KTH Engineering Sciences

Fel- och störningsanalys

1 Terminologi

Antag att x är ett exakt värde och $\tilde{x}$ är en approximation av x. Vi kallar då

$$\text{absoluta felet i } \tilde{x} \ = \tilde{x} - x, \qquad \text{relativa felet i } \tilde{x} \ = \frac{\tilde{x} - x}{x}.$$

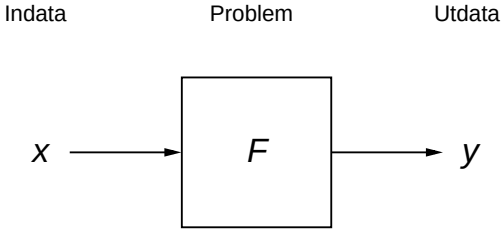
Ofta känner vi inte felet precis utan vet bara att det är mindre än en *felgräns*. Felgränsen kan vara antingen absolut, betecknad E_x, eller relativ, betecknad R_x, och innebär att

$$|\tilde{x} - x| \leq E_x, \qquad \left|\frac{\tilde{x} - x}{x}\right| \leq R_x.$$

Vi skriver ofta detta på formen $x = 2 \pm 0.1$ och menar då att $\tilde{x} = 2$ och $E_x = 0.1$. Om $\tilde{x}$ ges som ett korrekt avrundat decimaltal utskrivet med n siffror (på endera sidan av decimalkommat, men inledande nollor räknas inte) säger vi att $\tilde{x}$ har n korrekta, eller *signifikanta*, siffror. Exempelvis har 132,13 fem signifikanta siffror och 0,0311 har tre signifikanta siffror. Begreppet är nära relaterat till det relativa felet.[1]

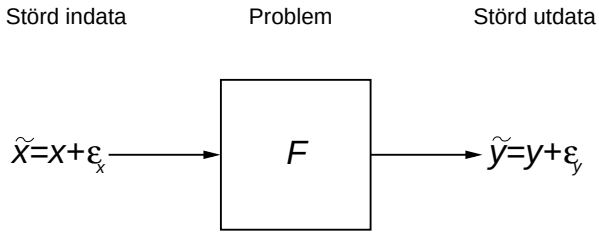
2 Störningsanalys

Indata till ett numeriskt problem innehåller i praktiken alltid (små) fel. Felen kan bero på tex mätfel, avrundningsfel eller pga att indata kommer från andra numeriska beräkningar som innehåller trunkationsfel. En viktig frågeställning är därför hur lösningen till problemet påverkas av dessa fel. Vi kan beskriva situationen såhär:



Här är F en abstrakt formulering av problemet. Notera att F är en funktion i matematisk mening, dvs att för varje indata x finns precis ett utdata y, så att $y = F(x)$. I allmänhet är dock F mycket komplicerad och kan inte skrivas i sluten form. Det kan tex vara lösningen till ett stort sammansatt ingenjörsproblem där x är en vektor med en mängd inparametrar. Frågan vi är intresserad av är vad som händer med utdata y om vi stör indata x med ett litet fel ε_x. Istället för x skickar vi in $\tilde{x} = x + \varepsilon_x$ och istället för y får vi ut $\tilde{y} = y + \varepsilon_y$. Hur beror då ε_y på ε_x?

[1] Genom att skriva om talet och felgränsen i normalform, tex $x = 0.00311 \pm 0.000005 = 3.11 \times 10^{-3} \pm 0.5 \times 10^{-5}$, kan man se att om talet har n signifikanta siffror maste relativa felet ligga mellan 0.5×10^{-n} och 5×10^{-n}. Vill man vara precis är det bättre att ange relativt fel än antal signifikanta siffror.



Felet i utdata ε_y kallas ibland "forward error", eftersom det motsvarar felet som propagerar framåt från indata till utdata. På motsvarande sätt kallas felet i indata ε_x ibland "backward error". Då tänker man sig att detta är felet som måste läggas på indata för att skapa ett givet fel i utdata.

2.1 Felfortplantning

Om F är en snäll funktion kan vi analysera "felfortplantningen" från indata till utdata med hjälp av lokal linjärisering runt $\tilde{x}$. I detta fall är $y = F(x)$ och $\tilde{y} = F(\tilde{x})$. Via taylorutvecklingen

$$F(x) = F(\tilde{x} - \varepsilon_x) = F(\tilde{x}) - \varepsilon_x F'(\tilde{x}) + \mathcal{O}(\varepsilon_x^2)$$

får vi om $\varepsilon_x \ll 1$,

$$\varepsilon_y = \tilde{y} - y = F(\tilde{x}) - F(x) = \varepsilon_x F'(\tilde{x}) + \mathcal{O}(\varepsilon_x^2) \quad \Rightarrow \quad \boxed{\varepsilon_y \approx \varepsilon_x F'(\tilde{x}).}$$

Detta ger direkt motsvarande formel för absoluta felgränserna E_x och E_y. Om $\varepsilon_x \in [-E_x, E_x]$ har vi att

$$\varepsilon_y \approx \varepsilon_x\, F'(\tilde{x}) \in \big[-E_x|F'(\tilde{x})|,\ E_x|F'(\tilde{x})|\big] \quad \Rightarrow \quad \boxed{E_y \approx E_x|F'(\tilde{x})|.}$$

Notera att F' här evalueras vid det approximativa värdet $\tilde{x}$. Sambandet gäller även om F' evalueras vid x men oftast är det $\tilde{x}$ som är tillgänglig, inte x.

Om F beror på flera variabler,

$$y = F(x_1, x_2, \ldots, x_n), \qquad \tilde{y} = F(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n),$$

får man på samma sätt

$$\boxed{\varepsilon_y \approx \varepsilon_{x_1} \tfrac{\partial F(\tilde{x}_1,\ldots,\tilde{x}_n)}{\partial x_1} + \cdots + \varepsilon_{x_n} \tfrac{\partial F(\tilde{x}_1,\ldots,\tilde{x}_n)}{\partial x_n},}$$

där ε_{x_j} är felet i x_j, samt

$$\boxed{E_y \approx E_{x_1} \left|\tfrac{\partial F(\tilde{x}_1,\ldots,\tilde{x}_n)}{\partial x_1}\right| + \cdots + E_{x_n} \left|\tfrac{\partial F(\tilde{x}_1,\ldots,\tilde{x}_n)}{\partial x_n}\right|},$$

där E_{x_j} är absoluta felgränsen för ε_{x_j}.

Exempel 1: Antag att $F(x) = 1 + \cos x$ och att indata x är given som $x = 0.1 \pm 0.005$. Med terminologin ovan är då $\tilde{x} = 0.1$ och $E_x = 0.005$. Beräkna först $\tilde{y}$,

$$\tilde{y} = 1 + \cos(0.1) \approx 0.0499958$$

Enligt formeln ovan blir felgränsen för $\tilde{y}$

$$E_y \approx E_x|F'(\tilde{x})| = 0.005 \cdot \sin(0.1) \approx 0.5 \cdot 10^{-3}.$$

Vi avrundar därför $\tilde{y}$ till tre decimaler och skriver

$$y = 0.050 \pm 0.0005.$$

Exempel 2: Låt x vara exakta roten till ekvationen $f(x) = 0$. Låt $\tilde{x}$ vara en approximation till roten given av Newtons metod med avbrottskriteriet

$$|f(\tilde{x})| \leq \delta,$$

där toleransen δ är ett litet tal. Vi vill beräkna felgränsen i $\tilde{x}$. I detta exempel är $y = f(x) = 0$ och $\tilde{y} = f(\tilde{x})$. Följdaktligen är $|\varepsilon_y| = |\tilde{y} - y| = |f(\tilde{x}) - 0| \leq \delta$, dvs $E_y = \delta$. Formeln för felgränserna ger då

$$E_y \approx E_x |f'(\tilde{x})| \quad \Rightarrow \quad E_x \approx \frac{\delta}{|f'(\tilde{x})|}.$$

Notera att om $f'(\tilde{x})$ är liten kan felet i $\tilde{x}$ vara stort även om toleransen δ är liten. Det är därför normalt svårt att lösa ekvationer med dubbelrötter där $f(x) = f'(x) = 0$.

Exempel 3: Antag att x är en rot till $\sin(ax) = x/2$, där parametern a inte är känd exakt utan med en viss osäkerhet: $a = 1 \pm 0.1$. Vad blir osäkerheten i lösningen x?

Här är a indata och $x = x(a)$ utdata.[2] Vi har $\tilde{a} = 1$ och felgränsen $E_a = 0.1$. Vi låter $\tilde{x} = x(\tilde{a}) = x(1)$, dvs $\sin(\tilde{x}) = \tilde{x}/2$, och vi vill nu räkna ut felgränsen E_x så att $x = \tilde{x} \pm E_x$. För att kunna använda felfortplantningsformeln ovan behöver vi derivatan dx/da, evaluerad i $\tilde{a} = 1$. Implicit derivering av $\sin(ax) - x = 0$ med avseende på a ger

$$x(a)\cos(ax(a)) + a\frac{dx(a)}{da}\cos(ax(a)) - \frac{1}{2}\frac{dx(a)}{da} = 0 \quad \Rightarrow \quad \frac{dx(a)}{da} = \frac{x(a)\cos(ax(a))}{\frac{1}{2} - a\cos(ax(a))}.$$

Eftersom $x(\tilde{a}) = \tilde{x}$ får vi

$$\frac{dx(\tilde{a})}{da} = \frac{x(\tilde{a})\cos(x(\tilde{a}))}{\frac{1}{2} - \cos(x(\tilde{a}))} = \frac{\tilde{x}\cos(\tilde{x})}{\frac{1}{2} - \cos(\tilde{x})}.$$

Och därmed

$$x = \tilde{x} \pm E_x, \qquad E_x \approx E_a \left|\frac{dx(\tilde{a})}{da}\right| = 0.1\frac{\tilde{x}\cos(\tilde{x})}{\frac{1}{2} - \cos(\tilde{x})}.$$

Med insatta siffror har vi tex $\tilde{x} \approx 1.895494267033981$ och $x'(1) \approx -0.738325684266218$, varför ett lämpligt avrundat svar är

$$x = 1.895 \pm 0.074.$$

2.2 Experimentell störningsanalys

I många fall är en sluten form för F inte känd, eller för komplicerad för att kunna deriveras. Det är då svårt att använda analysen ovan för att uppskatta felen i utdata. Exempelvis skulle $F(x)$ kunna vara given som lösningen till en ordinär differentialekvation vid en viss tid med begynnelsevärde x:

$$\frac{dy}{dt} = g(y), \qquad y(0) = x, \qquad F(x) := y(10).$$

I denna situation är det mer praktiskt att använda "experimentell" störningsanalys där $F(x)$ betraktas som en "svart låda". Det enda vi antar är att F är två gånger kontinuerligt deriverbar.

[2] Funktionen $x(a)$ är implicit given av sambandet $F(a, x) = \sin(ax) - x/2 = 0$. Den är väldefinerad i en omgivning av (a, x) så länge som $F'_x(a, x) = a\cos(ax) - 1/2 \neq 0$, enligt implicita funktionssatsen. Vi antar att detta gäller vid den aktuella roten.

Vi börjar med att beräkna $\tilde{y}$ från störda indata $\tilde{x}$ som tidigare. Sedan gör vi en "experimenträkning": en beräkning där vi medvetet stör indata ytterligare med värdet på indatafelgränsen E_x. Resultatet kallar vi y_{exp},

$$\tilde{y} = F(\tilde{x}), \qquad y_{\text{exp}} = F(\tilde{x} + E_x).$$

Skillnaden $|y_{\text{exp}} - \tilde{y}|$ är då en bra uppskattning av felgränsen E_y i utdata $\tilde{y}$ eftersom

$$|y_{\text{exp}} - \tilde{y}| = |F(\tilde{x} + E_x) - F(\tilde{x})| = |E_x F'(\tilde{x}) + \mathcal{O}(E_x^2)| \approx E_x |F'(\tilde{x})| \approx E_y.$$

Notera: Vi hade lika gärna kunnat avnända $F(\tilde{x} - E_x)$ istället för $F(\tilde{x} + E_x)$ när vi beräknade y_{exp}. Resultatet hade blivit detsamma, modulo $\mathcal{O}(E_x^2)$.

Exempel 4: Vi kan lösa Exempel 3 med hjälp av experimentell störningsräkning istället för att använda felfortplantingsformeln. Vi måste då beräkna $x(1)$ och $x_{\text{exp}} = x(1 + E_a)$. Felet i x ges sedan approximativt av $E_x \approx |x(1) - x(1 + E_a)|$. Vi noterar att vi redan har $x(1) = \tilde{x} \approx 1.895494267033981$ given ovan och $x_{\text{exp}} = x(1.1)$ är en rot till $\sin(1.1x) - x/2 = 0$. Vi kan lösa denna ekvation med tex fixpunktiteration eller Newtons metod. Det ger $x_{\text{exp}} \approx 1.818397416908629$. Resultatet blir då ungefär detsamma som ovan:

$$E_x \approx 0.0771.$$

Skillnaderna i värdena beror på att vi försummat högre ordningens termer i våra approximationer.

När F beror av två variabler stör vi dem en i sänder och beräknar,

$$y_{\text{exp},1} = F(x_1 + E_{x_1}, x_2), \qquad y_{\text{exp},2} = F(x_1, x_2 + E_{x_2}).$$

Felet i utdata uppskattas med

$$E_y \approx |\tilde{y} - y_{\text{exp},1}| + |\tilde{y} - y_{\text{exp},2}|.$$

Proceduren kan lätt generaliseras till funktioner av fler variabler. Om $y = F(x_1, x_2, \ldots, x_n)$ och $x_j = \tilde{x}_j \pm E_{x_j}$ gör vi

1. Beräkna $\tilde{y} = F(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$.
2. Stör x-variablerna en i sänder med sin felgräns och beräkna motsvarande störda y-värden:

$$\begin{aligned}
\tilde{y}_1 &= F(\tilde{x}_1 + E_{x_1}, \tilde{x}_2, \ldots, \tilde{x}_n), \\
\tilde{y}_2 &= F(\tilde{x}_1, \tilde{x}_2 + E_{x_2}, \ldots, \tilde{x}_n), \\
&\vdots \qquad\quad \vdots \\
\tilde{y}_n &= F(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n + E_{x_n}).
\end{aligned}$$

3. Uppskatta E_y som
$$E_y \approx |\tilde{y}_1 - \tilde{y}| + |\tilde{y}_2 - \tilde{y}| + \cdots + |\tilde{y}_n - \tilde{y}|.$$

Se tex ENM uppgift 8.8 för exempel.

Kommentar. Man skulle också kunna beräkna y för alla kombinationer av störningar. För två variabler skulle det bli:

$$\begin{aligned}
\tilde{y}_1 &= F(\tilde{x}_1 + E_{x_1}, \tilde{x}_2),\\
\tilde{y}_2 &= F(\tilde{x}_1 - E_{x_1}, \tilde{x}_2),\\
\tilde{y}_3 &= F(\tilde{x}_1 + E_{x_1}, \tilde{x}_2 + E_{x_2}),\\
\tilde{y}_4 &= F(\tilde{x}_1 - E_{x_1}, \tilde{x}_2 + E_{x_2}),\\
\tilde{y}_5 &= F(\tilde{x}_1 + E_{x_1}, \tilde{x}_2 - E_{x_2}),\\
\tilde{y}_6 &= F(\tilde{x}_1 - E_{x_1}, \tilde{x}_2 - E_{x_2}),\\
\tilde{y}_7 &= F(\tilde{x}_1, \tilde{x}_2 + E_{x_2}),\\
\tilde{y}_8 &= F(\tilde{x}_1, \tilde{x}_2 - E_{x_2}).
\end{aligned}$$

Felgränsen skulle sedan uppskattas med

$$E_y \approx \max_{1\le j\le 8} |\tilde{y} - \tilde{y}_j|.$$

Detta kan vara en mer noggrann metod om F varierar snabbt eller om felgränserna är stora. Det är dock en mycket dyrare metod när man har många variabler, dvs n stort. Antal funktionsevalueringar för att få med alla kombinationer är 3^n (inklusive beräkningen av $\tilde{y}$), jämfört med bara $n+1$ med den tidigare metoden. Om felgränserna är små kommer båda metoderna också ge samma svar, modulo $O(E_{x_j}^2)$, vilket ges av Taylorutveckling som tidigare.

3 Kondition och rättställdhet

Om felet i utdata är litet så snart felet i indata är litet brukar man säga att problemet är *välkonditionerat* eller *stabilt*; små fel i indata "spelar ingen större roll." Om felet i utdata kan bli stort även för relativt små fel i indata säger man att problemet är *illa konditionerat*. Den maximala förstärkningen av små fel i indata beskrivs av problemets *konditionstal* κ. Det absoluta konditionstalet κ_a är det största värdet på $|\varepsilon_y/\varepsilon_x|$ när $\varepsilon_x \ll 1$ varierar. En precision definition är $\kappa_a = \lim_{\delta\to 0} \max_{|\varepsilon_x|\le\delta} |\varepsilon_y/\varepsilon_x|$. Från beräkningarna ovan framgår att $\kappa_a = |F'(x)|$ när F är en snäll funktion. Ofta är man mer intresserad av det relativa konditionstalet κ_r, definierat som maximala förstärkningen av relativa felet, dvs $|\varepsilon_y/y|/|\varepsilon_x/x|$ när $\varepsilon_x \ll 1$ varierar. Den precisa definition är

$$\kappa_r = \lim_{\delta\to 0} \max_{|\varepsilon_x|\le\delta} \frac{|\varepsilon_y/y|}{|\varepsilon_x/x|}.$$

När F är en snäll funktion har vi då

$$\kappa_r = \frac{|\varepsilon_x F'(x)|/F(x)|}{|\varepsilon_x/x|} = \frac{|xF'(x)|}{|F(x)|}.$$

Vi har hittills hela tiden antagit att F motsvarar en snäll funktion, i själva verket en två gånger deriverbar funktion. Om F istället tex är en diskontinuerlig funktion kan ε_y bli stort även för godtyckligt små ε_x. Det betyder att även om vi har mycket små störningar i indata får vi ett utvärde som är helt fel. Formellt blir konditionstalet κ oändligt stort. Sådana problem kallas *icke rättställda* (ill-posed). Omvänt kallas problem där $F(x)$ är kontinuerlig för *rättställda* (well-posed) problem. Det är viktigt att se konsekvensen av icke rättställdhet: Eftersom vi i praktiken alltid har små störningar i indata *kan icke rättställda problem inte lösas med numeriska metoder* även om problemet i teorin har en entydig lösning. Begreppet rättställdhet är därför

centralt inom tillämpad matematik. Ett mycket enkelt exempel på ett icke rättställt problem är följande. Lös $f(x) = 0$ när

$$f(x) = \begin{cases} x, & x \neq 0,\ x \neq 1, \\ 1, & x = 0, \\ 0, & x = 1. \end{cases}$$

Det är uppenbart att den enda lösningen är $x = 1$, men en numerisk metod kommer inte kunna hitta den roten. Den numeriska svårigheten är ganska tydlig i det här enkla fallet. Den kan dock vara betydligt mer subtil i mer komplicerade problem. Speciellt svår är frågan om rättställdhet för partiella differentialekvationer där även till synes enkla ekvationer kan vara icke rättställda. I det fallet är både in- och utdata *funktioner* snarare än skalära tal eller vektorer.

4 Avrundningsfel

Beräkningar i en dator görs med ändlig precision. Det betyder att alla tal som lagras i datorn har ett litet *avrundningsfel*. Det betyder också att varje operation, som addition, multiplikation, etc. introducerar ett litet fel eftersom svaret måste lagras med ändlig precision. Oftast är avrundningsfelen insignifikanta. I dubbel precision, som MATLAB använder, är tex det relativa felet som härrör från avrundning mindre än $2^{-52} \approx 10^{-16}$ (kallas *maskinprecisionen*). MATLAB har därför normalt ca 15 korrekta siffror i sina beräkningar. Det finns dock ett par fall som man bör känna till, då effekten av avrundningsfel kan vara betydande:

- Subtraktion av två nästan lika stora tal.

  Detta kan leda till en stor reduktion av noggrannheten. Fenomenet kallas *kancellation*. Om vi låter X indikera osäkra siffror i en decimalutveckling, kan problemet illustreras med följande exempel:

```
   1.70604608801960XXXX
-  1.70604608801637XXXX
------------------------
   0.00000000000323XXXX
```

  De två talen ≈ 1.706 har båda 15 korrekta siffror, men differensen får bara tre korrekta siffror.

  Exempel 5: Antag att vi vill lösa andragradsekvationen $x^2 + 9^7 x + 3 = 0$. Vi vet att rötterna ges av

$$x_1 = -\frac{1}{2}\left(9^7 + \sqrt{9^{14} - 12}\right), \qquad x_2 = -\frac{1}{2}\left(9^7 - \sqrt{9^{14} - 12}\right).$$

  MATLAB ger följande svar för den andra roten

```
>> -(9^7-sqrt(9^14-12))/2

ans =

    -6.272457540035248e-07
```

Lösningen är i själva verket $\approx -6.272254743863892 \cdot 10^{-7}$. MATLABs svar har alltså inte mer än 4 korrekta siffror. Anledningen är att $9^7 = 4782969$ och $\sqrt{9^{14} - 12} \approx 4782968.999998746$ är nästan lika stora. Se Sauer kap 0.4 för en diskussion om hur problemet i det här fallet kan avhjälpas.

- Addition/subtraktion av tal som är av olika storleksordning.

  Den resulterande noggrannheten bestäms helt av det stora talet. Noggrannheten i det lilla talet försvinner. Fenomenet kallas *utskiftning* ("swamping" i Sauer). Med samma notation som ovan har vi tex

```
   5.00000000000000XXXX
+  0.00000000003017131781235 84XXXX
-----------------------------------
   5.00000000000301XXXX
```

  Exempel 6: I MATLAB får vi

```
>> a=5+3.142442387309e-14

a =

   5.000000000000031

>> a-5

ans =

     3.108624468950438e-14
```

  De sista resultatet har bara två korrekta siffror eftersom noggrannheten förstördes vid utskiftningen i första additionen.

I praktiska beräkningar är det i huvudsak vid lösning av stora linjära ekvationssystem med Gausselimination som ovanstående fenomen, speciellt utskiftning, kan ställa till problem. Strukturen där dikterar ibland att rader ska multipliceras med väldigt stora tal och därefter subtraheras från andra rader, vilket leder till försämrad noggrannhet. Ett sätt att undvika detta är att använda Gauss-elimination med så kallad *pivotering*. Se Sauer kap 2.3.2 och kap 2.4.1.