Rapid computation of Bernoulli and related numbers.

Goal.

The goal of this poster is to explore multisectioning and it's application to rapid computation of Bernoulli and related numbers

Definitions and Theorems.

Definition 1 (Poly-exponential function.) *Let λ_1, ..., λ_n $\in \mathbb{C}$ be constants and $p_1(x)$, ..., $p_n(x) \in \mathbb{C}[x]$ be polynomials. Then*

$$\sum_{i=1}^{n} p_i(x) e^{\lambda_i x},$$

*is a "*poly-exponential function*". Denote the set of all such functions by $\mathcal{P}$.*

Definition 2 (Rational poly-exponential function.) *Let* $s(x)$, $t(x) \in \mathcal{P}$ *and* $t(x) \neq 0$. *Then*

$$\frac{s(x)}{t(x)},$$

*is a "*rational poly-exponential function*". Denote the set of all such functions by* $\mathcal{R}$.

Definition 3 (Multisectioning.) *Let* $f(x)$ *be a function acting on a subset of* $\mathbb{C}$. *Define* $f_m^q(x) = \frac{1}{m}\sum_{i=0}^{m-1} \omega_m^{-iq} f(\omega_m^i x)$, *where* $\omega_m = e^{\frac{2\pi i}{m}}$.

Theorem 1 *Given a function* $f(x) \in \mathcal{R}$, $m, q \in \mathbb{Z}$, $0 \leq q < m$, *a recursion formula can be found for the* $mi + q$*-th coefficient of the exponential generating function of* $f(x)$ *that depends only on the* $mj + q$*-th coefficient, for* $j < i$, *and two lacunary recurrence relations.*

Proof: All equivalences are to be considered to be equivalences modulo m.

If $f(x) = \frac{s(x)}{t(x)}$ then

$$f_m^q(x) = \frac{S_m^q(x)}{T_m^0(x)}$$

where $S(x) = s(x)\prod_{i=1}^{m-1} t(x\omega_m^i)$ and $T(x) = \prod_{i=1}^{m} t(x\omega_m^i)$.

Further

$$S_m^q(x) = \sum_{i\equiv q} b_i \frac{x^i}{i!},$$

where the the coefficients b_i satisfy a linear recurrence relation.

Also

$$T_m^0(x) = \sum_{j\equiv 0} d_j \frac{x^j}{j!},$$

where the the coefficients d_i satisfy a linear recurrence relation.

So if $f(x) = \sum_{k=0}^{\infty} c_k \frac{x^k}{k!}$ then $f_m^q(x) = \sum_{k \equiv q} c_k \frac{x^k}{k!}$ and we have

$$\begin{aligned}
\sum_{k \equiv q} c_k \frac{x^k}{k!} &= \frac{\sum_{i \equiv q} b_i \frac{x^i}{i!}}{\sum_{j \equiv 0} d_j \frac{x^j}{j!}} \\
\sum_{j \equiv 0} d_j \frac{x^j}{j!} \sum_{k \equiv q} c_k \frac{x^k}{k!} &= \sum_{i \equiv q} b_i \frac{x^i}{i!} \\
\sum_{i \equiv q} \sum_{l \equiv 0} \binom{i}{l} d_l c_{i-l} \frac{x^i}{i!} &= \sum_{i \equiv q} b_i \frac{x^i}{i!} \\
\sum_{l \equiv 0} \binom{i}{l} d_l c_{i-l} &= b_i
\end{aligned}$$

Thus we get a recurrence relation for the c_i which depend only on those c_i where $i \equiv q \pmod{m}$ and two linear recurrence relations for the d_i and the b_i.

$\blacksquare$

Problem.

How do you determine the linear recurrence relation for b_is and the d_is in the proof?

Denominator.

To find the for the recurrence for the d_i, it is sufficient to calculate the recurrence for $T(x) = \prod_{i=1}^{m} t(x\omega_m^i)$. This can be done using a number of methods.

1. Let $P^t(x)$ and $P^s(x)$ be the recurrence polynomial associated with t and s respectively. Then
$$P^{st}(x) = \mathrm{Res}_y(P^s(x-y), P^t(y)).$$
Thus resultants can be used.

2. Given a bound on the size of the recurrence m of a function t, and $\hat{t}(x)$ a Taylor series approximate of t to degree $2m$, then the recurrence for t can be recovered via linear algebra. This combined with fast polynomial multiplication used to determine a recurrence for $\prod_{i=1}^{m} t(x\omega_m^i)$.

3. Given $T(x)$, and a bound for the size of its recurrence, a Taylor series can be determined for $T(x)$ via symbolic differentiation and the recurrence determined from this via linear algebra.

Numerator.

To find the recurrence for the b_i, it is sufficient to calculate the recurrence for $S(x) = (s(x)\prod_{i=1}^{m-1} t(x\omega_m^i))_m^q$. This can be done using the methods used for the d_is plus the additional method of

4. Knowing the recurrence for the d_i, a bound on the size of the recurrence for the b_i, and the first few initial values of the c_is then the terms of a Taylor series polynomial approximation of $S(x)$ can be determine, and linear algebra used to determine from this the recurrence for the b_is.

References

[1] *Cecm research projects*, http://www.cecm.sfu.ca/projects, 1999.

[2] Bruce C. Berndt, *Ramanujan's notebooks*, Springer-Verlag, New York, 1994.

[3] Jonathan Borwein, Peter Borwein, and Lennart Berggren, *Pi: A source book*, Springer, New York, 1997.

[4] Jonathan M. Borwein, David M. Bradley, and Richard E. Crandall, *Computational strategies for the Riemann zeta function*, unpublished, 1996.

[5] Mustapha Chellali, *Accélération de calcul de nombres de Bernoulli*, Journal of Number Theory (1988), 347–362.

[6] Louis Comtet, *Advanced combinatorics, the art of finite and infinite expansions*, D. Reidel Publishing Company, Boston, 1974.

[7] K.O. Geddes, G. Labahn, M. B. Monagan, and S. Vorketter, *The maple programming guide*, Springer-Verlag, New York, 1996.

[8] J. W. L. Glaisher, *On Eulerian numbers*, Quarterly Journal of Mathematics **45** (1914).

[9] Kevin Hare, *Multisectioning, rational poly-exponential functions and parallel computation*, Master's thesis, Simon Fraser University, Vancouver, April 1999.

[10] D.H. Lehmer, *Lacunary recurrence formulas for the numbers of Bernoulli and Euler*, Annals of Mathematics **36** (1935), no. 3.

[11] S Ramanujan, *Some properties of Bernoulli's numbers*, Indian Mathematical Journal (1911).

[12] Herbert S Wilf, *Generating functionology*, Academic Press, Inc., Toronto, 1990.

Bernoulli numbers.

Introduction.

```
> with(MS):
```

Consider the ***Bernoulli numbers*** B_i in the exponential generating function of $\frac{x}{\mathbf{e}^x - 1} = \sum_{i=0}^{\infty} \frac{B_i x^i}{i!}$. Let us find a ***linear recursion formula*** for the Bernoulli numbers where we need only every 8-th term starting at 0. (i.e. find the recurrence for the b_i and the d_i in the proof given that $m = 8$ and $q = 0$.)

Finding the linear recurrence relation for the denominator (the d_i).

First we need to find a ***linear recurrence relation*** for the denominator (the d_i) using method 2.

```
> bottom := `egf/strip` ( `bottom/ms/linalg/fft2` ( exp(x)
  - 1 , f , x , 8) , 8 , 0 );
```

$$bottom := f(x) = -6879707136\, f(x-80) + 7884395020288\, f(x-72) + 63251706062592\, f(x-64) - 39689600177520\, f(x-56) - 33612903319512\, f(x-48) + 2172936331881\, f(x-40) + 126650763\, f(x-32) + 216719850\, f(x-24) + 2419902\, f(x-16) - 1107\, f(x-8), f, x, [f(0)=0, f(8) = -40320, f(16) = 17297280, f(24) = -106871377536, f(32) = 150374230611840, f(40) = -421347447269274240, f(48) = 807075270214352233344, f(56) = -188043737150293282843200\,0, f(64) = 39431591157236750823116\,10240, f(72) = -874036508455963591861499966\,9376, f(80) = 188092962792295149304681722\,08012160]$$

```
> Bot := `egf/makeproc`( bottom ):
```

Finding the linear recurrence relation for the numerator (the b_i).

Next we find a ***linear recurrence relation*** for the numerator (the b_i) using method 4.

```
> top := `egf/strip`( `top/ms/linalg/know` ( Bot ,
  bernoulli , f , x , 8 , 0 , 22 ) , 8 , 0 );
```

$$top := f(x) = -4733037027712932249\,6\, f(x-160)$$

$+ 1084846573682364367503336\,\mathrm{f}(x-152)$

$- 6129337840881623464148992\,0\,\mathrm{f}(x-144)$

$- 99794897826037387129729843 2\,\mathrm{f}(x-136)$

$- 337538384169647459980758220 8\,\mathrm{f}(x-128)$

$+ 555093456177133043319518822 4\,\mathrm{f}(x-120)$

$+ 264261802405446570596992588 8\,\mathrm{f}(x-112)$

$- 294305124157521191333739187 2\,\mathrm{f}(x-104)$

$- 95736076024991683730299555 2\,\mathrm{f}(x-96)$

$+ 14605999716279143681757460 8\,\mathrm{f}(x-88)$

$- 4696241162714240047787457\,\mathrm{f}(x-80) + 14210988582915560301338\,\mathrm{f}(x-72)$

$- 77926079455936300631 7\,\mathrm{f}(x-64) - 1064597976665223703 2\,\mathrm{f}(x-56)$

$- 4283680302124744 2\,\mathrm{f}(x-48) - 104425531945635 6\,\mathrm{f}(x-40)$

$- 5375854640178\,\mathrm{f}(x-32) + 5791102728\,\mathrm{f}(x-24) + 3614355\,\mathrm{f}(x-16)$

$- 2214\,\mathrm{f}(x-8), f, x, [\,\mathrm{f}(0) = 0, \mathrm{f}(8) = -40320, \mathrm{f}(16) = 34594560,$

$\mathrm{f}(24) = -320614132608, \mathrm{f}(32) = 601496922447360, \mathrm{f}(40) = -2106737236346371200,$

$\mathrm{f}(48) = 4842451621286113400064, \mathrm{f}(56) = -13163061600520529799024000,$

$f(64) = 31545272925789400658492881920,$

$f(72) = -78663285761036723267534997024384,$

$f(80) = 188092962792295149304681722080121600,$

$f(88) = -452282900924712489348203849291361886080,$

$f(96) = 1069619229367364610812066520300211765467648,$

$f(104) = -2523127234942034251979935830223894510887411840,$

$f(112) = 5902739723040718443367768858532713037965504876800,$

$f(120) = -13755834202693720181988575320622318755673966511354240,$

$f(128) = 31893261425290046006071293170793426026128827955137607680,$

$f(136) = -73682333399140263527340042995210720310838170821015815885440,$

$f(144) =$

$16960617245919718801540278893915676661175529806005155236197555 2,$ $f(152)$

$= -38924257388673073389086502251182596145653458612381769079632225 9840$

]

Timings.

Introduction.

```
> with(MS):
```

Consider the ***Euler numbers*** E_i in the exponential generating function of $\frac{2}{\mathrm{e}^x + \mathrm{e}^{(-x)}} = \sum_{i=0}^{\infty} \frac{E_i\, x^i}{i!}$. Let us use the ***linear recurrence relations*** for the denominator (d_i) and the numerator (b_i) to calculate the first 3000 Euler numbers (every 12-th one).

```
> N := 3000:
```

Calculating using my code.

```
> tt := time():
> `calcul/normal`(N, Top, Bot, 12, 0):
> time_mine := time() - tt;
```

$$time\_mine := 450.979$$

Calculating using the default code.

```
> tt := time():
> euler(N):
> time_normal := time() - tt;
```

$$time\_normal := 4529.718$$

Comparison.

```
> time_normal/time_mine;
```

$$10.04418831$$

```
>
```

So we see that there is an improvement, even after taking into account that only 1/6-th of the information is calculated in the first case.