SIMONS FOUNDATION

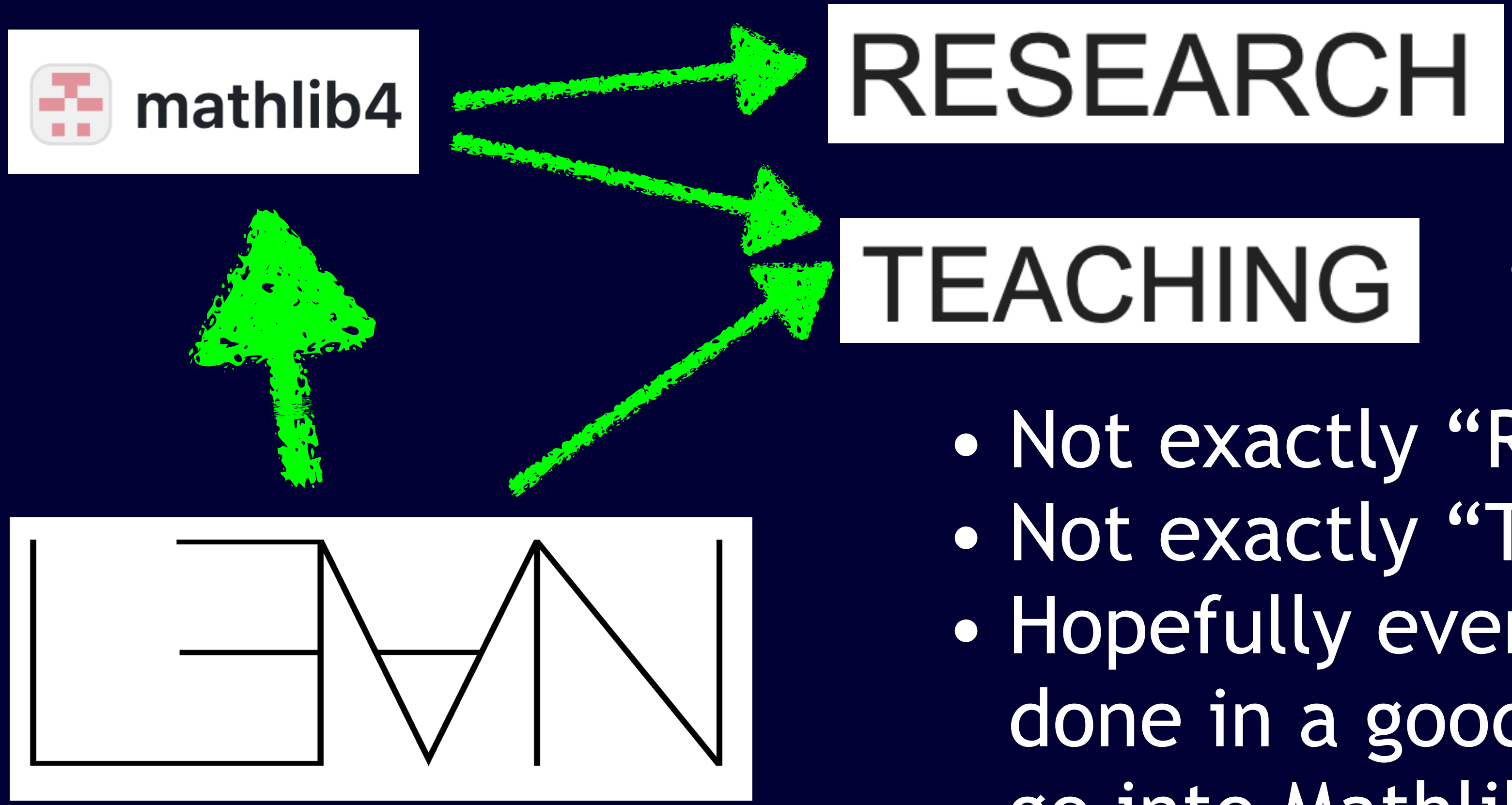
Workshop on LEAN

Alex Kontorovich

RUTGERS UNIVERSITY / IAS

SIMONS FOUNDATION

Workshop on LEAN



PNT+

- Started 1/31/24
- PNT proved 4/8/24

- Not exactly "Research"
- Not exactly "Teaching"
- Hopefully eventually will be done in a good enough way to go into Mathlib; not currently

PNT

- PNT+ co-organized with Terry Tao
- Goal: Fermat will need Chebotarev Density Theorem. Special case of that is Dirichlet's theorem (primes in progressions). Didn't even have Prime Number Theorem in Mathlib. So let's get to work!

- Note: PNT has been formalized before, many times in fact.
- 2005: Avigad et al in Isabelle (Erdos-Selberg method);
- 2009: Harrison in HOL-light (Newman's proof);
- 2016: Carniero in Metamath (Erdos-Selberg);
- 2018: Eberl-Paulson in Isabelle (Newman)
- We will want to do in it a way that extends to much more general settings.

- Organizational infrastructure: Github + Blueprint + Zulip

PNT

- Organizational infrastructure: Github + Blueprint + Zulip

- Original project comprised three attacks:
  - (Weak) using "Fourier" methods, Wiener-Ikehara Tauberian theorem. Work of Michael Stoll already reduced PNT to this. (Done!)
  - (Medium) developing Mellin transform API (David Loeffler), pulling infinite vertical contours past poles, picking up residues. And
  - (Strong) Getting a "classical" error savings of $\exp(c(\log x)^{1/2})$ using Hadamard factorization (or local versions)

- Meanwhile, Shuhao Song+Bowen Yao formalized (Strong) in Isabelle!
  - Posted mid-March '24 (?); paper says formalization took one month
  - Built on top of much bigger Complex Analysis library…

PNT

- These were all a great excuse to get more analysis into Mathlib
- We didn't have Fourier inversion (now we do, Sebastian Gouzel)
- We didn't have that Fourier transform of Schwartz function is Schwartz, now we do (Gouzel + K-Loeffler-Macbeth + Beffara)

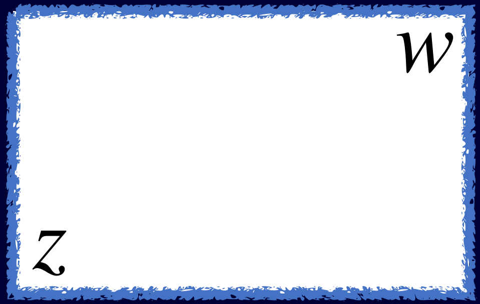
- We were also missing one of the least developed late undergrad / early grad areas of Mathlib (needed for lots of analytic number theory), namely: Complex Analysis

- Big Idea: Can do all the Complex Analysis we need just using Rectangles!

PNT

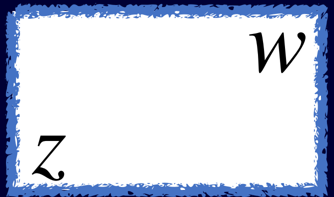
- Big Idea: Can do all the Complex Analysis we need just using Rectangles!
  - We have Green's Theorem in Mathlib (Yury Kudryashov), so the integral of a holomorphic function over a rectangle is zero.

```
/-%%
\begin{definition}\label{Rectangle}\lean{Rectangle}\leanok
A Rectangle has corners $z$ and $w \in \C$.
\end{definition}
%%-/
/-- A `Rectangle` has corners `z` and `w`. -/
def Rectangle (z w : ℂ) : Set ℂ := [[z.re, w.re]] ×ℂ [[z.im, w.im]]
```




```
noncomputable def RectangleIntegral
    {E : Type u_1} [NormedAddCommGroup E] [NormedSpace ℂ E] (f : ℂ → E)
    (z w : ℂ) :
```

PNT



- We have Green's Theorem in Mathlib (Yury Kudryashov), so the integral of a holomorphic function over a rectangle is zero.

```
noncomputable def RectangleIntegral
      {E : Type u_1} [NormedAddCommGroup E] [NormedSpace ℂ E] (f : ℂ → E)
      (z w : ℂ) :
```

```
RectangleIntegral f z w = HIntegral f z.re w.re z.im -
HIntegral f z.re w.re w.im + VIntegral f w.re z.im w.im -
VIntegral f z.re z.im w.im
```

```
HIntegral f x₁ x₂ y = ∫ (x : ℝ) in x₁..x₂, f (↑x + ↑y * Complex.I)
```

```
theorem HolomorphicOn.vanishesOnRectangle
      {E : Type u_1} [NormedAddCommGroup E] [NormedSpace ℂ E] {f : ℂ → E}
      {z w : ℂ} [CompleteSpace E] {U : Set ℂ} (f_holo : HolomorphicOn f U)
      (hU : z.Rectangle w ⊆ U) :
   RectangleIntegral f z w = 0
```

PNT

- We have Green's Theorem in Mathlib (Yury Kudryashov), so the integral of a holomorphic function over a rectangle is zero.

```
theorem HolomorphicOn.vanishesOnRectangle
    {E : Type u_1} [NormedAddCommGroup E] [NormedSpace ℂ E] {f : ℂ → E}
    {z w : ℂ} [CompleteSpace E] {U : Set ℂ} (f_holo : HolomorphicOn f U)
    (hU : z.Rectangle w ⊆ U) :
  RectangleIntegral f z w = 0
```

```
theorem ResidueTheoremOnRectangleWithSimplePole
    {f g : ℂ → ℂ} {z w p A : ℂ} (zRe_le_wRe : z.re ≤ w.re)
    (zIm_le_wIm : z.im ≤ w.im) (pInRectInterior : z.Rectangle w ∈ nhds p)
    (gHolo : HolomorphicOn g (z.Rectangle w))
    (principalPart :
     Set.EqOn (f - fun (s : ℂ) => A / (s - p)) g (z.Rectangle w \ {p}))
    :
  RectangleIntegral' f z w = A
```

PNT


```
theorem ResidueTheoremOnRectangleWithSimplePole
        {f g : ℂ → ℂ} {z w p A : ℂ} (zRe_le_wRe : z.re ≤ w.re)
        (zIm_le_wIm : z.im ≤ w.im) (pInRectInterior : z.Rectangle w ∈ nhds p)
        (gHolo : HolomorphicOn g (z.Rectangle w))
        (principalPart :
         Set.EqOn (f - fun (s : ℂ) => A / (s - p)) g (z.Rectangle w \ {p}))
        :
    RectangleIntegral' f z w = A
```

- How to find such a *g*? Riemann removable singularity theorem!

```
theorem existsDifferentiableOn_of_bddAbove                                  source
        {E : Type u_1} [NormedAddCommGroup E] [NormedSpace ℂ E] {f : ℂ → E}
        [CompleteSpace E] {s : Set ℂ} {c : ℂ} (hc : s ∈ nhds c)
        (hd : HolomorphicOn f (s \ {c})) (hb : BddAbove (norm ∘ f '' (s \ {c}))) :
    ∃ (g : ℂ → E), HolomorphicOn g s ∧ Set.EqOn f g (s \ {c})
```

- Let's do some Number Theory!

PNT

- Let's do some Number Theory!

"von Mangoldt" function:

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

```
#print vonMangoldt
```

```
def ArithmeticFunction.vonMangoldt :
ArithmeticFunction ℝ :=
{ toFun := fun n ↦ if IsPrimePow n then
Real.log ↑n.minFac else 0, map_zero' :=
vonMangoldt._proof_2 }
```

```
local notation "Λ" => vonMangoldt
```

PNT

"von Mangoldt" function: $\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$

```
#print vonMangoldt
```

```
def ArithmeticFunction.vonMangoldt :
ArithmeticFunction ℝ :=
{ toFun := fun n ↦ if IsPrimePow n then
Real.log ↑n.minFac else 0, map_zero' :=
vonMangoldt._proof_2 }
```

(Michael Stoll)

```
def ArithmeticFunction
      (R : Type u_1) [Zero R] :
    Type u_1
```

source

An arithmetic function is a function from `ℕ` that maps 0 to 0. In the literature, they are often instead defined as functions from `ℕ+`. Multiplication on `ArithmeticFunctions` is by Dirichlet convolution.

▼ Equations

- `ArithmeticFunction R = ZeroHom ℕ R`

PNT

"von Mangoldt" function: $$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

Chebyshev function: $$\psi(x) := \sum_{n \le x} \Lambda(n)$$

```
noncomputable def ChebyshevPsi (x : ℝ) : ℝ :=
  (Finset.range ⌊x + 1⌋₊).sum Λ
```

```
local notation "ψ" => ChebyshevPsi
```

PNT

"von Mangoldt" function: $\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$

Chebyshev function: $\psi(x) := \sum_{n \leq x} \Lambda(n)$

PNT (Medium Version):

$$|\psi(x) - x| \ll x \exp(-c(\log x)^{1/10}), \; x \to \infty$$

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

PNT

$$\psi(x) := \sum_{n \leq x} \Lambda(n)$$

PNT (Medium Version):

$$|\psi(x) - x| \ll x \exp(-c(\log x)^{1/10}),\ x \to \infty$$

```
theorem MediumPNT : ∃ c > 0,    declaration uses 'sorry'
    (ψ - id) =O[atTop]
      fun (x : ℝ) ↦ x * Real.exp (-c * (Real.log x) ^ ((1 : ℝ) / 10)) := by
```

```
f =O[l] g = ∃ (c : ℝ), Asymptotics.IsBigOWith c l f g
```

```
Asymptotics.IsBigOWith c l f g = ∀ᶠ (x : α) in l, ‖f x‖ ≤ c * ‖g x‖
```

PNT

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$|\psi(x) - x| \ll x \exp(-c(\log x)^{1/10}),\ x \to \infty$$

Zeta function:

$$\zeta(s) = \prod_p \left(1 - \frac{1}{p^s}\right)^{-1},\ \Re(s) > 1$$

```
/-- The Euler product for the Riemann ζ function, valid for `s.re > 1`.
This version is stated in terms of `tprod`. -/
theorem riemannZeta_eulerProduct_tprod (hs : 1 < s.re) :
    ∏' p : Primes, (1 - (p : ℂ) ^ (-s))⁻¹ = riemannZeta s :=
```

PNT

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$|\psi(x) - x| \ll x \exp(-c(\log x)^{1/10}),\ x \to \infty$$

Zeta function:

$$\zeta(s) = \prod_p \left(1 - \frac{1}{p^s}\right)^{-1},\ \Re(s) > 1$$

Log-deriv:

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

```
theorem ArithmeticFunction.LSeries_vonMangoldt_eq_deriv_riemannZeta_div
      {s : ℂ} (hs : 1 < s.re) :
    LSeries (fun (n : ℕ) => ↑(vonMangoldt n)) s = -deriv riemannZeta s /
        riemannZeta s
```

```
LSeries f s = ∑' (n : ℕ), LSeries.term f s n
```

```
LSeries.term f s n = if n = 0 then 0 else f n / ↑n ^ s
```

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

PNT

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

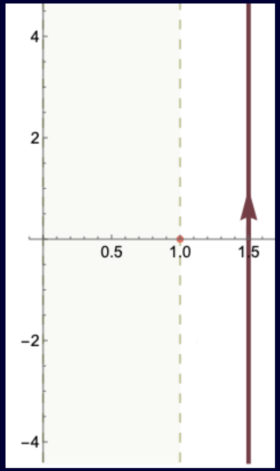
$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

Idea: “Perron formula”

For $\sigma > 0$,

$$\frac{1}{2\pi i} \int_{(\sigma)} \frac{y^{-s}}{s} ds = \mathbf{1}(y) := \begin{cases} 1 & \text{if } 0 < y < 1, \\ 0 & \text{else.} \end{cases}$$

Here $\int_{(\sigma)} = \int_{\sigma - i\infty}^{\sigma + i\infty} dt.$



Issues with convergence!… (Later)

PNT

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

Idea: "Perron formula" For $\sigma > 0$,

$$\frac{1}{2\pi i}\int_{(\sigma)} \frac{y^{-s}}{s} ds = \mathbf{1}(y) := \begin{cases} 1 & \text{if } 0 < y < 1, \\ 0 & \text{else.} \end{cases}$$

So:

$$\psi(x) := \sum_{n \le x} \Lambda(n) = \sum_n \Lambda(n)\, \mathbf{1}(n/x) = \sum_n \Lambda(n) \frac{1}{2\pi i}\int_{(\sigma)} \frac{(x/n)^s}{s} ds$$

$$= \frac{1}{2\pi i}\int_{(\sigma)} \left(\sum_n \Lambda(n) n^{-s} \right) \frac{x^s}{s} ds = \frac{1}{2\pi i}\int_{(\sigma)} \frac{\zeta'}{\zeta}(s)\, \frac{x^s}{s}\, ds$$

Need $\sigma > 1$

PNT

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$\psi(x) = \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \frac{x^s}{s} \, ds$$

Need $\sigma > 1$

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

None of this is OK, no absolute convergence. Nevermind!

Now, zeta has meromorphic continuation

$$\zeta(s) := \frac{\pi^{s/2}}{\Gamma(s/2)} \left[\int_1^\infty \left(2 \sum_{n=1}^\infty e^{-\pi n^2 u^2} \right) (u^s + u^{1-s}) \frac{du}{u} - \frac{1}{1-s} - \frac{1}{s} \right]$$

PNT

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$\psi(x) = \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \frac{x^s}{s} \, ds$$

Need $\sigma > 1$

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

None of this is OK, no absolute convergence. Nevermind!

Now, zeta has meromorphic continuation

$$\zeta(s) := \frac{\pi^{s/2}}{\Gamma(s/2)} \left[\int_1^\infty \left(2 \sum_{n=1}^\infty e^{-\pi n^2 u^2} \right) (u^s + u^{1-s}) \frac{du}{u} - \frac{1}{1-s} - \frac{1}{s} \right]$$

```
theorem differentiableAt_riemannZeta
    {s : ℂ} (hs' : s ≠ 1) :
    DifferentiableAt ℂ riemannZeta s
```

So we can “pull contours”

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

PNT

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

So we can "pull contours"

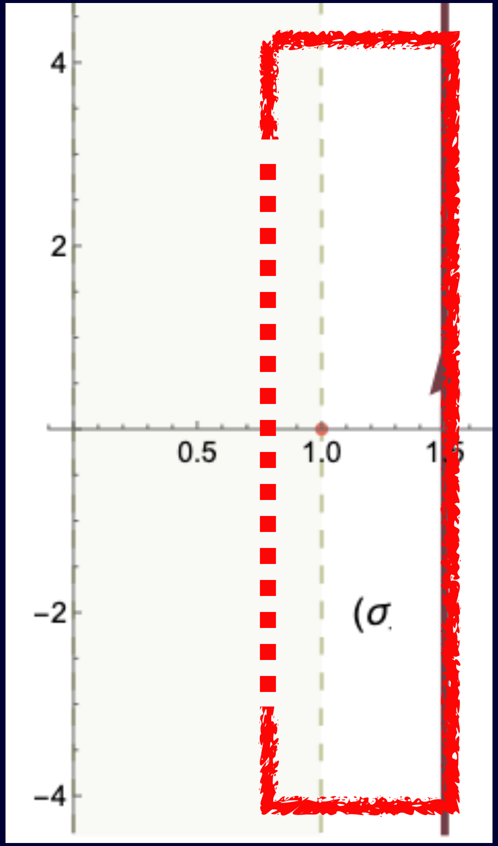
$$\psi(x) = \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \frac{x^s}{s} \, ds$$

Fact: ζ'/ζ is meromorphic on tall thin rectangle: $1 - \frac{A}{(\log T)^9} < \sigma$

```
theorem LogDerivZetaHolcLargeT :     declaration uses 'sorry'
    ∃ (A : ℝ) (_ : A ∈ Ioc 0 (1 / 2)), ∀ (T : ℝ) (_ : 3 < T),
    HolomorphicOn (fun (s : ℂ) ↦ deriv ζ s / (ζ s))
      (( [[ ((1 : ℝ) - A / Real.log T ^ 9), 2 ]] ×ℂ [[ -T, T ]]) \ {1}) := by
  sorry
```

And good bounds there (K-Sedlacek)

```
lemma LogDerivZetaBndUniform :     declaration uses 'sorry'
    ∃ (A : ℝ) (_ : A ∈ Ioc 0 (1 / 2)) (C : ℝ) (_ : 0 < C), ∀ (σ : ℝ) (T : ℝ) (t : ℝ) (_ : 3 < |t|)
    (_ : |t| ≤ T) (_ : σ ∈ Ico (1 - A / Real.log T ^ 9) 1),
    ‖deriv ζ (σ + t * I) / ζ (σ + t * I)‖ ≤ C * Real.log T ^ 9 := by
```



$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

PNT

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

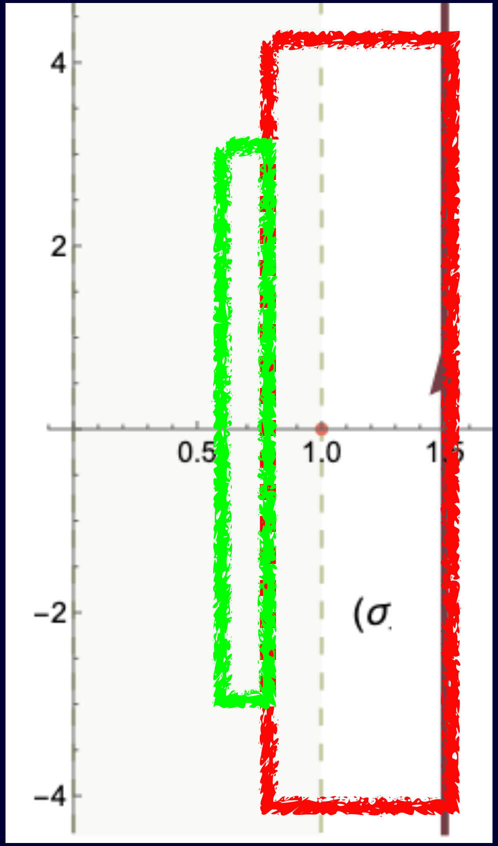
$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

```
lemma LogDerivZetaBndUniform :    declaration uses 'sorry'
    ∃ (A : ℝ) (_ : A ∈ Ioc 0 (1 / 2)) (C : ℝ) (_ : 0 < C), ∀ (σ : ℝ) (T : ℝ) (t : ℝ) (_ : 3 < |t|)
    (_ : |t| ≤ T) (_ : σ ∈ Ico (1 - A / Real.log T ^ 9) 1),
    ‖deriv ζ (σ + t * I) / ζ (σ + t * I)‖ ≤ C * Real.log T ^ 9 := by
```

$$\psi(x) = \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \frac{x^s}{s} \, ds$$

Near $s = 1$, ζ'/ζ blows up, so need to move away

```
theorem LogDerivZetaHolcSmallT :    declaration uses 'sorry'
    ∃ (σ₀ : ℝ) (_ : σ₀ < 1), HolomorphicOn (fun (s : ℂ) ↦ deriv ζ s / (ζ s))
      (( [[ σ₀, 2 ]] ×ℂ [[ -3, 3 ]]) \ {1}) := by
```



And then you need to estimate the resulting integrals.

How to make this all rigorous?

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

PNT

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

$$\psi(x) \not= \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \frac{x^s}{s}\, ds$$

How to make this all rigorous?

Let $\psi_\epsilon(x) := \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s)\, \widetilde{\nu}(\epsilon s)\, \frac{x^s}{s}\, ds = \sum_n \Lambda(n)\, \mathbf{1}_\epsilon(n/x)$

```
theorem SmoothedChebyshevDirichlet
        {SmoothingF : ℝ → ℝ} (diffSmoothingF : ContDiff ℝ 1 SmoothingF)
        (SmoothingFpos : ∀ x > 0, 0 ≤ SmoothingF x)
        (suppSmoothingF : Function.support SmoothingF ⊆ Set.Icc (1 / 2) 2)
        (mass_one : ∫ (x : ℝ) in Set.Ioi 0, SmoothingF x / x = 1) {X : ℝ}
        (X_gt : 3 < X) {ε : ℝ} (εpos : 0 < ε) (ε_lt_one : ε < 1) :
    SmoothedChebyshev SmoothingF ε X =
          ↑
          (∑' (n : ℕ), ArithmeticFunction.vonMangoldt n *
           Smooth1 SmoothingF ε (↑n / X))
```

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

PNT

$$\psi(x) := \sum_{n \leq x} \Lambda(n)$$

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

$$\psi(x) \cancel{=} \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \frac{x^s}{s} \, ds$$

How to make this all rigorous?

```
theorem SmoothedChebyshevDirichlet
    {SmoothingF : ℝ → ℝ} (diffSmoothingF : ContDiff ℝ 1 SmoothingF)
    (SmoothingFpos : ∀ x > 0, 0 ≤ SmoothingF x)
    (suppSmoothingF : Function.support SmoothingF ⊆ Set.Icc (1 / 2) 2)
    (mass_one : ∫ (x : ℝ) in Set.Ioi 0, SmoothingF x / x = 1) {X : ℝ}
    (X_gt : 3 < X) {ε : ℝ} (εpos : 0 < ε) (ε_lt_one : ε < 1) :
  SmoothedChebyshev SmoothingF ε X =
      ↑
      (∑' (n : ℕ), ArithmeticFunction.vonMangoldt n *
       Smooth1 SmoothingF ε (↑n / X))
```

Let $\psi_\epsilon(x) := \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \, \widetilde{\nu}(\epsilon s) \, \frac{x^s}{s} \, ds = \sum_n \Lambda(n) \, \mathbf{1}_\epsilon(n/x)$

Cost: $|\psi_\epsilon(x) - \psi(x)| \ll \epsilon \, x \log x$ (Preston Tranbarger)

```
theorem SmoothedChebyshevClose {SmoothingF : ℝ → ℝ}
    (diffSmoothingF : ContDiff ℝ 1 SmoothingF)
    (suppSmoothingF : Function.support SmoothingF ⊆ Icc (1 / 2) 2)
    (SmoothingFnonneg : ∀ x > 0, 0 ≤ SmoothingF x)
    (mass_one : ∫ x in Ioi 0, SmoothingF x / x = 1) :
    ∃ (C : ℝ), ∀ (X : ℝ) (_ : 3 < X) (ε : ℝ) (_ : 0 < ε) (_ : ε < 1) (_ : 2 < X * ε),
    ‖SmoothedChebyshev SmoothingF ε X - ChebyshevPsi X‖ ≤ C * ε * X * Real.log X := by
```

PNT

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

$$\psi(x) \neq \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \frac{x^s}{s} \, ds$$

How to make this all rigorous?

```
theorem SmoothedChebyshevDirichlet
    {SmoothingF : ℝ → ℝ} (diffSmoothingF : ContDiff ℝ 1 SmoothingF)
    (SmoothingFpos : ∀ x > 0, 0 ≤ SmoothingF x)
    (suppSmoothingF : Function.support SmoothingF ⊆ Set.Icc (1 / 2) 2)
    (mass_one : ∫ (x : ℝ) in Set.Ioi 0, SmoothingF x / x = 1) {X : ℝ}
    (X_gt : 3 < X) {ε : ℝ} (εpos : 0 < ε) (ε_lt_one : ε < 1) :
  SmoothedChebyshev SmoothingF ε X =
      ↑
       (∑' (n : ℕ), ArithmeticFunction.vonMangoldt n *
        Smooth1 SmoothingF ε (↑n / X))
```

Let $\psi_\epsilon(x) := \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \, \widetilde{\nu}(\epsilon s) \frac{x^s}{s} \, ds = \sum_n \Lambda(n) \, \mathbf{1}_\epsilon(n/x)$

Benefit: $|\widetilde{\nu}(\epsilon s)| \ll 1/(\epsilon |s|)$. So all integral exchanges are kosher

```
lemma MellinOfPsi {ν : ℝ → ℝ} (diffν : ContDiff ℝ 1 ν)
    (suppν : ν.support ⊆ Set.Icc (1 / 2) 2) :
    ∃ C > 0, ∀ (σ₁ : ℝ) (_ : 0 < σ₁) (s : ℂ) (_ : σ₁ ≤ s.re) (_ : s.re ≤ 2),
    ‖𝓜 (ν ·) s‖ ≤ C * ‖s‖⁻¹ := by
```

Cost: $|\psi_\epsilon(x) - \psi(x)| \ll \epsilon \, x \log x$ (Preston Tranbarger)

PNT

$$\Lambda(n) = \begin{cases} \log p & \text{if } n = p^k \\ 0 & \text{else} \end{cases}$$

$$\psi(x) := \sum_{n \le x} \Lambda(n)$$

$$\frac{\zeta'}{\zeta}(s) = \sum_n \frac{\Lambda(n)}{n^s}$$

$$\psi(x) \not= \frac{1}{2\pi i} \int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \, \frac{x^s}{s} \, ds$$

How to make this all rigorous?

```
theorem SmoothedChebyshevDirichlet
    {SmoothingF : ℝ → ℝ} (diffSmoothingF : ContDiff ℝ 1 SmoothingF)
    (SmoothingFpos : ∀ x > 0, 0 ≤ SmoothingF x)
    (suppSmoothingF : Function.support SmoothingF ⊆ Set.Icc (1 / 2) 2)
    (mass_one : ∫ (x : ℝ) in Set.Ioi 0, SmoothingF x / x = 1) {X : ℝ}
    (X_gt : 3 < X) {ε : ℝ} (εpos : 0 < ε) (ε_lt_one : ε < 1) :
  SmoothedChebyshev SmoothingF ε X =
      ↑
       (∑' (n : ℕ), ArithmeticFunction.vonMangoldt n *
        Smooth1 SmoothingF ε (↑n / X))
```

Let $\psi_\epsilon(x) := \dfrac{1}{2\pi i} \displaystyle\int_{(\sigma)} \frac{\zeta'}{\zeta}(s) \, \widetilde{\nu}(\epsilon s) \, \frac{x^s}{s} \, ds = \sum_n \Lambda(n) \, \mathbf{1}_\epsilon(n/x)$

Benefit: $|\widetilde{\nu}(\epsilon s)| \ll 1/(\epsilon |s|)$. So all integral exchanges are kosher

Cost: $|\psi_\epsilon(x) - \psi(x)| \ll \epsilon \, x \log x$ (Preston Tranbarger)

Let’s see how to coordinate everything over Zulip/Blueprint!