

CoCoALib - Slug #897

SimplestBigRatBetween: why is it so slow?

25 Jun 2016 18:58 - John Abbott

| | | | |
|--|------------------|------------------------|--------------------|
| Status: | Closed | Start date: | 25 Jun 2016 |
| Priority: | Normal | Due date: | |
| Assignee: | John Abbott | % Done: | 100% |
| Category: | Improving | Estimated time: | 2.49 hours |
| Target version: | CoCoALib-0.99800 | Spent time: | 2.45 hours |
| Description | | | |
| Investigations in issue #860 showed that SimplestBigRatBetween was mysteriously slower than expected (and than an equivalent implementation in RingTwinFloat). | | | |
| Find out why; and hopefully improve the code. | | | |
| Related issues: | | | |
| Related to CoCoALib - Bug #860: Check impl of RingTwinFloatImpl::mylsRational | | Closed | 30 Mar 2016 |

History

#1 - 25 Jun 2016 19:02 - John Abbott

- Related to Bug #860: Check impl of RingTwinFloatImpl::mylsRational added

#2 - 19 Oct 2020 15:22 - John Abbott

Why did I give no examples to test this on?

After reading [#860](#), a possible test might be

```
q := fibonacci(401)/fibonacci(400);  
eps := 2^(-553);  
q2 := SimplestRatBetween(q-eps,q+eps); // should be same as q
```

#3 - 19 Oct 2020 17:38 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

The profiler suggests that the reciprocal $1/(\text{myFrac}-\text{myQuot})$ is surprisingly costly. I shall modify operator/ to handle reciprocals specially to see if that makes it any faster.

#4 - 20 Oct 2020 11:39 - John Abbott

- % Done changed from 10 to 20

I have now put special handling in for reciprocals, and the code runs a bit faster.

Probably the best solution would be to make ContFracIter use BigInt instead of BigRat... should that be a new issue?

#5 - 20 Oct 2020 12:59 - John Abbott

Here is the speed test I used:

```
q := fibonacci(4001)/fibonacci(4000);
eps := 2^(-5553);
t0 := CpuTime();
for i := 1 to 1000 do
  q2 := SimplestRatBetween(q-eps,q+eps); // should be same as q
endfor;
println "Time: ", TimeFrom(t0);
```

2020-10-20: took about 7.5s (on my linux box)

#6 - 21 Oct 2020 13:38 - John Abbott

- Assignee set to John Abbott

- % Done changed from 20 to 70

I replaced used of BigRat (field myFrac) inside ContFractler by a pair of BigInt values (myNum & myDen). Now the speed test above takes 2.2s; I'm (pleasantly) surprised by the speed gain.

#7 - 22 Oct 2020 16:39 - John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99800

#8 - 03 Nov 2020 21:03 - John Abbott

- Status changed from In Progress to Feedback

- % Done changed from 70 to 90

I have checked in (despite the presence of some experimental changes).

#9 - 16 Sep 2021 13:12 - John Abbott

- % Done changed from 90 to 100

- Estimated time set to 2.49 h

Closing after 11 months in feedback.

#10 - 16 Sep 2021 13:47 - John Abbott

- Status changed from Feedback to Closed