Efficient approximation of the incomplete gamma function for use in cloud model applications

U. Blahak

Institute for Meteorology and Climate Research, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Correspondence to: U. Blahak (ulrich.blahak@dwd.de)

Published by Copernicus Publications on behalf of the European Geosciences Union.
m \sim D^3), the generalized gamma distribution is invariant under the transformation between the diameter- and the mass-representation; only the values of the parameters are different.

The parameters for Eq. (1) are not necessarily independent in natural precipitation. For example, in case of rain drops there has been observed some degree of correlation (Testud et al., 2001; Illingworth and Blackman, 2002).

Often, cloud microphysic parameters require the computation of (or are – in case of “bulk” parametrizations – based entirely on) moments of the PSD functions, leading, in case of infinite moments, to the ordinary gamma function, which is defined as

$$\Gamma(a) = \int_0^\infty e^{-t} t^{a-1} dt \quad \text{with} \quad a > 0. \quad (2)$$

For example, the mass content $L$, which is of primary importance, is the first moment of the distribution with respect to the mass representation. Generally the moments $M^{(i)}$ are defined as

$$M^{(i)} = \int_0^\infty y^i f(y) dy$$

$$= \int_0^\infty N_0 y^{i+\mu} e^{-\lambda y} dy = \frac{N_0 \Gamma \left( i + \frac{\mu + 1}{\nu} \right)}{\nu^{\frac{i+\mu+1}{\nu}} \lambda^{\frac{i}{\nu}}}, \quad (3)$$

hence the name “generalized gamma-distribution” for $f(y)$. Such infinite moments, also with non-integer $i$, enter state-of-the-art bulk (moment) cloud microphysical parametrizations in many ways, e.g., in the computation of collision rates, deposition/evaporation rates and so on, which is extensively described in textbooks (e.g., Pruppacher and Klett, 1997) or in the relevant literature (e.g., Lin et al., 1983; Seifert and Beheng, 2006; to name just a few).

However, many cloud microphysical processes involve some sort of spectral cut off, for example, collision processes like autoconversion or some 2- and 3-species collisions between solid and/or liquid particles. If, in bulk models, one wishes to parametrize such processes, e.g., the conversion of particles above a certain size/mass threshold to another species by a certain process (e.g., “wet growth”; collisions of ice particles with drops where the “outcome” depends on certain size ranges of the ice particles and the drops as proposed by Farley et al., 1989), this would require the computation of incomplete gamma functions. The lower incomplete gamma function is defined as

$$\gamma(a,x) = \int_0^x e^{-t} t^{a-1} dt \quad \text{with} \quad a > 0. \quad (4)$$

For example, consider the transformation of intermediate-density graupel particles to high-density hail particles in conditions of wet growth, which is important for hail formation. That is, if graupel particles are in an environment of high content of supercooled liquid water drops (high riming rate), then particles larger than a certain size/mass $m_{wg}$ cannot entirely freeze the collected supercooled water, because the latent heat of fusion cannot be transported away from the particles fast enough (e.g., Young, 1993). The non-frozen water might get incorporated into the porous ice skeleton of the graupel and might refreeze later, leading to an increase in bulk density (“hail”). Following Ziegler (1985), the corresponding loss of $L_g$ (graupel) to $L_h$ (hail) might be simply parameterized by

$$\frac{\partial L_g}{\partial t} \bigg|_{\text{wetgr}} = \frac{\partial L_h}{\partial t} \bigg|_{\text{wetgr}} = -\frac{1}{\Delta t} \int_{m_{wg}}^\infty m g f(m_g) dm_g$$

$$= - \frac{N_{0,g} \gamma_{\mu+2 \nu, \lambda g m_{wg}}}{\nu^2 \lambda_g^{\nu+2}} \frac{\mu g^{\nu-1}}{\lambda g^{\frac{\mu+1}{\nu}}} \Delta t \quad (5)$$

where the index $g$ denotes graupel and $\Delta t$ the numerical time step. On the right-hand side, now the upper incomplete gamma function $\gamma_{\mu}(a,x) = \Gamma(a) - \gamma_{\mu}(a,x)$ appears. Values of $m_{wg}$ are usually in a range equivalent to a diameter $\geq 1$ mm. $\mu_g$ is typically between $-0.5$ and $1$, and $\nu_g \approx 1/3$. Both $N_{0,g}$ and $\lambda_g$ are $> 0$ but quite variable, so that the corresponding value of $x$ in Eq. (4) might take on arbitrary values $> 0$.

Equation (5) is a simple but instructive example of a bulk cloud microphysical process parameterization, that comprises three typical features regarding the parameters $\mu$ and $\nu$ of the incomplete gamma function in cloud physics. First, the parameter $\mu$ depends on the shape parameters $\mu$ and $\nu$ of the assumed distribution Eq. (1) and not on $N_0$ and $\lambda$. Now, in most of the established one- or two-moment bulk schemes, $\mu$ and $\nu$ are fixed parameters which do not change during a particular model simulation, so that $\mu$ also remains fixed (only recently, authors start to make at least $\mu$ variable in a diagnostic way for some of the processes, e.g., Milbrandt and Yau, 2005a; Seifert, 2008). Second, the integration limit respectively size- or mass threshold $m$ of the process parameterization ($m_{wg}$ in the above Eq. (5) translates into the $x$-parameter $\lambda m^\nu$ of the incomplete gamma function. That means, even if $m$ is a fixed parameter, $x$ is not fixed because $\lambda$ is variable. Therefore, in cloud physics either $a$ is fixed and $x$ varies during a model simulation (which will be of importance later) or both $a$ and $x$ vary; the case of a variable $a$ and fixed $x$ has not yet occoured in literature to the best knowledge of the author. Third, $a$ attains non-integer values in most cases, depending on the choice of $\mu$ and $\nu$. As it is outlined later, an integer value of $a$ leads to an analytic expression for the incomplete gamma function, which facilitates its computation. Unfortunately, there is only a chance to get integer values of $a$ if one assumes exponential particle size- or mass-distributions, which is a serious and perhaps in many cases unphysical restriction.
Dividing Eq. (4) by the (ordinary) gamma function \( \Gamma(a) = \gamma(a, \infty) \) leads to the normalized function
\[
P(a, x) = \frac{\gamma(a, x)}{\Gamma(a)}
\]
with values monotonically increasing from 0 for \( x = 0 \) to 1 for \( x \to \infty \). The majority of the increase from 0 to 1 occurs at values of \( x \) around \( a \) with a “bandwidth” of about \( \sqrt{a} \) (see black curves in Fig. 1, in anticipation of the following).

Taking the complement \( 1 - P(a, x) \) leads to the so-called upper normalized incomplete gamma function \( Q(a, x) \), which, upon multiplication with \( \Gamma(a) \), gives the integral in Eq. (4) but with lower limit \( x \) and upper limit \( \infty \), which is denoted by \( \gamma_u(a, x) \). Computing any of \( \gamma_l(a, x), \gamma_u(a, x), P(a, x) \) or \( Q(a, x) \) will lead to any of the other functions by simple transformations.

Complete and incomplete gamma functions are well-known and treated extensively in the mathematical literature, and there are various ways to compute these functions in practice. To start with \( \Gamma(a) \), along with the well-known recurrence relations, Press et al. (1993) devise a very efficient and very accurate approximation for \( a > 0 \), which is sufficient for cloud microphysical applications, and which has been originally derived by Lanczos (1964). Incomplete gamma functions are analytic for integer values of \( a \) (see Eqs. 18–22 later in this paper); for arbitrary values of \( a \), a widely used method devised again by Press et al. (1993) uses a series expansion of \( \gamma_l(a, x) \) or a continued fraction expansion of \( \gamma_u(a, x) \), depending on whether \( x \) is larger or smaller than \( a + 1 \). These expansions are summed up to a certain number of terms until convergence is reached. The required number of terms depends on \( a \) and \( x \) and on the desired numerical accuracy.

While this is a very accurate method, the numerical burden is comparatively high within the framework of cloud models and is hard to predict because of the variable number of terms required to reach the desired accuracy, and, on vector machines, this method leads to vectorization problems, which might drastically lower the computing performance. We believe that, among other reasons, these practical problems and the supposedly high computational costs with regard to the low computing performance up to now prevented many cloud physicists from extensively using parametrizations which involve incomplete gamma functions, such as Eq. (5). In some cases where incomplete gamma functions have been used, simple analytical approximations for very special and fixed values of \( a \) were employed, e.g., a piecewise linear “ramp” function of \( x \) for non-integer \( a \) in Cotton et al. (1986). Or, as in Farley et al. (1989), finite integrals similar to the one in Eq. (5) — analytically resulting in incomplete gamma functions — have been calculated numerically (which is perhaps even more costly than Press’s method!).

However, if the parameter \( a \) is fixed during subsequent function evaluations (as is the case for many cloud microphysical process parametrizations in the context of bulk ap-proaches), the most efficient way in this case is perhaps interpolation from pre-computed regular lookup tables with respect to \( x \) at fixed \( a \). “Regular” is meant here in the sense that the indices of neighbouring table values can be computed from the interpolation point instead of a grid point search. Generally, this is achieved if the distance between tabulated values can be given by some functional form.

Perhaps the most simple example is an equidistant table, where the indices of the neighbouring values \( x_i \) and \( x_{i+1} \) in the table, required to obtain the interpolated value of \( \gamma_l \) at a point \( x \), can be explicitly computed from \( x \), the starting point \( x_1 \) of the table and the table increment \( \Delta x \),
\[
i = \text{INT} \left( \frac{x - x_1}{\Delta x} \right) + 1.
\]

This is much more efficient and predictable compared to a non-regular table, where a search loop with a nested if-clause is necessary to find \( i \). Another example of a regular lookup table would be logarithmically equidistant.

For the equidistant table, further efficiency is gained by pre-computing the constant \( 1/\Delta x \), so that the costly division is replaced by a cheap multiplication. If linear interpolation between neighbouring table values is used (accuracy might be gained from decreasing \( \Delta x \)), so that
\[
\gamma_l(x) \approx \gamma_l,i + \frac{\gamma_l,i+1 - \gamma_l,i}{\Delta x} (x - x_i),
\]
then altogether only one integer rounding operation and a few additions and multiplications have to be performed per \( \gamma_l \) evaluation. This method has been used, e.g., by Harrington et al. (1995) and Walko et al. (1995). Alternatively one could use more costly higher order interpolation schemes and at the same time reduce \( \Delta x \).

In case that \( a \) is not fixed, the lookup table would have to be two-dimensional, requiring two-dimensional interpolation techniques (bilinear in the most simple case).
The necessary span of the table with respect to $x$ at a certain value of $a$ may be taken from 0 to the value $x_{995}$ where $P(a,x_{995}) = 0.995$. $x_{995}$ as a function of $a$ may be estimated from the approximative formula

$$x_{995}(a) = g_1 (1 - \exp(g_2 a^{g_3})) + g_4 a$$

$$g_1 = 36.63 \quad g_2 = -0.1195$$

$$g_3 = 0.3393 \quad g_4 = 1.156$$

which has been obtained by nonlinear curve fitting and which is depicted in Fig. 2. At $x > x_{995}$, $P(a,x) \approx 1$ respectively $\gamma(a,x) \approx \Gamma(a)$.

Unfortunately such lookup tables also lead to vectorization problems because of memory bank conflicts, in case of many parallel vector tasks having to access the same table values at the same time. On such types of architectures, it is desirable to have an approximation formula at hand with a fixed number of mathematical terms, regardless of $a$ and $x$. At the same time, for many applications a reduced accuracy may be acceptable, e.g., approximation errors for $P$ within 0.01 absolute and/or 1% relative. Lanczos’s very accurate approximation to $\Gamma(a)$ does already fulfill the requirement of a fixed number of terms. The purpose of this paper is to develop such an approximation also for $\gamma(a,x)$, although at a reduced accuracy.

2 Approximation of the lower incomplete gamma function

We seek an approximation by means of nonlinear curve fitting, because this leads to the desired fixed number of terms, as opposed to relying on the convergence of series expansions. To motivate a regression ansatz, series expansions are however useful. A series expansion of $\gamma(a,x)$ can be obtained by plugging the Taylor series of the exponential function $\exp(-t)$ into Eq. (4) and integrating each term separately, which leads to (Abramowitz and Stegun, 1970)

$$\gamma(a,x) = x^a \sum_{i=0}^{\infty} (-1)^i \frac{x^i}{i!(i+a)}.$$  (10)

A different well-known representation (e.g., Press et al., 1993) is

$$\gamma(a,x) = \exp(-x) x^a \sum_{i=0}^{\infty} \frac{x^i}{(a+i)(a+i-1)\cdots a},$$  (11)

which can be obtained from Eq. (10) after tedious manipulation by separating the series representation of $\exp(-x)$ by means of the Cauchy-product and solving for the series coefficients by equating the pre-factors for each individual power of $x$.

It turns out that for $x \ll a$ the first two terms in Eq. (11) are sufficient to give a reasonable approximation. For larger $x$, an ever increasing number of terms is necessary. On the other hand, for $x \to \infty$, $P(a,x)$ approaches 1 seemingly similar to something like $1-c_4^{-x}$, with some positive number $c_4$. The former approximation is asymptotically correct for $x \to 0$, the latter for $x \to \infty$. Therefore, a 4-parametric ansatz $\tilde{\gamma}(a,x; c_1, c_2, c_3, c_4)$ is constructed which blends the former function (slightly modified by the the fitting parameter $c_1$) into the latter:

$$\gamma(a,x) \approx \tilde{\gamma}(a,x)$$

$$\approx \exp(-x) x^a \left( \frac{1}{a} + \frac{c_1 x}{a(a+1)} + \frac{(c_1 x)^2}{a(a+1)(a+2)} \right)$$

$$\left( 1 - W(x) \right) + \Gamma(a) W(x) \left( 1 - c_4^{-x} \right)$$  (12)

$$W(x) = \frac{1}{2} + \frac{1}{2} \tanh(c_2(x - c_3)).$$  (13)

For many fixed values of $a \in [0.1,30]$, nonlinear curve fitting by the Levenberg-Marquardt-method with respect to $x \in [0,x_{995}(a)]$ leads to a large number of coefficient sets $c_1$ to $c_4$, each coefficient being a function of $a$. $\tilde{\gamma}(a,x)$ denotes the resulting fit, $\tilde{P}(a,x) = \tilde{\gamma}(a,x)/\Gamma(a)$ and $\tilde{Q} = 1 - \tilde{P}$. The results of such curve fits have been found very pleasing, as the absolute approximation errors of the resulting normalized functions $P$ and $Q$ are generally less than 0.01 over the employed $a\times x$-domain. This corresponds to a relative error $|P - P|/P < 1\%$ for $x \lesssim a$ and $|Q - Q|/Q < 1\%$ for $x \lesssim a$. The relative errors might be larger when $P(a,x)$ and $Q(a,x)$ approach 0, but relative errors are not meaningful in that case. The fits show the same asymptotic behaviour as the original function and are found to always increase monotonically with increasing $x$, which is essential to be of practical use.
Table 1. Coefficients $P_1, q_1, r_1,$ and $s_1$ for the approximation functions $\hat{c}_1(a) \ldots \hat{c}_4(a)$ in Eq. (14).

<table>
<thead>
<tr>
<th>$i$</th>
<th>$P_1$</th>
<th>$q_1$</th>
<th>$r_1$</th>
<th>$s_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.4368392235E-03</td>
<td>1.1464706419E-01</td>
<td>0.0</td>
<td>1.0356711153E+00</td>
</tr>
<tr>
<td>2</td>
<td>-1.078266481E-04</td>
<td>2.6963429121E+00</td>
<td>1.1428711153E+00</td>
<td>2.3423452308E+00</td>
</tr>
<tr>
<td>3</td>
<td>-5.8969675295E-06</td>
<td>-2.9647038257E+00</td>
<td>-6.6981186438E-03</td>
<td>-3.6174503174E-01</td>
</tr>
<tr>
<td>4</td>
<td>2.8939523781E-07</td>
<td>2.1080724954E+00</td>
<td>1.0480765092E-04</td>
<td>-3.1376557650E+00</td>
</tr>
<tr>
<td>5</td>
<td>1.0043262986E-01</td>
<td>-</td>
<td>-</td>
<td>2.909230639E+00</td>
</tr>
<tr>
<td>6</td>
<td>5.6537848465E-01</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Fig. 3. Numerical values (grey stars) of the regression coefficients $c_1$ through $c_4$ in Eq. (12) as derived by nonlinear regression for fixed values of $a$. Black lines: nonlinear regression functions to each of $c_1$ through $c_4$ as functions of $a$ after ansatz (14) and the coefficients from Table 1.

As a last step, $c_1$ to $c_4$ are approximated as functions of $a$, again by nonlinear curve fitting. The results are

$$c_1(a) \approx \hat{c}_1(a) = 1 + P_1a + P_2a^2 + P_3a^3 + P_4a^4 + P_5(\exp(-P_6a) - 1)$$

(14)

$$c_2(a) \approx \hat{c}_2(a) = q_1 + q_2a + q_3a^2 + q_4a^3$$

(15)

$$c_3(a) \approx \hat{c}_3(a) = r_1 + r_2a + r_3a^2 + r_4a^3$$

(16)

$$c_4(a) \approx \hat{c}_4(a) = s_1 + s_2a + s_3a^2 + s_4a^3 + s_5a^4,$$

(17)

where the particular functional forms have been arrived at by guessing and experimentation. The values for the parameters $P_1 \ldots P_6, q_1 \ldots q_4, r_1 \ldots r_4$ and $s_1 \ldots s_5$ are given in Table 1. Replacing in Eq. (12) $c_1 \ldots c_4$ by $\hat{c}_1 \ldots \hat{c}_4$ leads to the final approximation $\hat{\gamma}(a,x)$. Figure 1 shows examples of the normalized $\hat{P}(a,x) = \hat{\gamma}(a,x)/\Gamma(a)$ (grey lines) as function of $x$ for some fixed values of $a$ in comparison to the original function $P(a,x)$ (black lines), demonstrating a reasonably good agreement.

However, Fig. 3 shows the coefficients $c_1$ to $c_4$ as obtained by the curve fits at constant $a$ (grey stars) along with the corresponding approximations $\hat{c}_1 \ldots \hat{c}_4$ given by Eqs. (14)–(17) (black lines), and some difficulties at the lower range of $a$-values ($a < 0.9$) are apparent. Therefore, $\hat{\gamma}(a,x)$ (resp. $\hat{P}(a,x)$) with the set of coefficients in Table 1 is not likely to be a good approximation to $\gamma(a,x)$ (resp. $P(a,x)$) for
Fig. 4. Relative (left) and absolute (right) error of the proposed approximation \(\hat{P}(a,x) = \gamma(a,x)/\Gamma(a)\) as function of \(a\) and \(x/(a+1)\). The parameter \(x\) has been scaled by \(a+1\) because the main variation of \(P(a,x)\) with \(x\) takes place at \(x\)-values around \(a+1\). The vertical grey lines at \(a = 0.9\) indicate the lower boundary with respect to \(a\) of the \((a,x)\)-range where errors are acceptable.

\[
a < 0.9, \text{ a fact that is clearly demonstrated in Fig. 4, which depicts the relative and absolute errors } (\hat{P} - P)/P \text{ (left plate) and } \hat{P} - P \text{ (right plate) as function of } a \text{ and } x. \text{ In these figures, } x \text{ has been scaled by } a+1 \text{ since } P \text{ varies mostly in the region of } x \text{ around } a+1.
\]

Where do these difficulties for small values of \(a\) come from? To understand this, it is instructive to look at the analytical formulas for \(\gamma(a,x)\) in case of \(a\) being an integer, starting with \(m = 1,\)

\[
\begin{align*}
\gamma(1,x) &= 1 - e^{-x} \\
\gamma(2,x) &= 1 - (x+1)e^{-x} \\
\gamma(3,x) &= 2 - (x^2 + 2x + 2)e^{-x} \\
\gamma(4,x) &= 6 - (x^3 + 3x^2 + 6x + 6)e^{-x} \\
&\vdots
\gamma(m,x) = \left[-t^{m-1}e^{-t}\right]_0^x + (m-1) \int_0^x t^{m-2}e^{-t} \, dt \\
&= -x^{m-1}e^{-x} + (m-1)\gamma(m-1,x) \\
&= \Gamma(m) - \left(\sum_{i=0}^{m-1} \frac{d}{dx}^i x^{m-1}\right) e^{-x}. \quad (22)
\end{align*}
\]

The last finite sum representation has been deduced from the representations at \(m = 1, 2, \ldots\) and can be proved by a) taking the first derivative with respect to \(x\), which yields \(x^{m-1} e^{-x}\) as it should, and b) by the fact that the highest derivative (last element in the sum) equals \((m-1)\Gamma(m)\), so that for \(x = 0\) the correct value \(\gamma(m,0) = 0\) is obtained. For small integer values of \(m\), this representation is an efficient way to evaluate \(\gamma\).

Now, concerning the difficulties for small values of \(a\), \(\gamma\) for \(a = 1\) (Eq. 18) is a simple exponential function which matches the ansatz (12) only in the limits \(c_2 \to \infty, c_3 = 0\) and \(c_4 = e\). As \(a \to 1\), the coefficients \(c_2\) to \(c_4\) go towards these values, as is shown in Fig. 3, with the singularity of \(c_2\) at \(a = 1\). Of course at \(a = 1\) the fitting algorithm leads to a “compromise”-solution with finite \(c_2\); otherwise, a kind of branch-cut of the coefficients around \(a = 1\) can be observed. For larger values of \(a\), the analytical formulation of \(\gamma\) gets more complicated (for \(a\) being an integer \(m\), more and more terms get involved into the finite sum in Eq. (22) as \(m\) increases) and is more likely to be well describable by the ansatz (12).

For most cloud microphysical applications, the branch for \(a > 1\) seems to be the important one. Therefore, the approximations (14)–(17) are developed to represent mainly that branch. By altering the regression functions to, e.g., rational functions, it would perhaps be possible to find approximations which encompass both branches.

It turns out, however, that the approximations (14)–(17) are applicable for \(a\)-values down to 0.9, because for \(a \geq 0.9\), the absolute error \(|\hat{P} - P|\) given in Fig. 4 (right figure) remains below 0.02 everywhere, and the relative error \(|(\hat{P} - P)/P|\) (left figure) is also quite small except when \(P\) gets close to 0. But here, the relative error is not a good error measure anyways. Beyond the range of \(a\)-values for which \(\hat{P}\) has been fitted, it turns out that the approximation error remains within the same small limits up to \(a = 45\). Above that, \(\hat{P}\) is not a good approximation.

A further condition for \(\hat{P}\) to be of practical use is that this function increases monotonically with respect to increasing \(x\) at fixed \(a\), as does the original function \(P\). One possibility to
check monotonicity is to look at the sign of the partial derivative of \( \hat{P} \) with respect to \( x \). Figure 5 shows \( \partial \hat{P}(a,x)/\partial x = 1/\Gamma(a) \partial \hat{\gamma}(a,x)/\partial x \) as function of \( a \) and \( x/(a+1) \). Same scaling of \( x \) and same grey line as in Fig. 4.

Concerning the efficiency of the proposed approximation, it has been found that, on our scalar linux desktop computer using the gfortran compiler and high optimization, it is faster by a moderate factor 4 on average compared to the efficient method of Press et al. (1993) mentioned in the introduction. Here, the speed-up depends on \( a \) and \( x \), that is, on the number of required terms and the desired accuracy in Press’s method. The latter has not been changed from its original values.

A more impressive speed-up is gained, however, in the case where subsequent evaluations at the same value of \( a \) are needed, as is the case in cloud physics: many coefficients of the terms in the proposed approximation only depend on \( a \) and may be pre-computed once, so that we were able to obtain a speed-up factor of about 15 on average. But the biggest advantage is perhaps on vector CPUs, where the fixed number of terms makes vectorization easy for programmers and compilers. Also, there might be advantages on massively parallel machines in that the method avoids load imbalances otherwise caused by different numbers of terms in series expansions to reach convergence at different grid points.

Further experiments, in which the hyperbolic tangent in Eq. (12) has been replaced by the simple but accurate and continuously differentiable rational approximation

\[
\tanh(x) \approx \begin{cases} 
-1 & x \leq -c_1 \\
\frac{9c_2^2x + 27x^3}{c_1^2 + 27c_1x^2} & -\frac{c_2}{3} < x < \frac{c_2}{3} \\
c_1 = 9.37532 & \quad c_1 = 9.37532 \\
1 & x \geq \frac{c_2}{3}
\end{cases}
\]

Eq. (23) might be tried instead. Observe that changing the value for \( c_1 \) would only rescale the function along the x-axis but preserve the differentiability and the “outer” function values −1 and 1, a fact that renders Eq. (23) a quite general blending function.

3 Summary

This paper describes an approximation to the lower incomplete gamma function which has been obtained by nonlinear curve fitting. It comprises a fixed number of terms and yields moderate accuracy (absolute approximation error of \( P \) is 0.02 in the range 0.9 ≤ \( a \) ≤ 45 and \( x \geq 0 \), which should be enough for most cloud microphysical applications, but may be problematic for other applications. The proposed approximation \( \hat{P} \) consists of Eq. (12) in combination with Eqs. (14)-(17) and coefficients given in Table 1. Monotonicity and asymptotic behaviour of the original incomplete gamma functions are preserved, which is important.

The method is generally only slightly more efficient in terms of the required number of floating point operations than the more accurate method of Press et al. (1993), but if subsequent evaluations at a certain fixed value of \( a \) are sought (which is often the case in cloud microphysics), then a significant performance increase can be obtained by pre-computing certain terms and coefficients which only depend on \( a \). On scalar architectures, however, the most efficient method to evaluate incomplete gamma functions is certainly interpolation of regular lookup tables as described in the introduction.

A great advantage of the proposed approximation \( \hat{P} \) is its applicability on vector CPUs, because the formula with its fixed number of terms is well suited for vectorization, as opposed to, e.g., Press’s method or other more accurate algorithms based on series- or continued-fraction representations.

With regard to massively parallel machines (where the use of equidistant lookup tables might pose memory issues), the fixed number of terms might be also beneficial to avoid load imbalances caused by different numbers of terms in series expansions to reach convergence at different grid points.

Acknowledgements. This work has been inspired by discussions with Axel Seifert (German Weather Service), who also suggested some improvements to the manuscript. Very valuable comments of two anonymous referees helped to improve and clarify the manuscript.

Edited by: O. Boucher
References


