot z} \sum_{j=0}^{k-1} \frac{(-x\,z)^j}{j!} dA(z) = \int_0^{\infty} H(xz, k) dA(z).$$

Teorema įrodyta.

Pritaikykime įrodytą teoremą nagrinėjam atvejui.

2.4 teorema. Tarkime, kad Nn yra geometrinis atsitiktinis dydis su parametru $p = p_n = \frac{1}{n}$. Tada

$$G(x,k) = \int_0^{\infty} H(xz,k) dA(z) = 1 - \left(\frac{x}{x-1}\right)^k, \ x < 0.$$

Įrodymas.

Žinoma kad, $P\left(\frac{N_n}{n} < z\right) \Rightarrow A(z) = 1 - e^{-x}, \ x > 0$. Tada

$$G(x,k) = \int_0^{\infty} H(xz,k) dA(z) = \int_0^{\infty} e^{x \cdot z} \sum_{j=0}^{k-1} \frac{(-x \cdot z)^j}{j!} \cdot e^{-z} dz = \sum_{j=0}^{k-1} \int_0^{\infty} \frac{(-xz)^j}{j!} e^{z(x-1)} dz$$

Pažymėję $I_j(x) = \int_0^{\infty} \frac{(-xz)^j}{j!} e^{z(x-1)} dz$, galime užrašyti, kad $G(x,k) = \sum_{j=0}^{k-1} I_j(x)$. Apskaičiavę integralus $I_j(x)$:

$$I_j(x) = \frac{1}{x-1} \int_0^{\infty} \frac{(-xz)^j}{j!} de^{z(x-1)} = \frac{1}{x-1} \left(e^{z(x-1)} \frac{(-xz)^j}{j!} \Bigg|_0^{\infty} - \int_0^{\infty} e^{z(x-1)} d \frac{(-xz)^j}{j!} \right) =$$

$$= \frac{x}{x-1} \int_0^{\infty} \frac{(-xz)^{j-1}}{(j-1)!} e^{z(x-1)} dz = \frac{x}{x-1} I_{j-1}(x)$$

$$I_j(x) = \frac{x}{x-1} I_{j-1}(x)$$

$$I_j(x) = \left(\frac{x}{x-1}\right)^j I_0(x)$$

$$I_0(x) = \int_0^\infty e^{z(x-1)} dz = \frac{1}{x-1} e^{z(x-1)}\Big|_0^\infty = \frac{1}{1-x}$$

$$I_j(x) = \frac{1}{1-x}\left(\frac{x}{x-1}\right)^j$$

Taigi, ribinė pasiskirstymo funkcija yra

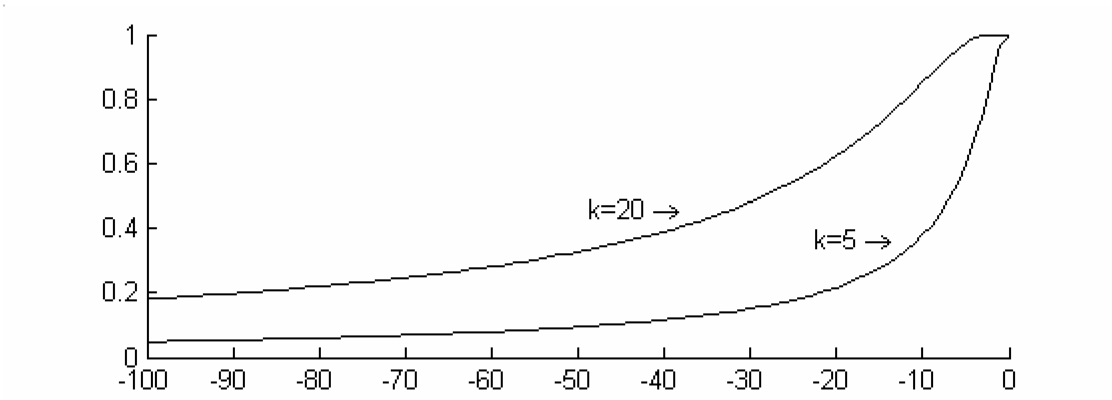
$$G(x,k) = \frac{1}{1-x}\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j, \quad x<0. \qquad \mathbf{(2.3)}$$

Pertvarkius (2.3) lygybę, gauname:

$$G(x,k) = 1 - \left(\frac{x}{x-1}\right)^{k-1}, \quad x<0.$$

Teorema įrodyta.

Šios funkcijos grafikas, esant skirtingiems parametro k reikšmėms, pateiktas 2.8 paveiksle.



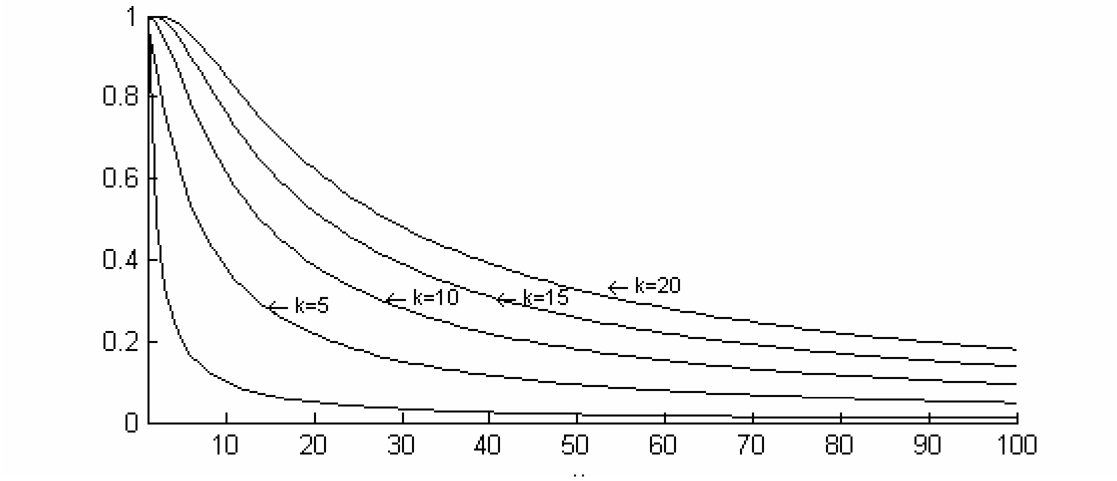
2.8 pav. Ribinė maksimumų sandaugos pasiskirstymo funkcija

2.2.2. KONVERGAVIMO GREITIS

Empiriškai įvertiname Absoliučiąsias paklaidas:

$$\Delta_2(x,n,k) = \left| G(x,k) - P\left(V_{N_n}^{(k)} < 1 + \frac{x}{n} \right) \right|.$$

Paklaidą įvertiname skaičiuodami maksimalų atstumą tarp ribinio skirstinio ir maksimumų sandaugos pasiskirstymo funkcijos, kai imčių didumai yra geometrinis atsitiktinis dydis su parametru $p = \frac{1}{n}$ (n kinta nuo 1 iki 100).



2.9 pav. Eksperimentinis paklaidos įvertinimas

Iš grafikų pavidalo spėjama, kad konvergavimo greičio eilė n atžvilgiu bus artima $\frac{1}{n}$, todėl palyginimui yra nubrėžta ir ši funkcija.

Ieškome konvergavimo greičio įverčio analitiškai.

2.5 teorema. $\Delta_2(x,n,k) = \dfrac{x^2 G(x,k)}{n + x - nx}$, $-n < x < 0.$

Įrodymas.

$$\Delta_2(x,n,k) = \left| G(x,k) - P\left(V_{N_n}^{(k)} < 1 + \frac{1}{n} \right) \right|$$

$$\Delta_2(x,n,k) = \left| \frac{1}{1-x} \sum_{j=0}^{k-1} \left(\frac{x}{x-1} \right)^j - \frac{n+x}{n^2} \sum_{j=0}^{k-1} \frac{\left(-\ln\left(1 + \frac{x}{n}\right)\right)^j}{j!} \sum_{m=1}^{\infty} m^j \left(\left(\frac{n+x}{n} \right) \left(\frac{n-1}{n} \right) \right)^{m-1} \right|, -n < x < 0$$

Nelygybė $G(x,k) \geq P\left(V_{N_n}^{(k)} < 1 + \frac{1}{n} \right)$ teisinga, esant bet kuriems $n,k \in N$, $x \in (-n;0)$, todėl modulio ženklas nėra būtinas. Panaudojus nelygybę $-\frac{x}{n} \leq -\ln\left(1 + \frac{x}{n}\right)$ konvergavimo greičio įvertį užrašome taip:

$$\Delta_2(x,n,k) = \frac{1}{1-x} \sum_{j=0}^{k-1} \left(\frac{x}{x-1} \right)^j - \frac{n+x}{n^2} \sum_{j=0}^{k-1} \frac{\left(-\ln\left(1 + \frac{x}{n}\right)\right)^j}{j!} \sum_{m=1}^{\infty} m^j \left(\left(\frac{n+x}{n} \right) \left(\frac{n-1}{n} \right) \right)^{m-1} \leq$$

$$\leq \frac{1}{1-x} \sum_{j=0}^{k-1} \left(\frac{x}{x-1} \right)^j - \frac{n+x}{n^2} \sum_{j=0}^{k-1} \frac{(-x)^j}{j! n^j} \sum_{m=1}^{\infty} m^j \left(\left(\frac{n+x}{n} \right) \left(\frac{n-1}{n} \right) \right)^{m-1}$$

Modeliuojant kompiuteriu buvo pastebėta, kad

$$\sum_{j=0}^{k-1}\frac{(-x)^j}{j!n^j}\sum_{m=1}^{\infty}m^j\left(\left(\frac{n+x}{n}\right)\left(\frac{n-1}{n}\right)\right)^{m-1}\geq\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j\sum_{m=1}^{\infty}\left(\left(\frac{n+x}{n}\right)\left(\frac{n-1}{n}\right)\right)^{m-1}.$$

Todėl

$$\Delta_2(x,n,k)\leq\frac{1}{1-x}\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j-\frac{n+x}{n^2}\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j\sum_{m=1}^{\infty}\left(\left(1+\frac{x}{n}\right)\left(1-\frac{1}{n}\right)\right)^{m-1}=$$

$$=\frac{1}{1-x}\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j-\frac{n+x}{n^2}\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j\sum_{m=1}^{\infty}\left(1-\frac{n+x-nx}{n^2}\right)^{m-1}=$$

$$=\frac{1}{1-x}\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j-\frac{n+x}{n^2}\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j\frac{n^2}{n+x-nx}=\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j\left(\frac{1}{1-x}-\frac{n+x}{n+x-nx}\right)=$$

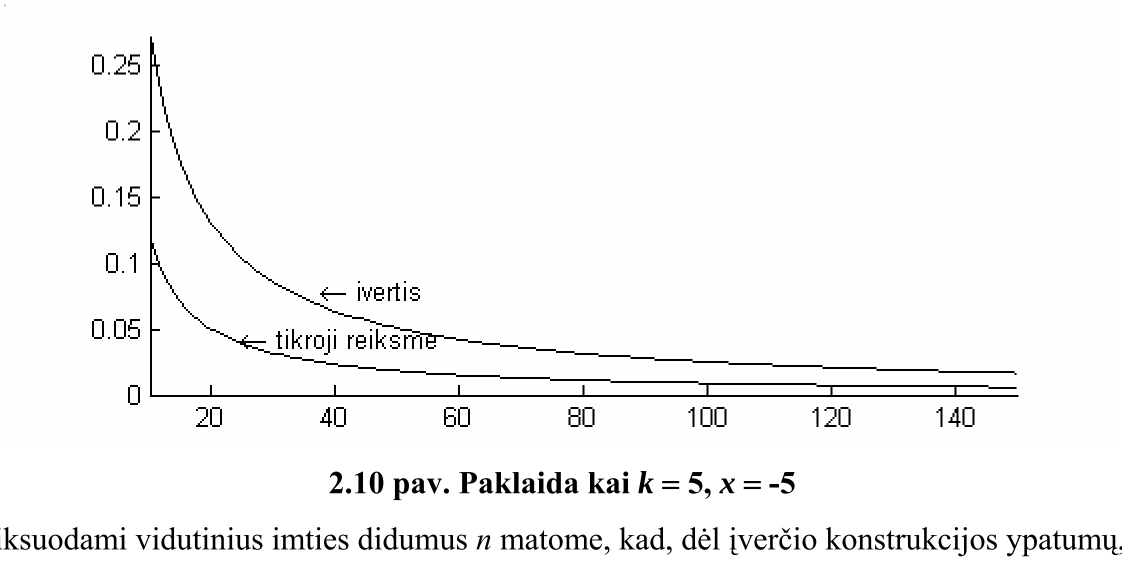
$$=\frac{x^2}{(1-x)(n+x-nx)}\sum_{j=0}^{k-1}\left(\frac{x}{x-1}\right)^j=\frac{x^2G(x,k)}{n+x-nx}.$$

Teorema įrodyta.

Gautas konvergavimo greičio įvertis $\Delta_2(x,n,k)=\dfrac{x^2G(x,k)}{n+x-nx}$. Matome, kad konvergavimo greičio eilė n atžvilgiu yra artima $\dfrac{1}{n}$. Šį rezultatą patvirtina ir aukščiau pateikti modeliavimo rezultatai.

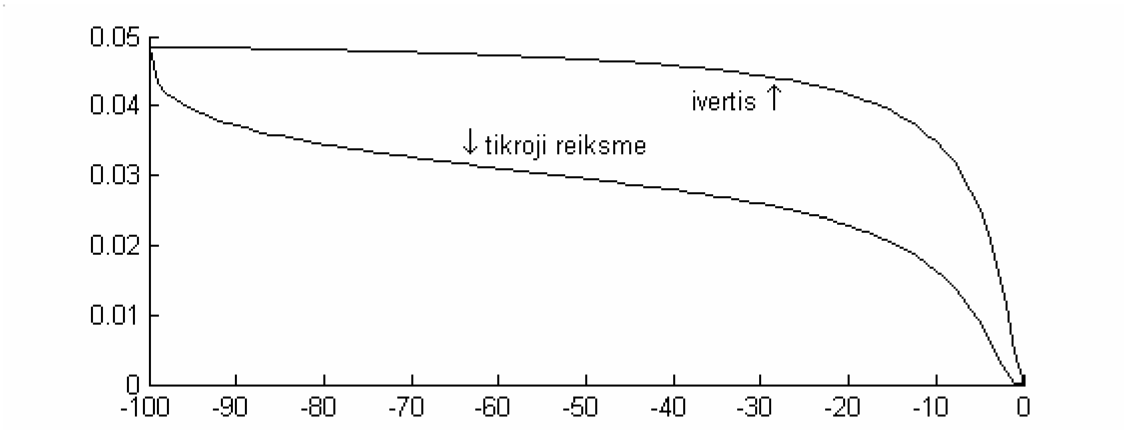
Bendru atveju gautas įvertis yra trijų kintamųjų funkcija. Todėl toliau nagrinėsime šios funkcijos projekcijas į tam tikras plokštumas.

Fiksuodami konkrečią x reikšmę bei keisdami parametrą p (t.y. keisdami vidutinius imčių didumus) matome kad didėjant n skirtumas tarp paklaidos tikrosios reikšmės ir įverčio nyksta, kaip to ir reikėjo tikėtis (2.10 paveikslas).



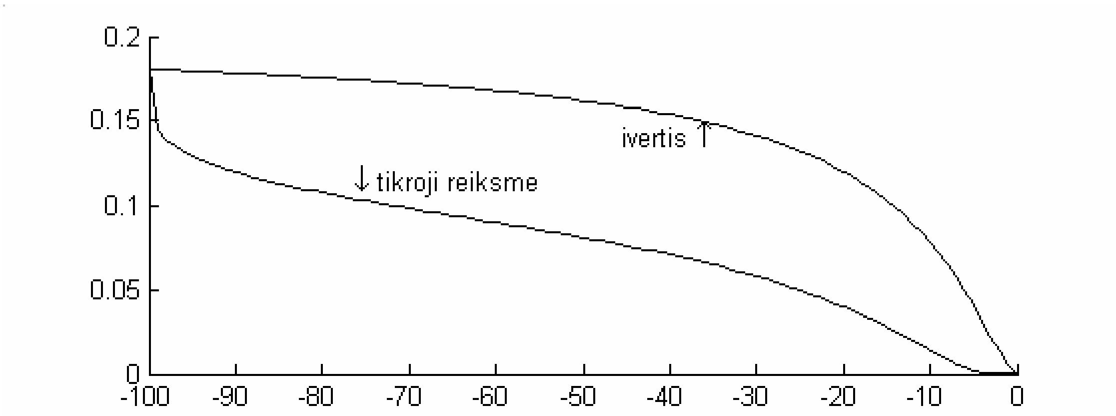
2.10 pav. Paklaida kai k = 5, x = -5

Fiksuodami vidutinius imties didumus n matome, kad, dėl įverčio konstrukcijos ypatumų, x kitimo intervalo rėžiuose įverčio reikšmė sutampa su tikrosios paklaidos reikšme (2.11 paveikslas).



2.11 pav. Paklaida kai *k* = 5; *n* = 100

Keisdami k reikšmes fiksuotam n matome, kad, priešingai nei fiksuoto imčių didumo atveju, gautasis įvertis visame x kitimo intervale išlieka sąlyginai artimas tikrajai paklaidos reikšmei (2.12 paveikslas). Todėl galime teigti, kad gautas įvertis yra tinkamas bet kurioms k ir n kombinacijos



2.12 pav. Paklaida kai *k* = 20, *n* = 100

2.3. DARBO SANTRAUKA

1. Rastas maksimumų sandaugos pasiskirstymo dėsnis, kai imčių didumai yra iš anksto žinomas dydis: $P\left(V_n^{(k)} < x\right) = x^n \cdot \sum_{j=0}^{k-1} \frac{(-n \cdot \ln x)^j}{j!}$;

2. Rastas neišsigimęs maksimumų sandaugos ribinis skirstinys (kai imčių didumai neaprėžtai didėja). Parinktos centravimo ir normavimo konstantos:

$$H(x,k) = \lim_{n\to\infty} F_{V_n^{(k)}}(a_n + b_n x) = e^x \cdot \sum_{j=0}^{k-1} \frac{(-x)^j}{j!},\ a_n = 1, b_n = \frac{1}{n};$$

3. Rastas konvergavimo į ribinį skirstinį greičio įvertis

$$\Delta_1(x,n,k) \le \frac{x^2 e^x}{2n} \sum_{j=0}^{k-1} \frac{(-x)^j}{j!},\ -n < x < 0;$$

4. Rastas maksimumų sandaugos pasiskirstymo dėsnis, kai imčių didumai yra geometrinis atsitiktinis dydis su parametru $p = \frac{1}{n}$:

$$P\left(V_{N_n}^{(k)} < x\right) = \frac{x}{n} \sum_{j=0}^{k-1} \frac{(-\ln x)^j}{j!} \sum_{m=1}^{\infty} m^j \left(x\left(\frac{n-1}{n}\right)\right)^{m-1};$$

5. Suformuluota ir įrodyta perkelimo teorema:

$$\lim_{n\to\infty} P\left(V_{N_n}^{(k)} < a_n + b_n x\right) = \int_0^{\infty} H(xz,k)\,dA(z),\ a_n = 1, b_n = \frac{1}{n}$$

6. Rastas neišsigimęs maksimumų sandaugos ribinis skirstinys, kai imčių didumai yra geometrinis atsitiktinis dydis:

$$G(x,k) = 1 - \left(\frac{x}{x-1}\right)^{k-1},\ x < 0;$$

7. Rastas konvergavimo į ribinį skirstinį greičio įvertis:

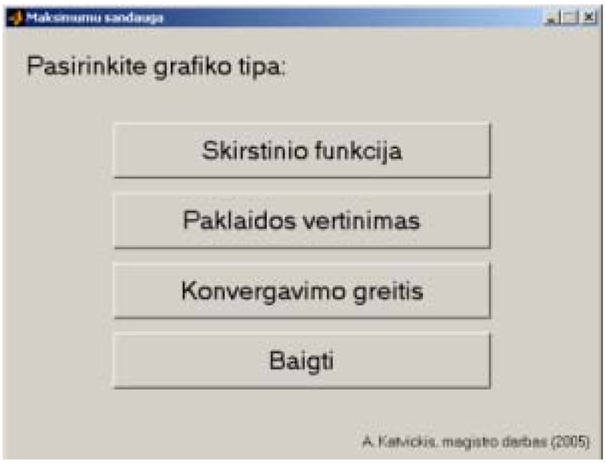
$$\Delta_2(x,n,k) = \frac{x^2 G(x,k)}{n + x - nx}.$$

3. PROGRAMINĖ REALIZACIJA IR INSTRUKCIJA VARTOTOJUI

MatLab 6 terpėje sukurta programa, pateikianti darbe gautus rezultatus grafikais. Matematinis paketas MatLab pasirinktas dėl plačių grafinių galimybių. Viena iš jų – galimybė sukurti grafinę vartotojo sąsają (Graphical User Interface (GUI) application).

Programa sukurta naudojantis standartinę MatLab standartine grafinės vartotojo sąsajos kūrimo terpe GUIDE (Graphical User Interface Development Environment).

Paleidus programa, ekrane pasirodo pagrindinis meniu (3.1 paveikslas).

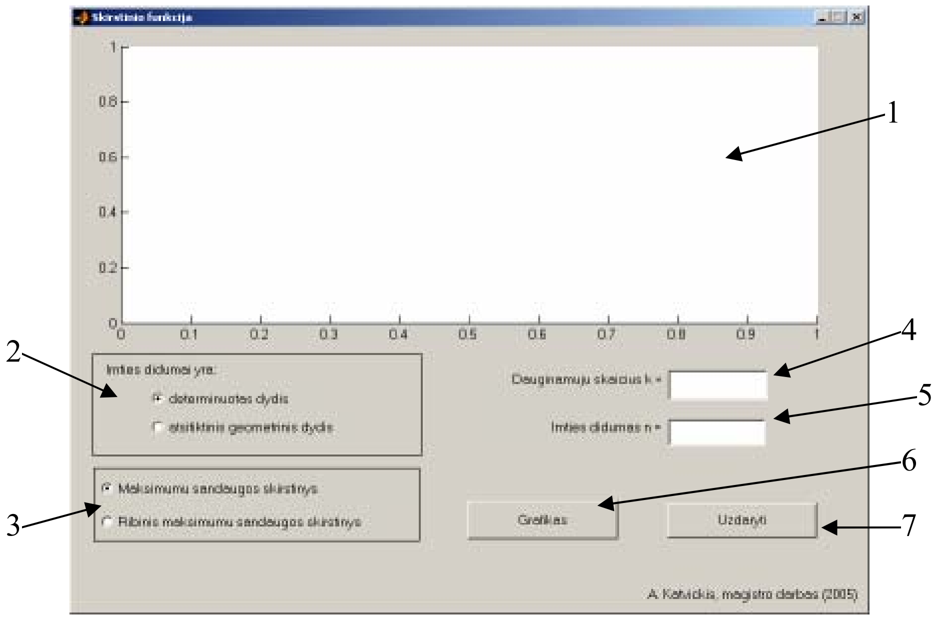


3.1 pav. Pagrindinis meniu

Pagrindiniame meniu galimi pasirinkimai:

- Skirstinio funkcija – braižoma maksimumų sandaugos skirstinio funkcija arba ribinės skirstinio funkcija;
- Paklaidos vertinimas – braižomas absoliučiosios paklaidos grafikas;
- Konvergavimo greitis – braižomas darbe gautas paklaidos įvertis bei tikroji paklaidos reikšmė, esant vienodoms parametrų reikšmėms;
- Baigti – programos darbo pabaiga.

Pasirinkę „Skirstinio funkcija“ ekrane atsiranda naujas langas (3.2 paveikslas).



3.2 pav. Skirstinio funkcijos braižymo langas

Skaičiais pažymėti objektai ir juose atliekami veiksmai:

1 – Pasirinktos skirstinio funkcijos grafikas.

2 – Pasirenkamas imties didumo tipas.

3 – Pasirenkamas skirstinio funkcijos tipas.

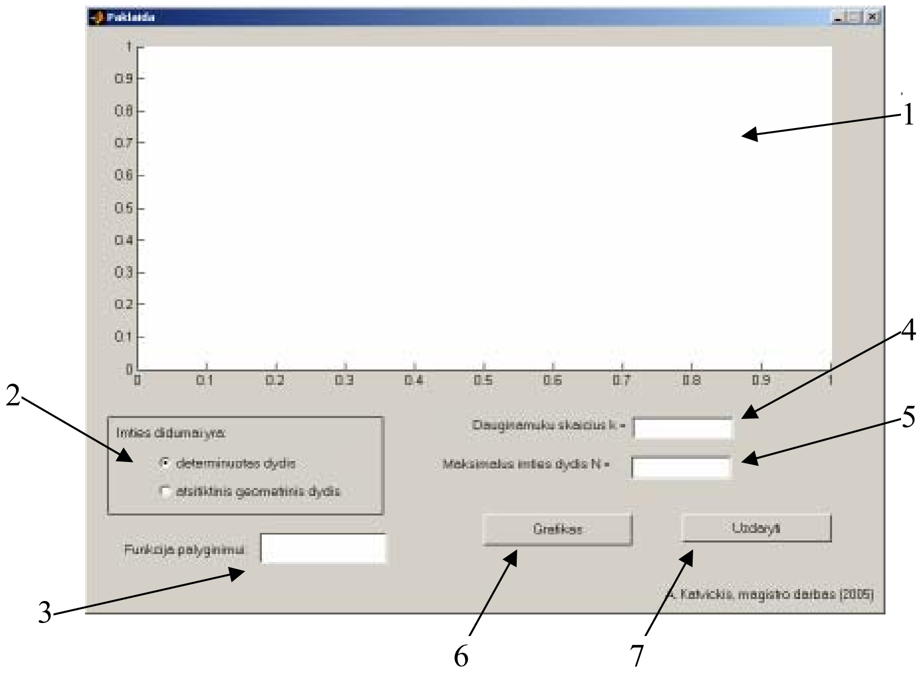
4 – Pasirenkamas maksimumų sandaugoje kiekis.

5 – Pasirenkamas imties didumas (imties didumas determinuotas dydis) arba vidutinis imties didumas (imties didumas geometrinis atsitiktinis dydis).

6 – Ištrina esamą grafiką (jei toks yra). Patikrina, ar nustatytos 2-5 laukuose parametrų reikšmės yra galimos, ir braižo naują grafiką. Jei nustatytos parametrų reikšmės yra neleistinos (pvz. neigiamas imties didumas), grafiko braižymas yra atšaukiamas ir vartotojas gauna pranešimą apie klaidos pobūdį.

7 – Uždaro šį langą.

Pagrindiniame meniu pasirinkę „Paklaidos vertinimas“ ekrane atsiranda naujas langas (3.3 paveikslas).



3.3 pav. Paklaidos vertinimo langas

Skaičiais pažymėti objektai ir juose atliekami veiksmai:

1 – Paklaidos grafikas.

2 – Pasirenkamas imties didumo tipas.

3 – Galima nurodyti funkciją, su kuria bus lyginamas gautas paklaidos grafikas. Jei funkcija nenurodyta, šis laukas ignoruojamas.

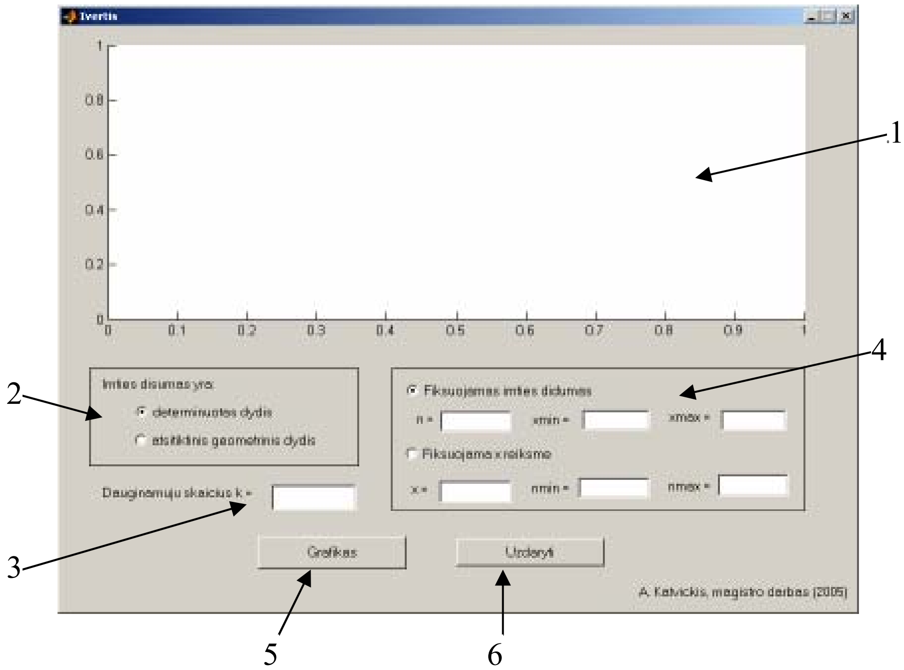
4 – Pasirenkamas maksimumų sandaugoje kiekis.

5 – Pasirenkamas maksimalus imties didumas N_{max} (paklaidos grafikas gaunamas keičiant imties didumą nuo 1 iki N_{max}).

6 – Ištrina esamą grafiką (jei toks yra). Patikrina, ar nustatytos 2-5 laukuose parametrų reikšmės yra galimos, ir braižo naują grafiką. Jei nustatytos parametrų reikšmės yra neleistinos (pvz. neigiamas imties didumas), grafiko braižymas yra atšaukiamas ir vartotojas gauna pranešimą apie klaidos pobūdį.

7 – Uždaro šį langą.

Pagrindiniame meniu pasirinkę „Konvergavimo greitis“ ekrane atsiranda naujas langas (3.4 paveikslas).



3.4 pav. Konvergavimo greičio įverčio braižymo langas

Skaičiais pažymėti tokie objektai ir juose atliekami veiksmai:

1 – Pasirinktos skirstinio funkcijos grafikas.

2 – Pasirenkamas imties didumo tipas.

3 – Pasirenkamas maksimumų sandaugoje kiekis.

4 – Pasirenkamas grafiko tipas. Galimi du variantai: 1) imties didumas n yra fiksuotas, x kinta nuo x_{min} iki x_{max}; 2) fiksuotas x, imties didumai kinta nuo n_{min} iki n_{max}.

5 – Ištrina esamą grafiką (jei toks yra). Patikrina, ar nustatytos 2-4 laukuose parametrų reikšmės yra galimos, ir braižo naują grafiką. Jei nustatytos parametrų reikšmės yra neleistinos (pvz. neigiamas imties didumas), grafiko braižymas yra atšaukiamas ir vartotojas gauna pranešimą apie klaidos pobūdį.

6 – Uždaro šį langą.

IŠVADOS

1. Kai imties didumas yra neatsitiktinis dydis *n*, konvergavimo greičio eilė *n* atžvilgiu yra $\frac{1}{n}$.

2. Įvertis $\Delta_1(x,n,k) \le \frac{x^2 e^x}{2n}\sum_{j=0}^{k-1}\frac{(-x)^j}{j!}, \ -n < x < 0$ yra taikytinas, kai dauginamųjų maksimumų skaičius $k << n$

3. Perkelimo teoremoje, kai imties didumas yra atsitiktinis geometrinis dydis su parametru $p = \frac{1}{n}$, konvergavimo greitis n atžvilgiu taip pat yra eilės $\frac{1}{n}$.

4. Įvertis $\Delta_2(x,n,k) = \frac{x^2 G(x,k)}{n + x - nx}$, skirtingai nei $\Delta_1(x,n,k)$, yra taikytinas maksimumų sandaugoms su bet kuriomis parametrų *k* ir *n* reikšmėmis.

5. Maksimumų sandaugos skirstinių asimptotinė analizė gali būti atlikta ir ne charakteristinių funkcijų metodu, bet tiesioginiu būdu (Priedas 2).

LITERATŪROS SARAŠAS

1. Галамбош Я (1984). Асимптотическая теория экстремальных порядковых статистик. Москва: Наука.

2. R. Vilkas, A. Aksomaitis (2001). Konvergavimo greičio radimo procedūra, *Lietuvos matematikos rinkinys, T. 41*, Vilnius, 485-490.

3. A. Aksomaitis, R. Vilkas (1998). Netiesinių stochastinių maksimumų struktūros, *LMD mokslo darbai*, Vilnius, 345-348.

4. A. Aksomaitis (2000). Tikimybių teorija ir statistika. Kaunas: Technologija.

5. Тихов М. С. (1997). Статистический анализ моделей высот интенсивных внутренних волн, *Обозрение прикладной и промышленной математики*, Т.4, вып.3, ТВП, 412-414.

Priedas 1. Spausdinimui pateiktas straipsnis

Imkime struktūrą (k maksimumų sandauga):

$$V_n^{(k)} = Z_n^{(1)} \cdot Z_n^{(2)} \cdot \ldots \cdot Z_n^{(k)};$$

čia

$Z_n^{(j)} = \max\left(X_1^{(j)}, X_2^{(j)}, \ldots, X_n^{(j)}\right), j = \overline{1,k}$, $(X_1^{(j)}, X_2^{(j)}, \ldots, X_n^{(j)})$ - paprastoji atsitiktinė imtis iš tolygiosios generalinės aibės $(X \sim T(0,1))$. Imtys nepriklausomos $(j = \overline{1,k})$.

Atsitiktinių dydžių maksimumų sandauga, kai $k = 2$ yra nagrinėta [2], [3] darbuose. Šiame darbe mes pateikiame tyrimus atveju, kai $k \geq 2$.

1 teorema. Dydžio $V_n^{(k)}$ pasiskirstymo tankis

$$p_{V_n^{(k)}}(x) = \frac{n^k}{(k-1)!} x^{n-1} \left(-\ln x\right)^{k-1}, \text{ čia } 0 < x < 1.$$

Įrodymas. Dydžio $V_n^{(k)}$ tankį ir pasiskirstymo funkciją rasime pasinaudodami charakteristinių funkcijų savybėmis.

Žinoma, kad

$$P(Z_n^{(j)} < x) = x^n, 0 < x < 1, j = \overline{1,k}$$

Pažymėję $W_n^{(k)} = \ln V_n^{(k)}, T_j = \ln Z_n^{(j)}, j = \overline{1,k}$ gauname:

$$W_n^{(k)} = \sum_{j=1}^{k} T_j$$

Perrašę šią lygybę charakteristinių funkcijų kalboje turime:

$$f_{W_n^{(k)}}(t) = \prod_{j=1}^{k} f_{T_j}(t)$$

Parodėme, kad

$$f_{T_j}(t) = \frac{n}{it+n}, j = \overline{1,k}.$$

Tada

$$f_{W_n^{(k)}}(t) = \left(\frac{n}{it+n}\right)^k$$

Pasinaudoję atgręžimo formule $p(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-itx} f(t)dt$, gauname dydžio $W_n^{(k)}$ tankį ir pasiskirstymo funkciją.:

$$p_{W_n^{(k)}}(x) = \frac{n^k}{(k-1)!} e^{nx} (-x)^{k-1},$$

$$F_{W_n^{(k)}}(x) = e^{nx} \sum_{j=0}^{k-1} \frac{(-nx)^j}{j!}, \ w < 0.$$

Kadangi $W_n^{(k)} = \ln V_n^{(k)}$, tai $F_{V_n^{(k)}}(x) = F_{W_n^{(k)}}(\ln x)$, t.y.

$$p_{V_n^{(k)}}(x) = \frac{n^k}{(k-1)!} x^{n-1} (-\ln x)^{k-1}, \ 0 < x < 1.$$

Teorema įrodyta.

Rasime ribinę maksimumų sandaugos pasiskirstymo funkciją

$$H_{V^{(k)}}(x) = \lim_{n \to \infty} F_{V_n^{(k)}}(a_n + b_n x);$$

čia a_n ir b_n - centravimo ir normavimo konstantos.

Pasirenkame $a_n = 1$ ir $b_n = \frac{1}{n}$. Tuomet :

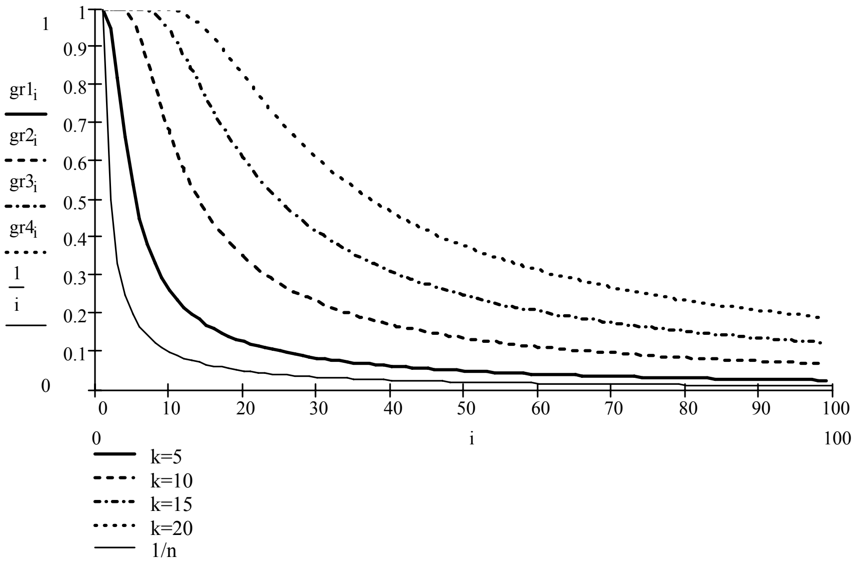
$$H_{V^{(k)}}(x) = \lim_{n \to \infty} F_{V_n^{(k)}}\left(1 + \frac{x}{n}\right) = \lim_{n \to \infty} \left(1 + \frac{x}{n}\right)^n \sum_{j=0}^{k-1} \frac{\left(-n \ln\left(1 + \frac{x}{n}\right)\right)^j}{j!},$$

$$H_{V^{(k)}}(x) = e^x \cdot \sum_{j=0}^{k-1} \frac{(-x)^j}{j!}, \ x < 0.$$

Kad žinotume, kiek ribinė pasiskirstymo funkcija atitinka maksimumų sandaugos skirstinį, įvertinsime paklaidą:

$$\Delta(x) = \left| H_{V^{(k)}}(x) - F_{V_n^{(k)}}(a_n + b_n x) \right|.$$

Šiame darbe konvergavimo greitis įvertintas eksperimentiškai, t.y. skaičiuojant maksimalų atstumą tarp ribinio skirstinio ir maksimumų sandaugos pasiskirstymo funkcijos, esant skirtingiems imties tūriams n ir dauginamųjų maksimumų skaičiams k. Grafikai gauti imant $n = \overline{1,100}$ ir $k = 5,10,15,20$.



Iš grafikų pavidalo spėjama, kad konvergavimo greičio eilė n atžvilgiu bus $\frac{1}{n}$, todėl palyginimui yra nubrėžta ir ši funkcija.

Iki šiol buvo nagrinėjamas atvejis, kai imčių $\left(X_1^{(j)}, X_2^{(j)}, \ldots, X_n^{(j)}\right)$, $j = \overline{1,k}$ didumai n yra iš anksto žinomos konstantos. Dabar nagrinėsime atvejį, kai imčių tūriai yra atsitiktinis dydis N_n, pasiskirstęs pagal geometrinį dėsnį, t.y. $P(N_n = m) = p \cdot (1-p)^{m-1}$, čia $p = \frac{1}{n}$.

Rasime atsitiktinio dydžio $V_{N_n}^{(k)}$ pasiskirstymo funkciją:

$$P\left(V_{N_n}^{(k)} < x\right) = \sum_{m=1}^{\infty} P(N_n = m) \cdot P\left(V_{N_n}^{(k)} < x \mid N_n = m\right) =$$

$$= \sum_{m=1}^{\infty} P(N_n = m) \cdot P\left(V_m^{(k)} < x\right).$$

Gauname, kad

$$P\left(V_{N_n}^{(k)} < x\right) = \frac{x}{n} \sum_{i=0}^{k-1} \frac{(-\ln x)^i}{i!} \sum_{m=1}^{\infty} m^i \left(x\left(\frac{n-1}{n}\right)\right)^{m-1}.$$

Šis rezultatas nėra vaizdus ir yra sunkiai pritaikomas skaičiavimuose. Todėl suformuluosime ir įrodysime perkėlimo teoremą.

2 teorema. $\lim_{n\to\infty} P\left(V_{N_n}^{(k)} < a_n + b_n x\right) = \int_0^{\infty} H(xz)\,dA(z)$.

Čia $x < 0$, *H(x)*, ir *A(x)* tokie kad $P\left(V_n^{(k)} < a_n + b_n x\right) \Rightarrow H(x)$ ir $P\left(\frac{N_n}{n} < z\right) \Rightarrow A(z)$, kai $n \to \infty$.

Normavimo ir centravimo konstantos gali būti parinktos $a_n = 1$ ir $b_n = \frac{1}{n}$.

Įrodymas.

$$P\left(V_{N_n}^{(k)}<1+\frac{v}{n}\right)=\sum_{m=1}^{\infty}P(N_n=m)\cdot P\left(V_m^{(k)}<1+\frac{v}{n}\right)=$$

$$=\sum_{m=1}^{\infty}P(N_n=m)\cdot\left(1+\frac{v}{n}\right)^m\cdot\sum_{i=0}^{k-1}\frac{\left(-m\cdot\ln\left(1+\frac{v}{n}\right)\right)^i}{i!}=$$

$$=\sum_{m=1}^{\infty}P\left(\frac{N_n}{n}=\frac{m}{n}\right)\cdot\left(1+\frac{v}{n}\right)^{n\cdot\frac{m}{n}}\cdot\sum_{i=0}^{k-1}\frac{\left(-n\cdot\frac{m}{n}\cdot\ln\left(1+\frac{v}{n}\right)\right)^i}{i!}$$

Įvedame keitinį $\frac{m}{n}=z$ ir skaičiuojame šio reiškinio ribą, kai $n\to\infty$. Gauname:

$$P\left(V_{N_n}^{(k)}<1+\frac{v}{n}\right)\xrightarrow[n\to\infty]{}\int_0^{\infty}e^{v\cdot z}\sum_{i=0}^{k-1}\frac{(-v\cdot z)^i}{i!}dA(z)=\int_0^{\infty}H(v\cdot z)dA(z).$$

Teorema įrodyta.

Pritaikykime įrodytą teoremą nagrinėjamam atvejui.

Žinoma kad, $P\left(\frac{N_n}{n}<z\right)\Rightarrow A(z)=1-e^{-x}$ [1], $x>0$, jei atsitiktinis dydis N_n yra pasiskirstęs pagal geometrinį dėsnį su parametru $p=\frac{1}{n}$.

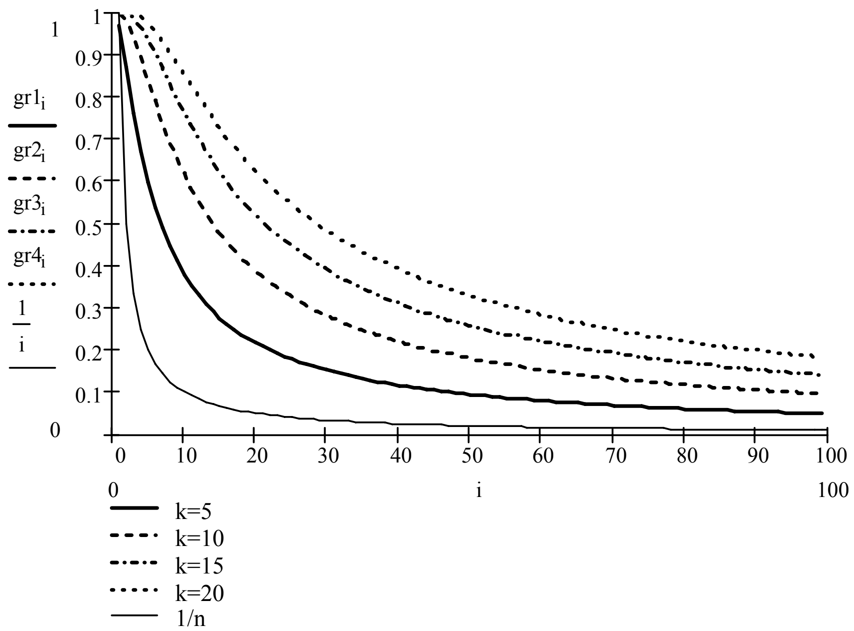
Pažymėkime ribinę funkciją $G(v)$.

$$G(v)=\int_0^{\infty}H(v\cdot z)dA(z)=\int_0^{\infty}e^{v\cdot z}\sum_{i=0}^{k-1}\frac{(-v\cdot z)^i}{i!}\cdot e^{-z}dz=1-\left(\frac{v}{v-1}\right)^k,\ v<0$$

Empiriškai įvertiname konvergavimo greitį

$$\Delta(v)=\left|G(v)-P\left(V_{N_n}^{(k)}<1+\frac{v}{n}\right)\right|$$

Konvergavimo greitį įvertiname skaičiuojant maksimalų atstumą tarp ribinio skirstinio ir maksimumų sandaugos pasiskirstymo funkcijos, kai imčių didumai yra geometrinis atsitiktinis dydis su parametru $p=\frac{1}{n}$ (n kinta nuo 1 iki 100).



Iš grafikų pavidalo spėjama, kad konvergavimo greičio eilė n atžvilgiu bus artima $\frac{1}{n}$, todėl palyginimui yra nubrėžta ir ši funkcija.

Priedas 2. Kitas 2.1 teoremos įrodymas

Įrodysime matematinės indukcijos metodu.

Kai $k=1$ gauname:

$$P\left(V_n^{(1)}<x\right)=P\left(Z_n^{(1)}<x\right)=P\left(X_1^{(1)}<x,\ldots,X_n^{(1)}<x\right)=P^n\left(X_1^{(1)}<x\right)=x^n$$

$$p_{V_n^{(1)}}(x)=\frac{d}{dv}P\left(V_n^{(1)}<x\right)=n\cdot x^{n-1},\ 0<x<1.$$

Kai $k=2$:

$$P\left(V_n^{(2)}<x\right)=P\left(Z_n^{(1)}\cdot Z_n^{(2)}<x\right).$$

Kadangi dydžių $Z_n^{(j)}$, $j=1,2$ pasiskirstymo funkcijos ir tankiai yra žinomi, tai dydžio $V_n^{(2)}$ pasiskirstymo funkcija išreiškiama dvilypių integralu:

$P\left(V_n^{(2)}<v\right)=\iint\limits_D p_{Z_n^{(1)}}(x)\cdot p_{Z_n^{(2)}}(y)dxdy$, čia D - sritis, ribojama kreivės $v=x\cdot y$ ir nelygybėmis $0<x<1$, $0<y<1$. Suintegravus, gauname

$$P\left(V_n^{(2)}<v\right)=v^n-n\cdot v^n\cdot\ln v.$$

$$p_{V_n^{(2)}}(v)=\frac{d}{dv}P\left(V_n^{(2)}<v\right)=-n^2\cdot v^{n-1}\cdot\ln v,\ 0<v<1.$$

Kai $k=m-1$:

$$p_{V_n^{(m-1)}}(u)=\frac{n^{m-1}}{(m-2)!}u^{n-1}(-\ln u)^{m-2}.$$

Įrodome, kad duotoji lygybė yra teisinga kai $k=m$:

$$P\left(V_n^{(m)}<v\right)=P\left(Z_n^{(1)}\cdot\ldots\cdot Z_n^{(m-1)}\cdot Z_n^{(m)}\right)=P\left(V_n^{(m-1)}\cdot Z_n^{(m)}<v\right),$$

$$P\left(V_n^{(m)}<v\right)=\iint\limits_D p_{V_n^{(m-1)}}(x)\cdot p_{Z_n^{(m)}}(y)dxdy.$$

Suintegravus, gauname:

$$P\left(V_n^{(m)}<v\right)=v^n\cdot\sum_{i=0}^{m-1}\frac{(-n\cdot\ln v)^i}{i!}.$$

$$p_{V_n^{(m)}}(v)=\frac{d}{dv}P\left(V_n^{(m)}<v\right)=\frac{n^m}{(m-1)!}\cdot v^{n-1}\cdot(-\ln v)^{m-1},\ 0<v<1.$$

Teorema įrodyta.

Priedas 3. Programinė realizacija MatLab terpėje

***sandauga.m**:*

```
function varargout = sandauga(varargin)
if nargin == 0
      fig = openfig(mfilename,'reuse');
      set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
      handles = guihandles(fig);
      guidata(fig, handles);
      if nargout > 0
            varargout{1} = fig;
      end
elseif ischar(varargin{1})
      try
            [varargout{1:nargout}] = feval(varargin{:});
      catch
            disp(lasterr);
      end
end

% --------------------------------------------------------------------
function varargout = start_Callback(h, eventdata, handles, varargin)
close(handles.figure1);
meniu;
```

meniu.m:

```
function varargout = meniu(varargin)
if nargin == 0
      fig = openfig(mfilename,'reuse');
      set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
      handles = guihandles(fig);
      guidata(fig, handles);
      if nargout > 0
            varargout{1} = fig;
      end
elseif ischar(varargin{1})
      try
            [varargout{1:nargout}] = feval(varargin{:});
      catch
            disp(lasterr);
      end
end

% --------------------------------------------------------------------
function varargout = funkcija_Callback(h, eventdata, handles, varargin)
funkcija;

% --------------------------------------------------------------------
function varargout = paklaida_Callback(h, eventdata, handles, varargin)
paklaida;

% --------------------------------------------------------------------
function varargout = ivertis_Callback(h, eventdata, handles, varargin)
ivertis;

% --------------------------------------------------------------------
function varargout = baigti_Callback(h, eventdata, handles, varargin)
close(handles.figure1);
```

funkcija.m:

```
function varargout = funkcija(varargin)
if nargin == 0
      fig = openfig(mfilename,'reuse');
      set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
      handles = guihandles(fig);
      guidata(fig, handles);
      if nargout > 0
            varargout{1} = fig;
      end
elseif ischar(varargin{1})
      try
            [varargout{1:nargout}] = feval(varargin{:});
      catch
            disp(lasterr);
      end
end

% --------------------------------------------------------------------
function varargout = fiksuotas_Callback(h, eventdata, handles, varargin)
set(handles.atsitiktinis, 'Value', 0);

% --------------------------------------------------------------------
function varargout = atsitiktinis_Callback(h, eventdata, handles, varargin)
set(handles.fiksuotas, 'Value', 0);

% --------------------------------------------------------------------
function varargout = paprastas_Callback(h, eventdata, handles, varargin)
set(handles.ribinis, 'Value', 0);

% --------------------------------------------------------------------
function varargout = ribinis_Callback(h, eventdata, handles, varargin)
set(handles.paprastas, 'Value', 0);

% --------------------------------------------------------------------
function varargout = braizyti_Callback(h, eventdata, handles, varargin)
cla;
klaida = 0;
k = str2num(get(handles.k, 'String'));
if isempty(k)
    set(handles.k, 'String', '');
    msgbox('k turi buti sveikas teigiamas skaicius', 'klaida', 'error');
    klaida = 1;
else
    if (k <= 0)|(round(k)~=k)
        set(handles.k, 'String', '');
        msgbox('k turi buti sveikas teigiamas skaicius', 'klaida', 'error');
        klaida = 1;
    end
end
n = str2num(get(handles.n, 'String'));
if isempty(n)
    set(handles.n, 'String', '');
    msgbox('n turi buti sveikas teigiamas skaicius', 'klaida', 'error');
    klaida = 1;
else
    if (n <= 0)|(round(n)~=n)
        set(handles.n, 'String', '');
        msgbox('n turi buti sveikas teigiamas skaicius', 'klaida', 'error');
        klaida = 1;
    end
end
```

```
poz = 2*get(handles.fiksuotas, 'Value')+get(handles.paprastas, 'Value');
if (klaida==0)
    hh = waitbar(0, 'Palaukite...');
    if mod(poz,2)==0
        x=-n:n/1000:0;
        for i=1:1001
            if poz==0
                y(i) = g_a(x(i), k);
            else
                y(i) = f_a(x(i), k);
            end
            waitbar(i/1001);
        end
    else
        x=0:0.001:1;
        for i=1:1001
            if poz==1
                y(i) = g_t(x(i), 1/n, k);
            else
                y(i) = f_t(x(i), n, k);
            end
            waitbar(i/1001);
        end
    end
    close(hh);
    plot(x, y);
end

% --------------------------------------------------------------------
function varargout = baigti_Callback(h, eventdata, handles, varargin)
close(handles.figure1);

% --------------------------------------------------------------------
function varargout = k_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = n_Callback(h, eventdata, handles, varargin)
```

paklaida.m:

```
function varargout = paklaida(varargin)
if nargin == 0
      fig = openfig(mfilename,'reuse');
      set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
      handles = guihandles(fig);
      guidata(fig, handles);
      if nargout > 0
            varargout{1} = fig;
      end
elseif ischar(varargin{1})
      try
            [varargout{1:nargout}] = feval(varargin{:});
      catch
            disp(lasterr);
      end
end



% --------------------------------------------------------------------
function varargout = fiksuotas_Callback(h, eventdata, handles, varargin)
set(handles.atsitiktinis, 'Value', 0);
```

```
% --------------------------------------------------------------------
function varargout = atsitiktinis_Callback(h, eventdata, handles, varargin)
set(handles.fiksuotas, 'Value', 0);


% --------------------------------------------------------------------
function varargout = k_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = Nmax_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = brezti_Callback(h, eventdata, handles, varargin)
cla;
klaida = 0;
k = str2num(get(handles.k, 'String'));
if isempty(k)
    set(handles.k, 'String', '');
    msgbox('k turi buti sveikas teigiamas skaicius', 'klaida', 'error');
    klaida = 1;
else
    if (k <= 0)|(round(k)~=k)
        set(handles.k, 'String', '');
        msgbox('k turi buti sveikas teigiamas skaicius', 'klaida', 'error');
        klaida = 1;
    end
end
Nmax = str2num(get(handles.Nmax, 'String'));
if isempty(Nmax)
    set(handles.Nmax, 'String', '');
    msgbox('n turi buti sveikas teigiamas skaicius', 'klaida', 'error');
    klaida = 1;
else
    if (Nmax <= 0)|(round(Nmax)~=Nmax)
        set(handles.Nmax, 'String', '');
        msgbox('n turi buti sveikas teigiamas skaicius', 'klaida', 'error');
        klaida = 1;
    end
end
poz = get(handles.fiksuotas, 'Value');
if (klaida==0)
    hh = waitbar(0, 'Palaukite...');
    x = -Nmax:Nmax/1000:0;
    n = 1:Nmax;
    for i=1:Nmax
        if poz==0
            for j=1:1001
                y1(j) = abs(g_a(x(j), k) - g_t((1+x(j)/i), 1/i, k));
            end
            y(i) = max(y1);
        else
            for j=1:1001
                y1(j) = abs(f_a(x(j), k) - f_t((1+x(j)/i), i, k));
            end
            y(i) = max(y1);
        end
        waitbar(i/Nmax);
    end
    close(hh);
    hold on
    plot(n, y);
    fplot(get(handles.Fx, 'String'), [1 Nmax], 'r');
    hold off
end
```

```
% --------------------------------------------------------------------
function varargout = finish_Callback(h, eventdata, handles, varargin)
close(handles.figure1);

% --------------------------------------------------------------------
function varargout = Fx_Callback(h, eventdata, handles, varargin)
```

ivertis.m:

```
function varargout = ivertis(varargin)
if nargin == 0
      fig = openfig(mfilename,'reuse');
      set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
      handles = guihandles(fig);
      guidata(fig, handles);
      if nargout > 0
            varargout{1} = fig;
      end
elseif ischar(varargin{1})
      try
            [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
      catch
            disp(lasterr);
      end
end


% --------------------------------------------------------------------
function varargout = fiksuotas_Callback(h, eventdata, handles, varargin)
set(handles.atsitiktinis, 'Value', 0);

% --------------------------------------------------------------------
function varargout = atsitiktinis_Callback(h, eventdata, handles, varargin)
set(handles.fiksuotas, 'Value', 0);

% --------------------------------------------------------------------
function varargout = k_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = imtis_Callback(h, eventdata, handles, varargin)
set(handles.taskas, 'Value', 0);

% --------------------------------------------------------------------
function varargout = taskas_Callback(h, eventdata, handles, varargin)
set(handles.imtis, 'Value', 0);

% --------------------------------------------------------------------
function varargout = n_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = xmin_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = xmax_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = brezti_Callback(h, eventdata, handles, varargin)
cla;
klaida = 0;
k = str2num(get(handles.k, 'String'));
if isempty(k)
```

```
        set(handles.k, 'String', '');
        msgbox('k turi buti sveikas teigiamas skaicius', 'klaida', 'error');
        klaida = 1;
    else
        if (k <= 0)|(round(k)~=k)
            set(handles.k, 'String', '');
            msgbox('k turi buti sveikas teigiamas skaicius', 'klaida', 'error');
            klaida = 1;
        end
    end
    if get(handles.imtis, 'Value')==1
        n = str2num(get(handles.n, 'String'));
        if isempty(n)
            set(handles.n, 'String', '');
            msgbox('n turi buti sveikas teigiamas skaicius', 'klaida', 'error');
            klaida = 1;
        else
            if (n <= 0)|(round(n)~=n)
                set(handles.n, 'String', '');
                msgbox('n turi buti sveikas teigiamas skaicius', 'klaida', 'error');
                klaida = 1;
            end
        end
        xmin = str2num(get(handles.xmin, 'String'));
        if isempty(xmin)
            set(handles.xmin, 'String', '');
            msgbox('xmin turi buti neigiamas skaicius', 'klaida', 'error');
            klaida = 1;
        else
            if (xmin >= 0)
                set(handles.xmin, 'String', '');
                msgbox('xmin turi buti neigiamas skaicius', 'klaida', 'error');
                klaida = 1;
            end
        end
        xmax = str2num(get(handles.xmax, 'String'));
        if isempty(xmax)
            set(handles.xmax, 'String', '');
            msgbox('xmax turi buti neigiamas skaicius', 'klaida', 'error');
            klaida = 1;
        else
            if (xmax >= 0)
                set(handles.xmax, 'String', '');
                msgbox('xmax turi buti neigiamas skaicius', 'klaida', 'error');
                klaida = 1;
            end
        end
        if (klaida==0) & (xmin>=xmax)
            set(handles.xmin, 'String', '');
            set(handles.xmax, 'String', '');
            msgbox('xmax turi buti didesnis uz xmin', 'klaida', 'error');
            klaida = 1;
        end
        if klaida==0
            hh = waitbar(0, 'Palaukite...');
            x = xmin:(xmax-xmin)/1000:xmax;
            for i = 1:1001
                if get(handles.fiksuotas, 'Value')==1
                    y1(i) = abs(f_a(x(i), k) - f_t(1+x(i)/n, n, k));
                    y2(i) = f_iv(x(i), n, k);
                else
                    y1(i) = abs(g_a(x(i), k) - g_t(1+x(i)/n, 1/n, k));
                    y2(i) = g_iv(x(i), n, k);
                end
```

```
                waitbar(i/1001);
            end
            hold on
            plot(x, y2)
            plot(x, y1)
            hold off
            close(hh);
        end
    else
        x = str2num(get(handles.x, 'String'));
        if isempty(x)
            set(handles.x, 'String', '');
            msgbox('x turi buti neigiamas skaicius', 'klaida', 'error');
            klaida = 1;
        else
            if (x >= 0)
                set(handles.x, 'String', '');
                msgbox('x turi buti neigiamas skaicius', 'klaida', 'error');
                klaida = 1;
            end
        end
        nmin = str2num(get(handles.nmin, 'String'));
        if isempty(nmin)
            set(handles.nmin, 'String', '');
            msgbox('nmin turi buti teigiamas sveikas skaicius', 'klaida', 'error');
            klaida = 1;
        else
            if (nmin <= 0)|(round(nmin)~=nmin)
                set(handles.nmin, 'String', '');
                msgbox('nmin turi buti teigiamas sveikas skaicius', 'klaida',
    'error');
                klaida = 1;
            end
        end
        nmax = str2num(get(handles.nmax, 'String'));
        if isempty(nmax)
            set(handles.nmax, 'String', '');
            msgbox('nmax turi buti teigiamas sveikas skaicius', 'klaida', 'error');
            klaida = 1;
        else
            if (nmax <= 0)|(round(nmax)~=nmax)
                set(handles.nmax, 'String', '');
                msgbox('nmax turi buti teigiamas sveikas skaicius', 'klaida',
    'error');
                klaida = 1;
            end
        end
        if (klaida==0) & (nmin>=nmax)
            set(handles.nmin, 'String', '');
            set(handles.nmax, 'String', '');
            msgbox('nmax turi buti didesnis uz nmin', 'klaida', 'error');
            klaida = 1;
        end
        if klaida==0
            hh = waitbar(0, 'Palaukite...');
            n = nmin:nmax;
            for i = 1:(nmax-nmin+1)
                if get(handles.fiksuotas, 'Value')==1
                    y1(i) = abs(f_a(x, k) - f_t(1+x/i, i, k));
                    y2(i) = f_iv(x, i, k);
                else
                    y1(i) = abs(g_a(x, k) - g_t(1+x/i, 1/i, k));
                    y2(i) = g_iv(x, i, k);
                end
```

```
            waitbar(i/(nmax-nmin+1));
        end
        hold on
        plot(n, y2)
        plot(n, y1)
        hold off
        close(hh);
    end
end


% --------------------------------------------------------------------
function varargout = pabaiga_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.pabaiga.
close(handles.figure1);


% --------------------------------------------------------------------
function varargout = x_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = nmin_Callback(h, eventdata, handles, varargin)

% --------------------------------------------------------------------
function varargout = nmax_Callback(h, eventdata, handles, varargin)
```

f_t.m:

```
function fun = f_t(x, n, k);
    if x <= 0
        fun = 0;
    else
        if x >= 1
            fun = 1;
        else
            for j = 1:k
                temp(j) = (-n*log(x))^(j-1)/factorial(j-1);
            end
            fun = (x^n)*sum(temp);
        end
    end;
```

f_a.m:

```
function fun = f_a(x, k);
    if x >= 0
        fun = 1;
    else
        for j = 1:k
            temp(j) = (-x)^(j-1)/factorial(j-1);
        end
        fun = exp(x)*sum(temp);
    end;
```

f_iv.m:

```
function fun = f_iv(x, n, k);
    if x >= 0
        fun = 0;
    else
        for j = 1:k
            temp(j) = (-x)^(j-1)/factorial(j-1);
```

```
        end
        if x <= -n
            fun = exp(x)*sum(temp);
        else
            fun = (x^2)*exp(x)*sum(temp)/(2*n);
        end
    end
```

g_t.m:

```
function fun = g_t(y, p, k);
    if y <= 0
        fun = 0;
    else
        if y >= 1
            fun = 1;
        else
            for j = 1:k
                temp(j) = s(j-1, y*(1-p), 0.00001)*(-log(y))^(j-1)/factorial(j-1);
            end
            fun = (y*p)*sum(temp);
        end
    end;

function fun = s(j, q, eps);
    d = 1;
    suma = 0;
    m = 1;
    while d > eps
        d = m^j*q^(m-1);
        m = m+1;
        suma = suma + d;
    end
    fun = suma;
```

g_a.m:

```
function fun = g_a(x, k);
    if x >= 0
        fun = 1;
    else
        fun = 1-(x^k)/(x-1)^k;
    end;
```

g_iv.m:

```
function fun = g_iv(x, n, k);
    if x >= 0
        fun = 0;
    else
        if x <= -n
            fun = g_a(x, k)
        else
            fun = ga(x,k)*(x^2)/(n+x-n*x);
        end
    end
```