Mathematical Miniatures

Tetration

Gottfried Helms - Univ Kassel 04 - 2008

"Exponential polynomial interpolation"

An interpolation technique for general iterates of the U-function

Abstract: For the fractional iteration of the function U(x) = exp(x)-1 (also named as "dxp()"-function, see[TF08-1]) a polynomial interpolation for the coefficients of the required powerseries is known and is described in various papers. For the more general case $U_t(x) = t^x - 1$ this polynomial interpolation cannot be applied. Instead, the diagonalization method for the according matrix-operator can be used.

Here I propose another matrix-based interpolation-method for the general case which does not need diagonalization. The core logic follows the idea of polynomial interpolation and it seems that it gives the same result as diagonalization – but the latter is not proved.

Gottfried Helms, 14.07.2008 Textversion 1.3

The decoding of the M-matrix appended. Heuristically it is the q-analogue of the pascal-matrix (see Pg 8-9) 09.09.2009

Contents

1. Definition and basic properties 2
1.1. Definition of the base-function 2
1.2. First step into the problem 3
1.3. Review: the ordinary polynomial interpolation 5
1.4. The matrix-approach to polynomial interpolation 6
1.5. An "exponential-polynomial" approach to interpolation 8
1.6. Comparision of "exponential polynomial interpolation" with diagonalization 10
1.7. Conclusion 10
2. Appendix 11
3. References 13

History: Textversion

1.4	*09.09.2009*	*Q-analogues inserted (chap 1.5),*
1.3	*16.09.2008*	*minor text-/style-edits,*
1.0	*14.07.2008*	*first version*

Project-Index	http://go.helms-net.de/math/tetdocs
Intro/notation	http://go.helms-net.de/math/tetdocs/intro.pdf

1. Definition and basic properties

1.1. Definition of the base-function

Define the function

$U(x) = \exp(x) - 1$

and the iteration-notation introducing the iteration-height parameter h:

$U^{\circ 0}(x) = x$
$U^{\circ 1}(x) = U(x)$
$U^{\circ h}(x) = U^{\circ h-1}(U(x))$

Then define for arbitrary t and $u=\log(t)$

$U_t(x) = t^x - 1$
$U_t^{\circ 0}(x) = x$
$U_t^{\circ 1}(x) = U_t(x)$
$U_t^{\circ h}(x) = U_t^{\circ h-1}(U_t(x))$

In the current text I first review the polynomial interpolation for $t=\exp(1)$, $u=1$ and shall then introduce a sort of "exponential polynomial interpolation" using general t (for the numerical examples I use $t=2$, $u=\log(2)$), which seems to be new.

All derivations are done in the matrix-notation; however the polynomial interpolation can be re-expressed in common serial notation; here, the matrix-notation is simply meant to keep the formulae concise. The used matrix-operators for the U-tetration are named $\mathbf{U}$ resp. $\mathbf{U_t}$ and the other involved standard-matrices are as follows:

$V(x)$:= $\text{column}_{r=0..inf}[1,x,x^2,x^3,...x^r,...]$
an infinite "vandermonde" (column-) vector of a variable x
$V(x)\sim$:= *the transpose ; the symbol is taken from the convention in Pari/GP*
${}^dV(x)$:= *the diagonal arrangement of V(x)*

VZ := $\text{matrix}_{r=0..inf,c=0..inf}[c^r]$
:= $[V(0),V(1),V(2),...]$
the collection of V()-vectors of consecutive parameters

P := $\text{matrix}_{r=0..inf,c=0..inf}[\text{binomial}(r,c)]$
which is the lower-triangular Pascal-matrix of infinite size

${}^dF$:= $\text{diag}(0!,1!,2!,...)$

$S2$:= $\text{matrix}_{r=0..inf,c=0..inf}[s2_{r,c}]$ // s2 Stirling-numbers 2nd kind
$S1$:= $\text{matrix}_{r=0..inf,c=0..inf}[s1_{r,c}]$ // s1 Stirling-numbers 1st kind

$M[,c]$:= *the $(c+1)^{th}$ column of a matrix M (=column c), c beginning at zero*

Then

U_t := $\text{matrix}_{r=0..inf,c=0..inf}[u^r * s2_{r,c}/r!\,] = {}^dV(u) * {}^dF^{-1} * S2$

to the effect that the function $U_t^{\circ h}(x)$ can be computed by the dot-product of $\mathbf{V}(x)$ and column 1 of the hth-power of $\mathbf{U_t}$ (which contains the required coefficients of the powerseries)

$V(x)\sim * U_t^h \quad = V(U_t^{\circ h}(x))\sim$

and finally

$V(x)\sim * U_t^h[,1] \quad = U_t^{\circ h}(x)$

1.2. First step into the problem

In some recent contributions in various forums I speculated that the commonly used polynomial interpolation-approach for the fractional iteration is not the only choice and may not be sufficient for cross-base-compatibility of **U**-tetration to give consistent results for **T**-tetration (if computed by different fixpoint-shifts).

Thinking about alternatives I developed the following interpolation concept which is actually not an alternative but an interesting generalization of the common polynomial interpolation on its own.

The starting point: if we consider the powerseries for $U_t^{\circ h}(x)$ for increasing heights h, we may arrange the coefficients in a matrix ***LIST*** where the rows are related to the powers of x and the h^{th} column contains the coefficients for the powerseries for a height h:

LIST: Table of coefficients of powerseries of $U_t^{\circ h}(x)$ (using u=log(t)) for consecutive heights h

	h=0	h=1	h=2	h=3	h=4	h=5
	$*u^{-1}$	$*u^{0}$	$*u^{1}$	$*u^{2}$	$*u^{3}$	$*u^{4}$
1*	0	0	0	0	0	0
xu*	1	1	1	1	1	1
$(xu)^2/2!*$	0	1	$1+ 1u$	$1+1u+1u^2$	$1+1u+1u^2+1u^3$	1+1u+…
$(xu)^3/3!*$	0	1	$1+ 3u+ 1u^2$	$1+3u+4u^2+3u^3+1u^4$	1+3u+…	1+3u+…
$(xu)^4/4!*$	0	1	$1+ 7u+ 6u^2+ 1u^3$	1+7u+…	1+7u+…	1+7u+…
$(xu)^5/5!*$	0	1	$1+15u+25u^2+10u^3+1u^4$	1+…	1+…	1+…
…	…	…	…	…	…	…

Here the entries become quickly complicated so I used periods; the entries result from consecutive powers of the matrix of stirling-numbers *2*nd-kind. The powerseries for the iterated function of a certain iteration-height h is constructed by the product of entries of the h'th column and the cofactors at the according column- and rowheads.

To have a better impression here is the table using $t=2$, $u = \log(t) = 0.6931\ldots$:

LIST: Table of coefficients of powerseries of $U_t^{\circ h}(x)$ (using t=2) for consecutive heights h

	h=0	h=1	h=2	h=3	h=4	h=5
	0	0	0	0	0	0
x*	1	0.69314718	0.48045301	0.33302465	0.23083510	0.16000270
x^2*	0	0.24022651	0.28192988	0.25087161	0.20053337	0.15179957
x^3*	0	0.055504109	0.13695810	0.16616714	0.15736842	0.13200242
x^4*	0	0.0096181291	0.060452839	0.10396073	0.11851749	0.11107001
x^5*	0	0.0013333558	0.024925637	0.062643486	0.087004184	0.091650610
x^6*	0	0.00015403530	0.0097157208	0.036673553	0.062686099	0.074594932
x^7*	0	0.000015252734	0.0036118574	0.020972661	0.044506375	0.060081259
x^8*	0	0.0000013215487	0.0012895308	0.011757701	0.031217336	0.047984406
x^9*	0	0.00000010178086	0.00044450779	0.0064780572	0.021669372	0.038052253
…	…	…	…	…	…	…

So, for example, the powerseries for $U_t^{\circ 2}(x)$ is defined by the coefficients of column $h=2$:

$$U_t^{\circ 2}(x) = 0.4804\ldots x + 0.2819\ldots x^2 + 0.1369\ldots x^3 + \ldots$$

The usual approach to determine the coefficients of fractional heights would be now to use a polynomial interpolation procedure for each row; for the fractional height $h=2.5$ for instance some interpolation between the coefficients in the columns $h=2$ and $h=3$ – applicable if a polynomial interpolation method yields finite polynomials.

But since the second row contain the powers of *log(t) (=u)* this is obviously not applicable here. Even if we divided each column by the according power of *u* we get no finite polynomials but an infinite composition by all columns from the third row on:

Table of useless coefficients for polynomial (series) computation of **LIST** *depending on h*

	*1	*h	*h^2	*h^2	*h^2	
	0	0	0	0		
x*	1	0	0	0	0	
x^2*	0	0.34657359	-0.10634708	0.032632902	-0.010013498	...
x^3*	0	0.080075502	0.12490934	-0.11599105	0.075941376	...
x^4*	0	0.013876027	0.098072610	-0.023674629	-0.035812013	...
x^5*	0	0.0019236258	0.048032196	0.038237187	-0.071925875	...
x^6*	0	0.00022222597	0.019777548	0.050123316	-0.048485204	...
x^7*	0	0.000022005043	0.0074735978	0.040489493	-0.014081651	...
x^8*	0	0.0000019065917	0.0026801761	0.027259549	0.010109643	...
x^9*	0	0.00000014683874	0.00092489104	0.016677072	0.021615598	...
...						

However – if we use u^h instead of *h* as interpolation-parameter, we actually find finite polynomials – but now in u^h –

Table of coefficients for polynomials in u^h for fractional iterates of $U_t^{\circ h}(x)$, t=2, u=log(2)

	*1	*u^h	*u^{2h}	*u^{3h}	*u^{4h}	
	0	0	0	0	0	
x*	0	1.0000000	0	0	0	
x^2*	0	1.1294457	-1.1294457	0	0	
x^3*	0	1.1985847	-2.5512951	1.3527103	0	
x^4*	0	1.2474591	-4.1482473	4.5834386	-1.6826504	
x^5*	0	1.2856301	-5.8758179	10.040759	-7.6018490	...
x^6*	0	1.3170719	-7.7093027	17.998582	-20.946010	...
x^7*	0	1.3439053	-9.6326054	28.687260	-45.427810	...
x^8*	0	1.3673703	-11.634375	42.307750	-85.216714	...
x^9*	0	1.3882575	-13.706120	59.039686	-144.89184	...
...		...	...	...	...	...

from which we may conclude that fractional heights just require to insert fractional values for *h*. I call this approach "*exponential polynomial interpolation*" since I've not seen this method before (I'd be grateful for hints...).

The values in the table above look somehow mysterious. But recall the symbolic eigen-decomposition as given in [Helms,08-1][1]: that exposition just defines polynomials in u^h at each power of *x* as well (which agree perfectly with these numerical values, see chapter below for a comparision).

Since the process of the computation of the table above is of special interest to me, I'll give it in matrix-notation in the following in contrast to the common polynomial interpolation (in matrix-notation), which I want to recall first.

---

[1] see also Appendix,1

1.3. Review: the ordinary polynomial interpolation

Please recall the polynomial-interpolation technique. For the example I use the same function *U*, simply with base *t=exp(1), u=1*.

What we are doing is to build the list of the coefficients of the occuring powerseries for $U^{\circ h}(x)$ at integer heights *h* and interpolate in each row for fractional *h*.

Here is the list of coefficients which are required for the powerseries for integer heights *h=0,1,2,3,4,...* of $U^{\circ h}(x)$. If they show polynomial growth with *h* we can then apply a common polynomial interpolation technique:

LIST: matrix of coefficients of powerseries for $U^{\circ h}(x)$ *for consecutive "heights" h,t=exp(1),u=1*

	h=0	h=1	h=2	h=3	h=4	h=5	h=6	h=7	...
	0	0	0	0	0	0	0	0	...
x*	1	1	1	1	1	1	1	1	...
x^2*	0	1/2	1	3/2	2	5/2	3	7/2	...
x^3*	0	1/6	5/6	2	11/3	35/6	17/2	35/3	...
x^4*	0	1/24	5/8	5/2	77/12	105/8	187/8	455/12	...
x^5*	0	1/120	13/30	179/60	163/15	691/24	1889/30	1211/10	...
x^6*	0	1/720	203/720	2471/720	287/16	4459/72	30049/180	137389/360	...
x^7*	0	1/5040	877/5040	3217/840	3247/112	132133/1008	45872/105	214139/180	...
...	...	...	...	...	...	...	...	...	...

To get the coefficients for fractional heights we would just interpolate along each row with the appropriate polynomial (which is depending on *h*)

Most suggestively the row at *x* gives *1* also for any fractional *h*. But consider an overlay with a sinusoidal function which is just zero at each integer: this would also be a possible idea for interpolation! So this idea is in no way an unique solution and if in general context, that choice of polynomial interpolation should be explicitely stated.

In row at x^2 it is similarly suggestive that we get *h/2* for the coefficient. So we may say that the powerseries for $U^{\circ h}(x)$ begins with

$$U^{\circ h}(x) = 1\,x + h/2\,x^2 + \ldots$$

and for the coefficients at next powers of *x* we have to apply the common polynomial interpolation techniques using the entries of ***LIST***. We may, for instance, simply use Pari/GP and ask (and get)

```
gp > polinterpolate([1/6, 5/6, 2, 11/3, 35/6 ],,'h)
 %1466 = 1/4*h^2 - 1/12*h
```

which remains constant however many terms of row *3* we feed into the interpolation procedure of Pari/GP.

When we collect the first few interpolation polynomials, we get (the powers of *h* reordered)

```
%1473 = 1                                                   \\ for row 1
%1474 =    + 1/2   *h                                       \\ for row 2
%1475 =    - 1/12  *h  +   1/4 *h^2                         \\ for row 3
%1476 =    + 1/48  *h  -   5/48*h^2  + 1/8   *h^3
%1477 =    - 1/180 *h  +   1/24*h^2 - 13/144 *h^3 + 1/16   *h^4
%1478 =   + 11/8640*h - 91/5760*h^2 + 89/1728*h^3 - 77/1152*h^4 + 1/32*h^5
```

and the matrix of the coefficients only is the table

	1	*h	$*h^2$	$*h^3$	$*h^4$	$*h^5$	...
1*	0						
x*	1						
x^2*	0	+1/2					
x^3*	0	-1/12	+1/4				
x^4*	0	+1/48	-5/48	+1/8			
x^5*	0	-1/180	+1/24	-13/144	+1/16		
x^6*	0	+11/8640	-91/5760	+89/1728	-77/1152	+1/32	
...	...	...	...	...	...	...	...

Let's call this matrix ***POLY*** .

To find the coefficients of the powerseries for $U^{\circ h}(x)$ for any value of h we have to insert the consecutive powers of h to get the final powerseries. In matrix notation using ***V****(h)* for the columnvector of consecutive powers of h this is

$$POLY * V(h) = U_h$$

where the resulting $\boldsymbol{U_h}$ is a column-vector containing the required coefficients for the powerseries of the h'th iterate of $\exp(x)-1$; the dot-product of the vandermondevector ***V****(x)~* with $\boldsymbol{U_h}$ gives the value for $U^{\circ h}(x)$ at x

$$\boldsymbol{V}(x)\sim * \boldsymbol{U_h} = U^{\circ h}(x) = 0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots$$

where the a_k are the consecutive coefficients contained in $\boldsymbol{U_h}$.

$$\begin{aligned} a_1 &= 1 \\ a_2 &= \quad 1/2\, h \\ a_3 &= -1/12\, h + 1/4\, h^2 \\ \dots &= \quad \dots \end{aligned}$$

The matrix-notation of the latter (where we keep h as indeterminate) is then

$$U^{\circ h}(x) \quad = \boldsymbol{V}(x)\sim * \boldsymbol{POLY} * \boldsymbol{V}(h)$$

1.4. The matrix-approach to polynomial interpolation

In the previous we used the fixed *polinterpolate()* procedure in Pari/GP. The unknown internal computation of the polynomial interpolation can equivalently and fully be expressed in matrix terms which I describe here to prepare the understanding of the next chapter.

The matrix-equation for one column in ***LIST*** – assuming an unknown triangular matrix ***POLY*** – which gives coefficients of powerseries for a function $f(x)$ such that $f^{\circ h}()$ at height $h=0,1,2,3,\dots$ occur as polynomials in h:

$$LIST[,h] = POLY * V(h)$$

To get the full list for any integer h we write

$$\begin{aligned} LIST &= POLY * [V(0),V(1),V(2),V(3),\dots] \\ LIST &= POLY * \qquad VZ \qquad // \text{collecting V()'s into a matrix} \end{aligned}$$

The polynomial approach – in matrix notation – is now to find ***POLY*** given ***LIST*** by inversion of ***VZ*** according to the structural hypothesis: ***LIST** = **POLY***VZ* hence we would write

$$LIST * VZ^{-1} \qquad = POLY$$

But this is impossible, since assuming infinite size the vandermonde-matrix ***VZ*** cannot be inverted (there would occur infinite series, summing to $\zeta(1)$).

One possibility to work around this is first to factor ***VZ*** into the well known components:

$$VZ = S2 * {}^{d}F * P\sim$$

This decomposition of ***VZ*** is simply the matrix-notation of the implicit definition of Stirlingnumbers 2^{nd} kind, as given for instance in [A&S].

So we have a description of LIST in terms of the still unknown ***POLY***:

$$LIST \qquad = POLY * (S2 * {}^{d}F * P\sim)$$

But now the newly introduced and separated components are all invertible triangular (or even diagonal) matrices and we may first use the reciprocal of ***P~*** only, to arrive at

$$LIST * P^{-1}\sim \qquad = POLY * S2 * {}^{d}F$$

Then – if ***LIST*** contains polynomially interpolatable entries – the matrix

$$LIST * P^{-1}\sim \ = X$$

is lower triangular, This is indeed the case – and this solves the problem.

From

$$X \qquad = POLY * S2 * {}^{d}F \qquad // \text{ X, S2 is lower triangular, F diagonal}$$

we can proceed by rearranging the other invertible factors:

$$X * {}^{d}F^{-1} * S2^{-1} \qquad = POLY$$
$$X * {}^{d}F^{-1} * S1 \qquad = POLY \qquad // \text{ since } S1 = S2^{-1}$$

where all entries in ***POLY*** are finitely computable by the lhs-matrix-formula.

Then – as in the chapter before – *POLY*V(h)* gives the coefficients for the powerseries for any fractional (or complex) *h* and *V(x)~* POLY * V(h)* gives the value for integer and fractional iterates $f^{\circ h}(x)$ at abscissa *x*.

Using the ***LIST***-matrix generated by $U^{\circ h}(x)$ at consecutive integer heights we get the coefficients for $U^{\circ h}(x)$ for any value of *h* and this version of ***POLY*** is exactly the version which we got by the fixed procedure *polinterpolate()* in Pari/GP.

Note that this matrix-notation is not completely new; for instance I found this in [Comtet, 1970], pg 144-148 in a very much similar expression. However, I didn't see the process consequently expressed in matrix-formulae like I do it here.

The expression *V(x)~* POLY * V(h)* can then be evaluated keeping one of the variables constant.

If *h* is kept constant, we get the powerseries in *x* for a certain height *h* of iteration of *exp(x)-1*.

If *x* is kept constant, we get a powerseries in *h* and which expresses – if for instance *x=1* – the tetrational function *dxp(h)* .

1.5. An "exponential-polynomial" approach to interpolation

In the previous chapter I used the parameters $t=exp(1)$, $u=1$ to get a polynomially interpolatable list depending on the height-parameter h. In the general case, where $t<>exp(1)$, $u<>1$, [1] the occuring matrix ***LIST*** is not polynomially interpolatable depending on h. The new idea is to replace h with the h^{th} powers u^h .

We try the analoguous equation with the again unknown matrix ***POLY***

$$
\begin{aligned}
LIST &= POLY * [V(u^0),V(u^1),V(u^2),V(u^3),...]\\
LIST &= POLY * \quad VV_U \qquad \text{// collecting V()'s into a matrix}
\end{aligned}
$$

Here $\boldsymbol{VV_U}$ is an infinite rectangular (and, btw, symmetric) matrix and – as well as ***VZ*** in the previous – cannot be inverted. The definition of its entries is:

$$VV_U \quad := matrix_{r=0..inf,c=0..inf}[u^{r*c}]$$

But the matrix can be LU-factored into two triangular matrices

$$VV_U \quad = L * R$$

where also ***R*** = ***L***~ since $\boldsymbol{VV_U}$ is symmetric.

We may then norm the columns of ***L*** and rows of ***R*** to get ones on their diagonals and collect the scaling factors in the then required diagonalmatrix ***D***:

$$VV_U \quad = M * D * M\sim$$

where ***M*** is lower triangular with unit in its diagonal and ***D*** is diagonal.

Here is the top-left segment of ***M***:

1	0	0	0	0	0
1	1.0000000	0	0	0	0
1	1.6931472	1.0000000	0	0	0
1	2.1736002	2.1736002	1.0000000	0	0
1	2.5066248	3.2179130	2.5066248	1.0000000	0
1	2.7374599	4.0526808	4.0526808	2.7374599	1.0000000
1	2.8974626	4.6845827	5.4023234	4.6845827	2.8974626
1	3.0083681	5.1481845	6.4836895	6.4836895	5.1481845
1	3.0852418	5.4818288	7.3074130	7.9803527	7.3074130
1	3.1385267	5.7190030	7.9153775	9.1495585	9.1495585
1	3.1754609	5.8862389	8.3550188	10.027417	10.613512

Proposal: (Heuristically:) this matrix is the q-analogue to the pascal-matrix.

With the common definitions for q-analogues (to a base u) and natural n (where I omit the usual brackets [] and also supplied the definition for $n=0$, which is required for the q-binomial)

$$q\text{-}n(u,n) = \qquad n_u = \begin{cases} n & if\ u=1 \\ \dfrac{1-u^n}{1-u} & if\ u\neq 1 \end{cases}$$

$$q\text{-}factorial\ (u,n) = \qquad n!_u = \begin{cases} 1 & if\ n=0 \\ 1_u * 2_u * ... * n_u & if\ n>0 \end{cases}$$

$$q\text{-}binomial(u,r,c) = \qquad \binom{r}{c}_u = \frac{r!_u}{c!_u\,(r-c)!_u}$$

---

[1] or: "hyperbolic case" of iteration according to current nomenclature in the "tetration-forum", see [TF08]

Then ***M*** to a base *u* (in the exampe $u=log(2)$) is defined by

$$M_u := \quad m_{r,c=0..\infty} = \binom{r}{c}_u \qquad \text{where "r" and "c" are the row- and column-indices}$$

and is thus the *q*-Pascalmatrix. The diagonalmatrix ***D*** also contains *q*-analogues; it is

$$D_u := \quad diag(d_{r=0..inf}) = r!_u *(u-1)^r *u^{r(r-1)/2}$$

The matrix ***M*** has the interesting aspect that the values in the columns converge to a fixed value (which can be seen if more rows are displayed). If we compute ***M*D*** the columns seem to converge to powers of $log(u)$ according to the following sequence:

$$(u^0, u^0 , u^1 , u^3 ,.u^6 , \dots , u^{binomial(c,2)}, \dots)$$

where *c* is the column-index beginning at zero. But leave this aside here.

The more interesting aspect is that in

$$LIST * M^{\sim -1} = X$$

the resulting matrix ***X*** is triangular, just as in the analoguous case of the polynomial interpolation in the previous chapter, and the solution can thus be exactly computed

$$\begin{aligned} POLY * (M * D * M^{\sim}) &= LIST \\ POLY * M * D &= LIST * M^{-1} \sim && \text{// the rhs is lower triangular!} \\ &= X \\ POLY &= X * D^{-1} * M^{-1} \end{aligned}$$

where – since ***X*** is triangular – we can compute each term of the result by finite calculation and the resulting ***POLY*** is hence triangular.

POLY: coefficient-matrix for polynomials by "exponential polynomial interpolation" t=2, u=log(2)

	$*u^0$	$*u^{1h}$	$*u^{2h}$	$*u^{3h}$	$*u^{4h}$	$*u^{5h}$	
1*	.	.	.	.	.	.	
x*	.	1	.	.	.	.	
x^2*	.	1.1294457	-1.1294457	.	.	.	
x^3*	.	1.1985847	-2.5512951	1.3527103	.	.	
x^4*	.	1.2474591	-4.1482473	4.5834386	-1.6826504	.	
x^5*	.	1.2856301	-5.8758179	10.040759	-7.6018490	2.1512781	
...		...	...	...	...	...	...

1.6. Comparision of "exponential polynomial interpolation" with diagonalization

The result seems to be the diagonalization in disguise: we get the same coefficients for the powerseries of, for instance, $U_t^{\circ 0.5}(x)$ as we would get by the diagonalization method.

Recall the diagonalization-formula for the matrix-operator $\boldsymbol{U_t}$

$$U_t^h = W_t^{-1} * D_t^h * W_t \qquad // \; D_t \text{ diagonal}$$

and the formal equivalents:

$$\begin{aligned} U_t^{\circ h}(x) &= V(x)\sim * \, U_t^h\,[,1] \\ &= V(x)\sim * \, W_t^{-1} * D_t^h * W_t[,1] \\ &= V(x)\sim * \, W_t^{-1} *^d V(u^h) * W_t[,1] \\ &= V(x)\sim * \, W_t^{-1} * diag(W_t[,1]) * V(u^h) \end{aligned}$$

(where *[,1]* means the second column) then let

$$Q = W_t^{-1} * diag(W_t[,1])$$

and

$$U_t^{\circ h}(x) = V(x)\sim * \, Q * V(u^h)$$

which is of the same form as ***POLY*** in the previous, only that ***Q*** is computed by the diagonalization formula.

Q (=?= POLY) : coefficient-matrix for polynomials by diagonalization-method

	$*u^0$	$*u^{1h}$	$*u^{2h}$	$*u^{3h}$	$*u^{4h}$	$*u^{5h}$	
1	0	.	.	.	.	.	
$x*$	.	1	.	.	.	.	
x^2*	.	1.1294457	-1.1294457	.	.	.	
x^3*	.	1.1985847	-2.5512951	1.3527103	.	.	
x^4*	.	1.2474591	-4.1482473	4.5834386	-1.6826504	.	
x^5*	.	1.2856301	-5.8758179	10.040759	-7.6018490	2.1512781	
...	.	...	...	...	...	...	...

This is - numerically to arbitrary precision - the same as what we got by the "exponential-polynomial interpolation" method for the matrix ***POLY***.

1.7. Conclusion

Although I was originally fiddling with alternate interpolation approaches[1] the given method occured "as an exercise" and seems to provide another legitimation for the diagonalization method.

Another spin-off of this analysis may be a better estimation procedure for the growth of the coefficients of the fractional-*h*-powerseries for *U*-tetration due to the convergence observation with the ***M***-matrix. This would subsequently allow to improve the summation techniques for such divergent series in general.

---

[1] I speculated that this is needed to make the *T*-tetration (using fixpoint-shifts via *U*-tetration) compatible for all different fixpoints

2. Appendix

[Pg 21 of "*Continuous iteration of powerseries-defined functions*" in [Helms,08-1] polynomials in u^h for general $U_t^{\circ h}(x)$-powerseries, found by diagonalization]

Let $U_t^{\circ h}(x)$ denote the h'th iterate of $U_t(x)$, then its powerseries is:

$$U_t^{\circ h}(x) = a_1 \frac{x}{1!} + a_2 \frac{u}{u-1}\frac{x^2}{2!} + a_3 \frac{u^2}{(u-1)(u^2-1)}\frac{x^3}{3!} + \ldots + a_k \frac{u^{k-1}}{\prod_{j=1}^{k-1}(u^j-1)}\frac{x^k}{k!} + \ldots$$

where

$$a_1 = 1\,u^h$$

$$\begin{aligned} a_2 = & -(1)\,u^h \\ & +(1)\,u^{2h} \end{aligned}$$

$$\begin{aligned} a_3 = & \;(1+2u)\,u^h \\ & -(3+3u)\,u^{2h} \\ & +(2+1u)\,u^{3h} \end{aligned}$$

$$\begin{aligned} a_4 = & -(1+6u+5u^2+6u^3)\,u^h \\ & +(7+18u+18u^2+11u^3)\,u^{2h} \\ & -(12+18u+18u^2+6u^3)\,u^{3h} \\ & +(6+6u+5u^2+1u^3)\,u^{4h} \end{aligned}$$

$$\begin{aligned} a_5 = & \;(1+14u+24u^2+45u^3+46u^4+26u^5+24u^6)\,u^h \\ & -(15+75u+130u^2+180u^3+165u^4+105u^5+50u^6)\,u^{2h} \\ & +(50+145u+230u^2+275u^3+215u^4+130u^5+35u^6)\,u^{3h} \\ & -(60+120u+170u^2+180u^3+120u^4+60u^5+10u^6)\,u^{4h} \\ & +(24+36u+46u^2+40u^3+24u^4+9u^5+1u^6)\,u^{5h} \end{aligned}$$

Note 1:

For integer h the denominators at each power of x are also factors of the numerators and can be cancelled. So we can use this formula even for the case $u=1$, $t=\exp(1)$. For $h=2$, $u=1$ we get

$$\begin{aligned} \text{at } x: \quad & u^2/1! \\ & = 1 \\ \text{at } x^2: \quad & (u^4-u^2)u/(u-1)/2! \\ & = u^2(u+1)u/2! \\ & = 2/2! \\ & = 1 \\ \text{at } x^3: \quad & ((1+2u)u^2 - (3+3u)u^4 + (2+1u)u^6)u^2/(u^3-u^2-u+1)/3! \\ & = u^2(1+2u-3u^2-3u^3+2u^4+1u^5)u^2/(u^3-u^2-u+1)/3! \\ & = (1u^2+3u+1)*u^4/3! \\ & = 5/3! \end{aligned}$$

Thus the powerseries begins with

$$U^{\circ 2}(x) = 1\,x + 1\,x^2 + 5/6\,x^3 + \ldots$$

which complies with the conventional computation.

Note 2:

Using the fixpoint-shift one may determine the coefficients for **T**-tetration for base

$b = t^{1/t} = \exp(u/t)$

and

$T_b(x) = b^x \qquad T_b^{\circ h}(x) = T_b^{\circ h-1}(b^x)$

and

$T_b^{\circ h}(x) = (U_t^{\circ h}(x/t - 1) + 1)*t$

by inserting:

$$\frac{T_b^{\circ h}(x)}{t} = 1 + a_1 \frac{\left(\frac{x}{t} - 1\right)}{1!} + a_2 \frac{u}{u-1} \frac{\left(\frac{x}{t} - 1\right)^2}{2!} + a_3 \frac{u^2}{(u-1)(u^2-1)} \frac{\left(\frac{x}{t} - 1\right)^3}{3!} + \ldots$$

expanding the parentheses $(x/t - 1)^k$ and collecting like powers. I'll present the example in the next version of this text.

3. References

[TF08-1]	<Tetration-forum>, Notations and definitions http://math.eretrandre.org/tetrationforum/showthread.php?tid=184
[Helms,08-1]	Helms, Gottfried; Continuous functional iteration pg 21, online at http://go.helms-net.de/math/tetdocs/ContinuousfunctionalIteration.pdf
[A&S]	Abramowitsch, M. and Stegun, I.A.; Handbook of mathematical functions 9th printing page 824; online at http://www.math.sfu.ca/~cbm/aands/
[Comtet,1970]	Comtet, Louis; Advanced Combinatorics D. Reidel Publishing Company; Dordrecht, Holland; 1970. pages 144-148

Gottfried Helms, first version: 13.07.2008