Quadpack Computation of IR-divergent integrals

E. de Doncker[1] J. Fujimoto[2] N. Hamaguchi [3]
T. Ishikawa[2] Y. Kurihara[2] Y. Shimizu[2] F. Yuasa[2]

[1]Department of Computer Science, Western Michigan University, Kalamazoo MI 49008, U. S.

[2]High Energy Accelerator Research Organization (KEK), Oho 1-1, Tsukuba, Ibaraki, 305-0801, Japan

[3]Hitachi, Ltd., Japan

Outline

1. Introduction
   - Infrared singularity
   - Digression on extrapolation
2. Application to massless one-loop vertex
   - Asymptotics one off-shell, two on-shell particles
3. Hypergeometric function and threshold singularity
4. Vertex with one on-shell and two off-shell particles

fu-logo

ur-logo

Outline

Outline

1. Introduction
   - Infrared singularity
   - Digression on extrapolation
2. Application to massless one-loop vertex
   - Asymptotics one off-shell, two on-shell particles
3. Hypergeometric function and threshold singularity
4. Vertex with one on-shell and two off-shell particles

Outline

Outline

Sample problem: Characteristics

Integral

$$I(\varepsilon) = \int_0^1 dx \int_0^1 dy \, \frac{1}{(x+y)^{2-\varepsilon}} = \frac{2^\varepsilon - 2}{(\varepsilon - 1)\,\varepsilon} = 2\frac{\varphi(\varepsilon)}{\varepsilon} \tag{1}$$

where $\varphi(\varepsilon) = (2^{\varepsilon-1} - 1)/(\varepsilon - 1)$

converges for $\varepsilon > 0$ and has a non-integrable singularity when $\varepsilon \leq 0$. Taylor expansion of $\varphi(\varepsilon)$ around $\varepsilon = 0$ results in

$$I(\varepsilon) \sim \frac{C_{-1}}{\varepsilon} + C_0 + C_1\varepsilon + \ldots . \tag{2}$$

with $C_{-1} = 1, \;\; C_0 = 1 - \log 2, \;\; C_1 = 1 - \log 2 - \frac{1}{2}\log^2 2.$

According to the asymptotic behavior, we can explore linear or nonlinear extrapolation for the computation of the coefficients of the leading terms.

Sample problem: Characteristics

Integral

$$I(\varepsilon) = \int_0^1 dx \int_0^1 dy \, \frac{1}{(x+y)^{2-\varepsilon}} = \frac{2^\varepsilon - 2}{(\varepsilon - 1)\,\varepsilon} = 2\,\frac{\varphi(\varepsilon)}{\varepsilon} \tag{1}$$

where $\varphi(\varepsilon) = (2^{\varepsilon-1} - 1)/(\varepsilon - 1)$

converges for $\varepsilon > 0$ and has a non-integrable singularity when $\varepsilon \leq 0$. Taylor expansion of $\varphi(\varepsilon)$ around $\varepsilon = 0$ results in

$$I(\varepsilon) \sim \frac{C_{-1}}{\varepsilon} + C_0 + C_1\varepsilon + \ldots. \tag{2}$$

with $C_{-1} = 1,\ C_0 = 1 - \log 2,\ C_1 = 1 - \log 2 - \frac{1}{2}\log^2 2$.

According to the asymptotic behavior, we can explore linear or nonlinear extrapolation for the computation of the coefficients of the leading terms.

Sample problem: Methods

Linear system

$$\varepsilon_\ell I(\varepsilon_\ell) = \sum_{k=0}^{n-1} C_{k-1}\,\varepsilon_\ell{}^k, \quad 1 \le \ell \le n \tag{3}$$

special case of $S_\ell = \sum_{k=0}^{n-1} C_{k-1}\,\varphi_k(\varepsilon_\ell), \quad 1 \le \ell \le n$

Numerical integration

$\hat{I}(\varepsilon) \approx I(\varepsilon)$ for $\varepsilon > 0$ is affected by the integrand singularity of (1) at $x = y = 0$ on the boundary of the integration domain, when $2 > 2 - \varepsilon > 0$.

Recursive (repeated) integration with the 1D integration code DQAGSE from QUADPACK [8] is equipped to handle algebraic end-point singularities in each coordinate direction.

tu-logo

ur-logo

Sample problem: Methods

Linear system

$$\varepsilon_\ell I(\varepsilon_\ell) = \sum_{k=0}^{n-1} C_{k-1}\, {\varepsilon_\ell}^k, \quad 1 \le \ell \le n \tag{3}$$

special case of $S_\ell = \sum_{k=0}^{n-1} C_{k-1}\, \varphi_k(\varepsilon_\ell), \quad 1 \le \ell \le n$

Numerical integration

$\hat{I}(\varepsilon) \approx I(\varepsilon)$ for $\varepsilon > 0$ is affected by the integrand singularity of (1) at $x = y = 0$ on the boundary of the integration domain, when $2 > 2 - \varepsilon > 0$.

Recursive (repeated) integration with the 1D integration code DQAGSE from QUADPACK [8] is equipped to handle algebraic end-point singularities in each coordinate direction.

tu-logo

ur-logo

Sample problem: Methods

Linear system

$$\varepsilon_\ell I(\varepsilon_\ell) = \sum_{k=0}^{n-1} C_{k-1}\, {\varepsilon_\ell}^k, \quad 1 \le \ell \le n \tag{3}$$

special case of $S_\ell = \sum_{k=0}^{n-1} C_{k-1}\, \varphi_k(\varepsilon_\ell), \quad 1 \le \ell \le n$

Numerical integration

$\hat{I}(\varepsilon) \approx I(\varepsilon)$ for $\varepsilon > 0$ is affected by the integrand singularity of (1) at $x = y = 0$ on the boundary of the integration domain, when $2 > 2 - \varepsilon > 0$.

Recursive (repeated) integration with the 1D integration code DQAGSE from QUADPACK [8] is equipped to handle algebraic end-point singularities in each coordinate direction.

Sample problem: Methods

Linear system

$$\varepsilon_\ell I(\varepsilon_\ell) = \sum_{k=0}^{n-1} C_{k-1}\, {\varepsilon_\ell}^k, \quad 1 \le \ell \le n \tag{3}$$

special case of $S_\ell = \sum_{k=0}^{n-1} C_{k-1}\, \varphi_k(\varepsilon_\ell), \quad 1 \le \ell \le n$

Numerical integration

$\hat{I}(\varepsilon) \approx I(\varepsilon)$ for $\varepsilon > 0$ is affected by the integrand singularity of (1) at $x = y = 0$ on the boundary of the integration domain, when $2 > 2 - \varepsilon > 0$.

Recursive (repeated) integration with the 1D integration code DQAGSE from QUADPACK [8] is equipped to handle algebraic end-point singularities in each coordinate direction.

Outline

Digression on extrapolation

- For sequence $\{S(\varepsilon_\ell)\}$, an extrapolation is performed to create sequences that convergence to the limit $\mathcal{S} = \lim_{\varepsilon_\ell \to 0} S(\varepsilon_\ell)$ faster than the given sequence, based on

Asymptotic expansion

$$S(\varepsilon) \sim \mathcal{S} + a_0\varphi_0(\varepsilon) + a_2\varphi_2(\varepsilon) + \dots$$

- ε may be a parameter of the problem or of the method. For example in Romberg integration, $\varepsilon_\ell = h_\ell$ is the step size.
- A linear extrapolation method solves (implicitly or explicitly [2]) linear systems of the form

$$S(\varepsilon_\ell) = a_0 + a_1\varphi_1(\varepsilon_\ell) + \dots a_\nu\varphi_\nu(\varepsilon_\ell), \quad \ell = 0, \dots, \nu;$$

of order $(\nu + 1) \times (\nu + 1)$ in unknowns $a_0, \dots a_\nu$ are solved for increasing values of ν.

Digression on extrapolation

- For sequence $\{S(\varepsilon_\ell)\}$, an extrapolation is performed to create sequences that convergence to the limit $\mathcal{S} = \lim_{\varepsilon_\ell \to 0} S(\varepsilon_\ell)$ faster than the given sequence, based on

Asymptotic expansion

$$S(\varepsilon) \sim \mathcal{S} + a_0\varphi_0(\varepsilon) + a_2\varphi_2(\varepsilon) + \dots$$

- ε may be a parameter of the problem or of the method. For example in Romberg integration, $\varepsilon_\ell = h_\ell$ is the step size.
- A linear extrapolation method solves (implicitly or explicitly [2]) linear systems of the form

$$S(\varepsilon_\ell) = a_0 + a_1\varphi_1(\varepsilon_\ell) + \dots a_\nu\varphi_\nu(\varepsilon_\ell), \quad \ell = 0, \dots, \nu;$$

of order $(\nu + 1) \times (\nu + 1)$ in unknowns $a_0, \dots a_\nu$ are solved for increasing values of ν.

fu-logo

ur-logo

Digression on extrapolation

- For sequence $\{S(\varepsilon_\ell)\}$, an extrapolation is performed to create sequences that convergence to the limit $\mathcal{S} = \lim_{\varepsilon_\ell \to 0} S(\varepsilon_\ell)$ faster than the given sequence, based on

Asymptotic expansion

$$S(\varepsilon) \sim \mathcal{S} + a_0\varphi_0(\varepsilon) + a_2\varphi_2(\varepsilon) + \ldots$$

- ε may be a parameter of the problem or of the method. For example in Romberg integration, $\varepsilon_\ell = h_\ell$ is the step size.
- A linear extrapolation method solves (implicitly or explicitly [2]) linear systems of the form

$$S(\varepsilon_\ell) = a_0 + a_1\varphi_1(\varepsilon_\ell) + \ldots a_\nu\varphi_\nu(\varepsilon_\ell), \quad \ell = 0, \ldots, \nu;$$

  of order $(\nu + 1) \times (\nu + 1)$ in unknowns $a_0, \ldots a_\nu$ are solved for increasing values of ν.

Digression on extrapolation

- The coefficients $\varphi_k(\varepsilon)$ need to be known explicitly in order to compute the system coefficients for linear extrapolation.
- If the functions of ε are not known, nonlinear extrapolation or convergence acceleration may be applied [9, 11, 10, 7, 4].
- As an example of a nonlinear extrapolation method, the ϵ-algorithm implements the sequence-to-sequence transformation by [9] recursively;
- can be applied if the φ functions are of the form

$$\varphi_k(\varepsilon) = \varepsilon^{\beta_k} \log^{\nu_k}(\varepsilon),$$

under some conditions on ν_k and β_k and if a geometric sequence is used for ε;

- but the actual form of the underlying ε-dependency does not need to be specified.

Digression on extrapolation

- The coefficients $\varphi_k(\varepsilon)$ need to be known explicitly in order to compute the system coefficients for linear extrapolation.
- If the functions of ε are not known, nonlinear extrapolation or convergence acceleration may be applied [9, 11, 10, 7, 4].
- As an example of a nonlinear extrapolation method, the ϵ-algorithm implements the sequence-to-sequence transformation by [9] recursively;
- can be applied if the φ functions are of the form

$$\varphi_k(\varepsilon) = \varepsilon^{\beta_k} \log^{\nu_k}(\varepsilon),$$

  under some conditions on ν_k and β_k and if a geometric sequence is used for ε;
- but the actual form of the underlying ε-dependency does not need to be specified.

tu-logo

ur-logo

Digression on extrapolation

- The coefficients $\varphi_k(\varepsilon)$ need to be known explicitly in order to compute the system coefficients for linear extrapolation.
- If the functions of ε are not known, nonlinear extrapolation or convergence acceleration may be applied [9, 11, 10, 7, 4].
- As an example of a nonlinear extrapolation method, the ϵ-algorithm implements the sequence-to-sequence transformation by [9] recursively;
- can be applied if the φ functions are of the form

$$\varphi_k(\varepsilon) = \varepsilon^{\beta_k} \log^{\nu_k}(\varepsilon),$$

under some conditions on ν_k and β_k and if a geometric sequence is used for ε;

- but the actual form of the underlying ε-dependency does not need to be specified.

Digression on extrapolation

- The coefficients $\varphi_k(\varepsilon)$ need to be known explicitly in order to compute the system coefficients for linear extrapolation.
- If the functions of ε are not known, nonlinear extrapolation or convergence acceleration may be applied [9, 11, 10, 7, 4].
- As an example of a nonlinear extrapolation method, the ϵ-algorithm implements the sequence-to-sequence transformation by [9] recursively;
- can be applied if the φ functions are of the form

$$\varphi_k(\varepsilon) = \varepsilon^{\beta_k} \log^{\nu_k}(\varepsilon),$$

under some conditions on ν_k and β_k and if a geometric sequence is used for ε;

- but the actual form of the underlying ε-dependency does not need to be specified.

Digression on extrapolation

- The coefficients $\varphi_k(\varepsilon)$ need to be known explicitly in order to compute the system coefficients for linear extrapolation.
- If the functions of ε are not known, nonlinear extrapolation or convergence acceleration may be applied [9, 11, 10, 7, 4].
- As an example of a nonlinear extrapolation method, the ϵ-algorithm implements the sequence-to-sequence transformation by [9] recursively;
- can be applied if the φ functions are of the form

$$\varphi_k(\varepsilon) = \varepsilon^{\beta_k} \log^{\nu_k}(\varepsilon),$$

under some conditions on ν_k and β_k and if a geometric sequence is used for ε;
- but the actual form of the underlying ε-dependency does not need to be specified.

ϵ-algorithm recursion

Extrapolation table

	τ_{00}			
0		τ_{01}		
	τ_{10}		τ_{02}	
0		τ_{11}		$\ldots$
		$\ldots$		$\ldots$
		$\ldots$		$\ldots$
0		$\tau_{\kappa-1,1}$		$\ldots$
	$\tau_{\kappa 0}$		$\tau_{\kappa-1,2}$	
0		$\tau_{\kappa 1}$		
	$\tau_{\kappa+1,0}$			

With original sequence S_κ, for $\kappa = 0, 1, \ldots$:

$$\tau_{\kappa,-1} = 0$$

$$\tau_{\kappa 0} = S_\kappa$$

$$\tau_{\kappa,\lambda+1} = \tau_{\kappa+1,\lambda+1} + \frac{1}{\tau_{\kappa+1,\lambda} - \tau_{\kappa\lambda}}$$

Sample problem: Linear extrapolation results

ℓ	Numerical Integration $(\text{Dqagse})^2$			Extrapolation relative errors		
	Rel. err.	# Evals	Time (s)	C_{-1}	C_0	C_1
1	1.90e-16	33,165	8.73e-03	1.28e-02	2.02e-01	
2	1.00e-15	36,915	7.43e-03	5.37e-04	1.55e-02	2.00e-01
3	3.29e-15	41,325	8.29e-03	1.21e-06	5.85e-04	1.42e-02
4	1.45e-14	41,775	8.41e-03	1.70e-07	1.27e-05	5.04e-04
5	1.39e-14	41,865	8.39e-03	1.37e-09	1.56e-07	9.72e-06
6	1.69e-14	42,045	8.42e-03	7.30e-12	1.20e-09	1.11e-07
7	2.18e-16	42,165	8.44e-03	1.50e-13	3.88e-11	4.56e-09
8	1.73e-14	67,755	1.35e-02	2.84e-13	7.18e-11	9.14e-09
9	1.01e-14	67,845	1.35e-02	5.06e-14	9.66e-12	7.20e-09
10	1.35e-14	140,655	1.13e-01	5.63e-13	3.19e-10	9.73e-08

Table: Integration and extrapolation results using $(\text{Dqagse})^2$ for IR sample problem (1)-(2), integ. error tolerances $t_r = 10^{-13}$ (outer), 5×10^{-14} (inner); solution based on linear system (3); integration and extrapolation relative errors are given.

Sample problem: Remarks

- The sequence of ε_ℓ is based on a type of sequence used by Bulirsch [3] in the context of Romberg integration: $\varepsilon = \varepsilon_\ell = 1/b_\ell$, $\ell = 1, 2, \ldots$, where $b_\ell = 2, 3, 4, 6, 8, 12, 16, 24, \ldots$ (alternating powers of 2 and 1.5 times the preceding power of 2).
- The Bulirsch sequence can be used with linear extrapolation, not with nonlinear extrapolation by the ϵ-algorithm;
- is in between geometric and harmonic sequence with repect to stability of the extrapolation, and also with respect to the expense in integrations for the system RHS.
- Extrapolation times negligible compared to integration;
- the successive linear systems are independent; only one is needed for solution but successive ones can be used for estimating accuracy.
- Similar results are obtained with geometric progressions of $\varepsilon_\ell = 1/b^\ell$ with, e.g., $b = 1.2, 1.5$ or 2. The amount of work in the integration, as reflected by the number of integrand evaluations, is considerably higher especially with $b = 2$.
- Using a geometric sequence, the ϵ-algorithm [9, 11] is able to transform the divergent sequence of $I(\varepsilon_\ell)$, $\ell = 1, 2, \ldots$ for a non-linear extrapolation to C_0.

Sample problem: Remarks

- The sequence of ε_ℓ is based on a type of sequence used by Bulirsch [3] in the context of Romberg integration: $\varepsilon = \varepsilon_\ell = 1/b_\ell, \ell = 1, 2, \ldots$, where $b_\ell = 2, 3, 4, 6, 8, 12, 16, 24, \ldots$ (alternating powers of 2 and 1.5 times the preceding power of 2).
- The Bulirsch sequence can be used with linear extrapolation, not with nonlinear extrapolation by the ϵ-algorithm;
- is in between geometric and harmonic sequence with repect to stability of the extrapolation, and also with respect to the expense in integrations for the system RHS.
- Extrapolation times negligible compared to integration;
- the successive linear systems are independent; only one is needed for solution but successive ones can be used for estimating accuracy.
- Similar results are obtained with geometric progressions of $\varepsilon_\ell = 1/b^\ell$ with, e.g., $b = 1.2, 1.5$ or 2. The amount of work in the integration, as reflected by the number of integrand evaluations, is considerably higher especially with $b = 2$.
- Using a geometric sequence, the ϵ-algorithm [9, 11] is able to transform the divergent sequence of $I(\varepsilon_\ell)$, $\ell = 1, 2, \ldots$ for a non-linear extrapolation to C_0.

Sample problem: Remarks

- The sequence of ε_ℓ is based on a type of sequence used by Bulirsch [3] in the context of Romberg integration: $\varepsilon = \varepsilon_\ell = 1/b_\ell,\ \ell = 1, 2, \ldots$, where $b_\ell = 2,\ 3,\ 4,\ 6,\ 8,\ 12,\ 16,\ 24, \ldots$ (alternating powers of 2 and 1.5 times the preceding power of 2).
- The Bulirsch sequence can be used with linear extrapolation, not with nonlinear extrapolation by the ϵ-algorithm;
- is in between geometric and harmonic sequence with repect to stability of the extrapolation, and also with respect to the expense in integrations for the system RHS.
- Extrapolation times negligible compared to integration;
- the successive linear systems are independent; only one is needed for solution but successive ones can be used for estimating accuracy.
- Similar results are obtained with geometric progressions of $\varepsilon_\ell = 1/b^\ell$ with, e.g., $b = 1.2, 1.5$ or 2. The amount of work in the integration, as reflected by the number of integrand evaluations, is considerably higher especially with $b = 2$.
- Using a geometric sequence, the ϵ-algorithm [9, 11] is able to transform the divergent sequence of $I(\varepsilon_\ell),\ \ell = 1, 2, \ldots$ for a non-linear extrapolation to C_0.

Sample problem: Remarks

- The sequence of ε_ℓ is based on a type of sequence used by Bulirsch [3] in the context of Romberg integration: $\varepsilon = \varepsilon_\ell = 1/b_\ell$, $\ell = 1, 2, \ldots$, where $b_\ell = 2, 3, 4, 6, 8, 12, 16, 24, \ldots$ (alternating powers of 2 and 1.5 times the preceding power of 2).
- The Bulirsch sequence can be used with linear extrapolation, not with nonlinear extrapolation by the ϵ-algorithm;
- is in between geometric and harmonic sequence with repect to stability of the extrapolation, and also with respect to the expense in integrations for the system RHS.
- Extrapolation times negligible compared to integration;
- the successive linear systems are independent; only one is needed for solution but successive ones can be used for estimating accuracy.
- Similar results are obtained with geometric progressions of $\varepsilon_\ell = 1/b^\ell$ with, e.g., $b = 1.2, 1.5$ or 2. The amount of work in the integration, as reflected by the number of integrand evaluations, is considerably higher especially with $b = 2$.
- Using a geometric sequence, the ϵ-algorithm [9, 11] is able to transform the divergent sequence of $I(\varepsilon_\ell)$, $\ell = 1, 2, \ldots$ for a non-linear extrapolation to C_0.

Sample problem: Remarks

- The sequence of ε_ℓ is based on a type of sequence used by Bulirsch [3] in the context of Romberg integration: $\varepsilon = \varepsilon_\ell = 1/b_\ell$, $\ell = 1, 2, \ldots$, where $b_\ell = 2, 3, 4, 6, 8, 12, 16, 24, \ldots$ (alternating powers of 2 and 1.5 times the preceding power of 2).
- The Bulirsch sequence can be used with linear extrapolation, not with nonlinear extrapolation by the ϵ-algorithm;
- is in between geometric and harmonic sequence with repect to stability of the extrapolation, and also with respect to the expense in integrations for the system RHS.
- Extrapolation times negligible compared to integration;
- the successive linear systems are independent; only one is needed for solution but successive ones can be used for estimating accuracy.
- Similar results are obtained with geometric progressions of $\varepsilon_\ell = 1/b^\ell$ with, e.g., $b = 1.2, 1.5$ or 2. The amount of work in the integration, as reflected by the number of integrand evaluations, is considerably higher especially with $b = 2$.
- Using a geometric sequence, the ϵ-algorithm [9, 11] is able to transform the divergent sequence of $I(\varepsilon_\ell)$, $\ell = 1, 2, \ldots$ for a non-linear extrapolation to C_0.

Outline

One off-shell, two on-shell particles

A massless one-loop vertex non-scalar integral for the case of one off-shell particle (with $p_3^2 \neq 0$) and two on-shell particles ($p_1^2 = p_2^2 = 0$) is given in [5] by

$$J_3\left(0,0,p_3^2;n_x,n_y\right) = \frac{1}{(4\pi)^2} \lim_{\varepsilon\to 0} I_3^{n_x,n_y}(\varepsilon)$$

Integral

$$I_3^{n_x,n_y}(\varepsilon) = \frac{\varepsilon\,\Gamma(-\varepsilon)}{(4\pi\mu_R^2)^{\varepsilon}} \int_0^1 dx \int_0^{1-x} dy \, \frac{x^{n_x} y^{n_y}}{\left(-p_3^2 xy - i0\right)^{1-\varepsilon}} \quad (4)$$

$$= \varepsilon\,\Gamma(-\varepsilon)\left(\frac{-\tilde{p_3}^2}{4\pi\mu_R^2}\right)^{\varepsilon} \frac{1}{-p_3^2} \frac{B(n_x+\varepsilon, n_y+\varepsilon)}{n_x+n_y+2\varepsilon}$$

$n_x, n_y \geq 0$ integers, μ_R^2 : energy scale normalization constant.

One off-shell, two on-shell particles

A massless one-loop vertex non-scalar integral for the case of one off-shell particle (with $p_3^2 \neq 0$) and two on-shell particles ($p_1^2 = p_2^2 = 0$) is given in [5] by

$$J_3\left(0,0,p_3^2;n_x,n_y\right) = \frac{1}{(4\pi)^2} \lim_{\varepsilon\to 0} I_3^{n_x,n_y}(\varepsilon)$$

Integral

$$I_3^{n_x,n_y}(\varepsilon) = \frac{\varepsilon\,\Gamma(-\varepsilon)}{(4\pi\mu_R^2)^{\varepsilon}} \int_0^1 dx \int_0^{1-x} dy \, \frac{x^{n_x} y^{n_y}}{(-p_3^2 xy - i0)^{1-\varepsilon}} \tag{4}$$
$$= \varepsilon\,\Gamma(-\varepsilon) \left(\frac{-\tilde{p_3}^2}{4\pi\mu_R^2}\right)^{\varepsilon} \frac{1}{-p_3^2} \frac{B(n_x+\varepsilon, n_y+\varepsilon)}{n_x+n_y+2\varepsilon}$$

$n_x, n_y \geq 0$ integers, μ_R^2 : energy scale normalization constant.

Asymptotics

There is no IR divergence and ε can be set to 0 when n_x and n_y are both non-zero. When one of n_x and n_y is zero and the other is not, e.g., $n_x = n > 0$, the asymptotic behavior of (4) is as that of the sample problem (2) shown previously, i.e.,

Asymptotic behavior

$$I_3^{n,0}(\varepsilon) \sim \frac{1}{p_3^2}\left(\frac{C_{-1}}{\varepsilon} + C_0 + \mathcal{O}(\varepsilon)\right), \tag{5}$$

with coefficients satisfying

$$C_{-1} = \frac{1}{n}$$

$$C_0 = -\frac{2}{n^2} + \frac{1}{n}\left(\log(-p_3^2) - \sum_{j=1}^{n-1}\frac{1}{j}\right)$$

Linear extrapolation results $n_x > 0, n_y = 0$

INT.#	NUMERICAL INTEGRATION (DQAGSE)^2			EXTRAPOLATION REL. ERR.	
	REL. ERR.	# EVALS	TIME (s)	C_{-1}	C_0
6	3.29e-15	175,935	3.89e-02	1.21e-05	3.35e-04
7	1.04e-15	203,115	4.52e-02	3.85e-07	1.53e-05
8	4.65e-15	230,055	5.02e-02	8.11e-09	4.73e-07
9	2.41e-15	249,885	5.40e-02	1.28e-10	1.06e-08
10	5.00e-16	270,135	5.86e-02	1.27e-12	1.54e-10
11	4.73e-15	283,995	6.21e-02	2.65e-13	3.04e-11
12	1.56e-14	305,805	6.73e-02	8.51e-13	1.59e-10
13	1.56e-14	363,915	8.11e-02	1.11e-12	3.19e-10
14	7.42e-15	439,065	1.00e-01	2.13e-14	4.02e-11

Table: Integration and extrapolation results using (DQAGSE)^2 for IR vertex (4)-(5) (real part), $n_x = 2, n_y = 0, p_3^2 = 100$; integ. error tolerances $t_r = 10^{-13}$ (outer), 5×10^{-14} (inner); Bulirsch sequence, integration $\#k$ corresponds to $\varepsilon = 1/b_{k+1}$; integration and extrapolation relative errors are given.

Asymptotics

When $n_x = n_y = 0$, J_3 behaves asymptotically as

Asymptotic behavior

$$I_3^{0,0}(\varepsilon) \sim \frac{1}{p_3^2}\left(\frac{C_{-2}}{\varepsilon^2} + \frac{C_{-1}}{\varepsilon} + C_0 + \mathcal{O}(\varepsilon)\right), \qquad (6)$$

with

$$C_{-2} = 1$$
$$C_{-1} = \log(-p_3^2)$$
$$C_0 = -\frac{\pi^2}{12} + \frac{1}{2}\log^2(-p_3^2)$$

In this case, the integrand of (4) is singular at both axes, while, for $n_x > 0$, the singularity occurs at the y-axis only. The adaptive numerical integration performs intensive subdivisions near the singularity.

Linear extrapolation results $n_x = n_y = 0$

INT.#	NUMERICAL INTEGRATION (DQAGSE)^2			EXTRAPOLATION RELATIVE ERROR		
	REL. ERR.	# EVALS	TIME (s)	C_{-2}	C_{-1}	C_0
6	7.86e-14	642,945	1.29e-01	3.74e-05	2.69e-03	7.64e-02
7	1.48e-13	1,139,385	2.29e-01	1.17e-06	2.81e-03	1.05e-03
8	7.48e-14	1,160,595	2.34e-01	2.44e-08	8.07e-07	4.71e-05
9	1.21e-13	1,984,485	4.03e-01	3.76e-10	1.77e-08	1.49e-06
10	2.30e-13	1,613,175	3.29e-01	3.06e-12	1.44e-10	7.99e-09
11	1.86e-13	1,454,535	2.99e-01	1.21e-12	1.47e-10	2.74e-08
12	6.43e-14	2,928,930	6.07e-01	3.98e-12	3.81e-10	6.59e-08

Table: Integration and extrapolation results using (DQAGSE)^2 for IR vertex (4), (6) (real part), $n_x = n_y = 0, p_3^2 = 100$; Bulirsch sequence, integration $\#k$ corresponds to $\varepsilon = 1/b_{k+1}$; integ. error tolerances $t_r = 10^{-13}$ (outer), 5×10^{-14} (inner); integration and extrapolation relative errors are given.

Hypergeometric function

Hypergeometric functions appear in expressions of massless one-loop integrals, as given in [5] for the tensor integrals of one-loop vertex and box functions.

Euler integral representation of the hypergeometric function

$$ {}_2F_1\,(a,b,c;z) = \frac{\Gamma(c)}{\Gamma(b)\,\Gamma(c-b)} \int_0^1 \frac{t^{\,b-1}(1-t)^{\,c-b-1}}{(1-t\,z)^{\,a}}\,dt \qquad (7) $$

where $\mathcal{R}c > \mathcal{R}b > 0$, denotes a one-valued analytic function in the complex plane cut along the real axis from 1 to ∞. The integral defines an analytic continuation of

Gauss series

$$ F(a,b,c;z) = \sum_{k=0}^{\infty} \frac{\Gamma(a+k)\,\Gamma(b+k)}{\Gamma(c+k)}\,\frac{z^k}{k!} $$

which has $|\,z\,| = 1$ as its circle of convergence [1]. The integrand of (7) has end-point singularities when $c-b-1<0$ and/or $b-1<0$.

Hypergeometric function

Hypergeometric functions appear in expressions of massless one-loop integrals, as given in [5] for the tensor integrals of one-loop vertex and box functions.

Euler integral representation of the hypergeometric function

$$ {}_2F_1\,(a,b,c;z) = \frac{\Gamma(c)}{\Gamma(b)\,\Gamma(c-b)} \int_0^1 \frac{t^{\,b-1}(1-t)^{\,c-b-1}}{(1-t\,z)^{\,a}}\,dt \tag{7} $$

where $\mathcal{R}c > \mathcal{R}b > 0$, denotes a one-valued analytic function in the complex plane cut along the real axis from 1 to ∞. The integral defines an analytic continuation of

Gauss series

$$ F(a,b,c;z) = \sum_{k=0}^{\infty} \frac{\Gamma(a+k)\,\Gamma(b+k)}{\Gamma(c+k)}\,\frac{z^k}{k!} $$

which has $|\,z\,| = 1$ as its circle of convergence [1]. The integrand of (7) has end-point singularities when $c-b-1<0$ and/or $b-1<0$.

Hypergeometric function: δ-extrapolation

We obtain numerical results using automatic integration and extrapolation,

δ-extrapolation

by replacing z by $z + i\delta$ and extrapolating to the limit of (7) as $\delta \to 0$. For the extrapolation, ${}_2F_1\,(a, b, c; z + i\delta_\ell)$ is computed for a sequence of δ_ℓ which tends to 0.

Linear method

Extrapolate

$$\sum_{k=0}^{n-1} C_k\, {\delta_\ell}^k = I(\delta_\ell) \tag{8}$$

for C_0, where $I(\delta_\ell) \approx {}_2F_1\,(a, b, c; z + i\delta_\ell), \quad 1 \le \ell \le n.$

Hypergeometric function: Test results

- We use results from [6] (NCI) as a reference, for an integral of the form (7) with $a = 1 + l,\ b = l + m,\ c = 1 + l + m + n$, at $z + i0 = 10 + i0$.
- For these tests, $l, m, n > 0$ are integers and $b \geq 1$ as well as $c - b \geq 1$, so that the integrand numerator is polynomial, thus without end-point singularities. We employ the general adaptive integrator DQAGE of QUADPACK using the Gauss-Kronrod rule pair with 10 Gauss and 21 Kronrod points. DQAGE lets the user select one of the Gauss-Kronrod pairs with 7-15, 10-21, 15-31, 20-41, 25-51 and 30-61 points as determined by an input parameter.

Hypergeometric function: Test results

- We use results from [6] (NCI) as a reference, for an integral of the form (7) with $a = 1 + l,\ b = l + m,\ c = 1 + l + m + n$, at $z + i0 = 10 + i0$.
- For these tests, $l, m, n > 0$ are integers and $b \geq 1$ as well as $c - b \geq 1$, so that the integrand numerator is polynomial, thus without end-point singularities. We employ the general adaptive integrator DQAGE of QUADPACK using the Gauss-Kronrod rule pair with 10 Gauss and 21 Kronrod points. DQAGE lets the user select one of the Gauss-Kronrod pairs with 7-15, 10-21, 15-31, 20-41, 25-51 and 30-61 points as determined by an input parameter.

Linear extrapolation results for ${}_2F_1\left(a,b,c;z+i\delta_\ell\right)$

Parameters l	m	n	Real/ Imag.	Extrapolated Result	Estim. Rel. Extr. Err.	n	Total Int. Time (s)
1	1	1	Real	-1.453322029e-02	1.10e-11	11	1.73e-02
			Imag.	1.507964474e-01	9.28e-12	10	2.91e-03
1	2	3	Real	8.417767169e-02	8.12e-11	10	2.37e-03
			Imag.	-2.290221044e-01	1.38e-11	10	3.10e-03
2	1	1	Real	1.087664688e-02	2.22e-11	11	2.07e-02
			Imag.	2.638937829e-02	3.61e-11	10	2.25e-03
2	3	4	Real	-2.890568082e-02	3.04e-11	11	9.30e-03
			Imag.	5.578978464e-02	2.37e-11	11	1.33e-02
3	1	2	Real	-1.026721798e-02	7.53e-10	14	3.23e-02
			Imag.	-5.654866773e-02	1.64e-09	13	3.84e-02
3	4	5	Real	8.121358827e-03	2.21e-10	13	3.00e-02
			Imag.	-1.274636264e-02	7.42e-11	11	2.55e-02

Table: Integration and extrapolation results for the hypergeometric function (7), $a=1+l,\ b=l+m,\ c=1+l+m+n$, at $z+i0=10+i0$; using $(\text{DQAGE})^2$ with rel. integ. error tol. of 5×10^{-14} and Bulirsch sequence for extrapolation to estim. rel. error 10^{-10}.

Hypergeometric function results – Remarks

- For the numerical tests of Table 4 we used $\delta_\ell = 1/b_\ell$ for an extension of the Bulirsch sequence with $b_\ell = 0.5, 0.75, 1, 1.5, 2, 3, 4, \ldots$.
- We compared the results of successive extrapolations to estimate the accuracy. Since the results in [6] are given to 10 digits, we performed extrapolations until the estimated relative error fell below 10^{-10}, which is listed in Table 4 together with the extrapolation result, the order n of the linear system (8) and the total time accumulated in the integrations needed for the right hand side in each case.
- For $(l, m, n) = (3, 1, 2)$, the best obtained estimated accuracy is reported; the obtained real part agrees with the exact value, whereas the imaginary part differs by 3×10^{-12}. In this case, the extrapolation was started at $\delta = 8$ ($b_0 = 0.125$) in view of the difficulty of the integration. For $(l, m, n) = (3, 4, 5)$, the real part absolute error is 3×10^{-12}. All other results agree with [6] to the required accuracy. The integrations for the right hand sides were performed to a tolerated relative error of 5×10^{-14}.

Hypergeometric function results – Remarks

- For the numerical tests of Table 4 we used $\delta_\ell = 1/b_\ell$ for an extension of the Bulirsch sequence with $b_\ell = 0.5, 0.75, 1, 1.5, 2, 3, 4, \ldots$.
- We compared the results of successive extrapolations to estimate the accuracy. Since the results in [6] are given to 10 digits, we performed extrapolations until the estimated relative error fell below 10^{-10}, which is listed in Table 4 together with the extrapolation result, the order n of the linear system (8) and the total time accumulated in the integrations needed for the right hand side in each case.
- For $(l, m, n) = (3, 1, 2)$, the best obtained estimated accuracy is reported; the obtained real part agrees with the exact value, whereas the imaginary part differs by 3×10^{-12}. In this case, the extrapolation was started at $\delta = 8$ ($b_0 = 0.125$) in view of the difficulty of the integration. For $(l, m, n) = (3, 4, 5)$, the real part absolute error is 3×10^{-12}. All other results agree with [6] to the required accuracy. The integrations for the right hand sides were performed to a tolerated relative error of 5×10^{-14}.

Hypergeometric function results – Remarks Cont'd

- Nonlinear extrapolation by the ϵ-algorithm can be applied with a geometric sequence of $\delta_\ell = \beta^{\sigma-\ell}$, e.g., with $\beta = 1.5$, $\ell = 0, 1, \ldots$ and a starting point determined by σ. The ϵ-algorithm is able to extrapolate the (divergent) sequence $I(\delta_\ell) \approx {}_2F_1\,(a, b, c; z + i\delta_\ell)$ as $\delta_\ell \to 0$ to C_0.
- However, linear extrapolation with a Bulirsch type sequence provides more efficient results (particularly) for the larger values of l, m and n.
- We note that the we successully used the same methods for the computation of the generalized hypergeometric function types of [6], related to ${}_3F_2\,(a, b, c; z)$ and ${}_4F_3\,(a, b, c; z)$, respectively. We obtained good convergence except for the case of nearly real z and $|z|$ very large (e.g., $z = 2000 + i10^{-15}$).

Hypergeometric function results – Remarks Cont'd

- Nonlinear extrapolation by the ϵ-algorithm can be applied with a geometric sequence of $\delta_\ell = \beta^{\sigma-\ell}$, e.g., with $\beta = 1.5$, $\ell = 0, 1, \ldots$ and a starting point determined by σ. The ϵ-algorithm is able to extrapolate the (divergent) sequence $I(\delta_\ell) \approx {}_2F_1\,(a, b, c; z + i\delta_\ell)$ as $\delta_\ell \to 0$ to C_0.
- However, linear extrapolation with a Bulirsch type sequence provides more efficient results (particularly) for the larger values of l, m and n.
- We note that the we successully used the same methods for the computation of the generalized hypergeometric function types of [6], related to ${}_3F_2\,(a, b, c; z)$ and ${}_4F_3\,(a, b, c; z)$, respectively. We obtained good convergence except for the case of nearly real z and $|z|$ very large (e.g., $z = 2000 + i10^{-15}$).

Hypergeometric function results – Remarks Cont'd

- Nonlinear extrapolation by the ϵ-algorithm can be applied with a geometric sequence of $\delta_\ell = \beta^{\sigma-\ell}$, e.g., with $\beta = 1.5$, $\ell = 0, 1, \ldots$ and a starting point determined by σ. The ϵ-algorithm is able to extrapolate the (divergent) sequence $I(\delta_\ell) \approx {}_2F_1\,(a, b, c; z + i\delta_\ell)$ as $\delta_\ell \to 0$ to C_0.
- However, linear extrapolation with a Bulirsch type sequence provides more efficient results (particularly) for the larger values of l, m and n.
- We note that the we successully used the same methods for the computation of the generalized hypergeometric function types of [6], related to ${}_3F_2\,(a, b, c; z)$ and ${}_4F_3\,(a, b, c; z)$, respectively. We obtained good convergence except for the case of nearly real z and $|z|$ very large (e.g., $z = 2000 + i10^{-15}$).

Vertex with one on-shell and two off-shell particles

For $p_1^2 = 0,\ p_2^2 \neq 0$ and $p_3^2 \neq 0$, $J_3\left(0, p_2^2, p_3^2; n_x, n_y\right) = \frac{1}{(4\pi)^2} \lim_{\varepsilon\to 0} I_{23}^{n_x,n_y}(\varepsilon)$ describes the case of one on-shell particle and two off-shell external legs [5], where

Integral

$$I_{23}^{n_x,n_y}(\varepsilon) = \frac{\varepsilon\,\Gamma(-\varepsilon)}{(4\pi\mu_R^2)^{\varepsilon}} \int_0^1 dx \int_0^{1-x} dy\, \frac{x^{n_x} y^{n_y}}{\left(-(p_3^2 - p_2^2)xy - p_2^2 y(1-y) - i0\right)^{1-\varepsilon}} \tag{9}$$

$$= J_3\left(0, 0, p_3^2; n_x, n_y\right)\ {}_2F_1\left(1, 1-\varepsilon, 2+n_x; \frac{p_3^2 - p_2^2}{\tilde{p_3}^2}\right). \tag{10}$$

IR divergence occurs when $n_x = n_y = 0$ or $n_x \neq 0,\ n_y = 0$ satisfying

Asymptotic behavior

$$I_{23}^{n_x,n_y}(\varepsilon) \sim \frac{1}{p_3^2}\left(\frac{C_{-1}}{\varepsilon} + C_0 + \mathcal{O}(\varepsilon)\right). \tag{11}$$

ε, δ-extrapolation

- Both forms (9) and (10) exhibit possible IR divergence as indicated by the ε-parameter. Furthermore, δ-extrapolation may be warranted for the computation of the hypergeometric function in (10), and in view of a possible zero denominator $D^{1-\varepsilon}$ in (9).
- It is possible to obtain results using a type of ε, δ-extrapolation, if necessary in case of IR divergence and $|z| < 1$ for the hypergeometric function or D becomes 0 within the domain of integration.
- We obtained preliminary results with an ε, δ-extrapolation in case that D was set to become 0 inside of the integration region.

ε, δ-extrapolation

- Both forms (9) and (10) exhibit possible IR divergence as indicated by the ε-parameter. Furthermore, δ-extrapolation may be warranted for the computation of the hypergeometric function in (10), and in view of a possible zero denominator $D^{1-\varepsilon}$ in (9).
- It is possible to obtain results using a type of ε, δ-extrapolation, if necessary in case of IR divergence and $|z| < 1$ for the hypergeometric function or D becomes 0 within the domain of integration.
- We obtained preliminary results with an ε, δ-extrapolation in case that D was set to become 0 inside of the integration region.

ε, δ-extrapolation

- Both forms (9) and (10) exhibit possible IR divergence as indicated by the ε-parameter. Furthermore, δ-extrapolation may be warranted for the computation of the hypergeometric function in (10), and in view of a possible zero denominator $D^{1-\varepsilon}$ in (9).
- It is possible to obtain results using a type of ε, δ-extrapolation, if necessary in case of IR divergence and $|z| < 1$ for the hypergeometric function or D becomes 0 within the domain of integration.
- We obtained preliminary results with an ε, δ-extrapolation in case that D was set to become 0 inside of the integration region.

Conclusions

- Extension to box integrals: preliminary results have been obtained for scalar problem using nonlinear extrapolation (and sector decomposition for the integral approximation).
- Vertex and box levels are important for one-loop N-point functions as IR divergence (if present) appears on these levels after reductions.
- Much more testing needed on ε, δ-extrapolations.

Conclusions

- Extension to box integrals: preliminary results have been obtained for scalar problem using nonlinear extrapolation (and sector decomposition for the integral approximation).
- Vertex and box levels are important for one-loop N-point functions as IR divergence (if present) appears on these levels after reductions.
- Much more testing needed on ε, δ-extrapolations.

Conclusions

- Extension to box integrals: preliminary results have been obtained for scalar problem using nonlinear extrapolation (and sector decomposition for the integral approximation).
- Vertex and box levels are important for one-loop N-point functions as IR divergence (if present) appears on these levels after reductions.
- Much more testing needed on ε, δ-extrapolations.

Bibliography

ABRAMOWITZ, M., AND STEGUN, I. S.
Handbook of Mathematical Functions.
Dover Publications, Inc., New York, 1965.

BREZINSKI, C.
A general extrapolation algorithm.
Numerische Mathematik 35 (1980), 175–187.

BULIRSCH, R.
Bemerkungen zur romberg-integration.
Numerische Mathematik 6 (1964), 6–16.

FORD, W., AND SIDI, A.
An algorithm for the generalization of the Richardson extrapolation process.
SIAM Journal on Numerical Analysis 24 (1987), 1212–1232.

KURIHARA, Y.
Dimensionally regularized one-loop tensor integrals with massless internal particles, 2005.
hep-ph/0504251 v3.

KURIHARA, Y., AND KANEKO, T.
Numerical contour integration for loop integrals.
Computer Physics Communications 174, 7 (2006), 530–539.
hep-ph/0503003 v1.

LEVIN, D., AND SIDI, A.
Two classes of non-linear transformations for accelerating the convergence of infinite integrals and series.
Appl. Math. Comp. 9 (1981), 175–215.

PIESSENS, R., DE DONCKER, E., ÜBERHUBER, C. W., AND KAHANER, D. K.
QUADPACK, A Subroutine Package for Automatic Integration.
Springer Series in Computational Mathematics. Springer-Verlag, 1983.

SHANKS, D.
Non-linear transformations of divergent and slowly convergent sequences.
J. Math. and Phys. 34 (1955), 1–42.

SIDI, A.
Convergence properties of some nonlinear sequence transformations.
Math. Comp. 33 (1979), 315–326.

WYNN, P.
On a device for computing the $e_m(s_n)$ transformation.
Mathematical Tables and Aids to Computing 10 (1956), 91–96.