Lean 2021

Complex Analysis
through a hierarchy of real-analysis structures

Marie Kerjean
CNRS, LIPN, Université Sorbonne Paris Nord.

Work in collaboration with
Reynald Affeldt, Cyril Cohen, Assia Mahboubi,
Damien Rouhling, Kazuhiko Sakaguchi, Pierre-Yves Strub

What this talk is about

"Holomorphy is differentiability on $\mathbb{R}$ with some condition on the partial devatives."

The Cauchy-Riemann Equations

11.1 The Operators ∂ and $\bar\partial$ Suppose f is a complex function defined in a plane open set Ω. Regard f as a transformation which maps Ω into R^2, and assume that f has a differential at some point $z_0 \in \Omega$, in the sense of Definition 8.22. For simplicity, suppose $z_0 = f(z_0) = 0$. Our differentiability assumption is then equivalent to the existence of two complex numbers α and β (the partial derivatives of f with respect to x and y at $z_0 = 0$) such that

(1) $$f(z) = \alpha x + \beta y + \eta(z)z \qquad (z = x + iy),$$

where $\eta(z) \to 0$ as $z \to 0$.

Since $2x = z + \bar z$ and $2iy = z - \bar z$, (1) can be rewritten in the form

(2) $$f(z) = \frac{\alpha - i\beta}{2} z + \frac{\alpha + i\beta}{2} \bar z + \eta(z)z.$$

This suggests the introduction of the differential operators

(3) $$\partial = \frac{1}{2}\left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right), \qquad \bar\partial = \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right).$$

Now (2) becomes

(4) $$\frac{f(z)}{z} = (\partial f)(0) + (\bar\partial f)(0) \cdot \frac{\bar z}{z} + \eta(z) \qquad (z \neq 0).$$

For real z, $\bar z / z = 1$; for pure imaginary z, $\bar z / z = -1$. Hence $f(z)/z$ has a limit at 0 if and only if $(\bar\partial f)(0) = 0$, and we obtain the following characterization of holomorphic functions:

11.2 Theorem *Suppose f is a complex function in Ω which has a differential at every point of Ω. Then $f \in H(\Omega)$ if and only if the Cauchy-Riemann equation*

(1) $$(\bar\partial f)(z) = 0$$

222

Do you remember 2019 ... ?

- I was a post-doc with Assia Mahboubi, who's all about computer algebra.
- I needed to define holomorphy: "Easy !".
  - The real-closed library of mathcomp provides us with $\mathbb{R}[i]$.
  - MathComp Analysis libraries provides us with differentiabilty
  - `Definition holomorphic (f : R[i] -> R[i]) := differentiable f.`
- But just to be sure ... Let's prove Cauchy-Riemann.

Spoiler : this was 16 month ago. I never started to look at computer algebra's proofs.

Mathematical-Components

A constructive library in **Coq** providing us with a rich theory of algebra and data structures:

- Algebraic structures : modules, fields, fields extensions, modules, ...

  ```
  h^-1 *: ( f (c + h) - f(c))
  ```

- Numerical structures : Rings and Fields with an order and a norm.

  ```
  `|h^-1 *: (f (c + h) - f(c))| < eps
  ```

- Real-Closed structures : real closures, algebraic closures, quantifier elimination.

  ```
  `|h^-1 *: ( f (c + h * i) - f(c)| < eps
  ```

MathComp Analysis is about adding topologies to these :

```
fun h => `|h^-1 *: ( f (c + h * i) - f(c)| @ 0 --> 'D_i f c
```

Mathematical-Components- Analysis

Why ?

- Because it's fun.
- Application in verification/robotics.

It reinterprets and extends the **Coquelicot** project. [Boldo and al, 2015]

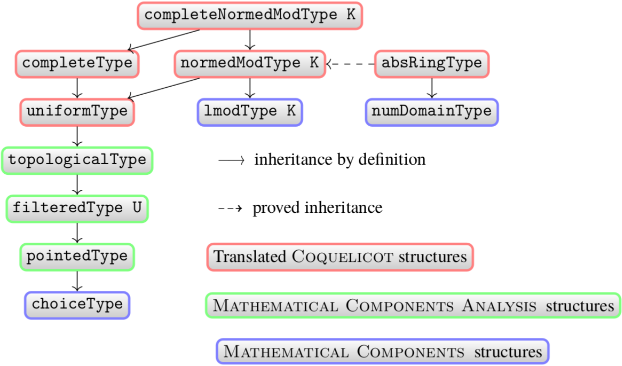


Figure: MATHEMATICAL COMPONENTS ANALYSIS hierarchy

[Cohen 2018]

Cauchy Riemann Equations

A function $f : \mathbb{C} \to \mathbb{C}$ is derivable at a point c if and only if it is differentiable along real vectors, and if $D_i f c = i \times D_1 f c$.

Differentiability in MathComp analysis in defined on functions defined between normed vector spaces.

- $\mathbb{C}$ as a field, endowed with the usual norm, is a vector space on itself.

```
Definition holomorphic (f : R[i] -> R[i]) := differentiable f.
```

- $\mathbb{C}$ a $\mathbb{R}$-module : a new normed vector space structure is defined on an alias `Rcomplex`.

```
Definition Rdifferentiable (f : R[i] -> R[i]) :=
differentiable (f : Rcomplex -> Rcomplex).
```

Algebraic proofs

Let us just take a small glance at the proofs at stake:

- From holomorphic to differentiability : a matter of proving that a function $\mathbb{C}$-linear is $\mathbb{R}$-linear.
- From holomorphic to Cauchy-Riemann equations: a matter of differentiating along real or imaginary axis and comparing.
- From Cauchy-Riemann and differentiability : a matter of proving $\mathbb{C}$-linearity from $\mathbb{R}$-linearity.

If the proofs must be algebraic, all the topological arguments must be hidden in the definition.

Topological definitions

- One handles two vector space, $\mathbb{C}$ and `Rcomplex`, each endowed with a norm.

```
‘|-| : Rcomplex -> R
‘|-| : R[i] -> R[i]
```

Allows for $v/`|-|$ *and a factorization of lemmas.*

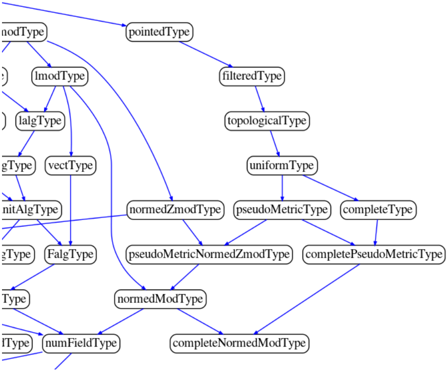
- Considering topologies directly induced by these norms leads to goals as:

```
nbhs ( 0 : C ) = nbhs (0 : Rcomplex)
```

and proofs that $|x| < e : \mathbb{C} \leftrightarrow |x| < \mathtt{Re}(e) : \mathbb{R}$.

- However, packed structures and forgetful inference allow for a topology *not induced but compatible* by the norm.

A packed hierarchy of topological structures



[Diagram by Kazuhiko Sakaguchi]

`topology.v` documentation:
UniformMixin _ _ _ _ nbhse ==
builds the mixin for a uniform space from the properties of entourages and the compatibility between entourages and nbhs .

Low-level topological proofs

How to avoid cutting epsilons in half :

```
pose t := tp%:num .
 exists (2*t^-1). split=> //.
 move=> x; case: x =P 0.
 - by move=> ->; rewrite f0 normr0 normr0 //= mul0r.
 - move/eqP=> xneq0 Fx.
 pose a : V := (`|x|^-1 * t/2 ) *: x.
 have Btp : ball 0 t a.
  apply : ball_sym ; rewrite -ball_normE /ball_ subr0.
  rewrite normmZ mulrC normrM.
  rewrite !gtr0_norm //= ; last by rewrite pmulr_lgt0 // invr_gt0 normr_gt0.
  rewrite mulrC -mulrA -mulrA ltr_pdivr_mull; last by rewrite normr_gt0.
  rewrite mulrC -mulrA gtr_pmull.
  rewrite invf_lt1 //=.
  by rewrite pmulr_lgt0 // !normr_gt0.
```

High-level topological proofs

[Affeldt, Cohen, Rouhling, Formalization Techniques for Asymptotic Reasoning in Classical Analysis]

- Topologies which are definitionaly equal.

```
Goal : continuous (df : Rc -> Rc) <-> continuous (df : C -> C)
by [].
```

- `near` tactics.

```
Lemma add_continuous : continuous (fun z : V * V => z.1 + z.2).
Proof.
move=> [/=x y]; apply/cvg_distP=> _/posnumP[e].
rewrite !near_simpl /=; near=> a b => /=; rewrite opprD addrACA.
by rewrite normm_lt_split //; [near: a|near: b]; apply: cvg_dist.
Grab Existential Variables. all: end_near. Qed.
```

- Landau notations.

```
Goal : f%:Rfun \o +%R^~ c = cst (f c) + df +o_ (0 : Rc) id <->
        f \o +%R^~ c = cst (f c) + df +o_ (0 : C) id
by rewrite littleoCo.
```

des hauts et des bas ...

From holomorphy to Rdifferentiability

```
Definition Rdifferentiable (f : C -> C) (c : C) := (differentiable f%:Rfun c%:Rc).

Lemma holo_differentiable (f : C -> C) (c : C) :
  holomorphic f c -> Rdifferentiable f c.
Proof.
move=> /holomorphicP /derivable1_diffP /diff_locallyP => -[cdiff holo].
have lDf : linear ('d f c : Rc -> Rc) by move=> t u v; rewrite !scalecr linearP.
pose df : Rc -> Rc := Linear lDf.
have cdf : continuous df by [].
have eqdf : f%:Rfun \o +%R^~ c = cst (f c) + df +o_ (0 : Rc) id
  by rewrite holo littleoCo.
by apply/diff_locallyP; rewrite (diff_unique cdf eqdf).
Qed.
```

From holomorphy to Cauchy-Riemann equations

- Restricting to the real or complex line is done by composing with the embedding of $\mathbb{R}$ in $\mathbb{C}$.

```
Definition realC : R^o -> C^o := (fun r => r%:C).

Lemma Rdiff1 (f : C^o -> C^o) c :
         lim ( (fun h : C^o => h^-1 *: ((f (c + h) - f c) ) )
                @ (realC @ (nbhs' 0)))
        = 'D_1 f%:Rfun c%:Rc :> C.

Lemma Rdiffi (f : C^o -> C^o) c:
        lim ( (fun h : C^o => h^-1 *: ((f (c + h * 'i) - f c)))
                @ (realC @ (nbhs' 0 )))
        = 'D_i f%:Rfun c%:Rc :> C.
```

```
Lemma holo_CauchyRiemann (f : C -> C) c:
holomorphic f c -> CauchyRiemannEq f c.
Proof.
move=> /cvg_ex; rewrite //= /CauchyRiemannEq -Rdiff1 -Rdiffi.
set quotC : C -> C := fun h : R[i] => h^-1 *: (f (c + h) - f c).
set quotR : C-> C: = fun h => h^-1 *: (f (c + h * 'i) - f c) .
move => [l holo].
have -> : quotR @ (realC @ nbhs' 0) = (quotR \o realC) @ nbhs' 0 by [].
have realC'0: realC @ nbhs' 0 --> nbhs' 0.
 by apply: within_continuous_withinNx; first by apply: continuous_realC.
have HR0:(quotC \o (realC) @ nbhs' 0) --> l.
 by apply: cvg_comp; last by exact: holo.
have lem : quotC \o *%R^~ 'i%R @ (realC @ (nbhs' (0 : R^o))) --> l.
  apply: cvg_comp; last by exact: holo
  (*...*)
have HRcomp: cvg (quotC \o *%R^~ 'i%R @ (realC @ (nbhs' (0 : R^o)))) .
  by apply/cvg_ex; simpl; exists l.
have ->: lim (quotR @ (realC @ (nbhs' (0 : R^o))))
  = 'i *: lim (quotC \o ( fun h => h *'i) @ (realC @ (nbhs' (0 : R^o)))).
  (*...*)
rewrite scalecM.
suff: lim (quotC @ (realC @ (nbhs' (0 : R^o))))
     = lim (quotC \o *%R^~ 'i%R @ (realC @ (nbhs' (0 : R^o)))) by move => -> .
suff -> : lim (quotC @ (realC @ (nbhs' (0 : R^o)))) = l.
  by apply/eqP; rewrite eq_sym; apply/eqP; apply: (cvg_map_lim _ lem).
by apply: (@cvg_map_lim [topologicalType of C^o]).
Qed.
```

From Rdifferentiability to holomorphy

```
Lemma Diff_CR_holo (f : C -> C) (c: C):
   (Rdifferentiable f c) /\ (CauchyRiemannEq f c)
  -> (holomorphic f c).

Proof.
move => [] /= /[dup] H /diff_locallyP => [[derc /eqaddoP diff]] CR.
apply/holomorphicP/derivable1_diffP/diff_locallyP.
pose Df := (fun h : C => h *: ('D_1 f%:Rfun c : C)).
have lDf : linear Df by move => t u v /=; rewrite /Df scalerDl scalerA scalecM.
pose df := Linear lDf.
have cdf : continuous df by apply: scalel_continuous.
have lem : Df = 'd (f%:Rfun) (c : Rc).
  apply: funext => z; rewrite /Df.
  set x := Re z; set y := Im z.
  have zeq : z = x *: 1 %:Rc + y *: 'i%:Rc.
  by rewrite [LHS]complexE /= realC_alg scalecr scalecM [in RHS]mulrC.
  rewrite [X in 'd _ _ X]zeq addrC linearP linearZ /= -!deriveE //.
  rewrite -CR (scalecAl y) -scalecM !scalecr /=.
  rewrite -(scalerDl (('D_1 f%:Rfun (c : Rc)) : C) _ (real_complex R x)).
  by rewrite addrC -!scalecr -realC_alg -zeq.
have holo: f \o shift c = cst (f c) + df +o_ ( 0 : C) id.
  apply/eqaddoP => eps /[dup] /gt0_realC epsr /realC_gt0 eps0; near=> x.
  by rewrite epsr realCM lecR /= lem; near:x; apply: diff.
have -> : 'd  f c = df:> ( _ -> _) by apply: diff_unique.
by split.
Grab Existential Variables. by end_near. Qed.
```

Conclusion

Issues:

- `nbhs' (0 : R^o)`

  In MathComp, any field is not yet a vector space on itself. Instead, one considers the regular algebra $\mathbb{C}^o$.
  This is enforced in a PR on MathComp Analysis and will be merged soon.
- MathComp Analysis comes with powerful tools for manipulating landau notations and neighborhoods : typing those is still a struggle sometimes.
- Slowdown introduces by Rcomplex : filteredTypes.

Soon in MathComp Analysis:

- Measure Theory, Lebesgue Integrals, ...

Maybe one day in MathComp Analysis:

- Distribution Theory, Topological vector spaces.

Thank you for your attention !