# Narrowing the gap between point-free topology and exact real computation (More questions than answers)

Paul Taylor
Honorary Senior Research Fellow,
School of Computer Science,
University of Birmingham

CCC 2022: Continuity, Computability, Constructivity
Università degli Studi di Padova
Lunedì 19 Settembre 2022
plus work done afterwards with Andrej Bauer at
Univerza v Ljubljani

## Abstract

In the continuity, computability and constructivity community, we have three well-established ways of presenting topology without points.

These correspond to different strengths of the foundations:

- classical point–set topology to the axiom of choice,
- locale theory to topos logic,
- formal topology to Martin-Löf type theory and
- abstract Stone duality to arithmetic universes.

The free arithmetic universe is composed of computable functions, which points towards computation.

On the other side, interval or "reliable" computation has a remarkable ability to give provably correct solutions to arbitrary precision to problems in integral calculus and solid modelling.

## Abstract (2)

However, whilst the programming is very clever, interval methods lack conceptual principles that can guide us past the original arithmetic operations in one real dimension.
Rather than defining some new kind of numbers, intervals should be seen as *using crude arithmetic to derive strong logical properties of functions.* Underlying the fact that it gives outer bounds to real-valued functions is that evaluating a *predicate* interval-wise provides an approximation to the *universal quantifier*. This is made precise by dividing up the interval sufficiently finely.
Using *back-to-front* intervals similarly approximates the *existential* quantifier, although this is poorly understood.
Similarly, the potentially chaotic behaviour of the Newton–Raphson algorithm may be tamed by replacing its *point-wise* differentials with interval-wise ones.

## Abstract (3)

These techniques were the basis of Andrej Bauer's *Marshall* calculator for evaluating Dedekind cuts. The operational principle is that each application of these ideas splits an interval into smaller ones on which some predicate is clearly true, clearly false or still indeterminate.
The next stage of this research is to devise new interval-like methods to extend these results to $\mathbb{R}^n$.
Dedekind cuts are easily equivalent to the formal points of point-free topology. These methods capture a problem in a *purely* mathematical way, whereas Cauchy sequences *pre-judge* the way a computation is to be done.
Beware that, because of this essential difference, *actual computation with cuts or formal points does not make traditional numerical analysis redundant, but demands that we re-formulate the whole of it within our subject.*

## Abstract (4)

That is the chasm, seen from the practical side, so what can we offer from the theoretical side?
In place of intervals, we need *parts* of a space that can be sub-divided.

- In Formal Topology, these capture basic *open* subspaces using an *infinitary* cover relation $\triangleleft$.
- In Abstract Stone Duality, they are intuitively *pairs* of subspaces, one *compact* and the other *open*, with a *finitary* cover relation $\ll$.

These axiomatisations are given respectively in the Formal Topology literature and my draft paper *Local Compactness and Bases in various formulations of Topology*,

`www.paultaylor.eu/ASD/loccbv`

and their similarity should promote a transfer of methods.

## Abstract (5)

Completeness, *i.e.* that an abstract $a \ll b$ actually captures $K_a \subset U_a$, for the four formulations of topology relies, as we expect, on the corresponding logic. The hardest case is the classical one, in general requiring not only Choice but proving the constructive one first, although the countable form can be done directly using Dependent Choice.
However, the account of $\ll$ goes further than completeness, refining the axioms so that a model of ASD can be constructed "with bare hands" from an arithmetic universe. In this, an auxiliary category naturally arises that is simpler than the one composed of continuous functions and has an *affine* product and related exponential.
The two are related by a structure-preserving functor. What is does concretely is to make the sufficiently fine sub-division that yields exact answers for ordinary interval methods,

## Abstract (6)

It is difficult to know where to go from here towards computation for locally compact spaces.
This is because $\mathbb{R}$ is the only space for which we know how to capture the topology entirely based on "arithmetic" operations plus the ASD axioms.
However, returning to theory, when the details of this "bare hands" model of ASD have been fully worked out, we should be able to state more precisely its relationship to the logic of arithmetic universes.
This question was first raised in my paper *Inside every model of Abstract Stone Duality lies an Arithmetic Universe* (namely its full subcategory of overt discrete spaces).

`www.paultaylor.eu/ASD/insema`

## Abstract (7)

The abstract for this talk *envisaged* discussing

- polishing the axioms for way below $\ll$
  (similar to those for covers $\triangleleft$)
  so that we could re-construct a category
  of locally compact spaces from these axioms
  with bare hands
  and hence get ideas for computation; and
- identifying the abstract ideas behind
  Andrej Bauer's Marshall calculator
  in order generalise them.

However, for time reasons, we can only talk about the second in this lecture.

## Levels of abstraction in topology

We study topology at many levels of generality and logical foundations, from the clouds down to the ground:

| | |
|---|---|
| descriptive set theory | |
| big CCCs | |
| point–set topology | |
| locale theory | formal points |
| formal topology | formal points |
| abstract Stone duality | formal points |
| $\mathbb{R}^n$ | |
| $\mathbb{R}$ | Dedekind reals |
| $\mathbb{R}$ | Cauchy reals |
| $\mathbb{N}$ | |

How can we exchange ideas amongst these levels
to derive practical calculation from abstract topology
and develop conceptual foundations for exact real and interval computation?

## What is the difference between Dedekind and Cauchy?

Dedekind reals are pairs $(D, U)$ of subsets of $\mathbb{Q}$ such that ...

Cauchy reals are sequences $(a_n)$ from $\mathbb{Q}$ such that ...

It's easy to turn a Cauchy real into a Dedekind real.

The converse uses Dependent Choice.

Wrong!

This misses the point of what they're for.

## What is the difference between Dedekind and Cauchy?

We learned in first year undergraduate real analysis that a Cauchy real is a sequence $(a_n)$ such that

$$\forall \epsilon > 0.\ \exists N.\ \forall mn > N.\ |a_n - a_m| < \epsilon.$$

However, as constructive mathematicians,
we re-learned that it needs a modulus of convergence

$$\mu : \mathbb{N}^{\mathbb{R}_+} \quad \text{such that} \quad \forall \epsilon > 0.\ \forall mn > \mu(\epsilon).\ |a_n - a_m| < \epsilon.$$

Moreover, $\mu$ may be chosen *once and for all* in our whole account of analysis. For example, we may choose

$$\mu(\epsilon) \equiv -\left\lfloor \log_{10}(\epsilon) \right\rfloor$$

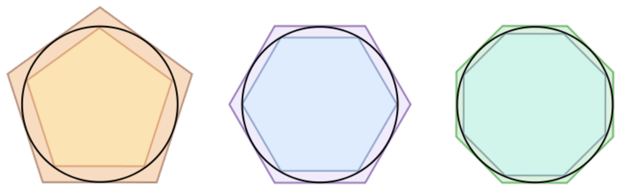
and then the Cauchy sequence is essentially a decimal expansion.

So a Cauchy real is a computation that
has already been done.

## What is the difference between Dedekind and Cauchy?

A Dedekind real is specified by its upper and lower bounds.

For example the areas or volumes of circum- and inscribed polygons or polyhedra:



This doesn't force the choice of any particular figures.
(Anyway, this Wikipedia image mis-represents Archimedes.)

## What is the difference between Dedekind and Cauchy?

A Dedekind cut specifies a computation
without prejudice as to how it is to be done.
(It could include many possible methods, thrown in together.)

So converting from a Dedekind real to a Cauchy real
amounts to choosing which way to do the computation.
(So obviously it requires some sort of "choice".)

## Dedekind reals and Formal Points

In Locale Theory, Formal Topology and Abstract Stone Duality,
a Formal Point is a predicate $P$ on (basic) opens $O$
such that $P(O)$ says whether the intended point is to lie in $O$.

(You know the axioms on $P$ that capture this.)

Dedekind cuts (in $\mathbb{R}^1$) yield formal points:

$$P(O) \quad\equiv\quad (D \cup O \cup U = \mathbb{R})$$

and *vice versa*:

$$\begin{aligned} d \in D &\quad\equiv\quad P(d, +\infty) \\ u \in U &\quad\equiv\quad P(-\infty, u) \end{aligned}$$

These formulae can be adapted to $\mathbb{R}^n$ too.

So Dedekind cuts and formal points are essentially the same.

## One of our intellectual gaps

Next we will look at evaluating formal points.

But only in $\mathbb{R}$ and maybe $\mathbb{R}^n$.

We can do this because the space $\mathbb{R}$ is
fully defined by its algebraic structure
(classically, a "complete ordered field")
together with the (ASD) axioms of topology:

```
        www.paultaylor.eu/ASD/dedras/axcomplete
```

We can do this for Cantor space $\mathbf{2}^{\mathbb{N}}$ too.

What other topological spaces
have some algebraic structure that fully defines them,
in an analogous way?

## The appropriate generality for Dedekind cuts

Any subsets $D, U \subset \mathbb{Q}$ or $\subset \mathbb{R}$ correspond to predicates on $\mathbb{Q}$ or $\mathbb{R}$.
Are these predicates of arbitrary logical complexity?

Forming a real number from a Dedekind cut
is an abstraction (in the sense of $\lambda$).

The corresponding application
is just arithmetic order, with $\beta$-rules

$$\mathsf{cut}\,(\delta, \upsilon) < d \iff \delta d \qquad \text{and} \qquad \mathsf{cut}\,(\delta, \upsilon) > u \iff \upsilon u$$

But $\{d \mid d < x\}$ and $\{u \mid x < u\}$ are open subsets of $\mathbb{R}$, not general ones.

The correspondence with Formal Points says the same.

(At least this is true in ASD. Do formal points in locale theory and formal topology also yield open cuts?)

## A language for Dedekind cuts

*A lambda calculus for real analysis* developed
a type theory for opens in $\mathbb{R}$, consisting of

- arithmetic expressions,
- arithmetic comparison $a < b$,
- finite conjunction and disjunction,
- universal quantification over compact subspaces, such as $[0, 1] \subset \mathbb{R}$,
- bounded existential quantification over overt subspaces, such as $[0, 1]$ or $(0, 1)$,
- unbounded existential quantification over overt subspaces, such as $\mathbb{N}$ or $\mathbb{R}$, and
- recursion for arithmetic and logic over $\mathbb{N}$.

## Computing with Dedekind cuts

Whenever we have a type theory we can try to do computation, or at least normalisation.

Given a pair of predicates (terms in this language),
we want to derive a convergent sequence of approximations
using simpler tests on rationals.

Thus we derive a Cauchy sequence from the Dedekind cut.

We start with alertsingle predicate,
consisting of a comparison between arithmetic expressions,
wlog $f(x) > 0$.

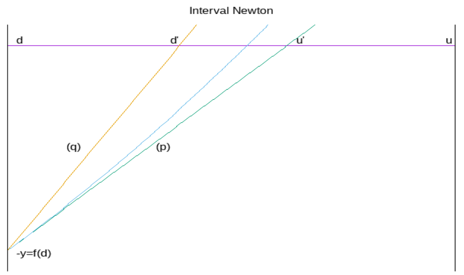
We extend this to single predicates in
the bounded and unbounded languages
and then to a pair of them forming a Dedekind cut.

## Arithmetical comparisons

These are the engine of the computation with Dedekind cuts.
Wlog we have a function $f : [d, u] \subset \mathbb{R} \to \mathbb{R}$
and want to find $d < d' < u' < u$ such that

$$f(x) < 0 \text{ if } d \le x \le d', \qquad \text{and} \qquad f(x) > 0 \text{ if } u' \le x \le u.$$

Suppose $f(d) \equiv -y < 0$ and $\forall x{:}[d, u].\ 0 < p < f'(x) < q$:



Iterating this gives the (Interval) Newton–Raphson algorithm.

## Interval Newton

In the Interval Newton algorithm,
we obtain the bounds $p$ and $q$ on $f'(x)$
by evaluating the differential interval-wise.

This eliminates the well known chaotic behaviour
of the traditional algorithm that uses point-wise differentials.

It converges quadratically,
so long as $f$ is twice differentiable,
just like the traditional algorithm,
*i.e.* it doubles the number of valid digits each time.

## Interval computation

Many "theoretical" treatments of interval computation say that interval-wise evaluation gives the set-theoretic image

$$f[d,u] \;=\; \{f(x) \mid d \le x \le y\}$$

Nonsense.

(As they well know) this is not even true for expressions, in particular if the variable is repeated.

It gives outer bounds on arithmetic expressions, but not tight bounds.

In fact it is better to assume that, if $d < u$, we always have

$$\lfloor f[d,u] \rfloor \quad < \quad f(x) \quad < \quad \lceil f[d,u] \rceil$$

for any $d \le x \le u$, even for constants and simple expressions.

## Quantifiers and domain theory

The formulae for the interval operations are not important.

They just provide some extension that

- agrees with the intended operations on the space ($\mathbb{R}$) itself, and
- preserves the order, which is reverse inclusion of intervals, or entailment between predicates.

From this it follows that, if $d \le x \le u$, so $[d,u] \supset [x,x]$, *i.e.* $[d,u] \le [x,x]$,

$$e < f[d,u] < t \quad\Longrightarrow\quad e < f(x) < t,$$

which we extend over the (order-preserving) logic to give

$$\phi[d,u] \quad\Longrightarrow\quad \forall x\!:\![d,u].\ \phi x.$$

So, interval-wise evaluation — however it is actually defined — is a lower approximation of universal quantification.

## What if we switch the order?

Suppose instead we define

$$\langle d,u\rangle \;\le\; \langle e,t\rangle \qquad \text{as} \qquad [d,u] \;\subset\; [e,t]\ ?$$

By the same reasoning, if $d \le x \le u$, so $\langle x,x\rangle \le \langle d,u\rangle$,

$$\phi x \;\equiv\; \phi\langle x,x\rangle \quad\Longrightarrow\quad \phi\langle d,u\rangle,$$

and therefore

$$\exists x\!:\![d,u].\ \phi x \quad\Longrightarrow\quad \phi\langle d,u\rangle.$$

So, reverse-interval evaluation — however it is defined — is an upper approximation of existential quantification.

Example (Ivo List)

$$\exists x\!:\![a,b].\ x > 0 \quad\Longrightarrow\quad \langle a,b\rangle > 0 \iff b > 0.$$

The extension of multiplication to *reverse* intervals is simply the reverse of Moore's multiplication for forward intervals.

The extension to *mixed* intervals was found by Edgar Kaucher in 1980.

## The big picture of the algorithm: framework

We have some predicate $\phi$ on a region $K$. Probably $K$ needs to be compact and overt.

We need a way of extending predicates to analogues of forward and reverse intervals.

In order to ensure that the approximations eventually get there, the interval operations must be Scott continuous.

For general overt and compact subspaces, we need the upper and lower powerdomains that Steve Vickers has studied extensively.

However, for "basic" intervals, the domain-theoretic construction in the completeness theorem for abstract bases provides an adequate analogue of the real interval domain.

## The big picture of the algorithm: engine

The engine of the algorithm is to split up the regions.

The predicate $\phi$ is built from finitely many arithmetic comparisons $f_i(x) > 0$ using the logical connectives.

We must split $K$ into sub-regions $K_n$ such that,
for each $K_n$ and $f_i$,

- it's true: $\forall x{:}K_n.\ f_i(x) > 0$,
- it's false: $\neg\exists x{:}K_n.\ f_i(x) > 0$,
- $f_i$ is still indeterminate on $K_n$, or
- we don't care about $f_i$ on $K_n$.

The value of whole expression $\phi$ throughout $K$ is then obtained using finitary $\wedge$ and $\vee$ from these sub-regions.
The reason why we *don't care* about some of the sub-regions is that $\top \vee ? \Leftrightarrow \top$ and $\bot \wedge ? = \bot$.
We want the indeterminate sub-regions to be smaller than some stated precision.

## How does this compute Dedekind cuts?

The cut is given by two predicates $(\delta, \upsilon)$
in the language that we have described.

The algorithm splits the whole line $\mathbb{R}$
into regions in which either

- $\delta$ is true and $\upsilon$ is false;
- they are indeterminate; or
- $\upsilon$ is true and $\delta$ is false.

These contain the (rational) numbers that are respectively

- less than the value of the cut;
- approximately equal to it; or
- greater than it.

## Splitting up regions in $\mathbb{R}^n$

Interval Newton shows how to do this in $\mathbb{R}^1$.

We need to generalise this to $\mathbb{R}^n$.

It's at least going to require linear algebra with intervals.

If you do a web search, you will find a whole downloadable book about linear algebra using intervals,

as if they were a new kind of number like the complex numbers.

Oh dear! Wrong again!

(So I'm not going to give the bibliographical details.)

## Splitting up regions in $\mathbb{R}^n$

The simplest way (given an orthodox maths education)
to prove correctness of the (1D) interval Newton algorithm
is to use the mean value theorem.

This has a textbook generalisation to $\mathbb{R}^n$,

$$\exists c{:}[0,1].\quad f(\vec{y}) \;=\; f(\vec{x}) \;+\; (\nabla f)(c\vec{x} + (1-c)\vec{y}) \cdot (\vec{y} - \vec{x}),$$

where $\nabla f$ is the gradient co-vector.

We need upper and lower bounds
on the components of $\nabla f$ over the given sub-region.

The discipline Optimisation provides these.

So it is apparently much more difficult than
just using the IEEE-defined rounding modes
of floating-point computation.

Then the dot-product is obtained by Moore-like operations.

## Splitting up regions in $\mathbb{R}^n$

By this method, for $f(\vec{y}) = 0$, this constrains $\vec{y}$ by hyper-planes defined by these components.

(Unfortunately, it's more complicated than this: there are higher analogues of Edalat's "bow ties".)

Even with that complication,
the regions that generalise intervals are
convex polyhedra bounded by hyper-planes.

The "indeterminate" regions are thin slabs
that bound segments of the surfaces $f_i(\vec{x}) = 0$,
quantifying the fact that $f_i$ is differentiable.

This is already going outside our community,
but that's only for $\mathbb{R}^n$.

What about more general spaces?
At least we have a conceptual structure.

## Back to the Marshall calculator

After CCC in Padova, I went to Ljubljana to work on these ideas further with Andrej Bauer.

In 2008, he made a prototype calculator, whose core code is in
`github.com/andrejbauer/marshall/tree/master/src/eval.ml`

This repeatedly refines a Dedekind cut, represented as a pair of predicates.

A key idea is to use upper and lower approximations,
in particular those for the quantifiers given by forward and reverse interval evaluation.

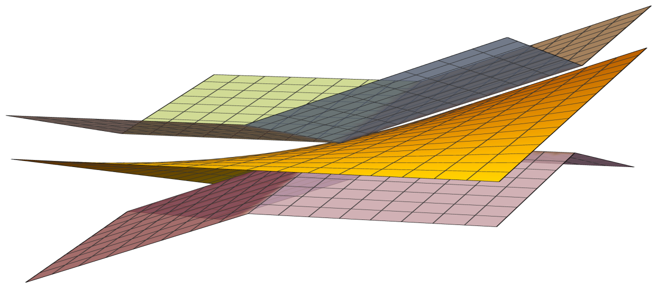
I have a different point of view, that any single predicate can be approximated.

Please note that these discussions are at a very early stage and that these notes are my own perspective on them, not necessarily his,

## Interval Newton in $\mathbb{R}^2$

Consider a function $f : \mathbb{R}^2 \to \mathbb{R}$,
whose graph is a curved (orange) surface in $\mathbb{R}^3$.

Whereas Interval Newton in 1D gives "bow ties", its analogue in 2D has more cases, arising from the way in which the "Moore dot product" of a vector with a gradient that is interval-valued in each coordinate has more cases.
So the curved surface is bounded by generalised "bow ties" like this:



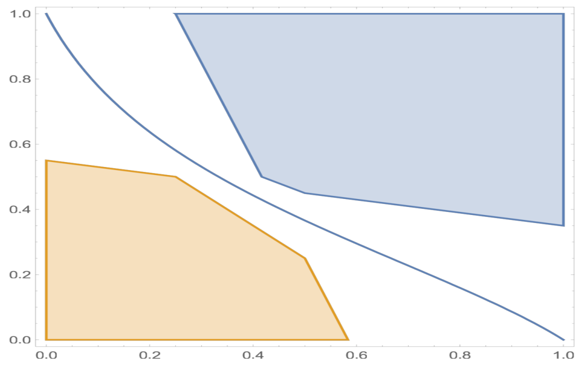
(This was drawn by Andrej Bauer.)

## Interval Newton in $\mathbb{R}^2$

When we intersect such a figure with the $f(x, y) = 0$ plane,
we get three polygonal regions, for which

$$f(x,y) < 0, \qquad \text{don't know}, \qquad f(x,y) > 0,$$

where the two definite cases mean that we have a *proof*.



(This too was drawn by Andrej Bauer, but his former student Ivo List had also obtained similar figures.)

## Interval Newton in $\mathbb{R}^2$

The white “don’t know” region is the one on which we will need to do further work by iterating the algorithm.
In particular it’s the region for which we will need bounds on the gradient (derivative).

But it’s much more complicated than the 1D version:

It’s not convex, so to avoid exponentially growing numbers of regions, we could form its convex closure.

It’s not parallel to the coordinate axes, so we must

- either rotate it
- or calculate bounds over a rhomboid.

This is now a task for (exact) numerical analysts!

## The bounded calculus

Next we add finite $\wedge$ and $\vee$
and bounded $\forall x : [a, b]$ and $\exists x : [a, b]$.

Point-wise, these are lattice operations over finitely many sub-regions.

$\forall$ and $\exists$ just ask whether all or some sub-regions are marked true.

Conceptually, this forms intersections and unions of sub-regions, but it may not be necessary to compute them explicitly.

Some “unknown” sub-regions get marked “don’t care” because they occur in $(-) \wedge \bot$ or $(-) \vee \top$ contexts.

The algorithm goes up and down the syntax tree,
invoking Interval Newton to split unknown regions further.

## Approximating opens in the bounded calculus

If the arithmetic comparisons are of polynomials with rational coefficients then (for $\mathbb{R}^1$, by 16th to 19th century algebra) decidably there is equality everywhere or there are finitely many sign-changes, with open intervals between them.

Interval Newton approximates this arbitrarily closely.

This result extends to the bounded calculus.

In particular every one-sided real so defined
has a partner forming a Dedekind cut
and Interval Newton generates
a Cauchy sequence with modulus of convergence.

This should also be valid for $\mathbb{R}^n$.

## The unbounded calculus

We know that there are one-sided cuts without partners.

They arise for computability reasons.

We would expect them to arise when we add

- unbounded quantification $\exists x : \mathbb{N}$ and $\exists x : \mathbb{R}$
- or general real coefficients
  expressed as embedded Dedekind cuts.

How do these affect the algorithm?

## The unbounded calculus

Any predicate (open subspace) in the bounded calculus
is a finite disjunction of conjunctions of basic ones.

The algorithm can approximate each one from the inside and outside, repeatedly, to obtain any degree of precision.
Interval Newton doubles the number of valid digits at each iteration, and this probably extends to the bounded calculus.

In the unbounded case there is an infinite disjunction.

Interval Newton gives no advice about the order in which to consider the disjuncts.

Probably any such open subspace can be approximated arbitrarily closely from the inside,
but with no modulus of convergence
and no outer bound.

## Locatedness in the unbounded calculus

Now consider a Dedekind cut, formed by
two open subspaces defined in the unbounded calculus.

Each gives an outer bound on the other.

But for any two distinct intermediate points,
at least one must be in either the upper or lower cut.

So (assuming we know how to evaluate such a predicate at a point) if we do both in parallel, one must terminate with a true value.
Then whichever disjunct succeeded
is the one to approximate next with Interval Newton.

This yields a Cauchy sequence with modulus that gives the value of the Dedekind cut.

## $\lambda$-terms and overt subspaces

Hitherto we have considered terms of type $\Sigma$
containing variable(s) of type $\mathbb{R}$.

But the ASD calculus was supposed to allow $\lambda$-abstraction.

It's strongly normalising, so assume that has been done.

What about free variables of type $\Sigma^{\mathbb{R}}$ (open subspaces)?

The most important case is when joins are preserved,
so it defines an overt subspace.

Then the variable can be a basic open region.

We could use the same methods to split it up.

This is (a bit) like solving for a point,
except there are many points.

## Generalising to other spaces

It is not difficult to see that this algorithm has

- an arithmetic part,
  consisting of Interval Newton, interval-wise evaluation and
  the 16th to 19th century algebra
  that splits regions on demand, and
- a logical part,
  that forms the lattice combinations,
  invokes region-splitting, and
  chooses amongst disjuncts.

The logical part knows nothing about $\mathbb{R}^n$
and can be formulated using
the axioms for abstract bases for locally compact spaces
(or maybe even the cover relation in Formal Topology).

## Division of intellectual labour

The arithmetic and logical parts of this algorithm
plainly belong to different communities.

It would be ridiculous for theorists to reproduce or compete
with what numerical analysts can do better.

When we theorists have
understood the logical side of this algorithm better
and defined a formal interface
to say what is required of the arithmetical part,
we can hand that over to our better-qualified colleagues.

If they then incorporate this into their tools, we will be better
able to work together to close the gap between point-free
topology and exact real computation.