# Numerical Optimization in DROP

**v4.21** 14 April 2019

# Introduction

## Framework Glossary

1. Hyperspace Search: Hyperspace search is a search to determine whether the entity is inside the zone of a range, e.g., bracketing search.
2. Hyperpoint Search: Hyperpoint searches are hard searches that search for an exact specific point (to within an appropriately established tolerance).
3. Iterate Nodes: This is the set of the traveled nodes (variate/Objective Function ordered pairs) that contain the trajectory traveled.
4. Iteration Search Primitives: The set of variate iteration routines that generate the subsequent iterate nodes.
5. Compound iterator search scheme: Search schemes where the primitive iteration routine to be invoked at each iteration are evaluated.
6. RunMap: Map that holds the program state at the end of each iteration, in the generic case, this is composed of the Wengert iterate node list, along with the corresponding root finder state.
7. Cost of Primitive (cop): This is the cost of invocation of a single variate iterator primitive.

## Document Layout

1. Base Framework
2. Search Initialization
    a. Bracketing
    b. Objective Function Failure
    c. Bracketing Start Initialization

# Framework

1. The root search given an objective function and its goal is achieved by iteratively evolving the variate, and involves the following steps:
    - Search initialization and root reachability determination: Searched is kicked off by spawning a root variate iterator for the search initialization process (described in detail in the next section).
    - Absolute Tolerance Determination.
    - Root Search Iteration: The root is searched iteratively according to the following steps:
        1. The iterator progressively reduces the bracket width.
        2. Successive iteration occurs using either a single primitive (e.g., using the bisection primitive), or using a selector scheme that picks the primitives for each step (e.g., Brent’s method).
        3. For Open Method, instead of 1 and 2, the routine drives towards convergence iteratively.
    - Search Termination Detection: The search termination occurs typically based on the following:
        - Proximity to the Objective Function Goal
        - Convergence on the variate
        - Exhaustion if the number of iterations
2. The flow behind these steps is illustrated in Figure 1.
3. The “Flow Control Variate” in root search is the “Objective Function Distance to Goal” Metric.

# Search Initialization

1. Broadly speaking, root finding approaches can be divided into a) those that bracket roots before they solve for them, and b) those that don’t need to bracket, opting instead to pick a suitable starting point.
2. Depending upon the whether the search is a bracketing or an open method, the search initialization does one the following:
    - Determine the root brackets for bracketing methods
    - Locate root convergence zone for open methods
3. Initialization begins by a search for the starting zone. A suitable starting point/zone is determined where, by an appropriate choice for the iterator, you are expected to reach the fixed-point target within a sufficient degree of reliability. Very general-purpose heuristics often help determine the search start zone.
4. Both bracketing and open initializers are hyperspace searches, since they search for something “IN”, not “AT”.

## Bracketing

1. Bracketing is the process of localizing the fixed point to within a target zone with the least required number of Objective Function calculations. Steps are:
    - Determine a valid bracketing search start
    - Choose a suitable bracket expansion
    - Limit attention to where the Objective Function is defined (more on this below).
2. Figure 2 shows the flow for the Bracketing routine.
3. Bracketing methods require that the initial search interval bracket the root (i.e. the function values at interval end points have opposite signs).

4. Bracketing traps the fixed point between two variate values, and uses the intermediate value theorem and the continuity of the Objective Function to guarantee the presence/existence of the fixed point between them.
5. Unless the objective function is discontinuous, bracketing methods guarantee convergence (although may not be within the specified iteration limit).
6. Typically, they do not require the objective function to be differentiable.
7. Bracketing iteration primitives' convergence is usually linear to super-linear.
8. Bracketing methods preserve bracketing throughout computation and allow user to specify which side of the convergence interval to select as the root.
9. It is also possible to force a side selection after a root has been found, for example, in sequential search, to find the next root.
10. Generic root bracketing methods that treat the objective function as a black box will be slower than targeted ones – so much so that they can constitute the bulk of the time for root search. This is because, to accommodate generic robustness coupled with root-pathology avoidance (oscillating bracket pairs etc.), these methods have to perform a full variate space sweep without any assumptions regarding the location of the roots (despite this most bracketing algorithms cannot guarantee isolation of root intervals). For instance, naïve examination of the Objective Function's "sign-flips" alone can be misleading, especially if you bracket fixed-points with even numbered multiplicity within the brackets. Thus, some ways of analyzing the Black Box functions (or even the closed form Objective Functions) are needed to better target/customize the bracketing search (of course, parsimony in invoking the number of objective function calls is the main limitation).
11. Soft Bracketing Zone: One common scenario encountered during bracketing is the existence of a soft preferred bracketing zone, one edge of which serves as a "natural edge". In this case, the bracketing run needs to be positioned to be able to seek out starting variate inside soft zone in the direction AWAY from the natural edge.
12. The first step is to determine a valid bracketing search start. One advantage with univariate root finding is that objective function range validity maybe established

using an exhaustive variate scanner search without worrying about combinatorial explosion.

## Objective Function Failure

1. Objective Function may fail evaluation at the specified variate for the following reason:
    - Objective Function is not defined at the specified variate.
    - Objective Function evaluates to a complex number.
    - Objective Function evaluation produces NaN/Infinity/Under-flow/Over-flow errors.
    - In such situations, the following steps are used to steer the variate to a valid zone.
2. Objective Function undefined at the Bracketing Candidate Variate: If the Objective Function is undefined at the starting variate, the starting variate is expanded using the variate space scanner algorithm described above. If the objective Function like what is seen in Figure 3, a valid starting variate will eventually be encountered.
3. Objective Function not defined at any of the Candidate Variates: The risk is that the situation in Figure 4 may be encountered, where the variate space scanner iterator "jumps over" the range over which the objective function is defined. This could be because the objective function may have become complex. In this case, remember that an even power of the objective function also has the same roots as the objective function itself. Thus, solving for an even power of the objective function (like the square) – or even bracketing for it – may help.

## Bracketing Start Initialization

1. Figure 5 shows the flow behind a general-purpose bracket start locator.

2. Once the starting variate search is successful, and the objective function validity is range-bound, then use an algorithm like bisection to bracket the root (as shown in Figure 6 below).
3. However, if the objective function runs out of its validity range under the variate scanner scheme, the following steps need to be undertaken:
    - If the left bracketing candidate fails, bracketing is done towards the right using the last known working left-most bracketing candidate as the “left edge”.
    - Likewise, if the right bracketing candidate fails, bracketing is done towards the left using the last known working right-most bracketing candidate as the “right edge”.
4. The final step is to trim the variate zone. Using the variate space scanner algorithm, and the mapped variate/Objective Function evaluations, the tightest bracketing zones are extracted (Figure 7).

## Open Search Initialization

1. Non-bracketing methods use a suitable starting point to kick off the root search. As is obvious, the chosen starting point can be critical in determining the fate of the search. In particular, it should be within the zone of convergence of the fixed-point root to guarantee convergence. This means that specialized methods are necessary to determine zone of convergence.
2. When the objective function is differentiable, the non-bracketing root finder often may make use of that to achieve quadratic or higher speed of convergence. If the non-bracketing root finder cannot/does not use the objective function’s differentiability, convergence ends up being linear to super-linear.
3. The typical steps for determining the open method starting variate are:
    - Find a variate that is proximal to the fixed point
    - Verify that it satisfies the convergence heuristic

4. Bracketing followed by a choice of an appropriate primitive variate (such as bisection/secant) satisfies both, thus could be a good starting point for open method searches like Newton's method.
5. Depending upon the structure of the Objective Function, in certain cases the chain rule can be invoked to ease the construction of the derivative – esp. in situations where the sensitivity to inputs are too high/low.

## Search/Bracketing Initializer Heuristic Customization

1. Specific Bracketing Control Parameters
2. Left/Right Soft Bracketing Start Hints: The other components may be used from the bracketing control parameters.
3. Mid Soft Bracketing Start Hint: The other components may be used from the bracketing control parameters.
4. Floor/Ceiling Hard Bracketing Edges: The other components may be used from the bracketing control parameters.
5. Left/Right Hard Search Boundaries: In this case, no bracketing is done – brackets are used to verify the roots, search then starts directly.

# Numerical Challenges in Search

1. Bit Cancellation
2. Ill-conditioning (e.g., see high order polynomial roots)
3. "domains of indeterminacy" – existence of sizeable intervals around which the objective function hovers near the target
4. Continuous, but abrupt changes (e.g., near-delta Gaussian objection function)
5. Under-flow/over-flow/round-off errors
6. root multiplicity (e.g., in the context of polynomial roots)
7. Typical solution is to transform the objective function to a better conditioned function – insight into the behavior of the objective can be used to devise targeted solutions.

# Variate Iteration

1.

$$\nu_{i+1} = I(\nu_i, \mathfrak{I}_i)$$

where $\nu_i$ is the i$^{th}$ variate and $\mathfrak{I}_i$ is the root finder state after the i$^{th}$ iteration.

2. Iterate nodes as Wengert variables: Unrolling the traveled iterate nodes during the forward accumulation process, as a Wengert list, is a proxy to the execution time, and may assist in targeted pre-accumulation and check-pointing.
3. Cognition Techniques of Mathematical Functions:
   - Wengert Variate Analysis => Collection of the Wengert variates helps build and consolidate the Objective Function behavior from the variate iterate Wengert nodes – to build a behavioral picture of the Objective Function.
   - Objective Function Neighborhood Behavior => With every Wengert variable, calculation of the set of forward sensitivities and the reverse Jacobians builds a local picture of the Objective Function without having to evaluate it.
4. Check pointing: Currently implemented using a roving variate/OF iterate node "RunMap"; this is also used to check circularity in the iteration process.
5. Compound Iterator RunMap: For compound iterations, the iteration circularity is determined the doublet $(\nu_i, \mathfrak{I}_i)$, so the Wengert RunMap is really a doublet Multi-Map.
6. Hyperpoint univariate fixed point search proximity criterion: For hyperpoint checks, the search termination check needs to explicitly accommodate a "proximity to target" metric. This may not be then case for hyperspace checks.
7. Regime crossover indicator: On one side the crossover, the variate is within the fast convergence zone, so you may use faster Open techniques like the Newton's methods. On the other side, continue using the bracketing techniques.

    a. Fast side of the crossover must be customizable (including other Halley's method variants); robust side should also be customizable (say False Position).

8. <u>Crossover indicator determination</u>: Need to develop targeted heuristics needed to determine the crossover indicator.
    - Entity that determines the crossover indicator may be determined from the relative variate shift change $\frac{x_{N+1}-x_N}{x_N-x_{N-1}}$ and the relative objective function change $\frac{y_{N+1}-y_N}{y_N-y_{N-1}}$.
9. <u>Types of bracketing primitives</u>:
    - Bracket narrower primitives (Bisection, false position), and interpolator primitives (Quadratic, Ridder).
    - Primitive's COP determinants: Expressed in terms of characteristic compute units.
        a. Number of objective function evaluation (generally expensive).
        b. Number of variate iterator steps needed.
        c. Number of objective function invocation per a given variate iteration step.
    - Bracket narrower primitives => Un-informed iteration primitives, low invocation cost (usually single objective function evaluation), but low search targeting quality, and high COP.
    - Interpolator primitives => Informed iteration primitives, higher invocation cost (multiple objective function evaluations, usually 2), better search targeting quality, and lower COP.
10. <u>Pre-OF Evaluation Compound Heuristic</u>: Heuristic compound variates are less informed, but rely heavily on heuristics to extract the subsequent iterator, i.e., pre-OF evaluation heuristics try to guide the evolution without invoking the expensive OF evaluations (e.g., Brent, Zheng).
11. <u>OF Evaluation Compound Heuristic</u>: These compound heuristics use the OF evaluations as part of the heuristics algorithm to establish the next variate => better informed

# Open Search Method: Newton's Method

1. Newton's method uses the objective function $f$ and its derivative $f'$ to iteratively evaluate the root.
2. Given a well-behaved function $f$ and it's derivative $f'$ defined over real $x$, the algorithm starts with an initial guess of $x_0$ for the root.
3. First iteration yields

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

4. This is repeated in

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

   till a value $x_n$ that is convergent enough is obtained.
5. If $\alpha$ is a simple root (root with multiplicity 1), and

$$\epsilon_n = x_n - \alpha$$

   and

$$\epsilon_{n+1} = x_{n+1} - \alpha$$

   respectively, then for sufficiently large $n$, the convergence is quadratic:

$$\epsilon_{n+1} \approx \frac{1}{2}\left|\frac{f''(x_n)}{f'(x_n)}\right| {\epsilon_n}^2$$

6. Newton’s method only works when $f$ has continuous derivatives in the root neighborhood.
7. When analytical derivatives are hard to compute, calculate slope through nearby points, but convergence tends to be linear (like secant).
8. If the first derivative is not well behaved/does not exit/undefined in the neighborhood of a particular root, the method may overshoot, and diverge from that root.
9. If a stationary point of the function is encountered, the derivative is zero and the method will fail due to division by zero.
10. The stationary point can be encountered at the initial or any of the other iterative points.
11. Even if the derivative is small but not zero, the next iteration will be a far worse approximation.
12. A large error in the initial estimate can contribute to non-convergence of the algorithm (owing to the fact that the zone is outside of the neighborhood convergence zone).
13. If $\alpha$ is a root with multiplicity $m > 1$, then for sufficiently large $n$, the convergence becomes linear, i.e.,

$$\epsilon_{n+1} \approx \frac{m-1}{m}\epsilon_n$$

14. When there are two or more roots that are close together then it may take many iterations before the iterates get close enough to one of them for the quadratic convergence to be apparent.
15. However, if the multiplicity $m$ of the root is known, one can use the following modified algorithm that preserves the quadratic convergence rate (equivalent to using successive over-relaxation)

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}$$

16. The algorithm estimates $m$ after carrying out one or two iterations, and then use that value to increase the rate of convergence. Alternatively, the modified Newton's method may also be used:

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{f'(x_n)f'(x_n) - f(x_n)f''(x_n)}$$

17. It is easy to show that if

$$f'(x_n) = 0$$

and

$$f''(x_n) \neq 0$$

the convergence in the neighborhood becomes linear. Further, if

$$f'(x_n) \neq 0$$

and

$$f''(x_n) = 0$$

convergence becomes cubic.

18. One way of determining the neighborhood of the root. Define

$$g(x) = x - \frac{f(x)}{f'(x)}$$

$$p_n = g(p_{n-1})$$

where $p$ is a fixed point of $g$, i.e.,

$$g \in C[a,b]$$

$k$ is a positive constant,

$$p_0 \in C[a,b]$$

and

$$g(x) \in C[a,b] \; \forall \; x \in C[a,b]$$

19. One sufficient condition for $p_0$ to initialize a convergent sequence $\{p_0\}_{k=0}^{\infty}$, which converges to the root

$$x = p$$

of

$$f(x) = 0$$

is that

$$x \in (p - \delta, p + \delta)$$

and that $\delta$ be chosen so that

$$\frac{f(x_n)f''(x_n)}{f'(x_n)f'(x_n)} \leq k < 1 \;\forall\; x \in (p - \delta, p + \delta)$$

20. It is easy to show that under specific choices for the starting variate, Newton's method can fall into a basin of attraction. These are segments of the real number line such that within each region iteration from any point leads to one particular root - can be infinite in number and arbitrarily small. Also, the starting or the intermediate point

can enter a cycle - the $n$-cycle can be stable, or the behavior of the sequence can be very complex (forming a Newton fractal).

21. Newton's method for optimization is equivalent to iteratively maximizing a local quadratic approximation to the objective function. But some functions are not approximated well by quadratic, leading to slow convergence, and some have turning points where the curvature changes sign, leading to failure. Approaches to fix this use a more appropriate choice of local approximation than quadratic, based on the type of function we are optimizing. [13] demonstrates three such generalized Newton rules. Like Newton's method, they only involve the first two derivatives of the function, yet converge faster and fail less often.
22. One significant advantage of Newton’s method is that it can be readily generalized to higher dimensions.
23. Also, Newton’s method calculates the Jacobian automatically as part of the calibration process, owing to the reliance on derivatives – in particular, automatic differentiation techniques can be effectively put to use.

# Closed Search Methods

## Secant

1. Secant method results on the replacement of the derivative in the Newton's method with a secant-based finite difference slope.
2. Convergence for the secant method is slower than the Newton's method (approx. order is 1.6); however, the secant method does not require the objective function to be explicitly differentiable.
3. It also tends to be less robust than the popular bracketing methods.

## Bracketing Iterative Search

1. Bracketing iterative root searches attempt to progressively narrow the brackets and to discover the root within.
2. The first set discusses the goal search univariate iterator primitives that are commonly used to iterate through the variate.
3. These goal search iterator primitives continue generating a new pair of iteration nodes (just like their bracketing search initialization counter-parts).
4. Certain iterator primitives carry bigger "local" cost, i.e., cost inside a single iteration, but may reduce global cost, e.g., by reducing the number iterations due to faster convergence.
5. Further, certain primitives tend to be inherently more robust, i.e., given enough iteration, they will find the root within – although they may not be fast.

6. Finally, the case of compound iterator search schemes, search schemes where the primitive iteration routine to be invoked at each iteration is evaluated on-the-fly, are discussed.
7. Iterative searches that maintain extended state across searches pay a price in terms of scalability – the price depending on the nature and the amount of state held (e.g., Brent's method carries iteration selection state, whereas Zheng's does not).

## Univariate Iterator Primitive: Bisection

1. Bisection starts by determining a pair of root brackets $a$ and $b$.
2. It iteratively calculates $f$ at

$$c = \frac{a+b}{2}$$

   then uses $c$ to replace either $a$ or $b$, depending on the sign. It eventually stops when $f$ has attained the desired tolerance.
3. Bisection relies on $f$ being continuous within the brackets.
4. While the method is simple to implement and reliable (it is a fallback for less reliable ones), the convergence is slow, producing a single bit of accuracy with each iteration.

## Univariate Iterator Primitive: False Position

1. False position works the same as bisection, except that the evaluation point $c$ is linearly interpolated; $f$ is computed at

$$c = \frac{bf(a) + af(b)}{f(a) + f(b)}$$

where $f(a)$ and $f(b)$ have opposite signs. This holds obvious similarities with the secant method.

2. False position method also requires that $f$ be continuous within the brackets.
3. It is simple enough, more robust than secant and faster than bisection, but convergence is still linear to super-linear.
4. Given that the linear interpolation of the false position method is a first-degree approximation of the objective function within the brackets, quadratic approximation using Lagrange interpolation may be attempted as

$$f(x) = \frac{(x - x_{n-1})(x - x_n)}{(x_{n-2} - x_{n-1})(x_{n-2} - x_n)} f_{n-2} + \frac{(x - x_{n-2})(x - x_n)}{(x_{n-1} - x_{n-2})(x_{n-1} - x_n)} f_{n-1} + \frac{(x - x_{n-2})(x - x_{n-1})}{(x_n - x_{n-2})(x_n - x_{n-1})} f_n$$

   where we use the three iterates, $x_{n-2}$, $x_{n-1}$ and $x_n$, with their function values, $f_{n-2}$, $f_{n-1}$ and $f_n$.
5. This reduces the number of iterations at the expense of the function point calculations.
6. Using higher order polynomial fit for the objective function inside the bracket does not always produce roots faster or better, since it may result in spurious inflections (e.g., Runge's phenomenon).
7. Further, quadratic or higher fits may also cause complex roots.

## Univariate Iterator Primitive: Inverse Quadratic

1. Performing a fit of the inverse $\frac{1}{f}$ instead of $f$ avoids the quadratic interpolation problem above. Using the same symbols as above, the inverse can be computed as

$$\frac{1}{f(y)} = \frac{(y - f_{n-1})(y - f_n)}{(f_{n-2} - f_{n-1})(f_{n-2} - f_n)} x_{n-2} + \frac{(y - f_{n-2})(x - f_n)}{(f_{n-1} - f_{n-2})(f_{n-1} - f_n)} x_{n-1} + \frac{(y - f_{n-2})(y - f_{n-1})}{(f_n - x_{n-2})(f_n - f_{n-1})} x_n$$

2. Convergence is faster than secant, but poor when iterates not close to the root, e.g., if two of the function values $f_{n-2}$, $f_{n-1}$ and $f_n$ coincide, the algorithm fails.

## Univariate iterator primitive: Ridder's

1. Ridders' method is a variant on the false position method that uses exponential function to successively approximate a root of $f$.
2. Given the bracketing variates, $x_1$ and $x_2$, which are on two different sides of the root being sought, the method evaluates $f$ at

$$x_3 = \frac{x_1 + x_2}{2}$$

3. It extracts exponential factor $\alpha$ such that $f(x)e^{\alpha x}$ forms a straight line across $x_{1,}$ $x_{2,}$ and $x_3$. A revised $x_2$ (named $x_4$) is calculated from

$$x_4 = x_3 + (x_3 - x_1)\frac{sign[f(x_1) - f(x_2)]f(x_3)}{\sqrt{f^2(x_3) - f(x_1)f(x_2)}}$$

2. Ridder's method is simpler than Brent's method, and has been claimed to perform about the same.
3. However, the presence of the square root can render it unstable for many of the reasons discussed above.

## Univariate compound iterator: Brent and Zheng

1. Brent's predecessor method first combined bisection, secant, and inverse quadratic to produce the optimal root search for the next iteration.
2. Starting with the bracket points $a_0$ and $b_0$, two provisional values for the next iterate are computed; the first given by the secant method

$$s = b_k - \frac{b_k - b_{k-1}}{f(b_k) - f(b_{k-1})} f(b_k)$$

   and the second by bisection

$$m = \frac{a_k + b_k}{2}$$

3. If $s$ lies between $b_k$ and $m$, it becomes the next iterate $b_{k+1}$, otherwise the $m$ is the next iterate.
4. Then, the value of the new contra-point is chosen such that $f(a_{k+1})$ and $f(b_{k+1})$ have opposite signs.
5. Finally, if

$$|f(a_{k+1})| < |f(b_{k+1})|$$

   then $a_{k+1}$ is probably a better guess for the solution than $b_{k+1}$, and hence the values of $a_{k+1}$ and $b_{k+1}$ are exchanged.
6. To improve convergence, Brent's method requires that two inequalities must be simultaneously satisfied.
   a) Given a specific numerical tolerance $\delta$, if the previous step used the bisection method, and if

$$\delta < |b_k - b_{k-1}|$$

the bisection method is performed and its result used for the next iteration. If the previous step used interpolation, the check becomes

$$\delta < |b_{k-1} - b_{k-2}|$$

b) If the previous step used bisection, if

$$|s - b_k| < \frac{1}{2}|b_k - b_{k-1}|$$

then secant is used; otherwise the bisection used for the next iteration. If the previous step performed interpolation

$$|s - b_k| < \frac{1}{2}|b_{k-1} - b_{k-2}|$$

is checked instead.

7. Finally, since Brent's method uses inverse quadratic interpolation, $s$ has to lie between $\frac{3a_k+b_k}{4}$ and $b_k$.
8. Brent’s algorithm uses three points for the next inverse quadratic interpolation, or secant rule, based upon the criterion specified above.
9. One simplification to the Brent’s method adds one more evaluation for the function at the middle point before the interpolation.
10. This simplification reduces the times for the conditional evaluation and reduces the interval of convergence.
11. Convergence is better than Brent’s, and as fast and simple as Ridder’s.

# Polynomial Root Search

1. This section carries out a brief treatment of computing roots for polynomials.
2. While closed form solutions are available for polynomials up to degree 4, they may not be stable numerically.
3. Popular techniques such as Sturm's theorem and Descartes' rule of signs are used for locating and separating real roots.
4. Modern methods such as VCA and the more powerful VAS use these with Bisection/Newton methods – these methods are used in Maple/Mathematica.
5. Since the eigenvalues of the companion matrix to a polynomial correspond to the polynomial's roots, common fast/robust methods used to find them may also be used.
6. A number of caveats apply specifically to polynomial root searches, e.g., Wilkinson's polynomial shows why high precision is needed when computing the roots – proximal/other ill-conditioned behavior may occur.
7. Finally, special ways exist to identify/extract multiplicity in polynomial roots – they use the fact that $f(x)$ and $f'(x)$ share the root, and by figuring out their GCD.

# Meta-heuristics

## Introduction

1. Definition: Meta-heuristic is a higher-level procedure or heuristic designed to find, generate, or select a lower level procedure or heuristic (partial search algorithm) that may provide a sufficiently good solution to an optimization problem, especially with incomplete or imperfect information or limited computation capacity (Bianchi, Dorigo, Gambardella, and Gutjahr (2009)).
2. Applicability: Meta-heuristics techniques make only a few assumptions about the optimization problem being addressed, so are usable across a variety of problem (Blum and Roli (2003)).
3. Underpinning Philosophy: Many kinds of meta-heuristics implement some kind of stochastic optimization, so the solution depends upon the random variables being generated. As such it does not guarantee that a globally optimal solution can be found over some class of problems.
4. Search Strategy: By searching over a large set of feasible solutions meta-heuristics often finds good solutions with less computation effort than other algorithms, iterative methods, or simple heuristics (see Glover and Kochenberger (2003), Goldberg (2003), Talbi (2009)).
5. Literature: While theoretical results are available (typically on convergence and the possibility of locating global optimum, see Blum and Roli (2003)), most results on meta-heuristics are experimental, describing empirical results based on computer experiments with the algorithms.
    - While high quality research exists (e.g., Sorensen (2013)), enormously numerous meta-heuristics algorithms published as novel/practical have been of flawed quality – often arising out of vagueness, lack of conceptual elaboration, and ignorance of previous literature (Meta-heuristics (Wiki)).

## Properties and Classification

1. Properties: This comes from Blum and Roli (2003):
    a. Meta-heuristics are strategies that guide the search process.
    b. The goal is to efficiently explore the search space in order to find near-optimal solutions.
    c. Techniques that constitute meta-heuristics range from simple local search procedures to complex learning processes.
    d. Meta-heuristic algorithms are approximate and usually non-deterministic.
    e. Meta-heuristics are not problem-specific.
2. Classification: These are taken from Blum and Roli (2003) and from Bianchi, Dorigo, Gambardella, and Gutjahr (2009):
    a. Classification based on the type of the search strategy
    b. Classification off-of single solution search vs. population based searches
    c. Classification off-of hybrid/parallel heuristics

## Meta-heuristics Techniques

1. Simple Local Search Improvements: In this family of techniques, the search strategy employed is an improvement on simple local search algorithms; examples include simulated annealing, tabu search, iterated local search, variable neighborhood search, and GRASP (Blum and Roli (2003)).
2. Search Improvements with Learning Strategies: The other type of search strategy has a learning component to the search; meta-heuristics of this type include ant colony optimization, evolutionary computation, and genetic algorithms (Blum and Roli (2003)).
3. Single Solution Searches: These focus on modifying and improving a single candidate solution; single solution meta-heuristics include iterated local search, simulated annealing, variable neighborhood search, and tabu search (Talbi (2009)).

4. Population based Searches: Population based searches maintain and improve multiple candidate solutions using population characteristics to guide the search. These meta-heuristics include evolutionary computation, genetic algorithms, and particle swarm optimization (Talbi (2009)).
5. Swarm Intelligence: Swarm intelligence is the collective behavior of de-centralized, self-organized agents in a particle or a swarm. Ant colony optimization (Dorigo (1992)), particle swarm optimization (Talbi (2009)), artificial bee colony (Karaboga (2010)) are all example algorithms of swarm intelligence.
6. Hybrid meta-heuristic: These combine meta-heuristics with other optimization approaches (these could come from e.g., mathematical programming, constraint programming, machine learning etc.). Components of the hybrid meta-heuristic run concurrently and exchange information to guide the search.
7. Parallel meta-heuristic: This employs parallel programming techniques to run multiple meta-heuristics searches in parallel; these may range from simple distributed schemes to concurrent search runs that interact to improve the overall solution.

## Meta-heuristics Techniques in Combinatorial Problems

1. Combinatorial Optimization Problems: In combinatorial optimization, an optimal solution is sought over a discrete search space. Typically the search-space of the candidate solution goes faster than exponentially as the problem-size increases, making an exhaustive search for the optimal solution infeasible (e.g., the Travelling Salesman Problem (TSP)).
2. Nature and Types of Combinatorial Problems: Multi-dimensional combinatorial problems (e.g., engineering design problems such as form-finding and behavior-finding (Tomoiaga, Chandris, Sumper, Sudria-Andrieu, and Villafafila-Robles (2013)) suffer from the usual curse of dimensionality, making them infeasible to analytical or exhaustive-search methods.
3. Meta-heuristics applied to Combinatorial Optimization: Popular meta-heuristic algorithms for combinatorial problems include genetic algorithms (Holland (1975)),

scatter search (Glover (1977)), simulated annealing (Kirkpatrick, Gelatt, and Vecchi (1983)), and tabu search (Glover (1986)).

## Key Meta-heuristics Historical Milestones

1. Contributions: Many different meta-heuristics are in existence, and new variants are being continually developed. The following key milestone contributions have been extracted from Meta-heuristics (Wiki).
2. 1950s:
    a. Robbins and Munro (1951) work on stochastic optimization methods.
    b. Barricelli (1954) carries out the first simulation of the evolutionary process and uses it on general optimization problems.
3. 1960s:
    a. Rastrigin (1963) proposes random search.
    b. Matyas (1965) proposes random optimization.
    c. Nelder and Mead (1965) propose a simplex heuristic, which later shown to converge to non-stationary points on some problems.
    d. Fogel, Owens, and Walsh (1966) propose evolutionary programming.
4. 1970s:
    a. Hastings (1970) proposes the Metropolis-Hastings algorithm.
    b. Cavicchio (1970) proposes adaptation of the control parameters for an optimizer.
    c. Kernighan and Lin (1970) propose a graph-partitioning method that is related to variable-depth search and prohibition based tabu search.
    d. Holland (1975) proposes genetic algorithm.
    e. Glover (1977) proposes scatter search.
    f. Mercer and Sampson (1978) propose a meta-plan for tuning the optimizer's parameters by using another optimizer.
5. 1980s:
    a. Smith (1980) describes genetic programming.
    b. Kirkpatrick, Gelatt, and Vecchi (1983) propose simulated annealing.

    c. Glover (1986) proposes tabu search, along with the first mention of the word meta-heuristic (Yang (2011)).
    d. Moscato (1989) proposes memetic algorithms.

6. 1990s:
    a. Dorigo (1992) introduce ant colony optimization in his PhD thesis.
    b. Wolpert and MacReady (1995) prove the no free lunch theorems (these are later extended by Droste, Jansen, and Wegener (2002), Igel and Toussaint (2003), and Auger and Teytaud (2010)).

## References

- Auger, A., and O. Teytaud (2010): Continuous Lunches are Free Plus the Design of Optimal Optimization Algorithms *Algorithmica* **57 (1)** 121-146.
- Barricelli, N. A (1954): Esempi Numerici di Processi di Evoluzione *Methodos* 45-68.
- Bianchi, L., M. Dorigo, L. M. Gambardella, and W. J. Gutjahr (2009): A Survey on Metaheuristics for Stochastic Combinatorial Optimization *Natural Computing: An International Journal* **8 (2)** 239-287.
- Blum, C., and A. Roli (2003): Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison *ACM Computing Surveys* **35 (3)** 268-308.
- Cavicchio, D. J. (1970): *Adaptive Search using Simulated Evolution* Technical Report **Computer and Communication Sciences Department, University of Michigan**
- Dorigo, M (1992): *Optimization, Learning, and Natural Algorithms* PhD Thesis **Politecnico di Milano**
- Droste, S., T. Jansen, and I. Wegener (2002): Optimization with Randomized Search Heuristics – The (A)NFL Theorem, Realistic Scenarios, and Difficult Functions *Theoretical Computer Science* **287 (1)** 131-144.
- Fogel, L., A. J. Owens, and M. J. Walsh (1966): *Artificial Intelligence Through Simulated Evolution* **Wiley**

- Glover, F. (1977): Heuristics for Integer Programming using Surrogate Constraints *Decision Sciences* **8 (1)** 156-166.
- Glover, F. (1986): Future Paths for Integer Programming and Links to Artificial Intelligence *Computers and Operations Research* **13 (5)** 533-549.
- Glover, F., and G. A. Kochenberger (2003): *Handbook of Metaheuristics* **57 Springer International Series in Operations Research and Management Science**.
- Goldberg, D. E. (1989): *Genetic Algorithms in Search, Optimization, and Machine Learning* **Kluwer Academic Publishers**
- Hastings, W. K. (1970): Monte Carlo Sampling Methods using Markov Chains and their Applications *Biometrika* **57 (1)** 97-109
- Holland, J. H. (1975): *Adaptation in Natural and Artificial Systems* **University of Michigan Press**
- Igel, C., and M. Toussaint (2003): On Classes of Functions for which the No Free Lunch Results hold *Information Processing Letters* **86 (6)** 317-321.
- Karaboga, D. (2010): Artificial Bee Colony Algorithm *Scholarpedia* **5 (3)** 6915.
- Kernighan, B. W., and S. Lin (1970): An Efficient Heuristic Procedure for Partitioning Graphs *Bell System Technical Journal* **49 (2)** 291-307
- Kirkpatrick, S., C. D. Gelatt Jr., M. P. Vecchi (1983): Optimization by Simulated Annealing *Science* **220 (4598)** 671-680.
- Matyas, J. (1965): Random Optimization *Automation and Remote Control* **26 (2)** 246-253.
- Mercer, R. E., and J. R. Sampson (1978): Adaptive Search using a Reproductive Meta-plan *Kybernetes* **7 (3)** 215-228.
- Moscato, P. (1989): *On Evolution, Search, Optimization, Genetic Algorithms, and Martial Arts: Towards Memetic Algorithms* Report 826 **Caltech Concurrent Computation Program**
- Nelder, J. A., and R. Mead (1965): A Simplex Method for Function Minimization *Computer Journal* **7** 308-313.

- Rastrigin, L. A. (1963): The Convergence of Random Search Method in the Extremal Control of a many Parameter System *Automation and Remote Control* **24 (10)** 1337-1342.
- Robbins, S., and H. Munro (1951): A Stochastic Approximation Method *Annals of Mathematical Statistics* **22 (3)** 400-407.
- Smith, S. F. (1980): *A Learning System based on Genetic Adaptive Algorithms* PhD Thesis **University of Pittsburgh**
- Sorensen, K. (2013): *Metaheuristics—the metaphor exposed*
- Talbi, E. G. (2009): *Metaheuristics: From Design to Implementation* **Wiley**.
- Tomoiaga, B., M. Chandris, A. Sumper, A. Sudria-Andrieu, and R. Villafafila-Robles (2013): Pareto Optimal Reconfiguration of Power Distribution Systems using a Genetic Algorithm based on NSGA-II *Energies* **6 (3)** 1439-1455.
- Wolpert, D. H., and W. G. MacReady (1995): *No Free Lunch Theorems for Search* Technical Report SFI-TR-95-02-010 **Santa Fe Institute**
- Yang, X. S. (2011): Metaheuristic Optimization *Scholarpedia* **6 (8)** 11472.

# Introduction and Overview

## Motivation, Background, and Setup

1. Mathematical Formulation and Framework: Convex Optimization is an important class of constrained optimization problems that subsumes linear and quadratic programming. Such problems are specified in the form

$$\underset{x \in \mathbb{R}^n}{min} f(x)$$

such that

$$x \in S$$

where the feasible set

$$S \subseteq \mathbb{R}^n$$

is a *convex* closed subset of $\mathbb{R}^n$ and $f$ is a *convex function* (Hauser (2012)).

2. Local Minima as Global Minima: Convex optimization problems are important because all local minima are also guaranteed to be globally minimal, although this value may not be reached necessarily at a unique point.

## Convex Sets and Convex Hull

1. Definition of a Convex Set: A convex set $S$ satisfies the following property. For any two points $x$ and $y$ in $S$ the point $(1-u)\cdot x+u\cdot y$ for

   $$u\in[0,1]$$

   lies in $S$ as well. In other words, the line segment between any two points in $S$ must also lie in $S$.
2. Definition of a Convex Hull: The smallest convex set containing another set $A$ is called the *convex Hull* of $C(A)$.
3. Definition of a Convex Function: A function of a scalar field $f$ is convex if it satisfies the following property:

   $$f\big((1-u)\cdot x+u\cdot y\big)\leq(1-u)\cdot f(x)+u\cdot f(y)$$

   for all $x$ and $y$ and any

   $$u\in[0,1]$$

   In other words, the function value between any two points $x$ and $y$ lies below the line of $f$ connecting $f(x)$ and $f(y)$. An equivalent definition is that on the line segment between $x$ and $y$ the area over the graph of $f$ forms a convex set.
4. Minima of a Convex Function: Convex functions are always somewhat “bowl shaped”, have no local maxima (unless constant), and have no more than one local minimum value. If a local minimum exists, then it is also a global minimum. Hence gradient descent and Newton methods (with line search) are guaranteed to produce the global minimum when applied to such functions.

## Properties of Convex Sets/Functions

1. Connection Property: Convex sets are connected.
2. Intersection Property: The intersection of any number of convex sets is also a convex set.
3. Axiomatic Convex Sets: The empty set, the whole space, any linear sub-spaces, and half-spaces are also convex.
4. Convex Set Dimensionality Reduction: If a convex set of $S$ is of a lower dimension than its surrounding space $\mathbb{R}^n$ then $S$ lies on a linear sub-space of $\mathbb{R}^n$.
5. Typical Convex Set Closure Properties: Some common convex functions include $e^x$, $-\log x$, $x^k$ where $k$ is an even number. Convex functions are closed under addition, the $max$ operator, monotonic transformations of the $x$ variable, and scaling by a non-negative constant.
6. Convex Set from Ellipsoidal Constraints: Constraints of the form

$$x^T A x \leq c$$

   produce a closed convex set (an ellipsoid) if $A$ is positive semi-definite, and $c$ is a non-negative number.
7. Transformation onto Semi-definite Programs: The set of positive semi-definite matrices is a closed convex set. Optimization problems with these constraints are known as *semi-definite programs* (SDPs). A surprising number of problems, including LPs and QPs, can be transformed into SDPs.

## Convex Optimization Problems

1. General Form of Convex Optimization: An optimization problem in general

$$\underset{x \in \mathbb{R}^n}{min} f(x)$$

such that

$$g_i(x) \le 0, i = 1, \cdots, m$$

and

$$h_j(x) = 0, j = 1, \cdots, p$$

is convex as long as $f$ is convex, all of the $g_i$ for

$$i = 1, \cdots, m$$

are convex, and all of the $h_j$ for

$$j = 1, \cdots, p$$

are linear.

2. Elimination of the Equality Constraints: In convex optimization we will typically eliminate the equalities before optimizing, either by converting them into two inequalities

$$h_j(x) \le 0$$

and

$$-h_j(x) \le 0$$

or by performing a null-space transformation. Hence the $h_j$'s can be dropped from typical treatments.

## References

- Hauser (2012): Convex Optimization and Interior Point Methods.

# Newton's Method in Optimization

## Method

1. Iterative Root Finding vs Optimization: In calculus Newton's method is an iterative method for finding the roots of a differentiable function $f$, i.e., solutions to the equation

$$f(x) = 0$$

   In optimization the Newton's method is applied to the derivative $f'$ of a twice-differentiable function $f$ to find the roots of the derivative, i.e., solutions to

$$f'(x) = 0$$

   also known as the stationary points of $f$.

2. The Stationary Point of $f$: In the $1D$ problem the Newton's method attempts to construct a sequence $x_n$ from an initial guess $x_0$ that converges towards some value $x^*$ satisfying

$$f(x^*) = 0$$

   This $x^*$ is a stationary point of $f$ (Newton's Method in Optimization (Wiki)).

3. Taylor Expansion of $f$ around $x_n$: One wishes to find $\Delta x$ such that $f(x_n + \Delta x)$ is a maximum. Thus one seeks to solve the equation that sets the derivative of the last expression with respect to $\Delta x$ equal to zero:

$$0 = \frac{\partial}{\partial \Delta x}\left\{f(x_n) + f'(x_n)\Delta x + \frac{1}{2}f''(x_n)[\Delta x]^2\right\} = f'(x_n) + f''(x_n)\Delta x$$

4. Conditions for approximating the Stationary Point: For the value of

$$\Delta x = -\frac{f'(x_n)}{f''(x_n)}$$

which is a solution to the above equation, typically

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

will be closer to the stationary point $x^*$. Provided that $f(x)$ is a twice-differentiable function and other technical conditions are satisfied, the sequence $x_1, \cdots, x_n$ will converge to the point $x^*$ satisfying

$$f(x^*) = 0$$

5. Geometric Interpretation of the Newton's Method: The geometric interpretation of the Newton's method is that at each iteration one approximates $f(\vec{x})$ by a quadratic function $\vec{x}_n$ and then takes a step towards a maximum/minimum of that quadratic function. In higher dimensions, this stationary point may also be a saddle point. Of course if $f(\vec{x})$ happens to *be* a quadratic function then the exact extremum is found in one step.

## Higher Dimensions

1. Determination of the Iteration Increment: The above iteration scheme can be generalized to several dimensions by replacing the derivative with a gradient $\vec{\nabla} f(\vec{x})$

and the reciprocal of the second derivative with the inverse of the Hessian matrix $\mathbb{H}f(\vec{x})$. One thus obtains the iteration scheme

$$\vec{x}_{n+1} = \vec{x}_n + [\mathbb{H}f(\vec{x})]^{-1}\vec{\nabla}f(\vec{x}), n \geq 0$$

2. Adjustment to the Iteration Increment: Often Newton's method is modified to include a small step size

   $$\gamma \in (0, 1)$$

   instead of

   $$\gamma = 1$$

   to result in

   $$\vec{x}_{n+1} = \vec{x}_n + \gamma[\mathbb{H}f(\vec{x})]^{-1}\vec{\nabla}f(\vec{x})$$

   This is often done to ensure that the Wolfe conditions are satisfied at each step

   $$\vec{x}_n \longrightarrow \vec{x}_{n+1}$$

   of the iteration.
3. Newton's Method - Speed of Convergence: Where applicable Newton's method converges much faster towards the local maximum or minimum than gradient descent. In fact every local minimum has a neighborhood $\mathbb{N}$ such that if one starts with

   $$x_0 \in \mathbb{N}$$

Newton’s method with step size

$$\gamma = 1$$

converges quadratically. The only requirements are that the Hessian be invertible, and it be a Lipschitz-continuous function of $\vec{x}$ in that neighborhood.

4. By-passing Explicit Hessian Computation: Finding the inverse of the Hessian in high dimensions can be an expensive exercise. In such cases, instead of directly inverting the Hessian it may be better to calculate the vector

$$\Delta\vec{x}_n = \vec{x}_{n+1} - \vec{x}_n$$

as a solution to the system of linear equations

$$[\mathbb{H}f(\vec{x}_n)]\vec{\nabla}f(\vec{x}_{n+1}) = -\vec{\nabla}f(\vec{x}_n)$$

This may be solved by various factorizations or approximately – and to great accuracy – using iterative methods.

5. Caveats of the Matrix Methods: Many of these methods are only applicable to certain types of equations – e.g., the Cholesky factorization and the conjugate gradient method will work only if $\mathbb{H}f(\vec{x}_n)$ is a positive definite matrix. While this may seem like a limitation, it is often a useful indicator of something gone wrong. For instance if a maximization problem is being considered and $\mathbb{H}f(\vec{x}_n)$ is not positive definite, the iterations end up converging to a saddle point and not to a minimum.
6. Stable Variants of these Methods: On the other hand if constrained optimization is being considered – for example using Lagrange multipliers – the problem may become one of saddle point finding, in which case the Hessian will be symmetric indefinite and a solution for $\vec{x}_{n+1}$ needs to be done with approaches that will work for such situations, such as the $LDL^T$ variant of the Cholesky factorization or the conjugate gradient method.

7. Techniques of Quasi-Newton Methods: There are exist various quasi-Newton methods where the approximation for the Hessian – or its direct inverse – is built up from the changes in the gradient.
8. Challenges with non-invertible Hessians: If a Hessian is close to a non-invertible matrix the inverted Hessian can be numerically unstable and the solution may diverge. In this case certain workarounds have been tried in the past, each of which have had varied success in differing setups.
9. Converting Hessian to Positive Definite: For example one can modify the Hessian by adding a correction matrix $B_n$ so as to make $\mathbb{H}f(\vec{x}_n) + B_n$ positive definite. On approach is to diagonalize $\mathbb{H}f(\vec{x}_n)$ and choose $B_n$ so that $\mathbb{H}f(\vec{x}_n) + B_n$ has the same eigenvectors as $\mathbb{H}f(\vec{x}_n)$ but with each negative eigenvalue replaced by

$$\epsilon > 0$$

10. Approach used by Levenberg-Marquardt: An approach exploited by the Levenberg-Marquardt algorithm – which uses an approximate Hessian – is to add a scaled identity matrix $\mu\mathbb{I}$ to the Hessian, with the scale adjusted at every iteration as needed.
11. Comparison with Gradient Descent Approach: For large $\mu$ and small Hessian, the iterations behave like gradient descent with a step size $\frac{1}{\mu}$. This results in slower but more reliable convergence when the Hessian doesn't provide useful information.

## Wolf Conditions

1. Motivation, Rationale, and Primary Purpose: In the unconstrained minimization problem setting, the **Wolf Conditions** are a set of inequalities for performing *inexact* line search, especially in quasi-Newton methods (Wolf (1969, 1971)).
2. Problem Setup - Function Extremization: In these methods the idea is to find

$$\underset{\vec{x}}{min}\, f(\vec{x})$$

for some smooth

$$f: \mathbb{R}^n \longrightarrow \mathbb{R}$$

Each step often involves approximately solving the sub-problem

$$\underset{\alpha}{min}\, f(\vec{x}_k + \alpha\vec{p}_k)$$

where $\vec{x}_k$ is the current best guess

$$\vec{p}_k \in \mathbb{R}^n$$

is a search direction, and

$$\alpha \in \mathbb{R}$$

is the step length.

3. Application of the Wolfe Conditions: The inexact line search provides an efficient way of computing an acceptable step length $\alpha$ that reduces the objective function sufficiently rather than minimizing the objective function over

$$\alpha \in \mathbb{R}^+$$

exactly. A line search algorithm can use the Wolfe conditions as a requirement for any guessed $\alpha$ before finding a new search direction $\vec{p}_k$.

## Armijo Rule and Curvature Condition

1. The Inequalities of Wolfe Condition: A step length $\alpha_k$ is said to satisfy the *Wolfe conditions*, restricted to the direction $\vec{p}_k$, if the following inequalities hold:

   $$f(\vec{x}_k + \alpha_k \vec{p}_k) \leq f(\vec{x}_k) + c_1 \alpha_k \vec{p}_k \cdot \vec{\nabla} f(\vec{x}_k)$$

   $$\vec{p}_k \cdot \vec{\nabla} f(\vec{x}_k + \alpha_k \vec{p}_k) \geq c_2 \vec{p}_k \cdot \vec{\nabla} f(\vec{x}_k)$$

   with

   $$0 < c_1 < c_2 < 1$$

   In examining the second condition once recalls that to ensure $\vec{p}_k$ is a descent direction one has

   $$\vec{p}_k \cdot \vec{\nabla} f(\vec{x}_k) < 0$$

2. Choices for $c_1$ and $c_2$: $c_1$ is usually chosen quite small while $c_2$ is much larger. Nocedal and Wright (2006) provide example values of

   $$c_1 = 10^{-4}$$

   and

   $$c_2 = 0.9$$

   for Newton or quasi-Newton methods, and

   $$c_2 = 0.1$$

for the non-linear conjugate gradient method. The first inequality is known as the **Armijo Rule** (Armijo (1966)) and ensures that the step length $\alpha_k$ decreases $f$ sufficiently. The second inequality is called the **Curvature Condition**; it ensures that the slope has reduced sufficiently.

3. Strong Wolfe Conditions on Curvature: Denoting by $\phi$ a univariate function restricted to the direction $\vec{p}_k$ as

$$\phi(\alpha) = f(\vec{x}_k + \alpha_k \vec{p}_k)$$

there are situations where the Wolfe conditions can result in a value for the step length that is not close to a minimizer of $\phi$. If one modifies the curvature condition to

$$\left|\vec{p}_k \cdot \vec{\nabla} f(\vec{x}_k + \alpha_k \vec{p}_k)\right| \leq c_2 \left|\vec{p}_k \cdot \vec{\nabla} f(\vec{x}_k)\right|$$

then the curvature condition along with the Armijo rule form the so-called **strong Wolfe conditions**, and force $\alpha_k$ to lie close to a critical point of $\phi$.

## Rationale for the Wolfe Conditions

1. Convergence of the Gradient to Zero: The principal reason for imposing the Wolfe conditions in an optimization algorithm where

$$\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{p}_k$$

is to ensure the convergence of the gradient to zero. In particular if the cosine of the angle between $\vec{p}_k$ and the gradient

$$\cos\theta = \frac{\vec{\nabla} f(\vec{x}_k) \cdot \vec{p}_k}{\left\|\vec{\nabla} f(\vec{x}_k)\right\| \|\vec{p}_k\|}$$

is bounded away from zero, and Armijo rule and curvature conditions (modified) hold,

$$\vec{\nabla} f(\vec{x}_k) \longrightarrow 0$$

2. <u>Positive Definiteness of Update Matrix</u>: An additional motivation, in the case of a quasi-Newton method, is that if

$$\vec{p}_k = {B_k}^{-1} \vec{\nabla} f(\vec{x}_k)$$

where the matrix $B_k$ is updated by the BFGS or the DFP formula, then if $B_k$ is positive, the (modified) curvature condition implies that $B_{k+1}$ is also positive definite.

## References

- Armijo, L. (1966): Minimization of Functions having Lipschitz Continuous First Derivatives *Pacific Journal of Mathematics* **16 (1)** 1-3.
- Newton's Method in Optimization (Wiki).
- Nocedal, J., and S. Wright (2006): *Numerical Optimization 2nd Edition* **Springer**.
- Wolfe, P. (1969): Convergence Conditions for Ascent Methods *SIAM Review* **11 (2)** 226-235.
- Wolfe, P. (1971): Convergence Conditions for Ascent Methods II: Some Corrections *SIAM Review* **13 (2)** 185-188.
- Wolfe Conditions (Wiki).

# Constrained Optimization

## Constrained Optimization – Definition and Description

1. Constrained Optimization Definition – Variable Constraints: In mathematical optimization **constrained optimization** is the process of optimizing an objective function with respect to some variables in the presence of constraints on those variables.
2. Constrained Optimization Definition - Objective Function: The objective function is either a cost function or an energy function that is to be minimized, or a reward function or a utility function that has to be maximized.
3. Hard vs Soft Variable Constraints: Constraints can either be *hard constraints* that set conditions on variables required to be satisfied, or *soft constraints* that have some variable values that are penalized in the objective function if – and based on the extent that – the conditions on the variables are not satisfied (Constrained Optimization (Wiki)).

## General Form

1. Constrained Optimization Problem – Mathematical Specification: A general constrained minimization problem may be written as follows:

$$\underset{x}{min}\, f(x)$$

subject to

$$g_i(x) = c_i$$

for

$$i = 1, \cdots, n$$

equality constraints and

$$h_j(x) \geq d_j$$

for

$$j = 1, \cdots, m$$

inequality constraints where

$$g_i(x) = c_i$$

for

$$i = 1, \cdots, n$$

and

$$h_j(x) \geq d_j$$

for

$$j = 1, \cdots, m$$

are the constraints to be satisfied; these are called the hard constraints.

2. Constrained Optimizers - Soft Objective Function: In some problems, often called *constraint optimization problems*, the objective function is a sum of the cost functions, each of which penalized the extent if any to which a soft constraint – a constraint that is preferred but not required to be satisfied - is violated.

## Solution Methods

1. Adaptation from Unconstrained Algorithms: Many unconstrained optimization algorithms can be adapted to the constrained case, often via the use of a penalty method. However, the search steps taken by the unconstrained method may be unacceptable for the constrained problem, leading to a lack of convergence. This is referred to as the Maratos effect (Sun and Yua (2010)).
2. Equality Constraints Lagrange Multiplier Technique: If the constrained problem has only equality constraints the method of Lagrange multipliers can be used to convert it to an unconstrained problem whose number of variables is the original number of variables plus the number of constraints.
3. Equality Constraints - Cross Variable Substitution: Alternatively if the constraints are all equality constraints and are all linear they can be solved for some of the variables in terms of the others, and the former can be substituted out of the objective function, leaving an unconstrained problem in a smaller number of variables.
4. Inequality Constraints - Conditions on Variables: With inequality constraints the problem can be characterized in terms of Geometric Optimality Conditions, Fritz-John Conditions, and Karush-Kuhn-Tucker Conditions in which simple problems may be directly solvable.
5. Linear Programming – Simplex/Interior Point: If the objective function and all of the hard constraints are linear, then the problem is a linear programming one. This can be solved by the Simplex method, which usually works in polynomial time in the

problem size but is not guaranteed to, or by interior point methods, which are guaranteed to work in polynomial time.

6. Quadratic Programming - Objective Function Constraints: If all the hard constraints are linear but the objective function is quadratic the problem is a quadratic programming problem.
7. Quadratic Programming Convex Objective Function: Quadratic Programming problems can be solved by the ellipsoid method in polynomial time if the objective function is convex; otherwise the problem is NP hard.

## Constraint Optimization: Branch and Bound

1. Idea behind Branch and Bound: Constraint optimization can be solved by branch-and-bound algorithms. These are back-tracking algorithms that store the cost of the best solution found during execution and use it eventually to avoid part of the search.
2. Back Tracking vs. Solution Extension: More precisely whenever the algorithm encounters a partial solution that cannot be extended to form a solution with better cost than the best stored cost, the algorithm back tracks instead of trying to extend this solution.
3. Efficient of Back Tracking Algorithms: Assuming that the cost is to be minimized the efficiency of these algorithms depends on how the cost can be obtained from extending a partial solution is evaluated. Indeed if the algorithms can back track from a partial solution, part of the search can be skipped. The lower the estimated cost the better the algorithm, as a lower estimated cost is more likely to be lower than the best cost of the solution found so far.
4. Algorithm Lower and Upper Bounds: On the other hand this estimated cost cannot be lower than the effective cost obtained by extending the solution, as otherwise the algorithm could backtrack while a solution better than the best found so far exists. As a result the algorithm requires an upper bound on the cost that can be obtained by extending a partial solution and this upper bound should be as small as possible.

5. Hansen's Branch-and-Bound Variation: A variation of the above method called Hansen's method uses interval methods (Leader (2004)). It inherently implements rectangular constraints.

## Branch-and-Bound: First-Choice Bounding Conditions

1. Separately treating each Soft Constraint: One way of evaluating this upper bound for a partial solution is to consider each soft constraint separately. For each soft constraint the maximal possible value for any assignment to the unassigned variables is assumed. The sum of these variables is an upper bound because the soft constraints cannot assume a higher value.
2. Exact Nature of Upper Bound: It is exact because the maximal nature of soft constraints may derive from different evaluations: a soft constraint may be maximal for

$$x = a$$

while another soft constraint is maximal for

$$x = b$$

## Branch-and-Bound Russian Doll Search

1. Branch-and-Bound Sub-problems: This method runs a branch-and-bound algorithm on $n$ problems where $n$ is the number of variables (Verfaillie, Lemaitre, and Schiex (1996)). Each such sub-problem is the sub-problem obtained by dropping the sequence $x_1, \cdots, x_i$ from the original problem along with the constraints containing them.

2. Sub-problem Cost Upper Bound: After the problem on variables $x_{i+1}, \cdots, x_n$ is solved its optimal cost can be used as an upper bound while solving the other problems.
3. Assigned/Unassigned Variables Sub-problem: In particular the cost of estimating a solution having $x_{i+1}, \cdots, x_n$ is added to the cost that derives from the evaluated variables. Virtually this corresponds to ignoring the evaluate variables and solving the problem on the unassigned ones, except that the latter problem has been already solved.
4. Sub-problem Total Cost Update: More precisely the cost of the constraints containing both the assigned and the unassigned variables is estimated as above, or using another arbitrary method; the cost of soft constraints using unassigned variables is instead estimated using the optimal solution of the corresponding problem, which is already known at this point.
5. Similarities with Dynamic Programming Approach: Like Dynamic Programming Russian Doll Search solves the sub-problems in order to solve the whole problem.
6. Differences with Dynamic Programming Approach: However, whereas Dynamic Programming directly combines the results obtained on the sub-problems to get the result of the whole program, the Russian Doll Search only uses them as bounds during the search.

## Branch-and-Bound – Bucket Elimination

1. Elimination of a Specified Variable: The bucket elimination algorithm can be used for constraint optimization. A given variable can be removed from the problem by replacing all the soft constraints containing it with a new soft constraint.
2. Removed Variable Constraint-Cost Expression: The cost of the constraint expression is estimated assuming the maximal objective function value for each of the removed variable. Formally if $x$ is the variable to be removed, $C_1, \cdots, C_n$ are the constraints containing it, and $y_1, \cdots, y_m$ are the variables containing $x$, the new soft constraint is defined by

$$C(y_1 = a_1, \cdots, y_n = a_n) = \underset{a}{max} \sum_i C_i(x = a, y_1 = a_1, \cdots, y_n = a_n)$$

3. Bucket of all Variable Constraints: Bucket elimination works with an arbitrary ordering of the variables. Every variable is associated with a bucket of constraints; the bucket of a variable contains all the constraints having the variable has the highest in the order.
4. Order of the Bucket Elimination: Bucket elimination proceeds from the last variable to the first. For each variable, all constraints of the bucket are replaced as above to remove the variable. The resulting constraint is then placed in the appropriate bucket.

## References

- Constrained Optimization (Wiki).
- Leader, J. J. (2004): *Numerical Analysis and Scientific Computation* **Addison Wesley**.
- Sun, W., and Y. X. Yua (2010): *Optimization Theory and Methods: Non-linear Programming* **Springer**.
- Verfaillie, G., M. Lemaitre, and T. Schiex (1996): Russian Doll Search for solving Constraint Optimization Problems *Proceedings of the 13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference* **Portland** 181-187.

# Lagrange Multipliers

## Motivation, Definition, and Problem Formulation

1. Purpose behind the Lagrange Methodology: In mathematical optimization the **method of Lagrange multipliers** (Lagrange (1811)) is a strategy for finding the local maxima and minima subject to equality constraints (Lagrange Multiplier (Wiki)).
2. Optimization Problem Mathematical Setup: Consider the optimization problem of maximizing $f(x, y)$ subject to

   $$g(x, y) = 0$$

   A new variable $\lambda$ is introduced (this is called the Lagrange multiplier) and the Lagrange function (or Lagrangian) defined by

   $$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda g(x, y)$$

   is studied. Here the term $\lambda$ may be either added or subtracted.
3. Lagrange Multipliers as Stationary Points: If $f(x_0, y_0)$ is the maximum of $f(x, y)$ for the original constrained problem, then there exists a $\lambda_0$ such that $(x_0, y_0, \lambda_0)$ is a stationary point for the Lagrangian function (stationary points are those where the partial derivatives of $\mathcal{L}$ are zero).
4. Necessary Conditions for Constrained Optimality: However not all stationary points yield a solution to the original problem. Thus, the method of Lagrange multipliers yields necessary conditions for optimality in constrained problems (Hiriart-Urruty and Lemarechal (1993), Bertsekas (1999), Vapnyarskii (2001), Lemarechal (2001), Lasdon (2002)).

5. Sufficient Conditions for Constrained Optimality: Sufficient conditions for a maximum or a minimum also exist. Sufficient conditions for a constrained local maximum or a minimum can be stated in terms of a sequence of principal minors, i.e., determinants of the upper-left-justified sub-matrices, of the bordered Hessian matrix of the second derivatives of the Lagrangian expression (Chiang (1984)).

## Introduction, Background, and Overview

1. Advantage of the Lagrange Multiplier Formulation: One of the most common problems in calculus is that of finding the extrema of a function, but it is often difficult to find a closed form for the function being extremized. Such difficulties often arise when one wishes to extremize the function subject to fixed outside constraints or conditions. The method of Lagrange multipliers is a powerful tool for solving this class of problems without the need to explicitly solve for the conditions and use them to eliminate the extra variables.
2. Intuition behind the Lagrange Multiplier Methodology: Consider the 2D problem introduced above: maximize $f(x, y)$ subject to

$$g(x, y) = 0$$

The method of Lagrange multipliers relies on the intuition that at the maximum $f(x, y)$ cannot be increasing in the direction of any neighboring point where

$$g(x, y) = 0$$

If it did, one could walk along

$$g(x, y) = 0$$

to get higher, meaning that the starting point wasn't actually the maximum.

3. Function Realization/Constraint Value Contours: The contours of $f$ given by

   $$f(x, y) = d$$

   for various values of $d$ as well as the contours of $g$ given by

   $$g(x, y) = 0$$

   may be visualized as being parallel/tangential to each other.

4. Invariance of $f(x, y)$ along the Constraint: Suppose one walks along the contour with

   $$g(x, y) = 0$$

   One is interested in funding points along $f(x, y)$ that do not change along the walk since these points may be maxima. There are two ways this can happen. First one could be walking along the contour line of $f(x, y)$ since by definition $f(x, y)$ does not change along the contour line. This would mean that the contour lines of $f(x, y)$ and $g(x, y)$ are parallel here. The second possibility is that one has reached a "level" part of $f(x, y)$ meaning that $f(x, y)$ cannot change in any direction.

5. Constraint/Objective Function Gradient Scaling: To check for the first possibility, one notices that the gradient of a function is perpendicular to its contour lines, and the contour lines of $f(x, y)$ and $g(x, y)$ are parallel if and only if the gradients of $f(x, y)$ and $g(x, y)$ are parallel. Thus one wants points $(x, y)$ where

   $$g(x, y) = 0$$

   and

   $$\vec{\nabla}_{x,y} f(x, y) = -\lambda \vec{\nabla}_{x,y} g(x, y)$$

for some $\lambda$ where

$$\vec{\nabla}_{x,y} f(x,y) = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

$$\vec{\nabla}_{x,y} g(x,y) = \left[\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right]$$

are the respective gradients. The constant $\lambda$ is required because although the two gradient vectors are parallel, the magnitudes of the gradient vectors are generally not equal. This constant is called the Lagrange multiplier (the negative sign is a convention).

6. The “Level” Plateau of $f$: This method also addresses the second possibility; if $f$ is level then its gradient is zero, and setting

$$\lambda = 0$$

is a solution regardless of $g$.

7. Formulation of the Lagrangian Function: To incorporate these conditions into one equation one introduces an auxiliary function

$$\mathcal{L}(x,y,\lambda) = f(x,y) - \lambda g(x,y)$$

and solves for

$$\vec{\nabla}_{x,y,\lambda} \mathcal{L}(x,y,\lambda) = 0$$

This is the method of Lagrange multipliers. Note that

$$\vec{\nabla}_{\lambda}\mathcal{L}(x, y, \lambda) = 0$$

implies

$$g(x, y) = 0$$

To summarize

$$\vec{\nabla}_{x,y,\lambda}\mathcal{L}(x, y, \lambda) = 0$$

results in

$$\vec{\nabla}_{x,y}f(x, y) = -\lambda\vec{\nabla}_{x,y}g(x, y)$$

and

$$g(x, y) = 0$$

The constrained extrema of $f$ are the *critical points* but they are not necessarily the *local extrema* of $\mathcal{L}$.

8. Alternate Formulations for the Lagrangian: One may re-formulate the Lagrangian as a Hamiltonian, in which case the solutions are the local minima for the Hamiltonian. This is done in optimal control theory in the form of Pontryagin's minimum principle.
9. Non-Extremum Solutions to Lagrangian: The fact that the solutions to the Lagrangian are not necessarily the extrema also poses difficulties for numerical optimization. This can be addressed by computing the *magnitude* of the gradient, as the zeroes of the magnitude are necessarily the local minima, as illustrated later in the part on numerical optimization.

## Handling Multiple Constraints

1. Notation Used for Multiple Constraints: Throughout this section the independent variables are denoted $x_1, \cdots, x_N$ and, as a group denoted as

$$p = (x_1, \cdots, x_N)$$

Also the function being analyzed will be denoted $f(p)$ and the constraints will be represented by the equations

$$\begin{gathered} g_1(p) = 0 \\ \vdots \\ g_M(p) = 0 \end{gathered}$$

2. Basic Idea behind the Constraints: The basic idea remains essentially the same: if we consider only satisfy the constraints (i.e., are *in* the constraints) then a point $\big(p, f(p)\big)$ is a stationary point (i.e., a point in a *flat* region) of $f$ if and only if the constraints at that point do not allow movement in a direction where $f$ changes value. Once the stationary points are located further tests are needed to see if it is a minimum, a maximum, or just a stationary point that is neither.
3. Multi-dimensional Invariance of $f$: Consider the level set of $f$ at $\big(p, f(p)\big)$. Let the set of vectors $\{v_L\}$ contain the directions in which one can move and still remain in the same level set, the directions where the value of $f$ does not change (i.e., the change equals zero). For every vector $v$ in $\{v_L\}$ the following relation must hold:

$$\frac{\partial f}{\partial x_1} v_1 + \cdots + \frac{\partial f}{\partial x_N} v_N = 0$$

where the notation $v_k$ above refers to the $k^{th}$ component of the vector $v$.

4. Invocation of the Gradient of $f$: The equation above can be re-written in a more compact geometric form that helps the intuition:

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_N} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} = 0$$

i.e.

$$\vec{\nabla} f^T \cdot \vec{v} = 0$$

This makes it clear that if one is at $p$ then *all* directions from this point that do *not* change the value of $f$ *must be perpendicular to* $\vec{\nabla} f(p)$ (the gradient of $f$ at $p$).

5. Usage of the Constraint Gradient: Each constraint limits the directions that one can move from a particular point and still satisfy the constraint. One uses the same set of procedures above to look for a set of vectors $\{v_C\}$ that contain the directions in which can move and still satisfy the constraint. As above, for every vector $v$ in $\{v_C\}$ the following relation must hold:

$$\frac{\partial g}{\partial x_1} v_1 + \cdots + \frac{\partial g}{\partial x_N} v_N = 0$$

i.e.

$$\vec{\nabla} g^T \cdot \vec{v} = 0$$

From this it can be seen that at point $p$ all directions from this point that will satisfy the constraint must be proportional to $\vec{\nabla} g(p)$.

6. Lagrange Multiplier Method Formal Definition: A point on $f$ is a constrained stationary point if and only if the direction that changes $f$ violates at least one of the constraints, i.e., it has no “component” in the legal space perpendicular to $\vec{\nabla} g(p)$.

Mathematically this means that the gradient of $f$ at this constrained stationary point is perpendicular to the space spanned by the set of vectors $\{v_C\}$ which in turn is perpendicular to the gradients of the constraints $g$.

7. Single Constraint $f/g$ Gradients: For a single constraint the above statement says that at stationary points the direction that changes $f$ is the same as that violates the constraint. To determine if two vectors are in the same direction, note that if two vectors start from the same point then open vector can always reach the other by changing its length and/or flipping or the opposite way along the same direction line. This requires that

$$\vec{\nabla}f(p) = \lambda\vec{\nabla}g(p)$$

implying that

$$\vec{\nabla}f(p) - \lambda\vec{\nabla}g(p) = 0$$

8. Single Constraint Case - Compact Formulation: On adding another simultaneous equation to guarantee that this test is performed only at the point that satisfies the constraint two simultaneous equations result, which when solved identify all the constrained stationary points.

$$g(p) = 0$$

implies that $p$ satisfies the constraint, thus

$$\vec{\nabla}f(p) - \lambda\vec{\nabla}g(p) = 0$$

is a stationary point.

9. Single Constraint Case Expanded Formulation: Fully expanded there are simultaneous equations that need to be solved for $N + 1$ variables, which are $x_1, \cdots, x_N$ and $\lambda$:

$$g(x_1, \cdots, x_N) = 0$$

$$\frac{\partial f(x_1, \cdots, x_N)}{\partial x_1} - \lambda \frac{\partial g(x_1, \cdots, x_N)}{\partial x_1} = 0$$

$$\vdots$$

$$\frac{\partial f(x_1, \cdots, x_N)}{\partial x_N} - \lambda \frac{\partial g(x_1, \cdots, x_N)}{\partial x_N} = 0$$

10. Multiple Constraints and their Gradients: For more than one constraint a similar reasoning applies. Each gradient function $g$ has a space of allowable directions at $p$ – the space of vectors perpendicular to $\vec{\nabla} g(p)$. The set of directions allowed by all constraints is thus the space of directions perpendicular to all of the constraint gradients. Denoting the space of allowable moves by $A$ and the span of gradients by $S$, using the discussion above

$$A = S^{\perp}$$

the space of vectors perpendicular to every element in $S$.

11. Multiple Constraints Case Gradient Formulation: If $p$ is an optimum any element not perpendicular to $\vec{\nabla} f(p)$ is not an allowable direction. One can show that this implies

$$\vec{\nabla} f(p) \in A^{\perp} = S$$

Thus there are scalars $\lambda_1, \cdots, \lambda_M$ such that

$$\vec{\nabla} f(p) = \sum_{k=1}^{M} \lambda_k \vec{\nabla} g_k(p)$$

implies

$$\vec{\nabla} f(p) - \sum_{k=1}^{M} \lambda_k \vec{\nabla} g_k(p) = 0$$

As before the simultaneous equations are added to guarantee that this test is performed only at a point that satisfies every constraint, and the resulting equations, when satisfied, identify all the constrained stationary points.

$$g_1(p) = \cdots = g_M(p) = 0$$

implies that $p$ satisfies all constraints; further

$$\vec{\nabla} f(p) - \sum_{k=1}^{M} \lambda_k \vec{\nabla} g_k(p) = 0$$

indicates that $p$ is a stationary point.

12. Multiple Gradient Case Lagrangian Formulation: The method is complete now from the standpoint of finding the stationary points, but these equations can be condensed into a more succinct and elegant form. The equations above look like partial derivatives of a larger scalar function $\mathcal{L}$ that takes all the $x_1, \cdots, x_N$ and all the $\lambda_1, \cdots, \lambda_M$ as inputs. Setting every equation to zero is exactly what one would have to do to solve for the *unconstrained* stationary points of the larger function. Finally the larger function $\mathcal{L}$ with partial derivatives that are exactly the ones that we require can be constructed very simply as

$$\mathcal{L}(x_1, \cdots, x_N, \lambda_1, \cdots, \lambda_M) = f(x_1, \cdots, x_N) - \sum_{k=1}^{M} \lambda_k \vec{\nabla} g_k(x_1, \cdots, x_N)$$

Solving the above equation for its *unconstrained* stationary points generates exactly the same stationary points as solving for the *constrained* stationary points of $f$ under the constraints $g_1, \cdots, g_M$

13. The Method of Lagrange Multipliers: In Lagrange's honor the above function is called a *Lagrangian*, the scalar multipliers $\lambda_1, \cdots, \lambda_M$ are called *Lagrange Multipliers*, and the method itself is called *The Method of Lagrange Multipliers*.
14. The Karush-Kuhn-Tucker Generalization: The method of Lagrange multipliers is generalized by the Karush-Kuhn-Tucker conditions, which also takes into account inequality constraints of the form

$$h(\vec{x}) \leq c$$

## Modern Formulation via Differentiable Manifolds

1. Local Extrema of $\mathbb{R}^d \rightarrow \mathbb{R}^1$: Funding the local maxima of a function

$$f: \mathbb{U} \rightarrow \mathbb{R}$$

where $\mathbb{U}$ is an open subset of $\mathbb{R}^d$ is done by finding all points $x \in \mathbb{U}$ such that

$$\mathbb{D}_x f = 0$$

and then checking whether all of the eigenvalues of the Hessian $\mathbb{H}_x f$ are negative. Setting

$$\mathbb{D}_x f = 0$$

is a non-linear problem and in general arbitrarily difficult. After funding the critical points checking for the eigenvalues is a linear problem and thus easy.

2. Extrema under the Level-Set Constraint: When

$$g \colon \mathbb{R}^d \to \mathbb{R}^1$$

is a smooth function such that

$$\mathbb{D}_x g \neq 0$$

for all $x$ in the level set

$$g(x) = c$$

then $g^{-1}(c)$ becomes an $n-1$ dimensional smooth manifold $\mathbb{M}$ by the level set theorem. Funding local maxima is by definition a local problem, so it can be done on the local charts of $\mathbb{M}$ after funding a diffeomorphism

$$\varphi \colon \mathbb{V} \to \mathbb{R}^{n-1}$$

from an open subset of

$$\mathbb{V} \subset \mathbb{M}$$

onto an open subset

$$\mathbb{U} \subset \mathbb{R}^{n-1}$$

and thus one can apply the algorithm in the previous part to the function

$$f' = f \circ \varphi^{-1} \colon \mathbb{U} \to \mathbb{R}$$

3. Approach of Lagrange Multiplier Method: While the above idea sounds good it is difficult to compute $\varphi^{-1}$ in practice. *The entire method of Lagrange multipliers reduces to skipping that step and finding the zeroes of $\mathbb{D}_x f'$ directly*. It follows from the construction in the level set theorem that $\mathbb{D}_x \varphi^{-1}$ is the inclusion map

   $$\ker \mathbb{D}_{\varphi^{-1}(x)} g \subseteq \mathbb{R}^n$$

   Therefore

   $$0 = \mathbb{D}_x f' = \mathbb{D}_x (f \circ \varphi^{-1}) = \mathbb{D}_{\varphi^{-1}(x)} f \circ \mathbb{D}_x \varphi^{-1}$$

   if and only if

   $$\ker \mathbb{D}_y g \subseteq \ker \mathbb{D}_y f$$

   from writing $y$ for $\varphi^{-1}(x)$

4. Existence of the Linear Map: By the first isomorphism theorem this is true if and only if there exists a linear map

   $$\mathcal{L} \colon \mathbb{R} \to \mathbb{R}$$

   such that

   $$\mathcal{L} \circ \mathbb{D}_y g = \mathbb{D}_y f$$

   As a linear map one must have that

$$\mathcal{L}(x) = \lambda x$$

for a fixed

$$\lambda \in \mathbb{R}$$

Therefore funding the critical point of $f'$ is equivalent to solving the system of equations

$$\lambda \mathbb{D}_y g = \mathbb{D}_y f$$

$$g(y) = c$$

in the variables

$$y \in \mathbb{R}^{n-1}$$

and

$$\lambda \in \mathbb{R}$$

This is in general a non-linear system of $n$ equations and $n$ unknowns.

5. Multiple Constraints Surjective Linear Map: In the case of general constraints one works with

$$g: \mathbb{R}^n \to \mathbb{R}^m$$

and replaces the condition

$$\mathbb{D}_x g \neq 0$$

for all

$$x \in g^{-1}(c)$$

with the requirement that $\mathbb{D}_x g$ be surjective at all such points. In this case $\mathcal{L}$ will be a linear map $\mathbb{R}^m \to \mathbb{R}$ i.e., a row vector with $m$ entries.

## Interpretation of the Lagrange Multipliers

1. As a Rate of Change Quantity: Often the Lagrange multipliers have an interpretation as some quantity of interest. For example if the Lagrangian expression is

   $$\begin{aligned}\mathcal{L}(x_1, \cdots, x_N, \lambda_1, \cdots, \lambda_M) \\ &= f(x_1, \cdots, x_N) + \lambda_1[c_1 - g_1(x_1, \cdots, x_N)] + \cdots \\ &+ \lambda_M[c_M - g(x_1, \cdots, x_N)]\end{aligned}$$

   then

   $$\frac{\partial \mathcal{L}}{\partial c_k} = \lambda_k$$

   So $\lambda_k$ is the rate of change of the quantity being optimized as a function of the constraint variable.
2. Examples of Lagrange Multiplier Roles: As examples, in Lagrangian mechanics the equations of motion are derived by finding stationary points of action, i.e., the time integral of the difference between the potential and the kinetic energies. Thus the force on a particle due to a scalar potential

$$\vec{F} = -\vec{\nabla} V$$

can be interpreted as a Lagrange multiplier determining the change in action – transfer of potential to kinetic energy – following a variation in the particle's constrained trajectory. In control theory this formulated instead as co-state equations.

3. Marginal Effect of the Constraint: Moreover by the control theorem the optimal value of a Lagrange multiplier has an interpretation as the marginal effect of the corresponding constraint constant upon on the optimal attainable value of the original objective function. Denoting the optimum with an asterisk it can be shown that

$$\frac{\partial f\left(x_1{}^*(c_1, \cdots, c_M), \cdots, x_N{}^*(c_1, \cdots, c_M)\right)}{\partial c_k} = \lambda_k{}^*$$

4. Lagrange Multiplier Usage in Economics: For example in economics the optimal profit to a player is calculated subject to a constrained space of actions, where the Lagrange multiplier is the change in the optimal value of the objective function (profit) due to the relaxation of a given constraint – e.g., through a change in the income. In such a context $\lambda^*$ is the marginal cost of the constraint, and is referred to as the shadow price.

## Lagrange Application: Maximal Information Entropy

1. Information Entropy Maximization – Problem Setup: Suppose on wishes to find the discrete probability distribution on the points $\{p_1, \cdots, p_n\}$ with maximal information entropy. This is the same as saying that one wishes to find the least biased probability on the points $\{p_1, \cdots, p_n\}$. In other words one wishes to maximize the Shannon entropy equation

$$f(p_1, \cdots, p_n) = -\sum_{j=1}^{n} p_j \log p_j$$

2. Cumulative Discrete Probability Normalization Constraint: For this to be a probability distribution the sum of the probabilities $p_j$ at each point $x_i$ must equal 1, so the constraint becomes

$$g(p_1, \cdots, p_n) = \sum_{j=1}^{n} p_j \log p_j$$

3. Application of the Lagrange Multipliers: Using Lagrange multipliers to find the point of maximum entropy $\vec{p}^*$ across all discrete probability distributions $\vec{p}$ on $\{x_1, \cdots, x_n\}$ requires that

$$\frac{\partial}{\partial \vec{p}} [f + \lambda(g - 1)] \bigg|_{\vec{p}=\vec{p}^*} = 0$$

which gives a system of $n$ equations with indices

$$k = 1, \cdots, n$$

such that

$$\frac{\partial}{\partial p_k} \left[ -\sum_{j=1}^{n} p_j \log p_j + \lambda \left( \sum_{j=1}^{n} p_j - 1 \right) \right] \Bigg|_{\vec{p}=\vec{p}^*} = 0$$

4. Solution to the Discrete Probability: Carrying out the differentiation on these $n$ equations one gets

$$-[1 + \log p_k{}^*] + \lambda = 0$$

This shows that all $p_k{}^*$ are the same because they depend only on $\lambda$. By using the constraint

$$\sum_{j=1}^{n} p_j = 1$$

one finds that

$$p_k{}^* = \frac{1}{n}$$

Hence the uniform distribution is the distribution with the greatest entropy, using distributions on $n$ points.

## Lagrange Application: Numerical Optimization Techniques

1. Local Minima vs. Saddle Points: The critical points of the Lagrangian occur at saddle points rather than the local minima or maxima (Heath (2005)). Unfortunately many numerical optimization techniques such as hill climbing, gradient descent, some of the quasi-Newton methods, among others, are designed to find local minima (or maxima) and not the saddle points.
2. Conversion to an Extremization Problem: For this reason one must modify the formulation to ensure that this is a minimization problem – for example by extremizing the square of the gradient of the Lagrangian – or use an optimization technique that finds stationary points and not necessarily extrema, e.g., such as Newton's method without an extremum seeking line search.

3. “Saddle” Critical Points - An Example: As a simple example consider the problem of finding the value of $x$ that minimizes the function

$$f(x) = x^2$$

constrained such that

$$x^2 = 1$$

Clearly this problem is pathological because there are only two values that satisfy the constraint, but it is useful for illustration purposes because the corresponding unconstrained function can be visualized in three dimensions.

4. Application of the Lagrange Multipliers: Using Lagrange multipliers this problem can be converted into an unconstrained optimization problem

$$\mathcal{L}(x, \lambda) = x^2 + \lambda(x^2 - 1)$$

The two critical points occur at the saddle points where

$$x = +1$$

and

$$x = -1$$

5. Transformation to Local Extremum Problem: In order to solve this problem with a numerical optimization technique one must first transform this problem such that the critical points occur at the local minima. This is done by computing the magnitude of the gradient of the unconstrained optimization problem.

6. <u>Objective Variate/Constraint Multiplier Jacobian</u>: First one computes the partial derivative of the unconstrained problem with respect to each variable;

$$\frac{\partial \mathcal{L}}{\partial x} = 2x + 2\lambda x$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = x^2 - 1$$

If the target function is not easily differentiable the differential with respect to each variable can be approximated using the divided differences technique, i.e.

$$\frac{\partial \mathcal{L}(x, \lambda)}{\partial x} \approx \frac{\mathcal{L}(x + \epsilon, \lambda) - \mathcal{L}(x - \epsilon, \lambda)}{2\epsilon}$$

$$\frac{\partial \mathcal{L}(x, \lambda)}{\partial \lambda} \approx \frac{\mathcal{L}(x, \lambda + \epsilon) - \mathcal{L}(x, \lambda - \epsilon)}{2\epsilon}$$

where $\epsilon$ is a small value.

7. <u>Computing the Magnitude of the Gradient</u>: Next one computes the magnitude of the gradient, which is the square root of the sum of squares of the partial derivatives:

$$h(x, \lambda) = \sqrt{(2x + 2\lambda x)^2 + (x^2 - 1)^2}$$
$$\approx \sqrt{\left[\frac{\mathcal{L}(x + \epsilon, \lambda) - \mathcal{L}(x - \epsilon, \lambda)}{2\epsilon}\right]^2 + \left[\frac{\mathcal{L}(x, \lambda + \epsilon) - \mathcal{L}(x, \lambda - \epsilon)}{2\epsilon}\right]^2}$$

Since the magnitude is always non-negative optimizing over the squared magnitude is equivalent to optimizing over the magnitude. Thus the square root may be omitted from these equations with no expected differences in the results of the optimization.

8. <u>Critical Points as Local Extrema</u>: The critical points of $h$ occur at

$$x = +1$$

and

$$x = -1$$

just as in $\mathcal{L}$. However the critical points in $h$ occur at local minima, so the numerical optimization techniques can be used to find them.

## Lagrange Multipliers – Common Practice Applications

1. Lagrange Multipliers Applied in Economics: Constrained optimization plays a central role in economics. For example, the choice problem for a customer is represented as one of maximizing a utility function subject to a budget constraint. The Lagrange multiplier has an interpretation as the shadow price associated with that constraint – in this instance the marginal utility of the income. Other examples include profit maximization for a firm, along with various macro-economic applications.
2. Lagrange Multipliers in Control Theory: In optimal control theory the Lagrange multipliers are represented as co-state variables and are re-formulated as the minimization of the Hamiltonian, e.g., they are the basis behind the Pontryagin's minimum principle.
3. Lagrange Multipliers in Non-linear Programming: The Lagrange multiplier method has several generalizations. In non-linear programming there are several multiplier rules, e.g., the Caratheodery-John Multiplier Rule and the Convex Multiplier Rule for inequality constraints (Pourciau (1980)).

## References

- Bertsekas, D. P. (1999): *Nonlinear Programming 2nd Edition* **Athena Scientific** Cambridge MA.
- Chiang, A. C. (1984): *Fundamental Methods of Mathematical Economics 3rd Edition* **McGraw-Hill**.
- Heath, M. T. (2005): *Scientific Computing – An Introductory Survey* **McGraw-Hill**.
- Hiriart-Urruty, J. B., and C. Lemarechal (1993): XII Abstract Duality for Practitioners, in: *Convex Analysis and Minimization Algorithms, Volume II: Advanced Theory and Bundle Methods* **Springer-Verlag** Berlin.
- Lagrange, J. L. (1811): *Mecanique Analytique*.
- Lagrange Multiplier (Wiki).
- Lasdon, L. S. (2002): *Optimization Theory for Large Systems* **Dover Publications** Mineola NY.
- Lemarechal, C. (1993): Lagrangian Relaxation, in: *Computational Combinatorial Optimization: Papers from the Spring School held in Schloss Dagstuhl (Editors: M. Junger and D. Naddef)* **Springer-Verlag** Berlin.
- Pourciau, B. H. (1980): Modern Multiplier Rules *American Mathematical Monthly* **87 (6)** 433-452.
- Vapnyarskii, I. B. (2001): Lagrange Multiplier, in: *Encyclopedia of Mathematics (editor: M. Hazewinkel)* **Springer**.

# Karush-Kuhn-Tucker Conditions

## Introduction, Overview, Purpose, and Motivation

1. KKT Conditions - Necessity and Scope: In mathematical optimization the Karush-Kuhn-Tucker (KKT) Conditions – also known as Kuhn-Tucker Conditions – are the first order necessary conditions for a solution in non-linear programming to be optimal provided some regularity conditions are satisfied (Karush-Kuhn-Tucker Conditions (Wiki)).
2. KKT Conditions vs. Lagrange Multipliers: Allowing for inequality constraints the KKT approach to non-linear programming generalizes the method of Lagrange multipliers, which only allows for equality constraints.
3. Mathematical Foundation behind Optimization Algorithms: The system of equations corresponding to the KKT conditions is usually not solved directly except in a few special cases where a closed-form solution may be derived analytically. In general many optimization algorithms can be interpreted as methods for numerically solving the KKT systems of equations (Boyd and van den Berghe (2009)).
4. KKT Conditions - Historical Attribution Revision: The KKT conditions were originally named after Harold W. Kuhn and Albert W. Tucker who first published the conditions in 1951 (Kuhn and Tucker (1951)). Later scholars discovered that then necessary conditions for this problem had been stated by William Karush in his master's thesis (Karush (1939), Kjeldsen (2000)).

## Necessary Conditions for Optimization Problems

1. KKT Statement - Non-linear Optimization Problem: Consider the following non-linear optimization problem: Maximize $f(x)$ subject to

$$g_i(x) \leq 0$$

and

$$h_j(x) = 0$$

where $x$ is the optimization variable, $f$ is the *objective* or the *utility* function, $g_i (i = 1, \cdots, m)$ are the inequality constraint functions, and $h_j (j = 1, \cdots, l)$ are the equality constraint functions. The numbers of the inequality and the equality constraints are denoted $m$ and $l$ respectively.

2. KKT Statement - The Necessary Condition: Suppose that the objective function

$$f: \mathbb{R}^n \rightarrow \mathbb{R}$$

and the constraint functions

$$g_i: \mathbb{R}^n \rightarrow \mathbb{R}$$

and

$$h_j: \mathbb{R}^n \rightarrow \mathbb{R}$$

are continuously differentiable at the point $x^*$. If $x^*$ is a local optimum that satisfies some regularity conditions below, there exist constants $\mu_i (i = 1, \cdots, m)$ and $\lambda_j (j = 1, \cdots, l)$ called KKT multipliers that have the following properties.

3. KKT Statement - Optimal Stationary Conditions: For maximizing $f(x)$

$$\vec{\nabla} f(x^*) = \sum_{i=1}^{m} \mu_i \vec{\nabla} g_i(x^*) + \sum_{j=1}^{l} \lambda_i \vec{\nabla} h_j(x^*)$$

For minimizing $f(x)$

$$-\vec{\nabla} f(x^*) = \sum_{i=1}^{m} \mu_i \vec{\nabla} g_i(x^*) + \sum_{j=1}^{l} \lambda_i \vec{\nabla} h_j(x^*)$$

4. KKT Statement - Primal Feasibility Conditions:

$$g_i(x^*) \leq 0$$

for all

$$i = 1, \cdots, m$$

and

$$h_j(x^*) = 0$$

for all

$$j = 1, \cdots, l$$

5. KKT Statement - Dual Feasibility Conditions:

$$\mu_i \geq 0$$

for all

$$i = 1, \cdots, m$$

6. KKT Statement – Complementary Slackness Conditions:

$$\mu_i g_i(x^*) = 0$$

for all

$$i = 1, \cdots, m$$

7. Reduction to the Lagrange Criterion: In the particular case

$$m = 0$$

where there are no inequality constraints the KKT conditions turn into the Lagrange conditions and the KKT multipliers become the Lagrange multipliers.

8. Non-differentiable Version of KKT: If some of the functions are non-differentiable sub-differentiable versions of the KKT conditions are available (Eustaquio, Karas, and Ribeiro (2008)).

## Regularity Conditions or Constraint Qualifications

1. The KKT Necessary Regularity Conditions: In order for a minimum point $x^*$ to satisfy the above KKT conditions the problem should satisfy some regularity conditions. The most common ones are listed below.
2. Linear Constraint Qualification: If $g_i$ and $h_j$ are affine functions the no other condition is needed to be satisfied.

3. Linear Independence Constraint Qualification (LICQ): The gradients of the active inequality constraints and the gradients of the equality constraints are linearly independent at $x^*$.
4. Mangasarian-Fromovitz Constraint Qualification (MFCQ): The gradients of the active inequality constraints and the gradients of the equality constraints are positive-linearly independent at $x^*$.
5. Constant Rank Constraint Qualification (CRCQ): For each subset of the gradients of the active inequality constraints and the gradients of the active equality constraints the rank at the vicinity of $x^*$ is constant.
6. Constant Positive Linear Dependence Constraint Qualification (CPLD): For each subset of the gradients of the active inequality constraints and the gradients of the equality constraints, if it is positive-linearly dependent at $x^*$, then it is positive-linearly dependent at the vicinity of $x^*$.
7. Quasi-Normality Constraint Qualification (QNCQ): If the gradients of the active inequality constraints and the gradients of the active equality constraints are positive-linearly dependent at $x^*$ with the associated multipliers $\lambda_i$ for equalities and $\mu_j$ for inequalities then there is no sequence

$$x_k \to x^*$$

such that

$$\lambda_i \neq 0$$

implies

$$\lambda_i h_i(x_k) > 0$$

AND

$$\mu_j \neq 0$$

implies

$$\mu_j g_j(x_k) > 0$$

8. Slater Condition: For a convex problem there exists a point $x$ such that

$$h_j(x) = 0$$

and

$$g_i(x) < 0$$

9. Positive Linear Dependency Condition Definition: The set of vectors $\{v_1, \cdots, v_n\}$ is positive linearly dependent if there exists

$$a_1 \geq 0, \cdots, a_n \geq 0$$

not all zero such that

$$a_1 v_1 + \cdots + a_n v_n = 0$$

10. Strength Order of Constraint Qualifications: It can be shown that

$$LICQ \Longrightarrow MFCQ \Longrightarrow CPLD \Longrightarrow QNCQ$$

and

$$LICQ \Longrightarrow CRCQ \Longrightarrow CPLD \Longrightarrow QNCQ$$

– the converses are not true – although MFCQ is not equivalent to CRCQ (Ruszczynski (2006)). In practice weaker constraint qualifications are preferred since they provide stronger optimality conditions.

## Sufficient Conditions

1. The Second Order Sufficiency Conditions: In some cases the necessary conditions are also sufficient for optimality. In general the necessary conditions are not sufficient for optimality and more information is needed, such as the Second Order Sufficiency Conditions (SOSC). For smooth functions SOSC involve the second derivatives, which explains its name.
2. When Necessary Conditions are Sufficient: The necessary conditions are sufficient for optimality of the objective function $f$ of a maximization problem is a concave function, the inequality constraints $g_i$ are continuously differentiable convex functions, and the equality constraints $h_j$ are affine functions.
3. The Type 1 Invex Functions: The broader class of functions in which the KKT conditions guarantee global optimality are the so-called Type 1 **invex functions** (Martin (1985), Hanson (1999)).
4. Second Order Sufficiency Conditions Formulation: For smooth non-linear optimization problems the second order sufficiency conditions are given as follows. Consider $x^*$, $\lambda^*$, and $\mu^*$ that find a local minimum using the Karush-Kuhn-Tucker conditions above. With $\mu^*$ such that the strict complementary condition is held at $x^*$, i.e.

$$\mu > 0$$

for all

$$s \neq 0$$

such that

$$\left[\frac{\partial g(x^*)}{\partial x}, \frac{\partial h(x^*)}{\partial x}\right] s = 0$$

the following equation must hold:

$$s^T {\nabla_{xx}}^2 \mathcal{L}(x^*, \lambda^*, \mu^*) s > 0$$

If the above condition is strictly met, the function is a strict constrained local minimum.

## KKT Conditions Application - Economics

1. KKT Models as Theoretical Tools: Often in mathematical economics the KKT approach is used in theoretical models in order to obtain qualitative results.
2. Minimum Profit Constraint Revenue Maximization: Consider a firm that maximizes its sales revenue subject to a minimum profit constraint. Letting $Q$ be the quantity of output produced – this is to be chosen - $R(Q)$ be the sales revenue with a positive first derivative and with a zero value at zero output, $C(Q)$ be the production cost with a positive first derivative and with a non-negative value at zero output, and $G_{min}$ be the positive minimal acceptable level of profit, then the problem is a meaningful one if the revenue function levels off so it is eventually less steep than the cost function.
3. Application of the KKT Condition: The problem can be expressed in the following minimization form: Minimize $-R(Q)$ subject to

$$G_{min} \leq R(Q) - C(Q)$$

$$Q \geq 0$$

and the KKT conditions are

$$Q\left[\frac{\partial R}{\partial Q}(1+\mu) - \mu\frac{\partial C}{\partial Q}\right] = 0$$

$$R(Q) - C(Q) - G_{min} \geq 0$$

$$\mu \geq 0$$

$$\mu[R(Q) - C(Q) - G_{min}] = 0$$

4. Revenue Growth vs. Cost Growth: Since

$$Q = 0$$

would violate the minimum profit constraint

$$Q > 0$$

must hold, and hence the third condition implies that the first condition holds with equality. Solving that equality gives

$$\frac{\partial R}{\partial Q} = \frac{\mu}{1+\mu}\frac{\partial C}{\partial Q}$$

5. Impact of the Minimum Profit Constraint: Because it was given that $\frac{\partial R}{\partial Q}$ and $\frac{\partial C}{\partial Q}$ are strictly positive, the inequality along with the non-negativity condition on $\mu$ is

positive and so the revenue-maximizing firm operates at a level of output at which the marginal revenue $\frac{\partial R}{\partial Q}$ is less than the marginal cost $\frac{\partial C}{\partial Q}$ - a result that is of interest because it contrasts the behavior of a profit maximizing firm which operates at a level at which they are equal.

## KKT Conditions Application – Value Function

1. Equality/Inequality Function Space Constraints: Reconsidering the optimization problem as a maximization problem with constant inequality constraints $v_1, \cdots, v_n$ Maximize $f(x)$ subject to

$$g_i(x) \le a_i$$

$$h_j(x) = 0$$

2. Definition of the Value Function: The value function is defined as

$$V(a_1, \cdots, a_n) = \underset{x}{sup}\, f(x)$$

subject to

$$g_i(x) \le a_i$$

$$h_j(x) = 0$$

$$j \in \{1, \cdots, l\}$$

$$i \in \{1, \cdots, m\}$$

So the domain of $V$ is

$$a \in \mathbb{R}^m$$

for some

$$x \in X$$

$$g_i(x) \leq a_i$$

$$i \in \{1, \cdots, m\}$$

3. Interpretation of $a_i$ and $\mu_i$: Give this definition each coefficient $\mu_i$ is the rate at which the value of the function increases as $a_i$ increases. Thus if each $a_i$ is interpreteed as a resource constraint the coefficients indicate how much increasing the resource increases the optimum value of $f$. This interpretation is especially important in economics and is used, for example, in utility maximization problems.

## Generalizations

1. KKT Extensions – Fritz John Conditions: With an extra constant multiplier $\mu_0$ which may be zero, in front of $\vec{\nabla}f(x^*)$ the KKT stationary conditions turn into

$$\mu_0\vec{\nabla}f(x^*) + \sum_{i=1}^{m} \mu_i\vec{\nabla}g_i(x^*) + \sum_{j=1}^{l} \lambda_i\vec{\nabla}h_j(x^*)$$

which are called the Fritz John conditions.

2. KKT Class as FONC Support: The KKT conditions belong to the wider class of the First Order Necessary Conditions (FONC) which allow for non-smooth functions using sub-derivatives.

## References

- Boyd, S., and L. van den Berghe (2004): *Convex Optimization* **Cambridge University Press**.
- Eustaquio, R., E. Karas, and A. Ribeiro (2008): *Constraint Qualification under Non-linear Programming* Technical Report **Federal University of Parana**.
- Hanson, M. A. (1999): Invexity and the Kuhn-Tucker Theorem *Journal of Mathematical Analysis and Applications* **236 (2)** 594-604.
- Karush, W. (1939): *Minima of Functions of Several Variables with Inequalities as Side Constraints* Master's Thesis **University of Chicago** Chicago Illinois.
- Karush-Kuhn-Tucker Conditions (Wiki).
- Kjeldsen, T. H. (2000): A Contextualized Analysis of the Kuhn-Tucker Theorem in Non-linear Programming: The Impact of World War II *Historia Mathematica* **27 (4)** 331-361.
- Martin, D. H. (1985): The Essence of Invexity *Journal of Optimization Theory and Applications* **47 (1)** 65-76.
- Ruszczynski, A. (2006): *Nonlinear Optimization* **Princeton University Press** Princeton NJ.

# Interior Point Method

## Motivation, Background, and Literature Survey

1. Definition of Interior Point Methods: Interior Point Methods (also known as *barrier methods*) are a class of algorithms that solves linear and non-linear convex optimization problem (Interior Point Method (Wiki)).
3. John von Neumann's Early Approach: John von Neumann suggested an interior point method of linear programming that was neither a polynomial time method not an efficient method in practice (Dantzig and Thapa (2003)). In fact it turned out to be slower in practice than the simplex method which is not a polynomial time method.
4. Karmarkar's Extension to the Simplex Method: Karmarkar (1984) developed a method for linear programming called the Karmarkar's algorithm which runs in provably polynomial time, and is also very efficient in practice. It enabled solutions to linear programming problems that were beyond the capabilities of the simplex method.
5. Traversal across the Feasible Region: Contrary to the Simplex method Karmarkar's algorithm reaches the best solution by traversing the interior of the feasible region. The method can be generalized to convex programming based on a self-concordant barrier function used to encode the convex set.
6. Transformation of the Convex Function: Any convex optimization problem can be transformed into minimizing (or maximizing) a linear function over a convex set by converting to an epigraph form (Boyd and van den Berghe (2004)). The idea of encoding the feasible set using a barrier and designing barrier methods was studied by Anthony V. Fiacco, Garth P. McCormick, and others in the early '60s. These ideas were mainly developed for general non-linear programming, but they were later abandoned due to the presence of more competitive methods for this class of problems (e.g., sequential quadratic programming).

7. Barriers to Encode Convex Set: Yuri Nesterov and Arkadi Nemirovskii came up with a special class of such barriers that can be used to encode any convex set. They provide guarantees that the number of iterations of the algorithm is bounded by a polynomial in the dimension and as well for the accuracy of the solution (Wright (2004)).
8. Interior Point Methods with Barriers: Karmarkar's breakthrough re-vitalized the study of interior point methods and barrier problems showing that it was possible to create an algorithm for linear programming characterized by polynomial time complexity, and, moreover, that was competitive with the simplex method. Already Khachiyan's ellipsoid method was a polynomial time algorithm; however it was too slow to be of practical interest.
9. Interior Point Method Sub-classes: The class of primal dual path-following interior point methods is considered the most successful. Mehrotra's predictor-corrector algorithm provides the basis for most implementations of this class of methods (Mehrotra (1992), Potra and Wright (2000)).

## Interior Point Methodology and Algorithm

1. Principle, Concept, and Approach Methodology: The main idea behind interior point methods is to iterate inside of the feasible set and progressively approach the boundary – if the minimum does lie on the boundary. This is done by transforming the original problem into a sequence of unconstrained optimization problems in which the objective function uses a *barrier function* that goes to $\infty$ at the boundaries of the feasible region. By reducing the strength of the barrier at each subsequent optimization the sequence of minima approach arbitrarily closely toward the minimum of the original problem.
2. Objective Function with Logarithmic Barriers: For the inequality constrained problem

$$\underset{x \in \mathbb{R}^n}{min} f(x)$$

such that

$$g_i(x) \leq 0, i = 1, \cdots, m$$

$f$ is modified to obtain a *barrier function*

$$f_\alpha(x) = f(x) - \alpha \sum_{i=1}^{m} \log g_i(x)$$

3. Impact of the Barrier Strength Parameter: Since the logarithm goes to $-\infty$ as its argument approaches 0 the modified barrier function becomes arbitrarily large as $x$ approaches the boundaries of the feasible region. The parameter $\alpha$ gives the barrier "strength". Its significance is that as $\alpha$ approaches 0 the optimum of $f_\alpha(x)$ approaches the optimum of $f(x)$. More precisely if $f^*$ optimizes $f(x)$ and

$$x_\alpha{}^* \equiv \underset{x}{\arg min}\, f_\alpha(x)$$

then

$$\underset{\alpha \to 0}{Limit}\, f(x_\alpha{}^*) = f^*$$

4. Gradient of the Barrier Function: Newton's method is employed to find the minimum of $f_\alpha(x)$ The gradient is

$$\nabla f_\alpha(x) = \nabla f(x) - \alpha \sum_{i=1}^{m} \frac{\nabla g_i(x)}{g_i(x)}$$

5. Numerical Stability Close to the Boundary: One concern is that as ${x_\alpha}^*$ proceeds closer and closer to the boundary the numerical stability of the unconstrained optimization problem becomes worse and worse because the denominator approaches zero. Thus it would be very challenging to apply the gradient descent to a small tolerance.
6. Use of KKT Type Multipliers: Therefore another set of KKT multiplier like variables

$$\vec{\lambda} = (\lambda_1, \cdots, \lambda_m)$$

with

$$\lambda_i \equiv \frac{\alpha}{g_i(x)}$$

is introduced. So in the $(x, \vec{\lambda})$ space one seeks the root of the equation

$$\nabla_x f_\alpha(x, \vec{\lambda}) = \nabla f(x) - \sum_{i=1}^{m} \lambda_i \nabla g_i(x) = 0$$

subject to the equality

$$\lambda_i g_i(x) = \alpha$$

for

$$i = 1, \cdots, m$$

7. Stability of the Modified Solution: This set of equations is far more numerically stable than

$$\nabla f_\alpha(x) = \nabla f(x) - \alpha \sum_{i=1}^{m} \frac{\nabla g_i(x)}{g_i(x)}$$

near the boundary. Note the similarity between these equations and the KKT equations. In fact if α were 0 one gets the KKT equations except with the equalities stripped away.

8. Objective/Constraint Function - Partial Derivatives: To find a root $\left(x_\alpha{}^*, \vec{\lambda}_\alpha{}^*\right)$ where both of the functions are satisfied the Newton's method is applied. To do so one needs the partial derivatives of the above functions.

$$\nabla_{xx}{}^2 f_\alpha\left(x, \vec{\lambda}\right) = \nabla^2 f(x) - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x)$$

$$\nabla_{\vec{\lambda}} \nabla_x f_\alpha(x) = -\sum_{i=1}^{m} \nabla g_i(x)$$

$$\nabla_x\left(\lambda_i g_i(x) - \alpha\right) = \lambda_i \nabla g_i(x)$$

$$\frac{\partial}{\partial \lambda_i}\left[\lambda_i g_i(x) - \alpha\right] = g_i(x)$$

for

$$i = 1, \cdots, m$$

9. Variate/Constraint Multipliers newton Increment: At the current iterate $\left(x_t, \vec{\lambda}_t\right)$ the Newton step $\left(\Delta x_t, \Delta \vec{\lambda}_t\right)$ is derived from the following system of equations.

$$\begin{bmatrix} \mathcal{H} & -\mathcal{G} \\ Diagonal\left(\vec{\lambda}_t\right)\cdot\mathcal{G}^T & Diagonal(\vec{g}) \end{bmatrix}\begin{bmatrix}\Delta x_t \\ \Delta\vec{\lambda}_t\end{bmatrix} = \begin{bmatrix} -\nabla f(x_t)+\mathcal{G}\cdot\vec{\lambda}_t \\ \alpha\mathbb{I} - Diagonal(\vec{g})\cdot\vec{\lambda}_t\end{bmatrix}$$

where $\mathcal{H}$ denotes the Hessian

$$\nabla_{xx}{}^{2} f_{\alpha}\left(x,\vec{\lambda}\right) = \nabla^2 f(x) - \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x)$$

evaluated at $\left(x_t,\vec{\lambda}_t\right)$, $\vec{g}$ denotes the $m \times 1$ vector of $g_i$'s, $\mathcal{G}$ denotes the $n \times m$ matrix whose $i^{th}$ column is $\nabla g_i(x_t)$, $Diagonal(\vec{v})$ produces a square matrix whose diagonal is the vector $\vec{v}$, and $\mathbb{I}$ denotes the $m \times 1$ vector of all 1's.

10. Variate Retention inside Feasible Region: Solving this system of equations provides a search direction that is then update via line search to avoid divergence. Note that to keep $x$ drifting out of the feasible set one must enforce for all $i$ the condition

$$g_i(x) \geq 0$$

or equivalently

$$\lambda_i \geq 0$$

during the line search.

11. Progressive Reduction of the Barrier Strength: Once the unconstrained search has converged $\alpha$ may be reduced (e.g., by multiplication by a small number) and the optimization can begin again. There is a trade-off in the convergence thresholds for each unconstrained search; too small and the algorithm must perform a lot of work even when $\alpha$ is high; but too large and the sequence of unconstrained optima does not converge quickly. A more balanced approach is to choose the threshold proportional to $\alpha$.

12. Application to Non-Convex Functions: The formulation above did not explicitly require that the problem be convex. Interior point methods can certainly be used in general non-convex problems, but like any local optimization problem they are not guaranteed to a minimum. Furthermore computing the Hessian matrix is quite often expensive.
13. Use in Linear/Quadratic Problems: They do, however, work quiet well in convex problems. For example in quadratic programming the Hessian is constant, and in linear programming the Hessian is zero.
14. Initialization at a Feasible Point: The interior point method must be initialized at an interior point, or else the barrier function is undefined. To find the initial feasible point $x$ the following optimization may be used.

$$\underset{x \in \mathbb{R}^n, s_1, \cdots, s_m}{max} \sum_{i=1}^{m} s_i$$

such that

$$g_i(x) - s_i \geq 0, i = 1, \cdots, m s_i \geq 0$$

If the problem is feasible then the optimal $s_i$ will all be $0$. It is easy to find an initial set of $s_i$ for any given $x$ simply by setting

$$s_i = \min(g_i(x), 0)$$

## References

- Boyd, S., and L. van den Berghe (2004): *Convex Optimization* **Cambridge University Press**.

- Dantzig, G. B., and M. N. Thapa (2003): *Linear Programming 2 – Theory and Extensions* **Springer-Verlag**.
- Interior Point Method (Wiki).
- Karmarkar, N. (1984): A New Polynomial-Time Algorithm for Linear Programming *Combinatorica* **4 (4)** 373-395.
- Mehrotra, S. (1992): On the Implementation of the Primal-Dual Interior Point Method *SIAM Journal on Optimization* **2 (4)** 575-601.
- Potra, F. A., and S. J. Wright (2000): Interior Point Methods *Journal of Computational and Applied Mathematics* **124 (1-2)** 281-302.
- Wright, M. H. (2004): The Interior-Point Revolution in Optimization; History, Recent Developments, and Lasting Consequences *Bulletin of the American Mathematical Society* **42 (1)** 39-56.

# Optimizer

## Constrained Optimization using Lagrangian

1. Base Set up: Use the Lagrangian objective function to optimize a multi-variate function $L(x, y)$ to incorporate the constraint

   $$g(x, y) = c$$

   as

   $$\Lambda(x, y, z) = L(x, y) + \lambda[g(x, y) - c]$$

   Here $\lambda$ is called the Lagrange multiplier, and use one Lagrange multiplier per constraint.

2. Optimize (Maximize/Minimize) the Lagrangian: Optimize (i.e., maximize/minimize) the Lagrangian with respect to $x$, $y$, and $\lambda$ – thus

   $$\frac{\partial\Lambda(x, y, z)}{\partial x} = 0$$

   $$\frac{\partial\Lambda(x, y, z)}{\partial y} = 0$$

   and

   $$\frac{\partial\Lambda(x, y, z)}{\partial \lambda} = 0$$

Notice that

$$\frac{\partial \Lambda(x, y, z)}{\partial \lambda} = 0$$

automatically implies the validity of the constraint

$$g(x, y) = c$$

thereby accommodating it in a natural way.

- Further

$$\frac{\partial^2 \Lambda(x, y, z)}{\partial \lambda^2} = 0$$

always, since $\Lambda(x, y, z)$ is linear in $\lambda$. Further, since the constraint is true by the optimizer construction, there should be no explicit dependence on $\lambda$.

3. <u>Comparison with unconstrained optimization</u>:

- Unconstrained Optimization results in

$$\frac{\partial L(x, y)}{\partial x} = 0$$

and

$$\frac{\partial L(x, y)}{\partial y} = 0$$

Constrained Optimization results in

$$\frac{\partial\Lambda(x,y,z)}{\partial x}=\frac{\partial L(x,y)}{\partial x}+\lambda\frac{\partial g(x,y)}{\partial x}$$

and

$$\frac{\partial\Lambda(x,y,z)}{\partial y}=\frac{\partial L(x,y)}{\partial y}+\lambda\frac{\partial g(x,y)}{\partial y}$$

$$\lambda=0$$

automatically reduces the constrained case to the unconstrained.

- Advantage of the Lagrange Multiplier Incorporation into Optimization => This ends up converting the constrained formulation space over on to the unconstrained, thereby providing with as many equations as the number of optimization unknowns, and one equation each per every constraint.
- Drawback of the Lagrange Multiplier Optimization Incorporation => While it lets you passively move to the unknowns' space from the constrained formulation space, this may end up impacting the application of certain boundary conditions, such as financial boundary, natural boundary conditions, Not-A-Knot boundary condition, etc.

4. Constraints of inequality $g(x,y)<c$ or $g(x,y)>c$: Solutions to these constraints exist either inside the unconstrained set, in which case the unconstrained equations can be solved, or they don't, in which case convert the inequality to an equality and give it the Lagrange multiplier treatment.
5. Comparison with Convex Optimization: Convex optimization is predicated on the presence of at least one minimum/maximum in the zone of interest. One example is variance/covariance constrained optimization, where both the variance/covariance and the constraints are both quadratic. Eigenization (just another type of constrained variance/covariance optimization) is another.
6. Penalizing Optimizer as a Constrained Optimization Setup: Naively put

$$\Lambda = Good - Bad$$

where, from a calibration point of view, "Good" refers to closeness of fit, and "Bad" to the curvature/smoothness penalty. Thus, constrained optimization here corresponds to "maximize the Good, and minimize the Bad".

- De-coupling "Good/Bad" from closeness of fit and curvature/smoothness penalty, respectively, can lead to alternate optimization framework formulations in finance, along with its insights.

## Least Squares Optimizer

1. Least Squares Optimization Formulation:

$$\mu_i(x_i) = y_i$$

$$\hat{\mu}_i(x_i) = \sum_{j=1}^{n} a_j B_{ij}(x_i)$$

$$S = \sum_{i=1}^{m} [\mu_i(x_i) - \hat{\mu}_i(x_i)]^2 = \sum_{i=1}^{m} \left[ y_i - \sum_{j=1}^{n} a_j B_{ij}(x_i) \right]^2$$

$$= \sum_{i=1}^{m} \left\{ {y_i}^2 - 2y_i \sum_{j=1}^{n} a_j B_{ij}(x_i) + \left[ \sum_{j=1}^{n} a_j B_{ij}(x_i) \right]^2 \right\}$$

$$\frac{\partial S}{\partial a_j} = 2 \left[ \sum_{i=1}^{m} a_j B_{ij}(x_i) - y_j \right] \left[ \sum_{i=1}^{m} a_j B_{ij}(x_i) \right]$$

2. Least Squares Matrix Formulation:

$$Y = [y_1, \cdots, y_m]^T$$

$$B_i(\vec{x}) = \left[B_{i,1}(\vec{x}), \cdots, B_{i,m}(\vec{x})\right]^T$$

$$a = [a_1, \cdots, a_m]^T$$

Then

$$\left[\frac{\partial \vec{S}}{\partial A}\right] = 2AB(\vec{x})B^T(\vec{x})A - 2YB(\vec{x})$$

As expected

$$y_i = \sum_{j=1}^{n} a_j B_{ij}(x_i)$$

is the optimized least squares tight fit.

# Multi-variate Distribution

1. Mean/Variance Location Dependence: The mean is sensitive to both translation and rotation, unless the distribution is mean centered. The variance along a given fixed direction, however, is not sensitive to rotation of the basis.
    - This also implies that the maximal/minimal variances are invariant to representational basis changes. They are, however, sensitive to scaling, though, as PCA itself is.
2. Dimensional Independence vs. Dimensional Realization Independence: Dimensions are distinct (pressure, temperature etc.), but the realizations in those dimensions need not be. Therefore, correlated unit vectors only apply to actual realizations (and they are NOT scale invariant).
3. Orthogonal Data Set in the Native Basis Representation: If a data set is orthogonal under a given basis, then

$$\langle x_i x_j \rangle = 0$$

    (See Figure 1).
4. Non-orthogonal Data Set in the Native Basis Representation: In this case, the native representation basis results in

$$\langle x_i x_j \rangle \neq 0$$

    (See Figure 2). Thus, an orthogonalization operation needs to be performed such that under the new basis

$$\langle x_i' x_j' \rangle = 0$$

(See Figure 3).

5. Orthogonalization as Principal Components Extraction: As can be seen from figures 2 and 3, under the new schematic axes

$$\langle x_i' x_j' \rangle = 0$$

Further these correspond to the principal components, i.e., orthogonal components where the variance is an extremum.

6. Orthogonalized Representation: Upshot of all these is that, if the representation basis is structured such that

$$\langle x_i x_j \rangle = 0$$

for all

$$i \neq j$$

then these representations automatically correspond to principal components as well.

7. Full Rank Matrix: This is simply an alternate term for multi-collinear matrix.

## Parallels between Vector Calculus and Statistical Distribution Analysis

1. DOT PRODUCT => The notion of dot product is analogous to covariance operation in statistical analysis, i.e., DOT PRODUCT is

$$\vec{x}_i \cdot \vec{x}_j = 0$$

if

$$\vec{x}_i \perp \vec{x}_j$$

and covariance is

$$\langle x_i x_j \rangle = 0$$

if $x_i$ and $x_j$ are orthogonal to each other.

2. Distance Metric => Vector Euclidean distance is $\sum[(x_1 - x_2)^2 + (y_1 - y_2)^2 + \cdots]$ is equivalent to the variance $\sum[(x_i - \mu_i)^2 + \cdots]$. Further, extremizing the Euclidean/Frobenius distance is analogous to variance minimization/maximization techniques.

# Linear Systems Analysis and Transformation

## Matrix Transforms

1. Co-ordinate Rotation and Translation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} + B$$

   where $A$ is the rotating transformer, and $B$ is the translator.

2. Scaling vs. Rotation: Say

$$A = \begin{bmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \end{bmatrix}$$

   If

$$a_{11} = a_{22} \neq 1$$

   and

$$a_{12} = a_{21} = 0$$

   it becomes pure scaling.

$$a_{11} \neq a_{22}$$

   and

$$a_{12} = a_{21} = 0$$

produces differential elongation/compression and

$$a_{12} \neq 0$$

or

$$a_{21} \neq 0$$

results in rotation.

3. Uses of Gaussian Elimination:
   - Linearization
   - Orthogonalization
   - Inversion
   - Diagonalization
   - Lower/Upper Triangle Decomposition (LU Decomposition)
   - Independent Component Extraction
   - Principal Component Analysis
4. Diagonal Identity Matrix Conception: Given a matrix $A$ what matrix $M$ should it be transformed by to get a diagonal identity matrix, i.e.

$$MA = I$$

Answer is

$$M = A^{-1}$$

Thus, diagonalization is also an inversion operation.

$$Diagonalization == Orthogonalization == Inversion == ICA$$

Diagonalization, of course, is unique only to a diagonal entry, whereas inversion corresponds to a specific choice of the diagonal entry.

## Systems of Linear Equations

1. Importance of Diagonal Dominance in Gauss-Seidel: Is diagonal dominance important because the dominant diagonal's contribution to the RHS drives the given equation's value, and therefore the iterative accuracy?
2. Eigenization Square Matrix Inversion Conceptualization: Given a source matrix

$$F_{SOURCE,0} = \{f_{ij}\}_{i,j=0}^{n-1}$$

and an initialized target inverse identity matrix

$$F_{INV,0} = \{I\}_{n \times n}$$

achieve a suitable set of $p$ transformations simultaneously on $F_{SOURCE}$ and $F_{INV}$ to eventually make

$$F_{SOURCE,p} = \{I\}_{n \times n}$$

so that the corresponding

$$F_{INV,p} = \{f_{INV,ij}\}_{i,j=0}^{n-1}$$

becomes the inverse.

3. Valid Inversion Rules: These are fairly straight forward application of the Gaussian elimination scheme:
    - Scale a single $F_{SOURCE}$ row by a constant => scale the corresponding $F_{INV}$ row by the same constant.
    - Add/subtract any pair of $F_{SOURCE}$ rows from each other => add/subtract the same pair of $F_{INV}$ rows from each other.
4. Matrix Inversion using Gaussian Elimination:
    - Scan across the diagonal entries.
    - Scan through the rows corresponding to each diagonal entry.
    - If a given row entry call value is zero, or its index corresponds to that of a diagonal, skip.
    - Calculate the $WorkColFactor$ as

$$WorkColFactor = \frac{DiagonalEntry}{CellValueEntry}$$

    - Scan all the cells in the column of the current cell.
    - Apply the $WorkColFactor$ product to each entry in the current working column of the source matrix.
    - Do the same as above to the inverse matrix.
    - Subtract the entries corresponding to the designated diagonal column from the working column to make the current entry zero.
    - Do the same as above to the inverse matrix as well.
    - After the completion of all such diagonal scans, row scans, and working column scans, re-scan the diagonal again.
    - Scale down each entry of the source matrix by itself, so that the source matrix entries now constitute an identity ($\{I\}_{n\times n}$) matrix.
    - Do the same as above to the inverse matrix as well.

5. Non-invertible Coefficient Matrix, but Solution exists: For unprocessed coefficient matrices, certain conditions (such as zero diagonal entries) may cause the coefficient matrix to be technically non-invertible, but that does not automatically mean that the system is unsolvable – a simply re-casting of the basic linear system set may be all that is required.
6. Linear Basis Re-arrangement: Sometimes, a singular coefficient matrix (with zero determinant, therefore non-invertible) may be re-arranged to create an invertible coefficient matrix. After inversion, it can be re-structured again to extract the inverse (which is just a coefficient Jacobian).
7. Rows/Columns as "Preferred Linear Basis Sequence" for Matrix Manipulation: Consider the solution to

$$AX = Z$$

where $X$ and $Z$ are columns. In this case, the notion of constraint linear representation is maintained exclusively in rows. Therefore, all elimination/scaling basis operations need to be applied on that basis. In considering the solution to

$$AX = Z$$

where $X$ and $Z$ are rows, columns now become the preferred linear basis sequence.
8. Regularization before Gauss Elimination: Before the Gauss Elimination can process, we need to diagonalize the matrix. Row swapping is a more robust way to diagonalize than row accumulation due to a couple of reasons:
    - Matrix Row Swap vs. Row Cumulate => In all cases, row swap can be transformed into row accumulation. When inverting, however, the row swapping AND accumulation should both be done on both the SOURCE and the TARGET matrices.
    - From a core linear operation set point-of-view, row accumulation is the inverse of the eventual of Gauss Elimination, so the danger is that the

    diagonalization gains of row swap may be undone during the intermediate stages of Gauss Elimination.
    - Even more important is that row swapping simply retains the original information, by just re-arranging the row set.

9. Row Swapping Caveats:
    - Always swap rows by retaining the directionality of the scan AND by retaining the scan initial node preceding the swap (to choose the pivot). One way to do this is by starting the scan at $row + 1$ - or from

$$row = 0$$

    if the edge has been reached - AND always keeping the scan sequence forward/backward.
    - Also ensure that the target swapped row is “valid”, i.e., it’s post-swapped diagonal entry should be non-zero. If this cannot be achieved through the scan, then that is an error condition.

10. Diagonalization/Inversion Algorithm: Work in terms of an intermediate transform variate $Z$ produced by re-arranging the original coefficient matrix and $Y$ such that the $A$ in

$$AX = Z$$

is now invertible. The following would be the steps:

- First, re-arrange the equation system set to identify a suitable pair $A$ and $Z$ such that $A$ is invertible, and $Z$ is estimated from

$$AX = Z$$

- The re-arranging linear operation set will produce $A$ such that

$$BY = Z \Longrightarrow AX = BY \Longrightarrow X = A^{-1}BY$$

11. **More General Inverse Transformation Re-formulation:** Given the coefficient matrix $\{a_{qj}\}_{j,q=0}^{n-1}$, the set of unknown variables $\{x_j\}_{j=0}^{n-1}$, and the RHS $\{y_q\}_{q=0}^{n-1}$, $y_p$, $y_q$, and the corresponding $z_p$ and $z_q$ are given as

$$\sum_{j=0}^{n-1} a_{qj} x_j = y_q$$

$$\sum_{j=0}^{n-1} a_{pj} x_j = y_p$$

and

$$\{z_j = y_j\}_{j=0}^{n-1}$$

On transformation (i.e., adding row $p$ to row $q$), you get

$$\sum_{j=0}^{n-1} (a_{pj} + a_{qj}) x_j = y_p + y_q$$

and therefore

$$\{z_j = y_j\}_{j=0;j\neq q}^{n-1}$$

along with

$$z_q = y_p + y_q$$

This clearly implies that

$$\boldsymbol{\frac{\partial z_q}{\partial y_p} = 1}$$

or

$$\boldsymbol{B_{qp} = B_{qp} + 1}$$

## Orthogonalization

1. <u>2D Orthogonalization</u>: In 2D, you need to fix one 1D orthogonal axis to be able to orthogonalize the other.
2. <u>2D Equation System</u>:

$$x_1 = a_{11}s_1 + a_{12}s_2$$

and

$$x_2 = a_{21}s_1 + a_{22}s_2$$

This has 4 unknowns, so one solution for this is as follows: Fix

$$a_{11} = 1$$

and

$$a_{12} = 0$$

This results in 2 unknowns. Setting

$$\langle {x_1}^2 \rangle = 1$$

$$\langle {x_2}^2 \rangle = 1$$

and

$$\langle x_1 x_2 \rangle = \rho$$

you get

$${a_{11}}^2 + {a_{12}}^2 = 1$$

and

$${a_{21}}^2 + {a_{22}}^2 = 1$$

resulting in

$$a_{21} = \rho$$

and

$$a_{21} = \sqrt{1 - \rho^2}$$

Thus, all unknowns are determined.

3. nD Orthogonalization:

$$\begin{bmatrix} x_0 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ \vdots & \ddots & \vdots \\ a_{n-1,0} & \cdots & a_{n-1,n-1} \end{bmatrix} \begin{bmatrix} s_0 \\ \vdots \\ s_{n-1} \end{bmatrix}$$

- Number of diagonal entries => $n$
- Number of non-diagonal entries => $n^2 - n$
- Net Number of equations => Number of diagonal entries + Number of non-diagonal entries =>

$$n + \frac{n(n-1)}{2} = \frac{n(n+1)}{2}$$

4. nD Unknowns Analysis:
   - Number of Unknowns => $n^2$.
   - Fix the first row, and the number of unknowns becomes => $n^2 - n$.
   - Fixing the row takes off one equation, so the number of equations => $\frac{n(n+1)}{2} - 1$
   - Number of equations = Number of Unknowns =>

$$\frac{n(n+1)}{2} - 1 = n^2 - n$$

results in

$$(n-1)(n-2) = 0$$

## Gaussian Elimination

1. n-D Gaussian Elimination: What does it work? If you fix a row, you can rotate the other rows to eliminate dependence on an ordinate of the fixed row.

2. Row Fixation as a Basis Choice: Row elimination does not automatically make the matrix diagonal or orthogonalize it – all it does is to eliminate dependence on a stochastic variate.
3. Number of Elimination Rotations: The first row fixation results in $n-1$ rotations, the second row fixation results in $n-2$ rotations, and so on. Thus, the total number of rotating transformations is $\frac{(n-1)(n-2)}{2}$ (i.e., the same as the number of unknowns seen earlier). The result of these transformations is a lower/upper triangular matrix.
4. Final Reversing Sweep: An additional reverse sweep would eliminate similar column dependencies as well – resulting in another $\frac{(n-1)(n-2)}{2}$ rotation choices.
5. Number of Sweep Operations: Total result of all the rotations and their corresponding choices =>

$$2\cdot\frac{(n-1)(n-2)}{2}=(n-1)(n-2)$$

# Rayleigh Quotient Iteration

## Introduction

1. Idea behind Rayleigh Quotient Iteration: **Rayleigh Quotient Iteration** is an eigenvalue algorithm that extends the idea of inverse iteration by using the Rayleigh Quotient to obtain increasingly accurate eigenvalue estimates (Wiki - Rayleigh Quotient Iteration (2018)).
2. Progressive Navigation towards "True" Selection: Rayleigh Quotient Iteration is an iterative method, i.e., it delivers a sequence of approximate solutions that converge to a true solution in the limit. This is true for all algorithms that compute eigenvalues; since the eigenvalues can be irrational numbers, there can be no general method for computing them in a finite number of steps.
3. Practicality of the Iterative Approach: Very rapid convergence is guaranteed and no more than a few iterations are needed in practice to obtain a reasonable solution.
4. Cubic Convergence for Typical Matrices: The Rayleigh Quotient algorithm converges cubically for Hermitian or symmetric matrices, given an initial vector that is sufficiently close to an eigenvector of the matrix that is being analyzed.

## The Algorithm

1. Update Eigenvalue using Rayleigh Quotient: The algorithm is very similar to inverse iteration, but replaces the estimated eigenvalues at the end of each iteration with the Rayleigh Quotient.

2. Initial Guess for Eigenvalue/Eigenvector: Begin by choosing a value $\mu_0$ as an initial eigenvalue guess for the Hermitian matrix $A$. An initial vector $b_0$ must also be supplied as the initial eigenvector guess.
3. Iterative Eigenvalue and Eigenvector: Calculate the subsequent approximation of the eigenvector $b_{i+1}$ by

   $$b_{i+1} = \frac{[A - \mu_i I]^{-1} b_i}{\|[A - \mu_i I]^{-1} b_i\|}$$

   where $I$ is the identity matrix, and set the next approximation of the eigenvalue to the Rayleigh quotient of the current iteration equal to

   $$\mu_i = \frac{{b_i}^* A b_i}{{b_i}^* b_i}$$

4. Deflation Techniques for Successive Eigenvalues: To compute more than one eigenvalue the algorithm can be combined with a deflation technique.
5. Treatment for Very Small Matrices: Note that for very small matrices it is beneficial to replace the matrix inverse with the adjugate, which will yield the same iteration because it is equal to the inverse to an irrelevant scale – specifically, the inverse of the determinant.
6. Advantages of using the Adjugate: The adjugate is easier to compute explicitly than the inverse – though the inverse is easier to apply to a vector for problems that aren't small – and is more numerically sound because it remains well-defined as the eigenvalue changes.

## References

- Wikipedia – Rayleigh Quotient Iteration (2018)

# Power Iteration

## Introduction

1. Power Iteration – Problem Statement/Definition: In mathematics, **power iteration** – also known as the *power method* – is an eigenvalue algorithm. Given a diagonalizable matrix $A$, the algorithm will produce a number $\lambda$, which is the greatest – in absolute value – eigenvalue of $A$, and a non-zero vector $v$, the corresponding eigenvector of $\lambda$, such that

$$Av = \lambda v$$

   The algorithm is also known as the von Mises iteration (von Mises and Pollazek-Geiringer (1929)).
2. Characteristics of the Power Iteration Algorithm: Power iteration is a very simple algorithm, but it may converge slowly. It does not compute a matrix decomposition, and hence can be used when $A$ is a very large sparse matrix.

## The Method

1. Starting Choice for Principal Eigenvector: The power iteration method starts with a vector $b_0$, which may be either an approximation to the dominant eigenvector or a random factor.
2. Recurrence Relation for Power Iteration: The method is described by the recurrence relation

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|}$$

So, at every iteration, the vector $b_k$ is mulplied by the matrix $A$ and normalized.

3. <u>Necessary Condition for Eigen-component Convergence</u>: If one assumes that $A$ has an eigenvalue that is greater in magnitude than its other eigenvalues and that the starting vector $b_0$ has a non-zero component in the direction of an eigenvector associated with the dominant eigenvalue, then a sub-sequence $\{b_k\}$ converges to an eigenvector associated with the dominant eigenvector.
4. <u>Recast of $\{b_k\}$ using Phase Shift</u>: Without the two assumptions above the sequence does not necessarily converge. In this sequence

$$b_k = e^{i\phi_k} v_1 + r_k$$

where $v_1$ is the eigenvector associated with the dominant eigenvalue, and

$$\|r_k\| \to 0$$

The presence of the term $e^{i\phi_k}$ implies that $\{b_k\}$ does not converge unless

$$e^{i\phi_k} \equiv 1$$

5. <u>Convergence to the Dominant Eigenvalue</u>: Under the two assumptions listed above, the sequence $\{\mu_k\}$ defined by

$$\mu_k = \frac{{b_k}^T A b_k}{{b_k}^T b_k}$$

converges to the dominant eigenvalue.

6. Eigenvector and Eigenvalue Sequences: The vector $b_k$ is the associated eigenvector. Ideally one should use the Rayleigh Quotient in order to get the associated eigenvalue.
7. Spectral Radius using the Rayleigh Quotient: The method can also be used to calculate the spectral radius – the largest eigenvalue of a matrix – by computing the Rayleigh Quotient

$$\frac{{b_k}^T A b_k}{{b_k}^T b_k} = \frac{{b_{k+1}}^T b_k}{{b_k}^T b_k}$$

## Analysis

1. Decomposition to Jordan Canonical Form: Let $A$ be decomposed into its Jordan Canonical Form

$$A = VJV^{-1}$$

where the first column of $V$ is an eigenvector of $A$ corresponding to the dominant eigenvalue $\lambda_1$.

2. The Principal Component Jordan Block: Since the dominant eigenvalue of $A$ is unique, the first Jordan block of $J$ is the $1 \times 1$ matrix $[\lambda_1]$, where $\lambda_1$ is the largest eigenvalue of $A$ in magnitude.
3. Initializing the Principal Component Eigenvector: The starting vector $b_0$ can be written as a linear combination of the columns of $V$:

$$b_0 = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n$$

By assumption, $b_0$ has a non-zero component in the direction of the dominant eigenvalue, so

$$c_1 \neq 0$$

4. k<sup>th</sup> Recurrence of the Initial Vector: The computationally recurrence relation for $b_{k+1}$ can be written as:

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|} = \frac{A^{k+1}b_0}{\|A^{k+1}b_0\|}$$

where the expression $\frac{A^{k+1}b_0}{\|A^{k+1}b_0\|}$ is amenable to the analysis below.

5. Principal Component Dependence of $b_k$:

$$\begin{aligned} b_k &= \frac{A^k b_0}{\|A^k b_0\|} = \frac{[VJV^{-1}]^k b_0}{\|[VJV^{-1}]^k b_0\|} = \frac{VJ^kV^{-1}b_0}{\|VJ^kV^{-1}b_0\|} \\ &= \frac{VJ^kV^{-1}(c_1v_1 + c_2v_2 + \cdots + c_nv_n)}{\|VJ^kV^{-1}(c_1v_1 + c_2v_2 + \cdots + c_nv_n)\|} \\ &= \frac{VJ^k(c_1\hat{e}_1 + c_2\hat{e}_2 + \cdots + c_n\hat{e}_n)}{\|VJ^k(c_1\hat{e}_1 + c_2\hat{e}_2 + \cdots + c_n\hat{e}_n)\|} \\ &= \left(\frac{\lambda_1}{|\lambda_1|}\right)^k \frac{c_1}{|c_1|} \frac{v_1 + \frac{1}{c_1}V\left(\frac{1}{\lambda_1}J\right)^k (c_2\hat{e}_2 + \cdots + c_n\hat{e}_n)}{\left\| v_1 + \frac{1}{c_1}V\left(\frac{1}{\lambda_1}J\right)^k (c_2\hat{e}_2 + \cdots + c_n\hat{e}_n)\right\|} \end{aligned}$$

6. Scaling Down the Principal Eigenvector: As

$$k \to \infty$$

the expression above simplifies to

$$\left(\frac{1}{\lambda_1}J\right)^k = \begin{pmatrix} [\mathbb{I}] & \cdots & & \\ \cdots & \left(\frac{1}{\lambda_1}J_2\right)^k & \cdots & \cdots \\ \cdots & \cdots & \ddots & \cdots \\ \cdots & \cdots & \cdots & \left(\frac{1}{\lambda_1}J_n\right)^k \end{pmatrix} \rightarrow \begin{pmatrix} [\mathbb{I}] & \cdots & \cdots & \cdots \\ \cdots & 0 & & \\ \cdots & \cdots & \ddots & \cdots \\ \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

7. <u>Limit of $\left(\frac{1}{\lambda_1}J_i\right)^k$ as $k \rightarrow \infty$</u>: The limit follows from the fact that the eigenvalue of $\left(\frac{1}{\lambda_1}J_i\right)^k$ is less than 1 in magnitude, so

$$\left(\frac{1}{\lambda_1}J_i\right)^k \rightarrow 0$$

as

$$k \rightarrow \infty$$

8. <u>Expansion of Higher Order Eigen Terms</u>: It follows that

$$\frac{1}{c_1}V\left(\frac{1}{\lambda_1}J\right)^k (c_2\hat{e}_2 + \cdots + c_n\hat{e}_n) \rightarrow 0$$

as

$$k \rightarrow \infty$$

9. <u>Reduction of $b_k$ for $k \rightarrow \infty$</u>: Using this fact, $b_k$ can be written in a form that emphasizes its relationship with $v_1$ when $k$ is large:

$$b_k = \left(\frac{\lambda_1}{|\lambda_1|}\right)^k \frac{c_1}{|c_1|} \frac{v_1 + \frac{1}{c_1} V \left(\frac{1}{\lambda_1} J\right)^k (c_2 \hat{e}_2 + \cdots + c_n \hat{e}_n)}{\left\| v_1 + \frac{1}{c_1} V \left(\frac{1}{\lambda_1} J\right)^k (c_2 \hat{e}_2 + \cdots + c_n \hat{e}_n) \right\|} = e^{i\phi_k} \frac{c_1}{|c_1|} \frac{v_1}{\|v_1\|} + r_k$$

where

$$e^{i\phi_k} = \left(\frac{\lambda_1}{|\lambda_1|}\right)^k$$

and

$$\|r_k\| \to 0$$

as

$$k \to \infty$$

10. <u>Uniqueness of the $\{b_k\}$ Sequence</u>: The sequence $\{b_k\}$ is bounded, so it contains a convergent sub-sequence. Note that the eigenvector corresponding to the dominant eigenvalue is unique only upto a scaler, so although the sequence $\{b_k\}$ may not converge, $b_k$ is nearly an eigenvector of $A$ for large $k$.
11. <u>Alternative Proof of the Sequence Convergence</u>: Alternatively, if $A$ is diagonalizable, then the following proof yields the same result. Let $\lambda_1, \lambda_2, \cdots, \lambda_m$ be the $m$ eigenvalues – counted with multiplicity – of $A$, and let $v_1, v_2, \cdots, v_m$ be the corresponding eigenvectors. Suppose that $\lambda_1$ is the dominant eigenvalue, so that

$$|\lambda_1| > |\lambda_j|$$

for

$$j > 1$$

12. Suitable Choice for the Initial Vector: As seen earlier, the initial vector $b_0$ is written as

$$b_0 = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n$$

If $b_0$ is chosen randomly – with uniform probability – the

$$c_1 \neq 0$$

with probability 1

13. Power Iteration over Initial Eigenvector: Now

$$A^k b_0 = c_1 A^k v_1 + c_2 A^k v_2 + \cdots + c_n A^k v_n = c_1 {\lambda_1}^k v_1 + c_2 {\lambda_2}^k v_2 + \cdots + c_n {\lambda_n}^k v_n$$
$$= c_1 {\lambda_1}^k \left[v_1 + \frac{c_2}{c_1}\left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \cdots + \frac{c_n}{c_1}\left(\frac{\lambda_n}{\lambda_1}\right)^k v_n\right] \rightarrow c_1 {\lambda_1}^k v_1$$

as long as

$$\left|\frac{\lambda_j}{\lambda_1}\right| < 1$$

for

$$j > 1$$

14. Convergence to the Principal Eigenvector: On the other hand,

$$b_k = \frac{A^k b_0}{\|A^k b_0\|}$$

Therefore $b_k$ converges to a multiple of the eigenvector $v_1$

15. Convergence Ratio of Power Iteration: The convergence ratio is geometric with the ratio $\left|\frac{\lambda_2}{\lambda_1}\right|$ where $\lambda_2$ denotes the second principal eigenvalue.
16. Consequence of Close Principal Eigenvectors: Thus, the method converges slowly if there is an eigenvalue close in magnitude to the dominant eigenvalue.

## Applications

1. Identification of the Dominant Eigenvector/Eigenvalue: Although the power iteration method approximates only one eigenvalue of a matrix, it remains useful for several computational problems.
2. Use in Google and Twitter: For instance, Google uses it to calculate the PageRank of documents in their search engine (Ipsen and Wills (2005)), and Twitter uses it to show users recommendations of who to follow (Gupta, Goel, Lin, Sharma, Wang, and Zadeh (2013)).
3. Space Advantage of Power Iteration: The power iteration is especially suitable for sparse matrices, such as the web matrix, or as the matrix-free method that does not require storing the coefficient matrix $A$ explicitly, but can instead access a function evaluating matrix-vector products $Ax$.
4. Power Iteration vs. Arnoldi Method: For non-symmetric matrices that are well-conditioned, the power iteration method can outperform the more complex Arnoldi method.
5. Power Iteration vs. Lanczos/LOBPCG: For symmetric matrices, the power iteration method is rarely used, since its convergence speed can be easily increased without sacrificing the smaller cost per iteration, e.g., Lanczos iteration and LOBPCG.

6. Variation in the Method - Inverse Iteration: Some of the more advanced algorithms can be understood as variations of the power iteration method. For instance, the inverse iteration method applies power iteration to the matrix $A^{-1}$.
7. Sub-space of Eigenvalues - Krylov Subspace: Other algorithms look at the whole subspace generated by the vectors $b_k$. This subspace is known as the Krylov subspace. It can be computed by Arnoldi iteration or Lanczos iteration.

## References

- Gupta, P., A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh (2013): WTF – The Who-To-Follow Service at Twitter
- Ipsen, I., and R. Wills (2005): Analysis and Computation of Google's PageRank
- Von Mises, R., and H. Pollaczek-Geiringer (1929): Praktische Verfahren der Gleichingsauflosung *Zeitschrift fur Andewandte Mathematik und Mechanik* **9** 152-164.

# Numerical Integration

## Introduction and Overview

1. Numerical Estimate of Definite Integrals: In numerical analysis, **numerical integration** comprises a broad family of algorithms for calculating the numerical value of a definite integral, and by extension, the term is also used sometimes to describe the numerical solution to differential equations.
2. Chapter Focus on Definite Integrals: This chapter focuses on the calculation of definite integrals.
3. Numerical Quadrature: One-Dimensional Integrals: The term **numerical quadrature** – often abbreviated to *quadrature* – is more or less a synonym for **numerical integration**, especially as applied to one-dimensional integrals (Wikipedia (2019)).
4. Curbature: Multi-Dimensional Numerical Integrals: Some authors refer to numerical integration over more than one dimension as *curbature*; others take *quadrature* to include higher dimensional integration.
5. Mathematical Specification of the Definite Integral: The basic problem in numerical integration is to compute an approximate solution to a definite integral $\int_a^b f(x)dx$ to a given degree of accuracy.
6. Accurately Estimating the Definite Integral: If $f(x)$ is a smooth function integrated over a small number of dimensions, and the domain of the integration is bounded, there are many methods for approximating the integral to the desired precision.

## Reasons for Numerical Integration

1. Discrete Sampling of the Integrand: There are several reasons for carrying out numerical integration. For instance, the integrand $f(x)$ may only be known at certain points, such as those obtained by sampling. Some embedded systems and other software applications may need numerical integration for this reason.
2. Anti-derivative not an Elementary Function: An expression for the integrand may be known, but it may be difficult or impossible to find an anti-derivative that is an elementary function. An example of such an integrand is

$$f(x) = e^{-x^2}$$

   the anti-derivative of which – the error function times a constant – cannot be written in an elementary form.
3. Series Approximation of the Anti-derivative: It may be possible to find an anti-derivative symbolically, but it may be easier to compute a numerical approximation than to compute the anti-derivative. That may be the case if the anti-derivative is computed as an infinite derivative or product, or it its evaluation requires a special function that is not available.

## Methods for One-Dimensional Integrals

1. Combining Evaluations of the Integrand: Numerical integration methods can generally be described as combining the evaluations of the integrand to get an approximation to the integral.

2. Weighted Sum at the Integration Points: The integrand is evaluated at a finite set of points called the *integration points* and a weighted sum of these values is used to approximate the integral. The integration points and the weights depend on the specified method used and the accuracy required from the approximation.
3. Approximation Error of the Method: An important part of the analysis of any numerical integration method is to study the behavior of the approximation error as a function of the number of integrand evaluations. A method that yields a small error for a small number of evaluations widely considered superior.
4. Reduced Number of Integrand Evaluations: Reducing the number of operations of the integrand reduces the number of arithmetic operations involved, and therefore reduces the total round-off error. Also, each evaluation takes time, and the integrand may be arbitrarily complicated.
5. Conditions for Brute Force Integration: A brute force kind of numerical integration can be done if the integrand is reasonably well-behaved, i.e., it is piece-wise continuous and of bounded variation, by evaluating the integrand with very small increments.

## Quadrature Rules Based on Interpolating Functions

1. Error of Integrating Interpolation Polynomials: A large class of quadrature rules can be derived by constructing interpolating functions that are easy to integrate. Typically, these interpolating functions are polynomials. In practice, some of these polynomials of very high degree tend to oscillate very wildly, only polynomials of low degree are used, typically linear and quadratic.
2. Mid-Point Rule: Zero Degree Polynomial: The simplest function of this type is to let the interpolating function be a constant function – a polynomial of degree 0 – that

passes through the point $\left[\frac{a+b}{2}, f\left(\frac{a+b}{2}\right)\right]$ This is the *mid-point rule* of the *rectangle rule*:

$$\int_a^b f(x)dx \approx (b-a)f\left(\frac{a+b}{2}\right)$$

3. Trapezoidal Rule: First Degree Polynomial: The interpolating function may be a straight line – an affine function, i.e., a polynomial of degree one – passing through the points $[a, f(a)]$ and $[b, f(b)]$. This is called the *trapezoidal rule*:

$$\int_a^b f(x)dx \approx (b-a)\left[\frac{f(a)+f(b)}{2}\right]$$

4. Sub division of the Integration Intervals: For either one of these rules, a more accurate representation can be made by breaking up the interval $[a, b]$ into some number $n$ of sub-intervals, computing an approximation for each sub-interval, then adding up all the results. This is called *composite rule*, *extended rule*, or *iterated rule*.
5. Example: Composite Trapezoidal Rule: For example, the composite trapezoidal rule can be stated as

$$\int_a^b f(x)dx \approx (b-a)\left[\frac{f(a)+f(b)}{2}+\sum_{k=1}^{n-1} f\left(a+k\frac{b-a}{n}\right)\right]$$

where the sub-intervals have the form

$$[a+kh, a+(k+1)h] \subset [a, b]$$

with

$$h = \frac{b - a}{n}$$

and

$$k = 0, \cdots, n - 1$$

Here the sub-intervals used have the same length $h$, but one could use intervals of varying length $h_k$

6. Definition of Newton-Cotes Formula: Interpolation with polynomials evaluated at equally spaced points in $[a, b]$ yields the Newton-Cotes formula, of which the rectangle rule and the trapezoidal rule are examples. Simpson's rule, which is based on a polynomial of order 2, is also a Newton-Cotes formula.
7. Quadrature Rules with Nesting Property: Quadrature rules with equally spaced points have the very convenient property of *nesting*. The corresponding rule with each interval sub-divided includes all the current points, so those integrand points can be re-used.
8. Integrand with Variable Interpolant Spaces: If one allows the intervals between the interpolation points to vary, one finds another group of quadrature formulas, such as the Gaussian quadrature formula.
9. Improved Accuracy with Gaussian Quadrature: Gaussian quadrature rule is typically more accurate than a Newton-Cotes rule, which requires the same number of function evaluations, if the integrand is smooth, i.e., if it is sufficiently differentiable.
10. Other Varying Interval Quadrature Rules: Other quadrature methods with varying intervals include Clenshaw-Curtis quadrature methods – also called Fejer quadrature – and these do nest.
11. Nestability of Gaussian Quadrature Rules: Gaussian quadrature rules do not nest, but the related Gauss-Kronrod quadrature formulas do.

## Generalized Mid-Point Rule Formulation

1. Generalized Mid-Point Rule Expression: A generalized mid-point rule formula is given by

$$\int_{0}^{1} f(x)dx = \sum_{m=1}^{M}\sum_{n=0}^{\infty}\frac{(-1)^{n}+1}{(2M)^{n+1}(n+1)!}\frac{d^{n}y}{dx^{n}}\Bigg\|_{x=\frac{m-\frac{1}{2}}{M}}$$

or

$$\int_{0}^{1} f(x)dx = \begin{matrix}Limit\\ N\rightarrow\infty\end{matrix}\sum_{m=1}^{M}\sum_{n=0}^{N}\frac{(-1)^{n}+1}{(2M)^{n+1}(n+1)!}\frac{d^{n}y}{dx^{n}}\Bigg\|_{x=\frac{m-\frac{1}{2}}{M}}$$

2. Example Expression for Inverse Tangent: For example, substituting

$$M = 1$$

and

$$f(x) = \frac{\theta}{1+\theta^{2}x^{2}}$$

in the generalized mid-point rule formula, one obtains the equation of the inverse tangent as

$$\tan^{-1} z = i\sum_{n=1}^{\infty}\frac{1}{2n-1}\left[\frac{1}{\left(1+\frac{2i}{z}\right)^{2n-1}}-\frac{1}{\left(1-\frac{2i}{z}\right)^{2n-1}}\right]$$

$$= 2\sum_{n=1}^{\infty}\frac{1}{2n-1}\frac{a_n(z)}{a_n{}^2(z)+b_n{}^2(z)}$$

where

$$i = \sqrt{-1}$$

is the imaginary unit, and

$$a_1(z) = \frac{2}{z}$$

$$b_1(z) = 1$$

$$a_n(z) = a_{n-1}(z)\left[1-\frac{4}{z^2}\right]+4\frac{b_{n-1}(z)}{z}$$

$$b_n(z) = b_{n-1}(z)\left[1-\frac{4}{z^2}\right]-4\frac{a_{n-1}(z)}{z}$$

3. <u>Eliminating the Series Odd Terms</u>: Since at each odd $n$ the numerator of the integrand becomes

$$(-1)^n + 1 = 0$$

the generalized mid-point rule formula can be re-organized as

$$\int_{0}^{1} f(x)dx = 2\sum_{m=1}^{M}\sum_{n=0}^{\infty}\frac{1}{(2M)^{2n+1}(2n+1)!}\frac{d^{2n}y}{dx^{2n}}\Bigg\|_{x=\frac{m-\frac{1}{2}}{M}}$$

4. Transforming $(a, b)$ Limits into $(0, 1)$: For a function $g(t)$ defined over the interval $(a, b)$ its integral is

$$\int_{a}^{b} g(t)dt = \int_{0}^{b-a} g(\tau + a)d\tau = (b - a)\int_{0}^{1} g([b - a]x + a)dx$$

Therefore, the generalized mid-point integration formula above can be applied assuming that

$$f(x) = g([b - a]x + a)dx$$

## Adaptive Algorithms

1. Lack of Suitable Derivative Points: If $f(x)$ does not sufficient derivatives at all points, or if the derivatives become large, the generalized mid-point quadrature is often insufficient. In such cases, the adaptive algorithm similar to the one outlined in Wikipedia (2019) will perform better.
2. Estimation of the Quadrature Error: Some details of the algorithm require careful thought. For many cases, estimating the error from quadrature over an interval for the function $f(x)$ is not obvious. One popular solution is to use two different rules for the quadrature, and use their difference as an estimate of the error from the quadrature.
3. Too Large Versus Small Errors: The other problem is deciding what *too large* or *very small* signify.

4. Local Criterion for *Too Large*: A *local* criterion for *too large* is that the quadrature error should not be larger than $t \cdot h$, where $t$, a real number, is the tolerance one wishes to set for the global error. However, if $h$ is too tiny, it may not be worthwhile to make it even smaller even if the quadrature error is apparently large.
5. Global Criterion for *Too Large*: A *global* criterion is that the sum of the errors on all the intervals should be less than $t$.
6. *A-Posteriori* Type of Error Analysis: This type of error analysis is typically called *a-posteriori* since the error is computed after having computed the approximation.
7. Forsythe Heuristics for Adaptive Quadrature: Heuristics for adaptive quadrature are discussed in Forsythe, Malcolm, and Moler (1977).

## Extrapolation Methods

1. Error Dependence on Evaluation Point Count: The accuracy of a quadrature rule of the Newton-Cotes type is generally a function of the number of evaluation points. The result is usually more accurate as the number of evaluation points increases, or, equivalently, the width of the step size between the points decreases.
2. Error Dependence on Step Width: It is natural to ask what the result would be if the step size were allowed to approach zero. This can be answered by extrapolating the result from two or more non-zero step sizes, using series acceleration methods such as Richardson extrapolation.
3. Details of the Extrapolation Methods: The extrapolation function may be either a polynomial or a rational function. Extrapolation methods are described in more detail by Stoer and Bulirsch (1980) and are implemented in many of the routines in the QUADPACK library.

## A Priori Conservative Error Estimation

1. <u>Bounded First Derivative over $[a, b]$</u>: Let $f$ have a bounded first derivative over $[a, b]$, i. e.,

   $$f \in \mathbb{C}^1(a, b)$$

   The mean-value theorem for $f$ where

   $$x \in (a, b]$$

   gives

   $$(x - a)\frac{df(\xi_x)}{dx} = f(x) - f(a)$$

   for some

   $$\xi_x \in (a, x]$$

   depending on $x$.

2. <u>Integrating Over $x$ on Both Sides</u>: On integrating in $x$ from $a$ to $b$ on both sides and taking the absolute values, one obtains

   $$\left|\int_a^b f(x)dx - (b - a)f(a)\right| \leq \left|\int_a^b (x - a)\frac{df(v_x)}{dx}dx\right|$$

3. Approximating the Right-Hand Side: The integral on the right-hand side can be further approximated by bringing the absolute value into the integrand, and replacing the term $\frac{df}{dx}$ by an upper bound:

$$\left| \int_a^b f(x)dx - (b-a)f(a) \right| \leq \frac{(b-a)^2}{2} \sup_{a \leq x \leq b} \left| \frac{df}{dx} \right|$$

   where the supremum has been used for the approximation.

4. Corresponding Approximation Applied to the LHS: Hence, if the integral $\int_a^b f(x)dx$ is approximated by the quadrature $(b-a)f(a)$, the error is no greater than the right-hand side above.

5. Converting the RHS into a Riemann Sum: This can be converted into an error analysis for the Riemann sum, giving an upper bound of

$$\frac{n^{-1}}{2} \sup_{0 \leq x \leq 1} \left| \frac{df}{dx} \right|$$

   for the error term of that particular approximation. Note that this precisely is the error obtained for the example

$$f(x) = x$$

6. Strict Upper Bounds on the Error: Using more derivatives, and by tweaking the quadrature, a similar analysis can be done using a Taylor series, using a partial sum with a remainder term, for $f$. This analysis gives a strict upper bound on the error, if the derivatives of $f$ are available.

7. Algorithmic Proofs and Verified Calculations: The integration method can be combined with interval arithmetic to produce computer proofs and *verified* calculations.

## Integrals Over Infinite Intervals

1. Standard Techniques for Unbounded Intervals: Several methods exist for approximate integration over unbounded intervals. The standard technique involves specially derived quadrature rules, such as Gauss-Hermitian quadrature for integrals on the whole real line, and Gauss-Laguerre quadrature for the integrals on the positive reals (Leader (2004)).
2. Changing Variables to Bounded Intervals: Monte Carlo methods can be used, or a change of variables to a finite interval can be applied; e.g., for the whole line, one could use

$$\int_{-\infty}^{+\infty} f(x)dx = \int_{-1}^{+1} f\left(\frac{1}{1-t^2}\right)\frac{1+t^2}{(1-t^2)^2}dt$$

and for semi-infinite intervals one could use

$$\int_{a}^{+\infty} f(x)dx = \int_{0}^{+1} f\left(a+\frac{t}{1-t}\right)\frac{1}{(1-t)^2}dt$$

$$\int_{-\infty}^{a} f(x)dx = \int_{0}^{+1} f\left(a-\frac{1-t}{t}\right)\frac{1}{t^2}dt$$

as possible transformations.

## Multi-dimensional Integrals

1. Fubini's Theorem: Curse of Dimensionality: The quadrature rules discussed so far are al designed to compute one-dimensional integrals. To compute integrals in multiple dimensions, one approach is to phrase the multiple integrals as repeated one-dimensional integrals by applying the Fubini's theorem – the tensor product rule. This approach requires the function evaluations to grow exponentially as the number of dimensions increases. Three methods described below are known to overcome this so-called *curse of dimensionality*.
2. Multi-dimensional Cubature Integration Rules: A great many additional techniques for forming multi-dimensional cubature integration rules for a variety of weighting functions are given in Stroud (1971).

## Monte Carlo

1. Potential Accuracy Improvement from MC: Monte Carlo and quasi-Monte Carlo methods are easy to apply to multi-dimensional integrals. They may yield greater accuracy for the same number of function evaluations than repeated integrations using one-dimensional methods.
2. Markov Chain Monte Carlo Algorithms: A large class of useful Monte Carlo methods are the so-called Markov Chain Monte Carlo algorithms, which include the Metropolis-Hastings algorithms and Gibbs sampling.

## Sparse Grids

Sparse grids were originally developed by Smolyak for the quadrature of high-dimensional functions. The method is always based on a one-dimensional quadrature rule, but performs a more sophisticated combination of univariate results. However, whereas the tensor product rule guarantees that the weight of all the quadrature points will be positive, Smolyak's rule does not guarantee that the weights will be positive.

## Bayesian Quadrature

Bayesian quadrature is a statistical approach to the numerical problem of computing integrals and falls under the field of probabilistic numerics. It can provide a full handling of the uncertainty over the range of the integral expressed as a Gaussian process posterior variance. It is also known to provide very fast convergence rates which can be upto exponential in the number of quadrature points $n$ (Briol, Oates, Girolami, and Osborne (2015)).

## Connections to Differential Equations

1. The ODE Initial Value Problem: The problem of evaluating the integral $F(x) = \int_a^x f(u)du$ can be reduced to an initial value problem for an ordinary differential equation by applying the first part of the fundamental theorem of calculus.
2. Differential Form of the ODE: By differentiating both sides of the above with respect to the argument $x$, it can be seen that the function $F$ satisfies $\frac{dF(x)}{dx} = f(x)\ F(a) = 0$
3. Applying the ODE Solution Schemes: Methods for ordinary differential equations, such as Runge-Kutta schemes, can be applied to the re-stated problem and thus be used to evaluate the integral. For instance, the standard fourth order Runge-Kutta method applied to the differential equation yields the Simpson's rule from above.
4. Separating the Independent/Dependent Variables: The differential equation $F(x) = f(x)$ has a special form; the right-hand side contains only the dependent variable (here $x$) and not the independent variable (here $F$). This simplifies the theory and the algorithms considerably.

## References

- Briol, F. X., C. J. Oates, M. Girolami, and M. A. Osborne (2015): Frank-Wolfe Bayesian Quadrature: Probabilistic Integration with Theoretical Guarantees **arXiv**
- Forsythe, G. E., M. A. Malcolm, and C. B. Moler (1977): *Computer Methods for Mathematical Computation* **Prentice Hall** Englewood Cliffs NJ
- Leader, J. J. (2004): *Numerical Analysis and Scientific Computation* **Addison Wesley**
- Stoer, J., and R. Bulirsch (1980): *Introduction to Numerical Analysis* **Springer-Verlag** New York
- Stroud, A. H. (1971): *Approximate Calculation of Multiple Integrals* **Prentice Hall** Englewood Cliffs NJ
- Wikipedia (2019): Numerical Integration

# Gaussian Quadrature

## Introduction and Overview

1. Quadrature Rule in Numerical Analysis: In numerical analysis, a **quadrature rule** is the approximation of the definite integral of a function, usually stated as a weighted sum of the function values at the specified points within the domain of integration (Wikipedia (2019)).
2. n-Point Gaussian Quadrature Rule: An n-point **Gaussian Quadrature Rule**, named after Carl Friedrich Gauss, is a quadrature rule constructed to yield the exact result for polynomials of degree $2n - 1$ or less using a suitable choice of nodes $x_i$ and weights $w_i$ for

$$i = 1, \cdots, n$$

3. Gaussian Quadrature Nodes and Weights: The most common domain of integration for such a rule is taken as $[-1, +1]$, so the rule can be stated as

$$\int_{-1}^{+1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

which is exact for polynomials of degree $2n - 1$ or less. This exact rule is known as the Gauss-Legendre quadrature rule.

4. Approximating $f(x)$ by a $2n - 1$ Polynomial: The quadrature rule will only be an approximation to the integral above if $f(x)$ is well approximated by a polynomial of degree $2n - 1$ or less in $[-1, +1]$

5. Integrands with End-Point Singularities: The Gauss-Legendre quadrature rule is not typically used for integrable functions with end-point singularities.
6. Alternative Specification of the Quadrature Rules: Instead, of the integrand can be written as

$$f(x) = (1 - x)^{\alpha}(1 + x)^{\beta} g(x)$$

$$\alpha, \beta > -1$$

   where $g(x)$ is well-approximated by a low-degree polynomial, then the alternative nodes $x_i'$ and weights $w_i'$ will usually give more accurate quadrature rules.
7. The Gauss-Jacobi Quadrature Rules: These are known as Gauss-Jacobi quadrature rules, i.e.,

$$f(x) = (1 - x)^{\alpha}(1 + x)^{\beta} g(x) \approx \sum_{i=1}^{n} w_i' f(x_i')$$

8. Gauss-Chebyshev Quadrature Weights: Common weights include $\frac{1}{\sqrt{1-x^2}}$ - referred to as Chebyshev-Gauss – and $\sqrt{1 - x^2}$.
9. Gauss-Laguerre and Gauss-Hermite Quadrature Rules: One may also want to integrate over semi-infinite intervals – the Gauss-Laguerre quadrature – or infinite intervals – the Gauss-Hermite quadrature.
10. Quadrature Nodes as Roots of Orthogonal Polynomials: It can be shown (Stoer and Bulirsch (2002), Press, Teukolsky, Vetterling, and Flannery (2007)) that the quadrature nodes $x_i$ are the roots of a polynomial belonging to a class of orthogonal polynomials, i.e., they belong to the class that is orthogonal with respect to a weighted inner-product. This is a key observation for computing Gauss quadrature nodes and weights.

## Gauss-Legendre Quadrature

1. Legendre Polynomials as Associated Orthogonals: For the simplest integration problem stated above, i.e., where $f(x)$ is well-approximated by polynomials in $[-1, +1]$, the associated orthogonal polynomials are Legendre polynomials, denoted by $P_n(x)$.
2. Weights of the Legendre Terms: With $n^{th}$ polynomial normalized to give

$$P_n(1) = 1$$

the $i^{th}$ Gauss node, $x_i$, is the $i^{th}$ root of $P_n$, and the weights are given by the formula (Abramowitz and Stegun (2007))

$$w_i = \frac{2}{(1 - x_i)^2[P_n{}'(x_i)]^2}$$

3. Low Order Nodes and Weights: Some low order quadrature rules are tabulated below over $[-1, +1]$. The next section contains other intervals.

| Number of Points | Points $x_i$ | Approximate $x_i$ | Weight $w_i$ | Approximate $w_i$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 2 | 2 |
| 2 | $\pm\sqrt{\frac{1}{3}}$ | $\pm 0.57735$ | 1 | 1 |
| 3 | 0 | 0 | $\frac{8}{9}$ | 0.888889 |
|  | $\pm\sqrt{\frac{3}{5}}$ | $\pm 0.774597$ | $\frac{5}{9}$ | 0.555556 |

| | | | | |
|---|---|---|---|---|
| 4 | $\pm\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$ | $\pm 0.339981$ | $\frac{18+\sqrt{30}}{36}$ | 0.652145 |
| | $\pm\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$ | $\pm 0.861136$ | $\frac{18-\sqrt{30}}{36}$ | 0.347855 |
| 5 | 0 | 0 | $\frac{128}{225}$ | 0.568889 |
| | $\pm\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$ | $\pm 0.538469$ | $\frac{322+13\sqrt{70}}{900}$ | 0.478629 |
| | $\pm\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$ | $\pm 0.906180$ | $\frac{322-13\sqrt{70}}{900}$ | 0.236927 |

## Change of Interval

1. Interval Change - $[a, b]$ to $[-1, +1]$: The integral over $[a, b]$ must be changed to the integral over $[-1, +1]$ before applying the Gaussian quadrature rule. The change of interval can be done in the following way:

$$\int_{a}^{b} f(t)dt = \frac{b-a}{2}\int_{-1}^{+1} f\left(\frac{b-a}{2}x+\frac{b+a}{2}\right)dx$$

2. Application of Gaussian Quadrature Rules: Applying the Gaussian quadrature rule then results in the following approximation:

$$\int_a^b f(t)dt \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}x_i + \frac{b+a}{2}\right)$$

## Other Forms

1. Generalization of the Integrand Quadrature: The integration problem can be expressed in a slightly more general way by introducing a positive weight function $\omega$ into the integrand, and allowing an interval other than $[-1, +1]$. That is, the problem is to calculate $\int_a^b \omega(x)f(x)dx$ for some choices of $a$, $b$, $c$, and $\omega$.
2. Parameter Choices for the Quadrature Generation: For

$$a = -1$$

$$b = 1$$

and

$$\omega(x) = 1$$

the problem is the same as the one considered above. Other choices lead to other integration rules, some of which are tabulated below.

| **Interval** | $\boldsymbol{\omega(x)}$ | **Orthogonal Polynomials** |
|---|---|---|
| $[-1, +1]$ | $1$ | Legendre Polynomials |
| $(-1, +1)$ | $(1-x)^{\alpha}(1+x)^{\beta}, \alpha, \beta > -1$ | Jacobi Polynomials |
| $(-1, +1)$ | $\frac{1}{\sqrt{1-x^2}}$ | Chebyshev Polynomials (First Kind) |

| | | |
|---|---|---|
| $[-1,+1]$ | $\sqrt{1-x^2}$ | Chebyshev Polynomials (Second Kind) |
| $[0,\infty)$ | $e^{-x}$ | Laguerre Polynomials |
| $[0,\infty)$ | $x^{\alpha}e^{-x}, \alpha > -1$ | Generalized Laguerre Polynomials |
| $(-\infty,+\infty)$ | $e^{-x^2}$ | Hermite Polynomials |

## Fundamental Theorem

1. n-Degree Polynomial Driving the Quadrature: Let $p_n$ be a non-trivial polynomial of degree $n$ such that

$$\int_{a}^{b} \omega(x) x^k p_n(x) dx = 0$$

for all

$$k = 0, \cdots, n-1$$

2. Nodes as Roots of the Polynomial: If the $n$ nodes $x_i$ are picked to be the zeros of $p_n$, then there exist $n$ weights $w_i$ which make the Gauss quadrature computed integral exact for polynomials $h(x)$ of degree $2n-1$ or less. Furthermore, all these nodes $x_i$ lie in the open interval $(a, b)$ (Stoer and Bulirsch (2002)).
3. Chosen as the Orthogonal Polynomial: The polynomial $p_n$ is said to be an orthogonal polynomial of degree $n$ associated with the weight function $\omega(x)$. It is unique to a constant normalization factor.

4. Decomposing $h(x)$ into Orthogonal Polynomials: The idea underlying the proof is that, because of its sufficiently low degree, $h(x)$ can be divided by $p_n(x)$ to produce a quotient $q(x)$ strictly lower than $n$, and a remainder of still lower degree, so that both will be orthogonal to $p_n(x)$, by the defining property of $p_n(x)$. Thus,

$$\int_a^b \omega(x)h(x)dx = \int_a^b \omega(x)r(x)dx$$

5. Quadrature Over Reduced Degree Polynomial: Because of the choice of nodes $x_i$, the corresponding relation

$$\sum_{i=1}^{n} w_i h(x_i) = \sum_{i=1}^{n} w_i r(x_i)$$

also holds. The exactness of the computed integral $h(x)$ then follows from the exactness of $r(x)$, i.e., for polynomials of degree $n$ or less.

## General Formula for the Weights

1. Generic Expression for the Quadrature Weights: The weights can be expressed as

$$w_i = \frac{a_n}{a_{n-1}} \frac{\int_a^b \omega(x){p_{n-1}}^2(x)dx}{{p_n}'(x_i)p_{n-1}(x_i)}$$

where $a_k$ is the coefficient of $x^k$ in $p_n(x)$.

2. Lagrange Polynomial Form for $r(x)$: To prove this, note that using the Lagrange interpolation, one can express $r(x)$ in terms of $r(x_i)$ as

$$r(x) = \sum_{i=1}^{n} r(x_i) \prod_{\substack{1 \le j \le n \\ j \ne i}} \frac{x - x_j}{x_i - x_j}$$

because $r(x)$ has a degree less than $n$ and is thus fixed by the value it attains at $n$ different points.

3. <u>Re-evaluation of the $r(x)$ Quadrature</u>: Multiplying both sides by $\omega(x)$ and integrating from $a$ to $b$ yields

$$\int_{a}^{b} \omega(x) r(x) dx = \sum_{i=1}^{n} r(x_i) \int_{a}^{b} \omega(x) \prod_{\substack{1 \le j \le n \\ j \ne i}} \frac{x - x_j}{x_i - x_j} dx$$

4. <u>Quadrature Weights Using Lagrange Polynomials</u>: The weights $w_i$ are thus given by

$$w_i = \int_{a}^{b} \omega(x) \prod_{\substack{1 \le j \le n \\ j \ne i}} \frac{x - x_j}{x_i - x_j} dx$$

5. <u>Quadrature Weights Using Orthogonal Polynomials</u>: This integral expression for $w_i$ can be expressed in terms of the orthogonal polynomials $p_n(x)$ and $p_{n-1}(x)$ as follows.
6. <u>Orthogonal Polynomial from Lagrange Numerator</u>: One can write

$$\prod_{\substack{1 \le j \le n \\ j \ne i}} (x - x_j) = \frac{\prod_{1 \le j \le n} (x - x_j)}{x - x_i} = \frac{p_n(x)}{a_n(x - x_i)}$$

where $a_n$ is the coefficient of $x^n$ in $p_n(x)$.

7. <u>Orthogonal Derivative from Lagrange Denominator</u>: Taking the limit of $x$ to $x_i$ yields, using L'Hopital's rule

$$\prod_{\substack{1 \le j \le n \\ j \ne i}} \left(x_i - x_j\right) = \frac{p_n{}'(x_i)}{a_n}$$

8. Weights from Polynomials and Derivatives: The integral expression for the weights can thus be written as

$$w_i = \frac{1}{p_n{}'(x_i)} \int_a^b \omega(x) \frac{p_n(x)}{x - x_i} dx$$

9. Reducing the Degree of Integrand: In the integrand, writing

$$\frac{1}{x - x_i} = \frac{1 - \left(\frac{x}{x_i}\right)^k}{x - x_i} + \left(\frac{x}{x_i}\right)^k \frac{1}{x - x_i}$$

yields

$$\int_a^b \omega(x) x^k \frac{p_n(x)}{x - x_i} dx = x_i{}^k \int_a^b \omega(x) \frac{p_n(x)}{x - x_i} dx$$

provided

$$k \le n$$

because $\frac{1 - \left(\frac{x}{x_i}\right)^k}{x - x_i}$ is a polynomial of degree $k - 1$ which is then orthogonal to $p_n(x)$.

10. Product of Lower Degree Polynomials: So, if $t(x)$ is a polynomial of at most degree $n$, one has

$$\int_{a}^{b} \omega(x)\frac{p_n(x)}{x - x_i}dx = \frac{1}{t(x_i)}\int_{a}^{b} \omega(x)\frac{t(x)p_n(x)}{x - x_i}dx$$

11. $\frac{p_n(x)}{x-x_i}$ as $n - 1$ Degree Polynomial: The integral on the right-hand side can be evaluated for

$$t(x) = p_{n-1}(x)$$

as follows. Because $\frac{p_n(x)}{x-x_i}$ is a polynomial of degree $n - 1$, one has

$$\frac{p_n(x)}{x - x_i} = a_n x^{n-1} + s(x)$$

where $s(x)$ is a polynomial of degree $n - 2$.

12. $n - 1$ Polynomial in the Weight Integral: Since $s(x)$ is orthogonal to $p_{n-1}(x)$, one has

$$\int_{a}^{b} \omega(x)\frac{p_n(x)}{x - x_i}dx = \frac{a_n}{p_{n-1}(x_i)}\int_{a}^{b} \omega(x)p_{n-1}(x)x^{n-1}dx$$

13. $x^{n-1}$ in Terms of $p_{n-1}$: One can then write

$$x^{n-1} = \left[x^{n-1} - \frac{p_{n-1}(x)}{a_{n-1}}\right] + \frac{p_{n-1}(x)}{a_{n-1}}$$

The term in the brackets is a polynomial of degree $n - 2$, which is therefore orthogonal to $p_{n-1}(x)$.

14. Weight Integral in Terms of ${p_{n-1}}^2$: The integral can thus be written as

$$w_i = \frac{a_n}{a_{n-1} p_{n-1}(x_i)} \int_a^b \omega(x) {p_{n-1}}^2(x) dx$$

15. Recovery of the Postulated Expression: According to

$$w_i = \frac{1}{p_n{}'(x_i)} \int_a^b \omega(x) \frac{p_n(x)}{x - x_i} dx$$

the weights are obtained by dividing the weight integral above by $p_n{}'(x_i)$, and that yields the expression

$$w_i = \frac{a_n}{a_{n-1}} \frac{\int_a^b \omega(x) {p_{n-1}}^2(x) dx}{p_n{}'(x_i) p_{n-1}(x_i)}$$

16. Alternate Expression Using the $p_{n+1}$ Term: $w_i$ can also be expressed in terms of the orthogonal polynomials $p_n(x)$ and $p_{n+1}(x)$. In a 3-term recurrence relation (see below)

$$p_{n+1}(x_i) = \rho_a p_n(x_i) + \rho_b p_{n-1}(x_i)$$

the $p_n(x_i)$ term vanishes, so $p_{n-1}(x_i)$ in

$$w_i = \frac{a_n}{a_{n-1}} \frac{\int_a^b \omega(x) {p_{n-1}}^2(x) dx}{p_n{}'(x_i) p_{n-1}(x_i)}$$

can be replaced by $\frac{p_{n+1}(x_i)}{\rho_b}$.

## Proof that the Weights are Positive

1. Remainder – Lagrange Squared Term Polynomial: Consider the following polynomial of degree $2n - 2$

$$l(x) = \prod_{\substack{1 \le j \le n \\ j \ne i}} \left( \frac{x - x_j}{x_i - x_j} \right)^2$$

where, as above, the $x_j$ are the roots of the polynomial $p_n(x)$.

2. Gaussian Quadrature for the above Polynomial: Clearly

$$l(x_j) = \delta_{ij}$$

Since the degree $l(x)$ is less than $2n - 1$, the Gaussian quadrature formula involving the weights and the nodes obtained from $p_n(x)$ applies.

3. Weights Have to be Positive: Since

$$l(x_j) = 0$$

for $j$ not equal to $i$, one has

$$\int_a^b \omega(x) l(x) dx = \sum_{j=1}^{N} w_j l(x_j) = \sum_{j=1}^{N} w_j \delta_{ij} = w_i > 0$$

Since both $\omega(x)$ and $l(x)$ are non-negative functions, it follows that

$$w_i > 0$$

## Computation of Gaussian Quadrature Rules

There are many algorithms for computing the nodes $x_i$ and the weights $w_i$ of Gaussian quadrature rules. The popular are the Golub-Welsch algorithm requiring $\mathcal{O}(n^2)$ operations, Newton's method for solving

$$p_n(x) = 0$$

using the three-term recurrence relation for evaluation requiring $\mathcal{O}(n^2)$ operations, and asymptotic formulas for large $n$ requiring $\mathcal{O}(n)$ operations.

## Recurrence Relation

1. Recurrence Relation for Orthogonal Polynomials: Orthogonal polynomials $p_r$ with

   $$[\![p_r, p_s]\!] = 0$$

   for

   $$r \neq s$$

   for a scalar product $[\![\cdot]\!]$,

   $$Degree(p_r) = r$$

   and leading coefficient one – i.e., monic orthogonal polynomials – satisfy the recurrence relation

$$p_{r+1}(x) = \left(x - a_{r,r}\right)p_r(x) - a_{r,r-1}p_{r-1}(x) - \cdots - a_{r,0}p_0(x)$$

where the scalar product is defined as

$$[\![p_r(x), p_s(x)]\!] = \int_a^b \omega(x) p_r(x) p_s(x) dx$$

for

$$r = 0, \cdots, n - 1$$

where $n$ is the upper bound on the degree – which can be taken to infinity – and where

$$a_{r,s} = \frac{[\![xp_r, p_s]\!]}{[\![p_r, p_s]\!]}$$

2. Proof of Recurrence using Induction: First of all, the polynomials defined by the recurrence relation starting with

$$p_0(x) = 1$$

have a leading coefficient of one and the correct degree (i. e, 0). Setting the starting polynomial by $p_0$, the orthogonality of $p_r$ can be demonstrated by induction.

3. Proof for the First Term: For

$$r = s = 0$$

one has

$$\llbracket p_1, p_0 \rrbracket = (x - a_{0,0})\llbracket p_0, p_0 \rrbracket = \llbracket xp_0, p_0 \rrbracket - a_{0,0}\llbracket p_0, p_0 \rrbracket = \llbracket xp_0, p_0 \rrbracket - \llbracket xp_0, p_0 \rrbracket$$
$$= 0$$

4. Proof for an Arbitrary Term: Now, if $p_0, \cdots, p_r$ are orthogonal, so is $p_{r+1}$, because in

$$\llbracket p_{r+1}, p_s \rrbracket = \llbracket xp_r, p_s \rrbracket - a_{r,r}\llbracket p_r, p_s \rrbracket - a_{r,r-1}\llbracket p_{r-1}, p_s \rrbracket - \cdots - a_{r,0}\llbracket p_0, p_0 \rrbracket$$

all scalar products vanish except for the first one and the one where $p_s$ meets the same orthogonal polynomial. Therefore,

$$\llbracket p_{r+1}, p_s \rrbracket = \llbracket xp_r, p_s \rrbracket - a_{r,s}\llbracket p_r, p_s \rrbracket = \llbracket xp_r, p_s \rrbracket - \llbracket xp_r, p_s \rrbracket = 0$$

5. Reduction to Three Term Recurrence: However, if the scalar product satisfies

$$\llbracket xp_r, p_s \rrbracket = \llbracket p_r, xp_s \rrbracket$$

- which is the case for Gaussian Quadrature – the above recurrence relation reduces to a three-term recurrence relation.

6. Serial Zeroing of Recurrence Terms: For

$$s < r - 1$$

$xp_s$ is a polynomial of degree less than or equal to $r - 1$. On the other hand, $p_r$ is orthogonal to every polynomial of degree less than or equal to $r - 1$. Therefore, one has

$$\llbracket xp_r, p_s \rrbracket = \llbracket p_r, xp_s \rrbracket = 0$$

and

$$a_{r,s} = 0$$

for

$$s < r - 1$$

7. Full Form of the Recurrence Relation: The recurrence relation then simplifies to

$$p_{r+1}(x) = \left(x - a_{r,r}\right)p_r(x) - a_{r,r-1}p_{r-1}(x)$$

or, with the convention

$$p_{-1}(x) \equiv 0$$

$$p_{r+1}(x) = (x - a_r)p_r(x) - b_r p_{r-1}(x)$$

where

$$a_r \doteqdot \frac{\llbracket xp_r, p_r \rrbracket}{\llbracket p_r, p_r \rrbracket}$$

$$b_r \doteqdot \frac{\llbracket xp_r, p_{r-1} \rrbracket}{\llbracket p_{r-1}, p_{r-1} \rrbracket} = \frac{\llbracket p_r, p_r \rrbracket}{\llbracket p_{r-1}, p_{r-1} \rrbracket}$$

where the final step occurs because

$$\llbracket xp_r, p_{r-1} \rrbracket = \llbracket p_r, xp_{r-1} \rrbracket = \llbracket p_r, p_r \rrbracket$$

since $xp_r$ differs from $p_r$ by a degree less than $r$.

## The Golub-Welsch Algorithm

1. <u>Three-Term Jacobi Recurrence Matrix</u>: The three-term recurrence relation can be written in the matrix form

   $$J\tilde{P} = x\tilde{P} - P_n(x) \times \hat{e}_n$$

   where

   $$\tilde{P} = [P_0(x), \cdots, P_{n-1}(x)]^T$$

   $\hat{e}_n$ is the $n^{th}$ standard basis vector, i.e.

   $$\hat{e}_n = [0, \cdots, 0, 1]^T$$

   and $J$ is the so-called Jacobi matrix.

   $$J = \begin{bmatrix} a_0 & 1 & 0 & \cdots & \cdots & \cdots \\ b_1 & a_1 & 1 & 0 & \cdots & \cdots \\ 0 & b_2 & a_2 & 1 & 0 & \cdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \cdots & \cdots & 0 & b_{n-2} & a_{n-2} & 1 \\ \cdots & \cdots & \cdots & 0 & b_{n-1} & a_{n-1} \end{bmatrix}$$

2. <u>The Golub-Welsch Algorithm</u>: The zeroes $x_j$ of the polynomials upto degree $n$, which are used as nodes for the Gaussian quadrature, can be found by computing the eigenvalues of this tri-diagonal matrix. This procedure is known as the Golub-Welsch algorithm.
3. <u>Elements of the Tri-diagonal Matrix</u>: For computing the weights and the nodes, it is preferable to consider the symmetric tridiagonal matrix $\mathcal{J}$ with elements

$$\mathcal{J}_{i,i} = J_{i,i} = a_{i-1}$$

$$i = 1, \cdots, n$$

$$\mathcal{J}_{i-1,i} = \mathcal{J}_{i,i-1} = \sqrt{J_{i,i-1} J_{i-1,i}} = \sqrt{b_{i-1}}$$

$$i = 2, \cdots, n$$

4. Quadrature Nodes from the Eigenvalues: $J$ and $\mathcal{J}$ are similar matrices, and therefore have the same eigenvalues – the quadrature nodes.
5. Quadrature Weights Extracted from Eigenvectors: The weights can be computed from the corresponding eigenvectors. If $\phi_j$ is a normalized eigenvector – i.e., an eigenvector with Euclidean norm equal to one – associated with the eigenvalue $x_j$, the corresponding weight can be computed from the first component of this eigenvector, namely:

$$w_j = \mu_0 \left[\phi_j(1)\right]^2$$

where $\mu_0$ is the integral of the weight function

$$\mu_0 = \int_a^b \omega(x) dx$$

Gil, Segura, and Temme (2007) contain further details.

## Error Estimates

1. Stoer-Bulirsch (2002) Error Estimate: Using the analysis presented in Stoer and Bulirsch (2002), the error of a Gaussian quadrature can be stated as follows. For an integrand that has $2n$ continuous derivatives,

$$\int_a^b \omega(x)f(x)dx - \sum_{i=1}^{n} w_i f(x_i) = \frac{f^{(2n)}(\xi)}{(2n)!}[\![p_n, p_n]\!]$$

for some $\xi$ in $(a, b)$, where $p_n$ is the monic orthogonal polynomial of degree $n$, and

$$[\![f(x), g(x)]\!] = \int_a^b \omega(x)f(x)g(x)dx$$

2. Kahaner, Moler, and Nash (1989) Error Estimate: In the important special case of

$$\omega(x) = 1$$

one has the error estimate (Kahaner, Moler, and Nash (1989))

$$\int_a^b \omega(x)f(x)dx - \sum_{i=1}^{n} w_i f(x_i) = \frac{(b-a)^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{(2n)}(\xi)$$

$$a < \xi < b$$

3. Conservative Nature of the Estimate: Stoer and Bulirsch (2002) remark that this error estimate is inconvenient in practice, since it may be difficult to estimate the order $2n$ derivatives. Furthermore, the actual error may be much less that the bound established by the derivative.
4. Error Estimate Using Different Orders: Another approach is to use two Gaussian quadrature rules of different orders, and to estimate the error as the difference

between these two results. For this purpose, the Gauss-Kronrod quadrature rules can be useful.

## Gauss-Kronrod Rules

1. Sub-divided Points do not Coincide: If the interval $[a, b]$ is sub-divided, the Gaussian evaluation points of the new sub-interval never coincide with the previous evaluation points – except at zero for odd numbers – and thus the integrand must be evaluated at every point.
2. Extensions to Gauss Quadrature Rules: *Gauss-Kronrod rules* are extensions to Gauss quadrature rules generated by adding $n + 1$ points to a $n$-point rule in such a way that the resulting rule is of the order $2n + 1$. This allows for computing higher-order estimates while re-using the function values of the lower-order estimates.
3. Estimation of the Quadrature Error: The difference between the Gauss quadrature rule and its Kronrod extension is often used an estimate of the approximation error.

## Gauss-Lobatto Rules

1. Gaussian vs. Lobatto Quadrature Rules: Also known as *Lobatto quadrature* (Abramowitz and Stegun (2007)), Gauss-Lobatto quadrature is named after the Dutch mathematician Rehuel Lobatto. It is similar to the Gaussian quadrature, except for the following differences:
    a. The integration points include the end-points of the integration interval.
    b. It is accurate for polynomials upto degree $2n - 3$, where $n$ is the number of integration points (Quatteroni, Sacco, and Saleri (2000)).
2. Gauss-Lobatto Quadrature Function Expression: Lobatto quadrature of function $f(x)$ in an interval is $\int_{-1}^{+1} f(x)dx = \frac{2}{n(n-1)}[f(-1) + f(+1)] + \sum_{i=2}^{N} w_i f(x_i) + R_N$

3. Gauss-Lobatto Quadrature Abscissa: The abscissae$x_i$ is the $(i-1)^{st}$ zero of $P_{n-1}{}'(x)$.

4. Gauss-Lobatto Quadrature Weights: $w_i = \frac{2}{n(n-1)[P_{n-1}(x_i)]^2}$ $x_i \neq \pm 1$

5. Gauss-Lobatto Quadrature Error Remainder: $R_N = \frac{-n(n-1)^3 2^{2n-1}[(n-2)!]^4}{(2n-1)[(2n-2)!]^3} f^{(2n-2)}(\xi)$ $-1 < \xi < +1$

6. Low Order Gauss-Lobatto Quadrature Weight Sample:

| **Number of Points $n$** | **Points $x_i$** | **Weights $w_i$** |
|---|---|---|
| 3 | $0$ | $\frac{4}{3}$ |
| | $\pm 1$ | $\frac{1}{3}$ |
| 4 | $\pm\sqrt{\frac{1}{5}}$ | $\frac{5}{6}$ |
| | $\pm 1$ | $\frac{1}{6}$ |
| 5 | $0$ | $\frac{32}{45}$ |
| | $\pm\sqrt{\frac{3}{7}}$ | $\frac{49}{90}$ |
| | $\pm 1$ | $\frac{1}{10}$ |
| 6 | $\pm\sqrt{\frac{1}{3} - \frac{2\sqrt{7}}{21}}$ | $\frac{14+\sqrt{7}}{30}$ |
| | $\pm\sqrt{\frac{1}{3} + \frac{2\sqrt{7}}{21}}$ | $\frac{14-\sqrt{7}}{30}$ |
| | $\pm 1$ | $\frac{1}{15}$ |
| 7 | $0$ | $\frac{256}{525}$ |

| | | |
|---|---|---|
| | $\pm\sqrt{\frac{5}{11}-\frac{2}{11}\sqrt{\frac{5}{3}}}$ | $\frac{124+7\sqrt{15}}{350}$ |
| | $\pm\sqrt{\frac{5}{11}+\frac{2}{11}\sqrt{\frac{5}{3}}}$ | $\frac{124-7\sqrt{15}}{350}$ |
| | $\pm 1$ | $\frac{1}{21}$ |

7. Implementation of the Adaptive Variant: An adaptive variant of this algorithm with 2 interior nodes (Gander and Gautschi (2000)) is found in GNU Octave and in MATLAB as *quadl* and *intergate*, respectively.

## References

- Abramowitz, M., and I. A. Stegun (2007): *Handbook of Mathematics Functions* **Dover Book on Mathematics**
- Gander, W., and W. Gautschi (2000): Adaptive Quadrature – Revisited *Bit Numerical Mathematics* **40 (1)** 84-101
- Gil, A., J. Segura, and N. M. Temme (2007): *Numerical Methods for Special Functions* **Society for Industrial and Applied Mathematics** Philadelphia
- Kahaner, D., C. Moler, and S. Nash (1989): *Numerical Methods and Software* **Prentice Hall**
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (2007): *Numerical Recipes: The Art of Scientific Computing 3$^{rd}$ Edition* **Cambridge University Press** New York
- Quatteroni, A., R. Sacco, and F. Saleri (2000): *Numerical Mathematics* **Springer Verlag** New York

- Stoer, J., and R. Bulirsch (2002): *Introduction to Numerical Analysis 3rd Edition* **Springer**
- Wikipedia (2019): Gaussian Quadrature

# Gauss-Kronrod Quadrature

## Introduction and Overview

1. Adaptive Method for Numerical Integration: The *Gauss-Kronrod quadrature formula* is an adaptive method for numerical integration.
2. Improved Accuracy by Re-using Less Accurate Results: It is a variant of the Gaussian quadrature, in which the evaluation points are chosen so that an accurate approximation can be computed by re-using the information produced by the computation of a less accurate approximation (Wikipedia (2019)).
3. Multi-Order Quadrature Rules and Errors: It is an example of what is called a nested quadrature rule; for the same set of function evaluation points, it has two quadrature rules, one higher order and one lower order – the latter is called an *embedded* rule. The difference between these two approximations is used to estimate the calculation error of the integral.

## Description

1. Numerical Approximation of Definite Integrals: The problem in numerical integration is to approximate $\int_a^b f(x)dx$
2. Use of n Point Gaussian Quadrature: Such integrals can be approximated, for example, by n-point Gaussian quadrature

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

where $x_i$ and $w_i$ are the points and the weights used to evaluate the function $f(x)$.

3. Consequences of Non-Matching Nodes: If the interval $[a, b]$ is sub-divided, the Gauss evaluation points of the new sub-intervals never coincide with the previous evaluation points – except at the mid-point for odd numbers of evaluation points – and thus the integrand must be evaluated at every point.
4. Node Extensions using Stieltjes Polynomials: Gauss-Kronrod formulas are extensions of the Gauss quadrature formulas generated by adding $n + 1$ points to a $n$ point rule in such a way that the resulting rule is of order $2n + 1$ (Laurie (1997)); the corresponding Gauss rule is order $2n - 1$. The extra points are zeros of Stieltjes polynomials.
5. Kronrod Extension as an Error Estimate: This allows for computing higher-order error estimates while re-using the function values of a lower order estimate. The difference between a Gauss quadrature rule and its Kronrod extension is used often as an estimate of the approximation error.

## Example

1. 7 Point Gauss Plus 15 Point Kronrod: A popular example combines a 7-point Gauss rule with a 15-point Kronrod rule (Kahaner, Moler, and Nash (1989)). Because the Gauss points are incorporated into the Kronrod points, a total of only 15 function evaluations are needed.
2. $(G7, K15)$ on $[-1, +1]$:

| Gauss Nodes | Weights |
|---|---|
| $\pm 0.94910\ \ 79123\ \ 42759$ | $0.12948\ \ 49661\ \ 68870$ |
| $\pm 0.74153\ \ 11855\ \ 99394$ | $0.27970\ \ 53914\ \ 89277$ |
| $\pm 0.40584\ \ 51513\ \ 77397$ | $0.38183\ \ 00505\ \ 05119$ |
| $\pm 0.00000\ \ 00000\ \ 00000$ | $0.41795\ \ 91836\ \ 73469$ |

| Kronrod Nodes | Weights |
|---|---|
| ±0.99145 53711 20813 | 0.02293 53220 10529 |
| ±0.94910 79123 42759 | 0.06309 20926 29979 |
| ±0.86486 44233 59769 | 0.10479 00103 22250 |
| ±0.74153 11855 99394 | 0.14065 32597 15525 |
| ±0.58608 72354 67691 | 0.16900 47266 39267 |
| ±0.40584 51513 77397 | 0.19035 05780 64785 |
| ±0.20778 49550 07898 | 0.20443 29400 75298 |
| ±0.00000 00000 00000 | 0.20948 21410 84728 |

3. Use in Quadrature Error Estimate: The integral is then estimated by the Kronrod rule $K15$ and the error can be estimated as $|G7 - K15|$.
4. Enhancements to the Quadrature Algorithm: Patterson (1968) showed how to find further extensions of this type. Monegato (1978) and Piessens, de Doncker-Kapenga, Uberhuber, and Kahaner (1983) proposed improved algorithms. Finally, the most efficient algorithm was proposed by Laurie (1997).
5. Tabulation of Quadruple Precision Coefficients: Quadruple Precision (34 decimal digits) coefficients for $(G7, K15)$, $(G10, K21)$, $(G15, K31)$, $(G20, K41)$, and others are computed and tabulated in Holoborodko (2011).

## Implementations

Routines for Gauss-Kronrod quadrature are provided by the QUADPACK library, the GNU Scientific Library, the NAG Numerical Libraries R, the C++ Boost Library, and DROP.

## References

- Holoborodko, P. (2011): Gauss-Kronrod Quadrature Nodes and Weights
- Kahaner, D., C. Moler, and S. Nash (1989): *Numerical Methods and Software* **Prentice Hall**
- Laurie, D. (1997): Calculation of Gauss-Kronrod Quadrature Rules *Mathematics of Computation* **66 (219)** 1133-1145
- Monegato, G. (1978): Some Remarks on the Construction of Extended Gaussian Quadrature Rules *Mathematics of Computation* **32 (141)** 247-252
- Patterson, T. N. L. (1968): The Optimum Addition of Points to Quadrature Formulae *Mathematics of Computation* **22 (104)** 847-856
- Piessens, R., E. de Doncker-Kapenga, C. W. Uberhuber, and D. K. Kahaner (1983): *QUADPACK – A Subroutine Package for Automatic Integration* **Springer-Verlag**
- Wikipedia (2019): Gauss-Kronrod Quadrature Formula

# Gamma Function

## Introduction and Background

1. Extension to the Factorial Function: The *gamma function* – represented by Γ - is one of a number of extensions to the factorial function with its argument shifted down by 1, to real and complex numbers (Wikipedia (2019)).
2. First Expression for Gamma Function: Derived by Daniel Bernoulli, if $n$ is positive,

$$\Gamma(n) = (n-1)!$$

   Although other extensions do exist, this particular function is the most popular and useful.

3. Defined over the Full Complex Plane: The gamma function is defined for all complex numbers except for non-positive numbers.
4. Second Expression for Gamma Function: For complex numbers with a real positive part, it is defined in the convergent improper integral

$$\Gamma(z) = \int_{0}^{\infty} x^{z-1} e^{-x} dx$$

5. Analytic Continuation onto Complex Plane: This integral function is extended by analytic continuation to all complex numbers except for non-positive integers – where this function has simple poles – yielding the meromorphic function that is now known as the gamma function.

6. Third Expression for Gamma Function: It has no zeros, so the reciprocal function $\frac{1}{\Gamma(z)}$ is a holomorphic function. In fact, the gamma function corresponds to the Mellin transform of the negative exponential function

$$\Gamma(z) = \{\mathcal{M}e^{-x}\}(z)$$

7. Applicable Domain of Gamma Function: The gamma function is a component in various probability distribution functions, and as such, it is applicable in the fields of probability and statistics, as well as combinatorics.

## Motivation

1. Gamma Function as Interpolation Problem: The gamma function can be seen as a solution to the following interpolation problem: *Find a smooth curve that connects the points* $(x, y)$ *given by*

$$y = (x - 1)!$$

   *at the positive integer values for* $x$.
2. Expression Representing the Factorial Function: A plot of the first few factorials makes it clear that such a curve can be drawn, but it would be preferable to have an expression that describes the curve, in which the number of operations do not depend on the size of $x$.
3. Inadequacy of Integer Factorial Expression: The simple formula

$$x! = 1 \times 2 \times \cdots \times x$$

   cannot be used for then factorial values of $x$ since it works only when $x$ is a natural number, i.e., a positive integer.

4. Factorial Representation for Real Numbers: There are, relatively speaking, no such simple solutions for factorials. No finite combinations, of sums, products, powers, exponential functions, or logarithms will suffice to express $x!$, but it is possible to find a general formula for factorials using tools from calculus such as integrals and limits. A good solution to this is the gamma function (Davis (1959)).
5. Infinite Solutions to the Problem: There are infinitely many continuous extensions to the factorials of non-integers; infinitely many curves can be drawn through any of isolated points.
6. Ways to Characterize Gamma Function: The gamma function is the most useful solution in practice, being analytic – except at non-positive integers – and can be characterized in several ways.
7. Origin of the Non-Uniqueness: However, it is not the only analytic function that extends the factorial, as adding it to any analytics function that is zero at the non-positive integers, such as $k \sin m\pi x$, with give another function with that property (Davis (1959)).
8. First Criterion for the Function: A more restrictive property than satisfying the above interpolation is to satisfy the recurrence relation defining a translated version of the factorial function;

$$f(1) = 1$$

$$f(x+1) = xf(x)$$

   for $x$ equal to any positive integer.
9. Insufficiency of the First Criterion: But this would allow for multiplication by any periodic analytic function which evaluates to one on the non-positive integers, such as $e^{k \sin m\pi x}$.
10. Conditions of Bohr-Mollerup Theorem: There is a final way to address all this ambiguity; Bohr-Mollerup theorem states that when the condition that $f$ be

logarithmically convex – or *super-convex* (Kingman (1961)) - is added, it uniquely determines $f$ for positive, real inputs.

*11.* Extension to Real/Complex Numbers: From there, the gamma function can be extended to all real and complex values – except the negative integers and zero – by using the unique analytic continuation of $f$.

*12.* Asymptotic Requirement of the Euler Product: Also see Euler's infinite product definition below where the properties

$$f(1) = 1$$

and

$$f(x + 1) = xf(x)$$

together with the asymptotic requirement that

$$\underset{n \to \infty}{Limit} \frac{(n - 1)!\, n^x}{f(n + x)} = 1$$

uniquely define the same function.

## Main Definition

1. Euler Integral of Second Kind: The notation $\Gamma(z)$ is due to Legendre (Davis (1959)). If the real part of the complex number $z$ is positive –

$$Re(z) > 0$$

– then the integral

$$\Gamma(z) = \int_{0}^{\infty} x^{z-1} e^{-x} dx$$

converges absolutely, and is known as the *Euler Integral of the Second Kind* – the Euler Integral of the First Kind defines the beta function (Davis (1959)).

2. Integrating Gamma Function by Parts: Using integration by parts, one sees that

$$\Gamma(z+1) = \int_{0}^{\infty} x^{z} e^{-x} dx = [-x^{z} e^{-x}]_{0}^{\infty} + \int_{0}^{\infty} z x^{z-1} e^{-x} dx$$

$$= \underset{x \to \infty}{Limit} [-x^{z} e^{-x}] - [0 e^{-0}] + z \int_{0}^{\infty} x^{z-1} e^{-x} dx$$

3. Recovery of the Recurrence Relation: Recognizing that

$$\underset{x \to \infty}{Limit} [-x^{z} e^{-x}] \to 0$$

$$\Gamma(z+1) = z \int_{0}^{\infty} x^{z-1} e^{-x} dx = z\Gamma(z)$$

4. $n = 1$ Limit of the Gamma Function: One can calculate $\Gamma(1)$ as

$$\Gamma(1) = \int_{0}^{\infty} x^{1-1} e^{-x} dx = [-e^{-x}]_{0}^{\infty} = \underset{x \to \infty}{Limit} [-e^{-x}] - [e^{-0}] = 0 - (-1) = 1$$

5. Proof by Induction for $n > 1$: Given that

$$\Gamma(1) = 1$$

and

$$\Gamma(n+1) = n\Gamma(n)$$

$$\Gamma(n) = 1 \cdot 2 \cdot 3 \cdots (n-1) = (n-1)!$$

for all positive integers $n$. This can be seen as an example of proof by induction.

6. Meromorphic Extension to Complex Plane: The identity

$$\Gamma(z) = \frac{\Gamma(z+1)}{z}$$

can be used – or yielding the same result, the analytic continuity can be used – to uniquely extend the integral formulation of $\Gamma(z)$ to a meromorphic function defined for all complex numbers $z$, except integers less than or equal to zero (Davis (1959)). It is this extended version that is commonly referred to as the gamma function (Davis (1959)).

## Alternate Definitions: Euler's Definition as an Infinite Product

1. Approximating $z!$ form Complex Numbers: When seeking to approximate $z!$ For a complex number $z$, it turns out that it is effective to compute first $n!$ For some large integer $n$, and then use that value to approximate a value for $(n+z)!$, and then use the recursion relation

$$m! = m(m-1)!$$

backwards $n$ times, to unwind it to an approximation for $z!$. Furthermore, this approximation is exact in the limit as $n$ goes to infinity.

2. <u>Integer Limits Extension to Complex Numbers</u>: Specifically, for a fixed integer $m$, it is the case that

$$\underset{n \to \infty}{Limit} \frac{n!\,(n+1)^m}{(n+m)!} = 1$$

and one can ask if the same expression is obeyed when the arbitrary integer $m$ is replaced by the arbitrary complex number $z$

$$\underset{n \to \infty}{Limit} \frac{n!\,(n+1)^z}{(z+m)!} = 1$$

3. <u>Infinite Product Expression for $z!$</u>: Multiplying both sides by $z!$ gives

$$z! = \underset{n \to \infty}{Limit}\, n! \frac{z!}{(z+m)!} (n+1)^z = \underset{n \to \infty}{Limit} \frac{1 \cdots n}{(1+z) \cdots (n+z)} (n+1)^z$$

$$= \underset{n \to \infty}{Limit} \frac{1 \cdots n}{(1+z) \cdots (n+z)} \left[\left(1+\frac{1}{1}\right)\left(1+\frac{1}{2}\right) \cdots \left(1+\frac{1}{n}\right)\right]^z$$

$$= \prod_{n=1}^{\infty} \left[\frac{1}{\left(1+\frac{z}{n}\right)} \left(1+\frac{1}{n}\right)^z\right]$$

4. <u>Convergence of the Infinite Product</u>: This infinite product converges for all complex $z$ except negative integers, which fails because using the recurrence relation

$$m! = m(m-1)!$$

backwards through the value

$$m = 0$$

involves a division by zero.

5. Gamma Function as Infinite Product: Similarly, for the gamma function, the definition as an infinite product due to Euler is valid for all complex numbers $z$ except for non-positive integers

$$\Gamma(z) = \frac{1}{z}\prod_{n=1}^{\infty}\left[\frac{\left(1+\frac{1}{n}\right)^{z}}{\left(1+\frac{z}{n}\right)}\right]$$

6. Uniqueness of Euler's Infinite Product: By this construction, the gamma function is the unique function that simultaneously satisfies

$$\Gamma(1) = 1$$

$$\Gamma(z+1) = z\Gamma(z)$$

for all complex numbers $z$ except the non-positive integers, and

$$\underset{n \to \infty}{Limit} \frac{\Gamma(z+n)}{(n-1)!\, n^{z}} = 1$$

for all complex numbers $z$ (Davis (1959)).

## Weierstrass Definition

The definition for the gamma function due to Weierstrass is also valid for all complex numbers $z$ except the non-positive integers:

$$\Gamma(z) = \frac{e^{-\gamma z}}{z} \prod_{n=1}^{\infty} \left[ \frac{e^{-\frac{z}{n}}}{\left(1 + \frac{z}{n}\right)} \right]$$

where

$$\gamma \cong 0.577216$$

is the Euler-Mascheroni constant.

## In Terms of Generalized Laguerre Polynomials

1. Laguerre Polynomials – Incomplete Gamma Function: A parametrization of the incomplete gamma function in terms of the generalized Laguerre polynomials is

$$\Gamma(z, x) = x^{z} e^{-x} \sum_{n=0}^{\infty} \frac{\mathcal{L}_{n}^{z}(x)}{n + z}$$

which converges for

$$Re(z) > -1$$

and

$$x > 0$$

(National Institute of Standards and technology (2019)).

2. Laguerre Polynomials - Complete Gamma Function: A somewhat unusual parametrization of the gamma function in terms of the Laguerre polynomials is given by

$$\Gamma(z) = t^z \sum_{n=0}^{\infty} \frac{\mathcal{L}_n^z(t)}{n+z}$$

which converges for

$$Re(z) < \frac{1}{2}$$

## General Properties

1. Euler's Reflection and Duplication Formula: Other important functional equations for the gamma function are the Euler's reflection formula

$$\Gamma(z)\Gamma(1-z) = \frac{\pi}{\sin \pi z}$$

$$z \notin \mathbb{Z}$$

which implies

$$\Gamma(\epsilon - n) = (-1)^{n-1} \frac{\Gamma(-\epsilon)\Gamma(1+\epsilon)}{\Gamma(n+1-\epsilon)}$$

and the duplication formula

$$\Gamma(z)\Gamma\left(z+\frac{1}{2}\right) = 2^{1-2z}\sqrt{\pi}\Gamma(2z)$$

2. Duplication Formula and Multiplication Theorem: The duplication formula is a special case of the multiplication theorem (National Institute of Standards and technology (2019))

$$\prod_{k=0}^{m-1}\Gamma\left(z+\frac{k}{m}\right) = m^{\frac{1}{2}-mz}(2\pi)^{\frac{m-1}{2}}\Gamma(2z)$$

3. Complex Conjugate of the Gamma Function: A simple but useful property, which can be seen from the limit definition, is

$$\overline{\Gamma(z)} = \Gamma(\bar{z}) \Rightarrow \Gamma(z)\Gamma(\bar{z}) \in \mathbb{R}$$

4. Modulus of the Gamma Function: In particular, with

$$z = a + bi$$

this product is

$$|\Gamma(a+bi)|^2 = |\Gamma(a)|^2 \prod_{k=0}^{\infty} \frac{1}{1+\frac{b^2}{(a+k)^2}}$$

$$|\Gamma(bi)|^2 = \frac{\pi}{b\sinh(\pi b)}$$

$$\left|\Gamma\left(\frac{1}{2}+bi\right)\right|^2 = \frac{\pi}{\cosh(\pi b)}$$

5. Special Gamma Function Value $x = \frac{1}{2}$: Perhaps the best-known value of the gamma function at a non-integer is

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$$

which can be found by setting

$$z = \frac{1}{2}$$

in the reflection or the duplication formulas, or by using the relation to the beta function given below with

$$x = y = \frac{1}{2}$$

or by simply making the substitution

$$u = \sqrt{x}$$

in the integral definition of the gamma function.

6. Special Gamma Function Value $x = \frac{1}{2} \pm n$: In general, for non-negative integer values of $n$ one has

$$\Gamma\left(\frac{1}{2} + n\right) = \frac{(2n)!}{4^n n!}\sqrt{\pi} = \frac{(2n-1)!!}{2^n}\sqrt{\pi} = \binom{n - \frac{1}{2}}{n} n!\,\sqrt{\pi}$$

$$\Gamma\left(\frac{1}{2} - n\right) = \frac{(-4)^n n!}{(2n)!}\sqrt{\pi} = \frac{(-2)^n}{(2n-1)!!}\sqrt{\pi} = \frac{\sqrt{n\pi}}{n!}$$

where $n!!$ denotes the double factorial of $n$ and, when

$$n = 0$$

$$n!! = 1$$

7. Special Gamma Function Value Rational $x$: It might be tempting to generalize the result that

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$$

by looking for a formula for other individual values $\Gamma(r)$, where $r$ is rational.

8. Transcendental Nature of the Gamma Function: It has been proved that $\Gamma(n + r)$ is a transcendental number and algebraically independent of $\pi$ for any integer $n$ and each of the fractions

$$r = \frac{1}{6}, \frac{1}{4}, \frac{1}{3}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}$$

(Waldschmidt (2008)). In general, when computing the values of the gamma function, one must settle for numerical approximations.

9. Asymptotic Approximation for the Gamma Function: Another useful limit for the asymptotic approximation is

$$\underset{n \to \infty}{Limit} \frac{\Gamma(n + \alpha)}{\Gamma(n) n^{\alpha}} = 1$$

$$\alpha \in \mathbb{C}$$

10. Polygamma Functions: Derivatives of Gamma: The derivatives of the gamma function are described in terms of the polygamma function. For example,

$$\Gamma'(z) = \Gamma(z)\psi_0(z)$$

11. Gamma Derivative for Positive Integers: For a positive integer $m$ the derivatives of the gamma function can be calculated as follows:

$$\Gamma'(m+1) = m!\left(-\gamma + \sum_{k=1}^{m}\frac{1}{k}\right)$$

12. Gamma Function Derivatives when $Re(z) > 0$: When

$$Re(z) > 0$$

the n<sup>th</sup> derivative of the gamma function is

$$\frac{\partial^n}{\partial z^n}\Gamma(z) = \int\limits_0^\infty t^{z-1}e^{-t}(\ln t)^n dt$$

This can be derived by differentiating the integral form of gamma with respect to $z$,the technique of differentiation under the integral sign.

13. Polynomial Series for Gamma Functions: Using the identity

$$\left.\frac{\partial^n}{\partial z^n}\Gamma(z)\right|_{z=1} = (-1)^n n! \sum_{\pi\vdash n}\prod_{i=1}^{r}\frac{\zeta^*(a_i)}{k_i!\cdot a_i}$$

$$\zeta^*(z) \coloneqq \begin{cases}\zeta(z) & z \neq 1 \\ \gamma & z = 1\end{cases}$$

where $\zeta(z)$ is the Riemann zeta function with partitions

$$\pi = \left(\underbrace{a_1, \cdots, a_1}_{k_1}, \cdots, \underbrace{a_r, \cdots, a_r}_{k_1}\right)$$

one has in particular

$$\zeta(z) = \frac{1}{z} - \gamma + \frac{1}{2}\left(\gamma^2 + \frac{\pi^2}{6}\right)z - \frac{1}{6}\left[\gamma^3 + \frac{\pi^2\gamma}{2} + 2\zeta(3)\right]z^2 + \mathcal{O}(z^3)$$

## Inequalities

1. Characterizing Strictly Logarithmically Convex Functionality: When restricted to positive real numbers, the gamma function is a strictly logarithmically convex function. This property may be stated in any of the following three equivalent ways.
2. Characterization via Exponentially Convex Inequality: For any two positive real numbers $x_1$ and $x_2$, and for any

$$t \in [0, 1]$$

$$\Gamma(tx_1 + [1 - t]x_2) \leq \Gamma^t(x_1)\Gamma^{1-t}(x_2)$$

Moreover, the inequality is strict for

$$t \in (0, 1]$$

3. Characterization via Spaced Point Pair: For any two positive numbers $x$ and $y$ with

$$y > x$$

$$\left[\frac{\Gamma(y)}{\Gamma(x)}\right]^{\frac{1}{y-x}} > e^{\frac{\Gamma'(x)}{\Gamma(x)}}$$

4. Characterization via First/Second Derivatives: For any positive real number $x$,

$$\Gamma''(x)\Gamma(x) > \Gamma'(x)$$

5. Strict Positivity of the First Derivative: The last of the statements is, essentially by definition, the same as the statement that $\psi_1(x)$ where $\psi_1$ is the polygamma function of order 1. To prove the logarithmic convexity of the gamma function, it therefore suffices to observe that $\psi_1$ has a series representation which, fort positive real $x$, consists of only positive terms.
6. Convexity Extension to Multi-point Interpolant: Logarithmic convexity and Jensen's inequality together imply, for any positive real numbers $x_1, \cdots, x_n$ and $a_1, \cdots, a_n$

$$\Gamma\left(\frac{a_1x_1 + \cdots + a_nx_n}{a_1 + \cdots + a_n}\right) \leq [\Gamma(x_1) \cdots \Gamma(x_n)]^{\frac{1}{a_1 + \cdots + a_n}}$$

7. Bounds on Gamma - Gautschi Inequality: There are also bounds on the ratios of gamma functions. The best-known is the Gautschi's inequality, which says that for any positive real number $x$ and

$$s \in (0,1)$$

$$x^{1-s} < \frac{\Gamma(x+1)}{\Gamma(x+s)} < (x+1)^{1-s}$$

## Stirling's Formula

1. Asymptotic Growth of the Gamma Function: The behavior of $\Gamma(z)$ for an increasing positive variable is simple; it grows quickly, faster than an exponential function.
2. Asymptotic Approximation using Stirling's Formula: Asymptotically, as

$$z \to \infty$$

the magnitude of the gamma function is given by the formula

$$\Gamma(z+1) \sim \sqrt{2\pi e}\left(\frac{z}{e}\right)^{z}$$

where the symbol ~ means that the ratio of the two sides converges to 1 (Davis (1959)) or asymptotically converges.

## Residues

1. Analytic Continuity into Negative Planes: The behavior of non-positive $z$ is more intricate. Euler's integral does not converge for

$$z \leq 0$$

but the function it defines in the positive convex half-plane has a unique analytic continuation to the negative half-plane.
2. Repeated Application of the Recurrence: One way to find that analytic continuation is to use Euler's integral for positive arguments and extend the domain to negative numbers by repeated application of the recurrence formula (Davis (1959))

$$\Gamma(z) = \frac{\Gamma(z+n+1)}{z(z+1)\cdots(z+n)}$$

choosing $n$ such that $z + n$ is positive.

3. Meromorphic in the Negative Half Plane: The product in the denominator is zero if $z$ equals any of the integers $0, -1, \cdots$ Thus, the gamma function must be undefined at these points to avoid division by zero; it is a meromorphic function with simple poles at the non-positive integers (Davis (1959)).
4. Residue Conducive Expansion for Gamma: The definition can be re-written as

$$(z + n)\Gamma(z) = \frac{\Gamma(z + n + 1)}{z(z + 1) \cdots (z + n - 1)}$$

5. Formal Definition of the Residue: For a function $f$ of a complex variable $z$, at a simple pole $c$, the residue of $f$ is given by:

$$Residue(f, c) = \frac{Limit}{z \to c} [(z - c)f(z)]$$

6. Residue Numerator and Denominator Values: When

$$z = -n$$

$$\Gamma(z + n + 1) = \Gamma(1) = 1$$

and

$$z(z + 1) \cdots (z + n - 1) = (-1)^n n!$$

7. Gamma Function Residue at Poles: So, the residues of the gamma function at those points are:

$$Residue(\Gamma, -n) = \frac{(-1)^n}{n}$$

8. Holomorphic Nature of Reciprocal Gamma: The gamma function is non-zero everywhere along the real line, although it comes arbitrarily close to zero as

$$z \to -\infty$$

There is in fact, no complex number $z$ for which

$$\Gamma(z) = 0$$

and hence the reciprocal gamma function $\frac{1}{\Gamma(z)}$ is an entire function, with zeros at

$$z = 0, -1, \cdots$$

(Davis (1959)).

## Minima

1. Gamma Function Minimum Inside $[0, 1]$: The gamma function has a local minimum at

$$z_{MIN} \approx 1.46163$$

where it attains the value

$$\Gamma(z_{MIN}) \approx 0.885603$$

2. Gamma Function Minimum between Poles: The gamma function must alternate the signs between the poles because the product in the forward recurrence contains an odd number of negative factors if the number of poles between $z$ and $z + n$ is odd, and even number of poles if their number is even.

## Integral Representations

1. Alternate Integral Representations of Gamma: There are many formulations, besides the Euler's Integrals of the second kind, that express gamma function as an integral.
2. Log Reciprocal Representation in $[0, 1]$: For instance, when the real part of $z$ is positive (Whittaker and Watson (1996)):

$$\Gamma(z) = \int_{0}^{1} \left(\log \frac{1}{t}\right)^{z-1} dt$$

3. Binet's First Integral Representation: Binet's first integral formula for the gamma function states that, when the real part of $z$ is positive, then (Whittaker and Watson (1996)):

$$\log \Gamma(z) = \left(z - \frac{1}{2}\right) \log z - z + \frac{1}{2} \log 2\pi + \int_{0}^{\infty} \left(\frac{1}{2} - \frac{1}{t} + \frac{1}{e^t - 1}\right) \frac{e^{-tz}}{t} dt$$

4. Laplacian of the Error Term: The integral on the right-hand side may be interpreted as a Laplace transform. That is,

$$\log \left[\Gamma(z) \left(\frac{e}{z}\right)^z \sqrt{2\pi e}\right] = \mathfrak{L}\left(\frac{1}{2t} - \frac{1}{t^2} + \frac{1}{t[e^t - 1]}\right)(z)$$

5. Binet's Second Integral Representation Form: Binet's second integral formula states that, again when the real part of $z$ is positive,

$$\log \Gamma(z) = \left(z - \frac{1}{2}\right)\log z - z + \frac{1}{2}\log 2\pi + 2\int_{0}^{\infty} \frac{\arctan\left(\frac{t}{z}\right)}{e^{2\pi t} - 1} dt$$

6. Reimann Sphere Hankel Contour Transform: Let $C$ be a Hankel contour, meaning a path the begins and ends at the point $\infty$ in the Riemann sphere, whose unit tangent vector converges to $-1$ at the start of the path and at $+1$ at the end, which has a winding number of 1 around 0, and which does not cross $[0, \infty)$.
7. Specification of the Contour Branch: Fix a branch of $\log(-t)$ to be real when $-t$ is on the negative real axis. Assume $z$ is not an integer.
8. Hankel's Formula for Gamma Function: Then the Hankel's formula for the gamma function is (Whittaker and Watson (1996)):

$$\Gamma(z) = -\frac{1}{2i\sin(\pi z)}\oint_{C}^{C}(-t)^{z-1}e^{-t}dt$$

where $(-t)^{z-1}$ is interpreted as $e^{(z-1)\log(-t)}$
9. Application of the Reflection Formula: The reflection formula leads to the closely related expression

$$\Gamma(z) = \frac{i}{2\pi}\oint_{C}^{C}(-t)^{-z}e^{-t}dt$$

again, valid wherever $z$ is not an integer.

## Fourier Series Expansion

The logarithm of the gamma function has the following Fourier series expansion for

$$0 < z < 1$$

$$\log\Gamma(z) = \left(z - \frac{1}{2}\right)(\gamma + \log z) + (1 - z)\log\pi - \frac{1}{2}\log(\sin(\pi z)) + \frac{1}{\pi}\sum_{n=1}^{\infty}\frac{\log n}{n}\sin(2\pi n z)$$

which was, for a long time, attributed to Ernst Kummer, who derived it in 1847 (Bateman and Erdelyi (1955), Shrivastava and Choi (2001)). However, Blagouchine (2014) discovered that Carl Johann Malmsten first derived this series in 1842.

## Raabe's Formula

In 1840, Joseph Ludwig Raabe proved that

$$\int_{a}^{a+1} \ln\Gamma(z)\,dz = \frac{1}{2}\log(2\pi) + a\log a - a$$

$$a > 0$$

In particular, if

$$a = 0$$

then

$$\int_0^1 \ln\Gamma(z)\,dz = \frac{1}{2}\log(2\pi)$$

## Pi Function

1. Definition of the Pi Function: A alternative notation, which was originally introduced by Gauss and which was sometimes used is the *Pi function*, which in terms of the gamma function is

$$\Pi(z) = \Gamma(z+1) = z\Gamma(z) = \int_0^\infty e^{-t}t^z dz$$

so that

$$\Pi(n) = n!$$

For every non-negative integer $n$.

2. Applying Reflection Formula/Multiplication Theorem: Using the Pi function, the reflection formula takes on the form

$$\Pi(z)\Pi(-z) = \frac{\pi z}{\sin(\pi z)} = \frac{1}{sinc\ z}$$

where $sinc\ z$ is the normalized sine function, while the multiplication theorem takes on the form

$$\Pi\left(\frac{z}{m}\right)\Pi\left(\frac{z-1}{m}\right)\cdots\Pi\left(\frac{z-m+1}{m}\right) = (2\pi)^{\frac{m-1}{2}}m^{-z-\frac{1}{2}}\Pi(z)$$

3. Reciprocal of the Pi Function: One also sometimes finds

$$\pi(z) = \frac{1}{\Pi(z)}$$

   which is entire function, defined for every complex number, just like the reciprocal gamma function. That $\pi(z)$ is entire entails that it has no poles, so $\Pi(z)$, like $\Gamma(z)$, has no zeros.

4. Volume of the $n$- Ellipsoid: The volume of the $n$-ellipsoid with radii $r_1, \cdots, r_n$ can be expressed as

$$V(r_1, \cdots, r_n) = \frac{\pi^{\frac{n}{2}}}{\Pi\left(\frac{n}{2}\right)} \prod_{k=1}^{n} r_k$$

## Relation to Other Functions

1. Upper/Lower Incomplete Gamma Functions: In the first integral above, which defines the gamma function, the limits of integration are fixed. The upper and the lower incomplete gamma functions are the functions obtained by allowing the lower or upper (respectively) limits of integration to vary.
2. Relation of Beta to Gamma: The gamma function is related to the Beta function by

$$\mathcal{B}(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

3. Gamma Function Derivatives - Digamma Polygamma: The logarithmic derivative of the gamma function is called the digamma function; higher derivatives are the polygamma functions.

4. <u>Gamma Function vs. Exponential Sum</u>: The analog of the gamma function over a finite field or a finite ring is the Gaussian sum, a type of exponential sum.
5. <u>Reciprocal of the Gamma Function</u>: The reciprocal gamma function is an entire function and has been studied as a specific, separate topic.
6. <u>Gamma vs. Reimann Zeta Function</u>: The gamma function also shows up in an important relation with the Reimann zeta function $\zeta(z)$.

$$\pi^{-\frac{z}{2}}\Gamma\left(\frac{z}{2}\right)\zeta(z) = \pi^{-\frac{1-z}{2}}\Gamma\left(\frac{1-z}{2}\right)\zeta(1-z)$$

7. <u>Product of Gamma and Reimann Zeta</u>: It also appears in the following formula:

$$\Gamma(z)\zeta(z) = \int_0^\infty \frac{u^z}{e^u - 1}\frac{du}{u}$$

which is only valid for

$$Re(z) > 1$$

8. <u>Log of the Gamma Function</u>: The logarithm of the gamma function satisfies the following relation:

$$\log\Gamma(z) = {\zeta_H}'(0, z) - \zeta'(0)$$

where $\zeta_H$ is the Hurwitz zeta function, $\zeta$ is the Reimann zeta function, and the prime denotes differentiation in the first variable.
9. <u>Moments of Stretched Exponential Function</u>: The gamma function is related to the stretched exponential function. For instance, moments of that function are

$$\langle \tau^n \rangle = \int_0^\infty dt\ t^{n-1} e^{-\left(\frac{t}{\tau}\right)^\beta} = \frac{\tau^n}{\beta} \Gamma\left(\frac{n}{\beta}\right)$$

## Particular Values

1. Some Specific Gamma Function Values: Some particular values of the gamma function are:

$$\Gamma\left(-\frac{3}{2}\right) = \frac{4}{3}\sqrt{\pi} \approx 2.633\ 271\ 801\ 207$$

$$\Gamma\left(-\frac{1}{2}\right) = -2\sqrt{\pi} \approx -3.544\ 907\ 701\ 811$$

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi} \approx 1.772\ 453\ 850\ 906$$

$$\Gamma(1) = 0! \approx 1$$

$$\Gamma\left(\frac{3}{2}\right) = \frac{1}{2}\sqrt{\pi} \approx 0.886\ 226\ 925\ 453$$

$$\Gamma(2) = 1! \approx 1$$

$$\Gamma\left(\frac{5}{2}\right) = \frac{3}{4}\sqrt{\pi} \approx 1.329\ 340\ 388\ 179$$

$$\Gamma(3) = 2! \approx 2$$

$$\Gamma\left(\frac{7}{2}\right) = \frac{15}{8}\sqrt{\pi} \approx 3.323\ 350\ 970\ 448$$

$$\Gamma(4) = 3! \approx 6$$

2. Undefined for Non-Positive Numbers: The complex gamma function is undefined for non-positive integers, but in these cases, the values can be defined in the Reimann sphere as ∞.
3. Holomorphic Nature of Reciprocal Gamma: The reciprocal gamma is well-defined, and analytic at these values – and in the entire complex plane:

$$\frac{1}{\Gamma(-3)} = \frac{1}{\Gamma(-2)} = \frac{1}{\Gamma(-2)} = \frac{1}{\Gamma(0)} = 0$$

## The Log-Gamma Function

1. Rationale behind the Log-Gamma Function: Because the gamma and the factorial functions grow so rapidly for moderately large arguments, many computing environments include a function that return the natural logarithm of the gamma function – often given as *lgamma* or *lngamma* in programming environments, or *gammaln* in spreadsheets – this grows much more slowly, and for combinatorial calculations allows adding and subtracting logs instead of multiplying and dividing by very large values.
2. Definition of Log-Gamma Function: It is often defined as

$$\log\Gamma(z) = -\gamma z - \log z + \sum_{k=1}^{\infty}\left[\frac{z}{k} - \log\left(1 + \frac{z}{k}\right)\right]$$

3. Digamma Function Derivative of Log Gamma: The digamma function, which is a derivative of this function, is also commonly seen.
4. Single Strip Function Value: In the context of technical and physical applications, e.g., wave propagation, the functional equation

$$\log\Gamma(z) = \log\Gamma(z+1) - \log z$$

   is often used since it allows one to determine the functional values in one strip of width 1 in $z$ from the neighboring strip.
5. Large $z$ as Starting Point: In particular, starting with a good approximation for a $z$ with large real part, one may go step-by-step down to the derived $z$.
6. Rocktaeschel Proposal for $\log\Gamma(z)$ Approximation: Following an indication of Carl Friedrich Gauss, Rocktaeschel (1922) proposed for $\log\Gamma(z)$ an approximation for large $Re(z)$:

$$\log\Gamma(z) \approx \left(z - \frac{1}{2}\right)\log z - z + \frac{1}{2}\log(2\pi)$$

7. Approximation for Smaller $z$: This can be used to accurately approximate $\log\Gamma(z)$ for $z$ with smaller $Re(z)$ via Bohmer (1939):

$$\log\Gamma(z-m) = \log\Gamma(z) - \sum_{k=1}^{m}\log(z-k)$$

8. Approximation using Higher Order Terms: A more accurate approximation can be obtained from the asymptotic expansions of $\log\Gamma(z)$ and $\Gamma(z)$, which are based on Stirling's approximation:

$$\Gamma(z) \sim z^{z-\frac{1}{2}}e^{-z}2\pi\left(1 + \frac{1}{12z} + \frac{1}{288z^2} - \frac{139}{51840z^3} - \frac{571}{2488320z^4}\right)$$

as

$$|z| \to \infty$$

at constant

$$|\arg z| < \pi$$

9. Approximation using Log Gamma Series: In a more *natural* presentation:

$$\log \Gamma(z) \approx z \log z - z + \frac{1}{12z} - \frac{1}{360z^3} + \frac{1}{1260z^5}$$

as

$$|z| \to \infty$$

at constant

$$|\arg z| < \pi$$

10. Coefficients in the Polynomial Expansion: The coefficients of the terms with

$$k > 1$$

of $z^{1-k}$ in the last expansion are simply $\frac{B_k}{k(k-1)}$ where $B_k$ are the Bernoulli numbers.

## The Log-Gamma Function Properties

1. Log-Gamma Bohr-Mollerup Theorem: The Bohr-Mollerup theorem states that, among all functions extending factorial functions to positive real numbers, only the gamma function is log-convex, that is, its natural logarithm is convex on the positive real axis.
2. Log-Gamma using Riemann Zeta: In a certain sense, the log gamma function is the more natural form; it makes some intrinsic attributes of the function clearer. A striking example of the Taylor series of $\log \Gamma$ around 1:

$$\log \Gamma(z) = -\gamma z + \sum_{k=2}^{\infty} \frac{\zeta(k)}{k} (-z)^k \ \ \forall \, |z| < 1$$

   with $\zeta(k)$ denoting the Riemann zeta function at $k$.
3. Integral Representation for Log-Gamma: So, using the following property

$$\zeta(z)\Gamma(z) = \int_0^{\infty} \frac{u^z}{e^u - 1} \frac{du}{u}$$

   one can find the integral representation for the $\log \Gamma(z)$ function:

$$\log \Gamma(1 + z) = -\gamma z + \int_0^{\infty} \frac{e^{-zt} - 1 + zt}{t(e^t - 1)} dt$$

   or, setting

$$z = 1$$

   and calculating $\gamma$,

$$\log\Gamma(1+z)=\int\limits_0^\infty \frac{e^{-zt}-ze^{-t}-1+z}{t(e^t-1)}dt$$

4. <u>Log-Gamma for Rational $z$</u>: There also exist special formulas for the logarithm of gamma function for rational $z$. For instance, if $k$ and $n$ are integers with

$$k<n$$

and

$$k\neq\frac{n}{2}$$

then

$$\log\Gamma\left(\frac{k}{n}\right)=\frac{(n-2k)\log(2\pi)}{2n}+\frac{1}{2}\left\{\log\pi-\log\left(\sin\frac{\pi k}{n}\right)\right\} \\ +\frac{1}{\pi}\sum_{r=1}^{n-1}\frac{\gamma+\log r}{r}\cdot\sin\frac{2\pi kr}{n}-\frac{1}{2\pi}\cdot\sin\frac{2\pi kr}{n}\int\limits_0^\infty\frac{e^{-nx}\log x}{\cosh x-\cdot\cos\frac{2\pi k}{n}}dx$$

(Blagouchine (2015)). This formula is sometimes used for numerical computation, since the integrand decreases very quickly.

## Integration Over Log-Gamma

1. <u>Log-Gamma Integral from Barnes G</u>: The integral $\int_0^z\Gamma(x)dx$ can be expressed in terms of the Barnes G-function (Alexejweski (1894), Barnes (1899))

$$\int_0^z \Gamma(x)dx = \frac{z}{2}\log 2\pi + \frac{z(1-z)}{2} + z\Gamma(z) - \ln G(z+1)$$

where

$$Re(z) > -1$$

2. Log-Gamma Integral from Hurwitz Zeta: It can also be written in terms of the Hurwitz zeta function (Gosper (1997), Adamchik (1998)):

$$\int_0^z \Gamma(x)dx = \frac{z}{2}\log 2\pi + \frac{z(1-z)}{2} - \zeta'(-1) + \zeta'(-1, z)$$

## Approximations

1. Arbitrary Precision using Stirling/Lanczos: Complex values of the gamma function can be computed numerically to arbitrary precision using Stirling's approximation or Lanczos approximation.
2. Fixed Precision Estimate for $Re(z) \in [1, 2]$: The gamma function can be computed to fixed precision for

$$Re(z) \in [1, 2]$$

by applying integration by parts to the Euler's integral.

3. Breaking Down the Integral Form: For any positive number $x$ the gamma function can be written as

$$\Gamma(z) = \int_{0}^{x} e^{-t} t^{z} \frac{dt}{t} + \int_{x}^{\infty} e^{-t} t^{z} \frac{dt}{t} = e^{-x} x^{z} \sum_{n=0}^{\infty} \frac{x^{n}}{z(z+1)\cdots(z+n)} + \int_{x}^{\infty} e^{-t} t^{z} \frac{dt}{t}$$

4. Custom N-bit Precision Estimate: When

$$Re(z) \in [1, 2]$$

and

$$x \geq 1$$

the absolute value of the last integral is smaller than $(x+1)e^{-x}$. By choosing a large enough $x$, the last expression can be smaller than $2^{-N}$ for any desired value $N$. Thus, the gamma function can be evaluated to $N$ bits of precision with the above series.

5. Karatsuba Algorithm for Euler Gamma: A fast algorithm for the calculation of the Euler gamma function for any algebraic argument – including rational – was constructed by Karatsuba (1991a, 1991b).
6. Estimates using Arithmetic-Geometric Iterations: For arguments that are integer multiples of $\frac{1}{24}$, the gamma function can also be evaluated quickly using arithmetic-geometric mean iterations – see the Section on Particular Values of the Gamma Function, as well as Borwein and Zucker (1992).

## Applications – Integration Problems

1. Special Nature of the Gamma Function: The gamma functions have been described as arguably the most common special function, or at least the *least* special of them. The other transcendental functions are special because one could conceivably avoid them

by staying away from many specialized mathematical topics. On the other hand, the gamma function is the most difficult to avoid.

2. Range of Gamma Function Application: The gamma function finds applications in such diverse areas such as quantum physics, astrophysics, and fluid dynamics (Chaudhry and Zubair (2001)).
3. Gamma Distribution Usage in Statistics: The gamma distribution, which is formulated in terms of the gamma function, is used in statistics to model a wide range of processes; for example, the time between the occurrences of the earthquakes (Rice (2010)).
4. Rationale Behind Gama Functions' Usefulness: The primary reason for the Gamma function's usefulness in such contexts is the prevalence of expressions of the type $f(t)e^{-g(t)}$ which describe processes that decay exponentially in space or time.
5. Reduction into Gamma-Type Integrals: Integrals of such expressions can be occasionally solved in terms of gamma function when no elementary solution exists.
6. Example - Power Source Function Decay: For example, if $f$ is a power function and $g$ is a linear function, a simple change of variables gives the evaluation

$$\int_{0}^{\infty} t^{b} e^{-at} dt = \frac{\Gamma(b+1)}{a^{b+1}}$$

7. Significance of Real Positive Integration: The fact that the integration is performed along the entire real positive line might signify that the gamma function represents the combination of a time-dependent process that continuous indefinitely, or the value might be the total of a distribution in an infinite space.
8. Complete vs. Incomplete Gamma Function: It is of course frequently useful to take limits of integration other $0$ ad $\infty$ to describe the cumulation of a finite process, in which case the ordinary gamma function is no longer a solution; the solution is then called an incomplete gamma function.

9. Gaussian Category of Exponentially Decaying Functions: An important category of exponentially decaying functions is that of the Gaussian functions $ae^{-\frac{(x-b)^2}{2}}$ and integrals thereof, such as the error function. There are many inter-relations between these functions and the gamma function, notably, $\sqrt{\pi}$ obtained by evaluating $\Gamma\left(\frac{1}{2}\right)$ is the *same* as that found in the normalizing factor of the error distribution and the normal distribution.
10. Gamma Function from Algebraic Integrals: The integrals discuss so far involve transcendental functions, but the gamma function also arises from integrals of purely algebraic functions.
11. Arc Lengths, Surfaces, and Volumes: In particular, the arc lengths of the ellipses and the lemniscates, which are curves defined by algebraic equations, are given by elliptic integrals that in special cases can be evaluated in terms of the gamma functions. The gamma function also be used to calculate *volume* and *area* of *n*-dimensional hyper-spheres.
12. Definition of the Beta Function: Another important special case is that of the beta function:

$$\mathcal{B}(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

## Calculating Products

1. Gamma Function as Generalized Factorial: The gamma function’s ability to generalize factorial products immediately leads to applications in many areas of mathematics, in combinatorics, and by extension in areas such as probability theory and the calculation of power series.

2. Gamma Function for Binomial Coefficient: Many expressions involving products of successive integers can be written as some combination of factorials, the most important example being that of the binomial coefficient

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

3. Interpretation Suitability for Binomial Coefficient: The example of binomial coefficient motivates why the properties of gamma functions are natural when extended to negative numbers. A binomial coefficient gives the number of ways to choose $k$ elements from $n$ elements; if

$$k > n$$

there are of course no ways. If

$$k > n$$

$(n-k)!$ is the factorial of a negative integer, and hence infinite if the gamma function's definition of the integral is used – dividing by infinity gives the expected value of 0.

4. Extension to Rational Function Products: One can replace the factorial by a gamma function to extend any such formula to complex numbers. Generally, this works for any product where each factor is a rational function of the index variable, by factoring the rational functions into linear expressions.

5. Polynomial Quotient Using Gamma Functions: If $P$ and $Q$ are monic polynomials with degree $m$ and $n$ and respective roots $p_1, \cdots, p_m$ and $q_1, \cdots, q_n$, one has

$$\prod_{i=a}^{b} \frac{P(i)}{Q(i)} = \left[\prod_{j=1}^{m} \frac{\Gamma\left(b-p_j+1\right)}{\Gamma\left(a-p_j\right)}\right]\left[\prod_{k=1}^{n} \frac{\Gamma(a-q_k)}{\Gamma(b-q_k+1)}\right]$$

6. Advantage of the Gamma Function Representation: If there is a way to calculate the gamma function numerically, it is a breeze to calculate numerical values of such products. The number of gamma functions on the right depend upon only the degree of polynomials, so it does not matter whether $b - a$ equals 5 or $10^5$
7. Handling Product Poles and Zeros: By taking appropriate limits, the equation can also be made to hold even when the left-hand product contains zeros and poles.
8. Extension to Infinite Rational Products: By taking limits, certain rational products of infinitely many factors can be evaluated in terms of gamma function as well. Due to the Weierstrass factorization theorem, analytic functions can be written as infinite products, and these can sometimes be represented as finite products or quotients of the gamma function.
9. Example - Sine Function Using Gamma: Already one striking example has been seen; the reflection formula essentially represents the sine function as a product of two gamma functions.
10. Exponential, Trigonometric, and Hyperbolic Functions: Starting from this expression, the exponential function as well as all the trigonometric and the hyperbolic functions can be expressed using gamma functions.
11. Mellin-Barnes Complex Contour Integrals: More functions yet, including the hypergeometric functions and special cases thereof, can be represented by means of complex contour integrals of the products and the quotients of the gamma functions, called the Mellin-Barnes integrals.

## Analytic Number Theory

1. Study of Riemann Zeta Function: An elegant and deep application of the gamma function is in the study of the Riemann zeta function.
2. Reimann Zeta Function Functional Equation: A fundamental property of the Riemann zeta function is its functional form:

$$\Gamma\left(\frac{s}{2}\right)\zeta(s)\pi^{-\frac{s}{2}} = \Gamma\left(\frac{1-s}{2}\right)\zeta(1-s)\pi^{-\frac{1-s}{2}}$$

3. Reimann Zeta Function Analytic Continuation: Among other things, this provides an explicit form for analytic continuation of the zeta function to a meromorphic function in the complex plane and leads to an immediate proof that the zeta function has infinitely many so-called *trivial* zeros on the real line.
4. Reimann Zeta Function - Property #2: Another property of the Riemann zeta function is

$$\Gamma(s)\zeta(s) = \int_{0}^{\infty} \frac{t^s}{e^t - 1}\frac{dt}{t}$$

5. Analytic Number Theory - Developmental Milestone: Both formulas were derived by Bernhard Riemann in his seminal 1859 paper (Riemann (1859)), one of the milestones in the development of analytic number theory – the branch of mathematics that studies prime numbers using the tools of mathematical analysis.
6. Riemann Extension to Factorial Numbers: Factorial numbers, considered as discrete objects, are an important concept in classical number theory, because they contain many prime factors, but Riemann found a use for their continuous extension that arguably turned out to be even more important.

## References

- Adamchik, V. S. (1998): Polygamma Functions of Negative Order *Journal of Computational and Applied Mathematics* **100 (2)** 191-199

- Alexejewski, W. P. (1894): Uber eine Klasse von Funktionen, die der Gammafunktion Analog Sid *Leipzig Weidmannsche Buchhandluns* **46** 268-275
- Barnes, E. W. (1899): The Theory of the G-Function *Quarterly Journal of Pure and Applied Mathematics* **31** 264-314
- Bateman, H., A. Erdelyi (1955): *Higher Transcendental Functions* **McGraw Hill**
- Blagouchine, I. V. (2014): Re-discovery of Malmsten's Integrals, their Evaluation by Contour Integration Methods, and some Related Results *Ramanujan Journal* **35 (1)** 21-110
- Blagouchine, I. V. (2015): A Theorem for the Closed-form Evaluation of the First Generalized Stieljes Constant at Rational Arguments and Related Summations **arXiv**
- Bohmer, P. E. (1939): *Differenzengleichungen und Bestimmte Integrale* **Kohler-Verlag** Leipzig
- Borwein, J. M., And I. J. Zucker (1992): Fast Evaluation of the Gamma Function for Small Rational Fractions using Complete Elliptical Integrals of the First Kind *IMA Journal of Mathematical Analysis* **12** 519-526
- Chaudhry, S. M., and M. A. Zubair (2003): *On a Class of Incomplete Gamma Functions with Applications* **Chapman & Hall/CRC**
- Davis, P. J. (1959): Leonhard Euler's Integral – A Historical Profile of the Gamma Function *American Mathematical Monthly* **66 (10)** 849-869
- Gosper, R. W. (1997): $\int_{\frac{m}{4}}^{\frac{n}{6}} \log \Gamma(z)\, dz$ in: *Special Functions, Q-Series, and Related Topics, Ismail, M., D. Masson, and M. Rahman, editors* **Fields Institute of Communication, American Mathematical Society**
- Karatsuba, E. A. (1991a): Fast Evaluation of Transcendental Functions *Problems of Information Transmission* **27 (4)** 339-360
- Karatsuba, E. A. (1991b): On a New Method for Fast Evaluation of Transcendental Functions *Russian Mathematical Surveys* **46 (2)** 246-247
- Kingman, J. F. C. (1961): A Convexity Property of Positive Matrices *Quarterly Journal of Mathematics* **12 (1)** 283-284

- National Institute for Standards and Technology (2019): Incomplete Gamma and Related Functions
- Rice, J. A. (2010): *Mathematical Statistics and Data Analysis 3rd Edition* **Duxbury Advanced Series**
- Riemann, B. (1859)): Ueber die Anzahl der Primzahlen Unter Einer Gegebenen Grosse
- Rocktaeschel, O. R. (1922): *Methoden zur Berechnung der Gammafunktion fur Komplexes Argument* **Technical University** Dresden
- Shrivastava, H. M., and J. Choi (2001): *Series Associated with Zeta and Related Functions* **Kluwer Academic** Netherlands
- Waldschmidt, M. (2006): Transcendence of Periods: The State of the Art *Pure and Applied Mathematics Quarterly* **2 (2)** 435-463
- Whittaker, E. T., and G. N. Watson (1996): *A Course on Modern Analysis 4th Edition* **Cambridge University Press**
- Wikipedia (2019): Gamma Function

# Stirling's Approximation

## Introduction and Overview

1. Approximation for Factorials and Gamma: *Stirling's approximation* (or *Stirling's formula*) is an approximation for factorials.
2. Accuracy for Small $n$ Values: It is a good approximation, leading to accurate results even for small values of $n$. It is named after James Stirling, although it was first stated by Abraham de Moivre (Pearson (1924), Le Cam (1986), Dutka (1991), Wikipedia (2019)).
3. Typical Version of Stirling's Approximation: The version of the formula typically used in applications is

$$\ln n! = n \ln n - n + \mathcal{O}(\ln n)$$

in the big-$\mathcal{O}$ notation, as

$$n \rightarrow \infty$$

By changing the base of the logarithm, for instance, in the worst case lower bound for comparison sorting, it becomes

$$log_2 n! = n\ log_2 n - (log_2 e)\ n + \mathcal{O}(log_2 n)$$

4. <u>Constant Term in the Approximation</u>: Specifying the constant in the $\mathcal{O}(\ln n)$ error term gives $\frac{1}{2}\ln(2\pi n)$ yielding the more precise formula

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

where the sign ~ means that the two quantities are asymptotic, i.e., their ratio tends to $1$ as $n$ tends to infinity.

5. <u>Upper/Lower Bounds on the Approximation</u>: One may also give simple bounds valid for all positive integers $n$, rather than for only large $n$:

$$\sqrt{2\pi}\, n^{n+\frac{1}{2}}\, e^{-n} \le n! \le e\, n^{n+\frac{1}{2}}\, e^{-n}$$

for

$$n = 1, 2, 3, \cdots$$

These follow from the more precise error bounds discussed below in this chapter.

## Derivation

1. <u>Summation of the Logs as an Integral</u>: Roughly speaking, the simplest version of Stirling's formula can be obtained by approximating the sum

$$\ln n! = \sum_{j=1}^{n} \ln j$$

by an integral

$$\sum_{j=1}^{n} \ln j \approx \int_{0}^{1} \ln x \, dx = n \ln n - n + 1$$

2. Sum of the Log Terms: The full formula, together with an estimate of its error, can be derived as follows. Instead of approximating $n!$, one considers its natural log as this slowly varying function:

$$\ln n! = \ln 1 + \cdots + \ln n$$

3. Trapezoidal Approximation of the Integral: The right-hand side of the equation minus

$$\frac{1}{2}(\ln 1 + \ln n) = \frac{1}{2} \ln n$$

is the approximation by the trapezoidal rule of the integral

$$\ln n! - \frac{1}{2} \ln n \approx \int_{0}^{1} \ln x \, dx = n \ln n - n + 1$$

and the error in this approximation is given by the Euler-MacLaurin formula

$$\begin{aligned} \ln n! - \frac{1}{2} \ln n &= \frac{1}{2} \ln 1 + \ln 2 + \cdots + \ln(n-1) + \frac{1}{2} \ln n \\ &= n \ln n - n + 1 + \sum_{k=2}^{m} \frac{(-1)^k B_k}{k(k-1)} + R_{m,n} \end{aligned}$$

where $B_k$ is the Bernoulli number and $R_{m,n}$ is the remainder term in the Euler-MacLaurin formula.

4. <u>Integration Error Terms as $n \to \infty$</u>: Take limits to find that

$$\lim_{n\to\infty}\left(\ln n! - n\ln n + n - \frac{1}{2}\ln n\right) = 1 - \sum_{k=2}^{m}\frac{(-1)^k B_k}{k(k-1)} + \lim_{n\to\infty} R_{m,n}$$

5. <u>Euler-MacLaurin Error Term Asymptote</u>: This limit is denoted as $y$. Because the remainder $R_{m,n}$ in the Euler-MacLaurin formula satisfies

$$R_{m,n} = \lim_{n\to\infty} R_{m,n} + \mathcal{O}\left(\frac{1}{n^m}\right)$$

where the big $\mathcal{O}$ notation is used again, combining the equations above yields the approximation formula in its logarithmic form:

$$\ln n! = n\ln\left(\frac{n}{e}\right) + \frac{1}{2}\ln n + y + \sum_{k=2}^{m}\frac{(-1)^k B_k}{k(k-1)} + \mathcal{O}\left(\frac{1}{n^m}\right)$$

6. <u>Exponentiation for the $n!$ Expression</u>: Taking exponential of both sides, and choosing a positive integer $m$, one gets an expression involving the unknown quantity $e^y$. For

$$m = 1$$

the expression becomes

$$n! = e^y\sqrt{n}\left(\frac{n}{e}\right)^n\left[1 + \mathcal{O}\left(\frac{1}{n}\right)\right]$$

7. Explicit Value of the Limit: The quantity $e^y$ can be found by taking the limit on both sides as $n$ tends to infinity and using Wallis' product, which shows that

$$e^y = \sqrt{2\pi}$$

Therefore, the Stirling's formula is obtained as

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left[1 + \mathcal{O}\left(\frac{1}{n}\right)\right]$$

## An Alternative Definition

1. Starting Point – The Gamma Function: An alternative formula for $n!$ using the gamma function is

$$n! = \int_0^{\infty} x^n e^{-x} dx$$

This can be verified by repeated integration by parts.

2. Change of $x$ to $ny$: Rewriting and arranging the variables as

$$x = ny$$

one gets

$$n! = \int_0^\infty e^{n\ln x - x}dx = ne^{n\ln n}\int_0^\infty e^{n(\ln y - y)}dy$$

3. Laplace's Method Recovers Stirling's Formula: Applying Laplace's method, one has

$$\int_0^\infty e^{n(\ln y - y)}dy \sim \sqrt{\frac{2\pi}{n}}e^{-n}$$

which recovers the Stirling formula

$$n! \sim ne^{n\ln n}\sqrt{\frac{2\pi}{n}}e^{-n} = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

4. Two Orders of Laplace Correction: Further corrections can be obtained by using the Laplace's method. For example, computing two-order expansion using Laplace's method yields

$$\int_0^\infty e^{n(\ln y - y)}dy \sim \sqrt{\frac{2\pi}{n}}e^{-n}\left(1 + \frac{1}{12n}\right)$$

and gives Stirling's formula to two-orders as

$$n! \sim ne^{n\ln n}\sqrt{\frac{2\pi}{n}}e^{-n}\left(1 + \frac{1}{12n}\right) = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1 + \frac{1}{12n}\right)$$

5. Complex Analysis Using Cauchy's Integral: A complex analysis version of this method (Flajolet and Sedgewick (2009)) is to consider $\frac{1}{n!}$ as a Taylor coefficient of the exponential function

$$e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!}$$

computed by Cauchy's integral formula as

$$\frac{1}{n!} = \frac{1}{2\pi i} \oint_{|z|=r}^{|z|=r} \frac{e^z}{z^{n+1}} dz$$

6. Approximation Using the Saddle Point Method: The line integral can then be approximated using the addle point method with an appropriate choice of the contour radius

$$r = r_n$$

The dominant portion of the integral near the saddle point is then approximated by a real integral and Laplace's method, while the remaining portion of the integral can be bounded above to give an error term.

## Speed of Convergence and Error Estimates

1. Stirling Series for Factorial Approximation: Stirling's formula is in fact the first approximation to the following series called the *Stirling Series* (National Institute of Standards and Technology (2018)):

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \frac{1}{12n} + \frac{1}{288n^2} - \frac{139}{51840n^3} - \frac{571}{2488320n^4} + \cdots\right)$$

2. Structure of the Stirling Errors: An explicit formula for the coefficients of this series was given by Nemes (2010). As illustrated in Wikipedia (2019), the relative error vs. $n$ for 1 through 5 terms listed above have very characteristic error structure features.
3. Asymptotic Expansion of Stirling Series: As

$$n \to \infty$$

the error in the truncated series is asymptotically equal to the first term. This is an example of an asymptotic expansion.

4. Stirling Series is not Convergent: It is not a convergent series; for any *particular* value of $n$ there are only so many terms of the series that improve the accuracy, after which point the accuracy actually gets worse.
5. Errors due to Alternating Signs: Rewriting Stirling's series in the form

$$\ln n! \sim n \ln n - n + \frac{1}{2}\ln(2\pi n) + \frac{1}{12n} - \frac{1}{360n^3} + \frac{1}{1260n^5} - \frac{1}{1680n^7} + \cdots$$

it is known that the error in truncating the series is always of the opposite sign and at most the same magnitude of the first omitted term.

6. Precise Estimates of Lower/Upper Bounds: More precise bounds, due to Robbins (1955), that are valid for all positive integers $n$ are:

$$\sqrt{2\pi}\, n^{n+\frac{1}{2}}\, e^{-n}\, e^{\frac{1}{12n+1}} \le n! \le \sqrt{2\pi}\, n^{n+\frac{1}{2}}\, e^{-n}\, e^{\frac{1}{12n}}$$

## Stirling's Formula for the Gamma Function

1. Review – Gamma Function Definition: For all positive integers

$$n! = \Gamma(n+1)$$

   where $\Gamma$ denotes the gamma function.

2. Differences between Gamma and Factorial: However, the gamma function, unlike the factorial, is more broadly defined for all complex numbers other than non-positive integers; nevertheless, Stirling's formula may still be applied.

3. Starting Expression for $\ln \Gamma(z)$ Term: If

$$\mathrm{Re}(z) > 0$$

   then

$$\ln \Gamma(z) = z \ln z - z + \frac{1}{2} \ln \frac{2\pi}{z} + \int_{0}^{\infty} \frac{2 \arctan \frac{t}{z}}{e^{2\pi t} - 1} dt$$

4. Repeated Integration by Parts: Repeated integration by parts results in

$$\ln \Gamma(z) \sim z \ln z - z + \frac{1}{2} \ln \frac{2\pi}{z} + \sum_{n=1}^{N-1} \frac{B_{2n}}{2n(2n-1)z^{2n-1}}$$

   where $B_n$ is the $n^{th}$ Bernoulli number. Note that the limit of the sum as

$$N \to +\infty$$

   is not convergent, so this expression is just an asymptotic expansion.

5. $z$ Validity Range for Asymptotic Expansion: The formula is valid for $z$ large enough in absolute value when

$$|\arg(z)| < \pi - \varepsilon$$

where $\varepsilon$ is positive with an error term of $\mathcal{O}(z^{-2N+1})$. The corresponding approximation may now be written as

$$\Gamma(z) = \sqrt{\frac{2\pi}{z}}\left(\frac{z}{e}\right)^{z}\left[1 + \mathcal{O}\left(\frac{1}{z}\right)\right]$$

6. <u>Riemann-Siegel Theta Function over Varying Im$(z)$</u>: A further application of this asymptotic expansion is for complex argument $z$ with constant Re$(z)$. See, for example, the Stirling formula applied in

$$\text{Im}(z) = t$$

of the Riemann-Siegel theta function on the straight line $\frac{1}{4} + it$.

## Error Bounds

1. <u>Explicit Expression for the Residuals</u>: For any positive integer $N$, the following notation is introduced:

$$\ln\Gamma(z) \sim z\ln z - z + \frac{1}{2}\ln\frac{2\pi}{z} + \sum_{n=1}^{N-1}\frac{B_{2n}}{2n(2n-1)z^{2n-1}} + R_N(z)$$

and

$$\Gamma(z) = \sqrt{\frac{2\pi}{z}}\left(\frac{z}{e}\right)^{z}\left[\sum_{n=0}^{N-1}\frac{a_n}{z^n}+\tilde{R}_N(z)\right]$$

2. Residuals of $\Gamma(z)$ and $\ln\Gamma(z)$: Then, from Schafke and Sattler (1990) and Memes (2015):

$$|R_N(z)| \leq \frac{B_{2N}}{2N(2N-1)|z|^{2N-1}}\begin{cases} 1 & |\arg(z)| \leq \dfrac{\pi}{4} \\ |\csc(\arg z)| & \dfrac{\pi}{4} < |\arg(z)| < \dfrac{\pi}{2} \\ \sec^{2N}\left(\dfrac{\arg(z)}{2}\right) & |\arg(z)| < \pi \end{cases}$$

$$\left|\tilde{R}_N(z)\right| \leq \left(\frac{|a_N|}{|z|^N}+\frac{|a_{N+1}|}{|z|^{N+1}}\right)\begin{cases} 1 & |\arg(z)| \leq \dfrac{\pi}{4} \\ |\csc(\arg z)| & \dfrac{\pi}{4} < |\arg(z)| < \dfrac{\pi}{2} \end{cases}$$

## A Convergent Version of the Sterling's Formula

1. Convergent Version Using Raabe's Integral: Obtaining a convergent version of the Sterling's formula entails evaluating Raabe's integral:

$$\int_0^{\infty}\frac{2\arctan\frac{t}{x}}{e^{2\pi t}-1}dt = \ln\Gamma(x) - x\ln x + x - \frac{1}{2}\ln\frac{2\pi}{x}$$

2. Use of Inverted Rising Exponentials: One way to do this is by means of a convergent series of inverted rising exponentials. If

$$z^{\bar{n}} = z(z+1)\cdots(z+n-1)$$

then

$$\int_0^\infty \frac{2\arctan\frac{t}{x}}{e^{2\pi t}-1}\,dt = \sum_{n=1}^{\infty} \frac{C_n}{(x+1)^{\bar{n}}}$$

where

$$C_n = \frac{1}{n}\int_0^1 x^{\bar{n}}\left(x-\frac{1}{2}\right)dx = \frac{1}{2}\sum_{k=1}^{n} \frac{k|S(n,k)|}{(k+1)(k+2)}$$

where $S(n,k)$ denotes the Stirling's numbers of the first kind.

3. Convergent Version of the Stirling Series: From this, one obtains a version of the Stirling's series

$$\ln\Gamma(x) = x\ln x - x + \frac{1}{2}\ln\left(\frac{2\pi}{x}\right) + \frac{1}{12(x+1)} + \frac{1}{12(x+1)(x+2)} + \frac{59}{360(x+1)(x+2)(x+3)} + \frac{29}{60(x+1)(x+2)(x+3)(x+4)} + \cdots$$

which converges when

$$\mathrm{Re}(z) > 0$$

## Versions Suitable for Calculators

1. Hyperbolic Sine Function Based Version: The approximation

$$\Gamma(z) \approx \sqrt{\frac{2\pi}{z}}\left(\frac{z}{e}\sqrt{z\sinh\frac{1}{z}+\frac{1}{810z^6}}\right)^z$$

   and its equivalent form

$$2\ln\Gamma(z) \approx \ln(2\pi) - \ln z\left[2\ln z + \ln\left(z\sinh\frac{1}{z}+\frac{1}{810z^6}\right) - 2\right]$$

   can be obtained by re-arranging Stirling’s extended formula and observing the coincidence between the resultant power series and the Taylor series expansion of the hyperbolic sine function.

2. Memory Efficient Computation of Gamma: This approximation is good to more than 8 decimal digits for $z$ with a real part greater than 8. Toth (2016) indicates that this was suggested by Robert Windschitl in 2002 for computing gamma functions with fair accuracy on calculators with limited program or register memory.
3. Nemes Simplified Gamma Function Version: Nemes (2010) proposed an approximation which gives the same number of exact digits as the Windschitl approximation, but is much simpler.

$$\Gamma(z) \approx \sqrt{\frac{2\pi}{z}}\left[\frac{1}{e}\left(z+\frac{1}{12z-\frac{1}{10z}}\right)\right]^z$$

   or equivalently

$$\ln\Gamma(z) \approx \frac{1}{2}(\ln 2\pi - \ln z) + z\left[\ln\left(z + \frac{1}{12z - \frac{1}{10z}}\right) - 1\right]$$

4. <u>Ramanujan's Simplified Expression for $\Gamma(1+x)$</u>: An alternative approximation for the gamma function stated by Ramanujan is

$$\Gamma(1+x) \approx \sqrt{\pi}\left(\frac{x}{e}\right)^x \left(8x^3 + 4x^2 + x + \frac{1}{30}\right)^{\frac{1}{6}}$$

for

$$x \geq 0$$

5. <u>Ramanujan's Simplified Expression for $\ln n!$</u>: The equivalent approximation for $\ln n!$ Has an approximate error of $\frac{1}{1400n^3}$ and is given by

$$\ln n! \approx n \ln n - n + \frac{1}{6}\ln\left(8x^3 + 4x^2 + x + \frac{1}{30}\right) + \frac{1}{2}\ln\pi$$

6. <u>Ramanujan's Approximation - Lower/Upper Bounds</u>: The approximation may be made more precise by giving paired upper and lower bounds; one such inequality is (Karatsuba (2001), Mortici (2011a, 2011b, 2011c))

$$\sqrt{\pi}\left(\frac{x}{e}\right)^x \left(8x^3 + 4x^2 + x + \frac{1}{100}\right)^{\frac{1}{6}} < \Gamma(1+x) < \sqrt{\pi}\left(\frac{x}{e}\right)^x \left(8x^3 + 4x^2 + x + \frac{1}{30}\right)^{\frac{1}{6}}$$

## References

- Dutka, J. (1991): The Early History of the Factorial Function *Archive for History of Exact Sciences* **43 (3)** 225-249
- Flajolet, P., and R. Sedgewick (2009): *Analytic Combinatorics* **Cambridge University Press** New York
- Karatsuba, E. (2001): On the Asymptotic Representation of the Euler Gamma Function by Ramanujan *Journal of Computational and Applied Mathematics* **135 (2)** 225-240
- Le Cam, L. (1986): The Central Limit Theorem around 1935 *Statistical Science* **1 (1)** 78-96
- Mortici, C. (2011a): Ramanujan's Estimate for the Gamma Function via Monotonicity Arguments *Ramanujan Journal* **25 (2)** 149-154
- Mortici, C. (2011b): Improved Asymptotic Formulas for the Gamma Function *Computers and Mathematics with Applications* **61 (11)** 3364-3369
- Mortici, C. (2011c): On Ramanujan's Large Argument Formula for the Gamma Function *Ramanujan Journal* **26 (3)** 185-192
- National Institute of Standards and Technology (2018): NIST Digital Library of Mathematical Functions
- Nemes, G. (2010): On the Coefficients of the Asymptotic Expansion of $n!$ **arXiv**
- Pearson, K. (1924): Historical Note on the Origin of the Normal Curve of Errors *Biometrika* **16 (3)** 402-404
- Robbins, H. (1955): A Remark on Stirling's Formula *American Mathematical Monthly* **62 (1)** 26-29
- Schafke, F. W., and A. Sattler (1990): Restgliedabschatzungen fur die Sterlingsche Reine *Note di Matematica* **10 (2)** 453-470
- Toth V. T. (2016): Programmable Calculators – The Gamma Function
- Wikipedia (2019): Stirling's Approximation

# Lanczos Approximation

## Introduction

1. Numerical Approximation to Gamma Function: The Lanczos is a method for computing the gamma function numerically. It is a practical alternative to the more popular Stirling's approximation for calculating the gamma function with a fixed precision (Wikipedia (2019)).
2. Principal Expression for the Lanczos Approximation: The Lanczos approximation consists of the formula

$$\Gamma(z+1)=\sqrt{2\pi}\left(z+g+\frac{1}{2}\right)^{z+\frac{1}{2}}e^{-\left(z+g+\frac{1}{2}\right)}A_g(z)$$

   for the gamma function, with

$$A_g(z)=\frac{1}{2}p_0(g)+p_1(g)\frac{z}{z+1}+p_2(g)\frac{z(z-1)}{(z+1)(z+2)}+\cdots$$

3. Usage of the Control Constant $g$: Here, $g$ is a constant that may be chosen arbitrarily subject to the restriction that'

$$Re(z)>\frac{1}{2}$$

(Pugh (2004)). The coefficients $p_i$ are slightly more difficult to calculate, as shown in the next section.

4. Extension to the Full Complex Plane: Although the expression as states here is only valid for the components in the right complex half-plane, it can be extended to the entire half plane by the reflection formula

$$\Gamma(1-z)\Gamma(z) = \frac{\pi}{\sin \pi z}$$

5. Truncation to obtain Suitable Precision: The series $A$ is convergent and may be truncated to obtain an approximation to the desired precision. By choosing an appropriate $g$ – typically a small integer – only $5 - 10$ terms of the series are required to compute the gamma function with typical single or double floating-point precision.
6. Pre-calculation of the Series Coefficients: If a fixed $g$ is chosen, the coefficients can be calculated in advance, and the sum is re-cast into the following form:

$$A_g = c_0 + \sum_{k=1}^{N} \frac{c_k}{z+k}$$

7. Popularization by *Numerical Recipes*: Thus, computing the gamma function becomes a matter of evaluating only a small number of elementary functions and multiplying by stored constants. The Lanczos approximation was popularized by Press, Teukolsky, Vetterling, and Flannery (2007), according to whom computing the gamma function becomes “not much more difficult than other built-in functions taken for granted, such as $\sin x$ or $e^x$”. This method is also implemented in the GNU Scientific Library.

## Coefficients

1. Expression for the Lanczos Coefficients: The coefficients are given by

$$p_k(g) = \sum_{a=0}^{k} C(2k+1, 2a+1) \frac{\sqrt{2}}{\pi} \left(a - \frac{1}{2}\right)! \left(a + g + \frac{1}{2}\right)^{-\left(a+\frac{1}{2}\right)} e^{a+g+\frac{1}{2}}$$

   with $C(i,j)$ denoting the $(i,j)^{th}$ element of the Chebyshev polynomial coefficient matrix which can be calculated recursively from the identities

$$C(1,1) = 1$$

$$C(2,2) = 1$$

$$C(i,1) = -C(i-2,1) \quad i = 3, 4, \cdots$$

$$C(i,j) = -2C(i-1,j-1) \quad i = j = 3, 4, \cdots$$

$$C(i,j) = -2C(i-1,j-1) - C(i-2,j) \quad i > j = 2, 3, \cdots$$

2. Paul Godfrey's Coefficient Computation Scheme: Godfrey (2001) describes how to obtain the coefficients as well as the value of the truncated series $A$ as a matrix product.

## Derivation

1. Lanczos derived the formula from the Euler's integral

$$\Gamma(z+1)=\int_0^\infty t^z e^{-t} dt$$

performing a basic sequence of manipulations to obtain

$$\Gamma(z+1)=(z+g+1)^{z+1}e^{-(z+g+1)}\int_0^e [v(1-\log v)]^z v^g dv$$

and then deriving a series for the integral.

## References

- Godfrey, P. (2001): Lanczos Implementation of the Gamma Function
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (2007): *Numerical Recipes: The Art of Scientific Computing 3rd Edition* **Cambridge University Press** New York
- Pugh, G. R. (2004): *An Analysis of the Lanczos Gamma Approximation* Ph. D. **University of British Columbia**
- Wikipedia (2019): Lanczos Approximation

# Incomplete Gamma Function

## Introduction and Overview

1. Lower/Upper Incomplete Gamma Functions: In mathematics, the *upper* and the *lower incomplete gamma functions* are types of special functions which arise as solutions to various mathematical problems, such as certain integrals (Wikipedia (2019b)).
2. Origin of the Term *Incomplete*: Their respective names stem from their integral definitions, which are defined similarly to the gamma functions, but with different or *incomplete* integrals.
3. Lower Gamma Function Integration Limits: The gamma function is defined as an integral from zero to infinity. This contrasts with the lower incomplete gamma function, which is defined as an integral from zero to a variable upper limit.
4. Upper Gamma Function Integration Limits: Similarly, the upper incomplete gamma function is defined as an integral from a variable lower limit to infinity.

## Definition

The upper incomplete gamma function is defined as

$$\Gamma(s,x) = \int_{x}^{\infty} t^{s-1}e^{-t}dt$$

whereas the lower incomplete gamma function is defined as

$$\gamma(s,x) = \int_{0}^{x} t^{s-1}e^{-t}dt$$

## Properties

1. Positive Real Part of $s$: In both cases above, $s$ is a positive complex number, such that the real part of $s$ is positive.
2. Recurrence for Lower/Upper Gamma: On integrating by parts, one finds the recurrence relations

   $$\Gamma(s+1,x) = s\Gamma(s,x) + x^{s}e^{-x}$$

   and

   $$\gamma(s+1,x) = s\gamma(s,x) - x^{s}e^{-x}$$

3. Equivalence across Upper/Lower/Parent: Since the ordinary gamma function is defined as

$$\Gamma(s) = \int_{0}^{\infty} t^{s-1} e^{-t} dt$$

one has

$$\Gamma(s) = \Gamma(s, 0) = \lim_{x \to \infty} \Gamma(s, x)$$

and

$$\Gamma(s, x) + \gamma(s, x) = \Gamma(s)$$

## Continuation to Complex Values

1. Development of the Holomorphic Equivalents: The lower and the upper incomplete gamma functions, as defined above for real, positive $s$ and $x$, can be developed into holomorphic functions for both $x$ and $s$, as well as defined for almost all combinations of complex $x$ and $s$ (National Institute of Standards and Technology (2019a)).
2. Complex Analysis of Holomorphic Counterparts: Complex analysis shows how the properties of real incomplete gamma functions extend to their holomorphic counterparts.

## Lower Incomplete Gamma Function – Holomorphic Extensions

1. Recurrence Applied to Lower Gamma: Repeated application of the recurrence relation to the *lower incomplete gamma* function leads to the power series expansion (National Institute of Standards and Technology (2019a))

$$\gamma(s,x) = \sum_{k=0}^{\infty} \frac{x^s e^{-x} x^k}{s(s+1)\cdots(s+k)} = x^s \Gamma(s) e^{-x} \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(s+k-1)}$$

2. Well-definedness of the Sum: Given the rapid growth of the absolute value of $\Gamma(z+k)$ when

$$k \to \infty$$

   and the fact that the reciprocal of $\Gamma(z)$ is an entire function, the coefficients in the right-most sum are well-defined, and locally the sum converges uniformly for all complex $s$ and $x$.

3. Property of the Limiting Function: By the theorem of Weierstrass (Marshall (2009)), the limiting function, sometimes denoted as $\gamma^*$

$$\gamma^*(s,z) = e^{-z} \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(s+k-1)}$$

   (National Institute of Standards and Technology (2019a)) is entire with respect to both $z$ for fixed $s$ and $s$ for fixed $z$ (National Institute of Standards and Technology (2019a)), and, thus holomorphic on $\mathbb{C} \times \mathbb{C}$ by Hartog's theorem (Garrett (2005)).

4. Holomorphicity of the Lower Gamma Function: Hence, the following *decomposition*

$$\gamma(s,x) = x^s e^{-x} \gamma^*(s,x)$$

(National Institute of Standards and Technology (2019a)) extends the real lower incomplete gamma function as a holomorphic function, both jointly and separately in $z$ and $s$.

5. Singularities and Zeroes of $\gamma$: It follows from the properties of $z^s$ and the $\Gamma$-function, that the first two terms capture the singularities of $\gamma$ – at

$$z = 0$$

or at $s$ a non-positive integer – whereas the last term $\gamma^*$ contributes to its zeroes.

## Multi-Valuedness

1. Multi-Valuedness Inherent in Complex Logarithms: The complex logarithm

$$\log z = \log|z| + i \arg z$$

is determined only upto a multiple of $2\pi i$, which renders it multi-valued. Functions involving complex logarithms typically inherit this property. Among these are the complex power, and since $z^s$ appears in its decomposition, the $\gamma$ function will too.
2. Indeterminacy of Multi-Valued Functions: The indeterminacy of multi-valued functions introduces complications, since it must be stated how to select the value. Two strategies are presented below to handle this.
3. Replacement using $\mathbb{C} \times \mathbb{C}$ Riemann Manifold: The most general way is to replace the domain $\mathbb{C}$ of the multi-valued function by a suitable manifold in $\mathbb{C} \times \mathbb{C}$ called the Reimann surface. While this removes multi-valuedness, aspects of the theory behind it have to be closely followed (Teleman (2003)).

4. Decomposition into Single-Valued Branches: Restrict the domain such that the multi-valued function decomposes into separate single-valued branches, which can be handled individually.
5. Terminology Used in this Section: The following set of rules are used to interpret the formulations of this Section, unless explicitly stated otherwise.

## Sectors

1. Domains Suitable for Complex Expressions: Sectors in $\mathbb{C}$ having their vertex at

   $$z = 0$$

   often prove to be appropriate domains for complex expressions.
2. Sub-sector within the Main Sector $\mathbb{C}$: A sector $D$ consists of all complex $z$ fulfilling

   $$z \neq 0$$

   and

   $$\alpha - \delta < \arg z < \alpha + \delta$$

   with some $\alpha$, and

   $$0 < \delta \leq \pi$$

   Often $\alpha$ can be arbitrarily chosen, and is not specified.

3. Default Value for $\delta$: If $\delta$ is not given, it is often assumed to be $\pi$, and the sector is in fact the whole plane $\mathbb{C}$, with the exception of a half-line originating at

$$z = 0$$

and pointing into the direction of $-\alpha$, usually serving as a branch-cut.

4. Sector Centered around Positive Real: Note: In many applications and texts, $\alpha$ is silently taken to be $0$, which centers the sector around the positive real axis.

## Branches

1. Single-Valued and Holomorphic Logarithm: In particular, a single-valued and holomorphic logarithm exists on any such sector $D$ having its imaginary part bounded to the range $(\alpha - \delta, \alpha + \delta)$.
2. Single Valued and Holomorphic Incomplete Gamma: Based on such a restricted logarithm, $z^s$ and the incomplete gamma functions collapse to single-valued, holomorphic functions on $D$ ($\mathbb{C} \times D$), called branches of their multi-valued counterparts on $D$.
3. Fixing the Branch by Setting $\alpha$: Adding a multiple of $2\pi$ to $\alpha$ yields a different set of correlated branches on the same set $D$. However, in any given context here, $\alpha$ is assumed fixed, and all branches involved are associated with it.
4. Defining the Principal Branch: If

$$|\alpha| < \delta$$

the branches are called principal, because they equal their real analogons on the positive real axis. Note: In many applications and texts, the formulations hold only for principal branches.

## Relation between Branches

The values of different branches of both the complex power function and the lower incomplete gamma function can be easily derived from each other by multiplication of $e^{2\pi iks}$ (National Institute of Standards and Technology (2019a)).

## Behavior near the Branch Point

1. Asymptotic Gamma Behavior near $z = 0$: The decomposition above further shows that $\gamma$ behaves near

$$z = 0$$

asymptotically as

$$\gamma(z, x) \to z^{s}e^{-z}\gamma^{*}(z, 0) = z^{s}\frac{\Gamma(s)}{\Gamma(s+1)} = \frac{z^{s}}{s}$$

2. Real/Complex Realm Behavioral Differences: For positive and real $x$, $y$, and $s$,

$$\frac{z^y}{y} \to 0$$

when

$$(x, y) \to (0, s)$$

This seems to justify setting

$$\gamma(s, 0) = 0$$

for real

$$s > 0$$

However, matters are somewhat different in the complex realm.

3. Conditions to be Satisfied for Convergence: Only if a) the real part of $s$ is positive, and b) the values $u^v$ are taken from just a finite set of branches, are they guaranteed to converge to zero, as

$$(u, v) \to (0, s)$$

in which case, so does $\gamma(u, v)$.

4. Results for a Single Branch: On a single branch of $\gamma$, the criterion b) is naturally fulfilled, so

$$\gamma(s, 0) = 0$$

for $s$ with positive real part, and it is a continuous limit. Also note that such a continuation is by no means an analytic one.

## Algebraic Relations

1. Algebraic Relations and Differential Equations: All algebraic relations and differential equations observed by the real $\gamma(s,z)$ hold for its holomorphic counterpart as well.
2. Holomorphic Extension using Identity Theorem: This is a consequence of the identity theorem that states that equations between holomorphic functions valid on a real interval hold everywhere.
3. Branch Preservation of Recurrences/Derivatives: In particular, the recurrence relation and

$$\frac{\partial\gamma(s,z)}{\partial z} = z^{s-1}e^{-z}$$

(National Institute of Standards and Technology (2019a)) are preserved on their corresponding branches.

## Integral Representation

1. $\gamma$ as a Holomorphic Anti-derivative: The last relation clearly indicates that, for a fixed $s$, $\gamma$ is a primitive or an anti-derivative of the holomorphic function $z^{s-1}e^{-z}$.
2. Integration inside a Single Branch: Consequently, for any complex

$$u, v \neq 0$$

$$\int_{u}^{v} t^{s-1}e^{-t}dt = \gamma(s, v) - \gamma(s, u)$$

holds as long as the path of the integration is contained entirely inside the domain of a branch of the integrand.
3. Complex Integral Definition of $\gamma$: If, additionally, the real part of $s$ is positive, then the limit

$$\gamma(s, u) \to 0$$

for

$$u \to 0$$

applies, finally aiming at the complex integral definition of $\gamma$

$$\gamma(s, z) = \int_{0}^{z} t^{s-1}e^{-t}dt$$

$$Re(s) > 0$$

(National Institute of Standards and Technology (2019a)).

4. Inclusion of $z = 0$ at Start: Any path of the integration containing only $0$ at the beginning, otherwise restricted to the domain of a branch of the integrand, is valid here, for example, the line connecting $0$ and $z$.

## Limit for $z \rightarrow \pm\infty$ - Real Values

Given the integral representation of the principal branch of $\gamma$, the equation holds for all positive real $s$ and $x$.

$$\Gamma(s, z) = \int_{0}^{\infty} t^{s-1} e^{-t} dt = \underset{x \rightarrow \infty}{Limit} \gamma(s, z)$$

(National Institute of Standards and Technology (2019b)).

## Limit for $z \rightarrow \pm\infty$ - Complex Values

1. Extending $z \rightarrow \pm\infty$ Limit to Complex $s$: This result extends to complex $s$. Assume first

$$1 \leq Re(s) \leq 2$$

and

$$1 < a < b$$

Then

$$|\gamma(s,b) - \gamma(s,a)| \leq \int_a^b |t^{s-1}| e^{-t} dt = \int_a^b t^{\Re s-1} e^{-t} dt \leq \int_a^b te^{-t} dt$$

where

$$|z^s| = \left|z^{\Re s}\right| \cdot e^{-\Im s \arg z}$$

(National Institute of Standards and Technology (2019c)) has been used in the middle.

2. Skip Convergence as $x \to +\infty$: Since the final integral becomes arbitrarily small if only $a$ is small enough, $\gamma(s,x)$ converges uniformly for

   $$x \to +\infty$$

   on the strip

   $$1 \leq Re(s) \leq 2$$

   towards a holomorphic function (Marshall (2009)), which must be $\Gamma(s)$ because of the identity theorem.

3. Convergence Outside the Strip as $x \to +\infty$: Taking the limit in the recurrence relation

   $$\gamma(s,x) = (s-1)\gamma(s-1,x) - x^{s-1}e^{-x}$$

   and noting that

$$\underset{x \to \infty}{Limit} x^{s-1}e^{-x} \to 0$$

for all $n$ shows that $\gamma(s,x)$ converges outside the strip too, towards a function obeying the recurrence relation of the $\Gamma$-function.

4. $\underline{\Gamma(s)}$ Limit Using the Identity Theorem: It follows that

$$\Gamma(s) = \underset{x \to \infty}{Limit} \gamma(s,x)$$

for all complex $s$ not a non-positive integer, $x$ real, and $\gamma$ principal.

## Sector-wise Convergence

1. $\underline{\gamma(s,u)}$ Convergence inside Principal Sector: Let $u$ be from the sector

$$\arg z < \delta < \frac{\pi}{2}$$

with some fixed $\delta$ keeping

$$\alpha = 0$$

$\gamma$ is the principal branch on this sector, and this section looks at

$$\Gamma(s) - \gamma(s,u) = \Gamma(s) - \gamma(s,|u|) + \gamma(s,|u|) - \gamma(s,u)$$

2. <u>Gamma Difference across Imaginary $u$</u>: As shown above, the first difference can be made arbitrarily small if $|u|$ is sufficiently large. The second difference allows for the following estimation:

$$\gamma(s, |u|) - \gamma(s, u) \leq \int_{u}^{|u|} |x^{z-1} e^{-z}| dz = \int_{u}^{|u|} |z|^{\Re s} \cdot e^{-\Im s \arg z} \cdot e^{-\Re s} dz$$

where the integral representation of $\gamma$ and the expression regarding $|z|^{s}$ have been used.

3. <u>Integrating across the Arc</u>: On integrating along the arc with radius

$$R = |u|$$

around 0 connecting $u$ and $|u|$, the last integral becomes

$$\int_{u}^{|u|} |z|^{\Re s} \cdot e^{-\Im s \arg z} \cdot e^{-\Re s} dz \leq R \cdot |\arg u| \cdot R^{\Re s - 1} \cdot e^{\Im s |\arg u|} \cdot e^{-R \cos(\arg u)}$$

$$\leq \delta \cdot R^{\Re s} \cdot e^{\Im s \delta} \cdot e^{-R \cos \delta} = M \cdot (R \cos \delta)^{\Re s} \cdot e^{-R \cos \delta}$$

where

$$M = \delta (\cos \delta)^{-Re(s)} e^{-Im(s\delta)}$$

is a constant independent of $u$ or $R$. Again, using

$$\underset{x \to \infty}{Limit}\, x^{s-1} e^{-x} \to 0$$

for all $n$, it can be seen that the last expression approaches $0$ as $R$ increases towards $+\infty$.

4. Convergence on the Principal Domain: In total, one now has

$$\Gamma(s) = \underset{|z| \to \infty}{Limit} \gamma(s, z)$$

$$|\arg z| < \frac{\pi}{2} - \epsilon$$

if $s$ is not a non-negative integer,

$$0 < \epsilon < \frac{\pi}{2}$$

is arbitrarily small but fixed, and $\gamma$ denotes the principal branch on this domain.

## Lower Incomplete Gamma Function – Overview

$\gamma(s, z)$ is:

i. Entire in $z$, for a fixed positive integer $s$
ii. Multi-valued holomorphic in $z$ for fixed $s$ not an integer
iii. On each branch, it is meromorphic in $s$ for fixed

$$z \neq 0$$

with simple poles at non-positive integers $s$.

## Upper Incomplete Gamma Function

1. <u>Upper Incomplete Gamma – Holomorphic Extension</u>: As for the *upper incomplete gamma function*, a holomorphic extension, with respect to $z$ or $s$, is given by

   $$\Gamma(s, z) = \Gamma(s) - \gamma(s, z)$$

   (National Institute of Standards and Technology (2019a)) at points $(s, z)$ where the right-hand side exists.
2. <u>Single-Valued, and Restricted to Principal Branch</u>: Since $\gamma$ is multi-valued, the same holds for $\Gamma$, but a restriction to principal values only yields the single-valued principal branch of $\Gamma$.
3. <u>Behavior under Non-Positive $s$</u>: When $s$ is a non-positive integer in the above equation, neither part of the difference is defined, and a limiting process, developed here for

   $$s \to 0$$

   fills in the missing values.
4. <u>Upper Gamma Bound as $s \to 0$</u>: Complex analysis guarantees holomorphicity, since $\Gamma(s, z)$ proves to be bounded in the neighborhood of

   $$s \to 0$$

   for fixed $z$.
5. <u>Power Series Expansion for $\gamma$</u>: To determine that limit, the power series of $\gamma^*$ at

$$z = 0$$

turns out useful. When replacing $e^{-x}$ by its power series in the integral definition of $\gamma$, one obtains – assuming $x$ and $s$ to be positive reals for now –

$$\gamma(s,z) = \int_0^x t^{s-1}e^{-t}dt = \int_0^x \sum_{k=0}^{\infty}(-1)^k \frac{t^{s+k-1}}{k!}dt = \sum_{k=0}^{\infty}(-1)^k \frac{x^{s+k}}{k!\,(s+k)}$$
$$= x^s \sum_{k=0}^{\infty} \frac{(-x)^k}{k!\,(s+k)}$$

or

$$\gamma^* = \sum_{k=0}^{\infty} \frac{(-x)^k}{k!\,(s+k)}$$

(National Institute of Standards and Technology (2019a)) which, as a series representation of the entire $\gamma^*$ function, converges for all complex $x$ (and all complex $s$ not a non-positive integer).

6. Extending from $s$ to $z$: With its restriction to real values lifted, the series allows the expansion:

$$\gamma(s,z) - \frac{1}{s} = -\frac{1}{s} + z^s \sum_{k=0}^{\infty} \frac{(-z)^k}{k!\,(s+k)} = \frac{z^s - 1}{s} + z^s \sum_{k=1}^{\infty} \frac{(-z)^k}{k!\,(s+k)}$$

$$\Re(s) > -1$$

$$s \neq 0$$

7. <u>$s \rightarrow 0$ Limit for Upper Gamma</u>: When

$$s \rightarrow 0$$

$$\frac{z^{s}-1}{s} \rightarrow \ln z$$

$$\Gamma(s)-\frac{1}{s}=\frac{1}{s}-\gamma+\mathcal{O}(s)-\frac{1}{s} \rightarrow -\gamma$$

(Wikipedia (2019a)) where $\gamma$ is the Euler-Mascheroni constant, hence

$$\Gamma(0, z)=\underset{s \rightarrow 0}{Limit}\left[\Gamma(s)-\frac{1}{s}-\left\{\gamma(s, z)-\frac{1}{s}\right\}\right]=-\gamma-\ln z-\sum_{k=1}^{\infty} \frac{(-z)^{k}}{k!\,k}$$

is the limiting function to the upper incomplete gamma function as

$$s \rightarrow 0$$

also known as the exponential integral $E_1(z)$ (National Institute of Standards and Technology (2019a)).

8. <u>Recurrence Relation Based Values for $\Gamma(-n, z)$</u>: By the way of recurrence relation, the values of $\Gamma(-n, z)$ for positive integers can be obtained from this result (National Institute of Standards and Technology (2019a))

$$\Gamma(-n, z)=\frac{1}{n!}\left[\frac{e^{-z}}{z^{n}} \sum_{k=0}^{n-1}(-1)^{k}(n-k-1)!\,z^{k}+(-1)^{n} \Gamma(0, z)\right]$$

Thus, the upper incomplete gamma function proves to exist and be holomorphic, with respect to both $z$ and $s$, for all $s$ and

$$z \neq 0$$

9. Characteristics of Upper Gamma Function: $\Gamma(s, z)$ is:
    a. Entire in $z$ for fixed positive integer $s$.
    b. Multi-valued holomorphic in $z$ for fixed $s$ not zero and not a positive integer, with a branch point at-

$$z = 0$$

c.

$$\Gamma(s, z) = \Gamma(s)$$

for $s$ with positive real part and

$$z = 0$$

– the limit when

$$(s_i, z_i) \rightarrow (s, 0)$$

but this is a continuous extension, not an *analytic* one, i.e., it does not hold for

$$s < 0$$

## Special Values

1. Compendium of Special $\Gamma(s,x)$ Values:

a.

$$\Gamma(s) = (s-1)!$$

if $s$ is a positive integer

b.

$$\Gamma(s,x) = (s-1)!\, e^{-x} \sum_{k=0}^{s-1} \frac{x^k}{k!}$$

if s is a positive integer

c.

$$\Gamma(s,0) = \Gamma(s)$$

$$\Re(s) > 0$$

d.

$$\Gamma(1,x) = e^{-x}$$

e.

$$\gamma(1,x) = 1 - e^{-x}$$

f.

$$\Gamma(0,x) = -Ei(-x)$$

for

$$x > 0$$

g.

$$\Gamma(s,x) = x^2 E_{1-s}(x)$$

h.

$$\Gamma\left(\frac{1}{2}, x\right) = \sqrt{\pi}\, erfc\left(\sqrt{x}\right)$$

i.

$$\gamma\left(\frac{1}{2}, x\right) = \sqrt{\pi}\, erf\left(\sqrt{x}\right)$$

2. <u>Terminologies Associated with Special Functions</u>: Here, $Ei$ is the exponential integral, $E_n$ is the generalized exponential integral, $erf$ is the error function, and $erfc$ is the complementary error function

$$erfc(x) = 1 - erf(x)$$

## Asymptotic Behavior

a.

$$\frac{\gamma(s, x)}{x^s} \rightarrow \frac{1}{s}$$

as

$$x \rightarrow 0$$

b.

$$\frac{\Gamma(s, x)}{x^s} = -\frac{1}{s}$$

as

$$x \to 0$$

and

$$\Re(s) > 0$$

For real $s$, the error of

$$\Gamma(s,x) = -\frac{x^s}{s}$$

is of the order of $\mathcal{O}\left(x^{\min(s+1,0)}\right)$ if

$$s \neq -1$$

and $\mathcal{O}(\ln x)$ if

$$s = -1$$

c.

$$\gamma(s,x) \to \Gamma(s)$$

as

$$x \to \infty$$

d.

$$\frac{\Gamma(s,x)}{x^{s-1}e^{-x}} \to 1$$

as

$$x \to \infty$$

e.

$$\Gamma(s,z) \to z^{s-1}e^{-z}\sum_{k=0}^{\infty}\frac{\Gamma(s)}{\Gamma(s-k)}z^{-k}$$

as an asymptotic series where

$$|z| \to \infty$$

and

$$|\arg z| < \frac{3\pi}{2}$$

(National Institute of Standards and Technology (2019a)).

## Evaluation Formulas

1. Lower Gamma Power Series Expansion: The lower incomplete gamma function can be evaluated using the power series expansion (National Institute of Standards and Technology (2019a))

$$\gamma(s,z) = \sum_{k=0}^{\infty} \frac{z^s e^{-z} z^k}{s(s+1)\cdots(s+k)} = z^s e^{-z} \sum_{k=0}^{\infty} \frac{z^k}{s^{\overline{k+1}}}$$

where $s^{\overline{k+1}}$ is the Pochhammer symbol.

2. Using Kummer's Confluent Hyper-geometric Function: An alternative expansion is

$$\gamma(s,z) = \sum_{k=0}^{\infty} \frac{(-1)^k z^{s+k}}{k(s+k)} = \frac{z^s}{s} M(s,s+1,z)$$

where $M$ is Kummer's Confluent Hyper-geometric function.

## Connection with Kummer's Confluent Hyper-geometric Function

1. Positive Real Part of $z$: When the real part of $z$ is positive,

$$\gamma(s,z) = s^{-1} z^s e^{-z} M(1,s+1,z)$$

where

$$M(1,s+1,z) = 1 + \frac{z}{s+1} + \frac{z^2}{(s+1)(s+2)} + \frac{z^3}{(s+1)(s+2)(s+3)} + \cdots$$

has an infinite radius of convergence.

2. Upper Gamma from Kummer's Identity: Again, using confluent hyper-geometric functions and employing Kummer's identity

$$\Gamma(s,z) = e^{-z}U(1-s,1-s,z) = \frac{z^s e^{-z}}{\Gamma(1-s)}\int_0^\infty \frac{e^{-u}}{u^s(z+u)}du = e^{-z}z^s U(1,1+s,z)$$

$$= e^{-z}\int_0^\infty e^{-u}(z+u)^{s-1}du = e^{-z}z^s\int_0^\infty e^{-zu}(1+u)^{s-1}du$$

3. <u>$\gamma$ from Continued Gauss Fraction</u>: For actual computation of numerical values, Gauss' continued fraction provides a useful expansion:

$$\gamma(s,z) = \cfrac{z^s e^{-z}}{s - \cfrac{sz}{s+1+\cfrac{z}{s+2-\cfrac{(s+1)z}{s+3+\cfrac{2z}{s+4-\cfrac{(s+2)z}{s+5+\cfrac{3z}{s+6-\ddots}}}}}}}$$

4. <u>Convergence for all Complex $z$</u>: The above continuous fraction converges for all complex $z$, provided that $s$ is not a negative integer.
5. <u>$\Gamma(s,z)$ from Continued Fraction Expressions</u>: The upper gamma function has the continued fraction expression (Abramowitz and Stegun (2007))

$$\Gamma(s,z) = \cfrac{z^s e^{-z}}{z+\cfrac{1-s}{1+\cfrac{1}{z+\cfrac{2-s}{1+\cfrac{2}{z+\cfrac{3-s}{1+\ddots}}}}}}$$

$$\Gamma(s,z)=\cfrac{z^s e^{-z}}{1+z-s+\cfrac{s-1}{3+z-s+\cfrac{2(s-2)}{5+z-3+\cfrac{3(s-3)}{7+z-s+\cfrac{4(s-4)}{9+z-s+\ddots}}}}}$$

## Multiplication Theorem

The following multiplication theorem holds true:

$$\Gamma(s,z)=\frac{1}{t^s}\sum_{i=0}^{\infty}\frac{\left(1-\frac{1}{t}\right)^i}{i!}\Gamma(s+i,tz)=\Gamma(s,tz)-(tz)^s e^{-tz}\sum_{i=0}^{\infty}\frac{\left(\frac{1}{t}-1\right)^i}{i}L_{i-1}^{s-1}(tz)$$

## Software Implementation

1. Availability in Computer Algebra Systems: The incomplete gamma functions are available in a variety of computer algebra systems.
2. $\gamma$ from Other Special Functions: Even if unavailable directly, however, incomplete function values can be calculated using functions commonly included in spreadsheets and computer algebra packages. In Excel, for example, these can be calculated using the Gamma function combined with the Gamma distribution function.
3. $\gamma$ from Cumulative Distribution Function: The lower incomplete function is given by

$$\gamma(s,x) = \exp\big(GammaLn(s)\big) \times Gamma.Dist(x,s,1,TRUE)$$

The upper incomplete function is given by

$$\Gamma(s,x) = \exp\big(GammaLn(s)\big) \times [1 - Gamma.Dist(x,s,1,TRUE)]$$

## Regularized Gamma Functions and Poisson Random Variables

1. Regularized Gamma Functions $P$ and $Q$: Two related functions are the regularized Gamma functions:

$$P(s,x) = \frac{\gamma(s,x)}{\Gamma(s)}$$

$$Q(s,x) = \frac{\Gamma(s,x)}{\Gamma(s)} = 1 - P(s,x)$$

2. CDF for Random Gamma Variables: $P(s,x)$ is the cumulative distribution function for Gamma random variables with shape parameter $s$ and scale parameter $1$.
3. CDF for Poisson Random Variables: When $s$ is an integer, $Q(s,\lambda)$ is the cumulative distribution function for Poisson random variables. If $x$ is a $Poisson(\lambda)$ random variable, then

$$\mathbb{P}[X,s] = \sum_{j<s} e^{-\lambda}\frac{\lambda^j}{j!} = \frac{\Gamma(s,\lambda)}{\Gamma(s)} = Q(s,\lambda)$$

This formula can be derived by repeated integration by parts.

## Derivatives

1. Upper Gamma Function wrt $x$: Using the integral representation above, the derivative of the upper gamma incomplete function $\Gamma(s, x)$ with respect to $x$ is

$$\frac{\partial \Gamma(s, x)}{\partial x} = -x^{s-1} e^{-x}$$

2. Upper Gamma Function wrt $s$: The derivative with respect to its first argument $x$ is given by (Geddes, Glasser, Moore, and Scott (1990))

$$\frac{\partial \Gamma(s, x)}{\partial s} = \ln x \Gamma(s, x) + x T(3, s, x)$$

and the second derivative by

$$\frac{\partial^2 \Gamma(s, x)}{\partial s^2} = \ln^2 x \Gamma(s, x) + 2x[\ln x T(3, s, x) + T(4, s, x)]$$

where the function $T(m, s, x)$ is a special case of the Meijer $G$-function

$$T(m, s, x) = G_{m-1,m}^{m,0}\left(\begin{matrix} 0, 0, \cdots, 0 \\ s-1, -1, \cdots, -1 \end{matrix} \middle| x\right)$$

3. Special Case of Meijer G-Function: This particular function has internal *closure* properties of its own because it can be used to express all successive derivatives.

4. Derivative of $\Gamma(s,x)$ wrt $s$: In general,

$$\frac{\partial^m \Gamma(s,x)}{\partial s^m} = \ln^m x\Gamma(s,x) + mx\ln xT(3+n,s,x)\sum_{n=0}^{m-1} P_n^{m-1}\ln^{m-n-1} xT(3+n,s,x)$$

where $P_j^n$ is the permutation defined by the Pochhammer symbol

$$P_j^n = \binom{n}{j} n! = \frac{n!}{(n-j)!}$$

5. $\Gamma(s,x)$ Derivatives in Succession Form: All such derivatives can be generated in the succession form

$$\frac{\partial T(m,s,x)}{\partial s} = \ln xT(m,s,x) + (m-1)T(m+1,s,x)$$

and

$$\frac{\partial T(m,s,x)}{\partial x} = -\frac{1}{x}[T(m-1,s,x) + T(m,s,x)]$$

6. Computing the Special Function $T(m,s,z)$: This function $T(m,s,z)$ can be computed from its series representation valid for

$$|z| < 1$$

$$T(m,s,z) = -\frac{(-1)^{m-1}}{(m-2)!}\left\{\frac{d^{m-2}}{dt^{m-2}}[\Gamma(s-t)z^{t-1}]\right\}\Bigg|_{t=0} + \sum_{n=0}^{\infty}\frac{(-1)^n z^{s-1+n}}{n!\,(-s-n)^{m-1}}$$

with the understanding that $s$ is not a negative number or zero.

7. Handling Negative or Zero $s$: In such a case, one my use a limit. Results for

$$|z| \geq 1$$

can be obtained by analytic continuation.

8. Simplification under Certain Special Cases: Some special cases of this function can be simplified. For example,

$$T(2, s, x) = \frac{T(s, x)}{x}$$

and

$$xT(3, 1, x) = E_1(x)$$

where $E_1(x)$ is the Exponential Integral.

9. Functions Dependent on $\Gamma(s, x)$ Derivatives: These derivatives and the function $T(m, s, x)$ provide exact solutions to a number of integrals by repeated differentiation of the integral definition of the upper incomplete gamma function (Milgram and Milgram (1985), Mathar (2010)). For example,

$$\int_x^\infty e^{-t} t^{s-1} \ln^m t \, dt = \frac{d^m}{ds^m} \int_x^\infty e^{-t} t^{s-1} dt = \frac{d^m}{ds^m} \Gamma(s, x)$$

10. Class of Laplace/Mellin Transforms: This formula can be further *inflated* or generalized to a huge class of Laplace transforms and Mellin transforms.

11. Definite Integrals in Computer Algebra: When combined with a computer algebra system, the exploitation of special functions provides a powerful method for solving definite integrals, in particular those encountered by practical engineering applications.

## Indefinite and Definite Integrals

1. Incomplete Gamma Function Power Integrand: The following indefinite integrals are readily obtained using integration by parts – with the constant of integration omitted in both cases.

$$\int x^{b-1}\gamma(s,x)dx = \frac{1}{b}[x^b\gamma(s,x) + \Gamma(s+b,x)]$$

$$\int x^{b-1}\Gamma(s,x)dx = \frac{1}{b}[x^b\Gamma(s,x) - \Gamma(s+b,x)]$$

2. Lower/Upper Gamma Fourier Connection: The lower and the upper incomplete Gamma function are connected via the Fourier transform

$$\int_{-\infty}^{+\infty} \frac{\gamma\left(\frac{s}{2}, \pi z^2\right)}{(\pi z^2)^{\frac{s}{2}}} e^{-2\pi ikz} dz = \frac{\gamma\left(\frac{1-s}{2}, \pi k^2\right)}{(\pi k^2)^{\frac{1-s}{2}}}$$

This follows, for example, using a suitable specialization of the technique illustrated in Gradshteyn, Ryzhik, Geronimus, Tseytlin, and Jeffrey (2015).

## References

- Abramowitz, M., and I. A. Stegun (2007): *Handbook of Mathematics Functions* **Dover Book on Mathematics**
- Garrett, P. (2005): Hartog's Theorem: Separate Analyticity Implies Joint
- Geddes, K. O., M. L. Glasser, R. A. Moore, and T. C. Scott (1990): Evaluation of Classes of Definite Integrals involving Elementary Functions via Differentiation of Special Functions *Applicable Algebra in Engineering, Communications, and Computing* **1 (2)** 149-165
- Gradshteyn, I. S., I. M. Ryzhik, Y. V. Geronimus, M. Y. Tseytlin, and A. Jeffrey (2015): *Tables of Integrals, Series, and Products* **Academic Press**
- Marshall, D. E. (2009): Complex Analysis
- Mathar, R. J. (2010): Numerical Evaluation of the Oscillatory Integral over $e^{i\pi x}x^{\frac{1}{x}}$ between 1 and ∞ **arXiV**
- Milgram, and M. S. Milgram (1985): The Generalized Integro-exponential Function *Mathematics of Computation* **44 (170)** 443-458
- National Institute of Standards and Technology (2019a): Incomplete Gamma and Related Functions
- National Institute of Standards and Technology (2019b): Gamma Functions
- National Institute of Standards and Technology (2019c): Elementary Functions
- Teleman, C. (2003): Riemann Surfaces
- Wikipedia (2019a): Gamma Function
- Wikipedia (2019b): Incomplete Gamma Functions

# Digamma Function

## Introduction and Overview

1. Definition of the Digamma Function: The *digamma function* is defined as the logarithmic derivative of the gamma function (Abramowitz and Stegun (2007), Wikipedia (2019)).

$$\psi(x) = \frac{d}{dx}\log\Gamma(x) = \frac{\Gamma'(x)}{\Gamma(x)}$$

   This is the first of the polygamma function.

2. Representation of the Digamma Function: The digamma function is often denoted as $\psi_0(x)$, $\psi^{(0)}(x)$, or F - the upper-case form of the archaic Greek consonant digamma meaning double gamma.

## Relation to the Harmonic Series

1. Equation of the Gamma Function: The gamma function obeys the equation

$$\Gamma(z+1) = z\Gamma(z)$$

2. <u>Derivative of the Gamma Function</u>: Taking derivative with respect to $z$ gives

$$\Gamma'(z+1) = z\Gamma'(z) + \Gamma(z)$$

3. <u>Recurrence Relation for the Digamma Function</u>: Dividing by $\Gamma(z+1)$ or the equivalent $z\Gamma(z)$ results in:

$$\frac{\Gamma'(z+1)}{\Gamma(z+1)} = \frac{\Gamma'(z)}{\Gamma(z)} + \frac{1}{z}$$

or

$$\psi(z+1) = \psi(z) + \frac{1}{z}$$

4. <u>Digamma Function using Harmonic Series</u>: Since the harmonic series are defined for positive integers $n$ as

$$H_n = \sum_{k=1}^{n} \frac{1}{k}$$

the digamma function is related to them by

$$\psi(n) = H_{n-1} - \gamma$$

where

$$H_0 = 0$$

and $\gamma$ is the Euler-Mascheroni constant.

5. Digamma Function for Half-Integers: For half-integer arguments, the digamma function takes the values

$$\psi\left(n+\frac{1}{2}\right) = -\gamma - 2\log 2 + \sum_{k=1}^{n} \frac{2}{2k-1}$$

## Integral Representations

1. Gauss Integral for Digamma Function: if the real part of $z$ is positive, then the digamma function has the following integral representation due to Gauss (Whittaker and Watson (1996)):

$$\psi(z) = \int_0^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-zt}}{1-e^{-t}}\right) dt$$

2. Incorporating Euler Mascheroni Integral Identity: Combining the above expression with an integral identity for the Euler-Mascheroni constant $\gamma$ gives:

$$\psi(z+1) = -\gamma + \int_0^\infty \frac{1-t^z}{1-t} dt$$

3. Difference between the Harmonic Terms: The integral is Euler's harmonic number $H_z$, so the previous formula may also be written as

$$\psi(z+1) = -\gamma + H_z$$

A consequence is the following generalization of the recurrence relation:

$$\psi(w+1) - \psi(z+1) = H_w - H_z$$

4. Digamma Function Dirichlet Integral Representation: An integral representation due to Dirichlet is (Whittaker and Watson (1996)):

$$\psi(z) = \int\limits_0^\infty \left(e^{-t} - \frac{1}{[1-t]^z}\right)\frac{dt}{t}$$

5. Asymptotic Integral for Digamma Function: Gauss' integral representation can be manipulated to give the start of the asymptotic expansion of $\psi(z)$ (Whittaker and Watson (1996)):

$$\psi(z) = \log z - \frac{1}{z} - \int\limits_0^\infty \left(\frac{1}{2} - \frac{1}{t} + \frac{1}{e^t - 1}\right) e^{-zt} dt$$

6. Gamma Function using Binet's First Integral: The above expression is a consequence of the Binet's first integral for the gamma function. The integral may be recognized as Laplace transform.
7. Gamma Function Binet's Second Integral: Binet's second integral for the gamma function gives a different formula for $\psi(z)$, which also gives the first few terms of the asymptotic expansion (Whittaker and Watson (1996)):

$$\psi(z) = \log z - \frac{1}{2z} - 2\int\limits_0^\infty \frac{t}{(z^2 + t^2)(e^{2\pi t} - 1)} dt$$

## Infinite Product Representation

The function $\frac{\psi(z)}{\Gamma(z)}$ is an entire function (Mezo and Hoffman (2017)), and it can be represented by the infinite product

$$\frac{\psi(z)}{\Gamma(z)} = -e^{2\gamma z} \prod_{k=0}^{\infty} \left(1 - \frac{z}{x_k}\right) e^{\frac{z}{x_k}}$$

Here $x_k$ is the $k^{th}$ zero of $\psi(z)$ (see below).

## Series Formula

1. Euler Product Formula Based Digamma Function: Euler's product formula for digamma function, combined with the functional equation and an identity for the Euler Mascheroni constant, yields the following expression for the digamma function, valid in the complex plane outside the negative integers (Abramowitz and Stegun (2007)):

$$\psi(z+1) = -\gamma + \sum_{n=1}^{\infty} \left(\frac{1}{n} - \frac{1}{n+z}\right) = -\gamma + \sum_{n=1}^{\infty} \frac{z}{n(n+z)}$$

$$z \neq -1, -2, \cdots$$

2. Alternate Euler Product Digamma Formula: Equivalently,

$$\psi(z) = -\gamma + \sum_{n=0}^{\infty}\left(\frac{1}{n+1} - \frac{1}{n+z}\right) = -\gamma + \sum_{n=1}^{\infty}\frac{z-1}{(n+1)(n+z)}$$

$$z \neq 0, -1, -2, \cdots$$

## Evaluation of Sums of Rational Functions

1. Sum of Rational Functions – Setup: The above identity can be used to evaluate sums of the form

$$\sum_{n=0}^{\infty} u_n = \sum_{n=0}^{\infty}\frac{p(n)}{q(n)}$$

where $p(n)$ and $q(n)$ are polynomials of $n$.

2. Criterion for Rational Series Convergence: For the series to converge,

$$\underset{n \to \infty}{Limit}\, n u_n \to 0$$

otherwise the series will be greater than the harmonic series, and thus diverge.

3. Digamma Based Rational Series Sum: Hence,

$$\sum_{k=1}^{m} a_k = 0$$

and

$$\begin{aligned}\sum_{n=0}^{\infty} u_n = \sum_{n=0}^{\infty}\sum_{k=1}^{m} \frac{a_k}{n+b_k} &= \sum_{n=0}^{\infty}\sum_{k=1}^{m} a_k\left(\frac{1}{n+b_k} - \frac{1}{n+1}\right) \\ &= \sum_{k=1}^{m}\left[a_k \sum_{n=0}^{\infty}\left(\frac{1}{n+b_k} - \frac{1}{n+1}\right)\right] = -\sum_{k=1}^{m} a_k[\psi(b_k)+\gamma] \\ &= -\sum_{k=1}^{m} a_k\psi(b_k)\end{aligned}$$

4. <u>Polygamma Based Rational Series Sum</u>: With the series expansion of higher rank polygamma function, a generalized formula may be given as

$$\sum_{n=0}^{\infty} u_n = \sum_{n=0}^{\infty}\sum_{k=1}^{m} \frac{a_k}{(n+b_k)^{r_k}} = \sum_{k=1}^{m} \frac{(-1)^{r_k}}{(r_k-1)!} a_k\, \psi^{r_k-1}(b_k)$$

provided the series on the left converges.

## Taylor Series

1. <u>Digamma using Rational Zeta Series</u>: The digamma has a rational zeta series, given by the Taylor series at

$$z = 1$$

This is

$$\psi(z+1) = -\gamma + \sum_{k=1}^{\infty} \zeta(k+1)(-z)^k$$

which converges for

$$|z| < 1$$

2. Derivation of the Taylor Series Above: Here, $\zeta(n)$ is the Riemann Zeta function. This series is easily derived from the corresponding Taylor's series and the Hurwitz zeta function.

## Newton Series

1. Newton/Stern Series for Digamma: The Newton series for digamma, sometimes referred to as the *Stern series* (Norlund (1924), Blagouchine (2018)), reads as

$$\psi(z+1) = -\gamma + \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \binom{s}{k}$$

where $\binom{s}{k}$ is the binomial coefficient.

2. Generalization of the above Series: It may also be generalized to

$$\psi(z+1) = -\gamma - \frac{1}{m}\sum_{k=1}^{\infty}\frac{m-k}{s+k} - \frac{1}{m}\sum_{k=1}^{\infty}\frac{(-1)^k}{k}\left\{\binom{s+m}{k+1} - \binom{s}{k+1}\right\}$$

## Series with Gregory's Coefficients, Cauchy Numbers, and Bernoulli Polynomials of the Second Kind

1. Using Rational Coefficients for Rational Series: There exist various series for the digamma containing only rational coefficients for rational arguments.
2. Digamma Function using Gregory's Coefficients: In particular, the series version of the Gregory's coefficient $G_n$ is

$$\psi(v) = \log v - \sum_{n=1}^{\infty}\frac{|G_n|(n-1)!}{(v)_n}$$

$$\Re(v) > 0$$

$$\psi(v) = 2\log\Gamma(v) - 2v\log v + 2v + 2\log v - \log 2\pi - \sum_{n=1}^{\infty}\frac{|G_n(2)|(n-1)!}{(v)_n}$$

$$\Re(v) > 0$$

$$\psi(v) = 3\log\Gamma(v) - 6\zeta'(-1,v) + 3v^2\log v - \frac{3}{2}v^2 + 6v\log v + 3v + 3\log v$$
$$+\frac{3}{2}\log 2\pi + \frac{1}{2} - 3\sum_{n=1}^{\infty}\frac{|G_n(3)|(n-1)!}{(v)_n}$$

$$\Re(v) > 0$$

where $(v)_n$ is the *rising factorial*

$$(v)_n = v(v+1)\cdots(v+n-k)$$

$G_n(k)$ are the Gregory coefficients of higher order with

$$G_n(1) = G_n$$

Γ is the gamma function, and ζ is the Hurwitz zeta function (Blagouchine (2016, 2018)).

3. Using Cauchy's Numbers of the Second Kind: Similar series with the Cauchy numbers of the second kind $C_n$ reads (Blagouchine (2016, 2018))

$$\psi(v) = \log(v-1) - \sum_{n=1}^{\infty} \frac{C_n(n-1)!}{(v)_n}$$

$$\Re(v) > 1$$

4. Digamma Function using Bernoulli Polynomials: A series with the Bernoulli polynomials of the second kind has the following form (Blagouchine (2018)):

$$\psi(v) = \log(v+a) - \sum_{n=1}^{\infty} \frac{(-1)^n \varphi_n(a)(n-1)!}{(v)_n}$$

$$\Re(v) > -a$$

where $\varphi_n(a)$ are the *Bernoulli polynomials of the Second Kind* defined by the generating equation

$$\frac{z(1+z)^a}{\log(1+z)} = \sum_{n=0}^{\infty} z^n \varphi_n(a)$$

$$|z| < 1$$

5. Generalization of the above Series: It may be generalized to

$$\psi(v) = \frac{1}{r}\sum_{l=0}^{r-1} \log(v+a+l) + \frac{1}{r}\sum_{n=1}^{\infty} \frac{(-1)^n N_{n,r}(a)(n-1)!}{(v)_n}$$

$$\Re(v) > -a$$

$$r = 1, 2, 3, \cdots$$

where the polynomials $N_{n,r}(a)$ are given by the following generating equation

$$\frac{(1+z)^{a+m} - (1+z)^a}{\log(1+z)} = \sum_{n=0}^{\infty} z^n N_{n,m}(a)$$

$$|z| < 1$$

so that

$$N_{n,l}(a) = \varphi_n(a)$$

(Blagouchine (2018)).

6. Series for Log Gamma Function: Similar expressions for the logarithm of the gamma function produce the following (Blagouchine (2018)):

$$\psi(v)=\frac{1}{v+a-\frac{1}{2}}\left\{\log\Gamma(v+a)+v-\frac{1}{2}\log 2\pi-\frac{1}{2}-\sum_{n=1}^{\infty}\frac{(-1)^n\varphi_{n+1}(a)(n-1)!}{(v)_n}\right\}$$

$$\Re(v)>-a$$

and

$$\psi(v)=\frac{1}{v+a-1+\frac{1}{2}r}\left\{\log\Gamma(v+a)+v-\frac{1}{2}\log 2\pi-\frac{1}{2}\right.$$
$$\left.+\frac{1}{r}\sum_{n=0}^{r-2}(r-n-1)\log(v+a+n)+\frac{1}{r}\sum_{n=1}^{\infty}\frac{(-1)^n N_{n+1,r}(a)(n-1)!}{(v)_n}\right\}$$

$$\Re(v)>-a$$

## Reflection Formula

The digamma function satisfies a reflection formula similar to that of the gamma function

$$\psi(1-z)-\psi(z)=\pi\cot\pi z$$

## Recurrence Formula and Characterization

1. Recurrence Relation for Digamma Function: The digamma function satisfies the recurrence relation

$$\psi(x+1) = \psi(x) + \frac{1}{x}$$

2. Digamma Function Telescoping for $\frac{1}{x}$: Thus, it can be said to *telescope* $\frac{1}{x}$, for one has

$$\Delta[\psi](x) = \frac{1}{x}$$

where $\Delta$ is the forward difference operator.

3. Partial Harmonic Sum Recurrence Relation: This satisfies the recurrence relation of a partial sum of the harmonic series, thus implying the formula

$$\psi_n = H_{n-1} - \gamma$$

4. Generalization of the Harmonic Sum: More generally, one has

$$\psi(1+z) = -\gamma + \sum_{k=1}^{\infty} \left(\frac{1}{k} - \frac{1}{z-k}\right)$$

for

$$\Re(z) > 0$$

5. Bernoulli Number Digamma Series Expansion: Another series expansion is

$$\psi(1+z) = \log(z) + \frac{1}{2z} - \sum_{j=1}^{\infty} \frac{B_{2j}}{2j\, z^{2j}}$$

where $B_{2j}$ are the Bernoulli numbers. This series diverges for all $z$ and is known as the *Stirling series*.

6. Uniqueness of the Telescoped Solution: Actually, $\psi$ is the only solution of the functional equation

$$F(x+1) = F(x) + \frac{1}{x}$$

that is monotonic on $\mathbb{R}^{+}$ and satisfies

$$F(1) = -\gamma$$

This follows immediately from the uniqueness of the gamma function, given its recurrence relation and the convexity restriction.

7. Digamma Function Difference Equation: This implies the useful difference equation

$$\psi(x+N) - \psi(x) = \sum_{k=0}^{N-1} \frac{1}{x+k}$$

## Some Finite Sums involving the Digamma Function

1. Digamma Function Gaussian Summation Formulas:

$$\sum_{r=0}^{m} \psi\left(\frac{r}{m}\right) = -m(\gamma + \log m)$$

$$\sum_{r=0}^{m} \psi\left(\frac{r}{m}\right) \cdot e^{\frac{2\pi i k r}{m}} = -m \log\left(1 - e^{\frac{2\pi i k}{m}}\right)$$

$$k \in \mathbb{Z}$$

$$m \in \mathbb{N}$$

$$k \neq m$$

$$\sum_{r=0}^{m} \psi\left(\frac{r}{m}\right) \cdot \cos\frac{2\pi k r}{m} = -m \log\left(2 \sin\frac{2\pi k}{m}\right) + \gamma$$

$$k = 1, 2, \cdots, m - 1$$

$$\sum_{r=0}^{m} \psi\left(\frac{r}{m}\right) \cdot \sin\frac{2\pi k r}{m} = \frac{\pi}{2}(2k - m)$$

$$k = 1, 2, \cdots, m - 1$$

are due to Gauss (Campbell (1966), Shrivastava and Choi (2001)).

2. Digamma Function Blagouchine Summation Formula: More complicated formulas, such as

$$\sum_{r=0}^{m-1} \psi\left(\frac{2r + 1}{m}\right) \cdot \cos\frac{(2r + 1)\pi k}{m} = m \log\left(\tan\frac{\pi k}{2m}\right)$$

$$k = 1, 2, \cdots, m-1$$

$$\sum_{r=0}^{m-1} \psi\left(\frac{2r+1}{m}\right) \cdot \sin\frac{(2r+1)\pi k}{m} = -\frac{\pi m}{2}$$

$$k = 1, 2, \cdots, m-1$$

$$\sum_{r=0}^{m-1} \psi\left(\frac{r}{m}\right) \cdot \cot\frac{\pi r}{m} = -\frac{\pi(m-1)(m-2)}{6}$$

$$\sum_{r=0}^{m-1} \psi\left(\frac{r}{m}\right) \cdot \frac{r}{m} = -\frac{\gamma}{2}(m-1) - \frac{m}{2}\log m - \frac{\pi}{2}\sum_{r=1}^{m-1} \frac{r}{m} \cdot \cot\frac{\pi r}{m}$$

$$\sum_{r=1}^{m-1} \psi\left(\frac{r}{m}\right) \cdot \cos\frac{(2l+1)\pi r}{m} = -\frac{\pi}{m}\sum_{r=1}^{m-1} \frac{r \cdot \sin\frac{2\pi r}{m}}{\cos\frac{2\pi r}{m} - \cos\frac{(2l+1)\pi}{m}}$$

$$l \in \mathbb{Z}$$

$$\begin{aligned}
&\sum_{r=1}^{m-1} \psi\left(\frac{r}{m}\right) \cdot \sin\frac{(2l+1)\pi r}{m} \\
&\qquad = -(\gamma + \log 2m)\cot\frac{(2l+1)\pi}{2m} \\
&\qquad + \sin\frac{(2l+1)\pi}{m}\sum_{r=1}^{m-1} \frac{\log\sin\frac{\pi r}{m}}{\cos\frac{2\pi r}{m} - \cos\frac{(2l+1)\pi}{m}}
\end{aligned}$$

$$l \in \mathbb{Z}$$

$$\sum_{r=1}^{m-1} \psi^2\left(\frac{r}{m}\right) = (m-1)\gamma^2 + m(2\gamma + \log 4m)\log m - m(m-1)\log^2 2$$

$$-\frac{\pi^2(m^2-3m+2)}{12} + m\sum_{l=1}^{m-1} \log^2 \sin\frac{\pi l}{m}$$

are due to the works of certain modern authors (e.g., Blagouchine (2015)).

## Gauss Digamma Theorem

For positive integers $r$ and $m$

$$r < m$$

the digamma function may be expressed in terms of Euler's constant and a finite number of elementary functions.

$$\psi\left(\frac{r}{m}\right) = -\gamma - \log 2m - \frac{\pi}{2}\cot\frac{\pi r}{m} + 2\sum_{n=1}^{\left\lfloor\frac{m-1}{2}\right\rfloor} \frac{r}{m}\cdot\cos\frac{2\pi nr}{m}\log\sin\frac{\pi n}{m}$$

## Asymptotic Expansion

1. Asymptotic Expansion using Bernoulli/Reimann Zeta: The digamma function has the asymptotic expansion

$$\psi(z) \to \log(z) - \frac{1}{2z} + \sum_{j=1}^{\infty} \frac{\zeta(1-2n)}{z^{2j}} = \log(z) - \frac{1}{2z} - \sum_{j=1}^{\infty} \frac{B_{2j}}{2j\, z^{2j}}$$

where $B_k$ is the $k^{th}$ Bernoulli number, and $\zeta$ is the Riemann zeta function.

2. Asymptotic Expansion of Series Terms: The first few terms of this expansion are:

$$\psi(z) \to \log(z) - \frac{1}{2z} - \frac{1}{12z^2} + \frac{1}{120z^4} - \frac{1}{252z^6} + \frac{1}{240z^8} - \frac{5}{660z^{10}} + \frac{691}{32760z^{12}} - \frac{1}{12z^{14}} + \cdots$$

3. Convergence of the above Series: Although the infinite sum does not converge for any $z$, any finite partial sum becomes increasingly accurate as $z$ increases.
4. Infinite Series using Euler MacLaurin Formula: The above series can be found by applying the Euler-MacLaurin formula to the sum (Bernardo (1976)) $\sum_{n=1}^{\infty}\left(\frac{1}{n} - \frac{1}{z+n}\right)$
5. Binet's Second Integral Series Expansion: The expansion can also be derived from the integral representation coming from Binet's second integral formula for the gamma function.
6. Application of Bernoulli's Numbers to Geometric Series: Expanding $\frac{t}{(z^2+t^2)}$ as a geometric series, and substituting the integral representation of the Bernoulli numbers leads to the same asymptotic series as above. Furthermore, expanding only finitely many terms of the series gives a formula with an explicit error term:

$$\psi(z) = \log(z) - \frac{1}{2z} - \sum_{j=1}^{\infty} \frac{B_{2j}}{2j\, z^{2j}} + (-1)^{N+1} \frac{2}{z^{2N}} \int_0^{\infty} \frac{t^{2N+1}}{(z^2+t^2)(e^{2\pi t}-1)} dt$$

## Inequalities

1. LTE Inequality for Digamma Function: When

$$x > 0$$

   the function $\log(z) - \frac{1}{2z} - \psi(z)$ is completely monotonic, and in particular, positive.

2. Consequence of Bernstein's Theorem: This is a consequence of Bernstein's theorem on monotone functions applied to the integral representation coming from Binet's first integral for the gamma function.
3. Upper Bound on the Integral: Additionally, by the convexity inequality

$$1 + t \leq e^t$$

   the integrand in this representation is bounded above by $\frac{e^{-tz}}{2}$.

4. GTE Inequality for Digamma Function: Consequently, $\frac{1}{z} - \log(z) + \psi(z)$ is also completely monotone.
5. Alzer Inequality for Digamma Differences: This recovers the theorem of Alzer (1997), who also showed that, for

$$s \in (0, 1)$$

$$\frac{1-s}{1+s} < \psi(z+1) - \psi(z+s)$$

6. Elezovic-Giordano-Pecaric Digamma Bounds: Related bounds were obtained by Elezovic, Giordano, and Pecaric (2000), who proved that, for

$$z > 0$$

$$\log\left(z + \frac{1}{2}\right) - \frac{1}{z} < \psi(z) < \log(z + e^{-\gamma}) - \frac{1}{z}$$

where $\gamma$ is the Euler-Mascheroni constant. The constants appearing in these bounds are the best possible (Qi and Guo (2009)).

7. Gautschi Inequality for Digamma Ratio: The mean-value theorem implies the following analog of Gautschi's inequality. If

$$z > c$$

where

$$c \approx 1.461$$

is the unique positive real root of the digamma function, and if

$$s > 0$$

then

$$e^{(1-s)\frac{\psi'(z+1)}{\psi(z+1)}} \leq \frac{\psi(z+1)}{\psi(z+s)} \leq e^{(1-s)\frac{\psi'(z+s)}{\psi(z+s)}}$$

Moreover, the equality holds if and only if

$$s = 1$$

(Laforgia and Natalini (2013)).

8. Digamma Harmonic Mean Value Inequality: Inspired by the harmonic mean value inequality for the classical gamma function, Alzer and Jameson (2017) showed, among other things, a harmonic mean value inequality for the digamma function:

$$-\gamma \leq \frac{2\psi(z)\psi\left(\frac{1}{z}\right)}{\psi(z) + \psi\left(\frac{1}{z}\right)}$$

for

$$z \geq 0$$

The inequality holds if and only if

$$z = 1$$

## Computation and Approximation

1. Asymptotic Expansion using Recurrence Relation: The asymptotic expansion gives an easy way to compute $\psi(z)$ when the real part of $z$ is large. To compute $\psi(z)$ for small $z$, the recurrence relation

$$\psi(z) + 1 = \psi(z) + \frac{1}{z}$$

can be used to shift the value of $z$ higher.

2. Usage for Higher $z$ Values: Beal (2003) suggests using the above recurrence to shift to a value greater than $6$ and then applying the above expansion with terms over the $z^{14}$ cutoff, which yields *more than enough precision* – at least $12$ digits except near the zeros.
3. Bounds on $\psi(z)$ as $z \rightarrow \infty$: As $z$ goes to infinity, $\psi(z)$ gets arbitrarily close to both $\log\left(z - \frac{1}{2}\right)$ and $\log z$. Going down from $z + 1$ to $z$, $\psi(z)$ decreases by $\frac{1}{z}$, $\log\left(z - \frac{1}{2}\right)$ decreases by $\frac{\log\left(z + \frac{1}{2}\right)}{z - \frac{1}{2}}$, which is more than $\frac{1}{z}$, and $\log z$ decreases by $\log\left(1 + \frac{1}{z}\right)$, which is less than $\frac{1}{z}$.
4. $\psi(z)$ Bounds for $z > \frac{1}{2}$: From this, it can be seen that for any positive $z$ greater than $\frac{1}{2}$,

$$\psi(z) \in \left(\log\left(z - \frac{1}{2}\right), \log z\right)$$

   or, for any positive $z$,

$$e^{\psi(z)} \in \left(z - \frac{1}{2}, z\right)$$

5. $e^{\psi(z)}$ as $z \rightarrow \infty$ and $z \rightarrow 0$: The exponential $e^{\psi(z)}$ is approximately $z - \frac{1}{2}$ for large $z$, but gets closer to $z$ at small $z$, approaching $0$ at

$$z = 0$$

6. $\psi(z)$ Bounds for $z \in (0, 1)$: For $z < 1$ the limits can be calculated based on the fact that, between $1$ and $2$, as

$$\psi(z) \in (-\gamma, 1 - \gamma)$$

so

$$\psi(z) \in \left(-\frac{1}{z} - \gamma, 1 - \frac{1}{z} - \gamma\right)$$

$$z \in (0, 1)$$

or

$$e^{\psi(z)} \in \left(e^{-\frac{1}{z}-\gamma}, e \cdot e^{-\frac{1}{z}-\gamma}\right)$$

7. Asymptotic Series Expansion for $e^{\psi(z)}$: From the above asymptotic series for $\psi(z)$, one can derive an asymptotic series for $e^{-\psi(z)}$. The series matches the overall behavior well, that is, it behaves asymptotically as it should for large arguments, and has poles of unbounded multiplicity at the origin.

$$\frac{1}{e^{\psi(z)}} \sim \frac{1}{x} + \frac{1}{2 \cdot x^2} + \frac{5}{4 \cdot 3! \cdot x^3} + \frac{3}{2 \cdot 4! \cdot x^4} + \frac{47}{48 \cdot 5! \cdot x^5} - \frac{5}{16 \cdot 6! \cdot x^6} + \cdots$$

8. Convergence Behavior of the Above Series: This is similar to the Taylor expansion of $e^{-\psi\left(\frac{1}{y}\right)}$ at

$$y = 0$$

but it does not converge. However, if it converged to a function $f(y)$, then $\log\frac{f(y)}{y}$ would have the same MacLaurin series as $\log\frac{1}{y} - \psi\left(\frac{1}{y}\right)$ But this does not converge because the series given above does not converge – the function is not analytic at infinity. A similar series exists for $e^{\psi(z)}$, which starts at

$$e^{\psi(z)} \sim z - \frac{1}{2}$$

9. Asymptotic Series for $\psi\left(z+\frac{1}{2}\right)$ and $e^{\psi\left(z+\frac{1}{2}\right)}$: If one calculates the asymptotic series for $\psi\left(z+\frac{1}{2}\right)$ it turns out there are no odd powers of $z$, i.e., there is no $z^{-1}$ term. This leads to the following asymptotic expansion, which saves computing terms of even order:

$$e^{\psi\left(z+\frac{1}{2}\right)} \sim x + \frac{1}{4! \cdot x} - \frac{37}{8 \cdot 6! \cdot x^3} + \frac{10313}{72 \cdot 8! \cdot x^5} - \frac{5509121}{384 \cdot 10! \cdot x^7} + \cdots$$

## Special Values

1. Special Values from Gauss Theorem: The digamma function has values in closed form for rational numbers, as a result of Gauss' theorem. Some are listed below:

$$\psi(1) = -\gamma$$

$$\psi\left(\frac{1}{2}\right) = -2\ln 2 - \gamma$$

$$\psi\left(\frac{1}{3}\right) = -\frac{\pi}{2\sqrt{3}} - \frac{3\ln 3}{2} - \gamma$$

$$\psi\left(\frac{1}{4}\right) = -\frac{\pi}{2} - 3\ln 2 - \gamma$$

$$\psi\left(\frac{1}{6}\right) = -\frac{\pi\sqrt{3}}{2} - 2\ln 2 - \frac{3\ln 3}{2} - \gamma$$

$$\psi\left(\frac{1}{8}\right) = -\frac{\pi}{2} - 4\ln 2 - \frac{\pi + \log\left(2+\sqrt{2}\right) - \log\left(2-\sqrt{2}\right)}{\sqrt{2}} - \gamma$$

2. Values at Imaginary Unit: Moreover, by the series representation, it can be easily deduced that at the imaginary unit,

$$\Re\left(\psi(i)\right) = -\gamma - \sum_{n=0}^{\infty} \frac{n-1}{n^3+n^2+n+1}$$

$$\Im\left(\psi(i)\right) = \sum_{n=0}^{\infty} \frac{1}{n^2+1} = \frac{1}{2} + \frac{\pi}{2}\coth\pi$$

## Regularization

The digamma function appears in the regularization of divergent integrals $\int_0^\infty \frac{dx}{x+a}$, this integral can be approximated by a divergent general Harmonic series, but the following value can be attached to the series

$$-\psi(a) = \sum_{n=0}^{\infty} \frac{1}{n+a}$$

## Roots of the Digamma Function

1. Complex Gamma Function Saddle Point: The roots of the digamma function are the saddle points of the complex-valued gamma function. Thus, they all lie on the real axis.
2. Positive and Negative Digamma Roots: The only one on the positive real axis is the unique minimum of the real-valued gamma function at $\mathbb{R}^+$ at

$$x_0 = 1.461\ 632\ 144\ 968\ \cdots$$

All others occur between the single poles on the negative axis.

$$x_1 = -0.504\ 083\ 008\ \cdots$$

$$x_2 = -1.573\ 498\ 473\ \cdots$$

$$x_3 = -2.610\ 720\ 868\ \cdots$$

$$x_4 = -3.635\ 293\ 366\ \cdots$$

3. Hermite Approximation of Digamma Roots: Charles Hermite (1881) observed that

$$x_n = -n + \frac{1}{\ln n} + \mathcal{O}\left(\frac{1}{\ln^2 n}\right)$$

holds asymptotically.

4. Analytic Improvement over Hermite Roots: A better approximation of the location of the roots is given by

$$x_n \approx -n + \frac{1}{\ln n} + \frac{1}{\pi}\arctan\left(\frac{\pi}{\ln n}\right)$$

$$n \geq 2$$

and using a further term it becomes still better

$$x_n \approx -n + \frac{1}{\ln n} + \frac{1}{\pi}\arctan\left(\frac{\pi}{\ln n + \frac{1}{8n}}\right)$$

$$n \geq 1$$

which both spring off of the reflection formula via

$$0 = \psi(1 - x_n) = \psi(x_n) + \frac{\pi}{\tan \pi x_n}$$

and substituting $\psi(x_n)$ by its most convergent asymptotic expansion.

5. Adjusted Digamma Expansion for Roots: The correct second term of this expansion is $\frac{1}{2n}$, where the given one works good to approximate roots with small $n$.
6. Enhancement to the Hermite Roots: Another improvement of the Hermite's formula was given by Mezo and Hoffman (2017):

$$x_n = -n + \frac{1}{\ln n} - \frac{1}{2n\ \ln^2 n} + \mathcal{O}\left(\frac{1}{n^2 \ln^2 n}\right)$$

7. Digamma Roots - Infinite Sum Identities: Regarding the zeros, the following infinite sum identities were recently proved by Mezo and Hoffman (2017):

$$\sum_{n=0}^{\infty} \frac{1}{{x_n}^2} = \gamma^2 + \frac{\pi^2}{2}$$

$$\sum_{n=0}^{\infty} \frac{1}{{x_n}^3} = -4\zeta(3) - \gamma^3 - \frac{\pi^2 \gamma}{2}$$

$$\sum_{n=0}^{\infty} \frac{1}{{x_n}^4} = \gamma^4 + \frac{\pi^4}{9} + \frac{2\pi^2\gamma^2}{3} + 4\gamma\zeta(3)$$

8. Digamma Roots - Higher Order Terms: In general, the function

$$\varpi(k) = \sum_{n=0}^{\infty} \frac{1}{{x_n}^k}$$

can be determined, and it has been studied in detail by Mezo and Hoffman (2017).

9. Reciprocal Quadrature Root Polynomial Sums: Further, Mezo and Hoffman (2017) show that

$$\sum_{n=0}^{\infty} \frac{1}{{x_n}^2 + x_n} = -2$$

and

$$\sum_{n=0}^{\infty} \frac{1}{{x_n}^2 + x_n} = \gamma + \frac{\pi^2}{6\gamma}$$

also holds true.

## References

- Abramowitz, M., and I. A. Stegun (2007): *Handbook of Mathematics Functions* **Dover Book on Mathematics**
- Alzer, H. (1997): On some Inequalities for the $\Gamma$ and the $\psi$ Functions *Mathematics of Computation* **66 (217)** 373-389
- Alzer, H., and G. Jameson (2017): A Harmonic Mean Inequality for the Digamma Function and Related Results *Rendiconti del Seminaro Matematico della Universita di Padova* **70 (201)** 203-209
- Bernardo, J. M. (1976): Algorithm AS03 $\psi$ (Digamma Function) Computation *Applied Statistics* **25** 315-317

- Blagouchine, I. V. (2015): A Theorem for the Closed-Form Evaluation of the First Generalized Stieltjes Constant at Rational Arguments and some Related Summations **arXiv**
- Blagouchine, I. V. (2016): Two Series Expansions for the Logarithm of the Gamma Function involving Stirling Numbers and containing only Rational Coefficients for certain argument related $\pi^{-1}$ **arXiv**
- Blagouchine, I. V. (2018): Three Notes on Ser's and Hasse's Representations for the Zeta-Functions **arXiv**
- Campbell, R. (1966): *Les Integrales Euleriennes et Leurs Applications* **Dunod** Paris
- Elezovic, N., C. Giordano, J. Pecaric (2000): The Best Bounds in Gautschi's Inequality *Mathematical Inequalities and Applications* **3 (2)** 239-252
- Hermite, C. (1881): Sur l'integrale Eulerienne de Seconde Espace *Journal fur die Reine und Angewandte Mathematik* **90** 332-338
- Laforgia, A., and P. Natalini (2013): Exponential, Gamma, and Polygamma Functions: Simple Proofs of Classical and New Inequalities *Journal of Mathematical Analysis and Applications* **407 (2)** 495-504
- Mezo, I., and M. E. Hoffman (2017): Zeros of the Digamma Function and its Barnes G-function Analogue *Integral Transforms and Special Functions* **28 (28)** 846-858
- Norlund, N. E. (1924): *Vorlesungen uber Differenzenrechnung* **Springer** Berlin
- Qi, F., and B. N. Guo (2009): Sharp inequalities for the psi function and harmonic numbers **arXiv**
- Shrivastava, H. M., and J. Choi (2001): *Series Associated with the Zeta and the Related Functions* **Kluwer Academic Publishers** Netherlands
- Whittaker, E. T., and G. N. Watson (1996): *A Course on Modern Analysis 4th Edition* **Cambridge University Press**
- Wikipedia (2019): Digamma Function

# Error Function

## Introduction and Overview

1. What is an Error Function? The **error function** – also called the **Gaussian Error Function** – is a special non-elementary function of sigmoidal shape that occurs in probability, statistics, and partial differential equations describing diffusion (Wikipedia (2019)).
2. Definition of the Error Function: It is defined as (Greene (1993), Andrews (1998))

$$\mathrm{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt$$

3. Interpretation of the Error Function: In statistics, for non-negative values of $x$, the error function has the following interpretation: for a random variable $Y$ that is normally distributed with a mean $0$ and variance $0.5$, describes the probability of $Y$ falling in the range $[-x, x]$.
4. Function Related to erf: There are several closely related functions, such as the complementary error function, the imaginary error function, and others.

## Name

1. Origin of erf and erfc: The name *error function* and its abbreviation erf were proposed by J. W. L. Glaisher in 1871 on account of its connection with “the Theory of Probability, and notably the Theory of Errors” (Glaisher (1871a)). The error function complement was also discussed by Glaisher in a separate publication in the same year (Glaisher (1871b)).
2. Error Density - *Law of Facility*: For the *Law of Facility* of errors whose density is given by a normal distribution

$$f(x) = \sqrt{\frac{c}{\pi}} e^{-cx^2}$$

   Glaisher calculates the chance of an error lying between $p$ and $q$ as

$$\sqrt{\frac{c}{\pi}} p \int_{p}^{q} e^{-cx^2} dx = \frac{1}{2}\left[\operatorname{erf}\left(q\sqrt{c}\right) - \operatorname{erf}\left(p\sqrt{c}\right)\right]$$

## Applications

1. Error Statistics Distribution from Measurements: When the results of a series of experiments is described by a normal distribution with standard deviation $\sigma$ and expected value 0, then erf$\left(\frac{a}{\sigma\sqrt{2}}\right)$ is the probability that the error of a single measurement lies between $-a$ and $+a$ for a positive $a$. This is useful, for example, in determining the bit error-rate if a digital communication system.

2. Heaviside Step Function Boundary Conditions: The error and the complementary error functions occur, for example, in solutions to the heat transfer equation when the boundary conditions are given by the Heaviside step function.
3. Cumulative Probability such that $X < L$: The error function and its approximations can be used to estimate results that hold with high probability. Given a variable

$$X \sim \mathcal{N}(\mu, \sigma)$$

   and constant

$$L < \mu$$

$$\mathbb{P}[X \leq L] = \frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{L - \mu}{\sqrt{2}\sigma}\right) \approx Ae^{-B\left(\frac{L-\mu}{\sigma}\right)^2}$$

   where $A$ and $B$ are certain numeric constants.
4. Asymptote of the Cumulative Probability: If $L$ is sufficiently far from the mean, i.e., if

$$\mu - L \geq \sigma\sqrt{\ln k}$$

   then

$$\mathbb{P}[X \leq L] \leq Ae^{-B\ln k} = \frac{A}{k^B}$$

   so the probability goes to 0 as

$$k \to \infty$$

## Properties

1. erf is an Odd Function: The property

$$\operatorname{erf}(-z) = -\operatorname{erf}(z)$$

means that the error function is an odd function. This directly comes about from the fact that the integrand $e^{-t^2}$ is an even function.

2. erf is Self-Complex Conjugate: For any complex number $z$

$$\operatorname{erf}(\bar{z}) = \overline{\operatorname{erf}(z)}$$

where $\bar{z}$ is the complex conjugate of $z$.

3. erf Asymptotic Behavior at $\pm\infty$: The error function at $+\infty$ is exactly $1$. At the real axis, $\operatorname{erf}(z)$ approaches unity at

$$z \rightarrow +\infty$$

and $-1$ at

$$z \rightarrow -\infty$$

At the imaginary axis, it tends to $\pm i\infty$

## Taylor Series

1. erf Holomorphic Everywhere Except at ±∞: The error function is an entire function; it has no singularities except those at infinity, and its Taylor expansion always converges.
2. MacLaurin Polynomial Series for erf: The defining integral cannot be evaluated in closed form in terms of elementary functions, but by expanding the integrand $e^{-x^2}$ into its MacLaurin series and integrating it term by term, one obtains the error function's MacLaurin series as:

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n z^{n+1}}{n!\,(2n+1)} = \frac{2}{\sqrt{\pi}} \left( z - \frac{z^3}{3} + \frac{z^5}{10} - \frac{z^7}{42} + \frac{z^9}{216} - \cdots \right)$$

   which holds for every complex number $z$.
3. Alternative Formulation of the MacLaurin Series: For iterative calculation of the above series, the following alternative formulation may be useful.

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \left[ z \prod_{k=1}^{n} \frac{-(2k-1)z^2}{k(2k+1)} \right] = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \left[ \frac{z}{2n+1} \prod_{k=1}^{n} \frac{-z^2}{k} \right]$$

   because $\frac{-(2k-1)z^2}{k(2k+1)}$ expresses the multiplier to turn the $k^{th}$ term into the $(k+1)^{th}$ term, considering $z$ as the first term.
4. Imaginary Error Function MacLaurin Series: The imaginary error function has a very similar MacLaurin series, which is

$$\operatorname{erfi}(z) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n z^{n+1}}{n!\,(2n+1)} = \frac{2}{\sqrt{\pi}} \left( z + \frac{z^3}{3} + \frac{z^5}{10} + \frac{z^7}{42} + \frac{z^9}{216} + \cdots \right)$$

   which holds for every complex number $z$.

## Derivative and Integral

1. Derivative of the Error Function: The derivative of the error function follows immediately from its definition:

$$\frac{d}{dz}\operatorname{erf}(z)=\frac{2}{\sqrt{\pi}}e^{-z^2}$$

2. Derivative of the Imaginary Error Function: The derivative of the imaginary error function follows immediately from its definition:

$$\frac{d}{dz}\operatorname{erfi}(z)=\frac{2}{\sqrt{\pi}}e^{z^2}$$

3. Anti-derivative of the Error Function: An anti-derivative of the error function, obtainable by integration by parts, is $z\operatorname{erf}(z)+\frac{e^{-z^2}}{\sqrt{\pi}}$
4. Anti-derivative of the Imaginary Error Function: An anti-derivative of the error function, obtainable by integration by parts, is $z\operatorname{erfi}(z)-\frac{e^{z^2}}{\sqrt{\pi}}$
5. Higher Order Error Function Derivatives: Higher order derivatives are given by

$$\operatorname{erf}_k(z)=\frac{2}{\sqrt{\pi}}\frac{d^{k-1}}{dz^{k-1}}e^{-z^2}=\frac{2(-1)^{k-1}}{\sqrt{\pi}}H_{k-1}(z)e^{-z^2}$$

where

$$k=1,2,\cdots$$

and $H$ are the physicists' Hermite polynomials.

## Burmann Series

1. The Hans Heinrich Burmann Theorem: An expansion, which converges more rapidly for all real values of $x$ than the Taylor expansion, is obtained by using the Hans Heinrich Burmann's theorem (Schopf and Supancic (2014)):

$$\begin{aligned}\operatorname{erf}(x) &= \frac{2}{\sqrt{\pi}} sgn(x)\sqrt{1-e^{-x^2}}\left[1-\frac{1}{12}\left(1-e^{-x^2}\right)-\frac{7}{480}\left(1-e^{-x^2}\right)^2\right. \\ &\quad \left.-\frac{5}{896}\left(1-e^{-x^2}\right)^3-\frac{787}{276480}\left(1-e^{-x^2}\right)^4-\cdots\right] \\ &= \frac{2}{\sqrt{\pi}} sgn(x)\sqrt{1-e^{-x^2}}\left(\frac{\sqrt{\pi}}{2}+\sum_{k=1}^{\infty} c_k e^{-kx^2}\right)\end{aligned}$$

2. Approximation using the Two Leading Terms: By keeping only the first two coefficients and choosing

$$c_1 = \frac{31}{200}$$

and

$$c_2 = -\frac{341}{8000}$$

the resulting approximation shows its largest relative error at

$$x = \pm 1.3796$$

where it is less than $3.6127 \times 10^{-3}$:

$$\operatorname{erf}(x) \approx \frac{2}{\sqrt{\pi}} sgn(x) \sqrt{1 - e^{-x^2}} \left( \frac{\sqrt{\pi}}{2} + \frac{31}{200} e^{-x^2} - \frac{341}{8000} e^{-2x^2} \right)$$

## Inverse Functions

1. <u>Non-Unique Imaginary-Valued Solutions</u>: Given a complex number $z$, there is not a *unique* complex number $w$ satisfying

   $$\operatorname{erf}(w) = z$$

   so a true inverse function would be multi-valued.
2. <u>Unique Real-Valued Inverse Solutions</u>: However, for

   $$-1 < x < 1$$

   there is a unique *real* number denoted $\operatorname{erf}^{-1}(x)$ satisfying

   $$\operatorname{erf}(\operatorname{erf}^{-1}(x)) = x$$

3. <u>Domain of the Inverse Error Function</u>: The **inverse error function** is usually defined in the domain $(-1, 1)$, and it is restricted to this domain in many computer algebra systems.

4. <u>Complex Domain of Error Functions</u>: However, it can be extended to the disk

$$|z| < 1$$

of the complex plane, using the MacLaurin series

$$\text{erf}^{-1}(z) = \sum_{k=0}^{\infty} \frac{c_k}{2k+1} \left(\frac{\sqrt{\pi}}{2} z\right)^{2k+1}$$

where

$$c_0 = 1$$

and

$$c_k = \sum_{m=0}^{k-1} \frac{c_m c_{k-1-m}}{(m+1)2(m+1)} = \left\{1, 1, \frac{7}{6}, \frac{127}{90}, \frac{4369}{2520}, \frac{34807}{16200}, \cdots\right\}$$

5. <u>Polynomial Series Expansion of $\text{erf}^{-1}(z)$</u>: Thus, one has the following series expansion. Note that the common factors have been canceled from the numerator and the denominator:

$$\text{erf}^{-1}(z) = \frac{\sqrt{\pi}}{2} \left(z + \frac{\pi}{12} z^3 + \frac{7\pi^2}{480} z^5 + \frac{127\pi^3}{40320} z^7 + \frac{4369\pi^4}{5806080} z^9 + \frac{34807\pi^5}{182476800} z^{11} + \cdots\right)$$

Note that the error function’s value at $\pm\infty$ is equal to $\pm 1$.

6. <u>Caveat on Recovering the $z$</u>: For

$$|z| < 1$$

one has

$$\mathrm{erf}(\mathrm{erf}^{-1}(z)) = z$$

7. The Inverse Complementary Error Function: The **inverse complementary error function** is defined as

$$\mathrm{erfc}^{-1}(1 - z) = \mathrm{erf}^{-1}(z)$$

8. The Inverse Imaginary Error Function: For *real* $x$, there is a unique *real* number $\mathrm{erfi}^{-1}(x)$ satisfying

$$\mathrm{erfi}(\mathrm{erfi}^{-1}(x)) = x$$

The **inverse imaginary error function** is defined as $\mathrm{erfi}^{-1}(x)$ (Bergsma (2006)).

9. MacLaurin Series Expansion for $x$: For any real $x$, Newton's method can be used to compute $\mathrm{erfi}^{-1}(x)$, and for

$$-1 \le x \le 1$$

the following MacLaurin series converges:

$$\mathrm{erfi}^{-1}(x) = \sum_{k=0}^{\infty} \frac{(-1)^k c_k}{2k + 1} \left(\frac{\sqrt{\pi}}{2} x\right)^{2k+1}$$

where $c_k$ is defined as above.

## Asymptotic Expansion

1. Asymptotic Expansion for erfc$(x)$: A useful asymptotic expansion of the complementary error function – and therefore also of the error function – for large real $x$ is

$$\operatorname{erfc}(x) = \frac{e^{-x^2}}{x\sqrt{\pi}}\left[1 + \sum_{n=1}^{\infty}(-1)^n \frac{1 \cdot 3 \cdot 5 \cdots (2n-1)}{(2x^2)^n}\right] = \frac{e^{-x^2}}{x\sqrt{\pi}}\left[\sum_{n=1}^{\infty}(-1)^n \frac{(2n-1)!!}{(2x^2)^n}\right]$$

   where $(2n-1)!!$ is the double factorial of $(2n-1)$, which is the product of all odd numbers up to $(2n-1)$.

2. Finite erfc Remainder Error in the Landau Notation: This series diverges for every finite $x$, and its meaning as asymptotic expansion is that, for any

$$N \in \mathbb{N}$$

   on has

$$\operatorname{erfc}(x) = \frac{e^{-x^2}}{x\sqrt{\pi}}\left[\sum_{n=1}^{\infty}(-1)^n \frac{(2n-1)!!}{(2x^2)^n}\right] + R_N(x)$$

   where the remainder, in Landau notation, is

$$R_N(x) = \mathcal{O}\left(x^{1-2N}e^{-x^2}\right)$$

   as

$$x \to \infty$$

3. Exact Value of the Error Remainder: Indeed, the exact value of the remainder is

$$R_N(x) \doteqdot \frac{(-1)^N}{\sqrt{\pi}} 2^{1-2N} \frac{(2N)!}{N!} \int_x^{\infty} t^{-2N} e^{-t^2} dt$$

   which follows easily by induction, writing

$$e^{-t^2} = -(2t)^{-1}\left(e^{-t^2}\right)'$$

   and integrating by parts.

4. Approximation for Small/Large $x$: For large enough values of $x$, only the first few terms of this asymptotic expansion are needed to obtain a good approximate of $\text{erfc}(x)$ while for not too large values for $x$, it can be noted that the above Taylor expansion at 0 provides a very fast convergence.

## Continued Fraction Expansion

A continued fraction expansion of the complementary error function is (Cuyt, Petersen, Vigdis, Verdonk, Waadeland, and Jones (2008)):

$$\text{erfc}(x) = \frac{z}{\sqrt{\pi}} e^{-x^2} \cfrac{1}{z^2 + \cfrac{a_1}{1 + \cfrac{a_2}{z^2 + \cfrac{a_3}{1 + \cdots}}}}$$

$$a_m = \frac{m}{2}$$

## Integral of Error Function with Gaussian Density Function

$$\int_{-\infty}^{\infty} \operatorname{erf}(ax+b)\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}dx = \operatorname{erf}\left(\frac{a\mu+b}{\sqrt{1+2a^2\sigma^2}}\right)$$

$$a, b, \mu, \sigma \in \mathbb{R}$$

## Factorial Series

1. The erfc Inverse Factorial Series: The inverse factorial series

$$\operatorname{erfc}(z) = \frac{e^{-z^2}}{z\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n Q_n}{(z^2+1)^{\bar{n}}} = \frac{e^{-z^2}}{z\sqrt{\pi}}\left[1-\frac{1}{2}\frac{1}{(z^2+1)}+\frac{1}{2}\frac{1}{(z^2+1)(z^2+2)}+\cdots\right]$$

converges for

$$Re(z^2) > 0$$

2. Definition of Rising Factorial Components:

$$Q_n \triangleq \frac{1}{\Gamma\left(\frac{1}{2}\right)} \int_0^\infty \tau(\tau-1)\cdots(\tau-n+1)\frac{e^{-\tau}}{\sqrt{\tau}}\,d\tau = \sum_{k=0}^{n} \left(\frac{1}{2}\right)^{\bar{k}} s(n,k)$$

$z^{\bar{n}}$ denotes the rising factorial, and $s(n,k)$ denotes a signed Stirling number of the first kind (Schlomilch (1859), Nielson (1906)).

## Numerical Approximations – Approximation with Elementary Functions

1. Abramowitz and Stegun Family of Functions: Abramowitz and Stegun (2007) provide several approximations of varying accuracy. This allows one to choose the fastest approximation suitable for a given application. In order of increasing accuracy, they are:

   a.

$$\operatorname{erf}(x) \approx 1 - \frac{1}{(1 + a_1x + a_2x^2 + a_3x^3 + a_4x^4)^4}$$

$$x \geq 0$$

with maximum error of $5 \times 10^{-4}$ where

$$a_1 = 0.278393$$

$$a_2 = 0.230389$$

$$a_3 = 0.000972$$

$$a_4 = 0.078108$$

b.

$$\mathrm{erf}(x) \approx 1 - (1 + a_1 t + a_2 t^2 + a_3 t^3)e^{-x^2}$$

$$t = \frac{1}{1 + px}$$

$$x \geq 0$$

with maximum error of $2.5 \times 10^{-5}$ where

$$a_1 = 0.3480242$$

$$a_2 = -0.0958798$$

$$a_3 = 0.7478556$$

$$p = 0.47047$$

c.

$$\mathrm{erf}(x) \approx 1 - \frac{1}{(1 + a_1 x + a_2 x^2 + \cdots + a_6 x^6)^{16}}$$

$$x \geq 0$$

with maximum error of 3 × $10^{-7}$ where

$$a_1 = 0.0705230784$$

$$a_2 = 0.0422820123$$

$$a_3 = 0.0092705272$$

$$a_4 = 0.0001520143$$

$$a_5 = 0.0002765672$$

$$a_6 = 0.0000430638$$

d.

$$\operatorname{erf}(x) \approx 1 - (1 + a_1 t + a_2 t^2 + \cdots + a_5 t^5) e^{-x^2}$$

$$t = \frac{1}{1 + px}$$

$$x \geq 0$$

with maximum error of 1.5 × $10^{-7}$ where

$$a_1 = 0.254829592$$

$$a_2 = -0.284496736$$

$$a_3 = 1.421413741$$

$$a_4 = -1.453152027$$

$$a_5 = 1.061405429$$

$$p = 0.3275911$$

2. Usage for Positive Negative $x$: All of the approximations are valid for

$$x \geq 0$$

To use these approximations for negative $x$, use the fact that $\mathrm{erf}(x)$ is an odd function, so

$$\mathrm{erf}(x) = -\mathrm{erf}(-x)$$

3. Pure Exponential Approximation and Bounds: Exponential bounds and a pure exponential approximation for the complementary error function are given by Chiani, Dardari, and Simon (2003):

$$\mathrm{erfc}(x) \leq \frac{1}{2}e^{-2x^2} + \frac{1}{2}e^{-x^2} \leq e^{-x^2}$$

$$x > 0$$

$$\mathrm{erfc}(x) \approx \frac{1}{2}e^{-x^2} + \frac{1}{2}e^{-\frac{4}{3}x^2}$$

$$x > 0$$

4. Karagiannidis and Lioumpas erfc Approximation: A tight approximation of the complementary error function for

$$x \in [0, \infty)$$

is given by Karagiannidis and Lioumpas (2007) who showed that, for the appropriate choice of parameters $\{A, B\}$

$$\mathrm{erfc}(x) \approx e^{-x^2} \frac{1 - e^{-Ax}}{B\sqrt{\pi}x}$$

They determined that

$$\{A, B\} = \{1.98, 1.135\}$$

which gave a good approximation for all

$$x \geq 0$$

5. <u>Single Term Lower erfc Bound</u>: A single-term lower bound is from Chang, Cosman, and Milstein (2011):

$$\mathrm{erfc}(x) \geq \sqrt{\frac{2e}{\pi}} \frac{\sqrt{\beta - 1}}{\beta} e^{-\beta x^2}$$

$$x \geq 0$$

$$\beta > 1$$

where the parameter $\beta$ can be picked to minimize error on the desired interval of approximation.

6. <u>Winitzki Approximation for erf</u>: Another approximation is given by

$$\operatorname{erf}(x) \approx sgn(x)\sqrt{1 - e^{-x^2 \frac{\frac{4}{\pi}+ax^2}{1+ax^2}}}$$

where

$$a = \frac{8(\pi - 3)}{3\pi(4 - \pi)} \cong 0.140012$$

7. Performance of the Winitzki Approximation: This is designed to be very accurate in the neighborhood of $0$ and in the neighborhood of infinity, and the error is less than $0.00035$ for all $x$. Using the alternate value

$$a \approx 0.147$$

reduces the maximum error to about $0.00012$ (Winitzki (2008)).

8. Winitzki Approximation for $\operatorname{erf}^{-1}$: This approximation can also be inverted to calculate the inverse error function:

$$\operatorname{erf}^{-1}(x) \approx sgn(x)\sqrt{\sqrt{\left[\frac{2}{\pi a} + \frac{\ln(1 - x^2)}{2}\right]^2 - \frac{\ln(1 - x^2)}{a}} - \left[\frac{2}{\pi a} + \frac{\ln(1 - x^2)}{2}\right]}$$

## Polynomial

An approximation with a maximal error of $1.2 \times 10^{-7}$ for any real argument (Press, Teukolsky, Vetterling, and Flannery (2007))

$$\operatorname{erf}(x) = \begin{cases} 1 - \tau & x \geq 0 \\ \tau - 1 & x < 0 \end{cases}$$

with

$$\begin{aligned} \tau = t \cdot \exp\{-x^2 - 1.26551223 + 1.00002368t + 0.37409196t^2 + 0.09678418t^3 \\ - 0.18628806t^4 + 0.27886807t^5 - 1.13520398t^6 + 1.48851587t^7 \\ - 0.82215223t^8 + 0.17087277t^9\} \end{aligned}$$

and

$$t = \frac{1}{1 + 0.5|x|}$$

## Table of Values

| **x** | $\mathbf{erf}(\boldsymbol{x})$ | $\mathbf{1 - erf}(\boldsymbol{x})$ |
|---|---|---|
| 0.00 | 0 | 1 |

| | | |
|---|---|---|
| 0.02 | 0.022 564 575 | 0.977 435 425 |
| 0.04 | 0.045 111 106 | 0.954 888 894 |
| 0.06 | 0.067 621 594 | 0.932 378 406 |
| 0.08 | 0.090 078 126 | 0.909 921 874 |
| 0.10 | 0.112 462 916 | 0.887 537 084 |
| 0.20 | 0.222 702 589 | 0.777 297 411 |
| 0.30 | 0.328 626 759 | 0.671 343 241 |
| 0.40 | 0.428 392 355 | 0.571 607 645 |
| 0.50 | 0.520 499 878 | 0.479 500 122 |
| 0.60 | 0.603 856 091 | 0.396 143 909 |
| 0.70 | 0.677 801 194 | 0.322 198 806 |
| 0.80 | 0.742 100 965 | 0.257 899 035 |
| 0.90 | 0.796 908 212 | 0.203 091 788 |
| 1.00 | 0.842 700 793 | 0.157 299 207 |
| 1.10 | 0.880 205 070 | 0.119 794 930 |
| 1.20 | 0.910 313 978 | 0.089 686 022 |
| 1.30 | 0.934 007 945 | 0.065 992 055 |
| 1.40 | 0.952 285 120 | 0.047 714 880 |
| 1.50 | 0.966 105 146 | 0.033 894 854 |
| 1.60 | 0.976 348 383 | 0.023 651 617 |

| | | |
|---|---|---|
| 1.70 | 0.983 790 459 | 0.016 209 541 |
| 1.80 | 0.989 090 502 | 0.010 909 498 |
| 1.90 | 0.992 790 429 | 0.007 209 571 |
| 2.00 | 0.995 322 265 | 0.004 677 735 |
| 2.10 | 0.997 020 533 | 0.002 979 467 |
| 2.20 | 0.998 137 154 | 0.001 862 846 |
| 2.30 | 0.998 856 823 | 0.001 143 177 |
| 2.40 | 0.999 311 486 | 0.000 688 514 |
| 2.50 | 0.999 593 048 | 0.000 406 952 |
| 3.00 | 0.999 977 910 | 0.000 022 090 |
| 3.50 | 0.999 999 257 | 0.000 000 743 |

## Related Functions – Complementary Error Function

1. Scaled/Unscaled Complementary Error Function: The **complementary error function**, denoted erfc, is defined as

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-t^2} dt = e^{-x^2} \operatorname{erfcx}(x)$$

which also defines erfcx, the **scaled complementary error function** (Cody (1993)), which can be used instead of erfc to avoid arithmetic underflow (Cody (1993), Zaghloul (2007)).

2. Craig's Formula Version of erfc: Another form of $\text{erfc}(x)$ for non-negative $x$ is known as Craig's formula, after its discoverer (Craig (1991)):

$$\text{erfc}(x|x \geq 0) = \frac{2}{\pi}\int_{0}^{\frac{\pi}{2}} e^{-\frac{x^2}{\sin^2\theta}} d\theta$$

3. Benefits of using Craig's Formula: This expression is valid only for positive values of $x$, but it can be used in conjunction with

$$\text{erfc}(x) = 2 - \text{erfc}(-x)$$

to obtain $\text{erfc}(x)$ for negative values. This form is advantageous in that the imaginary range of integration is fixed and finite.

## Imaginary Error Function

1. Definition of the Imaginary Error Function: The **imaginary error function**, denoted erfi, is defined as

$$\text{erfi}(x) = -i \cdot \text{erf}(ix) = \frac{2}{\sqrt{\pi}}\int_{0}^{x} e^{t^2} dt = \frac{2}{\sqrt{\pi}} e^{x^2} D(x)$$

where $D(x)$ is the Dawson function, which can be used instead of erfi to avoid arithmetic overflow (Cody (1993)).

2. erfi when $x$ is Real: Despite the name *imaginary error function*, $\text{erfi}(x)$ is real when $x$ is real.
3. Faddeeva Complex Error Function Definition: When the error function is evaluated for arbitrary complex arguments $z$, the resulting complex error function is usually discussed in a scaled form as the Faddeeva function:

$$w(z) = e^{-z^2}\text{erfc}(-iz) = \text{erfcx}(-iz)$$

## Cumulative Distribution Function

1. Standard Normal Cumulative Distribution Function: The error function is essentially identical to the standard normal cumulative distribution function, denoted $\Phi$, also named $\text{norm}\ (x)$ by software languages, as they differ only by scaling and translation.
2. Relation between CDF and erf: Indeed

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int\limits_{-\infty}^{x} e^{-\frac{t^2}{2}} dt = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] = \frac{1}{2}\text{erf}\left(-\frac{x}{\sqrt{2}}\right)$$

or, re-arranging for erf and erfc:

$$\text{erf}(x) = 2 \cdot \Phi\left(x\sqrt{2}\right) - 1$$

$$\text{erfc}(x) = 2 \cdot \Phi\left(-x\sqrt{2}\right) = 2\left[1 - \Phi\left(x\sqrt{2}\right)\right]$$

3. Relation between erf and Q-function: Consequently, the error function is also closely related to the Q-function, which is the tail probability of the standard normal distribution. The Q-function can be expressed in terms of the error function as

$$Q(x) = \frac{1}{2}\left[1 - \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right] = \frac{1}{2}\operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right)$$

4. Relation between $\operatorname{erf}^{-1}$ and Probit: The inverse of $\Phi$ is known as the normal quantile function, or the probit function, and may be expressed in terms of the inverse error function as

$$Probit(p) = \Phi^{-1}(p) = \sqrt{2}\operatorname{erf}^{-1}(2p-1) = -\sqrt{2}\operatorname{erfc}^{-1}(2p)$$

5. Usage of CDF and erf: The standard normal CDF is used more often in probability and statistics, and the error function is used more often in other branches of mathematics.
6. Special Case of Mittag-Lefler Function: The error function is a special case of the Mittag-Lefler function, and can also be expressed as a confluent hypergeometric function (Kummer's function):

$$\operatorname{erf}(x) = \frac{2x}{\sqrt{\pi}}\mathcal{M}\left(\frac{1}{2}, \frac{3}{2}, -x^2\right)$$

This has a simple expression in terms of the Fresnel integral.

7. Relation between erf and Gamma Function: In terms of the regularized gamma function $P$ and the regularized gamma function $\gamma$

$$\operatorname{erf}(x) = sgn(x)\mathcal{P}\left(\frac{1}{2}, x^2\right) = \frac{sgn(x)}{\sqrt{\pi}}\gamma\left(\frac{1}{2}, x^2\right)$$

where $sgn(x)$ is the sign function.

## Generalized Error Functions

1. Expression for Generalized Error Function: Some authors discuss the more general functions:

$$E_n(x) = \frac{n!}{\sqrt{\pi}} \int_0^x e^{-t^n} dt = \frac{n!}{\sqrt{\pi}} \sum_{p=0}^{\infty} (-1)^p \frac{x^{np+1}}{(np+1)p!}$$

2. Special Cases of Generalized Error Functions:
    a. $E_0(x)$ is a straight line through the origin:

$$E_0(x) = \frac{x}{\sqrt{\pi}}$$

    b. $E_2(x)$ is the error function $\text{erf}(x)$
3. Similarity Among Odd/Even Error Exponents: After division by $n!$, all $E_n$ for odd $n$ look similar – but not identical – to each other. Similarly, all $E_n$ for even $n$ look similar – but not identical – to each other after a division by $n!$. All other generalized error functions look similar to each other on the positive $x$ side of the graph.
4. From Standard/Incomplete Gamma Function: These generalized functions can be equivalently expressed for

$$x > 0$$

using the gamma function and the incomplete gamma function:

$$E_n(x) = \frac{1}{\sqrt{\pi}} \Gamma(n) \left[ \Gamma\left(\frac{1}{n}\right) - \Gamma\left(\frac{1}{n}, x^n\right) \right]$$

$$x > 0$$

5. <u>erf from Incomplete Gamma Functions</u>: Therefore, the error function can be defined in terms of the incomplete Gamma function:

$$\text{erf}(x) = 1 - \frac{1}{\sqrt{\pi}} \Gamma\left(\frac{1}{2}, x^2\right)$$

## Iterated Integrals of the Complementary Error Function

1. <u>Iterated Integrals of erfc - Definition</u>: The iterated integrals of the complementary error function are defined by (Carslaw, H. S., and J. C. Jaeger (1959)):

$$i^n \text{erfc}(z) = \int_z^\infty i^{n-1} \text{erfc}\,(\zeta) d\zeta$$

$$i^0 \text{erfc}(z) = \text{erfc}(z)$$

$$i^1 \text{erfc}(z) = i\text{erfc}(z) = \frac{1}{\sqrt{\pi}} e^{-z^2} - z\text{erfc}(z)$$

$$i^2\mathrm{erfc}(z) = \frac{1}{4}[\mathrm{erfc}(z) - 2zi\mathrm{erfc}(z)]$$

2. <u>General Recurrence Formula for erfc</u>: The general recurrence formula is

$$2ni^n\mathrm{erfc}(z) = i^{n-2}\mathrm{erfc}(z) - 2zi^{n-1}\mathrm{erfc}(z)$$

3. <u>Power Series Representation for Iterated erfc</u>: These have the power series

$$i^n\mathrm{erfc}(z) = \sum_{j=0}^{\infty} \frac{(-z)^j}{2^{n-j}j!\,\Gamma\left(1+\frac{n-j}{2}\right)}$$

from which follow the symmetry properties

$$i^{2m}\mathrm{erfc}(-z) = -i^{2m}\mathrm{erfc}(z) + \sum_{q=0}^{m} \frac{z^{2q}}{2^{2(m-q)-1}(2q)!\,(m-q)!}$$

and

$$i^{2m+1}\mathrm{erfc}(-z) = i^{2m+1}\mathrm{erfc}(z) + \sum_{q=0}^{m} \frac{z^{2q+1}}{2^{2(m-q)-1}(2q+1)!\,(m-q)!}$$

## References

- Abramowitz, M., and I. A. Stegun (2007): *Handbook of Mathematics Functions* **Dover Book on Mathematics**

- Andrews, L. C. (1998): *Special Functions of Mathematics for Engineers* **SPIE Press**
- Bergsma, W. (2006): A New Correlation Coefficient, its Orthogonal Decomposition, and Associated Tests of Independence **arXiv**
- Chang, S. H., P. C. Cosman, L. B. Milstein (2011): Chernoff-Type Bounds for Gaussian Error Function *IEEE Transactions on Communications* **59 (11)** 2939-2944
- Chiani, M., D. Dardari, and M. K. Simon (2003): New Exponential Bounds and Approximations for the Computation of Error Probability in Fading Channels *IEEE Transactions on Wireless Communications* **2 (4)** 840-845
- Cody, W. J. (1991): Algorithm 715: SPECFUN – A Portable FORTRAN Package of Special Function Routines and Test Drivers *ACM Transactions on Mathematical Software* **19 (1)** 22-32
- Craig, J. W. (1991): A New, Simple, and Exact Result for Calculating the Probability of Error For Two-Dimensional Signal Constellations
- Glaisher, J. W. L. (1871a): On a Class of Definite Integrals – Part I *London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 4* **42 (277)** 294-302
- Glaisher, J. W. L. (1871b): On a Class of Definite Integrals – Part II *London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 4* **42 (279)** 421-436
- Greene, W. H. (1993): *Econometric Analysis 5th Edition* **Prentice-Hall**
- Karagiannidis, G. K., and A. S. Lioumpas (2007): An Improved Approximation for the Gaussian Q-function *IEEE Communications Letters* **11 (8)** 644-646
- Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (2007): *Numerical Recipes: The Art of Scientific Computing 3rd Edition* **Cambridge University Press** New York
- Schopf, H. M., and P. H. Supancic (2014): On Burmann's Theorem and its Application to Problems of Linear and Non-linear Heat Transfer and Diffusion
- Schlomilch, O. X. (1859): Ueber Facultatenreihen *Zeitschrift fur Mathematik und Physik* **4** 390-415

- Nielson, N. (1906): *Handbuch der Theorie der Gammafunktion* **B. G. Teubner** Leipzig
- Wikipedia (2019): Error Function
- Winitzki, S. (2008): A Handy Approximation for the Error Function and its Inverse
- Zaghloul, M. R. (2007): On the Calculation of the Voigt Line Profile: A Single Proper Integral with a Damped Sign Integrand *Monthly Notices of the Royal Astronomical Society* **375 (3)** 1043-1048

**Figure #1**
**Fixed Point Search SKU Flow**

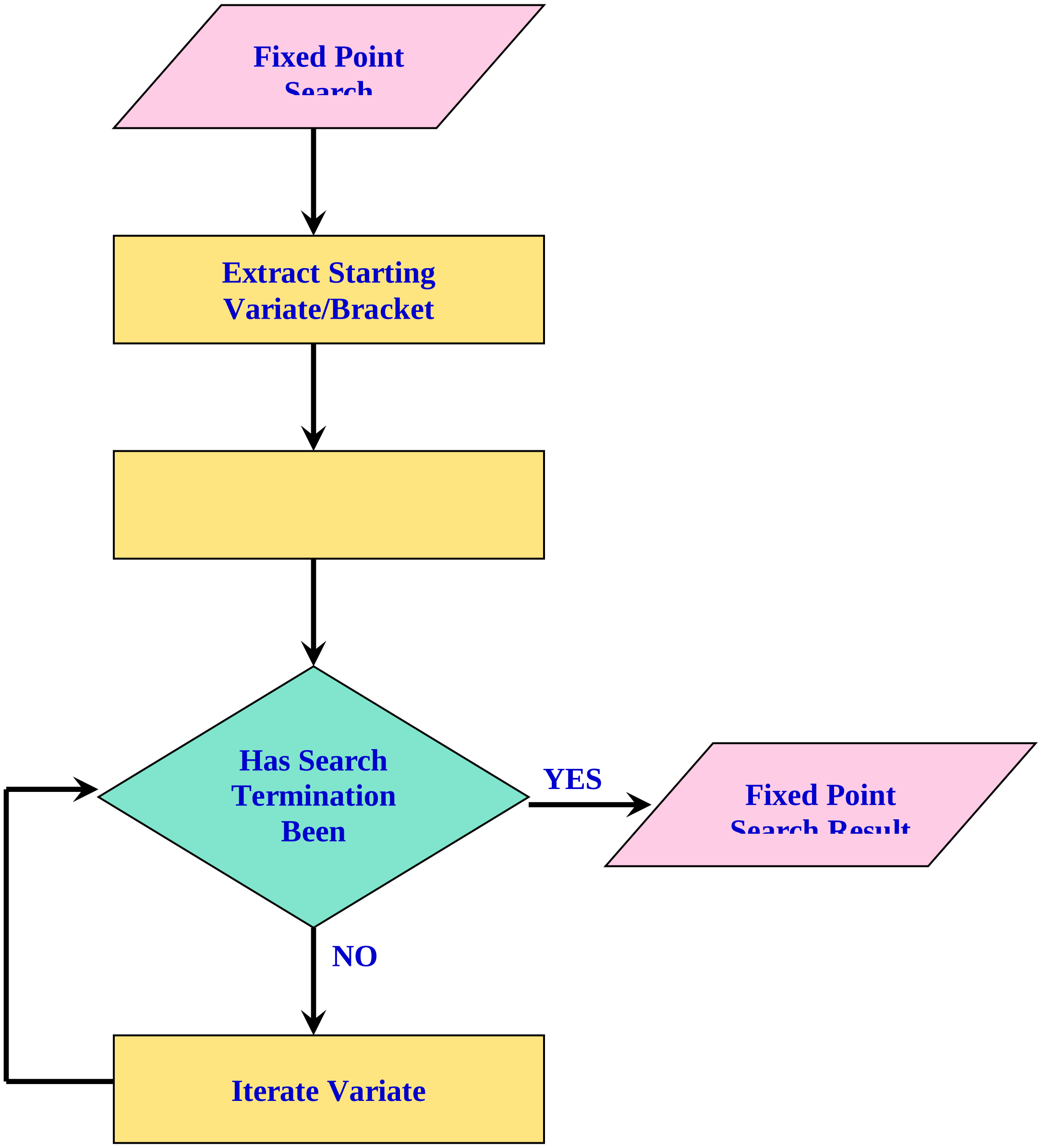
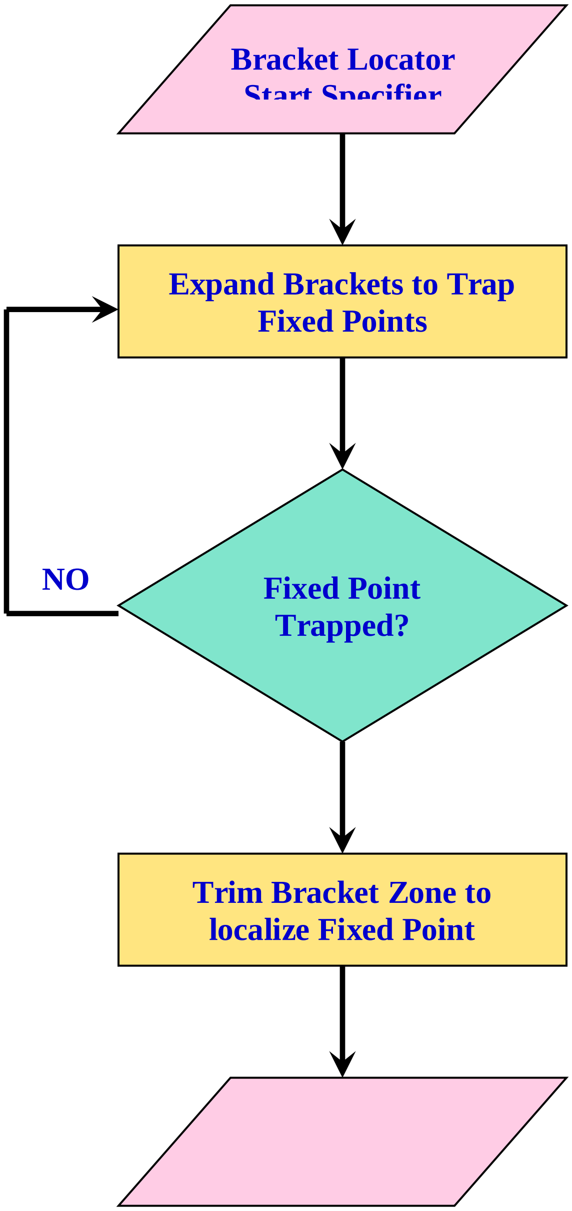
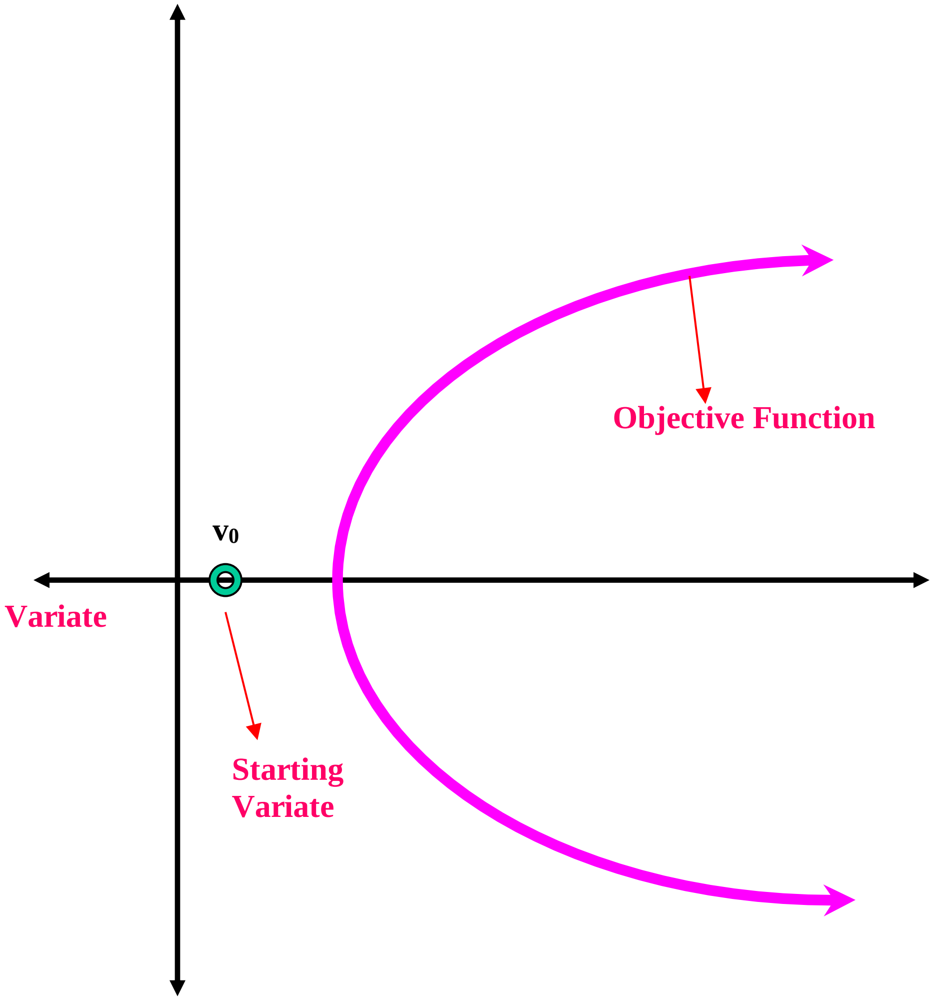


**Figure #2**
**Bracketing SKU Flow**



## Figure #3
## Objective Function Undefined at the Starting Variate



# Figure #4
# Objective Function Undefined at any of the Candidate Variates

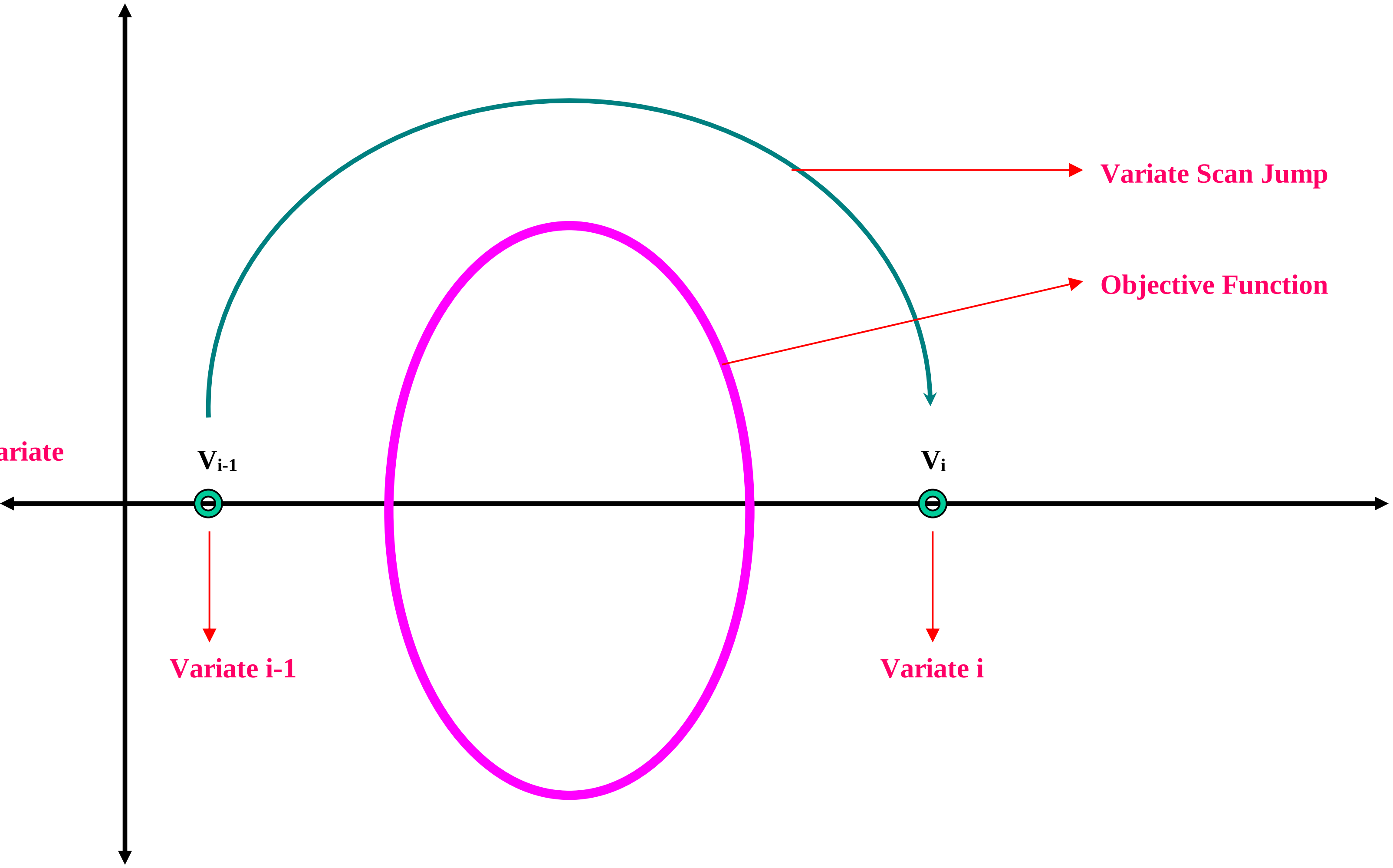
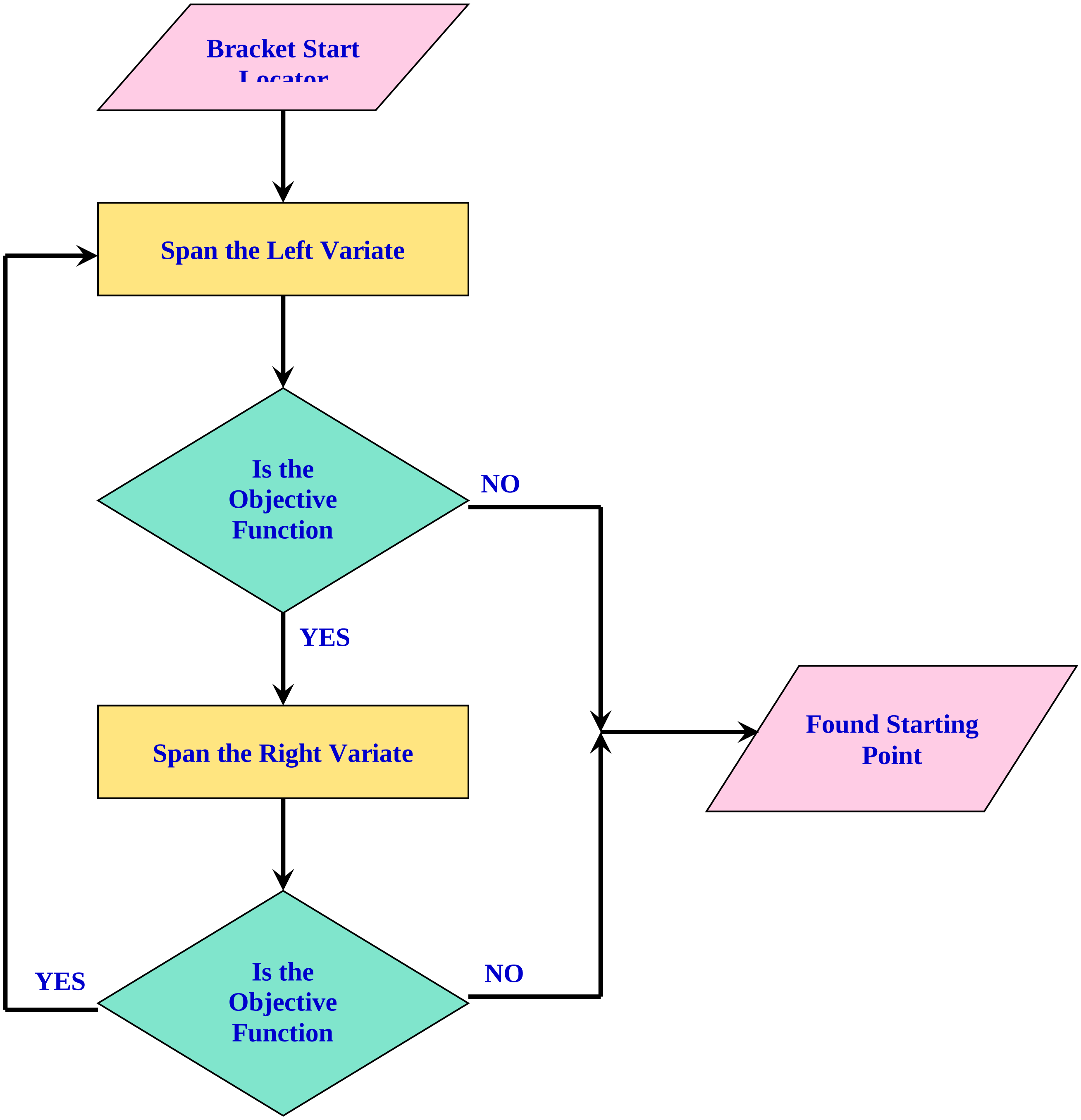
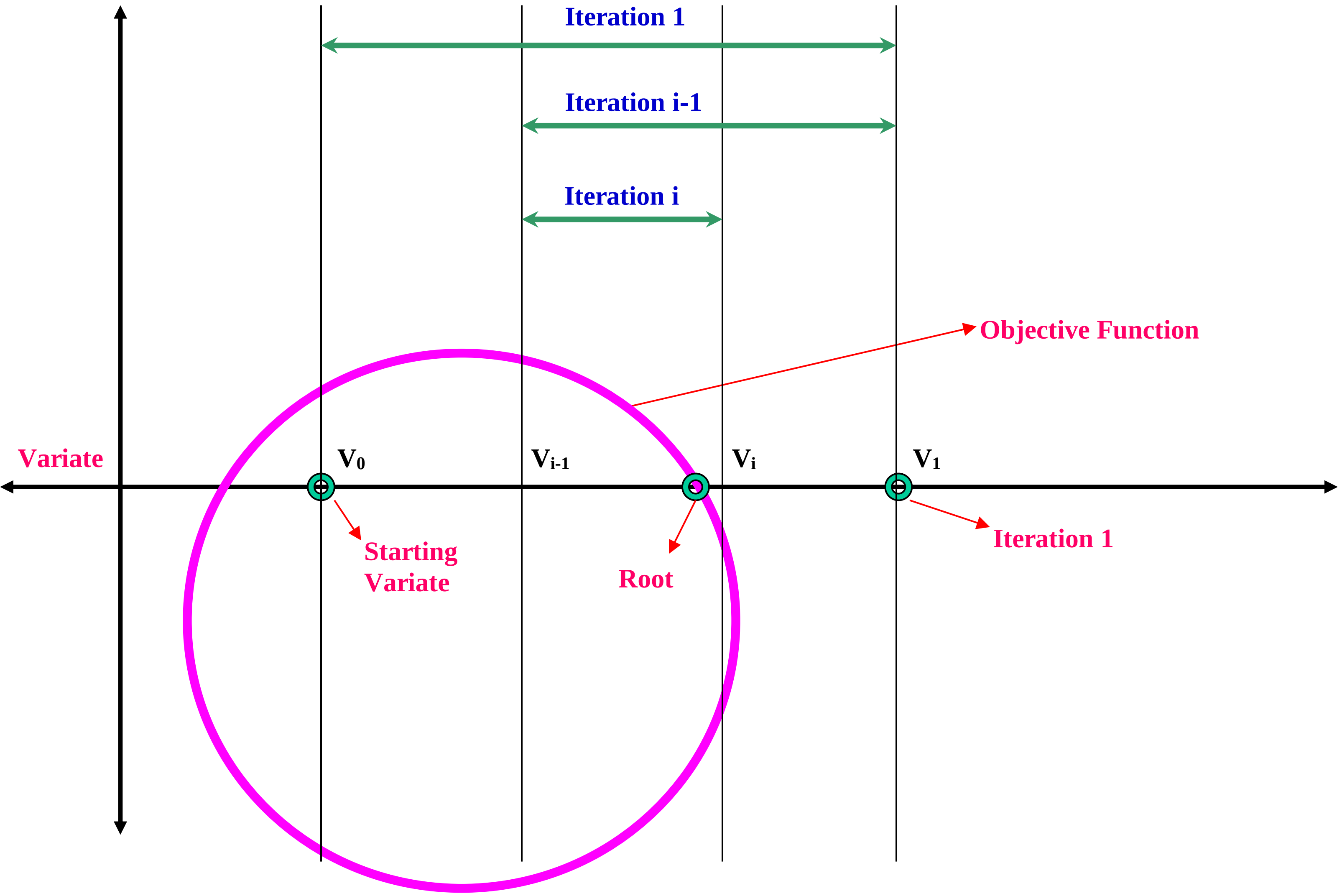


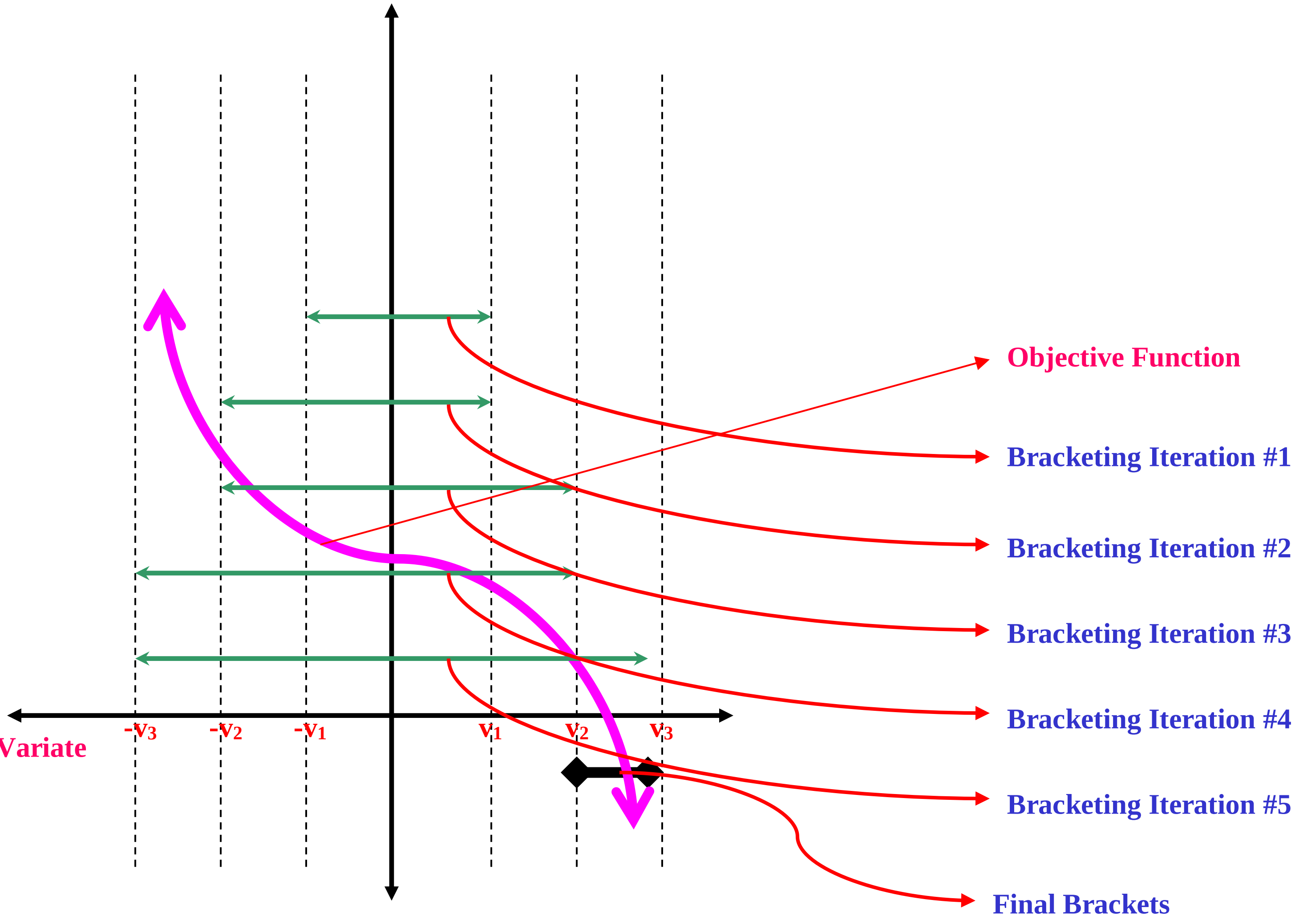
**Figure #5**
**General Purpose Bracket Start Locator**



## Figure #6
## Bracketing when Objective Function Validity is Range-bound



## Figure #7
## Objective Function Fixed Point Bracketing


Objective Function
Bracketing Iteration #1
Bracketing Iteration #2
Bracketing Iteration #3
Bracketing Iteration #4
Bracketing Iteration #5
Final Brackets
Variate
-V3
-V2
-V1
V1
V2
V3