Numerička matematika
2. predavanje

Saša Singer

singer@math.hr

web.math.pmf.unizg.hr/~singer

PMF – Matematički odsjek, Zagreb

Sadržaj predavanja

- Uvodna priča o greškama (nastavak):
  - Analiza pojedinih vrsta grešaka:
    - Greške metode — teorija aproksimacija.
    - Greške u podacima — teorija perturbacija.
  - Uvjetovanost problema.
  - Mjerenje grešaka — razne norme.
  - Uvjetovanost višedimenzionalnog problema.
  - Primjeri:
    - Uvjetovanost problema.
    - Izbjegavanje kraćenja.
  - Približno računanje i perturbacije podataka.
    - Greške zaokruživanja — direktna i obratna analiza.
  - Stabilni i nestabilni algoritmi. Primjeri.

Sadržaj predavanja (nastavak)

- Rješavanje linearnih sustava:
  - Uvod i ponavljanje teorije o linearnim sustavima.
  - Gaussove eliminacije.
  - Zamjene jednadžbi (redaka) — parcijalno pivotiranje.
  - Zamjene redaka i stupaca — potpuno pivotiranje.

Informacije

Trenutno nema bitnih informacija.

Greške i uvjetovanost

Greške — ponavljanje

Pri numeričkom rješavanju nekog problema javljaju se različiti tipovi grešaka:

- greške modela — svođenje realnog problema na neki "matematički" problem,
- greške u ulaznim podacima (mjerenja i sl.),
- greške numeričkih metoda za rješavanje "matematičkog" problema,
- greške "približnog" računanja — obično su to
  - greške zaokruživanja u aritmetici računala.

Greške modela su "izvan" dosega numeričke matematike.

- Spadaju u fiziku, kemiju, biologiju, tehniku, ekonomiju, ...

Greške (nastavak)

Sljedeće tri kategorije (podaci, metoda, računanje) su vezane za “matematički” problem, i

- spadaju u domenu numeričke matematike!

O njima nešto “moramo reći”.

Skica numeričkog rješavanja nekog problema sliči algoritmu:

Posebno, ako dozvolimo da, umjesto riječi “algoritam”,

- piše i riječ “metoda”.

Zamislite da pojam “algoritam” uključuje

- metodu i stvarno računanje rezultata!

Greške (nastavak)

$$\xrightarrow{\text{ULAZ}} \boxed{\text{ALGORITAM}} \xrightarrow{\text{IZLAZ}}$$

Sve tri vrste grešaka — podaci, metoda, računanje,

- rezultiraju nekom greškom u konačnom rezultatu!

Ta greška nas “zanima”.

Uočite da greške u ulaznim podacima možemo gledati

- neovisno o metodi ili algoritmu za rješenje problema,
- i tako dolazimo do pojma uvjetovanosti problema.

Za razliku od toga, greške metode i računanja, naravno,

- ovise o metodi, odnosno, algoritmu za rješenje problema.

Analiza grešaka

Greška metode

Gruba podjela numeričkih metoda — prema greškama:

Egzaktne metode

- daju egzaktno rješenje u konačnom broju "koraka", odnosno, računskih operacija.

Primjer:

- Gaussove eliminacije ili LR faktorizacija za linearne sustave.

Greška takvih metoda je nula, uz egzaktno računanje.

Približne ili neegzaktne metode

- daju približno rješenje problema, u konačnom broju "koraka" (računskih operacija).

Greška metode — približne metode

Mogu biti egzaktne — na nekom limesu!

Primjeri:

- zamjena kompliciranog modela jednostavnijim,
- greške diskretizacije (numerička integracija),
- greške odbacivanja/rezanja, konačne iteracije (rješavanje nelinearnih jednadžbi)

Analiza ovih grešaka spada u teoriju aproksimacija.

Pošteno, to je standardni predmet proučavanja numeričke matematike, u širem smislu,

- numerička analiza, funkcionalna analiza, itd.

Time se bavimo veći dio kolegija!

Greške u podacima

Ključno svojstvo problema je

- ovisnost rješenja o greškama ili pertubacijama ulaznih podataka.

To spada u teoriju perturbacije.

Da bi problem uopće imao smisla, očekujemo

- neku vrstu neprekidnosti rješenja,
- ili barem ograničenu osjetljivost rješenja na perturbacije.

Inače imamo "loše" postavljen (engl. "ill-posed") problem!

Osjetljivost se obično mjeri tzv. brojem uvjetovanosti problema (engl. "condition number"). Može ih biti i više.

Uvjetovanost problema

Neformalno rečeno, uvjetovanost problema mjeri

- osjetljivost problema na greške u podacima.

Osnovno svojstvo uvjetovanosti:

- Ne ovisi o konkretnoj numeričkoj metodi za rješenje problema, već samo o problemu.

Svrha uvjetovanosti = daje odgovor na pitanje:

- Koju točnost rezultata možemo očekivati
- pri točnom računanju, bez grešaka zaokruživanja,
- s (malo) pomaknutim — netočnim podacima?

Velika uvjetovanost $\longleftrightarrow$ nestabilan problem.

Model problema

Matematički model problema, zovimo ga P:

- za zadani ulaz — podatak $x \in \mathcal{X}$,
- dobivamo izlaz — rezultat $y \in \mathcal{Y}$.

Slikica modela je

$$x \longrightarrow \boxed{P} \longrightarrow y$$

Problem P interpretiramo kao računanje vrijednosti funkcije

$$f : \mathcal{X} \to \mathcal{Y},$$

gdje su $\mathcal{X}$ i $\mathcal{Y}$ odgovarajući matematički objekti. Na primjer, vektorski prostori, a vrlo često su i normirani prostori (treba nam mjera za grešku). Najčešće, $\mathcal{X} = \mathbb{R}^m$, $\mathcal{Y} = \mathbb{R}^n$.

Uvjetovanost problema (nastavak)

Ideja uvjetovanosti:

$$\text{greška u rezultatu} \approx \text{uvjetovanost} \cdot \text{greška u podacima}$$

Ovisi o obje vrijednosti: točnoj x i približnoj $\hat{x}$.
Za $m > 1$ ili $n > 1$, ovisi i o tome kako mjerimo greške.

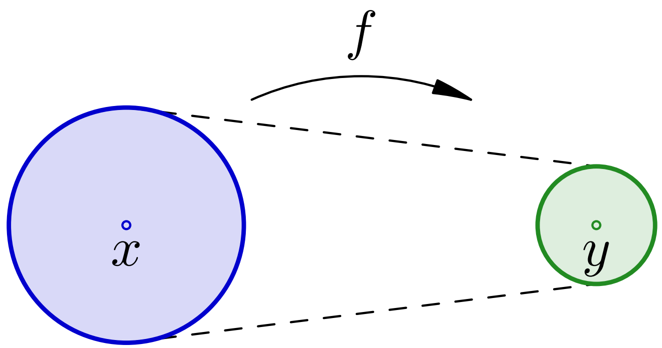
Napomene:

- Obično nas uvjetovanost posebno zanima za male perturbacije (greške, smetnje) podataka.
- Ako je f dovoljno glatka funkcija, možemo koristiti Taylorov razvoj u okolini točnog ulaznog podatka x
- i dobiti procjenu uvjetovanosti preko derivacija!

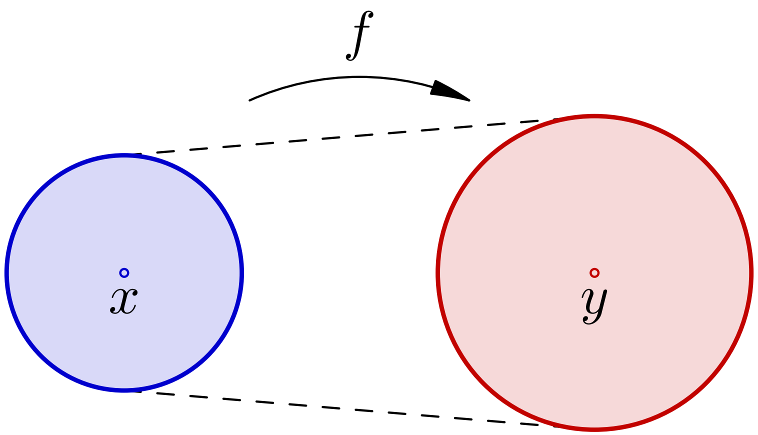
Više detalja malo kasnije, kad "sredimo" mjerenje grešaka!

Uvjetovanost — prigušivač i pojačalo grešaka

Mala uvjetovanost $\longleftrightarrow$ funkcija f je "prigušivač" grešaka:



Velika uvjetovanost $\longleftrightarrow$ funkcija f je "pojačalo" grešaka:



Norme i uvjetovanost

Kako mjeriti grešku?

Kad x i $y = f(x)$ nisu brojevi, nego vektori ili matrice, grešku možemo mjeriti na više načina.

- Po svakoj od komponenata vektora/matrica,
  - što je "vrlo precizno",
  - međutim, to je malo previše brojeva.
- Kao neku "ukupnu ili najveću" grešku,
  - što je samo jedan broj — pa se lakše nalazi,
  - iako može biti "neprecizno" (sažeta informacija).

  Ovo se radi korištenjem vektorskih i/ili matričnih normi.

Prisjetite se: vektorski prostor na kojem je definirana norma zove se normirani prostor.

Vektorske norme

"Vektorska" norma na vektorskom prostoru V (nad poljem F, gdje je $F = \mathbb{R}$ ili $F = \mathbb{C}$) je

- svaka funkcija $\| \cdot \| : V \to \mathbb{R}$

koja zadovoljava sljedeća svojstva:

1. $\|x\| \geq 0, \quad \forall x \in V,$
   a jednakost vrijedi ako i samo ako je $x = 0$,
2. $\|\alpha x\| = |\alpha| \, \|x\|, \quad \forall \alpha \in F, \ \forall x \in V,$
3. $\|x + y\| \leq \|x\| + \|y\|, \quad \forall x, y \in V$
   (nejednakost poznata pod imenom nejednakost trokuta).

Najpoznatije vektorske norme

Kad je vektorski prostor konačnodimenzionalan, $V = \mathbb{R}^n$ ili $V = \mathbb{C}^n$, najčešće se koriste sljedeće tri norme:

- 1-norma ili ℓ_1 norma $\|x\|_1 = \sum_{i=1}^{n} |x_i|$,
- 2-norma ili ℓ_2 norma ili euklidska norma
$$\|x\|_2 = (x^* x)^{1/2} = \sqrt{\sum_{i=1}^{n} |x_i|^2},$$
- ∞-norma ili ℓ_∞ norma $\|x\|_\infty = \max_{i=1,\ldots,n} |x_i|$.

Samo je 2-norma izvedena iz skalarnog produkta.

Norme na prostoru funkcija

Vektorski prostor V ne mora biti konačnodimenzionalan.

Na primjer, norme definirane na vektorskom prostoru $C[a, b]$ neprekidnih funkcija f na segmentu $[a, b]$, definiraju se slično normama na $\mathbb{R}^n$ (suma $\mapsto$ integral):

- L_1 norma $\|f\|_1 = \int\limits_a^b |f(t)|\,dt,$
- L_2 norma $\|f\|_2 = \left(\int\limits_a^b |f(t)|^2\,dt \right)^{1/2},$
- L_∞ norma $\|f\|_\infty = \max\{\,|f(x)| \mid x \in [a, b]\,\}.$

Ekvivalentnost normi

Može se pokazati da vrijedi sljedeći teorem.

Teorem. Na svakom konačnodimenzionalnom vektorskom prostoru V sve su norme ekvivalentne, tj. za svake dvije norme $\|\ \|_a$ i $\|\ \|_b$, postoje konstante c i C, takve da za sve $v \in V$ vrijedi

$$c\|v\|_a \leq \|v\|_b \leq C\|v\|_a.$$

∎

Na primjer,

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\,\|x\|_\infty,$$

za sve $x \in \mathbb{R}^n$.

Razlika između teorije i prakse — kad je n ogroman.

Matrične norme

Zamijenimo li u definiciji vektorske norme, čisto formalno, vektor x matricom A, dobivamo matričnu normu.

Matrična norma je svaka funkcija $\| \cdot \| : \mathbb{C}^{m\times n} \to \mathbb{R}$ koja zadovoljava sljedeća svojstva:

1. $\|A\| \geq 0, \quad \forall A \in \mathbb{C}^{m\times n}$,
   a jednakost vrijedi ako i samo ako je $A = 0$,
2. $\|\alpha A\| = |\alpha| \, \|A\|, \quad \forall \alpha \in \mathbb{C}, \ \forall A \in \mathbb{C}^{m\times n}$,
3. $\|A + B\| \leq \|A\| + \|B\|, \quad \forall A, B \in \mathbb{C}^{m\times n}$.

Tome se često dodaje i zahtjev konzistentnosti

4. $\|AB\| \leq \|A\| \, \|B\|$, kad god je produkt AB definiran.

Matrične norme (nastavak)

Matrične norme nastaju na dva načina.

- Matricu A promatramo kao vektor s $m \times n$ elemenata i za taj vektor koristimo odgovarajuću vektorsku normu.

  Najpoznatija takva norma odgovara vektorskoj 2-normi i zove se euklidska, Frobeniusova, Hilbert–Schmidtova, ili Schurova norma

$$\|A\|_F = (\mathrm{tr}(A^* A))^{1/2} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}.$$

- Operatorske norme:

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \quad \text{ili} \quad \|A\| = \max_{\|x\|=1} \|Ax\|.$$

Najpoznatije operatorske matrične norme

Uvrštavanjem odgovarajućih vektorskih normi, dobivamo

- matrična 1-norma, "maksimalna stupčana norma"

$$\|A\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{m} |a_{ij}|,$$

- matrična 2-norma, spektralna norma

$$\|A\|_2 = (\rho(A^* A))^{1/2} = \sigma_{\max}(A),$$

ρ je spektralni radijus, a σ singularna vrijednost matrice,

- matrična ∞-norma, "maksimalna retčana norma"

$$\|A\|_\infty = \max_{i=1,\ldots,m} \sum_{j=1}^{n} |a_{ij}|.$$

Matrične norme (nastavak)

Svojstva:

- Za matrične norme, također, vrijedi ekvivalentnost.
- Matrična 2-norma se teško računa u praksi — uobičajeno se procjenjuje korištenjem ostalih normi.
- Za svaku operatorsku normu vrijedi

  $$\|Ay\| \leq \|A\| \, \|y\|,$$

  za svaki vektor y. To se često koristi kod ocjena. Ova formula direktno izlazi iz definicije operatorske norme.
- Unitarna invarijantnost spektralne i Frobeniusove norme: za bilo koje unitarne matrice U (reda m) i V (reda n) vrijedi $\|UAV\|_2 = \|A\|_2$, $\|UAV\|_F = \|A\|_F$.

Uvjetovanost

Vrste uvjetovanosti — kratki pregled

Prema vrsti (tipu) greške koju gledamo:

- Apsolutna, relativna — po x, odnosno, po $y = f(x)$.

Prema načinu mjerenja greške (u više dimenzija):

- Po pojedinim komponentama ili po normi cijelog vektora.

Po dozvoljenoj "varijaciji" argumenata x i $\hat{x}$:

- U "fiksnim" točkama — tj. x i $\hat{x}$ su zadani. Nema puno smisla kao informacija o funkciji f, jer su točke fiksne.
- Lokalno oko x — $\hat{x}$ varira u nekoj zadanoj okolini oko x.
- Lokalno u točki x, za male perturbacije — na limesu kad $\hat{x} \to x$, tj. $\Delta x \to 0$, ako limes postoji. Ovisi samo o x.
- Globalno po x — (obično) kao najgori slučaj po svim x iz nekog skupa ili cijelog prostora. Ovisi samo o f.

Apsolutna greška i apsolutna uvjetovanost

Apsolutna, odnosno, relativna uvjetovanost problema mjeri koliko je problem osjetljiv na odgovarajuće promjene polaznih podataka.

- Apsolutna greška: $\|\Delta x\|$, $\|\Delta y\|$, (svaka norma u svom prostoru), gdje je

$$\Delta x = \hat{x} - x, \qquad \Delta y = \hat{y} - y.$$

- Apsolutna uvjetovanost:

$$\kappa_{\text{abs}}(x) := \frac{\|\Delta y\|}{\|\Delta x\|}.$$

Za male greške $\|\Delta x\|$, veza s derivacijom $f'(x)$ je očita!

Relativna greška i relativna uvjetovanost

U praksi se češće koristi relativna mjera za grešku (na primjer, zbog aritmetike računala).

- Relativna greška (po normi):

$$\delta_x := \frac{\|\Delta x\|}{\|x\|}, \qquad \delta_y := \frac{\|\Delta y\|}{\|y\|}.$$

- Relativna uvjetovanost (po normi):

$$\kappa_{\text{rel}}(x) := \frac{\delta_y}{\delta_x}.$$

Problem je dobro uvjetovan u relativnom smislu ako je

- κ_{rel} što je moguće manji, (barem) za $\delta_x \to 0$.

Landauov simbol — red veličine

Za zapis "reda veličine" vrijednosti neke funkcije u okolini neke točke koristimo tzv. Landauov simbol.

Definicija. Neka su $g, h : \mathbb{R}^m \to \mathbb{R}^n$ funkcije, $\|\cdot\|_{\mathbb{R}^m}$ i $\|\cdot\|_{\mathbb{R}^n}$ norme i neka je $x_0 \in \mathbb{R}^m$.

Ako postoje konstante $\delta > 0$ i $C > 0$, takve da za sve x vrijedi

$$\|x - x_0\|_{\mathbb{R}^m} \leq \delta \quad \Longrightarrow \quad \|g(x)\|_{\mathbb{R}^n} \leq C \, \|h(x)\|_{\mathbb{R}^n},$$

onda kažemo da je

"funkcija g reda veličine $\mathcal{O}$ od h, kad x teži prema x_0"

i to pišemo ovako

$$g(x) = \mathcal{O}(h(x)) \quad (x \to x_0).$$

Landauov simbol (nastavak)

Napomena. Umjesto znaka “$=$”, korektno bi bilo pisati $\in$, tj.

$$g(x) \in \mathcal{O}(h(x)) \quad (x \to x_0).$$

Također, često se piše “veliko” O, umjesto “pisanog” $\mathcal{O}$.

Primjer. Za $m = n = 1$ vrijedi:

$$\sin x = \mathcal{O}(x), \quad \sin x = x + \mathcal{O}(x^3) \quad (x \to 0),$$

$$x^2 + 3x = \mathcal{O}(x) \quad (x \to 0),$$

$$x^2 - x - 6 = \mathcal{O}(x - 3), \quad (x \to 3).$$

Zadnje dvije relacije opisuju ponašanje polinoma u okolini jednostruke nultočke, a izlaze iz zapisa

$$x^2 + 3x = x(x + 3), \quad x^2 - x - 6 = (x - 3)(x + 2).$$

Uvjetovanost i Taylorov teorem

Istražimo uvjetovanost problema računanja vrijednosti funkcije $f : \mathbb{R} \to \mathbb{R}$ u nekoj točki x.

Promatramo ponašanje funkcije f za male perturbacije Δx u okolini točke x. Neka je Δy pripadna perturbacija funkcijske vrijednosti $y = f(x)$, tj.

$$f(x + \Delta x) = y + \Delta y.$$

Neka je f još dva puta neprekidno derivabilna oko x. Korištenjem Taylorovog polinoma stupnja 1 dobivamo da je

$$\begin{aligned} \Delta y &= f(x + \Delta x) - f(x) \\ &= f'(x)\, \Delta x + \frac{f''(x + \vartheta \Delta x)}{2!}\, (\Delta x)^2, \quad \vartheta \in (0, 1). \end{aligned}$$

Apsolutna uvjetovanost i Taylorov teorem

Za male perturbacije Δx, apsolutni oblik ove relacije je

$$\Delta y = f'(x)\,\Delta x + O((\Delta x)^2).$$

Odavde slijedi da je apsolutna uvjetovanost funkcije f jednaka $f'(x)$ ili $|f'(x)|$, za male perturbacije Δx,

$$\kappa_{\mathrm{abs}}(x) = |f'(x)|.$$

Relativna uvjetovanost i Taylorov teorem

Ako je $x \neq 0$ i $y \neq 0$, dijeljenjem s y izlazi relativna forma

$$\frac{\Delta y}{y} = \frac{x f'(x)}{f(x)} \cdot \frac{\Delta x}{x} + O\left(\left(\frac{\Delta x}{x}\right)^2\right).$$

Ovdje koristimo da su $1/x$ i $1/y$ ograničene, pa za sve dovoljno male relativne pertubacije $\Delta x/x$ vrijedi

$$\frac{(\Delta x)^2}{y} = \frac{x^2}{y} \left(\frac{\Delta x}{x}\right)^2 = O\left(\left(\frac{\Delta x}{x}\right)^2\right).$$

Onda relativnu uvjetovanost funkcije f možemo definirati kao

$$\kappa_{\mathrm{rel}}(x) = (\mathrm{cond}\, f)(x) := \left|\frac{x f'(x)}{f(x)}\right|.$$

Uvjetovanost — posebni slučajevi oko nule

Ako je $x = 0$ i $y \neq 0$, onda relativna greška u x nema smisla (nije ograničena). Zato gledamo apsolutnu grešku u x i relativnu u y. Pripadni tzv. “miješani” broj uvjetovanosti je

$$(\operatorname{cond} f)(x) := \left| \frac{f'(x)}{f(x)} \right|.$$

Analogno, za $x \neq 0$ i $y = 0$, pripadni broj uvjetovanosti je

$$(\operatorname{cond} f)(x) := |x f'(x)|.$$

Ako je $x = y = 0$, onda gledamo samo apsolutne greške, pa je

$$(\operatorname{cond} f)(x) := |f'(x)|.$$

Uvjetovanost — primjer

Primjer. Relativna uvjetovanost funkcije

$$f(x) = \ln x$$

jednaka je

$$(\operatorname{cond} f)(x) = \left| \frac{x f'(x)}{f(x)} \right| = \left| \frac{1}{\ln x} \right|,$$

što je veliko za $x \approx 1$, kada je $\ln x \approx 0$.

Pitanje: Apsolutna uvjetovanost?

Uvjetovanost višedimenzionalnog problema

Što u više dimenzija?

Kad je $f : \mathbb{R}^m \to \mathbb{R}^n$, problem postaje složeniji. Uz oznake

$$x = (x_1, x_2, \ldots, x_m)^T \in \mathbb{R}^m, \quad y = (y_1, y_2, \ldots, y_n)^T \in \mathbb{R}^n,$$

preslikavanje f možemo komponentno zapisati kao

$$y_k = f_k(x_1, x_2, \ldots, x_m), \quad k = 1, 2, \ldots, n.$$

Ponovno, pretpostavljamo da svaka funkcija f_k ima

- neprekidne parcijalne derivacije po svim komponentnim varijablama x_ℓ u točki x, do barem drugog reda.

Najdetaljniju analizu dobivamo gledajući promjene

- svake komponentne funkcije f_k po svakoj varijabli x_ℓ.

Finija analiza — svaki izlaz po svakom ulazu

Promjena koju uzrokuje mala relativna perturbacija varijable x_ℓ u funkciji f_k ista je kao za funkciju jedne varijable.

Relativna uvjetovanost tog problema je

$$\gamma_{k\ell}(x) := (\mathrm{cond}_{k\ell}\, f)(x) := \left| \frac{x_\ell}{f_k(x)} \cdot \frac{\partial f_k}{\partial x_\ell} \right| .$$

Ako to napravimo za sve varijable x_ℓ i za svaku funkciju f_k, dobivamo matricu brojeva uvjetovanosti

$$\Gamma(x) = [\, \gamma_{k\ell}(x) \,] \in \mathbb{R}_+^{n \times m}.$$

Da bismo iz matrice $\Gamma(x)$ dobili jedan broj, koristimo neku normu i definiramo

$$(\mathrm{cond}\, f)(x) := \|\Gamma(x)\|.$$

Primjer — uvjetovanost aritmetičkih operacija

Zadatak. Osnovne aritmetičke operacije $\circ = +, -, *, /$, na realnim brojevima gledamo kao računanje vrijednosti funkcije $f_\circ : \mathbb{R}^2 \to \mathbb{R}$, gdje je

$$f_\circ(x_1, x_2) = x_1 \circ x_2.$$

Izračunajte pripadne matrice $\Gamma_\circ(x_1, x_2)$ za svaku operaciju $\circ$ i nađite pripadnu relativnu uvjetovanost u ∞–normi. ■

Rješenje.

$$\|\Gamma_\pm(x_1, x_2)\|_\infty = \frac{|x_1| + |x_2|}{|x_1 \pm x_2|},$$

$$\|\Gamma_*(x_1, x_2)\|_\infty = \|\Gamma_/(x_1, x_2)\|_\infty = 2.$$

To odgovara ranijim rezultatima za (vrlo) male relativne greške u polaznim podacima, tj. na limesu kad $\varepsilon \to 0$!

Grublja analiza — po normi

Grublju analizu — s manje parametara, dobivamo po ugledu na jednodimenzionalnu, promatranjem

- apsolutnih i relativnih perturbacija vektora u smislu norme, pri čemu je $\|\cdot\|$ bilo koja vektorska norma.

Relativnu perturbaciju vektora $x \in \mathbb{R}^m$ "po normi" definiramo kao

$$\frac{\|\Delta x\|}{\|x\|}, \qquad \Delta x = (\Delta x_1, \Delta x_2, \ldots, \Delta x_m)^T,$$

Pretpostavljamo da su komponente Δx_ℓ vektora perturbacije Δx male u odnosu na pripadne komponente x_ℓ vektora x. Tada je i $\|\Delta x\|/\|x\|$ malo (obrat ne vrijedi).

Isto napravimo i za vektor $y \in \mathbb{R}^n$, tj. gledamo $\|\Delta y\|/\|y\|$.

Taylorov razvoj komponentnih funkcija

Sada možemo pokušati povezati relativnu perturbaciju od y s relativnom perturbacijom od x.

Za male perturbacije Δx, iz početka Taylorovog razvoja funkcije f_k dobivamo

$$\Delta y_k = f_k(x + \Delta x) - f_k(x) \approx \sum_{\ell=1}^{m} \frac{\partial f_k}{\partial x_\ell} \Delta x_\ell.$$

Ovu relaciju možemo zapisati u vektorsko-matričnom obliku

$$\Delta y \approx \frac{\partial f}{\partial x} \cdot \Delta x,$$

gdje je $\dfrac{\partial f}{\partial x} = J_f(x)$ Jacobijeva matrica funkcije f u točki x.

Jacobijeva matrica preslikavanja

Jacobijeva matrica $J_f(x)$ sadrži parcijalne derivacije svih komponentnih funkcija po svim varijablama:

$$[\, J_f(x) \,]_{k\ell} = \frac{\partial f_k}{\partial x_\ell}, \quad k = 1, \ldots, n, \quad \ell = 1, \ldots, m,$$

ili

$$J_f(x) = \frac{\partial f}{\partial x} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \cdots & \dfrac{\partial f_1}{\partial x_m} \\[2ex] \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \cdots & \dfrac{\partial f_2}{\partial x_m} \\[2ex] \cdots & \cdots & \cdots & \cdots \\[2ex] \dfrac{\partial f_n}{\partial x_1} & \dfrac{\partial f_n}{\partial x_2} & \cdots & \dfrac{\partial f_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{n \times m}.$$

Apsolutne perturbacije po normi

Iz približne jednakosti za male perturbacije

$$\Delta y \approx \frac{\partial f}{\partial x} \cdot \Delta x,$$

uzimanjem bilo koje operatorske ili konzistentne norme izlazi

$$\|\Delta y\| \approx \left\| \frac{\partial f}{\partial x} \cdot \Delta x \right\| \lesssim \left\| \frac{\partial f}{\partial x} \right\| \cdot \|\Delta x\|.$$

Za operatorske norme, prethodna nejednakost oštra, tj. postoji perturbacija Δx za koju se ona dostiže.

Odavde vidimo da normu Jacobijeve matrice možemo uzeti kao apsolutnu uvjetovanost "po normi".

Uočite: Za $m = n = 1$ dobivamo isto kao i ranije!

Relativne perturbacije po normi

Kao i ranije, ako je $x \neq 0$ i $y \neq 0$, dijeljenjem s $\|y\|$ dobivamo da za relativne perturbacije po normi vrijedi

$$\frac{\|\Delta y\|}{\|y\|} \lesssim \frac{\|x\|}{\|f(x)\|} \cdot \left\| \frac{\partial f}{\partial x} \right\| \cdot \frac{\|\Delta x\|}{\|x\|}.$$

To opravdava definiciju relativne uvjetovanosti “po normi” u obliku

$$(\operatorname{cond} f)(x) := \frac{\|x\|}{\|f(x)\|} \cdot \left\| \frac{\partial f}{\partial x} \right\|.$$

Ova uvjetovanost je mnogo grublja nego $\|\Gamma(x)\|$, jer norma pokušava “uništiti” detalje o komponentama vektora.

Ako su komponente bitno različitih redova veličina, samo najveće po apsolutnoj vrijednosti igraju neku ulogu.

Primjer uvjetovanosti problema

Problem — računanje integrala (Gautschi)

Ispitajmo uvjetovanost problema računanja integrala

$$I_n = \int_0^1 \frac{t^n}{t+5}\, dt,$$

za zadani nenegativni cijeli broj $n \in \mathbb{N} \cup \{0\} = \mathbb{N}_0$.

U ovom obliku, problem je napisan kao preslikavanje iz $\mathbb{N}_0$ u $\mathbb{R}$ i ne "paše" ranijem pojmu problema.

- Domena ovdje nije $\mathbb{R}$, nego $\mathbb{N}_0$ (diskretan skup), pa nema smisla govoriti o neprekidnosti, derivabilnosti i sl.

Zato prvo transformiramo problem.

Rekurzija za integral

Nađimo vezu između I_k i I_{k-1}, s tim da I_0 znamo izračunati

$$I_0 = \int\limits_0^1 \frac{1}{t+5}\,dt = \ln(t+5)\,\Big|_0^1 = \ln\frac{6}{5}.$$

Za početak, očito vrijedi da je

$$\frac{t}{t+5} = 1 - \frac{5}{t+5}.$$

Množenjem obje strane s t^{k-1} dobivamo

$$\frac{t^k}{t+5} = t^{k-1} - 5\,\frac{t^{k-1}}{t+5}.$$

Rekurzija za integral (nastavak)

Na kraju, integracijom na segmentu $[0, 1]$ izlazi

$$I_k = \int_0^1 t^{k-1}\, dt - 5I_{k-1} = \frac{1}{k} - 5I_{k-1}, \quad k = 1, 2, \ldots, n.$$

Dakle, I_k je rješenje (linearne, nehomogene) diferencijske jednadžbe prvog reda

$$y_k = -5y_{k-1} + \frac{1}{k}, \quad k = 1, 2, \ldots,$$

uz početni uvjet $y_0 = I_0$.

Ovo gore je dvočlana rekurzivna relacija za y_k (pogledajte priču o rekurzivnim relacijama u Diskretnoj matematici).

Rekurzija unaprijed — zapis funkcijama

Varijacija početnog uvjeta definira niz funkcija f_k, $y_k = f_k(y_0)$.

Zanima nas relativna uvjetovanost funkcije f_n u točki $y_0 = I_0$, u ovisnosti o $n \in \mathbb{N}_0$. Razlog:

- I_0 nije egzaktno prikaziv u računalu,
- umjesto I_0, spremi se aproksimacija $\widehat{I}_0$,
- konačni rezultat — neka aproksimacija $\widehat{I}_n = f_n(\widehat{I}_0)$.

Indukcijom ili supstitucijom unatrag lako se dokaže da vrijedi

$$y_n = f_n(y_0) = (-5)^n y_0 + p_n, \quad p_n = \sum_{k=1}^{n} \frac{(-5)^{n-k}}{k},$$

gdje je p_n ovisi samo o nehomogenim članovima rekurzije, ali ne i o početnom uvjetu y_0.

Rekurzija unaprijed — relativna uvjetovanost

Relativna uvjetovanost funkcije f_n u točki y_0 je

$$(\mathrm{cond}\, f_n)(y_0) = \left| \frac{y_0 f_n'(y_0)}{y_n} \right| = \left| \frac{y_0 (-5)^n}{y_n} \right|.$$

Iz definicije integrala slijedi: I_n monotono padaju po n, čak

$$\lim_{n \to \infty} I_n = 0.$$

Zbrajanjima dobivamo sve manje i manje brojeve! U $y_0 = I_0$ je

$$(\mathrm{cond}\, f_n)(I_0) = \frac{I_0 \cdot 5^n}{I_n} > \frac{I_0 \cdot 5^n}{I_0} = 5^n.$$

Zaključak: f_n je vrlo loše uvjetovana u $y_0 = I_0$, i to tim gore kad n raste.

Rekurzija unaprijed — rezultati

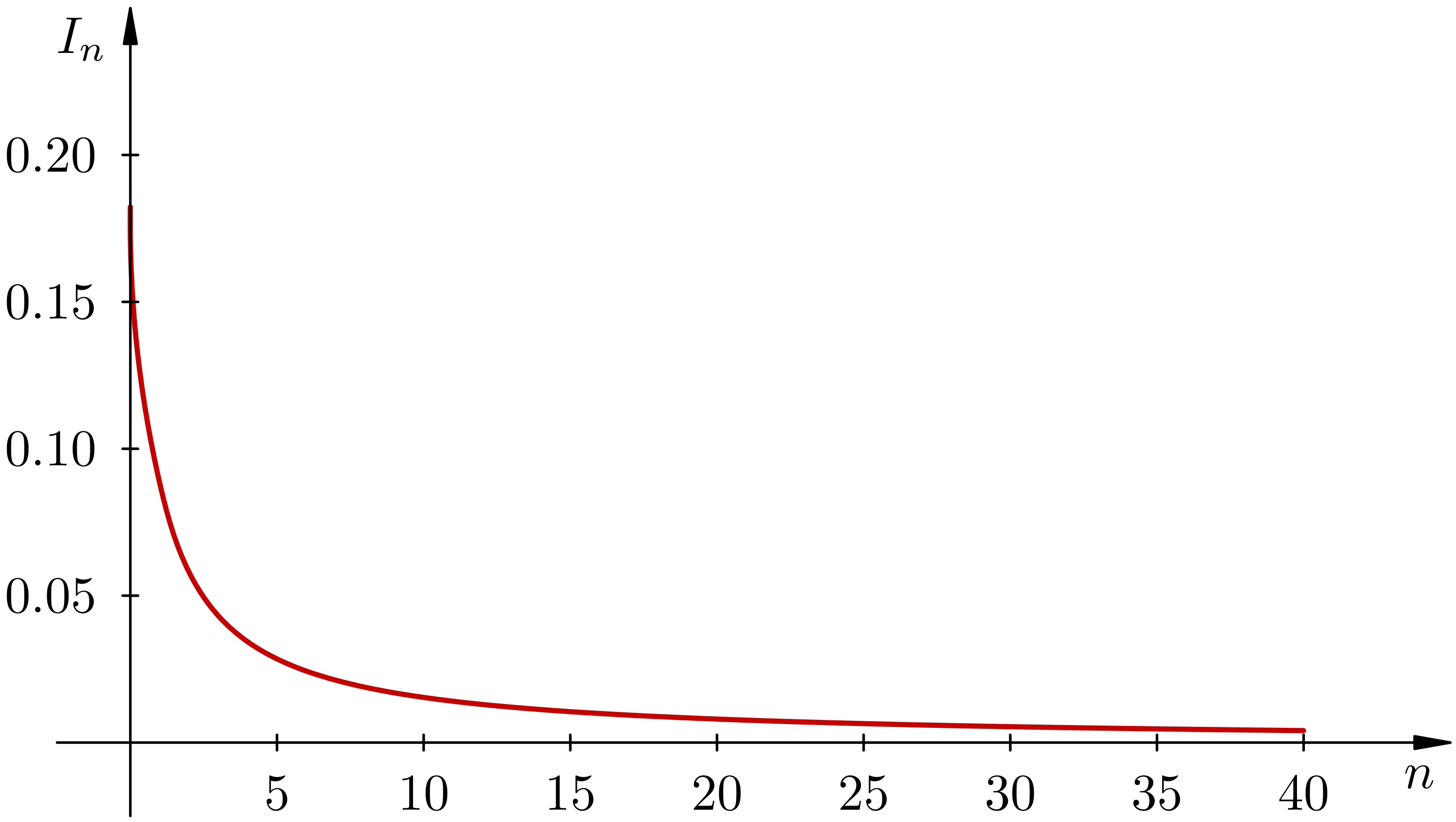
Pitanje: Kako se loša uvjetovanost vidi, kad stvarno računamo $f_n(I_0)$ u aritmetici računala?

Algoritam unaprijed, za zadani n (pseudokôd):

```
k = 0;
y = ln( 6.0 / 5.0 );     // y_0
ispisi k, y;
za k = 1 do n radi {
    y = -5.0 * y + 1.0 / k;     // y_k
    ispisi k, y;
}
```

Slikice! Pokaži program i rezultate!

Točne vrijednosti integrala I_n



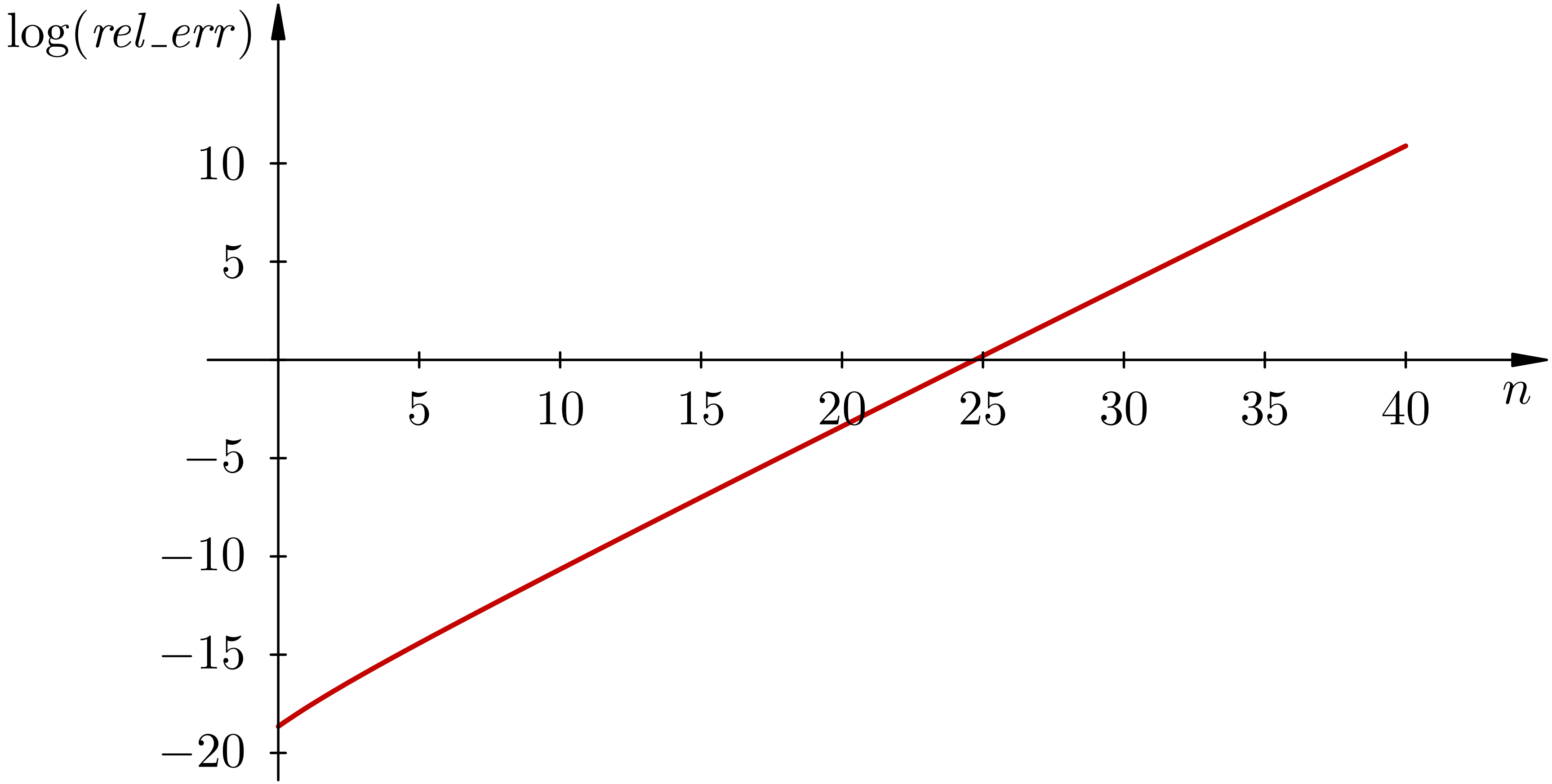
egzaktne/točne vrijednosti integrala I_n

Rekurzija unaprijed — numerički rezultati

Izračunate vrijednosti u tipu `extended` ($u \approx 5.42 \cdot 10^{-20}$).
Crvene znamenke su pogrešne!

n	$\widehat{I}_n$	I_n	rel. greška
0	1.82321556793954626E−1	1.82321556793954626E−1	2.2E−19
1	8.83922160302268688E−2	8.83922160302268689E−2	−2.0E−18
2	5.80389198488656562E−2	5.80389198488656553E−2	1.5E−17
..	...	...	...
22	7.38035060732479776E−3	7.29738306511145509E−3	1.1E−02
23	6.57650783294122857E−3	6.99134554400794192E−3	−5.9E−02
24	8.78412750196052383E−3	6.70993894662695705E−3	3.1E−01
25	−3.92063750980261915E−3	6.45030526686521474E−3	−1.6E+00
..	...	...	...
39	−6.32992112791892692E+7	4.18374034921478077E−3	−1.5E+10
40	3.16496056420946346E+8	4.08129825392609613E−3	7.8E+10

Rekurzija unaprijed za I_n — relativne greške



($\log_{10}$) relativne greške izračunate vrijednosti integrala I_n rekurzijom unaprijed

Rekurzija unaprijed — komentar rezultata

Izračunata vrijednost $\widehat{I}_0$ ima

- vrlo malu relativnu grešku — samo nekoliko u.

Međutim, ta mala greška "eksplodira" vrlo brzo,

- jer se pojačava s faktorom 5 u svakoj iteraciji.

Isto vrijedi i za sve greške zaokruživanja iza toga, samo je ukupni faktor pojačanja malo manji (kasnije su nastale).

Stvarni problem i bitna razlika od primjera "$\sin 24\pi$":

- Ovdje nema velikih omjera brojeva u algoritmu.

Brojevi I_n relativno sporo padaju — omjer I_0/I_{40} je ispod 50. Po tome, očekivali bismo gubitak točnosti od oko 2 decimale,

- a stvarno imamo užasno i još "nevidljivo" kraćenje.

Rekurzija unatrag — zapis funkcijama

Može li se loša uvjetovanost izbjeći?

- Može — okretanjem rekurzije, unaprijed $\mapsto$ unatrag!

Treba uzeti neki $\nu > n$ i "silazno" računati

$$y_{k-1} = \frac{1}{5}\left(\frac{1}{k} - y_k\right), \qquad k = \nu, \nu - 1, \ldots, n + 1.$$

Ovo, u principu, smijemo koristiti i za $n = 0$, tj. računati y_0.

Problem: Kako izračunati početnu vrijednost y_ν?

Nova rekurzija definira niz funkcija $g_{n,\nu}$, koje vežu y_n i y_ν, uz $\nu > n$, tj.

$$y_n = g_{n,\nu}(y_\nu).$$

Rekurzija unatrag — relativna uvjetovanost

Relativna uvjetovanost za $g_{n,\nu}$ je

$$(\mathrm{cond}\, g_{n,\nu})(y_\nu) = \left| \frac{y_\nu \, (-1/5)^{\nu-n}}{y_n} \right|, \quad \nu > n.$$

Za $y_\nu = I_\nu$ dobivamo da je $y_n = I_n$, a iz monotonosti I_n slijedi

$$(\mathrm{cond}\, g_{n,\nu})(I_\nu) = \frac{I_\nu}{I_n} \cdot \left(\frac{1}{5} \right)^{\nu-n} < \left(\frac{1}{5} \right)^{\nu-n}, \quad \nu > n,$$

što je ispod 1, tj. relativne greške se prigušuju.

- Prigušenje grešaka ide s faktorom $1/5$ po svakoj iteraciji!
- To vrijedi i za greške zaokruživanja napravljene u ranijim iteracijama (u aritmetici računala).

Rekurzija unatrag — početna vrijednost

Ako je $\widehat{I}_\nu$ neka aproksimacija za I_ν, onda za relativne perturbacije vrijedi

$$\left|\frac{\widehat{I}_n - I_n}{I_n}\right| = (\operatorname{cond} g_{n,\nu})(I_\nu) \cdot \left|\frac{\widehat{I}_\nu - I_\nu}{I_\nu}\right| < \left(\frac{1}{5}\right)^{\nu-n} \cdot \left|\frac{\widehat{I}_\nu - I_\nu}{I_\nu}\right|.$$

Zbog linearnosti funkcije $g_{n,\nu}$, ova relacija vrijedi za bilo kakve perturbacije, a ne samo za male.

- Početna vrijednost $\widehat{I}_\nu$ uopće ne mora biti blizu prave I_ν.
- Možemo uzeti $\widehat{I}_\nu = 0$, čime smo napravili relativnu grešku od 100% (tj. 1) u početnoj vrijednosti ...

Rekurzija unatrag — točnost i start ν

- ... a još uvijek dobivamo $\widehat{I}_n$ s relativnom greškom

$$\left|\frac{\widehat{I}_n - I_n}{I_n}\right| < \left(\frac{1}{5}\right)^{\nu-n}, \quad \nu > n.$$

- Povoljnim izborom ν, ocjenu na desnoj strani možemo napraviti po volji malom — ispod tražene točnosti ε.
- Dovoljno je uzeti $\widehat{I}_\nu = 0$ i

$$\nu \geq n + \frac{\log(1/\varepsilon)}{\log 5},$$

a zatim računamo vrijednosti

$$\widehat{I}_{k-1} = \frac{1}{5}\left(\frac{1}{k} - \widehat{I}_k\right), \quad k = \nu, \nu-1, \ldots, n+1.$$

Rekurzija unatrag — rezultati

Pitanje: Kako se dobra uvjetovanost vidi, kad stvarno računamo $g_{n,\nu}(I_\nu)$ u aritmetici računala?

Pokaži program i rezultate za $\varepsilon = 10^{-19}$!

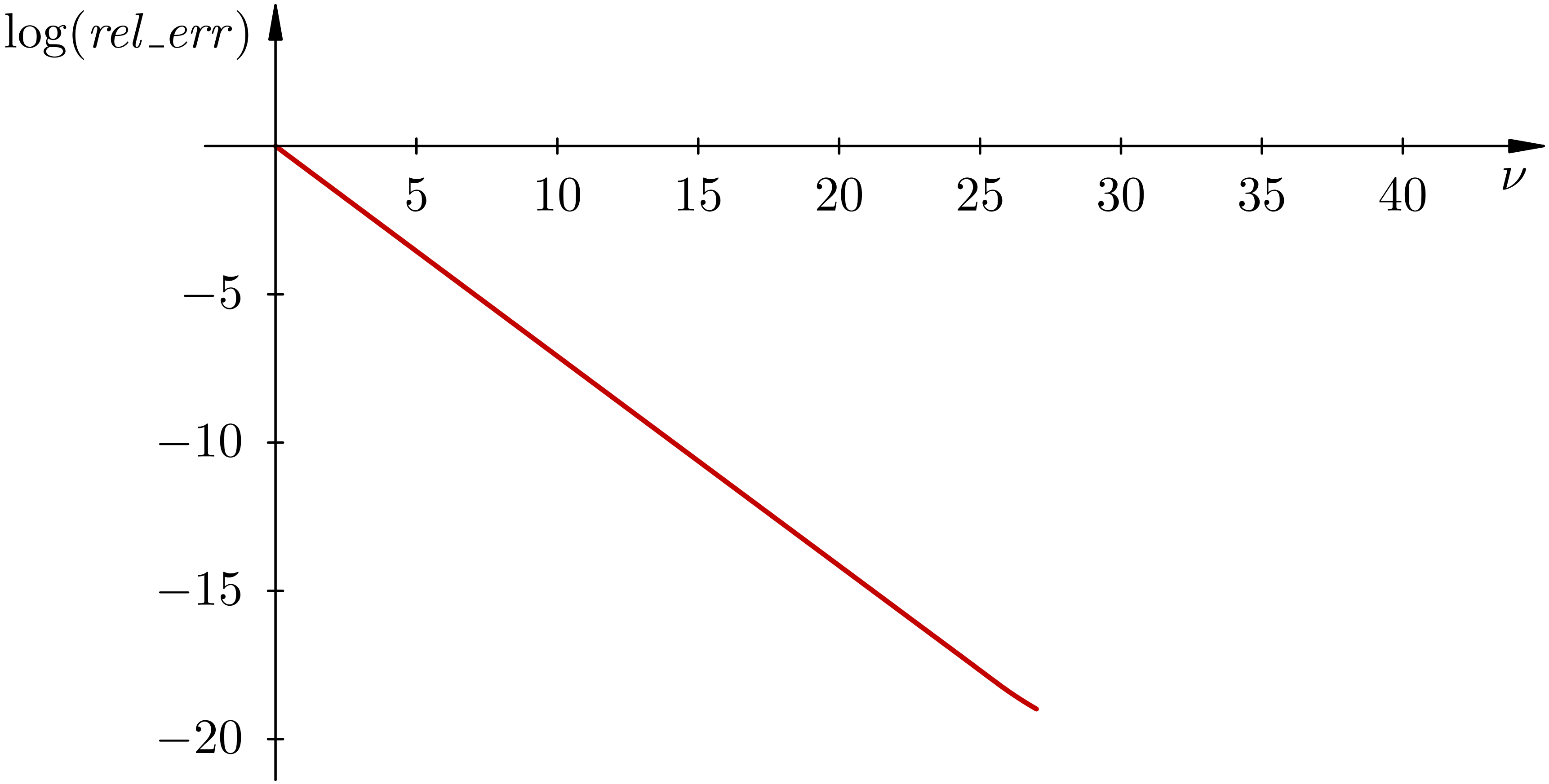
- Početna vrijednost je $\widehat{I}_\nu = 0$.
- Za ovaj ε dobijemo

$$\nu \geq n + \frac{\log(1/\varepsilon)}{\log 5} \approx n + 28.$$

Dakle, "silazno" računamo 28 vrijednosti.

Usput, to su vrijednosti za I_n iz ranije tablice i sve prikazane znamenke su točne (ima ih 18). Dakle, "ništa se ne vidi". Greška je samo nekoliko u, jer je $\varepsilon \approx 2u$ i imamo 3 operacije.

Rekurzija unatrag za I_{40} — ovisno o startu ν



(log$_{10}$) relativne greške izračunate vrijednosti integrala I_{40} obratnom rekurzijom za $I_{40+\nu} = 0$

Rekurzivno računanje — završne napomene

Rekurzije prvog i (posebno) drugog reda se vrlo često koriste u praksi

- ne samo za računanje vrijednosti integrala,
- već za računanje raznih specijalnih funkcija (poput Besselovih) i ortogonalnih polinoma (v. kasnije).

Zato oprez ...

- treba znati nešto o stabilnosti rekurzije, prije računanja!

"Trik" okretanja rekurzije poznat je kao Millerov algoritam. Prvi puta je iskorišten baš za računanje Besselovih funkcija.

Primjeri izbjegavanja kraćenja

Primjer:
Kvadratna jednadžba

Kvadratna jednadžba

Uzmimo da treba riješiti (realnu) kvadratnu jednadžbu

$$ax^2 + bx + c = 0,$$

gdje su a, b i c zadani, i vrijedi $a \neq 0$.

Matematički gledano, problem je lagan: imamo 2 rješenja

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Numerički gledano, problem je mnogo izazovniji:

- ni uspješno računanje po ovoj formuli,
- ni točnost izračunatih korijena,

ne možemo uzeti "zdravo za gotovo".

Kvadratna jednadžba — standardni oblik

Za početak, jer znamo da je $a \neq 0$, onda jednadžbu možemo podijeliti s a, tako da dobijemo tzv. "normalizirani" oblik

$$x^2 + px + q = 0, \qquad p = \frac{b}{a}, \quad q = \frac{c}{a}.$$

Po standardnim formulama, rješenja ove jednadžbe su

$$x_{1,2} = \frac{-p \pm \sqrt{p^2 - 4q}}{2}.$$

Međutim, u praksi, stvarno računanje se radi po formuli

$$x_{1,2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q},$$

s tim da na početku izračunamo i zapamtimo $p/2$ ili $-p/2$. Ovim postupkom štedimo jedno množenje (ono s 4).

Kvadratna jednadžba — problem

Primjer. Rješavamo kvadratnu jednadžbu $x^2 - 56x + 1 = 0$.

U dekadskoj aritmetici s $p = 5$ značajnih znamenki dobijemo

$$x_1 = 28 - \sqrt{783} = 28 - 27.982 = 0.018000,$$

$$x_2 = 28 + \sqrt{783} = 28 + 27.982 = 55.982.$$

Točna rješenja su

$$x_1 = 0.0178628\ldots \quad \text{i} \quad x_2 = 55.982137\ldots.$$

Apsolutno manji od ova dva korijena — x_1, ima samo dvije točne znamenke (kraćenje), relativna greška je $7.7 \cdot 10^{-3}$!

Apsolutno veći korijen x_2 je "savršeno" točan.

Kvadratna jednadžba — popravak

Prvo izračunamo većeg po apsolutnoj vrijednosti, po formuli

$$x_2 = \frac{-(b + \operatorname{sign}(b)\sqrt{b^2 - 4ac})}{2a} = -\frac{p}{2} - \operatorname{sign}(p)\sqrt{\left(\frac{p}{2}\right)^2 - q},$$

a manjeg po apsolutnoj vrijednosti, izračunamo iz

$$x_1 \cdot x_2 = \frac{c}{a} = q$$

(Vièteova formula), tj. formula za x_1 je

$$x_1 = \frac{c}{x_2\, a} = \frac{q}{x_2}.$$

Opasnog kraćenja za x_1 više nema!

Kvadratna jednadžba (nastavak)

Ovo je bila samo jedna, od (barem) tri “opasne” točke za računanje. Preostale dvije su:

- “kvadriranje” pod korijenom — mogućnost za overflow. Rješenje — “skaliranjem”.
- oduzimanje u diskriminanti s velikim kraćenjem — nema jednostavnog rješenja. Naime, “krivac” nije aritmetika.
  - To je samo odraz tzv. nestabilnosti problema. Tad imamo dva bliska korijena, koji su vrlo osjetljivi na male promjene (perturbacije) koeficijenata jednadžbe.
  - Na primjer, pomak c = pomak grafa “gore–dolje”. Mali pomak rezultira velikom promjenom korijena!

Neki primjeri izbjegavanja kraćenja

Primjer. Treba izračunati

$$y = \sqrt{x + \delta} - \sqrt{x},$$

gdje su x i δ zadani ulazni podaci, s tim da je $x > 0$,

- a $|\delta|$ vrlo mali broj.

U ovoj formuli, očito, dolazi do velike greške zbog kraćenja — zaokruživanje korijena prije oduzimanja.

Ako formulu "deracionaliziramo" u oblik

$$y = \frac{\delta}{\sqrt{x + \delta} + \sqrt{x}},$$

problema više nema!

Neki primjeri izbjegavanja kraćenja

Primjer. Treba izračunati

$$y = \cos(x + \delta) - \cos x,$$

gdje su x i δ zadani ulazni podaci, s tim da je $|\cos x|$ razumno velik,

- a $|\delta|$ vrlo mali broj.

Opet, dolazi do velike greške zbog kraćenja.

Ako formulu napišmo u "produktnom" obliku

$$y = -2 \sin \frac{\delta}{2} \sin \left(x + \frac{\delta}{2} \right),$$

problema više nema!

Primjer za nultočke polinoma

Svojstvene vrijednosti i nultočke polinoma

U linearnoj algebri, svojstvene vrijednosti zadane matrice A se računaju "na ruke" kao

- nultočke karakterističnog polinoma te matrice

$$k_A(\lambda) = \det(\lambda I - A) = 0.$$

Prvo, računanjem determinante, nađemo "standardni" oblik karakterističnog polinoma, preko koeficijenata

$$k_A(\lambda) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0,$$

a onda tražimo nultočke $\lambda_1, \ldots, \lambda_n$ tog polinoma.

Oprez: Nultočke polinoma mogu biti vrlo osjetljive na male perturbacije u koeficijentima polinoma.

Primjer — Wilkinsonov polinom

Primjer. Uzmimo tzv. Wilkinsonov polinom stupnja $n = 20$,

$$P_{20}(\lambda) = (\lambda - 1) \cdot (\lambda - 2) \cdots (\lambda - 19) \cdot (\lambda - 20).$$

Iz ovog "multiplikativnog" oblika odmah čitamo da su nultočke tog polinoma, redom, prirodni brojevi

$$\lambda_i = i, \quad i = 1, \ldots, 20.$$

Ovaj oblik polinoma — kao produkt linearnih faktora, je

- idealno stabilan na male perturbacije "polaznih" podataka,
- jer su ti podaci upravo nultočke polinoma!

Wilkinsonov polinom — razvijen po potencijama

Kad polinom P_{20} "razvijemo" po potencijama od λ, tj. zapišemo preko koeficijenata c_j, dobivamo

$$P_{20}(\lambda) = \lambda^{20} + c_{19}\lambda^{19} + \cdots + c_1\lambda + c_0,$$

s koeficijentima:

$$\begin{aligned} c_{19} &= -(1 + 2 + \cdots + 19 + 20) = -210, \\ &\vdots \\ c_0 &= (-1) \cdot (-2) \cdots (-19) \cdot (-20) = 20! \end{aligned}$$

Baš to je oblik kojeg bismo, na primjer, izračunali iz pripadne matrice. Poanta:

- Ovdje se nultočke baš i "ne vide" odmah ...

Treba ih izračunati!

Egzaktni koeficijenti Wilkinsonovog polinoma

Točne vrijednosti koeficijenata c_j su

$c_0 =$	$2432\,90200\,81766\,40000$	$c_{10} =$	$1\,30753\,50105\,40395$
$c_1 =$	$-8752\,94803\,67616\,00000$	$c_{11} =$	$-13558\,51828\,99530$
$c_2 =$	$13803\,75975\,36407\,04000$	$c_{12} =$	$1131\,02769\,95381$
$c_3 =$	$-12870\,93124\,51509\,88800$	$c_{13} =$	$-75\,61111\,84500$
$c_4 =$	$8037\,81182\,26450\,51776$	$c_{14} =$	$4\,01717\,71630$
$c_5 =$	$-3599\,97951\,79476\,07200$	$c_{15} =$	$-16722\,80820$
$c_6 =$	$1206\,64780\,37803\,73360$	$c_{16} =$	$533\,27946$
$c_7 =$	$-311\,33364\,31613\,90640$	$c_{17} =$	$-12\,56850$
$c_8 =$	$63\,03081\,20992\,94896$	$c_{18} =$	20615
$c_9 =$	$-10\,14229\,98655\,11450$	$c_{19} =$	-210

Koeficijenti su "jedva" prikazivi u tipu `extended`, a sigurno nisu egzaktno prikazivi u manjim tipovima, poput `double`.

Mala perturbacija koeficijenta c_{19}

U polinomu P_{20} napravimo

- jednu jedinu perturbaciju veličine 2^{-23} u koeficijentu c_{19},

tako da dobijemo polinom

$$\widetilde{P}_{20}(\lambda) = P_{20}(\lambda) - 2^{-23}\lambda^{19}.$$

Pripadna relativna perturbacija koeficijenta c_{19} je

- reda veličine 2^{-30}, odnosno, ispod 10^{-9}.

Reklo bi se — zaista mala perturbacija!

Kako izgledaju nultočke tog perturbiranog polinoma $\widetilde{P}_{20}$, tj.

- jesu li se i nultočke "malo" promijenile?

Nažalost, ne!

Nestabilnost nultočaka Wilkinsonovog polinoma

Egzaktne nultočke polinoma $\widetilde{P}_{20}$, na 9 decimala, su

$1.00000\,0000$	$6.00000\,6944$	$10.09526\,6145 \pm 0.64350\,0904\, i$
$2.00000\,0000$	$6.99969\,7234$	$11.79363\,3881 \pm 1.65232\,9728\, i$
$3.00000\,0000$	$8.00726\,7603$	$13.99235\,8137 \pm 2.51883\,0070\, i$
$4.00000\,0000$	$8.91725\,0249$	$16.73073\,7466 \pm 2.81262\,4894\, i$
$4.99999\,9928$	$20.84690\,8101$	$19.50243\,9400 \pm 1.94033\,0347\, i$

Od 20 realnih nultočaka polinoma P_{20}, dobili smo

- samo 10 realnih — prvih 9 i zadnja,
- i 5 parova konjugirano kompleksnih, s vrlo "nezanemarivim" imaginarnim dijelovima.

Ni govora o "maloj" perturbaciji!

Svojstvene vrijednosti matrica — pouka

Zato se, u praksi, svojstvene vrijednosti matrice A

- nikad (ili gotovo nikad) ne računaju kao
- nultočke karakterističnog polinoma k_A.

Za taj problem postoji gomila raznih numeričkih metoda, ovisno o vrsti matrice i raznim drugim stvarima.

Približno računanje i perturbacije podataka

Interpretacija grešaka zaokruživanja

Kod približnog računanja — na pr. u aritmetici računala, imamo greške zaokruživanja. One nastaju

- spremanjem ulaznih podataka u algoritam,
- kao rezultat svake pojedine aritmetičke operacije.

Ključna stvar za analizu tih grešaka je

- svođenje na teoriju perturbacija, u smislu
- egzaktnog računanja s perturbiranim polaznim podacima!

Kako to ide? Ilustracija na IEEE standardu.

Greške prikaza i aritmetike

Ako je ulazni podatak $x \in \mathbb{R}$

- unutar raspona brojeva prikazivih u računalu,

onda se, umjesto x, sprema zaokruženi prikazivi broj $f\ell(x)$, tako da vrijedi

$$f\ell(x) = (1 + \varepsilon)x, \quad |\varepsilon| \le u,$$

gdje je

- ε relativna greška napravljena tim zaokruživanjem,
- a u je jedinična greška zaokruživanja.

Imamo malu relativnu grešku, a računalo dalje računa

- s perturbiranim polaznim podatkom $f\ell(x)$.

Slična stvar vrijedi i za aritmetičke operacije.

Zaokruživanje u aritmetici

Osnovna pretpostavka za realnu aritmetiku u računalu:

- za sve četiri osnovne aritmetičke operacije vrijedi ista ocjena greške zaokruživanja kao i za prikaz brojeva.

Isto vrijedi i za neke matematičke funkcije, poput $\sqrt{\ }$, ali ne mora vrijediti za sve funkcije (na pr. za sin oko 0, ili ln oko 1).

Preciznije: Neka $\circ$ označava bilo koju operaciju $+$, $-$, $*$, $/$. Za prikazive brojeve u dozvoljenom rasponu $x, y \in \mathcal{F}$, takve da je i egzaktni rezultat $x \circ y$ u dozvoljenom rasponu (tj. u $\mathcal{F}$), vrijedi ocjena relativne greške

$$fl(x \circ y) = (1 + \varepsilon)\,(x \circ y), \quad |\varepsilon| \leq u.$$

Broj ε ovisi o x, y, operaciji $\circ$ i aritmetici računala.

Širenje grešaka zaokruživanja

Kad imamo puno operacija — nastaje problem:

- greške se šire i
- treba procijeniti grešku u rezultatu.

Kako to napraviti?

Ponovimo, za aritmetiku računala ne vrijedi:

- asocijativnost zbrajanja i množenja,
- distributivnost množenja prema zbrajanju.

Posljedica: poredak izvršavanja operacija je bitan!

Jedino što vrijedi je:

- komutativnost za zbrajanje i množenje.

Širenje grešaka zaokruživanja (nastavak)

Za analizu grešaka zaokruživanja ne možemo koristiti nikakva “normalna” pravila za aritmetičke operacije u računalu, jer ti zakoni naprosto ne vrijede.

Stvarna algebarska struktura je izrazito komplicirana i postoje debele knjige na tu temu.

- Vrijede neka “zamjenska” pravila,

ali su neupotrebljiva za analizu iole većih proračuna.

Međutim, analiza pojedinih operacija postaje bitno lakša, ako uočimo da:

- greške zaokruživanja u aritmetici računala možemo interpretirati i kao egzaktne operacije, ali na “malo” pogrešnim podacima!

Širenje grešaka zaokruživanja (nastavak)

Kako? Dovoljno je faktor $(1 + \varepsilon)$ u ocjeni greške

$$fl(x \circ y) = (1 + \varepsilon)\,(x \circ y), \quad |\varepsilon| \le u,$$

"zalijepiti" na x i/ili y. To je isto kao da operand(i) ima(ju) neku relativnu grešku na ulazu u operaciju, a operacija $\circ$ je egzaktna. Dakle,

- izračunati (ili "zaokruženi") rezultat jednak je egzaktnom rezultatu, ali za malo promijenjene (tj. perturbirane) podatke (u relativnom smislu).

Što dobivamo ovom interpretacijom?

- Onda možemo koristiti "normalna" pravila egzaktne aritmetike za analizu grešaka.

Širenje grešaka zaokruživanja (nastavak)

Ne zaboravimo još da ε ovdje ovisi o x, y, i operaciji $\circ$. Kad takvih operacija ima više, pripadne greške obično označavamo nekim indeksom u ε.

Na primjer, ako je $\circ$ zbrajanje $(+)$, onda je

$$\begin{aligned} f\ell(x+y) &= (1+\varepsilon_{x+y})\,(x+y) \\ &= [\,(1+\varepsilon_{x+y})\,x\,] + [\,(1+\varepsilon_{x+y})\,y\,], \end{aligned}$$

uz $|\varepsilon_{x+y}| \leq u$, ako su x, y i $x+y$ u prikazivom rasponu.

Potpuno ista stvar vrijedi i za oduzimanje.

Kod množenja i dijeljenja možemo birati kojem ulaznom podatku ćemo “zalijepiti” faktor $(1+\varepsilon)$.

Širenje grešaka zaokruživanja (nastavak)

Za množenje možemo pisati

$$\begin{aligned} f\ell(x * y) &= (1 + \varepsilon_{x*y})\,(x * y) \\ &= \big[\,(1 + \varepsilon_{x*y})\,x\,\big] * y = x * \big[\,(1 + \varepsilon_{x*y})\,y\,\big], \end{aligned}$$

a za dijeljenje

$$\begin{aligned} f\ell(x \,/\, y) &= (1 + \varepsilon_{x/y})\,(x \,/\, y) \\ &= \big[\,(1 + \varepsilon_{x/y})\,x\,\big] \,/\, y = x \,/\, \big[\,y/(1 + \varepsilon_{x/y})\,\big]. \end{aligned}$$

Postoje i druge varijante. Na primjer, da svakom operandu "zalijepimo" $\sqrt{1+\varepsilon}$ (odnosno $1/\sqrt{1+\varepsilon}$), ali to nije naročito važno. Bitno je samo da je izračunati rezultat egzaktan za malo perturbirane podatke.

Širenje grešaka (bilo kojih)

Zasad nije vidljivo koja je točno korist od ove interpretacije. Stvar se bolje vidi tek kad imamo više operacija zaredom.

Međutim, ova ideja s “malo pogrešnim podacima” je

- baš ono što nam treba za analizu širenja grešaka,

i to bez obzira na uzrok grešaka, čim se sjetimo da

- rezultati ranijih operacija
- s nekom greškom ulaze u nove operacije.

Naime, uzroka grešaka može biti mnogo, ovisno o tome što računamo. Od grešaka modela i metode, preko grešaka mjerenja (u ulaznim podacima), do grešaka zaokruživanja.

Širenje grešaka u aritmetici računala

Dosad smo gledali širenje grešaka u egzaktnoj aritmetici.

U aritmetici računala postupamo na potpuno isti način. Samo treba zgodno iskoristiti onu raniju interpretaciju da je

- izračunati (ili "zaokruženi") rezultat jednak egzaktnom, ali za malo perturbirane podatke (u relativnom smislu).

A širenje grešaka u egzaktnoj aritmetici znamo.

Ukratko, bez dokaza:

Svaka pojedina aritmetička operacija u računalu samo

- povećava perturbaciju svojih ulaznih podataka za jedan faktor oblika $(1 + \varepsilon)$, uz ocjenu $|\varepsilon| \leq u$,

ovisno o tome kojim operandima "zalijepimo" taj faktor.

Natuknice o analizi grešaka

Bilo koji algoritam gledamo kao preslikavanje:

$$\text{ulaz (domena)} \;\to\; \text{izlaz (kodomena)}.$$

Naravno, zanima nas

- greška u izračunatom rezultatu — u kodomeni,
- uz približno računanje aritmetikom računala.

Ova greška zove se greška unaprijed (engl. forward error).

Nažalost, postupak "direktne" analize grešaka je težak,

- relativno rijetko "ide" i često daje loše ocjene greške.

Primjer. Obična norma vektora u $\mathbb{R}^2$ (i još dodaj "scaling"). (v. Z. Drmač, članak u MFL-u).

Natuknice o analizi grešaka (nastavak)

U praksi se puno češće koristi tzv. “obratna” analiza grešaka. Osnovna ideja je ista kao i za pojedine operacije:

- izračunati rezultati algoritma mogu se dobiti egzaktnim računanjem,
- ali na perturbiranim ulaznim podacima — u domeni.

Ova greška u domeni zove se greška unatrag ili obratna greška (engl. backward error).

Prednost: ocjena tih perturbacija u domeni je bitno lakša,

- jer se akumulacija onih faktora oblika $(1 + \varepsilon)$ prirodno radi unatrag — od rezultata prema polaznim podacima.

U protivnom, moramo znati grešku za operande, a to je greška unaprijed za prethodni dio algoritma.

Natuknice o analizi grešaka (nastavak)

Postupak "unatrag" za nalaženje grešaka u izračunatim rezultatima ide u dva koraka.

- Prvo se obratnom analizom naprave ocjene perturbacija polaznih podataka u domeni,
- a zatim se koristi matematička teorija perturbacije, koja daje ocjene grešaka rezultata u kodomeni. Ovaj izvod ide za egzaktni račun, pa vrijede sva normalna pravila.

Tako stižemo do pojmova:

- stabilno i nestabilno računanje ili algoritam $=$ "prigušivač" ili "pojačalo" grešaka.

Slikice (skripta NA, Higham) — su na sljedećoj stranici.

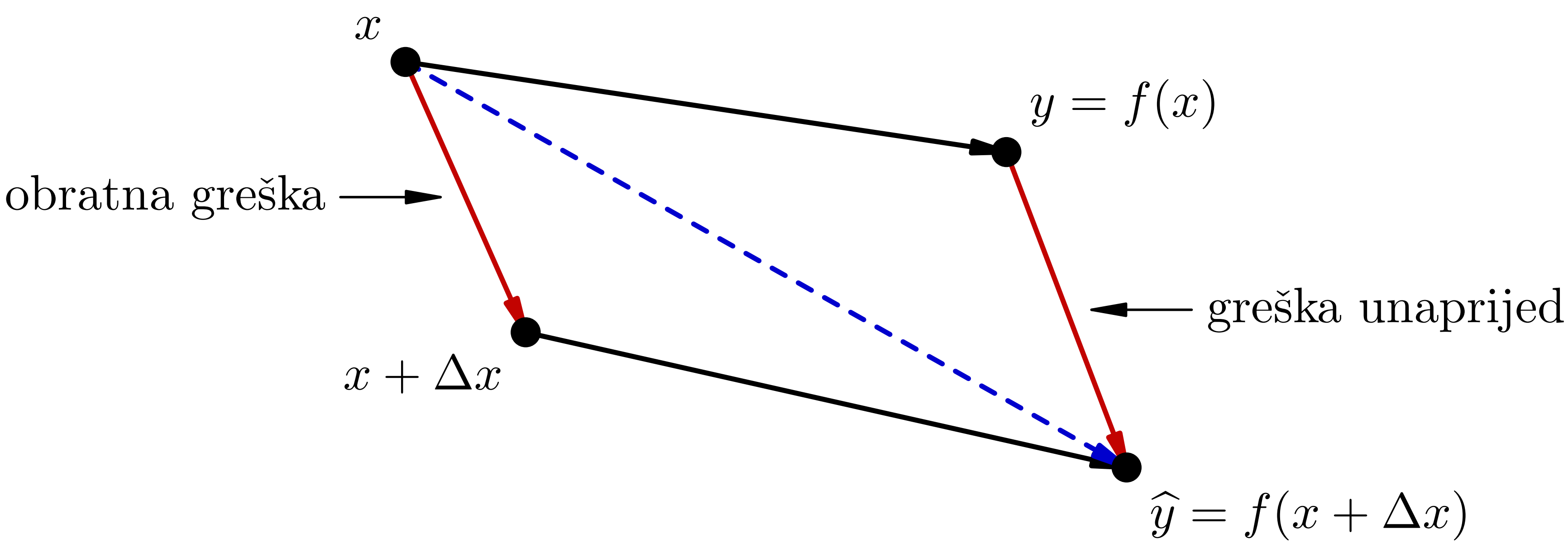
- Primjeri nestabilnosti — uklonjivi i NEuklonjivi.

Greška unaprijed i obratna greška

Uzmimo da algoritam “rješava” problem P.

$$x \longrightarrow \boxed{P} \longrightarrow y$$

Ako problem P intepretiramo kao računanje funkcije f, onda grešku unaprijed i obratnu grešku možemo prikazati ovako:



Zbrajanje brojeva

Primjer: računanje sume u aritmetici računala

Primjer. Računamo sumu (zbroj) $s_n = x_1 + x_2 + \cdots + x_n$, uz pretpostavku da su svi brojevi x_i prikazivi u računalu, tj. vrijedi $x_i = f\ell(x_i)$, za $i = 1, \ldots, n$.

Algoritam. Zbrajamo unaprijed — redom po indeksima:

```
s = x_1;
za i = 2 do n radi
    s = s + x_i;
```

Oznake. Razlikujemo egzaktne i izračunate parcijalne sume:

- Egzaktne parcijalne sume su $s_i = s_{i-1} + x_i = x_1 + \cdots + x_i$,
- Izračunate parcijalne sume su $\hat{s}_i = f\ell(\hat{s}_{i-1} + x_i)$,

za $i = 2, \ldots, n$. Za početnu sumu vrijedi $s_1 = \hat{s}_1 = x_1$.

Greške zaokruživanja u aritmetici računala

Prema IEEE standardu, za svaki izračunati rezultat vrijedi

$$\hat{s}_i = f\ell(\hat{s}_{i-1} + x_i) = (1 + \varepsilon_{i-1})\,(\hat{s}_{i-1} + x_i), \quad i = 2, \ldots, n,$$

s tim da je $|\varepsilon_{i-1}| \leq u$, uz pretpostavku da su svi $\hat{s}_i$ unutar prikazivog raspona (nadalje smatramo da je tako).

Jedino razumno je izraziti završni rezultat $\hat{s}_n$ u terminima

- polaznih vrijednosti $x_1, \ldots, x_n$.

Kad to napravimo i sredimo po x_i, dobivamo

$$\begin{aligned}\hat{s}_n = {} & (1 + \varepsilon_1) \cdots (1 + \varepsilon_{n-1})\, x_1 + (1 + \varepsilon_1) \cdots (1 + \varepsilon_{n-1})\, x_2 \\ & + (1 + \varepsilon_2) \cdots (1 + \varepsilon_{n-1})\, x_3 + \cdots \\ & + (1 + \varepsilon_{n-2})\,(1 + \varepsilon_{n-1})\, x_{n-1} + (1 + \varepsilon_{n-1})\, x_n.\end{aligned}$$

Zapis izračunate sume

Izračunatu sumu $\hat{s}_n$ možemo napisati u obliku

$$\hat{s}_n = (1+\eta_1)\,x_1 + (1+\eta_2)\,x_2 + \cdots + (1+\eta_n)\,x_n,$$

gdje je

$$\begin{aligned} \eta_1 = \eta_2 &= (1+\varepsilon_1)\cdots(1+\varepsilon_{n-1}) - 1 \\ \eta_3 &= (1+\varepsilon_2)\cdots(1+\varepsilon_{n-1}) - 1 \\ &\;\vdots \\ \eta_{n-1} &= (1+\varepsilon_{n-2})\,(1+\varepsilon_{n-1}) - 1 \\ \eta_n &= (1+\varepsilon_{n-1}) - 1 = \varepsilon_{n-1}. \end{aligned}$$

Iz $|\varepsilon_i| \leq u$ dobivamo ocjene (v. “velika” skripta, str. 135–136)

$$|\eta_i| \leq (1+u)^{n+1-i} - 1 \leq \gamma_{n+1-i} := \frac{(n+1-i)u}{1-(n+1-i)u},$$

za $i = 2, \ldots, n$, i $|\eta_1| = |\eta_2| \leq \gamma_{n-1}$ (uz uvjet $(n-1)u < 1$).

Što ćemo s tim — interpretacija unatrag

Pogled unatrag u domenu — obratna greška, stabilnost unatrag (engl. backward error, backward stability), iz relacije

$$\hat{s}_n = (1 + \eta_1)\, x_1 + (1 + \eta_2)\, x_2 + \cdots + (1 + \eta_n)\, x_n.$$

Izračunati rezultat $\hat{s}_n$ je egzaktna suma

- malo "perturbiranih" polaznih podataka $x_1, \ldots, x_n$,
- s obratnim ili polaznim relativnim greškama $\eta_1, \ldots, \eta_n$.

To kaže da je algoritam zbrajanja stabilan "unatrag".

Pogled unaprijed = teorija perturbacije za greške $\eta_1, \ldots, \eta_n$.
Prava greška izračunate sume je

$$\hat{s}_n - s_n = \eta_1 x_1 + \eta_2 x_2 + \cdots + \eta_n x_n.$$

Ocjena relativne greške unaprijed

Odavde slijedi

$$\begin{aligned} |\hat{s}_n - s_n| &\le (|x_1| + \cdots + |x_n|) \cdot \max_{i=1,\ldots,n} |\eta_i| \\ &\le (|x_1| + \cdots + |x_n|) \cdot \gamma_{n-1}. \end{aligned}$$

Za relativnu grešku izračunate sume dobivamo ocjenu

$$\frac{|\hat{s}_n - s_n|}{|s_n|} \le \frac{|x_1| + \cdots + |x_n|}{|x_1 + \cdots + x_n|} \cdot \gamma_{n-1} = \mathrm{cond}(s_n) \cdot \gamma_{n-1},$$

gdje je

$$\mathrm{cond}(s_n) = \frac{|x_1| + \cdots + |x_n|}{|x_1 + \cdots + x_n|}.$$

Zbrajanje brojeva istog predznaka $\Longrightarrow$ $\mathrm{cond}(s_n) = 1$.

Poredak zbrajanja za iste predznake

Zbog prijelaza $|\eta_i| \leq \max_{i=1,\ldots,n} |\eta_i|$, prethodna ocjena vrijedi

- za bilo koji algoritam — neovisno o poretku sumanada!

Prave obratne greške η_i, naravno, ne znamo. Međutim, znamo da za ocjene tih grešaka vrijedi:

$$|\eta_1| = |\eta_2| \leq \gamma_{n-1}, \quad |\eta_i| \leq \gamma_{n+1-i}, \quad i = 2, \ldots, n,$$

tj. najveća je ocjena za prva dva sumanda x_1, x_2, i ocjena pada kako indeksi rastu (sumandi kasnije ulaze u zbrajanja).

Kad zbrajamo brojeve istog predznaka (tj. nema kraćenja), najmanju ocjenu greške dobivamo tako da

- sumande x_i poredamo uzlazno po apsolutnoj vrijednosti!
  Razlog: veće (ocjene) greške idu na manje sumande.

Pogledati primjer za parcijalne sume harmonijskog reda.

Rješavanje linearnih sustava

Općenito o linearnim sustavima — teorija

Neka je $\mathcal{F} = \mathbb{R}$ polje realnih brojeva (može i $\mathcal{F} = \mathbb{C}$).

Zadani su:

- (pravokutna) matrica $A \in \mathcal{F}^{m\times n}$ i vektor $b \in \mathcal{F}^m$.

Tražimo rješenje linearnog sustava

$$Ax = b.$$

Teorem Kronecker–Capelli kaže da linearni sustav $Ax = b$

- ima rješenje $x \in \mathcal{F}^n$ — ako i samo ako je rang matrice A, u oznaci r, jednak rangu proširene matrice $\widehat{A} = [\, A \mid b \,]$,
- rješenje sustava je jedinstveno ako je $r = n$.

Znamo čak i malo više!

Linearni sustavi — teorija (nastavak)

Opće rješenje sustava $Ax = b$ (ako postoji) ima oblik

$$x = x_p + \mathcal{N}(A),$$

gdje je

- x_p jedno partikularno rješenje polaznog sustava $Ax = b$,
- $\mathcal{N}(A)$ je nul-potprostor od A, ili opće rješenje pripadnog homogenog sustava $Ax = 0$.

Iz teorema o rangu i defektu za matricu A

$$r + \dim \mathcal{N}(A) = n,$$

slijedi tvrdnja o jedinstvenosti rješenja:

- $\dim \mathcal{N}(A) = 0 \iff r = n.$

Linearni sustavi — od teorije prema praksi

U praksi se najčešće rješavaju linearni sustavi $Ax = b$ kod kojih je matrica A kvadratna i regularna.

- A kvadratna — znači da je $m = n$ (A je reda n).
- A regularna — znači, na primjer, $\det A \neq 0$.

Iz teorema Kronecker–Capelli onda izlazi da

- rješenje x takvog sustava postoji i jedinstveno je.

$\Longrightarrow$ ima smisla promatrati algoritme za računanje rješenja.

Nema smisla računati nešto što (možda) i ne postoji, ili nije jedinstveno (koje od mnogo rješenja računamo).

Oprez: To što unaprijed znamo da je A regularna

- ne znači da to vrijedi i numerički!

Kako naći rješenje? — Inverz matrice

Teorija (1). Možemo naći inverz matrice A, tj. matricu A^{-1}

- i slijeva pomnožiti sustav $Ax = b$ matricom A^{-1}.

Dobivamo

$$x = A^{-1}b.$$

Samo je pitanje: kako ćemo izračunati inverz A^{-1}?

Zaključak: Lakši problem sveli smo na teži — u prijevodu, pali smo s konja na magarca.

Zašto? Jednostavno, zato što je

- j-ti stupac inverza, upravo, rješenje sustava $Ax = e_j$.

Dakle, za n stupaca od A^{-1} treba riješiti n linearnih sustava. A krenuli smo od jednog (sustava)! Ne tako!

Kako naći rješenje? — Cramerovo pravilo

Teorija (2). Iz linearne algebre znate za Cramerovo pravilo:

- j-ta komponenta rješenja sustava je

$$x_j = \frac{\det A_j}{\det A}, \qquad j = 1, \ldots, n,$$

- pri čemu je matrica A_j jednaka matrici A,
- osim što je j-ti stupac u A_j zamijenjen desnom stranom b.

Treba još “samo” izračunati determinante — i to $n+1$ njih.
A kako ćemo to?

“Klasični” odgovor: pa ... recimo, Laplaceovim razvojem.

Jao, jao ... Bilo kako, samo ne tako!

Kako naći rješenje? — Zaboravite Cramera

Zašto? Laplaceovim razvojem dobijemo (kao u definiciji det)

- “samo” $n!$ pribrojnika u svakoj determinanti,
- a svaki pribrojnik je produkt od n faktora.

Prava “sitnica”. I tako to, još $n+1$ puta ...

Zaključak: Ako determinante računamo na ovaj način,

- složenost Cramerovog pravila za rješavanje linearnog sustava je eksponencijalna u n (dokažite to!)
- i nikad se ne koristi kao metoda numeričkog rješavanja.

Zaboravite determinante i Cramera — finale

Komentar: Determinante možemo računati i puuuno brže,

- tako da matricu svedemo na trokutasti oblik,
- postupkom sličnim Gaussovim eliminacijama.

Naime, determinanta trokutaste matrice (gornje ili donje) je

- produkt dijagonalnih elemenata,

pa se lako računa!

No, isti postupak eliminacija koristimo i za

- rješavanje "cijelog" linearnog sustava $Ax = b$,
- i to samo jednom, a ne $n + 1$ puta.

Dakle, Cramerovo pravilo se ne isplati ni kad ovako računamo determinante.

Kako naći rješenje? — Gaussove eliminacije!

Najjednostavnija metoda za rješavanje linearnih sustava su

- Gaussove eliminacije, odnosno,
- slične metode svođenja na trokutastu formu.

Ideja: Sustav $Ax = b$ se ekvivalentnim transformacijama svodi na sustav oblika

$$Rx = y,$$

gdje je

- R trokutasta matrica (recimo, gornja),

iz kojeg se lako, tzv. povratnom supstitucijom, nalazi rješenje.

Oznaka R — "right" (desna) = gornja trokutasta matrica.

Ekvivalentne transformacije sustava

Ekvivalentne transformacije sustava su

- one koje ne mijenjaju rješenje sustava.

Standardne ekvivalentne transformacije su (v. LA):

- zamjena poretka jednadžbi (nužno!),
- množenje jednadžbe brojem različitim od nule,
  - ova transformacija "skaliranja" se obično ne koristi, ili se vrlo pažljivo koristi — za povećanje stabilnosti,
- množenje jedne jednadžbe nekim brojem i dodavanje drugoj jednadžbi (ključno!),

  $=$ dodavanje linearne kombinacije preostalih jednadžbi, s tim da uzmemo samo jednu preostalu jednadžbu.

Gaussove eliminacije

Gaussove eliminacije — algoritam

Označimo $A^{(1)} := A$, $b^{(1)} := b$ na početku prvog koraka.

U skraćenoj notaciji, bez pisanja nepoznanica x_i, linearni sustav $Ax = b$ možemo zapisati proširenom matricom, kao

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right].$$

Svođenje na trokutastu formu radimo u $n - 1$ koraka.

Gaussove eliminacije — algoritam (nastavak)

1. korak.

- U prvom stupcu matrice $A^{(1)}$ poništimo sve elemente, osim prvog.

Kako se to radi?

Ako je element $a_{11}^{(1)} \neq 0$, onda redom, možemo

- od i-te jednadžbe oduzeti
- prvu jednadžbu pomnoženu s

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \quad i = 2, \ldots, n.$$

Pritom se prva jednadžba se ne mijenja.

Gaussove eliminacije — algoritam (nastavak)

Prva jednadžba — kao redak proširene matrice $[\,A^{(1)} \mid b^{(1)}\,]$, je

$$\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \end{array}.$$

Polazna i-ta jednadžba — pisana na isti način, za $i = 2, \ldots, n$

$$\begin{array}{cccc:c} a_{i1}^{(1)} & a_{i2}^{(1)} & \cdots & a_{in}^{(1)} & b_i^{(1)} \end{array}.$$

Nova i-ta jednadžba — pisana na isti način, za $i = 2, \ldots, n$

$$\begin{array}{cccc:c} a_{i1}^{(2)} & a_{i2}^{(2)} & \cdots & a_{in}^{(2)} & b_i^{(2)} \end{array}.$$

Relacije za nove elemente su

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1} a_{1j}^{(1)}, \quad b_i^{(2)} = b_i^{(1)} - m_{i1} b_1^{(1)},$$

za $j = 1, \ldots, n$, $i = 2, \ldots, n$. Prvi redak ($i = 1$) ostaje isti.

Gaussove eliminacije — algoritam (nastavak)

Iz ovih relacija za nove elemente (prepisane još jednom)

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1} a_{1j}^{(1)}, \quad b_i^{(2)} = b_i^{(1)} - m_{i1} b_1^{(1)},$$

vidimo da su multiplikatori

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \quad i = 2, \ldots, n.$$

odabrani upravo tako da je $a_{i1}^{(2)} = 0$, za $i = 2, \ldots, n$.

Dakle, nakon prvog koraka dobivamo proširenu matricu $[\, A^{(2)} \mid b^{(2)} \,]$ u kojoj

- prvi stupac ima nule (strogo) ispod dijagonale, tj. gornju trokutastu formu.

Gaussove eliminacije — algoritam (nastavak)

Time smo dobili ekvivalentni linearni sustav $A^{(2)}x = b^{(2)}$ s proširenom matricom

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right].$$

Postupak poništavanja možemo nastaviti s drugim stupcem matrice $A^{(2)}$ — na isti način.

Ako je $a_{22}^{(2)} \neq 0$, biramo faktore m_{i2} tako da poništimo sve elemente drugog stupca ispod dijagonale. I tako redom.

Gaussove eliminacije — algoritam (nastavak)

Općenito, k–ti korak izgleda ovako, za $k = 1, \ldots, n-1$:

- iz proširene matrice $[\, A^{(k)} \mid b^{(k)} \,]$ dobivamo novu proširenu matricu $[\, A^{(k+1)} \mid b^{(k+1)} \,]$,
- tako da poništimo sve elemente ispod dijagonale u k–tom stupcu matrice $A^{(k)}$.

Relacije za nove elemente koje treba izračunati u matrici $A^{(k+1)}$ i vektoru $b^{(k+1)}$ su

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \qquad b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)},$$

za $i, j = k+1, \ldots, n$, a multiplikatori m_{ik} su

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \qquad i = k+1, \ldots, n.$$

Gaussove eliminacije — algoritam (nastavak)

Prvih k redaka u $[\,A^{(k+1)} \mid b^{(k+1)}\,]$ ostaju isti kao u $[\,A^{(k)} \mid b^{(k)}\,]$.

Konačno, ako su svi $a_{ii}^{(i)} \neq 0$, za $i = 1, \ldots, n-1$, završni linearni sustav $[\,A^{(n)} \mid b^{(n)}\,]$, ekvivalentan polaznom, je

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & a_{nn}^{(n)} & b_n^{(n)} \end{array}\right].$$

Dobili smo gornju trokutastu matricu $R = A^{(n)}$ (nule u strogo donjem trokutu matrice R ne pišemo).

Gaussove eliminacije — algoritam (nastavak)

Uz pretpostavku da je $a_{nn}^{(n)} \neq 0$, ovaj se linearni sustav lako rješava tzv. povratnom supstitucijom (supstitucijom unatrag)

$$x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}},$$

$$x_i = \frac{1}{a_{ii}^{(i)}} \left(b_i^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)} x_j \right), \quad i = n-1, \ldots, 1.$$

Pitanje: Ako je A kvadratna i regularna matrica,

- moraju li svi elementi $a_{ii}^{(i)}$ biti različiti od nule?

To je nužno (i dovoljno) da algoritam "prođe" u ovom obliku.

Gaussove eliminacije — primjedba na algoritam

Odgovor: Ne!

Primjer. Linearni sustav $Ax = b$ s proširenom matricom

$$[\,A \mid b\,] = \left[\begin{array}{cc:c} 0 & 1 & 1 \\ 1 & 0 & 1 \end{array}\right]$$

je regularan ($\det A = -1$), sustav ima jedinstveno rješenje $x_1 = x_2 = 1$,

- a ipak ga ne možemo riješiti Gaussovim eliminacijama,
- ako ne mijenjamo poredak jednadžbi.

Zaključak: Moramo dozvoliti promjenu poretka jednadžbi.

Gaussove eliminacije s pivotiranjem

Pitanje: Dozvolimo li promjene poretka jednadžbi — tzv. "pivotiranje" u stupcu kojeg sređujemo,

- može li se Gaussovim eliminacijama s pivotiranjem riješiti svaki sustav kojemu je matrica kvadratna i regularna?

Odgovor: Ako dozvolimo pivotiranje

- zamjenom "ključne" jednadžbe i bilo koje druge koja ima ne-nula element (u tom stupcu),
- Gaussovim eliminacijama rješiv je svaki regularni kvadratni linearni sustav.

Objašnjenje: Ako u prvom stupcu nemamo ne-nula elemenata, matrica je singularna. Isto vrijedi i za svaki sljedeći korak (dokaz = Laplaceov razvoj determinante!).

Gaussove eliminacije — kako pivotiramo?

Pitanje: Kako vršiti pivotiranje, tj. zamjene jednadžbi?

- Zamjenom "ključne" jednadžbe i bilo koje druge koja ima ne-nula element (u tom stupcu)?

Odgovor: Tu je ključna razlika između egzaktnog i približnog računanja (kad imamo greške zaokruživanja).

- U teoriji — kod egzaktnog računanja, dovoljno je naći bilo koji ne-nula element (u tom stupcu).
- U praksi — kad računamo približno, to može dovesti do potpuno pogrešnog rezultata.

Jedna jedina operacija može upropastiti rezultat!

- Postoji i puno bolja strategija za pivotiranje, kojom se to (barem dijelom) može izbjeći.

Gaussove eliminacije — primjer

Primjer. Zadan je linearni sustav

$$
\begin{aligned}
0.0001\, x_1 + x_2 &= 1 \\
x_1 + x_2 &= 2.
\end{aligned}
$$

Matrica sustava je regularna, pa postoji jedinstveno rješenje

$$x_1 = 1.\dot{0}00\dot{1}, \quad x_2 = 0.\dot{9}99\dot{8}.$$

Riješimo taj sustav "računalom" koje ima 4 decimalne znamenke mantise i 2 znamenke eksponenta (ovo nije bitno).

Uočiti: Broj $0.0001 = 10^{-4}$ je "mali", ali nije nula. Po teoriji,

- možemo ga uzeti kao prvi (ili bilo koji) ne-nula element.

Gaussove eliminacije — primjer (nastavak)

Sustav zapisan u takvom "računalu" pamti se kao

$$
\begin{aligned}
1.000 \cdot 10^{-4}\, x_1 + 1.000 \cdot 10^{0}\, x_2 &= 1.000 \cdot 10^{0} \\
1.000 \cdot 10^{0}\quad x_1 + 1.000 \cdot 10^{0}\, x_2 &= 2.000 \cdot 10^{0}.
\end{aligned}
$$

Množenjem prve jednadžbe s -10^4 i dodavanjem drugoj, dobivamo novu drugu jednadžbu

$$(1.000 \cdot 10^{0} - 1.000 \cdot 10^{4})\, x_2 = 2.000 \cdot 10^{0} - 1.000 \cdot 10^{4}.$$

Oduzimanje u računalu se vrši tako da manji eksponent postane jednak većem, a mantisa se denormalizira. Dobivamo

$$
\begin{aligned}
1.000 \cdot 10^{0} = 0.100 \cdot 10^{1} &= 0.010 \cdot 10^{2} = 0.001 \cdot 10^{3} \\
&= 0.000|1 \cdot 10^{4}.
\end{aligned}
$$

Gaussove eliminacije — primjer (nastavak)

Za zadnju jedinicu nema mjesta u mantisi, pa je mantisa postala 0, tj. prvi broj "nema" utjecaja na rezultat. Slično se dogodi i s desnom stranom (i 2 je "zanemariv" prema 10^4).

Dakle, nova druga jednadžba glasi

$$-1.000 \cdot 10^4 \, x_2 = -1.000 \cdot 10^4.$$

Rješenje ove jednadžbe je očito $x_2 = 1.000 \cdot 10^0$. Uvrštavanjem u prvu jednadžbu, dobivamo

$$\begin{aligned} 1.000 \cdot 10^{-4} \, x_1 &= 1.000 \cdot 10^0 - 1.000 \cdot 10^0 \cdot 1.000 \cdot 10^0 \\ &= 0.000 \cdot 10^0, \end{aligned}$$

pa je $x_1 = 0$, što nije niti približno točan rezultat.

Gaussove eliminacije — primjer (nastavak)

Razlog za ogromnu relativnu grešku (100%):

- prvu jednadžbu množimo velikim brojem -10^4 (po apsolutnoj vrijednosti) i dodajemo drugoj,
- što "uništava" drugu jednadžbu.

Drugim riječima, utjecaj polazne druge jednadžbe

- postaje zanemariv u novoj drugoj jednadžbi.

U polaznoj drugoj je moglo pisati "bilo što" (dovoljno malo)!

Isto bi nam se dogodilo za bilo koju drugu jednadžbu oblika

$$x_1 + \alpha x_2 = \beta,$$

gdje su $|\alpha|, |\beta| < 5$.

Gaussove eliminacije s pivotiranjem — primjer

Promijenimo li poredak jednadžbi, dobivamo

$$
\begin{aligned}
1.000 \cdot 10^0 \;\; x_1 + 1.000 \cdot 10^0 \, x_2 &= 2.000 \cdot 10^0 \\
1.000 \cdot 10^{-4} \, x_1 + 1.000 \cdot 10^0 \, x_2 &= 1.000 \cdot 10^0.
\end{aligned}
$$

Množenjem prve jednadžbe s -10^{-4} i dodavanjem drugoj, dobivamo novu drugu jednadžbu

$$(1.000 \cdot 10^0 - 1.000 \cdot 10^{-4}) \, x_2 = 1.000 \cdot 10^0 - 2.000 \cdot 10^{-4}.$$

Ovdje nema oduzimanja — drugi broj s 10^{-4} je “zanemariv” prema 1, na obje strane. Dakle, nova druga jednadžba sad glasi

$$1.000 \cdot 10^0 \, x_2 = 1.000 \cdot 10^0.$$

Gaussove eliminacije s pivotiranjem — primjer

Ponovno dobivamo rješenje $x_2 = 1.000 \cdot 10^0$. Međutim, uvrštavanjem u prvu jednadžbu dobivamo

$$
\begin{aligned}
1.000 \cdot 10^0\, x_1 &= 2.000 \cdot 10^0 - 1.000 \cdot 10^0 \cdot 1.000 \cdot 10^0 \\
&= 1.000 \cdot 10^0,
\end{aligned}
$$

pa je $x_1 = 1.000 \cdot 10^0$, što je točan rezultat — korektno zaokruženo egzaktno rješenje na četiri decimalne znamenke!

Razlog za vrlo malu relativnu grešku:

- prvu jednadžbu sad množimo malim brojem -10^{-4} (po apsolutnoj vrijednosti) i dodajemo drugoj,
- što nema utjecaja na drugu jednadžbu — tj. ovdje nema "uništavanja" jednadžbi.

Gaussove eliminacije s pivotiranjem — primjer

Kao i ranije, u koraku eliminacije,

- (bivša) druga jednadžba nema utjecaja na (bivšu) prvu.

Međutim, nakon zamjene

- prva jednadžba (bivša druga) ostaje netaknuta u prvom koraku eliminacije i uredno utječe na rješenje.

Zaključak: Sigurno nije dovoljno uzeti

- prvi (bilo koji) ne-nula element u stupcu

kao ključni element za eliminacije,

- jer možemo dobiti potpuno pogrešan rezultat.

Gaussove eliminacije s pivotiranjem — primjer

Primjer. Usporedimo izračunata rješenja sustava

$$\begin{aligned} \varepsilon\, x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2, \end{aligned}$$

za $\varepsilon = 10^{-1}, \dots, 10^{-25}$, Gaussovim eliminacijama bez zamjena i sa zamjenom poretka jednadžbi, u aritmetici računala.

Računanjem u najvećoj mogućoj preciznosti (`extended`) dobivamo sljedeću tablicu.

U tablici je x_2 naveden samo jednom — jer ga obje metode izračunaju jednako (i točno)!

U prvom stupcu pišu samo eksponenti p, pri čemu je $\varepsilon = 10^p$.

Gaussove eliminacije s pivotiranjem — primjer

p	x_1 bez pivotiranja	x_1 s pivotiranjem	x_2
-4	1.00010001000100000	1.00010001000100010	0.99989998999899990
-5	1.00001000010000200	1.00001000010000100	0.99998999989999900
-6	1.00000100000099609	1.00000100000100000	0.99999899999900000
-7	1.00000009999978538	1.00000010000001000	0.99999989999999000
$\vdots$	$\vdots$	$\vdots$	
-17	0.99746599868666408	1.00000000000000001	0.99999999999999999
-18	0.97578195523695399	1.00000000000000000	1.00000000000000000
-19	1.08420217248550443	1.00000000000000000	1.00000000000000000
-20	0.00000000000000000	1.00000000000000000	1.00000000000000000
$\vdots$	isto	isto	isto

Gaussove eliminacije s parcijalnim pivotiranjem

Standardni naziv: pivotni element = element koji se prije k-tog koraka eliminacije dovodi na dijagonalno mjesto $a_{kk}^{(k)}$.

U praksi se obično bira korištenjem parcijalnog pivotiranja.

- U k-tom koraku, pivotni element je po apsolutnoj vrijednosti najveći u "ostatku" k-tog stupca — na glavnoj dijagonali ili ispod nje.

Preciznije, ako je u k-tom koraku

$$|a_{rk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|,$$

onda ćemo prvo zamijeniti r-ti i k-ti redak, a zatim početi korak eliminacije elemenata k-tog stupca.

Gaussove eliminacije s parcijalnim pivotiranjem

Motivacija: elementi "ostatka" linearnog sustava koje treba izračunati u matrici $A^{(k+1)}$ u k-tom koraku transformacije su

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)},$$

za $i, j = k+1, \ldots, n$, a multiplikatori m_{ik} su

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1, \ldots, n.$$

Ako je multiplikator m_{ik} velik, u aritmetici računala može doći do kraćenja najmanje značajnih znamenki $a_{ij}^{(k)}$, tako da izračunati $a_{ij}^{(k+1)}$ može imati veliku relativnu grešku.

Gaussove eliminacije s parcijalnim pivotiranjem

Sasvim općenito, ideja pivotiranja je minimizirati "korekcije" elemenata pri prijelazu s $A^{(k)}$ na $A^{(k+1)}$. Dakle,

- multiplikatori m_{ik} trebaju biti što manji, po apsolutnoj vrijednosti.

Ekvivalentno, pivotni element treba biti što veći, jer ulazi u nazivnik multiplikatora.

Za multiplikatore kod parcijalnog pivotiranja vrijedi

$$|m_{ik}| \leq 1, \quad i = k+1, \ldots, n.$$

U praksi, parcijalno pivotiranje funkcionira izvrsno, ali matematičari su konstruirali primjere kad ono "nije savršeno" (v. malo kasnije).

Gaussove eliminacije s potpunim pivotiranjem

Osim parcijalnog, može se provoditi i potpuno pivotiranje.

- U k-tom koraku, bira se najveći element u cijelom "ostatku" matrice $A^{(k)}$, a ne samo u k-tom stupcu.

Ako je u k-tom koraku

$$|a_{rs}^{(k)}| = \max_{k \leq i,j \leq n} |a_{ij}^{(k)}|,$$

onda ćemo prvo zamijeniti r-ti i k-ti redak, s-ti i k-ti stupac, a zatim početi korak eliminacije elemenata k-tog stupca.

Oprez: zamjenom s-tog i k-tog stupca zamijenili smo ulogu nepoznanica (varijabli) x_s i x_k.

Ovo nisu jedine mogućnosti pivotiranja kod rješavanja linearnih sustava.