Fine-tuning the Laporta approach

Thomas Luthe

Bielefeld University
in collaboration with York Schröder

Radcor-Loopfest 2015
Motivation

- Previous talk: many applications for tadpole integrals
- Push computational limits to 5 loops, starting with fully massive tadpoles
- Low maintenance approach, one method for a complete set of integrals with little human input needed

Outline

- Short review of difference equation and factorial series
- Improvements
- Results
Difference equations and factorial series \cite{Laporta '01}

- \( l(x) = \int \frac{1}{D_1^x D_2^{b_2} \cdots D_n^{b_n}} \), here all \( D_i \) massive with \( m = 1 \)

- \( \sum_{k=0}^{R} q_k(x) l(x + k) = \sum_{i}^{R_i - 1} \sum_{k=0}^{R_i - 1} p_{ik}(x) J_i(x + k), \ J_i \in \text{subsectors} \)

- \( l(x) = \sum_{s=0}^{\infty} \frac{\Gamma(x + 1)}{\Gamma(x + s + d/2 + 1)} a_s \)

- \( \sum_{k=0}^{R'} q'_k(s) a_{s+k} = \sum_{i}^{R'_i - 1} \sum_{k=0}^{R'_i - 1} p'_{ik}(s) a_{i,s+k} \)

- \( a_0 \) from large-\( x \) behaviour in terms of lower loop integrals

- \( a_0 \xrightarrow{\text{rec. rel.}} a_{s_{\text{max}}}, \sum_{s=0}^{s_{\text{max}}} \frac{\Gamma(x_{\text{max}} + 1)}{\Gamma(x_{\text{max}} + s + d/2 + 1)} a_s, \ l(x_{\text{max}}) \xrightarrow{\text{diff. eq.}} l(1) \)
Difference equations and factorial series [Laporta ‘01]

Advantages

- Everything can be automated
- Works well also with divergent integrals and does not depend on a special class of functions
- High precision results for arbitrarily many orders in $\epsilon$
- Can expand around any dimension
- Cross-checks by putting $x$ on different propagators

Typical problems and limitations

- Usually only numeric results → limited use for integrals with multiple scales
- Complexity of the coefficients in high order equations
- High orders of the recurrence relation
- Divergence of the factorial series in numerical evaluation
Coupled vs decoupled equations

Typically generate equations via IBP: \(0 = l_j(x + k)\) 

[Chetyrkin, Tkachov '81]

Decoupled equations

- simple solve algorithm
- need to solve only one integral numerically
- coeffs. grow large very quickly with \(R\)

Coupled equations

- more involved solve algorithm
- need to solve \(R\) integrals simultaneously
- coeffs. grow less quickly with \(R\)
- can choose master integral basis
Coupled vs decoupled equations

Example: Difference eq. 29703#3
Propagators: 7
Order: 8
Integrals: 1396 + sub-topologies
Input equations: 1400

<table>
<thead>
<tr>
<th></th>
<th>decoupled equations</th>
<th>coupled equations</th>
<th>coupled eqs. + opt. basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle \text{deg}_x \rangle / \text{coeff.}$</td>
<td>33.54</td>
<td>5.63</td>
<td>4.65</td>
</tr>
<tr>
<td>$\langle \text{deg}_d \rangle / \text{coeff.}$</td>
<td>31.59</td>
<td>6.51</td>
<td>4.51</td>
</tr>
<tr>
<td>$\langle \text{size as string} \rangle / \text{eq.}$</td>
<td>$5.3 \cdot 10^5$</td>
<td>$7.0 \cdot 10^3$</td>
<td>$3.9 \cdot 10^3$</td>
</tr>
<tr>
<td>$\langle # \text{ coefficients} \rangle / \text{eq.}$</td>
<td>17.97</td>
<td>17.44</td>
<td>16.06</td>
</tr>
<tr>
<td>$# \text{ steps to solve}$</td>
<td>$6.1 \cdot 10^4$</td>
<td>$5.9 \cdot 10^4$</td>
<td>$5.6 \cdot 10^4$</td>
</tr>
<tr>
<td>time to solve</td>
<td>$\sim 2d$</td>
<td>68s</td>
<td>48s</td>
</tr>
</tbody>
</table>
Coupled vs decoupled equations

Scaling of the homogeneous part of equations with order $R$:

![Graph showing scaling of coupled and decoupled equations with order $R$. The graph plots the average number of equations against $R$. There are two sets of data points: red circles for equation sizes and blue triangles for degree in $x$. As $R$ increases, both the number of equations and the degree in $x$ grow significantly.]
Recurrence relations

\[ \sum_{i}^{R'} \sum_{k=0}^{s} q_{ik}(s) a_{i,s+k} = 0 \]

- Can be reduced with the same algorithms as the difference equations!
- Translation from difference equations:
  - order \( R \), \( x \)-degree \( N \) \( \rightarrow \) order \( R' \) with \( N \leq R' \leq N + R \)
  - \( \Rightarrow \) Translation loses information with every order of \( x \)
- Need input equations with \( R \& N \) minimal.
- IBP: \( R \leq 2, N \leq 1 \), but not good enough.
  - \( \Rightarrow \) Reduce IBP-equations without multiplying or dividing by \( x \) & try to factor out \((x + \alpha)\).
Divergence factors and precision

- The numerical error grows by a factor $F_D$ ($F_R$) with each iteration of the difference equations (recurrence relations).

- $D_{\text{end}} \approx \log_{10} \left[ \left( \frac{x_{\text{max}} + s_{\text{max}}}{s_{\text{max}}} \right) F_D \right]^{x_{\text{max}}} \quad \text{(precision of } I(1))$

- $D_{\text{start}} \approx \log_{10} \left[ \left( \frac{x_{\text{max}} + s_{\text{max}}}{s_{\text{max}}} \right) F_R \right]^{s_{\text{max}}} \quad \text{(precision of } a_0)$

<table>
<thead>
<tr>
<th>Loops</th>
<th>$F_D$</th>
<th>$F_R$</th>
<th>$x_{\text{max}}$</th>
<th>$s_{\text{max}}$</th>
<th>$D_{\text{end}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>300000</td>
<td>1000000</td>
<td>300000</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>300000</td>
<td>870000</td>
<td>145000</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>1</td>
<td>110000</td>
<td>900000</td>
<td>45000</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>1.125</td>
<td>21500</td>
<td>1000000</td>
<td>20000</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>12.928</td>
<td>700</td>
<td>18000</td>
<td>300</td>
</tr>
</tbody>
</table>
Results \((d = 4)\)

\[
\frac{\mathcal{E}}{5} = -\frac{441}{40} \zeta(7) \epsilon^4 \quad [\text{Kazakov '83}]
\]

\[
+ 181.782239286123408207907882360186 \ldots \epsilon^5
\]

\[
- 1725.99961374035208059516739924421 \ldots \epsilon^6
\]

\[
+ 12797.9998737268240466855516903376 \ldots \epsilon^7
\]

\[
- 82986.8526925813821605590471473909 \ldots \epsilon^8
\]

\[
+ 496710.272856148328215231508903586 \ldots \epsilon^9
\]

\[+ \ldots\]

See also zig-zag conjecture [Broadhurst, Kreimer '95][Brown, Schnetz '12]
Conclusions

- **Improvements:**
  - Choose coupled over decoupled eqs. to simplify coefficients
  - Reduction of recurrence relations
  - Avoid divergence in factorial series by increased precision

- Everything implemented in C++, except polynomial algebra (Fermat [Lewis]), all time-critical code parallelised

- At the 5-loop level have produced difference equations up to order 20, recurrence relations up to order $28 +$ inhomogeneous parts

- Solved all fully massive master integrals for 37 out of 48 vacuum 5-loop diagrams with $\sim 300$ digits precision, $\geq 10$ orders in $\epsilon$ around $d = 4 - 2\epsilon, d = 3 - 2\epsilon$
Divergence factors

\[ I^{(\text{hom})}(x) = \sum_{m=1}^{n} \mu_m^x \sum_{s=0}^{\infty} \frac{\Gamma(x + 1)}{\Gamma(x + s - K_m + 1)} a_{m,s} \]

\( \mu_m \) are roots of the characteristic polynomial \( p(\mu) \).

For decoupled eq. \( \sum_{i=0}^{N} \sum_{k=0}^{R} p_{ik} x^i I(x + k) = 0 \):

\[ p(\mu) = \sum_{k=0}^{R} p_{Nk} \mu^k \]

\[ F_{P}^{(m)} = \max_{i} \left| \frac{\mu_m}{\mu_i} \right| \]

\[ F_{R}^{(m)} = \max \left\{ 1, \max_{i, \mu_i \neq \mu_m} \left| \frac{\mu_m}{\mu_m - \mu_i} \right| \right\} \]