Towards numerical integration in Coq

Bas Spitters (jww Eelis van der Weegen)

Radboud University Nijmegen

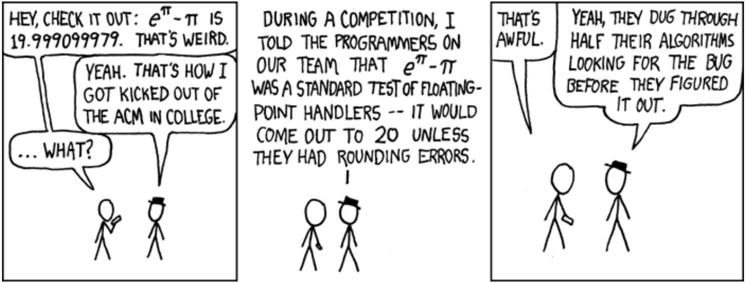
November 10, 2010

Context

Gap between theory and implementation of numerics.
The interval community started to narrow this gap.
Mathematically correct, but not formally provably so.
Are open to help from formal mathematics.

Context

Gap between theory and implementation of numerics.
The interval community started to narrow this gap.
Mathematically correct, but not formally provably so.
Are open to help from formal mathematics.
We need computations in formal proofs.


HEY, CHECK IT OUT: $e^{\pi} - \pi$ IS 19.999099979. THAT'S WEIRD.
YEAH. THAT'S HOW I GOT KICKED OUT OF THE ACM IN COLLEGE.
... WHAT?
DURING A COMPETITION, I TOLD THE PROGRAMMERS ON OUR TEAM THAT $e^{\pi} - \pi$ WAS A STANDARD TEST OF FLOATING-POINT HANDLERS -- IT WOULD COME OUT TO 20 UNLESS THEY HAD ROUNDING ERRORS.
THAT'S AWFUL.
YEAH, THEY DUG THROUGH HALF THEIR ALGORITHMS LOOKING FOR THE BUG BEFORE THEY FIGURED IT OUT.

Kantorovich

The Newton-Kantorivich theorem gives sufficient conditions for the convergence of Newton 's method.

Theorem: Let X and Y be Banach spaces and $F : D \subset X \to Y$. Suppose that on an open convex set $D_0 \subset D$, F is Frechet differentiable and

$$||F'(x) - F'(y)|| \leq K||x - y||, x, y \in D_0.$$

For some $x_0 \in D_0$, assume that $\Gamma_0 = [F'(x_0)]^{-1}$ is defined on all of Y and that $h := \beta K \eta \leq \frac{1}{2}$ where $||\Gamma_0|| \leq \beta$ and $||\Gamma_0 F x_0|| \leq \eta$. Set

$$t^* = \frac{1}{\beta K}(1 - \sqrt{1 - 2h}), \quad t^{**} = \frac{1}{\beta K}(1 + \sqrt{1 - 2h})$$

and suppose that $S := \{x \mid ||x - x_0|| \leq t^*\} \subset D_0$. Then the Newton iterates $x_{k+1} := x_k - [F'(x_k)]^{-1} F x_k$, $k = 0, 1, \ldots,$ are well defined, lie in S and converge to a solution x^* of $Fx = 0$ which is unique in $D_0 \cap \{x \mid ||x_0 - x|| < t^{**}\}$. Moreover, if $h < \frac{1}{2}$ the order of convergence is quadratic.

Methodology

Bishop: use contructive analysis as a programming language for numerical analysis
Martin-Löf: type theory as a language for constructive mathematics
Verified exact numerical analysis running inside Coq
Clean implementation first, speed up later

Overview

- Experiment building a library using type classes
- Faster real computation
- Numerical integration
- Picard method

Experiment building a library

Request for input

Three libraries: stdlib, corn, ssr.

ssr: solves many problems, but discrete

corn: computational continuous structures, needs updating

Experiment using type classes.

To be integrated with canonical structures $\rightarrow$ unification hints?

Dyadics

Improve efficency of the reals.
The current implementation (O'Connor) is fast, but can be improved.
Use dyadics instead of rationals, use machine integers (Krebbers)
Code refactoring, data structures ...
Example: verified plot of a circle.

Numerical integration

Riemann very slow, but general and verified!

Numerical integration

Riemann very slow, but general and verified!
Newton-Cotes:
Approximate a function by a polynomial and integrate this.

Lagrange polys

Definition

Let $x_1, \ldots, x_n$ be distinct and $y_1, \ldots, y_n$ arbitrary, then a unique polynomial L of degree at most $n-1$ exists with $L(x_k) = y_k$.

This polynomial is called the Lagrange polynomial.

Explicitly, $L(x) := \sum_j y_j \prod_{i, j \neq i} \frac{x - x_i}{x_j - x_i}$.

```
Definition L: cpoly CRasCRing :=
 Sigma (map (fun p => let '((x, y), rest) := p in
  _C_ y [*] Pi (map (fun xy' => (' (- fst xy') [+X*] One)
   [*] _C_ (' (/ (x - fst xy')))) rest)) (separates qpoints)).
```

Theorem (Lagrange error formula)

Let f be n times differentiable. Then for all x,
$|f(x) - P_n(x)| \leqslant \frac{\prod(x-x_k)}{n!} \sup|f^{(n)}|$.

Proof uses generalized Rolle's theorem.

This is a paradigmatic example.

Generalized Rolle

Theorem (Classical Rolle's theorem)

Let f be differentiable and have two zeroes in an interval $[a, b]$. Then f' has a zero in (a, b).

Theorem (Classical generalized Rolle's theorem)

Let f be n times differentiable and have $n+1$ zeroes in an interval $[a, b]$. Then $f^{(n)}$ has a zero in $[a, b]$.

Is not constructive, i.e. does not compute inside Coq.

Generalized Rolle

Three solutions:

- Approximate (ϵ) version
  Was used before in corn, Ugly
  The reason we have two libraries for reals in Coq?

Generalized Rolle

Three solutions:

- Approximate (ϵ) version
  Was used before in corn, Ugly
  The reason we have two libraries for reals in Coq?

Generalized Rolle

Three solutions:

- Approximate (ϵ) version
  Was used before in corn, Ugly
  The reason we have two libraries for reals in Coq?
- Generic zeroes using sheaf models
  Computational interpretation of classical logic a la Hilbert program
  Beautiful, but too early

Generalized Rolle

Bas Spitters (jww Eelis van der Weegen)

Three solutions:

- Approximate (ϵ) version
  Was used before in corn, Ugly
  The reason we have two libraries for reals in Coq?
- Generic zeroes using sheaf models
  Computational interpretation of classical logic a la Hilbert program
  Beautiful, but too early
- **Divided differences** (Thanks Henri)

Hermite-Genocchi formula

Replace Generalized Rolle by Hermite-Genocchi.
Let R be a field and $f : R \to R$. The interpolation polynomial in the Newton form is a linear combination of Newton basis polynomials

$$N(x) := \sum_{j=0}^{k} a_j n_j(x)$$

with the Newton basis polynomials defined as

$$n_j(x) := \prod_{i=0}^{j-1}(x - x_i)$$

and the coefficients defined as $a_j := f[x_0, \dots, x_j]$, where $f[x_0, \dots, x_j]$ is the notation for divided differences:

Hermite-Genocchi formula

divided differences defined recursively by:

$$f[a] = f(a)$$

$$f[a, b] = f(a) - f(b)/a - b$$

$$f[a, b, c] = f[a, c] - f[b, c]/a - b$$

and in general, $f[a : b : l] := f[a : l] - f[b : l]/a - b$.

Hermite-Genocchi formula

divided differences defined recursively by:

$$f[a] = f(a)$$

$$f[a, b] = f(a) - f(b)/a - b$$

$$f[a, b, c] = f[a, c] - f[b, c]/a - b$$

and in general, $f[a : b : l] := f[a : l] - f[b : l]/a - b$.
Would like: induction-recursion.
Program at Type level (=Equations?)

Hermite-Genocchi formula

divided differences defined recursively by:

$$f[a] = f(a)$$

$$f[a,b] = f(a) - f(b)/a - b$$

$$f[a,b,c] = f[a,c] - f[b,c]/a - b$$

and in general, $f[a:b:l] := f[a:l] - f[b:l]/a - b$.
Would like: induction-recursion.
Program at Type level (=Equations?)
Separate logic and computation:
lists without duplication, dummy values :-(

Hermite-Genocchi formula

divided differences defined recursively by:

$$f[a] = f(a)$$

$$f[a, b] = f(a) - f(b)/a - b$$

$$f[a, b, c] = f[a, c] - f[b, c]/a - b$$

and in general, $f[a : b : l] := f[a : l] - f[b : l]/a - b$.

Would like: induction-recursion.

Program at Type level (=Equations?)

Separate logic and computation:

lists without duplication, dummy values :-(

The Newton polynomial can be written as

$$N(x) := f[x_0]+f[x_0, x_1](x-x_0)+\cdots+f[x_0, ..., x_k](x-x_0)\cdots(x-x_{k-1})$$

Newton polynomial

```
Notation QPoint := (Q ∗ CR).
Fixpoint divdiff_l (a: QPoint) (xs: list QPoint): CR :=
 match xs with
  | nil => snd a
  | cons b l => (divdiff_l a l − divdiff_l b l) ∗ ' / (fst a −
      fst b)
 end.

Definition divdiff (l: ne_list QPoint): CR :=
 divdiff_l (head l) (tail l).

Let an (xs: ne_list QPoint): cpoly CRasCRing :=
 _C_ (divdiff xs) [∗] Pi (map (fun x => ' (− fst x) [+X∗]
     One) (tl xs)).

Definition N: cpoly CRasCRing := Sigma (map an (tails
     qpoints)).
```

Hermite-Genocchi formula

The Newton polynomial coincides with the Lagrange polynomial.

The divided difference $f[a_1, \ldots, a_n]$ is the coefficient of x^n in the (Newton) polynomial that interpolates f at $a_1, \ldots, a_n$.

Hermite-Genocchi formula

The Newton polynomial coincides with the Lagrange polynomial.
The divided difference $f[a_1, \ldots, a_n]$ is the coefficient of x^n in the (Newton) polynomial that interpolates f at $a_1, \ldots, a_n$.

$$f[a, b] = \frac{f(a) - f(b)}{a - b} = \int_0^1 f'(a + (b - a)t)dt.$$

Generally,

$$f[a_1, ..., a_n] = \iint_{n-1} f^{(n-1)}(u_1 a_1 + ... + u_n a_n) du_1 \cdots du_{n-1}$$

with $u_1 + \cdots + u_n = 1$ and $0 \leqslant u_i \leqslant 1$.

Hermite-Genocchi formula

The Newton polynomial coincides with the Lagrange polynomial.
The divided difference $f[a_1, \ldots, a_n]$ is the coefficient of x^n in the (Newton) polynomial that interpolates f at $a_1, \ldots, a_n$.

$$f[a,b] = \frac{f(a)-f(b)}{a-b} = \int_0^1 f'(a+(b-a)t)dt.$$

Generally,

$$f[a_1, \ldots, a_n] = \iint_{n-1} f^{(n-1)}(u_1 a_1 + \ldots + u_n a_n) du_1 \cdots du_{n-1}$$

with $u_1 + \cdots + u_n = 1$ and $0 \leqslant u_i \leqslant 1$.
Corollary,

$$f(x) - P_n f(x) = \prod_{i=1}^{n}(x - x_i) \iint_{n-1} f^{(n-1)}(u_1 a_1 + \ldots + u_n a_n) d\vec{u}$$

Hermite-Genocchi formula

The Newton polynomial coincides with the Lagrange polynomial.
The divided difference $f[a_1, \ldots, a_n]$ is the coefficient of x^n in the (Newton) polynomial that interpolates f at $a_1, \ldots, a_n$.

$$f[a,b] = \frac{f(a) - f(b)}{a - b} = \int_0^1 f'(a + (b-a)t)dt.$$

Generally,

$$f[a_1, ..., a_n] = \iint_{n-1} f^{(n-1)}(u_1 a_1 + ... + u_n a_n) du_1 \cdots du_{n-1}$$

with $u_1 + \cdots + u_n = 1$ and $0 \leqslant u_i \leqslant 1$.
Corollary,

$$f(x) - P_n f(x) = \prod_{i=1}^{n} (x - x_i) \iint_{n-1} f^{(n-1)}(u_1 a_1 + ... + u_n a_n) d\vec{u}$$

Replace differentiation by integration.

Simpson's rule

Corollary (Simpson's rule)

If $|f^{(4)}| \leqslant M$, *then*

$$|\int_a^b f(x)dx - \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]| \leqslant \frac{(b-a)^5}{2880}M.$$

The right hand side is the integral of the Lagrange polynomial P_3 at $a, \frac{a+b}{2}, b$. For the error we adopt the classical proof, but replace the use of Rolle's theorem and the Mean Value Theorem by the Hermite-Genocchi formula.

Simpson's rule

Corollary (Simpson's rule)

If $|f^{(4)}| \leqslant M$, *then*

$$|\int_a^b f(x)dx - \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]| \leqslant \frac{(b-a)^5}{2880}M.$$

The right hand side is the integral of the Lagrange polynomial P_3 at $a, \frac{a+b}{2}, b$. For the error we adopt the classical proof, but replace the use of Rolle's theorem and the Mean Value Theorem by the Hermite-Genocchi formula. Define $F(t) := f(\frac{a+b}{2} + \frac{b-a}{2}t)$. This reduces the problem to showing that $|\int_{-1}^{1} F(\tau)d\tau - \frac{1}{3}(F(-1) + 4F(0) + F(1)]| \leqslant N/90$, where $|F^{(4)}| \leqslant N$

Simpson's rule

Define

$$G(t) = \int_{-t}^{t} F(\tau)d\tau - \frac{t}{3}(F(-t) + 4F(0) + F(t)]$$

We need to prove that $90G(1) \leqslant \|F^{(4)}\|$.

Simpson's rule

Define

$$G(t) = \int_{-t}^{t} F(\tau)d\tau - \frac{t}{3}(F(-t) + 4F(0) + F(t)]$$

We need to prove that $90G(1) \leqslant \|F^{(4)}\|$.To do so, define $H(t) := G(t) - t^5 G(1)$. Then

$$H(0) = H(1) = H'(0) = H''(0) = 0.$$

Simpson's rule

Define

$$G(t) = \int_{-t}^{t} F(\tau)d\tau - \frac{t}{3}(F(-t) + 4F(0) + F(t)]$$

We need to prove that $90G(1) \leqslant \|F^{(4)}\|$.To do so, define $H(t) := G(t) - t^5G(1)$. Then

$$H(0) = H(1) = H'(0) = H''(0) = 0.$$

Hence, $H[0,0,0,1] = -(H[0,0,0] - H[0,0,1]) = 0 + (-H[0,0] + H[0,1]) = 0$.
Moreover, $H^{(3)}(t) = -\frac{t}{3}(F^{(3)}(t) - F^{(3)}(-t)) - 60t^2G(1) = -\frac{t}{3}(\int_{-t}^{t} F^{(4)}) - 60t^2G(1)$.

Simpson's rule

This shows that

$$
\begin{aligned}
0 = H[0,0,0,1] &= \int_0^1 H^{(3)} \\
&= \int_0^1 -\frac{t}{3}\Big(\int_{-t}^{t} F^{(4)}\Big) - 60t^2 G(1) \\
&\geqslant \int_0^1 -\frac{t}{3} 2tN - 60t^2 G(1) \\
&= -\frac{2}{3}(N + 90G(1)) \int_0^1 t^2 \\
&= -\frac{2}{3}(N + 90G(1)) \frac{1}{3}.
\end{aligned}
$$

Simpson's rule

This shows that

$$
\begin{aligned}
0 = H[0,0,0,1] &= \int_0^1 H^{(3)} \\
&= \int_0^1 -\frac{t}{3}\left(\int_{-t}^{t} F^{(4)}\right) - 60t^2 G(1) \\
&\geqslant \int_0^1 -\frac{t}{3} 2tN - 60t^2 G(1) \\
&= -\frac{2}{3}(N + 90G(1)) \int_0^1 t^2 \\
&= -\frac{2}{3}(N + 90G(1))\frac{1}{3}.
\end{aligned}
$$

Hence, $N \geqslant -90G(1)$. Similarly, $0 \leqslant -\frac{2}{9}(-N + 90G(1))$. Consequently, $90G(1) \leqslant N$. We conclude that $|90G(1)| \leqslant N$.

Differentiation over general fields [Bertrand, Glöckner, Neeb]

> *The proofs are 'algebraic' in nature and in this way become often simpler and more transparent even than the usual proofs in $\mathbb{R}^n$ because we avoid the repeated use of the Mean Value Theorem (or of the Fundamental Theorem) which are no longer needed once they are incorporated in [the definition of the derivative by a difference quotient].*

Picard existence theorem

Given the initial value problem:

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad t \in [t_0 - \alpha, t_0 + \alpha]$$

Suppose f is Lipschitz continuous in y and continuous in t. Then, for some $\varepsilon > 0$, there exists a unique solution y(t) to the initial value problem within the range [t0 $-$ ε,t0 $+$ ε].

Proof of Picard method

Picard iteration:
Set $\varphi_0(t) = y_0$ and

$$\varphi_i(t) = y_0 + \int_{t_0}^{t} f(s, \varphi_{i-1}(s))\, ds.$$

The sequence of Picard iterates φ_i is convergent and that the limit is a solution to the problem. The width of the interval where the local solution is defined is entirely determined by the Lipschitz constant.

Picard iteration

Consider a concrete C^∞ function, say $\lambda x.sin(sinx)$
To compute the integral we need an upper bound on the derivative.
Cruz-Filipe's tactic automatically finds a provable derivative.
The sup function (O'Connor/S) computes bound.
Finally, we apply Simpson's rule.

Demo

Simpson's rule in Coq.