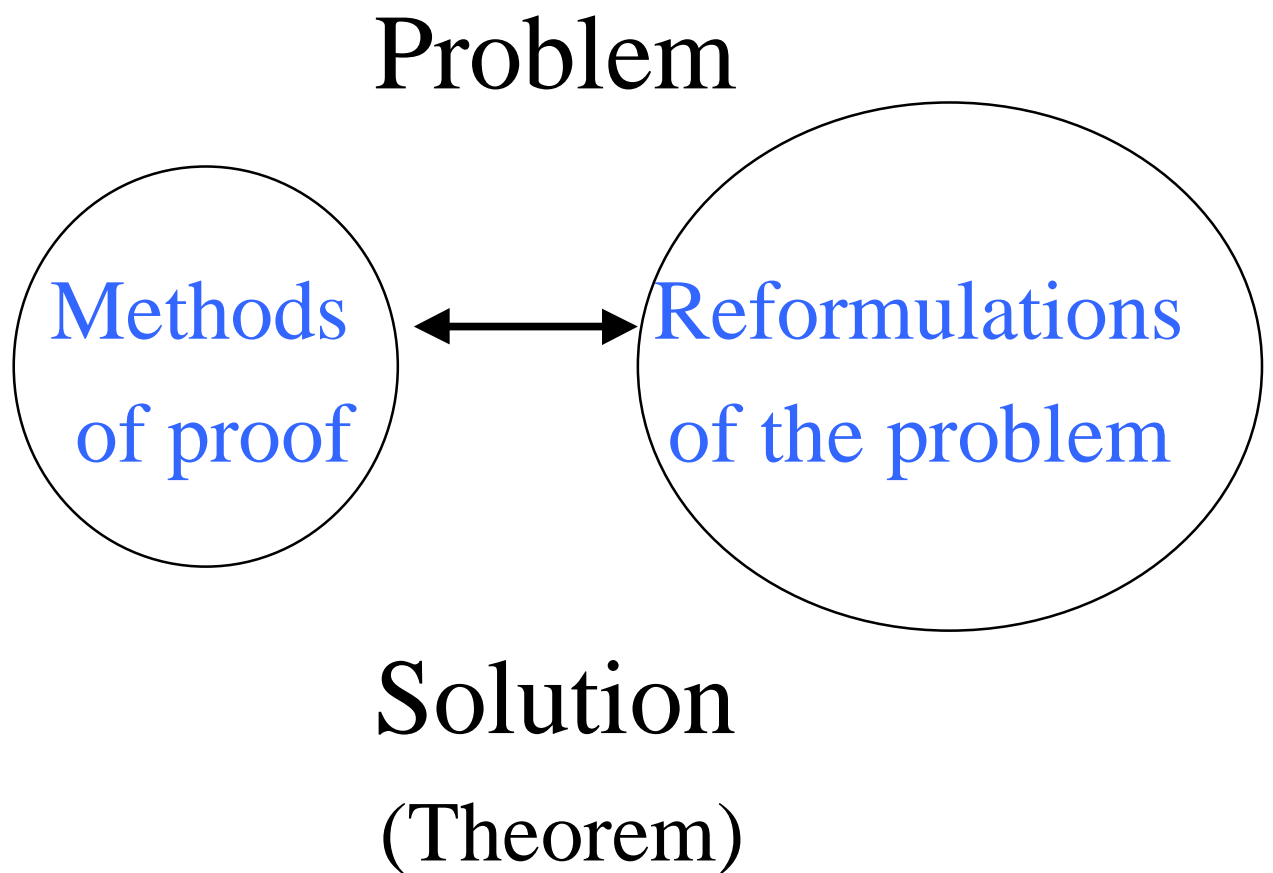


An Introduction to the Method of Computer-Assisted Proofs in Analysis

Peter Wittwer

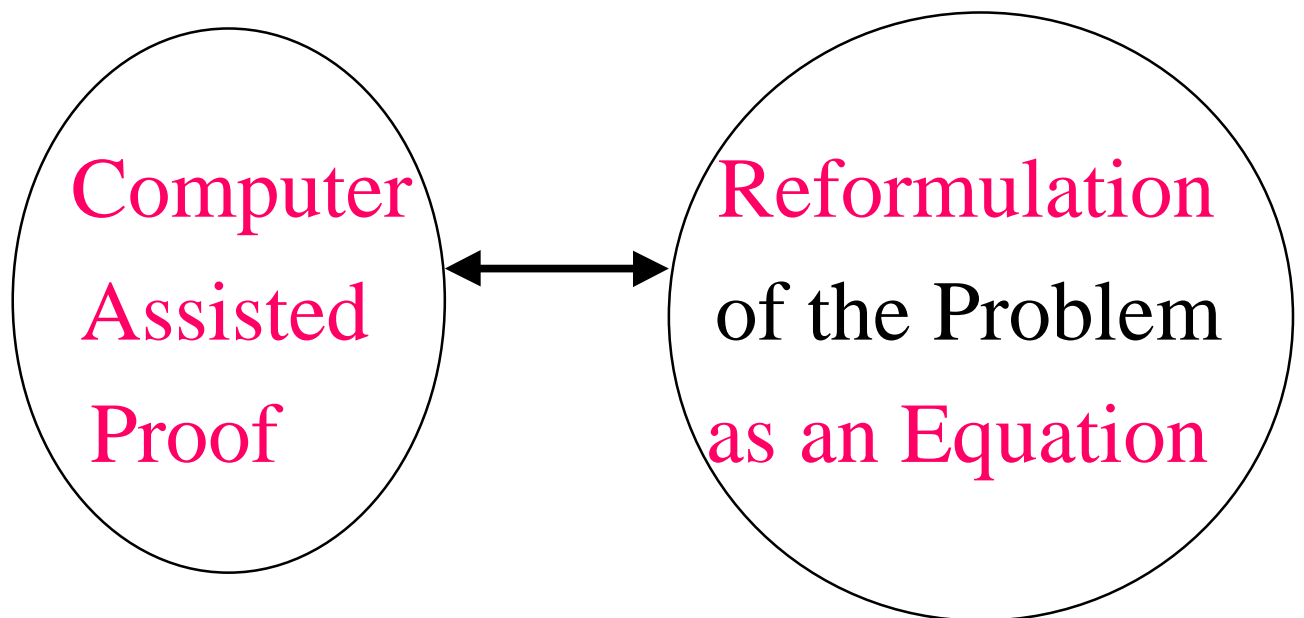
Université de Genève

How do we prove theorems?



What is a Computer- Assisted Proof?

Problem



Solution
(Theorem)

What Kind of Equations?

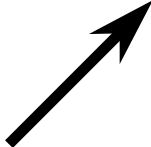
Prove existence of
solutions for:

$$N(f) = f$$

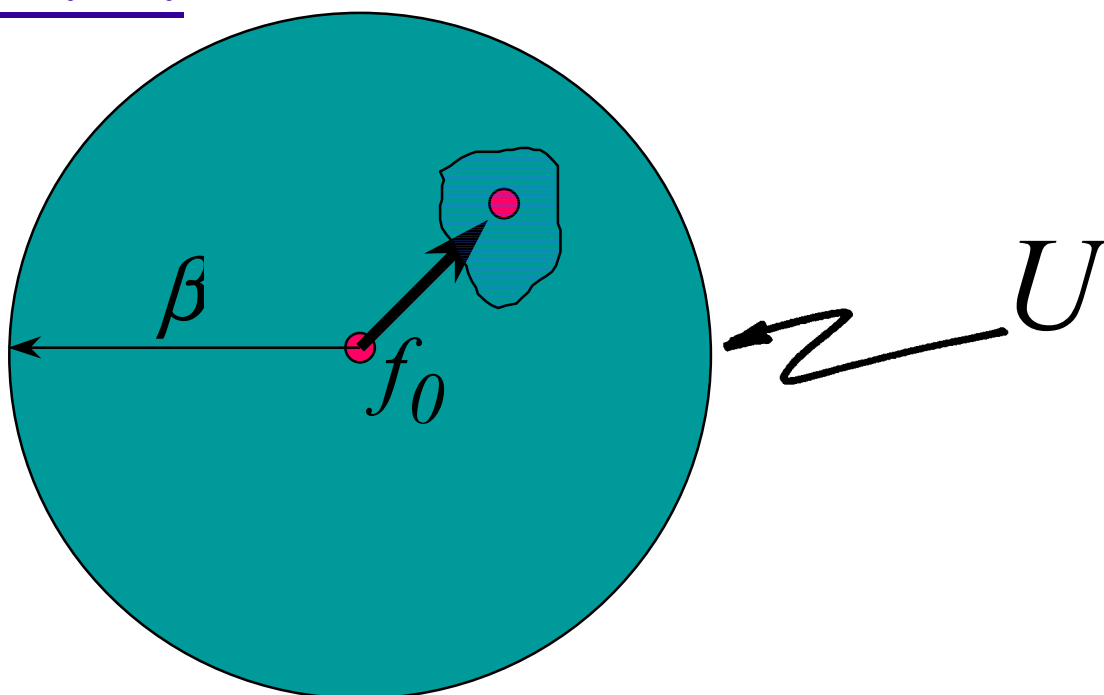
$f \in B$ Banach space

$DN(f)$ reasonable

C.M.P. can be put into
practice near the solution



C.M.P.



$$\|DM(f)h\| \leq \rho < 1$$

where $\{f \in U\}$ and $\{h \in B \mid \|h\| \leq 1\}$

$$\|M(f_0) - f_0\| \leq \varepsilon < (1 - \rho)\beta$$

Compute with Sets

How to Proceed

- f_0 approximate solution
- $M(f) = f - L(N(f) - f)$

$$L \approx (DN(f_0) - 1)^{-1}$$

$$DM(f) \approx 0$$

- hypothesis of the **C.M.P.** satisfied

“Building Blocks”

- Decompose

$$M = M_1 \circ M_2 \cdots$$

- Bound “Factors”

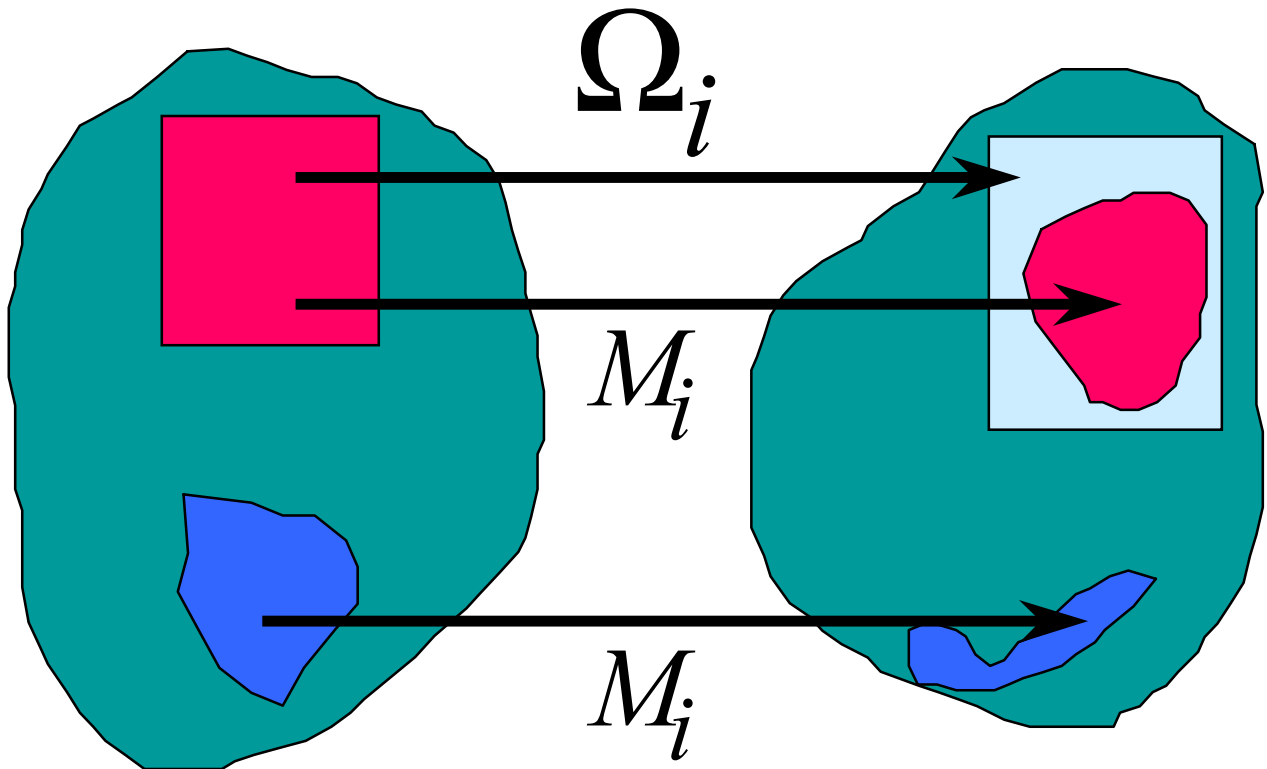
$$M_i, \quad DM_i$$

- Reassemble

$$M = M_1 \circ M_2 \cdots$$

$$DM = \text{apply chain rule}$$

Bound = Set-Map



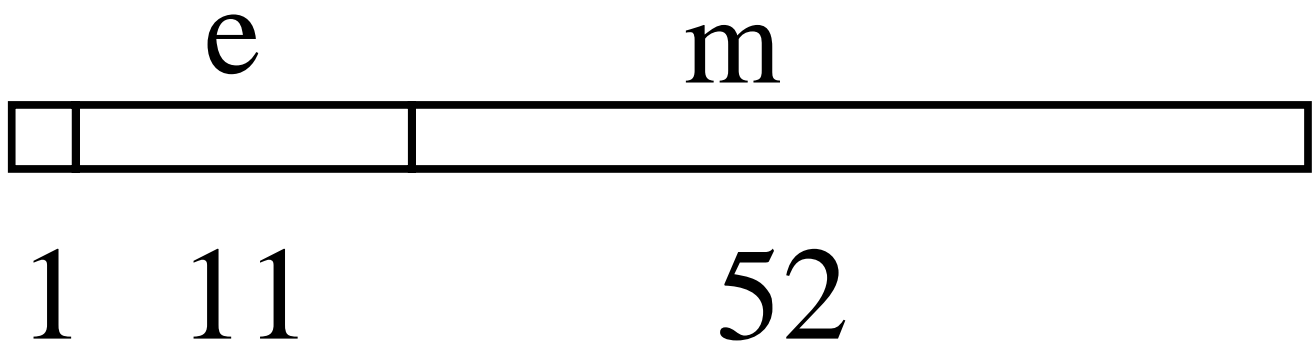
Def.: $\Omega_i \geq M_i$

$$D(\Omega_i) \subseteq D(M_i)$$

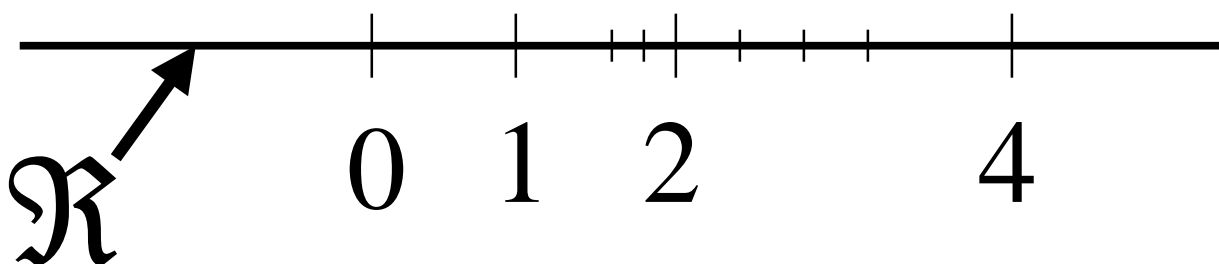
- $\Omega_i(S) \supseteq M_i(S)$

$$\forall S \in D(\Omega_i)$$

Floating Point Arithmetic (IEEE)



$$r = \pm 1.m 2^{e-1023}$$



$$r_1 \oplus_C r_2 = \text{rounding}(r_1 \oplus r_2)$$

Standard Sets for \mathbf{R}

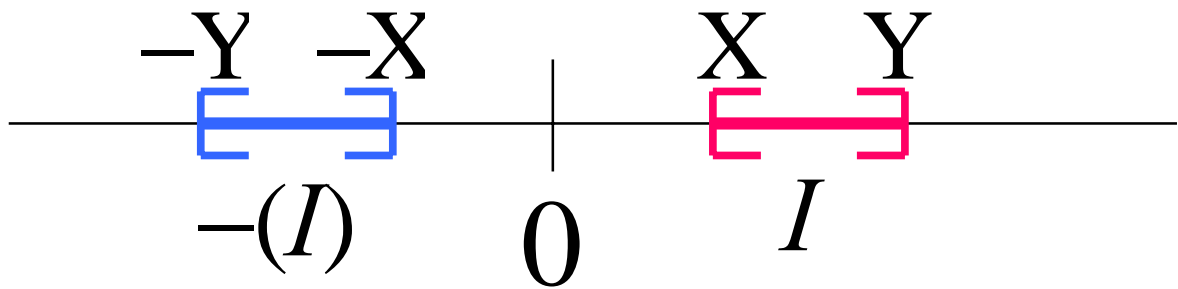
$\text{std}(\mathbf{R})$ is the collection of all closed intervals $i(X, Y)$ of the form

$$i(X, Y) = \{r \in \mathbf{R} \mid X \leq r \leq Y\}$$

with $X \leq Y$ elements of \mathfrak{R}

Example of a Bound

$-I = \{r \in \mathbb{R} \mid r = -x \text{ for some } x \in I\}$.



**$i(X1, Y1)$ contains $-i(X, Y)$:-
 $X1$ is $-Y$,
 $Y1$ is $-X$,
!.**

↗
Prolog

Bounds on + and *

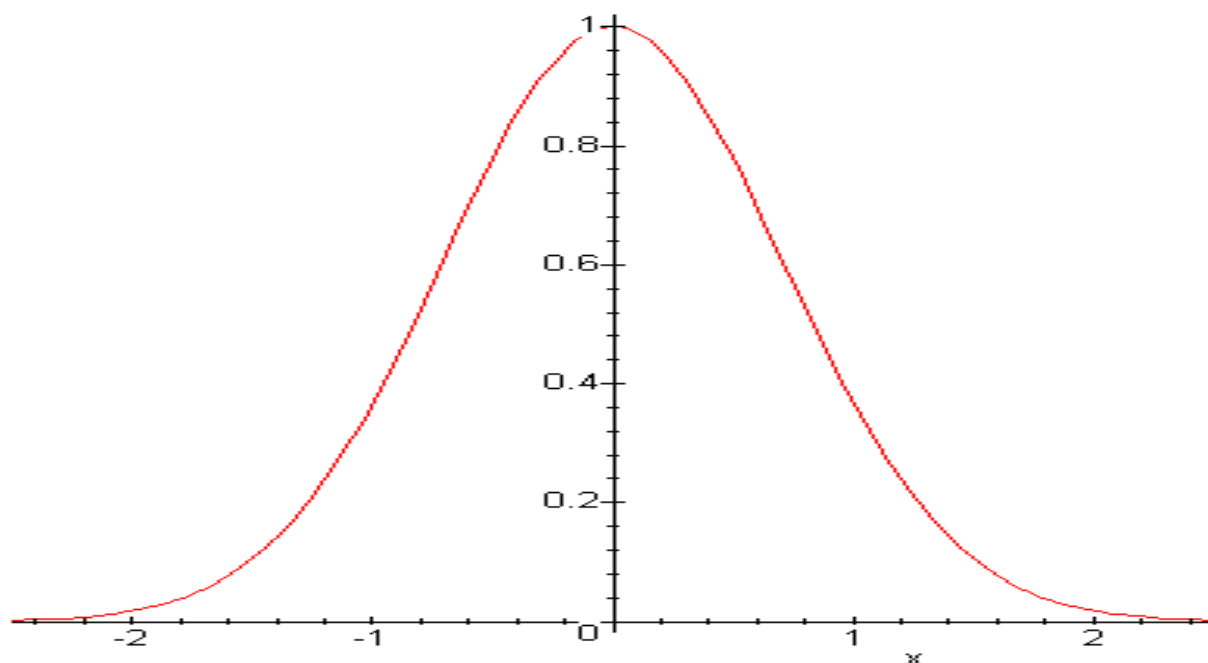
**$i(X_1, Y_1)$ contains $i(X_2, Y_2) + i(X_3, Y_3)$:-
X is $X_2 + X_3$,
Y is $Y_2 + Y_3$,
 $i(X_1, Y_1)$ enlarges $i(X, Y)$,
!.**

**$i(X_1, Y_1)$ contains $i(X_2, Y_2) * i(X_3, Y_3)$:-
A is $X_2 * X_3$,
B is $X_2 * Y_3$,
C is $Y_2 * X_3$,
D is $Y_2 * Y_3$,
 $\text{sort_numbers}([A, B, C, D], [X, _, _, Y])$,
 $i(X_1, Y_1)$ enlarges $i(X, Y)$,
!.**

Stable Distributions

$$R = R_1 + R_2$$

$$f(x) = \frac{1}{\sqrt{\pi}} \exp(-x^2)$$



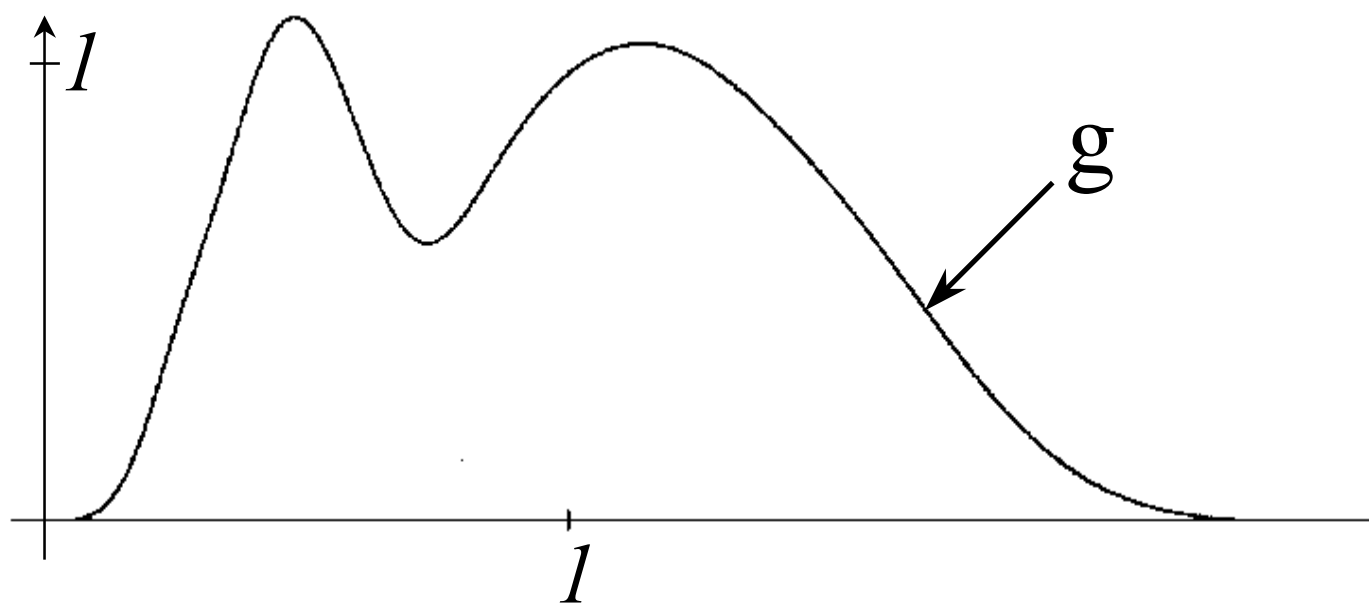
$$f(x) = \sqrt{2} f^{*2}(\sqrt{2}x)$$

Alain Schenker \swarrow Jan Wehr

Theorem [SWW]

$$R = \frac{1}{\frac{1}{R_1 + R_2} + \frac{1}{R_3 + R_4}}$$

$$f(x) = a \cdot \delta_\infty(x) + (1-a)g(x)$$



$$g(x) = \mu(c_1 g^{*2} + c_2 T(T(g^{*2})^{*2})(\mu x))$$

$$Tg(x) = \frac{1}{x^2} g\left(\frac{1}{x}\right)$$

Constants

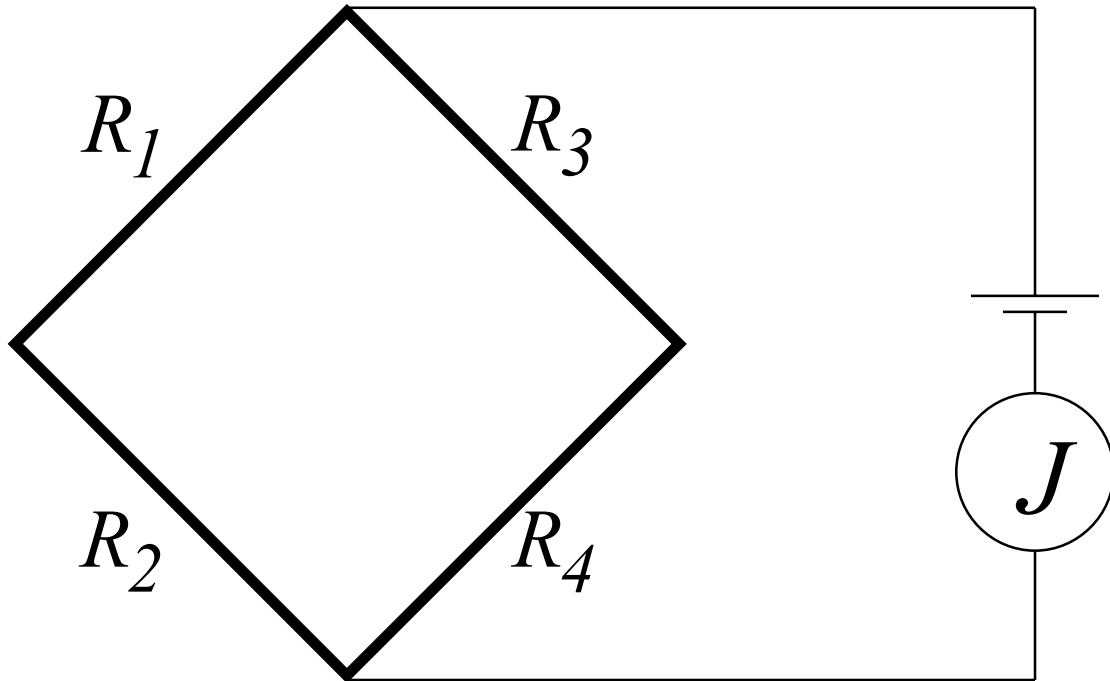
$$a = 0.618\dots = (\sqrt{5} - 1) / 2$$

$$c_1 = 1 - a^3$$

$$c_2 = a^3$$

$$\exists \mu \in [1.7562036, 1.7562047]$$

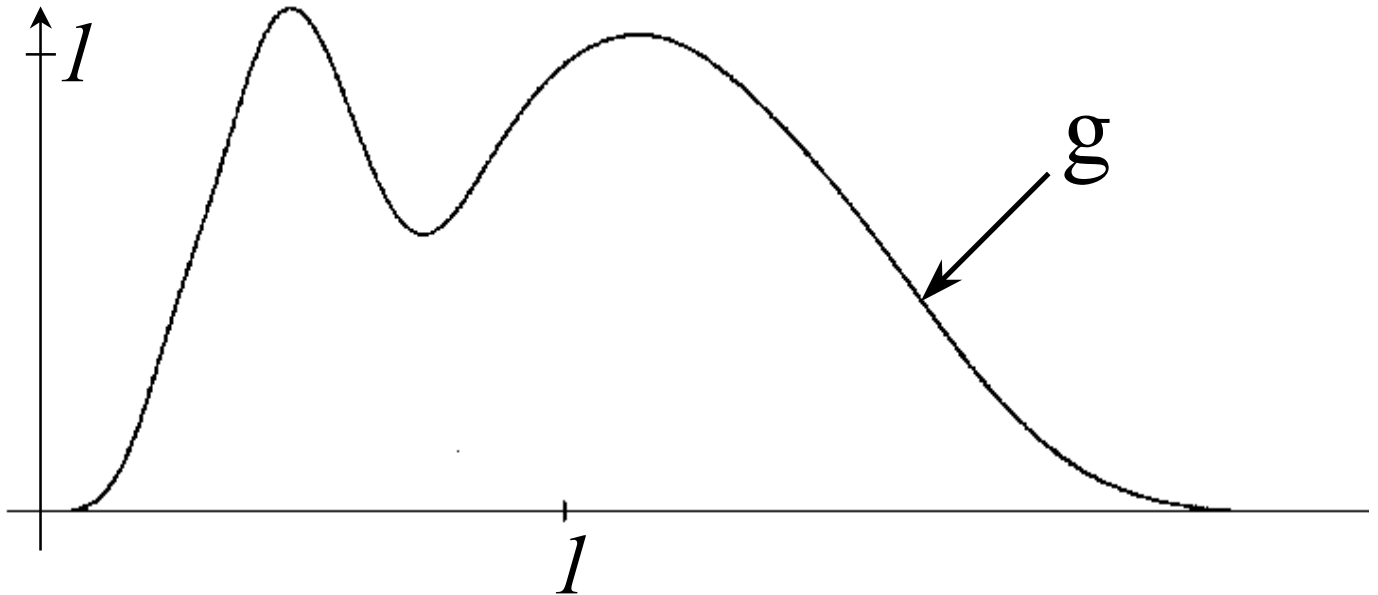
Motivation



Kirchhoff 's laws:

$$R = \frac{1}{\frac{1}{R_1 + R_2} + \frac{1}{R_3 + R_4}}$$

Choice of Function Space



$$Ng(x) = \mu(c_1 g^{*2} + c_2 T(T(g^{*2})^{*2}))(\mu x)$$

$$Tg(x) = \frac{1}{x^2} g\left(\frac{1}{x}\right) \quad \mu = 1.756\dots$$

$$B_{\alpha,\beta} = L_1(R^+, w_{\alpha,\beta}(x) dx)$$

$$w_{\alpha,\beta}(x) = \exp(\alpha/x + \beta x)$$

Decomposition

$$\begin{aligned} N(g)(x) &= \mu(c_1(g^{*2}) + c_2T(T(g^{*2})^{*2})(\mu x)) \\ &= ((S \circ A \circ M \circ \Gamma \circ P_2 \circ U \circ \Gamma \circ P_1)(g))(x) \end{aligned}$$

$$P_1g = (p(g), p(g)),$$

$$\Gamma(f, g) = (g, T(g)),$$

$$U(f, g) = (f, g),$$

$$P_2(f, g) = (f, p(g)),$$

$$M(f, g) = (c_1f, c_2g),$$

$$A(f, g) = f + g,$$

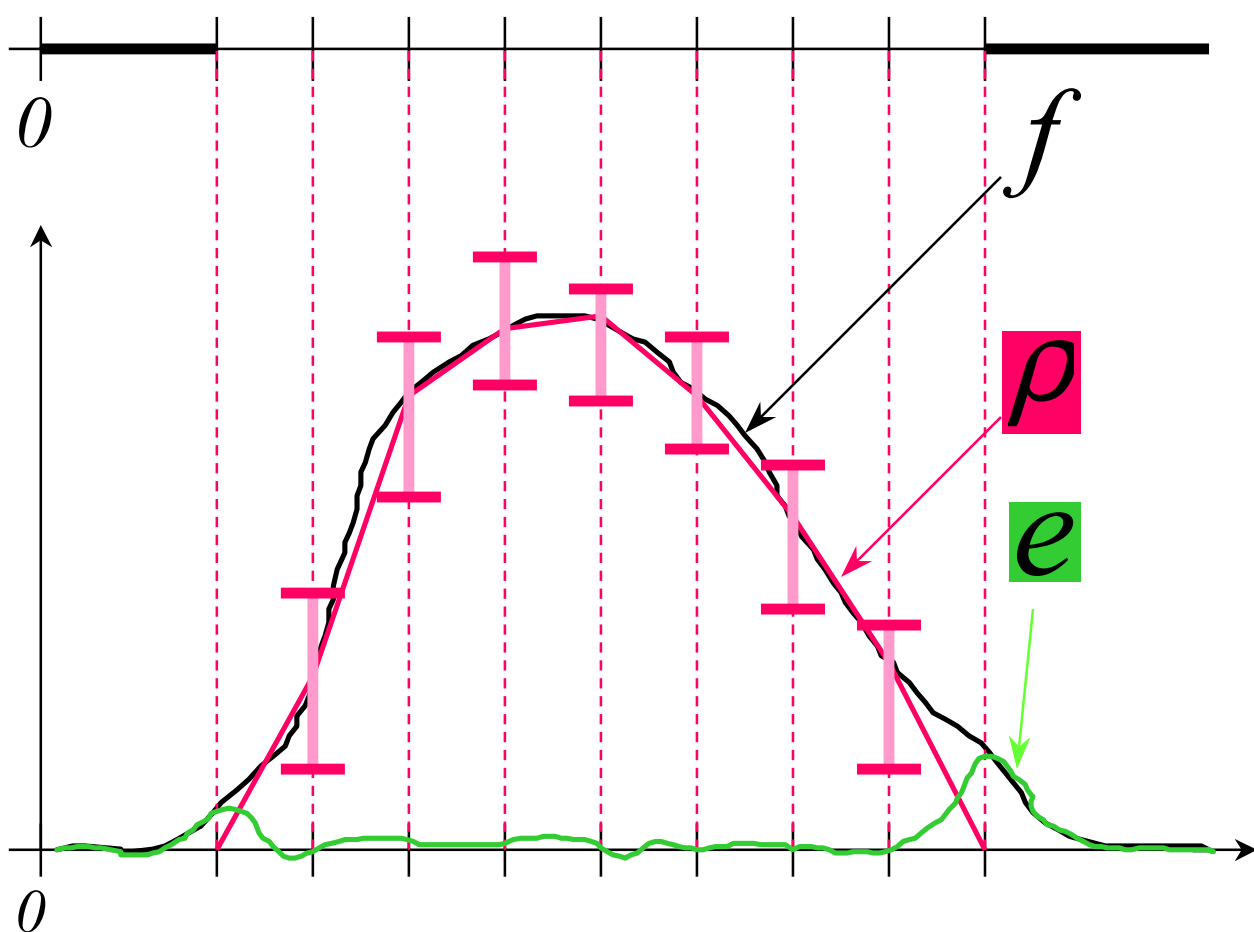
$$S(f)(x) = \mu f(\mu x),$$

$$B_{\alpha,\beta} \rightarrow B_{4\alpha,\beta} : p(g)(x) = g^{*2}(x)$$

Standard Sets (Choice)

$$B = L_1(\mathbb{R}^+, w(x) dx)$$

$$w(x) = \exp(1/x + x)$$



$$f = \rho + e, \quad \rho \in \dots,$$

$$\|e\| \leq \varepsilon$$

Bound on Convolution

$$p(f)(x) = (f * f)(x)$$

$$f = \rho + e$$

$$\begin{aligned} f * f &= \rho * \rho + 2 * \rho * e + e * e \\ &= \tilde{\rho} + \tilde{e} \end{aligned}$$

Lemma: $\rho * \rho$ an explicit cubic spline.

$$\|\tilde{e}\| \leq \dots$$

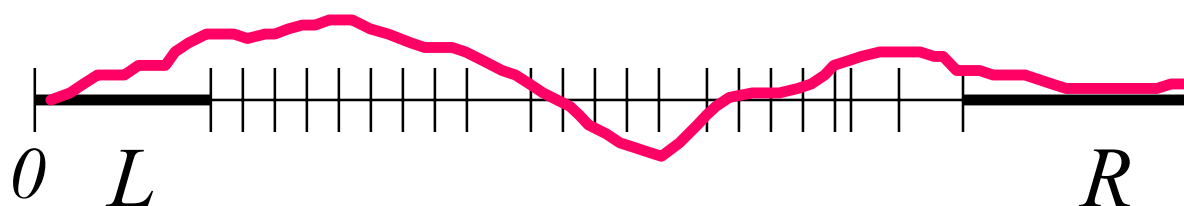


this is the only work
one does!

Bound on Derivative

$$(Dp(f)h)(x) = 2(f * h)(x)$$

$$f = \rho + e, \quad \|e\| \leq \beta$$



$$h = h_L + h_R + h_{\perp} + \sum_{i=0}^N h_i \chi_i$$

Lemma:

$$\|\rho * h_{\perp}\| < \delta^2 \cdot \|\rho''\| \cdot \|h_{\perp}\|$$

Choice of Programming Language

130

```
326:C..calculate sTGJDZF and sTDZFOGAMMA
327:   sT3=sDIFF(sINV(sZ),sQUOT(sPOWER(sRHOP,2),sR))
328:   sT3=sINV(sPOWER(sSQRT(sT3),7))
329:   sT5=sQUOT(sPROD(sT3,sT4)
330:   *   ,sPROD(sPOWER(sETA,4),sPOWER(sDIFF(sONE,sT1),3)))
331:   sT7=sQUOT(sQUOT(sICNST(7),sTWO),sPOWER(sSQRT(sT6),9))
332:   sT8=sQUOT(sPROD(sSUM(sTWO,sT1),sT7)
333:   *   ,sPROD(sPOWER(sETA,6),sPOWER(sDIFF(sONE,sT1),5)))
334:   sFAC=sPROD(sQUOT(sICNST(5),sTWO),sPROD(s3HALF,sPROD(sHALF
335:   *   ,sSQRT(sPI))))
336:   sT9=sPROD(sPROD(sPROD(sTWO,sFAC),sCTT),sSUM(sT5,sT8))
337:   sTGJDZF=sUPPER(sPROD(sPROD(sPOWER(sRHOP,3),sT9),sGGAMMA))
338:   sTDZFOGAMMA=sPROD(sTWO,sPROD(sPROD(sCTT,sFAC)
339:   *   ,sPROD(sQUOT(sPOWER(sRHOP,4),sPOWER(sETA,3)),sT3)))
340:C..calculate coefficients sJ1, sJ2 and sJ3
341:   sT1=sQUOT(sGGAMMA,sPOWER(sRHOP,3))
342:   sT1=DCMPLX(-rMAXABS(sT1),rMAXABS(sT1))
343:   sT2=sQUOT(sT1,sPOWER(sRHOGAMMA,2))
344:   sJ1H=sPROD(sFP00,sPROD(sFOUR,sT1))
345:   sJ1 =sSUM(sJ01,sJ1H)
346:   sJ2H=sPROD(sFP01,sPROD(sICNST(5),sT1))
347:   sJ2 =sSUM(sJ02,sJ2H)
348:   sJ3H=sSUM(sPROD(sFP00,sSUM(sPROD(sT1,sQUOT(sICNST(12)
349:   *   ,sKAPPA)),sPROD(sT2,sICNST(6))))),sPROD(sFP01
350:   *   ,sPROD(sICNST(6),sT1)))
351:   sJ3 =sSUM(sJ03,sJ3H)
352:C..calculate bound for general
353:   sT1=sSUM(sSUM(sSUM(sGDZJ,sPROD(sSUM(sPROD(sABS(sFP00),sGAMMA)
354:   *   ,sSUM(sPROD(sABS(sFP01),s2GAMMA),sPROD(sABS(sFP02)
355:   *   ,sPROD(sGAMMA,s2GAMMA))))),sGDZGAMMA))
356:   *   ,sSUM(sPROD(sTDZFOGAMMA,sGDZGAMMA)
357:   *   ,sPROD(sSUM(sABS(sFP00),sSUM(sPROD(sABS(sFP01),sSUM(sGAMMA
358:   *   ,sGAMMA)),sPROD(sABS(sFP02),sSUM(sPROD(sTHREE,s2GAMMA)
359:   *   ,sSUM(sPROD(sTHREE,sPROD(sGAMMA,sGHGAMMA))
360:   *   ,sPOWER(sGHGAMMA,2)))))),sPROD(sDZGAMMA,sGGAMMA)))
361:   *   ,sPROD(sTGJDZF,sDZGAMMA))
362:   sT2=sSUM(sSUM(sPROD(sJ1H,sPOWER(sRHOP,3))
363:   *   ,sPROD(sJ2H,sPOWER(sRHOP,4))),sPROD(sJ3H
364:   *   ,sPOWER(sRHOP,5)))
365:   sGDZJTOT=sCONST(rMAXABS(sSUM(sT1,sT2)))
366:C..calculate bound for higher order
367:   sT1=sSUM(sSUM(sHDZJ,sSUM(sPROD(sDZFOGAMMA,sHDZGAMMA)
368:   *   ,sPROD(sSUM(sABS(sFP00),sSUM(sPROD(sABS(sFP01)
369:   *   ,sSUM(sGAMMA,sGAMMA)),sPROD(sABS(sFP02),sSUM(sPROD(sTHREE
370:   *   ,s2GAMMA),sSUM(sPROD(sTHREE,sPROD(sGAMMA,sGHGAMMA))
371:   *   ,sPOWER(sGHGAMMA,2)))))),sPROD(sDZGAMMA,sHGAMMA)))
372:   *   ,sPROD(sQUOT(sCT,sC),sPROD(sHJDZF,sDZGAMMA))
373:   sHDZJTOT=sCONST(rMAXABS(sT1))
374:C..invert Ecalle's equation: lower order coefficients
375:   sDZFD01=sQUOT(sJ1,sPROD(sFOUR,sGA3))
376:   sDZFD02=sQUOT(sJ2,sPROD(sICNST(5),sGA3))
377:   sDZFD03=sQUOT(sDIFF(sJ3,sPROD(sSUM(sPROD(s3HALF
378:   *   ,sQUOT(sGA5,sGA3)),sPROD(sQUOT(sTHREE,sFOUR),sGA3))
379:   *   ,sJ1)),sPROD(sICNST(6),sGA3))
380:C..add additional contribution to general
381:   CALL FsDILAT(vGAMMA,sRH01,vT1)
382:   CALL FsMULT(vT1,sRHOGAMMA,vT1)
383:   CALL FDZsDILAT(vGAMMA,sRH01,vT2)
384:   CALL FMULT(vT1,vT2,vT2)
385:   CALL FEQUAL(vT2,vT3)
```

Prolog

(PROgramming in LOGic)

even if r_1 and r_2 are in S , as we shall now assume. However, as far as bounds are concerned, it is sufficient to find an interval $i(X1, Y1)$ that contains $r_1 \# r_2$, given an interval $i(X, Y)$ that contains the computed value for $r_1 \# r_2$. The following predicate `enlarges/2` serves this purpose:

```
i(X1, Y1) enlarges i(X, Y) :-  
  X1 is_a_safe_lower_bound_on X,  
  Y1 is_a_safe_upper_bound_on Y,  
  !.
```

To be more precise, if $X \leq Y$ are given representable numbers (not necessarily in the safe range), then $i(X1, Y1)$ enlarges $i(X, Y)$ is satisfied if and only if the indicated safe lower bound $X1$ and safe upper bound $Y1$ can be found, in which case $i(X1, Y1)$ is a standard set with the above-mentioned property, as explained in §3. A typical application of `enlarges/2` is given in the next clause.

Consider now the sum function from $\mathbb{R} \times \mathbb{R}$ to \mathbb{R} . The corresponding set map assigns to a pair (I_2, I_3) in $\mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R})$ the set

$$(5.3) \quad I_2 + I_3 = \{r \in \mathbb{R} \mid r = x + y \text{ for some } x \in I_2, y \in I_3\}$$

in $\mathcal{P}(\mathbb{R})$. The following clause defines a bound on this map:

```
i(X1, Y1) contains i(X2, Y2) + i(X3, Y3) :-  
  X is X2 + X3,  
  Y is Y2 + Y3,  
  i(X1, Y1) enlarges i(X, Y),  
  !.
```

As indicated in §4, the domain of this bound is the set of all pairs (I_2, I_3) in $\text{std}(\mathbb{R}) \times \text{std}(\mathbb{R})$ for which I_1 contains $I_2 + I_3$ is true. In this case, the domain is determined by the predicate `enlarges/2`, which is used in order to compensate for possible rounding errors introduced by `is/2` and to ensure that the returned result I_1 is again a standard set.

The same description applies to our bound on the product of (sets of) real numbers if $+$ is replaced by $*$. The bound itself is defined as follows:

```
i(X1, Y1) contains i(X2, Y2) * i(X3, Y3) :-  
  A is X2 * X3,  
  B is X2 * Y3,  
  C is Y2 * X3,  
  D is Y2 * Y3,  
  sort_numbers([A, B, C, D], [X, _, _, Y]),  
  i(X1, Y1) enlarges i(X, Y),  
  !.
```

Here, we have used that the product of two standard sets I_2 and I_3 is an interval and that the

Summary

- Bounds are set-maps
- Bounds are defined on “standard sets”
- Can compute with bounds
- Bounds can be composed to form new bounds
- Can use chain rule to compute derivative of complicated maps
- Need only worry about simple maps
- Let computer do the rest
(Prolog)
- SIAM Review, 38, 4, 1996