ORIGINAL PAPER

A series acceleration algorithm for the gamma-Pareto (type I) convolution and related functions of interest for pharmacokinetics

Carl A. Wesolowski[1,2] · Jane Alcorn[1] · Geoffrey T. Tucker[3]
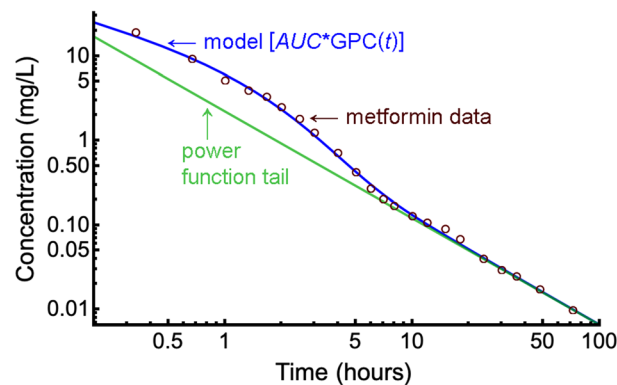
Abstract
The gamma-Pareto type I convolution (GPC type I) distribution, which has a power function tail, was recently shown to describe the disposition kinetics of metformin in dogs precisely and better than sums of exponentials. However, this had very long run times and lost precision for its functional values at long times following intravenous injection. An accelerated algorithm and its computer code is now presented comprising two separate routines for short and long times and which, when applied to the dog data, completes in approximately 3 min per case. The new algorithm is a more practical research tool. Potential pharmacokinetic applications are discussed.

Graphic abstract

Gamma-Pareto convolution (GPC) new algorithm regression run time example using 72 h intravenous metformin 22 sample dog 1 data.

New run time: 3:00 (min:s)
Old run time: 33:10 (min:s)

✉ Carl A. Wesolowski
carl.wesolowski@gmail.com

1   College of Pharmacy and Nutrition, University of Saskatchewan, 104 Clinic Place, Saskatoon, SK S7N 2Z4, Canada

2   Department of Medical Imaging, Royal University Hospital, College of Medicine, University of Saskatchewan, 103 Hospital Drive, Saskatoon, SK S7N 0W8, Canada

3   Department of Human Metabolism, Medical and Biological Sciences, University of Sheffield, Sheffield, UK

Introduction

Gamma-Pareto convolutions (GPC), the convolution of a gamma distribution with some type of Pareto distribution, are increasingly used for modelling diverse random processes like traffic patterns [1], flood rates, fatigue life of aluminium, confused flour beetle populations [2], and extreme rainfall events [3]. Although there are multiple possible GPC models and different nomenclatures used to describe them, a natural classification would arise from Pareto distribution classification, types I through IV, and the Lomax distribution, a type II subtype, which is the

classification scheme of reference [4] and the Mathematica computer language [5].[1]

Convolution was first introduced to pharmacokinetics in 1933 by Gehlen who used the convolution of two exponential distributions,

$$
\begin{aligned}
\mathrm{EDC}(t; b, \beta) &= b e^{-bx} * \beta e^{-\beta x}(t) \\
&= \begin{cases}
\left. \begin{array}{ll}
b\beta \dfrac{e^{-\beta t} - e^{-bt}}{b - \beta} & b \neq \beta \\[3mm]
b^2 t\, e^{-bt} & b = \beta
\end{array} \right\} & t \geq 0 \\[6mm]
\left. \begin{array}{ll}
0 &
\end{array} \right\} & t < 0
\end{cases}
\end{aligned}
\tag{1}
$$

to describe plasma concentration-time data, and as originally developed in 1910 by Bateman to model radioactive decay [6–8]. Much later, in 2006, the Bateman equation was generalised as an exact gamma-gamma convolution (GDC) by Di Salvo [9]. Ten years later, this was then applied to 90 min continuous recordings of radioactivity in human thyroid glands following injection of ^{99m}Tc-MIBI [10].[2] In 1919, Widmark identified integration of a monoexponential as a model for constant infusion [11]. That integration from zero to t to find a constant infusion model can be applied not just to exponential functions, but applies equally well to any area under the curve (AUC) scaled density-function (pdf)[3] model, was shown subsequently by Wesolowski et al. [12, 13]. Recently, the disposition of metformin was described precisely using the type I GPC model, which as it is asymptotically a Pareto function, has a power function tail [13]. Using direct comparison rather than classification, power function tails were shown to be always heavier than exponential tails, see the Appendix Subsection entitled Relative tail heaviness of reference [13]. A power function tail, in turn, implies an underlying fractal structure, where fractal in this context signifies a scale invariant model of vascular arborisation [14].

The GPC computer algorithm used in 2019 had long run times and was not accurate beyond 96 h [13]. These problems were corrected in order to make predictions for multiple dosing over longer times [15, 16]. Since computer implementation of new functions is highly specialised, not easily arrived at by induction and yet indispensable for any practical application, documentation of a more practical

type I GPC algorithm may facilitate its more widespread implementation. Accordingly, we now present a series acceleration computer implementation of a more generally applicable GPC type I function, with markedly shorter run times.

Background

The gamma-Pareto convolution distribution function family

A classification for gamma-Pareto convolutions (GPC) is proposed that arises from the types of Pareto distributions [4]. These are types I through IV plus the Lomax distribution, a subtype of II. The Pareto type I distribution is

$$
\mathrm{PD}(t; \alpha, \beta) = \frac{\alpha}{t}\left(\frac{\beta}{t}\right)^{\alpha} \theta(t - \beta) ,
\tag{2}
$$

where α is the shape parameter, β is a scale parameter and $\theta(\cdot)$ is the unit step function such that $\theta(t - \beta)$ is the unit step function time-delayed by β, and is used to make a product that is non-zero only when $t > \beta$.[4]

A type II Pareto distribution can be written as

$$
\mathrm{PD}_{\mathrm{II}}(t; \alpha, \beta, \mu) = \frac{\alpha}{\beta}\left(1 + \frac{t - \mu}{\beta}\right)^{-\alpha - 1} \theta(t - \mu) .
\tag{3}
$$

Setting $\mu = 0$, this becomes the Lomax distribution; $\mathrm{PD}_{\mathrm{Lomax}}(t; \alpha, \beta) = \frac{\alpha}{\beta}\left(1 + \frac{t}{\beta}\right)^{-\alpha - 1}\theta(t)$, which was used to derive a Lomax gamma-Pareto distribution [1]. The relevance of this is that the GPC type I and Lomax GPC derivations are similar. As yet, the type II (not Lomax) through type IV gamma-Pareto convolutions have not been published. These convolutions are likely to be infinite sums and may require series acceleration to be of practical use. By substitution and reduction of the number of parameters, there are closed form GPC-like expressions, types II through IV, that are different distributions [2]. As a full set of solutions for the entire GPC function family has not been characterised, it is not known what additional applications there could be for the GPC family of functions.

Unlike the Lomax GPC, the GPC type I does not start at $t = 0$, but at $t = \beta$. For pharmacokinetic modelling, $\beta > 0$ is a measure of the circulation time between injection of an intravenous bolus of drug ($t = 0$), and its arrival at a

---

[2] 740 MBq technetium-99m labeled hexakis-methoxy-isobutyl-isonitrile.

[3] We retain the acronym pdf without a probability; p, but use $f(t)$ preferentially. Concentration models are the product of area-under-the-curve of concentration and density functions whose total area-under-the-curve is 1 (dimensionless dose fraction). This balances the classical mechanical units of Mass, Length, and Time, as follows, $C(t) = AUC \times f(t)$; $\left[\frac{M}{L^3} = \frac{M\,T}{L^3} \times \frac{1}{T}\right]$.

[4] The unit step function, $\theta(x)$, is zero for $x < 0$ and 1 for $x \geq 0$, such that $\theta(x)$ is continuous everywhere except at $x = 0$. When $x = t - \beta$ and $\beta > 0$, then $\theta(t - \beta)$ is a unit step function shifted to later time (i.e., to the right) by β units in the new coordinate system; t. The unit step function is faster for numerical computations than the Heaviside theta function, which later is sometimes also symbolised as $\theta(x)$. The Heaviside theta is more mathematically useful when it is continuous everywhere such that its derivative and Laplace transform are defined.

peripheral venous sampling site ($t = \beta$). The four-parameter gamma Pareto (type I) convolution (GPC) density function was developed to model the disposition of metformin in dogs, which exhibited an unexpectedly heavy tail poorly described by an exponential decay [13]. This heavy tail implies a prolonged buildup of the body burden of the drug [15, 17] that may require dose tapering on long-term use, especially in patients with renal impairment [16].

The Gamma-Pareto type I convolution and related functions

GPC type I: To form a GPC type I model, the type I Pareto distribution, Eq. (2), is convolved with a gamma distribution,

$$GD(t; a, b) = \frac{1}{t} \frac{e^{-bt}(bt)^a}{\Gamma(a)} \theta(t) ,\qquad (4)$$

where a is a dimensionless shape parameter, b is a rate per unit time, is the reciprocal of its scale parameter, and $\Gamma(\cdot)$ is the gamma function.[5] This yields the GPC function,

$$GPC(t; a, b, \alpha, \beta)$$
$$= \theta(t - \beta) \frac{b^a \alpha \beta^\alpha}{\Gamma(a)} t^{a-\alpha-1} \sum_{n=0}^{\infty} \frac{(-bt)^n}{n!} B_{1-\frac{\beta}{t}}(a + n, -\alpha) ,$$
$$(5)$$

where $B_z(\cdot, \cdot)$ is the incomplete beta function.[6] This is a density function (a pdf, or more simply an f, with units per time; t^{-1}). Equation (5) is from convolution following Maclaurin series expansion of e^{-bt}, i.e., it is analytic. An analytic function has any number of sequential multiple integrals and derivatives, as illustrated in the following equations. Compared to their prior expression [13], the equations that follow have been put in simpler terms.

GPC type I integral: Equation (6) is the cumulative density function (CDF) of the GPC, symbolised by F, the integral of the $f(t)$ density; $F(t) = \int_0^t f(\tau) d\tau$,

$$GPC_F(t; a, b, \alpha, \beta)$$
$$= \theta(t - \beta) \frac{b^a \alpha \beta^\alpha}{\Gamma(a)} t^{a-\alpha} \sum_{n=0}^{\infty} \frac{(-bt)^n}{(a+n)n!} B_{1-\frac{\beta}{t}}(1 + a + n, -\alpha) .$$
$$(6)$$

This equation, because it is a CDF, expresses the dimensionless fraction of a unit drug dose eliminated from the body as a function of time, and was used to calculate a prolonged retention of metformin in dogs and to explain its

incomplete urinary recovery at 72 h following intravenous injection in humans [13, 15, 17].

GPC type I double integral: Equation (7) is the double integral of the density function, f, which is also the single integral of F, the CDF, and is sometimes called a "supercumulative" distribution [18]. It is symbolised by $\mathcal{F}$, i.e., $\mathcal{F}(t) = \int_0^t F(\tau) d\tau = \int_0^t \int_0^\tau f(x) dx d\tau$. The $GPC_{\mathcal{F}}$ in least terms is

$$GPC_{\mathcal{F}}(t; a, b, \alpha, \beta) = \theta(t - \beta) \frac{b^a \alpha \beta^\alpha}{\Gamma(a)} t^{a-\alpha+1}$$
$$\sum_{n=0}^{\infty} \frac{(-bt)^n}{(a+n)(1+a+n)n!} B_{1-\frac{\beta}{t}}(2 + a + n, -\alpha) .$$
$$(7)$$

This equation (units t) was used to construct an intravenous bolus multidose loading regimen that maintains the same mean amount of metformin in the body during successive dose intervals [13] and to predict metformin buildup during constant multidosing in humans both with normal renal function and with renal insufficiency [16]. A further use of this equation is to predict the cumulative distribution function following a period of constant infusion given only its bolus intravenous-concentration, fit function.

GPC type I derivative: Equation (8) is the derivative of the GPC density, GPC', or in general an f',

$$GPC'(t; a, b, \alpha, \beta)$$
$$= \theta(t - \beta) \frac{b^a \alpha \beta^\alpha}{\Gamma(a)} t^{a-\alpha-2}$$
$$\sum_{n=0}^{\infty} \frac{(a + n - 1)(-bt)^n}{n!} B_{1-\frac{\beta}{t}}(a + n - 1, -\alpha) .$$
$$(8)$$

This equation (units t^{-2}) is useful for finding the peaks of the GPC function by searching for when it equals zero, and for calculating disposition half-life from its general definition,

$$t_{1/2}; f(t) \overset{\text{def}}{=} -\ln(2) \frac{f(t)}{f'(t)} ,$$

which is Eq. (6) of reference [13]. Note that there is a pattern in the sequential integrals and derivatives that illustrates the analyticity of the GPC function. The integrals and derivatives above follow directly from integration or differentiation of the GPC formula, for which the following identity from integration by parts[7]

$$B_z(A + 1, B) = \frac{A}{A + B} B_z(A, B) - \frac{z^A (1 - z)^B}{A + B} ,$$

is useful for simplifying the results.

Methods, algorithms for GPC type I series acceleration and their computation

Data sources and regression methods

The source data for regression analysis and subsequent algorithm testing consists of seven intravenous bolus metformin studies performed in healthy mixed-breed dogs [19]. The 19 to 22 samples per case drawn between 20 min to 72 h postinjection are listed as open data in *Supplementary material 1 (SLSX 49kb)* in [13].[8] The regression target was the so-called $1/C^2$ weighted ordinary least squares (OLS) method, implemented as minimisation of the proportional norm, which is also the relative root mean square (rrms) error, as per the *Concentration data and fitting it* Appendix Subsection of [13].[9] The loss function chosen to be minimised agreed with the error type the measurement system assay calibration curve. Both the metformin assay (5.2% rrms) [20], and the GPC residuals (8.6% rrms) exhibited proportional error. The reuse of assay loss functions for regression loss functions is systemically consistent and appears in these references [10, 13]. The regression method used was Nelder-Mead *Constrained Global Numerical Minimisation* as implemented in Mathematica, a global search technique [5].[10] For 20 significant figure results for all parameters used was the Mathematica routine NMinimize with the options: PrecisionGoal → 30, AccuracyGoal → 30, WorkingPrecision → 65, MaxIterations → 20010, Method → {"NelderMead", "PostProcess" → False}. Post processing is disallowed because it launches a constrained convex gradient solution refinement protocol; the interior point method, which does not converge. The use of parameter starting value ranges close to the solution helps speed up convergence. Note that regression can start with 65 significant figure accuracy but finish with less than half of that for some parameter values due to error propagation from the fit function itself and/or the regression process. In order to calculate the confidence intervals (CI) of the parameters, model-based bootstrap [21] was performed, as follows. Care was taken to verify the normality of fit residuals and the homoscedasticity of residuals—see [13]—as suggested by [22]. Those conditions allow for the residuals to be randomly sampled with replacement, then added to the model at the sample-times to create synthetic data having the same properties as the original data, but which have

altered regression parameter solutions. The bootstrap parameter values so obtained can provide more information than gradient method parameter CV's, as the latter only provides single case-wise estimates, which are not as statistically useful as case-wise distributed parameter information [13]. Table 1 shows both case-wise and population-wise coefficients of variation from an early version of a GPC algorithm. The table was amalgamated from Tables 1, 3, and 12 of [13] representing 24 h of 8-core parallel processing of 42 time-sample serum curves. There is thus an obvious need for a faster algorithm for regression analysis.

GPC type I primary definition: the short-*t* algorithm

The primary definition of a gamma-Pareto type I convolution, Eq. (5), is

$$
\begin{aligned}
\mathrm{GPC}&(t; a, b, \alpha, \beta) \\
&= \mathrm{GD}(x; a, b) * \mathrm{PD}(x; \alpha, \beta) \ (t) \\
&= \left[\frac{(b\,x)^a\, e^{-b\,x}}{x\,\Gamma(a)} \theta(x) \right] * \left[\frac{\alpha}{x} \left(\frac{\beta}{x} \right)^{\alpha} \theta(x - \beta) \right] (t) \qquad . \quad (9) \\
&= \theta(t - \beta)\, t^{a-\alpha-1} \frac{\alpha\, b^a \beta^{\alpha}}{\Gamma(a)} \sum_{n=0}^{\infty} \frac{(-b\,t)^n}{n!} B_{1-\frac{\beta}{t}}(a + n, -\alpha)
\end{aligned}
$$

This contains alternating terms in the summation such that the sum is rapidly convergent for t not much greater than its lower limit, β. However, for sufficiently large values of t, the individual terms of the summation both alternate in sign and become extremely large in *magnitude* (i.e., absolute value) before absolute series convergence. For absolute convergence of an alternating series the infinite sum of the absolute values is bounded above, which permits rewrite of the summation sequence of infinite sums. This, and the ratio test [23] for it are shown in the Short-*t* GPC convergence Appendix Subsection. Thus, the order of infinite summation can be changed to obtain shorter run times when $t \gg \beta$, and the algorithm is accelerated through an algebraic rewrite of Eqs. (9) as (10) below. Alternating infinite series with large magnitude terms occurring before absolute convergence are common, for example, the infinite-series, primary definition of $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots$ has that same property for larger magnitude x-values. Acceleration for the sine function could include converting the x-values to be principal sine values ($-\frac{\pi}{2}$ to $\frac{\pi}{2}$), and adjusting the output accordingly.[11] For the $\mathrm{GPC}(t)$ function a similar result, i.e., adjusting the

---

Table 1 Shown are parameters from gamma densities (GD), Pareto densities (PD) and both from Gamma-Pareto convolution (GPC) fitting of concentrations data for 7 dogs

Functions	GD				PD				GPC				
Parameters	a		b		α		β		AUC		CL		Fit error
Units[a]	None	%	$\frac{1}{h}$	%	None	%	s	%	$\frac{mg \cdot h}{L}$	%	$\frac{ml}{min \cdot kg}$	%	%
Dog 1	0.3493	8.09	0.7318	4.84	0.2644	5.13	25	–	31.16	6.22	9.8	6.42	8.7
Dog 2	0.8112	10.5	0.9993	6.89	0.1365	4.90	25	–	28.18	1.09	11.5	1.09	6.3
Dog 3	0.6689	8.67	0.9107	5.95	0.2010	3.81	25	–	12.15	3.77	26.7	3.72	5.9
Dog 4	0.6092	22.3	0.8062	15.9	0.1726	9.80	25	–	16.73	5.65	19.4	5.89	13.8
Dog 5	0.6435	20.6	1.1035	11.4	0.1199	6.11	25	–	26.21	5.37	12.4	5.14	9.5
Dog 6	0.5194	7.42	0.6137	4.93	0.1929	5.85	30	–	28.43	2.15	11.4	2.10	6.1
Dog 7	0.7629	17.7	1.0518	20.3	0.1571	5.82	30	–	22.10	2.31	14.7	2.34	10.0
Case%[b]		14.8		11.5		6.17		–		4.22		4.26	8.2
Pop.%		29.7		25.0		25.9		–		28.8		39.5	7.5[c]

[a]Units row: *None* means *dimensionless*. As β was constrained to be within 25 to 30 s, its variability for the 5 realisations per case is not meaningful

[b]Used with model-based bootstrap root mean square case-wise (Case%, $n = 5$) and population-wise (Pop.%, $n = 35$) coefficients of variation, and fit error

[c]Geometric mean (GM) was used to calculate group error of fit because the 35 model-based bootstrap fit errors errors were log-normally distributed for which the GM was 7.5%, not significantly different from the original data GM error of 8.2%. The original data values are case-wise listed for fit error, and for the parameter values

algorithmic behaviour to be accelerated for long-t values, can be obtained as follows.

GPC type I secondary definition: the long-t algorithm

Theorem 1 *The long-t algorithm is*

$$\text{GPC}(t; a, b, \alpha, \beta)$$
$$= -\theta(t - \beta) \frac{\alpha b^a}{\Gamma(a)} t^{a-1} \sum_{k=1}^{\infty} \left(\frac{\beta}{t}\right)^k \frac{(1-a)_k}{k!(k-\alpha)} {}_1F_1(a, a-k; -bt)$$
$$+ \theta(t - \beta) \left[\frac{b^a}{\Gamma(a)} e^{-bt} t^{a-1} \right.$$
$$\left. - \pi \csc(\pi \alpha) \frac{b^a \beta^\alpha}{\Gamma(\alpha)} t^{a-\alpha-1} {}_1\tilde{F}_1(a, a-\alpha; -bt) \right] . \quad (10)$$

Proof This is shown by substitution of the identities,[12] $B_z(A, B) = B(A, B) - B_{1-z}(B, A)$, and $B(A, B) = \frac{\Gamma(A)\Gamma(B)}{\Gamma(A+B)}$ into the incomplete beta function of Eq. (9) above and yields,

$$B_{1-\frac{\beta}{t}}(a+n, -\alpha) = \frac{\Gamma(-\alpha)\Gamma(a+n)}{\Gamma(a+n-\alpha)} - B_{\frac{\beta}{t}}(-\alpha, a+n) . \quad (11)$$

Substituting this into the right hand side of Eq. (9) yields,

$$\theta(t - \beta) \frac{\alpha b^a \beta^\alpha}{\Gamma(a)} t^{a-\alpha-1}$$
$$\sum_{n=0}^{\infty} \left[\frac{(-bt)^n}{n!} \frac{\Gamma(-\alpha)\Gamma(a+n)}{\Gamma(a+n-\alpha)} - \frac{(-bt)^n}{n!} B_{\frac{\beta}{t}}(-\alpha, a+n) \right] , \quad (12)$$

the left hand summand of which simplifies to a GPC asymptote for long times, t,

$$\theta(t - \beta) \alpha b^a \beta^\alpha \Gamma(-\alpha) t^{a-\alpha-1} {}_1\tilde{F}_1(a, a-\alpha; -bt) ,$$

where ${}_1\tilde{F}_1(\cdot, \cdot; z)$ is the regularised confluent hypergeometric function.[13] The above formula, as[14] $-\pi \csc(\pi \alpha) = \Gamma(-\alpha)\Gamma(\alpha+1) = \alpha\Gamma(-\alpha)\Gamma(\alpha)$, can be written alternatively as

---

[12] http://functions.wolfram.com/06.19.17.0008.01. and http://functions.wolfram.com/06.18.02.0001.01.

[13] Where ${}_1\tilde{F}_1(a; b; z) = {}_1F_1(a; b; z)/\Gamma(b)$, where ${}_1F_1(a; b; z) = \sum_{k=0}^{\infty} z^k (a)_k / [k!(b)_k]$ is the not regularised version, and where $(a)_k = \Gamma(a+k)/\Gamma(a)$ is the Pochhammer, also called the descending factorial.

[14] http://functions.wolfram.com/06.05.16.0001.01 and http://functions.wolfram.com/06.05.16.0002.01.

$$-\theta(t-\beta)\pi\csc(\pi\alpha)\frac{b^a\beta^\alpha}{\Gamma(\alpha)}t^{a-\alpha-1}{}_1\tilde{F}_1(a,a-\alpha;-bt)\ ,\quad(13)$$

which obviates having to use a $\Re[\Gamma(-\alpha)]$ computer command to truncate a zero magnitude imaginary machine number carry, e.g., $\Re(x+0\times i)=x$, such that Eq. (9) can be rewritten as

$$\mathrm{GPC}(t;a,b,\alpha,\beta)$$
$$=-\theta(t-\beta)\frac{\alpha\,b^a\beta^\alpha}{\Gamma(a)}t^{a-\alpha-1}\sum_{n=0}^{\infty}\frac{(-b\,t)^n}{n!}B_{\frac{\beta}{t}}(-\alpha,a+n)\qquad(14)$$
$$-\theta(t-\beta)\pi\csc(\pi\alpha)\frac{b^a\beta^\alpha}{\Gamma(\alpha)}t^{a-\alpha-1}{}_1\tilde{F}_1(a,a-\alpha;-b\,t)\ ,$$

where $\alpha\neq 0,1,2,3,\dots$, which is Eq. (25) of the first type I GPC publication [13]. Note that not only is the summation of the above absolutely convergent, but as the last line above is an asymptote for $t\to\infty$ of the GPC function [13], the summation converges to zero as $t\to\infty$ relatively more rapidly than the asymptote.

The summation terms,

$$-\frac{\alpha\,b^a\beta^\alpha}{\Gamma(a)}t^{a-\alpha-1}\sum_{n=0}^{\infty}\frac{(-b\,t)^n}{n!}B_{\frac{\beta}{t}}(-\alpha,a+n)\ ,$$

are rearranged for acceleration at long times using the infinite series definition of the incomplete beta function,[15]

$$B_{\frac{\beta}{t}}(-\alpha,a+n)=\left(\frac{\beta}{t}\right)^{-\alpha}\sum_{k=0}^{\infty}\frac{\left(\frac{\beta}{t}\right)^k(1-a-n)_k}{k!(k-\alpha)}\qquad(15)$$
$$\text{for }\left|\frac{\beta}{t}\right|<1\text{ and }\alpha\neq0,1,2,3,\dots,$$

by substituting it into the summation and simplifying to yield,

$$-\frac{\alpha\,b^a}{\Gamma(a)}t^{a-1}\sum_{n=0}^{\infty}\frac{(-b\,t)^n}{n!}\sum_{k=0}^{\infty}\frac{\left(\frac{\beta}{t}\right)^k(1-a-n)_k}{k!(k-\alpha)}\ .\qquad(16)$$

Given absolute convergence (Short-t GPC convergence Appendix Subsection) the order of infinite summation can be changed with impunity by distributing the outer sum over the inner sum, and factoring, as follows,

$$-\frac{\alpha\,b^a}{\Gamma(a)}t^{a-1}\sum_{k=0}^{\infty}\sum_{n=0}^{\infty}\frac{(-b\,t)^n}{n!}\frac{\left(\frac{\beta}{t}\right)^k(1-a-n)_k}{k!(k-\alpha)}\ ,$$
$$-\frac{\alpha\,b^a}{\Gamma(a)}t^{a-1}\sum_{k=0}^{\infty}\frac{\left(\frac{\beta}{t}\right)^k}{k!(k-\alpha)}\sum_{n=0}^{\infty}\frac{(-b\,t)^n(1-a-n)_k}{n!}\ .\qquad(17)$$

Fortunately, the inner sum in the above formula simplifies to a closed form, allowing it to be rewritten as

15 http://functions.wolfram.com/06.19.06.0002.01.

$$-\frac{\alpha\,b^a}{\Gamma(a)}t^{a-1}\sum_{k=0}^{\infty}\left(\frac{\beta}{t}\right)^k\frac{(1-a)_k}{k!(k-\alpha)}{}_1F_1(a,a-k;-b\,t)\ .\qquad(18)$$

The $k=0$ term of that sum simplifies to be the gamma distribution function part of the GPC convolution. Splitting off that term and adjusting the lower summation index from $k=0$ to $k=1$ yields,

$$\frac{b^a}{\Gamma(a)}e^{-b\,t}t^{a-1}$$
$$-\frac{\alpha\,b^a}{\Gamma(a)}t^{a-1}\sum_{k=1}^{\infty}\left(\frac{\beta}{t}\right)^k\frac{(1-a)_k}{k!(k-\alpha)}{}_1F_1(a,a-k;-b\,t)\qquad(19)$$

Next, the quickly convergent sum term, Eq. (19), is added to the gamma distribution plus asymptotic formula Eq. (13) to create a series accelerated algorithm rewrite of Eq. (5) for long t-values,

$$\mathrm{GPC}(t;a,b,\alpha,\beta)$$
$$=-\theta(t-\beta)\frac{\alpha\,b^a}{\Gamma(a)}t^{a-1}\sum_{k=1}^{\infty}\left(\frac{\beta}{t}\right)^k\frac{(1-a)_k}{k!(k-\alpha)}{}_1F_1(a,a-k;-b\,t)$$
$$+\theta(t-\beta)\left[\frac{b^a}{\Gamma(a)}e^{-b\,t}t^{a-1}\right.$$
$$\left.-\pi\csc(\pi\alpha)\frac{b^a\beta^\alpha}{\Gamma(\alpha)}t^{a-\alpha-1}{}_1\tilde{F}_1(a,a-\alpha;-bt)\right].$$

This is identically Eq. (10), which completes the proof of the long-t theorem. $\square$

The term beginning with $+\theta(t-\beta)$ of the above equation is an asymptote of the GPC function. The above equation's first line when written as a list of terms to be summed has all negative elements when $k>\alpha$, which was the case for metformin [13]. If $k<\alpha$ for the first few k, then the simplified summation terms are initially positive until $k>\alpha$, but in any case the magnitude of those terms is strictly monotonically decreasing such that increasing precision to sum those terms is unnecessary. The confluent hypergeometric functions in those terms and their effects on convergence are presented in detail in the Long-t GPC convergence rapidity Appendix Subsection, which shows that the absolute value of the ratio of the $(k+1)$th to kth terms is approximately $\frac{\beta}{k\,t}$, where the k in the denominator insures that the absolute values of the simplified terms of the summand for the above formula are monotonically decreasing, and that each $(k+1)^{\text{st}}$ term is many times closer to zero than the kth term, such that it is unnecessary to test for convergence using the sum to infinity of all the remainder terms, i.e., in practice it is sufficient to test the absolute value of the last term and to stop the summation when that magnitude is less than the desired precision (e.g., $<10^{-65}$).

Other long-t functions; the integrals and derivative

GPC type I long-t integral: The derivation of a similarly accelerated series for $t \geq 4\beta$ of the CDF of GPC, i.e., its 0 to t integral, GPC_F, follows from its primary definition, Eq. (6), using the same procedure as Eqs. (9) to (10), leading to,

$$GPC_F(t; a, b, \alpha, \beta)$$

$$= -\theta(t - \beta) \frac{\alpha b^a}{\Gamma(1 + a)} t^a$$

$$\sum_{k=1}^{\infty} \left(\frac{\beta}{t}\right)^k \frac{(-a)_k}{k!(k - \alpha)} {}_1F_1(a, a - k + 1; -b t)$$

$$+ \theta(t - \beta) \left[1 - Q(a, b t) \right.$$

$$\left. - \pi \csc(\pi \alpha) \frac{b^a \beta^\alpha}{\Gamma(\alpha)} t^{a - \alpha} {}_1\tilde{F}_1(a, a - \alpha + 1; -b t) \right],$$

$$(20)$$

where $Q(a, b t) = \frac{\Gamma(a, b t)}{\Gamma(a)}$ is the regularised upper incomplete gamma function, and is the complementary cumulative density function (CCDF = 1−CDF) of the gamma distribution.[16] Note that GPC_F is a CDF, such that the upper limit of Eq. (20) as t increases is 1 or 100% of initial dose eliminated from the body.

GPC type I long-t double integral: Similarly, the super cumulative distribution, i.e., the integral from $\tau = 0$ to t of the CDF is,

$$GPC_{\mathcal{F}}(t; a, b, \alpha, \beta)$$

$$= -\theta(t - \beta) \frac{\alpha b^a t^{a+1}}{\Gamma(a + 1)}$$

$$\sum_{k=2}^{\infty} \left(\frac{\beta}{t}\right)^k \frac{(-a)_k}{k!(k - \alpha)(a - k + 1)} {}_1F_1(a, a - k + 2; -b t)$$

$$+ \theta(t - \beta) \left\{ \frac{t e^{-b t} (b t)^a}{\Gamma(a + 1)} - \frac{\alpha \beta}{\alpha - 1} [1 - Q(a, b t)] \right.$$

$$+ \left(t - \frac{a}{b}\right) [1 - Q(a + 1, b t)]$$

$$\left. - \pi \csc(\pi \alpha) \frac{b^a \beta^\alpha}{\Gamma(\alpha)} t^{a - \alpha + 1} {}_1\tilde{F}_1(a, a - \alpha + 2; -b t) \right\}.$$

$$(21)$$

Note that the sum term is now indexed from $k = 2$, for which each simplified summation element has a negative value when $k > \alpha$, and a multiplied out positive first term when $\alpha < 2$.

---

[16] CCDF is sometimes loosely referred to as a survival function, $S(t)$.

GPC type I long-t derivative: The GPC derivative's algorithm for $t > 4\beta$, i.e., long-t, is

$$GPC'(t; a, b, \alpha, \beta)$$

$$= \theta(t - \beta) b^a t^{a-2} \left\{ \frac{a - b t - 1}{\Gamma(a)} e^{-b t} \right.$$

$$+ \frac{\pi \alpha \csc(\pi \alpha) \left(\frac{\beta}{t}\right)^\alpha}{\Gamma(\alpha + 1)} \left[(\alpha + 1) {}_1\tilde{F}_1(a; a - \alpha; -b t) \right.$$

$$\left. - a {}_1\tilde{F}_1(a + 1; a - \alpha; -b t) \right]$$

$$+ \frac{\alpha}{\Gamma(a)} \sum_{k=1}^{\infty} \frac{(1 - a)_k \left(\frac{\beta}{t}\right)^k}{k!(k - \alpha)} \left[b t {}_1F_1(a; a - k; -b t) \right.$$

$$\left. \left. - (a - k - 1) {}_1F_1(a - 1; a - k - 1; -b t) \right] \right\}.$$

$$(22)$$

The combined short- and long-t algorithm for GPC series acceleration

There are now two algorithms, an algorithm that converges quickly only for short t-values, and another that converges quickly only when t-values are long. This section describes how the algorithms are combined to produce a new accelerated algorithm for any value of t. A full set of functions for the derivative and integrals of the GPC algorithm follows the same pattern as the Mathematica source code of the GPC type I accelerated algorithm Appendix Subsection. The two algorithms are combined by choosing $t = 4\beta$ as the floor (least) value for use of the long-t algorithm, makes the next term at worst approximately 1/4 that of the current term. Given a next term fraction of $\frac{\beta}{k t}$ times the current term, the $t = 4\beta$ floor value is not critical, the trick is to avoid second to first term ratios that initially approach 1 as $t \to \beta$, for which the short-t algorithm has fewer terms and converges faster. See the Choosing when to use the short-t and long-t algorithms Appendix Subsection for further information (Fig. 1).

The program uses so-called infinite magnitude numbers such that numbers like $\pm 10^{\pm 100000}$ can be used without overflow or underflow (code: \$MaxExtraPrecision = ∞). However, there is another concern; precision. Machine precision was 53 bits, or approximately 16 significant figures. When 10^{-100} and 1 are added, one has to have a precision of 100 significant figures to avoid truncation. For the short-t algorithm the extended precision needed is precalculated using machine precision of large numbers, which are stored as simplified terms, and are searched to

find the largest magnitude number (code: Ordering[storage,−1][[1]] − 1). It is then that number as a rounded base 10 exponent (code: Round[Log10[Abs[outmax]]]) plus 65 significant figures that is used as the required precision of the computation. The terms of the summand are then recalculated to that high precision, then summed, such that the result has approximately 65 significant figures remaining even though the calculation itself may have needed a thousand or more significant figures to yield that result. The same approach could be used to calculate $\sin(x)$ from its infinite series definition. As mentioned above, in practice that is not used, and instead the equivalent principal sine values of x are computed. For the GPC(t) computation, one can invert the range related extra precision problem by reordering the series to make it increasingly less demanding to calculate long-t values by direct application of Eq. (10) and that is precisely what the long-t GPC type I algorithms does. The value $t = 4\beta$ is used to transition between shorter t-values for use by the short-t algorithm, and longer t-values for use with the long-t algorithm. As mentioned, that time of transition between long and short algorithms is not critical and is more formally presented in the Choosing when to use the short-t and long-t algorithms Appendix Subsection.

Results

This Results section shows examples for GPC algorithm run times and diagnostics, of how it can and should be used including the use for extended same dose multidosing, and a subsection illustrating confidence interval (CI) and coefficient of variation (CV) diagnostic quality assurance.

Algorithm run time analysis

The GPC combined short- and long-t algorithm was defined in terms of how to calculate it efficiently, as above. Implementation of the combined short and long time algorithm using Mathematica 12.3 without parallel processing on a 2.3 GHz 8-Core Intel i9 processor allows long t-value execution times of around 1.2 millisecond with typical 63 to 67 decimal place accuracy. (The full range of run times is approximately from 42 to 1.2 milliseconds for t-values ranging 30 s to 1/2 year.) This contrasts with the short-t implementation of GPC Eq. (9), which, as t increases, needs more terms and higher precision to maintain a given precision of the final result, with a processing time that progressively becomes intractably long. Figure 2 shows the relative performance of the algorithms in these respects using the GPC parameters from fitting metformin data for dog 1 [13]. This dog showed the median regression error of 8.7% of the seven studied. Despite having the fastest

elimination at 72 h, the concentration level for that dog was predicted to be 2×10^{-7} of peak at one year, a small number but much larger than could be produced assuming a terminal exponential tail. For the short-t algorithm the run-time to calculate concentration at one-half year following injection was 1809 s, versus 1.2 milliseconds for the new algorithm. This difference is because the short-t algorithm used at long times had 8883 terms to sum, and the call to **gpcshort** was used twice; once at machine precision to find the maximum absolute value term $(1.0796 * 10^{1392})$ of all of the summand terms in order to calculate that 1457 place precision was required to obtain 65 place precision in the output, and once again to do the 1457 place summation arithmetic. For the combined (new) algorithm this is not needed as for short times the short-t algorithm does not have large oscillating terms, and the long-t algorithm has monotonically decreasing term magnitude both for each sequentially summed term, and as t increases, for each first term magnitude. For example, the first (and only) term of the long-t algorithm's summand at one-half year was negligible $(-1.851 * 10^{-1403})$. These effects are illustrated in Fig. 3.

For our test case example, the two algorithms, short-t and long-t, agreed to within 63 to 67 decimal places. In practice, the short-t algorithm is used for short times and the long-t algorithm is used for long times. It makes little difference what cut-point time between short- and long-t algorithms is used, and the time 4β, albeit around 100–120 s, was chosen as a division point between algorithms short enough to ensure that extra precision padding for the short-t algorithm would be unnecessary.

Regression processing elapsed times and extended multidosing

For evaluating the 72 h data for seven dogs, the new, combined short- and long-t algorithm run time for curve fitting was approximately 1:15 to 3:00 (min:s) average values, the program prior version with hardware and software accelerations for the short-t algorithm and without sufficiently extended precision (despite using at least 65 place arithmetic) had run times in the approximate range of 34 to 35 min, but with occasional errors in parameter values of $\pm 2 \times 10^{-14}$. With proper precision extension the error dropped below 10^{-20} for all 5 parameters and 7 cases, but the run time increased to 50 min, using a partly accelerated short-t algorithm (Eq. (14)) and 8-core hardware acceleration. The current combined short- and long-t algorithm does not use those additional accelerations. Forty model-based bootstrap cases generated for the first dog's data—see next Subsection—took 49:45 (min:s), or 1:15 per case. That is a lot faster than the 33:51 per case it took to generate 35 bootstrap models using the old software
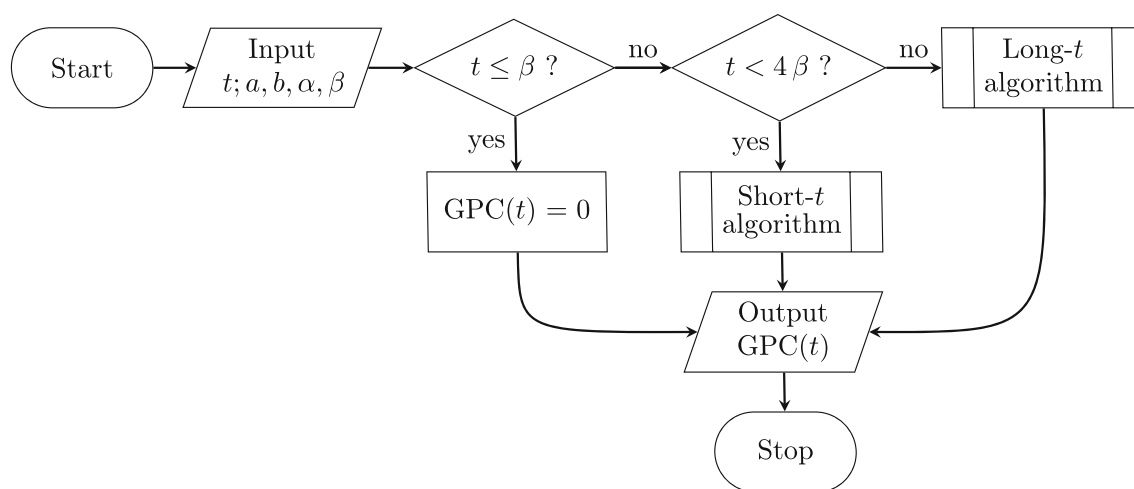
Fig. 1 Standard flow chart for the Mathematica source code of the GPC type I accelerated algorithm Appendix Subsection. The predefined short- and long-t routines are also described in the text

(19:44:55). Overall, the run time is very approximately 27 times faster than prior, but is variable depending on the problem being solved, the computer hardware used, and the other software running on the computer at that time. For example, Fig. 4a, with a current run time of 7.1 s, could not be computed at all using the earlier software version.

Notice that if we wish to glean information during metformin multidosing with plasma or serum sampling, the best time to do so is just prior to the next scheduled dosing as those concentrations change for each dose interval, whereas the peak concentration change over time is very small. However, because the tissue dosage[17] accumulates, the amount of drug in the body (Fig. 4b) cannot be predicted from serum (or plasma) concentration alone. Note that approximately one entire dose has accumulated in tissue by 14 days despite most of the cumulative dosage having been eliminated over that time. That is, during the first dose interval, the mean drug mass remaining in the body was 0.175 doses, and during the 14th dose interval the mean drug mass remaining in the body was 1.118 doses, where 12.88 dose masses were eliminated.

Which are better, confidence intervals or coefficients of variation?

With reference to Table 2, confidence intervals (CI) of the mean were extracted from model-based bootstrap with 40 cases generated for the first dog's data. For calculating CI's of the mean, the Student's-t method was used (Verified assumptions: Central Limit Theorem, $n > 30$, light-tailed distributions). However, as a result of extensive testing the degrees of freedom were set at n rather than the more typical $n − 1$, as it was found that for smaller values of n,

physically impossible results were obtained, whereas even for $n = 2$, when n was used, rather than $n − 1$, the results were accurate. For $n = 40$ it made very little difference whether $n − 1$ or n were used. Also shown are CI's of the model based bootstrap (A.K.A., parametric bootstrap) results calculated directly from the $n = 40$ data using the nonparametric quantile (A.K.A, percentile) method of Weibull [24].[18] Note that the Pareto rate parameter, $β$ was not presented. Since many (38 of 40) of the results were at the constraint boundaries of 25 to 30 s, one already knows what the confidence interval largely is; the constraint values themselves. Another situation entirely exists for coefficients of variation (CV). Note in the table that when $n = 5$ as during the prior study, that the values so obtained were too small. It is theoretically possible to use bootstrap (in our case that would be bootstrap of model-based bootstrap) to obtain confidence interval quantiles for the median CV, and although median values of CV's have shown sufficient robustness to construct confidence intervals for n sufficiently large [25], the correction for n-small is problematic as per Table 2 and the Discussion Section that follows.

Discussion

Wise [26] first proposed that power functions or gamma distributions should be used in pharmacokinetic modelling as superior alternatives to sums of exponential terms. This has been reinforced more recently, for example by Macheras [27]. While convolution models and fractal

---

[17] For a single dose, body drug mass is $M(t) =$ Dose $[1 − GPC_F(t)]$.

[18] This uses the Weibull method for extracting confidence intervals, which in Microsoft Excel (2007) would format for the lower tail as PERCENTILE.EXC(A1:A40,0.025) and from Mathematica 12.3 [5] as Quantile[data, 0.025, {{0, 1}, {0, 1}}], https://mathworld.wolfram.com/Quantile.html.
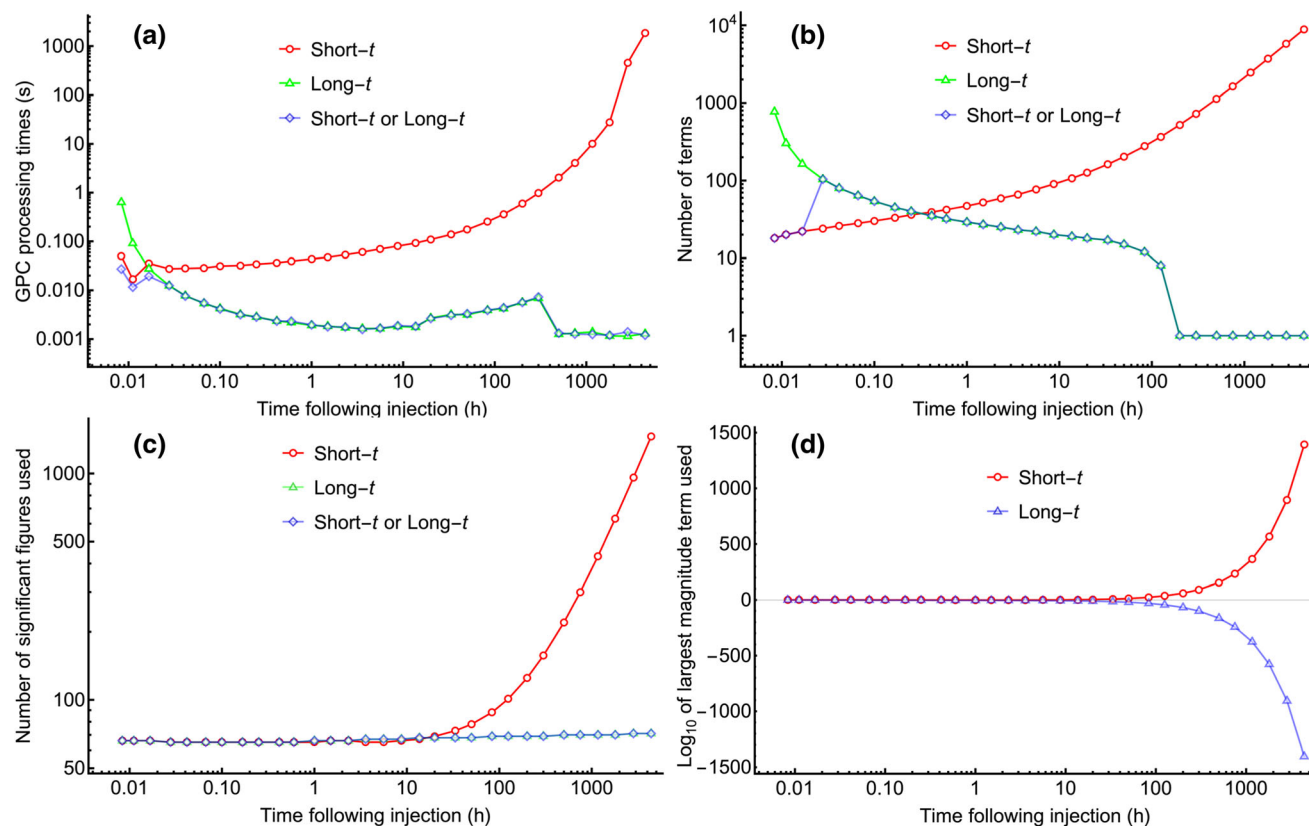
Fig. 2 Log–log plots comparing the performance of the short time (red connected open circles), the long time algorithm (green connected open triangles) and the either short or long time (blue connected open diamonds) algorithms' performance for the GPC model of metformin disposition in dog 1 of reference [13]. Panel **a** shows that the long or short-t algorithm is more than one million times faster than the short-t algorithm when the time after injection is very long, e.g., for predicting a serum concentration at one half year (4396 h), and more than 20 times faster than the long-t algorithm for $t = 30$ s. Panel **b** shows the number of summed terms (n) for each method. Note that for long t-values, the combined algorithm used the long-t method and only calculates one term, whereas the short-t algorithm would have used many terms. Panel **c** shows the precision needed to accommodate the largest of the terms for each algorithm. As t increases, the short-t function uses an increasing number of significant figures, but for the long-t or combined algorithms that number increases only slightly to preserve accuracy for lower concentrations at longer times. Panel **d** shows the largest term magnitude as powers of 10 for the short- and long-t algorithms. For long times, the short-t algorithm alternating sign intermediate terms reach quite large magnitude, while for the long-t algorithm the largest magnitude term collapses to vanishingly small values. (Color figure online)

consistent models have been shown to be superior models in some cases and find occasional use [10, 12, 13, 28, 29] compartmental modelling software is widely available and is used by default. For example, compared to biexponential clearance evaluation of 412 human studies using a glomerular filtration rate agent, adaptively regularised gamma distribution (Tk-GV method [30, 31]) testing was able to reduce sampling from 8 to 4 h and from nine to four samples for a more precise and more accurate, yet more easily tolerated and simplified clearance test [29]. Despite this, few institutions have implemented the Tk-GV method at present. In the case of metformin, a highly polar ionised base, the extensive, obligatory active transport of the drug into tissue produces a rate of expansion of the apparent volume of distribution having the same units as renal clearance, yielding the Pareto (power function) tail. This

heavy tail, and Fig. 4, help to explain why metformin control of plasma glucose levels had delayed onset, e.g., following 4-weeks of oral dosing [32], and provides hints concerning the lack of a direct correlation between drug effect and blood metformin concentrations [33]. Other basic drugs whose active transport dominates their disposition may show similar behaviour. The long tail in the disposition of amiodarone may be a reflection of its very high lipid solubility rather than, or in association with, active tissue uptake. Weiss [34] described its kinetics after a 10 min intravenous infusion with an s-space Laplace transform convolution of a monoexponential cumulative distribution with an inverse Gaussian distribution and a Pareto type I density, which lacked a real or t-space inverse transform such that the modelling information had to be extracted numerically. A real space $f(t)$ model convolution
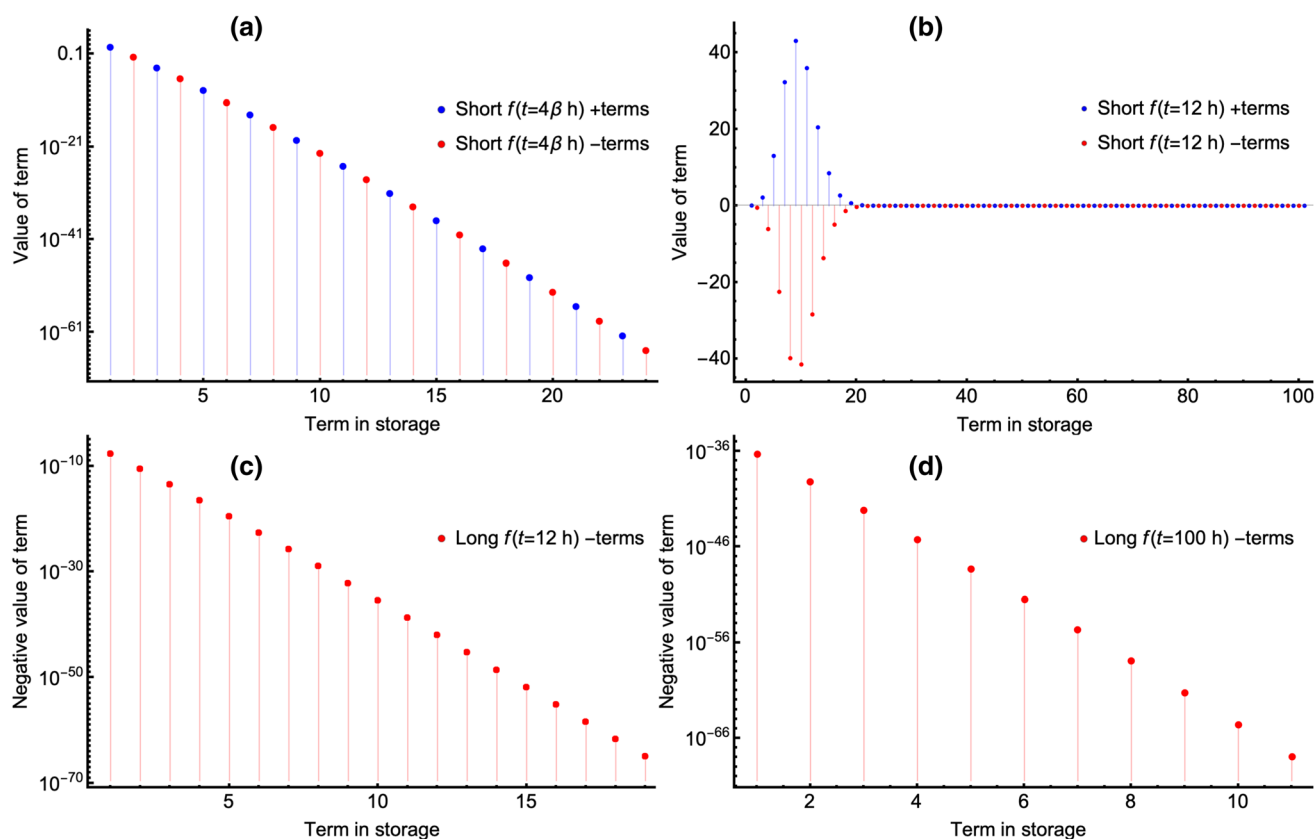
Fig. 3 Individual terms of the sums for the short-t and long-t algorithms for the GPC model of metformin disposition in dog 1 of reference [13]. These terms are stored in a list variable called, somewhat unimaginatively, *storage*. Note that $f(t = 4\beta\,h)$, is an explicit replacement rule meaning that *an $f(t)$ is evaluated $f(4\beta)$, where $t = 4\beta\ h$*. Panels **a** and **b** show the values of the short-t algorithm's summand terms. Panel **a** shows the values for the short-t algorithm at the upper limit of t of its usage in the combined, new, algorithm at $t = 4\beta$ (100 s in this case). The blue dots are positive values, and the red dots are negative values of the summand of Eq. (5). Panel **b** shows what happens when the short-t algorithm is used at 12 h. That is the oscillatory terms would have intermediate values that grow in magnitude before they converge. While this poses

little problem at 12 h, this time is not used in the new, combined algorithm and in the extreme the oscillatory intermediate terms of the summand grow to very large magnitude as t-values increase. Thus, before the summand is calculated for long t-values when using the short-t algorithm, preliminary calculation of the required extended precision is necessary, which prologues execution time markedly. Panel **c** shows the region of values for the long-t algorithm at 12 h. Note that there are fewer terms than for the short-t algorithm at that same time (panel **b**), that all terms are negative (red), and that they decrease by one or more orders of magnitude between successive terms. Panel **d** shows that by 100 h for the long-t algorithm, there are fewer terms than at 12 h, and that their magnitude is very small even for the largest magnitude, first, term. (Color figure online)

of time-limited infusion effects of a GPC type I distribution is simple to construct and would be the same as Weiss's model in the one essential aspect that matters; testing of the amiodarone power function tail hypothesis, for which a GPC derived model would have the advantage of being more transparently inspectable. Similarly, Claret et al. [35] used finite time difference power functions to investigate cyclosporin kinetics for which GPC and related model testing may be appropriate.

We were able to use Nelder–Mead global search regression model-based bootstrap to provide more information and better information for parameter variability than would be available from a gradient matrix. Some readers would prefer to use the Levenberg–Marquardt algorithm convex gradient regression method, so that the

gradients can be used to estimate case-wise coefficients of variation. The logarithm of sums of exponential terms is always convex. The GPC-metformin loss function is non-convex, as shown by failure of the interior point method to improve on solutions as reported in the Data sources and regression Methods Subsection. Constrained nonconvex gradient methods are comparatively rarely implemented; there appears to be no such implementation in Mathematica at present.

Correction of standard deviation (SD) for small numbers ($n < 30$) using bootstrap of model-based bootstrap and χ^2 were used as mentioned elsewhere [36], and led to using n rather than $n - 1$ for Student's-t degrees of freedom. Whereas variance is unbiased, when the square-root of variance is taken, the result, standard deviation becomes
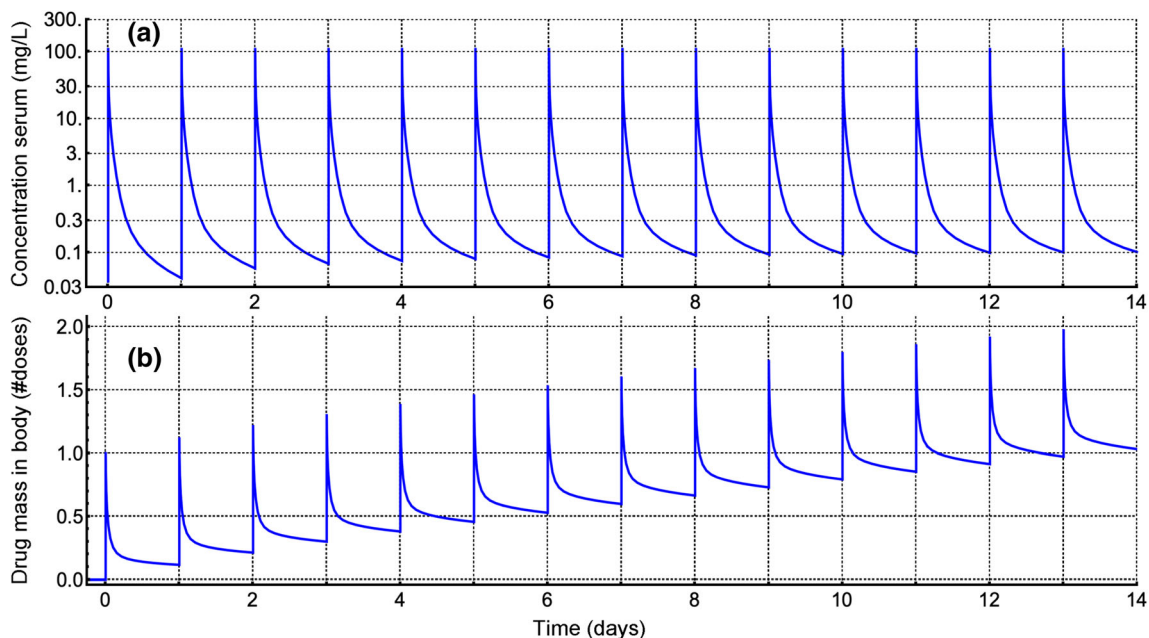
Fig. 4 Multidosing of the GPC function for dog 1 with one 18.248 mg/kg body weight dose every 24 h. Panel **a** shows the predicted concentration curve. Note that the peak concentrations did not increase much following 14 doses (0.089%), but that the trough values increased rather more substantially (2.48 times) For panel **b** showing number of doses retained in the body, one first sees a 1 dose peak increasing to a 1.97 dose peak for the 14th dose, whereas the trough initially at 0.117 doses, increased 8.85 fold to finish at a 1.03 dose trough

Table 2 Example parameter results and quality assurance are shown for dog 1

	Primary result	95% CI bootstrap	Mean	95% CI of mean	SD	CV%	Prior CV%
n	1	40	40	40	40	40	5
rrms %	8.71	4.72 to 9.88	7.52	7.16 to 7.89	1.15	15.3	–
R^2	0.99872	0.99773 to 0.99950	0.99872	0.99860 to 0.99884	0.00038	0.039	–
a	0.349	0.198 to 0.516	0.350	0.324 to 0.376	0.0802	22.9	8.09
b	0.732	0.558 to 0.889	0.735	0.708 to 0.761	0.0821	11.2	4.84
α	0.264	0.235 to 0.322	0.268	0.261 to 0.275	0.021	7.85	5.13
AUC	31.2	26.8 to 38.0	31.3	30.4 to 32.2	2.77	8.84	6.22
CL	9.76	8.01 to 11.3	9.78	9.52 to 10.05	0.829	8.47	6.42

The relative root mean square error of fit (rrms %) and R^2 values are acceptable. Confidence intervals (CI) are shown for the parameter values and the mean parameter values

biased. Arising from χ^2, a standard deviation from only two samples, is on average only 79.8%, $\sqrt{\frac{2}{\pi}}$, of the population standard deviation [24].[19] Gradient methods lack *pre hoc* testing of the implicit assumption of residual normality and do not *post hoc* provide any parameter distribution

information. From the trace of the gradient matrix, one obtains a standard deviation with degrees of freedom that are $n - p - 1$ (*n*-samples, *p* parameters) [36]. For standard deviations in the case where $n - p - 1$ is small, the corrections for standard deviation are large. Overall, the ratio between gradient based error propagation results and that from bootstrap is not unusually a factor of two larger or smaller [37]. Moreover, average fit errors using any loss function $> 10\%$, for assay methods with errors $< 10\%$ may suggest that the algorithm/data combination is suspect [10, 13, 22, 38], and for the metformin dog data that is the case for two- and three-compartment models, but not for

---

[19] Given only two samples, the population mean is not located midway between them, however, the midpoint (mean) is used to estimate the population mean in the standard deviation formula. The correction formula multiplier for an unbiased estimator ($\hat{\sigma}$) of population standard deviation (σ) from sample standard deviation (s) is $\hat{\sigma} = c_n s$, where $c_n = \sqrt{\frac{n-1}{2}} \Gamma\left(\frac{n-1}{2}\right) \Gamma\left(\frac{n}{2}\right)^{-1}$ [24].

the GPC model, which latter model was the only one to fit the data better than 10% (average 8.6% rrms with assay error of 5.2% rrms), as well as being the only model to exhibit normality and homoscedasticity of residuals [13]. When the fit error is >10%, one should, at a minimum, test residuals for homoscedasticity and normality, and if these are not present, a better fit model should be sought for its own sake, and bootstraping becomes problematic [22].

The use of coefficients of variation is sometimes problematic. Suppose that we have lots of data, but because $CV = SD/mean$, if by chance in a particular case especially if we have small n, some of the multiply generated mean values may approach zero, which injects some erratically high CV-values into a distribution of values. It is for that reason, numerically instability, that the more data one has, the worse the mean CV-value can be, with the solution being to first calculate many CV values, and then take their median value [25]. Even though the mean value may be not useful, the median may be, and confidence intervals for CV could be established using bootstrap quantiles, but not by using the gradient matrix approach because correction for n-small is problematic. That is, for mean values that can be rewritten as being proportional and having an established maximum range, e,g., Likert scale minus 1 variables, correcting CV underestimation for small values of n is possible. However, if, as is the case here, there is no theoretical maximum CV, one needs to invent a correction based upon the observed confidence intervals of the mean [39], such that CI-values are unavoidable for determining the meaning of the preponderance of CV results. Finally, comparison for significant differences between parameters for one subject versus another are easy to construct using CI, but more difficult to obtain for CV. Thus, CV-values cited without explicit quality assurance should be regarded as qualitative results.

Limitations

A major deficiency of the first article that applied and compared the gamma-Pareto type I convolution (GPC) model to other models [13] was the lack of an algorithm that could be used for independent investigation and/or for application to other drugs. The accelerated algorithm presented herein is the first publication of code for a gamma-Pareto type I convolution (GPC). As such, the algorithm was kept in a simple form without using all possible acceleration tools or stopping conditions. While it could be optimised for even shorter run-times using vector addition of subsets of the summands, by using Eq. (14) to reduce summand magnitudes for the short-t algorithm and/or combining partial sums of the summands for the short- or long-t algorithms, by eliminating diagnostic parameters such as run-time calculations, by compiling it, and by

multiple other means. However, that would be at the expense of clarity and/or simplicity of presentation. It is complicated to compute the values of functions like the $sin(x)$ efficiently. For example, an answer with precalculated exact precision can be quickly generated for $sin x$ using the CORDIC procedure, which is optimised for binary register operations at the machine level [40]. At a much higher and slower level, compared to the GPC(t) short-t algorithm, the $sin(t)$ function's series expansion has even larger magnitude terms for long t-values. In its current form, the combined short- and long-t GPC algorithm is so much faster than the previously published run times using the seven dogs 72 h data and more generally valid that it is now a practical algorithm. The current implementation is no longer limited as to how long t is, and the propagated error of up to 2×10^{-14} for parameter values obtained from regression of 72 h data has been reduced to $< 10^{-20}$. That error demonstrates the major extent to which errors from 65 decimal place precision can propagate during processing of tens of thousands of calculations, especially during regression, which typically, by default, halves the number of significant figures—see the Data sources and regression Methods Subsection. This does not affect any of the parameter values listed in Table 1, but the ability to quickly calculate a larger number of model-based bootstrap results would improve the parameter CI estimates. Another consideration is how to obtain exactly n significant figures precision when n are requested. Currently, for 65 significant figures requested, a result precise to several significant figures greater or lesser than 65 is returned and the algorithm is written only for 65 significant figure precision. Generalising this to request to obtain an arbitrary specific precision for a GPC functional height awaits the next several generations of algorithmic refinement.

Conclusions

The new GPC type I algorithm consists of two parts, one for very early times, and another for late times. At times less than approximately 4β, i.e., 100-120 s for the metformin data, the short-t algorithm is actually faster than the long-t algorithm. For early data, the short-t algorithm has alternating sign terms of monotonically decreasing magnitude. However, when used at long times, the short-t GPC algorithm required precalculation of the precision needed for later summation, which represents an improvement over the algorithm previously used [13]. In the newly-proposed, combined short and long-t algorithm this precalculation is unnecessary because of the long-t algorithm usage for all but the shortest t-values, resulting in markedly

accelerated convergence, and the new ability to predict concentration at any time, no matter how long.

Appendix

This section provides information concerning convergence of the short- and long-t algorithms, when they should be used, and how to encode them in the Mathematica [5] language.

Short-t GPC convergence

The short-t algorithm is an alternating series sum. For alternating series one can distinguish two types of convergence. Conditional convergence in which the value of the infinite sum depends on the order in which summation is performed as shown by the Riemann rearrangement theorem, and absolute convergence for which any order, or permutation, of summation process yields the same, unique, sum. Convergence is defined as conditional when an alternating series converges but its absolute value does not [41]. For example, the alternating harmonic series $\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n}$ has an absolute value ratio of next term to current term of $\frac{n}{n+1}$, whose limit as $n \to \infty$ is 1. That means that as n increases, the next term approaches the same size as the nth term, such that the absolute sum of terms is not bounded above, and the order of addition of the original series determines what the total sum is, making changes in order of summation yield different, i.e., ambiguous, results. If a limiting term ratio is less than 1, for example $\frac{1}{2}$, the series is absolutely convergent, e.g., the limiting infinite sum ratio of $\frac{1}{2}$ for some eventual term is, in binary arithmetic, $0.111111\ldots_2 \to 1_2 = 1$. It is fair to call series, whose limiting absolute value term ratio is 0, eventually-rapidly convergent. In the case of the short-t algorithm the infinite sum of its absolute values is, for sufficiently large values of t, a very large number. However, for any real valued time, t, no matter how large, there is a real number, M, that is greater than the magnitude of the infinite sum of absolute values of Eq. (9). That M can be fantastically large, but never infinite, makes it difficult to use Eq. (9) without precision that is explicitly extended for the purpose of accurately forming the infinite sum for certain values of the parameters and long times, but it does not make convergence conditional.

Lemma 1 *In the case of the short-t GPC algorithm, convergence is absolute and eventually rapid, such that the Riemann rearrangement theorem prohibition for resequencing infinite sums does not apply.*

Proof To show absolute convergence of the short-t GPC type I algorithm, we construct the absolute value of the ratio of the $(n + 1)$th to nth term. First, we take the infinite series definition of the incomplete beta function,[20]

$$B_z(A, B) = z^A \sum_{j=0}^{\infty} \frac{z^j (1 - B)_j}{j!(A + j)}$$
$$= z^A \left[\frac{1}{A} - \frac{B - 1}{A + 1} z + \frac{(B - 1)(B - 2)}{2!(A + 2)} z^2 - \cdots \right];$$
$$|z| < 1 \wedge \neg(-a \in \mathbb{Z} \wedge -a \geq 0) .$$

Although this is an alternating sign series with a restricted range of convergence, we term by term, *without permutation*, substitute into it the incomplete beta function parameters of Eq. (9)'s nth and $(n + 1)$th terms; $B_{1-\frac{\beta}{t}}(a + n, -\alpha)$, and $B_{1-\frac{\beta}{t}}(a + n + 1, -\alpha)$, and substitute that into the absolute value of the $(n + 1)$th to nth term ratio of the summand of Eq. (9), and simplify to yield,

$$\frac{b(t - \beta)}{n + 1} \frac{\frac{1}{a+n+1} + \frac{(\alpha+1)}{a+n+2}\left(1 - \frac{\beta}{t}\right) + \frac{(\alpha+1)(\alpha+2)}{2!(a+n+3)}\left(1 - \frac{\beta}{t}\right)^2 + \cdots}{\frac{1}{a+n} + \frac{(\alpha+1)}{a+n+1}\left(1 - \frac{\beta}{t}\right) + \frac{(\alpha+1)(\alpha+2)}{2!(a+n+2)}\left(1 - \frac{\beta}{t}\right)^2 + \cdots}$$
$$< \frac{b(t - \beta)}{n + 1} .$$

As $\infty > t > \beta$, neither the infinite series numerator or denominator is alternating, their ratio is absolutely convergent as $\frac{b(t-\beta)}{n+1}$ is an asymptote of, and upper bound for, the ratio of consecutive absolute value terms as $n \to \infty$. While $b(t - \beta) > n + 1$, if that occurs, for example for long t-values, we would expect the magnitude of the terms of the summands to increase for n small enough, but as n increases $b(t - \beta) \ll n + 1$ eventually, and the $(n + 1)$th relative term magnitude can be made as asymptotically close to zero as desired, and convergent by the ratio test [23]. $\square$

Thus, the magnitude of alternating terms is eventually monotonically decreasing such that the absolute error of summation from truncating at an nth term for n sufficiently large is less than the magnitude of the $(n + 1)$th term by the alternating series remainder theorem. Moreover, the first term of the summand is some definite positive real number proportional to 1. Setting the first term to be 1, we conclude that the sum of the absolute value of the summands of Eq. (9) is proportional to a number bounded above by

$$M \propto \sum_{k=0}^{\infty} \frac{[b(t - \beta)]^k}{k!} = e^{b(t-\beta)} ,$$
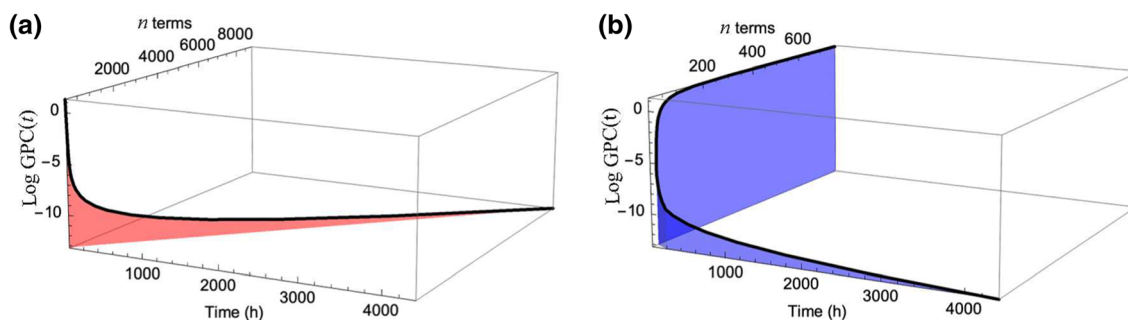
Fig. 5 The tradeoff between the time elapsed following bolus intravenous injection (x-axis), the number of terms to be summed for calculating the GPC function's value (y-axis), and the natural logarithm of the GPC function at that time (z-axis). Panel **a** shows short-t algorithm performance and panel **b** shows the long-t algorithm performance. Note that only for very early elapsed times does the short-t algorithm have fewer terms than the long-t algorithm

such that the sum of absolute values of summands of Eq. (9) is bounded above by some positive constant value times an exponential function of t, and Eq. (9) is absolutely convergent.

Long-t GPC algorithm convergence rapidity

This subsection examines the rapidity of convergence of the long-t GPC algorithm. In Lemma 1 directly above, it was shown that the short-t algorithm is absolutely convergent. Therefore, its infinite series rewrite as the long-t Theorem 1, Eq. (10), is also convergent but how many summation terms are needed for convergence and which parameters determine this convergence can be clarified using the substituted definition of the confluent hypergeometric series[21] as follows.

$$_1F_1(a; a-k; -bt) = \sum_{j=0}^{\infty} \frac{(a)_j(-bt)^j}{j!(a-k)_j}$$

$$= 1 - \frac{abt}{a-k} + \frac{a(a+1)(bt)^2}{2!(a-k)(a-k+1)} - \cdots .$$

Note that in the limit as $k \to \infty$ the above equation is asymptotically ($\sim$) 1. Next, the ratio of the $(k+1)$th to kth term is asymptotic to $\frac{\beta}{kt}$ for k sufficiently large,

$$\frac{\beta}{t} \frac{\alpha - k}{(k+1)(\alpha-k-1)} \frac{1 - \frac{a}{a-k-1}bt + \frac{a(a+1)}{2(a-k-1)(a-k)}(bt)^2 - \cdots}{1 - \frac{a}{a-k}bt + \frac{a(a+1)}{2(a-k)(a-k+1)}(bt)^2 - \cdots} \sim \frac{\beta}{kt} .$$

For that reason, for longer t-values, one can expect faster convergence of the long-t algorithm with fewer terms summed.

Choosing when to use the short-t and long-t algorithms

As above, the absolute value of the ratio of the next term to the current term for the short-t algorithm is bounded above by $\frac{b(t-\beta)}{n+1}$. For the long-t algorithm, the ratio of the $(k+1)$th to kth term approaches $\frac{\beta}{kt}$ for k sufficiently large. Note that these are in the opposite direction, that is, while t-values increase, $\frac{b(t-\beta)}{n+1}$ increases and $\frac{\beta}{kt}$ decreases. It is not critical exactly at what t value one elects to use the short- and long-t algorithms, as the major cost in computational time and number of terms needed occurs at the extreme values of t, but in an opposite direction for each algorithm. Figure 5 shows the tradeoff for dog 1 of the metformin series between numbers of terms for summation, time following bolus injection, and the magnitude of the natural logarithm of GPC(t), where GPC(t) = $\frac{C(t)}{AUC}$. Selecting $t = 4\beta$ as a cut point for switching between algorithms means that the short-t algorithm absolute sum of terms is bounded above, from substitution into $e^{b(t-\beta)}$, by $e^{3b\beta}$ times the first term's value, not a large number, and the long-t algorithm has an approximate maximum $(k+1)$th to kth term ratio of $\frac{1}{4k}$ for the shortest t-value used, which can still be made as small as desired for k sufficiently large.

Mathematica source code of the GPC type I accelerated algorithm

Computer implementation of the GPC type I function is provided as a notebook (nb file type) in the Mathematica language as Supplementary Materials 1. Without access to Mathematica itself that file cannot be easily reviewed. Therefore, an image of that file's contents is provided below.

[21] http://functions.wolfram.com/07.20.06.0002.01.

```mathematica
(*.....................The gpc[t] function a gamma-Pareto type I convolution fast calculation algorithm.........................*)
(*..........................................Copyright Carl A.Wesolowski, 2021..............................................*)
(* To fit disposition data, minimise the loss function using AUC×gpc[t] as the model, which returns AUC as a value. To use gpc[t]   *)
(* enter the constrained>0 coefficients a, b, α, and β to 65 decimal place accuracy.a and α are dimensionless, b is a rate and β is *)
(* a time. For example,the metformin dog 1 GPC type I parameters using b (h⁻¹) and β (h) are                                        *)
a = N[Rationalize[0.34931003807815571524792421542558602868248355919027496611955665616], 65];
b = N[Rationalize[0.73182479199387479660419087183394451163091958778927254273673996698], 65];
α = N[Rationalize[0.26437129139517680335740710070693267536710608361890151476103695922], 65];
β = N[Rationalize[0.00694444444444444444444444444444444444444444444444444444444444444444444], 67];
(* Note the explicit input precision as 65 digits, which must be specified as such for the algorithm to function properly. During   *)
(* regression analysis the parameters values above would change to minimise a loss function, the ones above were used as realistic  *)
(* values for the algorithm execution timing trial of the Results Section.*)
$MaxExtraPrecision = Infinity;
$MinPrecision = 0;  (*This default machine precision value is adjusted in the Block commands below to $MinPrecision=desirprec*)


(*........................................Section for short times, β < t < 4β, where 4β ≈ 100 to 120 s...............................*)
gpcshort[t_?NumericQ] := Quiet[Unevaluated[k = 0; storage =.; (*storage is an array for later summation*)

   conscale = (bᵃ α βᵅ)/Gamma[a]  tᵃ⁻ᵅ⁻¹;  (*This is the constant multiplier of the sum and is calculated only once*)

   target = 10⁻⁶⁵/conscale; storage = conscale First@Last@Reap@While[Abs[Sow[((-b t)ᵏ Beta[1 - β/t, a + k, -α])/k!  (*gpc function*)]] > target, k++];

   nn = Ordering[storage, -1][[1]] - 1; (*Finds nn, the index of the largest magnitude term in the storage array*)

   outmax = conscale ((-b t)ⁿⁿ Beta[1 - β/t, a + nn, -α])/nn!; (*This saves the maximum term*)

   lastn = Length[storage]; xprec = Round[Log10[Abs[outmax]]];

   If[xprec === Indeterminate, xprec = 0]; (*Exact times, e.g., t=e^(1/1000) for n=1 may need correction.*)desirprec = 65 + Abs[xprec]]];


(*..................................................t ≥ 4β section with asymptote for long values of t....................................*)
gpcsetup[t_?NumericQ] := Quiet[Unevaluated[k = 1; storage =.; conscale = (bᵃ t⁻¹⁺ᵃ α)/Gamma[a]; target = 10⁻⁶⁵/conscale;

   storage = -conscale First@Last@Reap@While[Abs[Sow[(β/t)ᵏ/((k - α) k!)  Hypergeometric1F1[a, a - k, -b t] Pochhammer[1 - a, k]]] > target, k++];

   nn = Ordering[Abs[storage], -1][[1]];

   outmax = -conscale (β/t)ⁿⁿ/((nn - α) nn!)  Hypergeometric1F1[a, a - nn, -b t] Pochhammer[1 - a, nn];

   lastn = Length[storage];

   asympt[z_?NumericQ] := N[(bᵃ e⁻ᵇ ᶻ z⁻¹⁺ᵃ)/Gamma[a] - (π bᵃ βᵅ α Csc[α π])/Gamma[1 + α]  z⁻¹⁺ᵃ⁻ᵅ Hypergeometric1F1Regularized[a, a - α, -b z], 70];
   SetAttributes[asympt, Listable];
   xprec = Round[Log10[Abs[outmax + asympt[t]]]];
   If[xprec === Indeterminate, xprec = 0]; desirprec = 65 + Abs[xprec]]];


(*.................................GPC combined short and long time function call (use this one)..........................*)
gpc[t_?NumericQ] := Quiet[.
   If[t ≤ β, 0, If[t < 4 β, Block[{$MinPrecision = desirprec, $MaxExtraPrecision = Infinity}, gpcshort[t]; Sum[storage[[j]], {j, 1, lastn}]],
      Block[{$MinPrecision = desirprec, $MaxExtraPrecision = Infinity}, gpcsetup[t]; asympt[t] + Sum[storage[[j]], {j, 1, lastn}]]]]];


(*..........Using the code above, the short time algorithm is............*)
gpcslow[t_?NumericQ] := Quiet[
   If[t ≤ β, 0, Block[{$MinPrecision = 0, $MaxExtraPrecision = Infinity}, gpcshort[t]];
      Block[{$MinPrecision = desirprec, $MaxExtraPrecision = Infinity}, gpcshort[t]; Sum[storage[[j]], {j, 1, lastn}]]]];


(*...........Using the code above, the long time algorithm is............*)
gpclong[t_?NumericQ] := Quiet[
   If[t ≤ β, 0,
      Block[{$MinPrecision = desirprec, $MaxExtraPrecision = Infinity}, gpcsetup[t]; Sum[storage[[j]], {j, 1, lastn} + asympt[t]]]]];
```

References

1. Nadarajah S, Kotz S (2007) On the convolution of Pareto and gamma distributions. Comput Netw 51(12):3650–3654. https://doi.org/10.1016/j.comnet.2007.03.003

2. Alzaatreh A, Famoye F, Lee C (2012) Gamma-Pareto distribution and its applications. J Mod App Stat Methods 11(1):7

3. Hanum H, Wigena AH, Djuraidah A, Mangku IW (2015) Modeling extreme rainfall with Gamma-Pareto distribution. Appl Math Sci 9(121):6029–6039. https://doi.org/10.12988/ams.2015.57489

4. Kotz S, Balakrishnan N, Johnson NL (2004). 52, Multivariate Pareto distributions, Section 2 In: Continuous multivariate distributions, Models and applications, vol 1. Wiley, pp 577–9

5. Wolfram Research, Inc (2021) Mathematica, version 12.3, Champaign, IL Version: 12.3 ed. Wolfram Research Champaign, IL. https://www.wolfram.com/mathematica

6. Bateman H (1910). In: The solution of a system of differential equations occurring in the theory of radioactive transformations, vol 15, pp 423–427. https://ia802809.us.archive.org/18/items/cbarchive_122715_solutionofasystemofdifferentia1843/solutionofasystemofdifferentia1843.pdf

7. Gladtke E (1988) History of pharmacokinetics. In: Pharmacokinetics. Springer, pp 1–9. https://rd.springer.com/chapter/10.1007/978-1-4684-5463-5_1

8. Gehlen W (1933) Wirkungsstärke intravenös verabreichter Arzneimittel als Zeitfunktion. Naunyn-Schmiedebergs Archiv für experimentelle Pathologie und Pharmakologie 171(1):541–554. https://doi.org/10.1007/BF01981291.pdf

9. Di Salvo F (2006) In: The exact distribution of the weighted convolution of two gamma distributions, pp 511–4. http://www.old.sis-statistica.org/files/pdf/atti/Spontanee2006_511-514.pdf

10. Wesolowski CA, Wanasundara SN, Wesolowski MJ, Erbas B, Babyn PS (2016) A gamma-distribution convolution model of ^{99m}Tc-MIBI thyroid time-activity curves. EJNMMI Phys 3(1):31

11. Widmark EMP (1919) Studies in the concentration of indifferent narcotics in blood and tissues. Acta Medica Scand 52(1):87–164. https://doi.org/10.1111/j.0954-6820.1919.tb08277.x

12. Wesolowski CA, Wesolowski MJ, Babyn PS, Wanasundara SN (2016) Time varying apparent volume of distribution and drug half-lives following intravenous bolus injections. PLoS ONE 11(7):e0158798

13. Wesolowski CA, Wanasundara SN, Babyn PS, Alcorn J (2020) Comparison of the gamma-Pareto convolution with conventional methods of characterising metformin pharmacokinetics in dogs. J Pharmacokinet Pharmacodyn 47(1):19–45. https://doi.org/10.1007/s10928-019-09666-z

14. West GB, Brown JH, Enquist BJ (1999) The fourth dimension of life: fractal geometry and allometric scaling of organisms. Science 284(5420):1677–1679. https://doi.org/10.1126/science.284.5420.1677

15. Tucker GT, Wesolowski CA (2020) Metformin disposition—a 40-year-old mystery. Br J Clin Pharmacol 86(8):1452–1453. https://doi.org/10.1111/bcp.14320

16. Tucker GT, Wesolowski CA (2021) Comment on: the pharmacokinetics of metformin in patients receiving intermittent haemodialysis by Sinnappah, et al. Br J Clin Pharmacol 87(8):3370–3371. https://doi.org/10.1111/bcp.14683

17. Tucker GT, Casey C, Phillips PJ, Connor H, Ward JD, Woods HF (1981) Metformin kinetics in healthy subjects and in patients with diabetes mellitus. Br J Clin Pharmacol 12(2):235–246. https://doi.org/10.1111/j.1365-2125.1981.tb01206.x

18. Avdis E, Watanabe M (2017) Rational-expectations whiplash. SSRN Electron J. https://doi.org/10.2139/ssrn.2933935

19. Johnston CA, Dickinson VSM, Alcorn J, Gaunt MC (2017) Pharmacokinetics and oral bioavailability of metformin hydrochloride in healthy mixed-breed dogs. Am J Vet Res 78(10):1193–1199

20. Michel D, Gaunt MC, Arnason T, El-Aneed A (2015) Development and validation of fast and simple flow injection analysis-tandem mass spectrometry (FIA-MS/MS) for the determination of metformin in dog serum. J Pharm Biomed Anal 107:229–235

21. Bollen KA, Stine RA (1992) Bootstrapping goodness-of-fit measures in structural equation models. Sociol Methods Res 21(2):205–229

22. Zhang X, Savalei V (2016) Bootstrapping confidence intervals for fit indexes in structural equation modeling. Struct Equ Model 23(3):392–408

23. Laugwitz D, Neuenschwander E (1994) Riemann and the Cauchy-Hadamard formula for the convergence of power series. Historia Math 21(1):64–70

24. Gurland J, Tripathi RC (1971) A simple approximation for unbiased estimation of the standard deviation. Am Stat 25(4):30–32

25. Brody JP, Williams BA, Wold BJ, Quake SR (2002) Significance and statistical errors in the analysis of DNA microarray data. Proc Natl Acad Sci 99(20):12975–12978

26. Wise ME (1985) Negative power functions of time in pharmacokinetics and their implications. J Pharmacokinet Biopharm 13(3):309–346. https://doi.org/10.1007/BF01065658

27. Dokoumetzidis A, Magin R, Macheras P (2010) Fractional kinetics in multi-compartment systems. J Pharmacokinet Pharmacodyn 37(5):507–524. https://doi.org/10.1007/s10928-010-9170-4

28. Garrett ER (1994) The Bateman function revisited: a critical reevaluation of the quantitative expressions to characterize concentrations in the one compartment body model as a function of time with first-order invasion and first-order elimination. J Pharmacokinet Pharmacodyn 22(2):103–128. https://doi.org/10.1007/BF02353538

29. Wanasundara SN, Wesolowski MJ, Barnfield MC, Waller ML, Murray AW, Burniston MT, Babyn PS, Wesolowski CA (2016) Accurate and precise plasma clearance measurement using four

^{99m}Tc-DTPA plasma samples over 4 h. Nucl Med Commun 37(1):79

30. Wesolowski CA, Puetter RC, Ling L, Babyn PS (2010) Tikhonov adaptively regularized gamma variate fitting to assess plasma clearance of inert renal markers. J Pharmacokinet Pharmacodyn 37(5):435–474. https://doi.org/10.1007/s10928-010-9167-z

31. Wesolowski CA, Babyn PS, Puetter RC, inventors; Carl A. Wesolowski, assignee (2014) Method for evaluating renal function. US Patent 8,738,345. https://patents.google.com/patent/US8738345B2/en

32. Buse JB, DeFronzo RA, Rosenstock J, Kim T, Burns C, Skare S, Baron A, Fineman M (2016) The primary glucose-lowering effect of metformin resides in the gut, not the circulation: results from short-term pharmacokinetic and 12-week dose-ranging studies. Diabetes Care 39(2):198–205

33. Stepensky D, Friedman M, Raz I, Hoffman A (2002) Pharmacokinetic-pharmacodynamic analysis of the glucose-lowering effect of metformin in diabetic rats reveals first-pass pharmacodynamic effect. Drug Metab Dispos 30(8):861–868

34. Weiss M (1999) The anomalous pharmacokinetics of amiodarone explained by nonexponential tissue trapping. J Pharmacokinet Biopharm 27(4):383–396. https://doi.org/10.1023/A:1020965005254

35. Claret L, Iliadis A, Macheras P (2001) A stochastic model describes the heterogeneous pharmacokinetics of cyclosporin. J Pharmacokinet Pharmacodyn 28(5):445–463. https://doi.org/10.1023/A:1012295014352

36. Friedman J, Hastie T, Tibshirani R et al (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer series in statistics, New York

37. Green R, Hahn W, Rocke D (1987) Standard errors for elasticities: a comparison of bootstrap and asymptotic standard errors. J Bus Econ Stat 5(1):145–149

38. Burger D, Ewings F, Kabamba D, L'homme R, Mulenga V, Kankasa C, Thomason M, Gibb DM (2010) Limited sampling models to predict the pharmacokinetics of nevirapine, stavudine, and lamivudine in HIV-infected children treated with pediatric fixed-dose combination tablets. Ther Drug Monit 32(3):369–372. https://doi.org/10.1097/ftd.0b013e3181d75e47

39. Smithson M (1982) On relative dispersion: a new solution for some old problems. Qual Quant 16(3):261–271

40. Volder J (1959) In: The CORDIC Computing Technique vol 1 Los Alamitos, CA, USA: IEEE Computer Society, p 257. https://doi.ieeecomputersociety.org/10.1109/AFIPS.1959.57

41. Agana MJ (2015) The classical theory of rearrangements. Master of Science in Mathematics Thesis, Boise State University. https://scholarworks.boisestate.edu/cgi/viewcontent.cgi?article=2052&context=td