

Gaussian basis functions for solving differential equations

A. Allison¹ D. Abbott² C. E. M. Pearce³

(Received 26 March 2010; revised 22 September 2010)

Abstract

We derive approximate numerical solutions for an ordinary differential equation common in engineering using two different types of basis functions, polynomial and Gaussian, and a maximum discrepancy error measure. We compare speed and accuracy of the two solutions. The basic finding for our example is that while Gaussian basis functions can be used, the computational effort is greater than that required for a polynomial basis given the same degree of error.

Contents

1 Introduction C748

2 Design choices for approximate solutions C751

<http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/2630> gives this article, © Austral. Mathematical Soc. 2010. Published October 26, 2010. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

<i>1</i>	<i>Introduction</i>	C748
3	A simple RC circuit	C752
3.1	Global basis functions and least squares	C753
4	Numerical examples	C754
4.1	Polynomial basis functions	C754
4.2	The use of weighting functions	C757
4.3	Gaussian basis functions	C759
5	Summary and prospective developments	C763
	References	C765

1 Introduction

The most common aim of numerical approximation is to approximate a function that cannot be calculated exactly, in terms of other functions that can, using the available computing environment. The success of an approximate solution may be evaluated in terms of the accuracy of the solution and the amount of computational effort required to produce it.

One approach to numerical solutions of ordinary differential equations (ODEs) is representation as a sum of basis functions. For example, Taylor series use polynomial functions as basis functions, the coefficients being chosen to match derivatives at an initial value. This is effectively collocation, within a range that includes the values of the function and a number of its derivatives. This is shown by Abramowitz [1], for example.

By the Weierstrass approximation theorem, subsequently extended and simplified by Stone [10, 11, 12], polynomial functions are dense in the space $\mathcal{C}[\mathbf{a}, \mathbf{b}]$ of real continuous functions on an interval $[\mathbf{a}, \mathbf{b}]$, and a transcendental function on $[\mathbf{a}, \mathbf{b}]$ can be approximated by the use of polynomials of sufficiently high order to an arbitrary degree of accuracy. Gaussian functions are also dense in function space [4] and can be used as basis functions.

A common difficulty for initial value collocation with Taylor series is that solutions often become inaccurate for large arguments. This can be avoided by limiting the maximum value of $|t|$, expanding the function over narrow sub-intervals $|t - \mu_k| < \tau/2$ with common length τ . However with this approach the conditions all need to be matched at the boundaries of each sub-interval. It is necessary in computation to store all boundary values, apply conditions of continuity and then solve for the required parameters. This is computationally expensive because of the amount of linear algebra. The division of a domain into sub-domains leads to spline methods [8] and finite element methods [6]. However, the use of sub-intervals and internal boundary conditions has been successfully avoided by spectral methods [13].

For $f(t) = \exp(-t)$, the truncated Taylor series $g(t) = \sum_{l=0}^L (-t)^l/l!$ with $L = 16$ has the derivatives $g^{(i)}(0) = f^{(i)}(0)$ for $0 \leq i \leq 16$. This is initial value collocation. Figure 1 shows $g(t) - f(t)$ over $0 \leq t \leq 7$. The approximation is good for $0 \leq t \leq 5$ but the error grows significantly on $5 < t \leq 7$. For $t > 7$, the errors are even larger. To obviate this problem we use a minimal square integral method over an entire domain to construct numerical solutions.

A minimal square integral approach can lead to ill-posed problems if the basis functions are chosen inappropriately. Basis functions of the form t^k have the problem that they become very large in absolute terms as $t \rightarrow \infty$. This means that they are inappropriate when approximating a function like $\exp(-\omega_1 t)$ as $t \rightarrow \infty$, since $\lim_{t \rightarrow \infty} \exp(-\omega_1 t) = 0$ when $\omega_1 > 0$. In addition, powers t^k , while linearly independent, are rather similar in behaviour for large positive values of k and t . Thus the resulting linear equations form systems that are nearly singular. They have high condition numbers and tend to be ill-conditioned. Small round-off errors in the arithmetic lead to very large errors in the calculated solutions. To avoid this problem, we consider the use of Gaussian basis functions. These are quasi-orthogonal for appropriate positions and widths of the Gaussian functions. The resulting matrices are diagonally dominant and the linear algebra problems ought to be well conditioned, but we find life is not quite so simple.

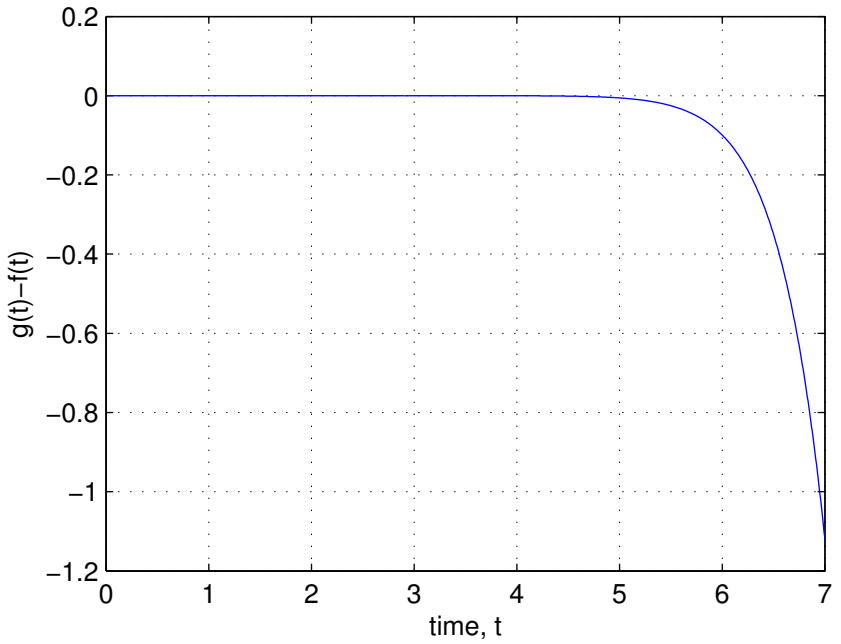


FIGURE 1: Errors in a truncated Taylor series.

TABLE 1: Design choices for approximate solutions to ODEs.

Concept	Approach
Approximate solution	linear weighted sum of basis functions $g(t) = \sum_{k=0}^{N-1} b_k \phi_k(t)$
Domain of approximation	one complete domain, of time, t , $T_1 \leq t \leq T_2$, with no subdivision
Method of approximation	minimal square integral over a domain (not collocation at points) defined by the conservation laws giving the ODE
Error measurement	maximum absolute deviation, or discrepancy
Initial conditions	enforced using constraints and making use of the Lagrange multiplier method

In summary, we consider the use of Gaussian basis functions as global basis functions, using a minimal square integral for constructing approximate solutions to ODEs.

2 Design choices for approximate solutions

There are many possible choices that can be made when constructing approximate solutions. Our general paradigm, depicted in Table 1, serves as a testing environment for comparing different basis functions. We consider polynomial functions $\phi_k(t) = t^k$ and Gaussian (normal) functions

$$\phi_k(t) = \mathcal{N}(t, \mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(t - \mu_k)^2}{2\sigma_k^2}\right).$$

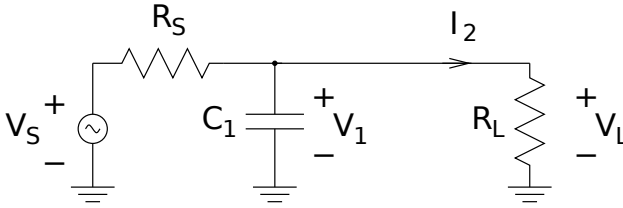


FIGURE 2: Circuit diagram for a first order Butterworth filter.

For the present, we choose a constant variance width ($\sigma_k^2 = \sigma^2$) and use arrays of Gaussian functions with equal spacing

$$\mu_{k+1} - \mu_k = \tau. \quad (1)$$

We use Gaussian functions rather than Hermite functions, which are Gaussian functions multiplied by Hermite polynomial functions [3].

3 A simple RC circuit

We consider a problem that has engineering applications and is simple enough to have a closed form solution but which presents a meaningful challenge to our numerical methods. Figure 2 shows a standard ladder circuit used to realise the simplest Butterworth filter. The source impedance is R_s and the load impedance R_L . The circuit has a single time constant τ_1 , corresponding to a single corner angular frequency $2\pi f_1 = \omega_1 = 1/\tau_1$. There is one state variable, the voltage V_1 across C_1 . The current $I_2 = V_1/R_L$ is not an independent state variable, since it can be expressed in terms of V_1 . The output voltage V_L equals V_1 and so also is not an independent state variable.

Standard circuit analysis produces the equation of state

$$\frac{dV_1(t)}{dt} + \omega_1 V_1(t) = \omega_s V_s(t), \quad (2)$$

where

$$\omega_1 = \frac{1}{C_1} \left(\frac{1}{R_s} + \frac{1}{R_l} \right) \quad \text{and} \quad \omega_s = \frac{1}{R_s C_1}.$$

Equation (2) is a linear first order ODE and is solved readily using an integrating factor or by Laplace transformation. Treating only impulsive transient responses ($V_s(t) = 0$ for $t > 0$), we specify the initial value for V_1 as $V_{1,0} = \lim_{t \rightarrow 0^+} V_1(t)$. There is a closed form solution

$$V_1(t) = V_{1,0} \exp(-\omega_1 t). \quad (3)$$

We could evaluate the solution using a reciprocal Taylor series

$$V_1(t) \approx \frac{V_{1,0}}{\sum_{l=0}^L (\omega_1 t)^l / l!}.$$

The issue is whether we can approximate the solution (3) using a weighted sum of basis functions, and whether more generally we can solve ODEs that are more complicated than (2) and which may not have closed form solution.

For simplicity, we consider the corresponding normalised and dimensionless ODE $f'(t) - \alpha f(t) = 0$, with solution $f(t) = f_0 e^{\alpha t}$. This may be approximated by various functions $g(t) = \sum_{k=0}^{N-1} b_k \phi_k(t)$. For comparison with the Taylor series of Figure 1, we choose $\alpha = -1$.

3.1 Global basis functions and least squares

We seek to minimise the square integral

$$H^2 = \int_{T_1}^{T_2} w(t) (g'(t) - \alpha g(t))^2 dt$$

where $g(t) = \sum_{k=0}^{N-1} b_k \phi_k(t)$ and b_k is real. For a polynomial basis it is convenient to take $w(t) = 1$ on the interval $[T_1, T_2]$. For a Gaussian basis

functions analysis is simplified if the weighting function can be expressed in terms of basis functions $\phi_j(t)$ as $w(t) = \sum_{j=J_1}^{J_2} v_j \phi_j(t)$, where $0 \leq J_1 < J_2 \leq N - 1$.

In general, for $w(t)$ a linear combination of basis functions, we obtain

$$\frac{\partial H^2}{\partial \mathbf{b}_l} = 2 \int_{T_1}^{T_2} (\phi'_l(t) - \alpha \phi_l(t)) \left(\sum_{j=J_1}^{J_2} v_j \phi_k(t) \right) \sum_{k=0}^{N-1} \mathbf{b}_k (\phi'_k(t) - \alpha \phi_k(t)) dt.$$

We also impose a constraint $\mathbf{g}(T_1) = \sum_{k=0}^{N-1} \mathbf{b}_k \phi_k(T_1) = f(T_1)$ to represent the initial condition. Our least squares problem is a constrained minimisation, so we use a Lagrange multiplier λ to obtain

$$\frac{\partial H^2}{\partial \mathbf{b}_l} - \lambda \phi_l(T_1) = 0, \quad 0 \leq l \leq N - 1.$$

The constants \mathbf{b}_k then satisfy

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N-1} & -\phi_0(T_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{N-1,0} & A_{N-1,1} & \cdots & A_{N-1,N-1} & -\phi_{N-1}(T_1) \\ \phi_0(T_1) & \phi_1(T_1) & \cdots & \phi_{N-1}(T_1) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{b}_0 \\ \vdots \\ \mathbf{b}_{N-1} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f(T_1) \end{bmatrix} \tag{4}$$

where the coefficients $A_{l,k}$ depend on the choice of basis function.

4 Numerical examples

4.1 Polynomial basis functions

For the polynomial basis function, we have

$$A_{l,k} = 2lk \frac{T_2^{l+k-1} T_1^{l+k-1}}{l+k-1} - \alpha(l+k) \frac{T_2^{l+k} T_1^{l+k}}{l+k} + \alpha^2 \frac{T_2^{l+k+1} T_1^{l+k+1}}{l+k+1}. \tag{5}$$

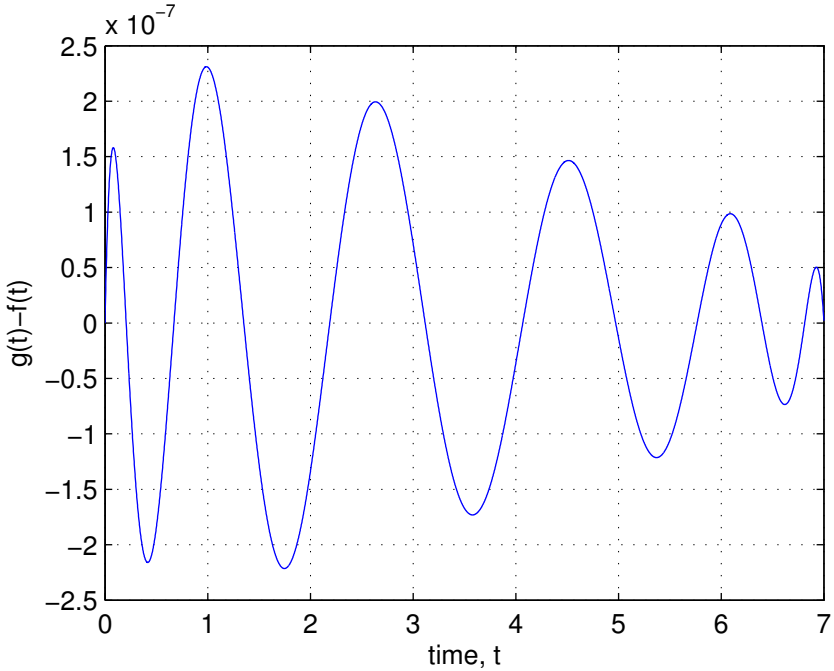


FIGURE 3: A least squares polynomial approximation.

There are some special cases, handled using limits: $A_{00} = 2\alpha^2 (T_2 - T_1)$ and $A_{01} = A_{10} = 2\alpha (T_2 - T_1) (\alpha (T_2 + T_1) - 2)$. The linear equations represented by (4) can then be solved to obtain polynomial coefficients \mathbf{b}_k .

In Figure 3 $f(t)$ is approximated using a polynomial function $g(t)$, the polynomial coefficients being chosen to minimize the mean square deviation $H^2 = \int_{T_1}^{T_2} (g'(t) - \alpha g(t))^2 dt$ subject to $g(T_1) = f(T_1)$. As shown in Figure 3, the maximum absolute deviation on the interval is less than 2×10^{-7} .

In Figure 4 this is plotted as a function of the number of basis functions. We obtain almost one additional decade of performance for each basis function until we reach $N = 12$, after which the precision is limited by the precision of the arithmetic. It is possible to show that the precision of the arithmetic

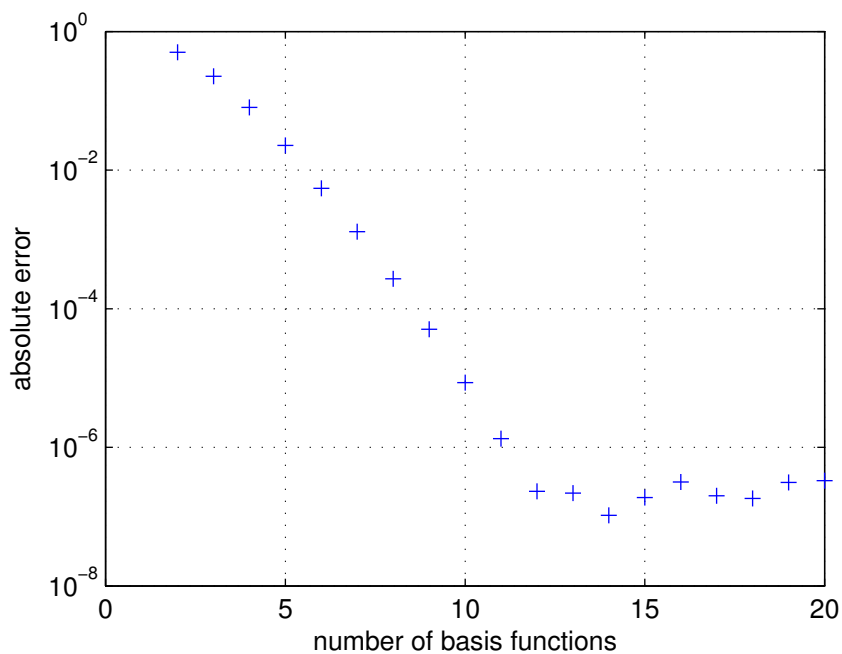


FIGURE 4: Error of approximation for polynomials.

is the limiting factor here, by running the same computer code, but with reduced (single precision) arithmetic. It is reasonable to assume that, as mentioned by Trefethen [13, p.6], precision could be improved by running the same algorithm in a multiple precision environment, such as LISP, MAPLE or MACSYMA but we have not yet carried out this experiment.

4.2 The use of weighting functions

For polynomial basis functions, the weighting function was effectively a rectangular window. It is not possible to integrate along the whole real line without a window because the resulting integral would not converge. Integrating Gaussian functions over a finite integral is a computationally expensive operation. It seems better to approximate the rectangular window by a sum $w(t) = \sum_{j=1}^{J_2} v_j \phi_k(t)$ of Gaussian functions $\phi_k(t) = \mathcal{N}(t, \mu_k, \sigma_k^2)$. The positions μ_k and widths σ_k of the basis functions need to be chosen to give a good flat window. With equal spacing $\mu_{j+1} - \mu_j = \tau$ and equal width $\sigma^2 = \tau^2/2$ we obtain

$$w(t) = \tau \sum_{j=0}^{N-1} \mathcal{N}(t, \mu_j, \sigma^2).$$

As illustrated in Figure 5(a) this gives very suitable windows. Use of more Gaussian functions should provide more accurate approximation, as shown in Figure 5(b).

The user has a free choice for the relationship between the inter-function spacing τ and the mean square width σ of each basis function. A very good approximation can be obtained when

$$\sigma^2 = \tau^2/2. \tag{6}$$

The windowing function can then be written as $w(t) = \tau \sum_{j=0}^{N-1} \mathcal{N}(t, \mu_j, \sigma^2)$. Other functions can be approximated by multiplying these Gaussian window-

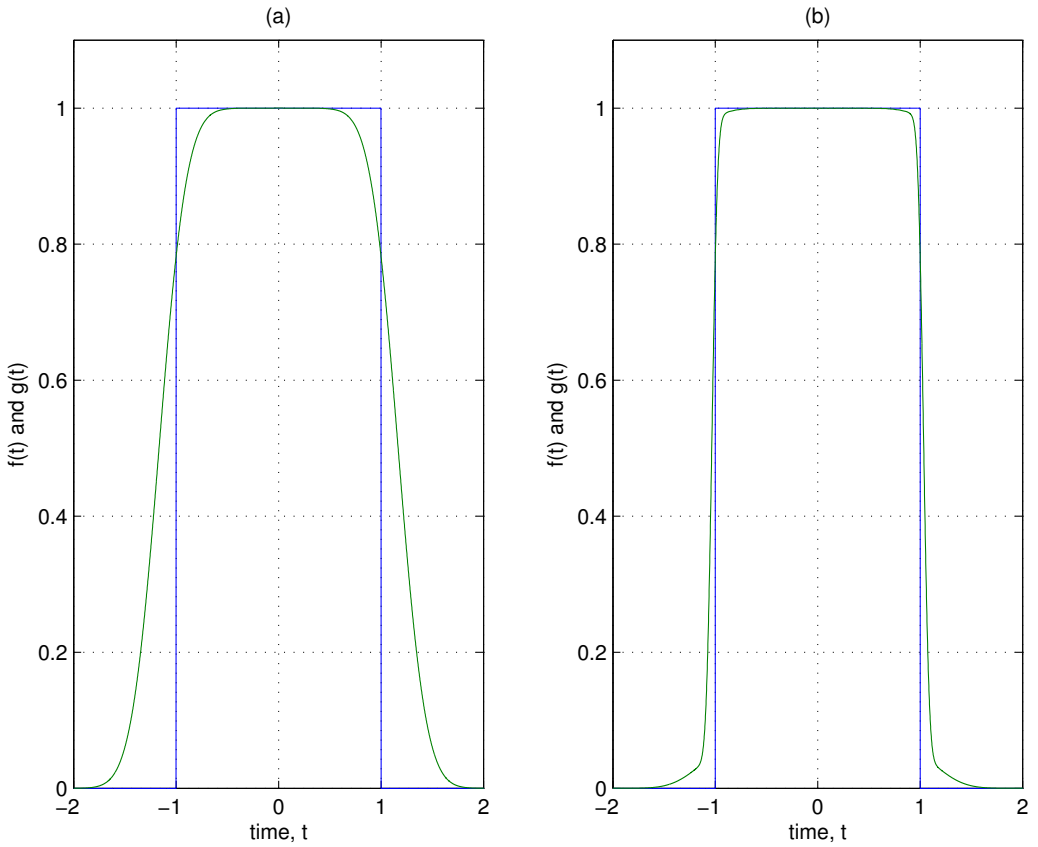


FIGURE 5: A Gaussian approximation to a rectangular window.

ing function by the required target function as

$$f(t) \approx \sum_{j=0}^{N-1} \tau f(\mu_j) \mathcal{N}(t, \mu_j, \sigma^2).$$

This approximation is simple and practical, but not guaranteed to be optimal in the least squares sense.

4.3 Gaussian basis functions

For the Gaussian basis function,

$$A_{l,k} = \frac{4}{\tau^2} \sqrt{\frac{2}{\pi}} \mathcal{F}, \quad (7)$$

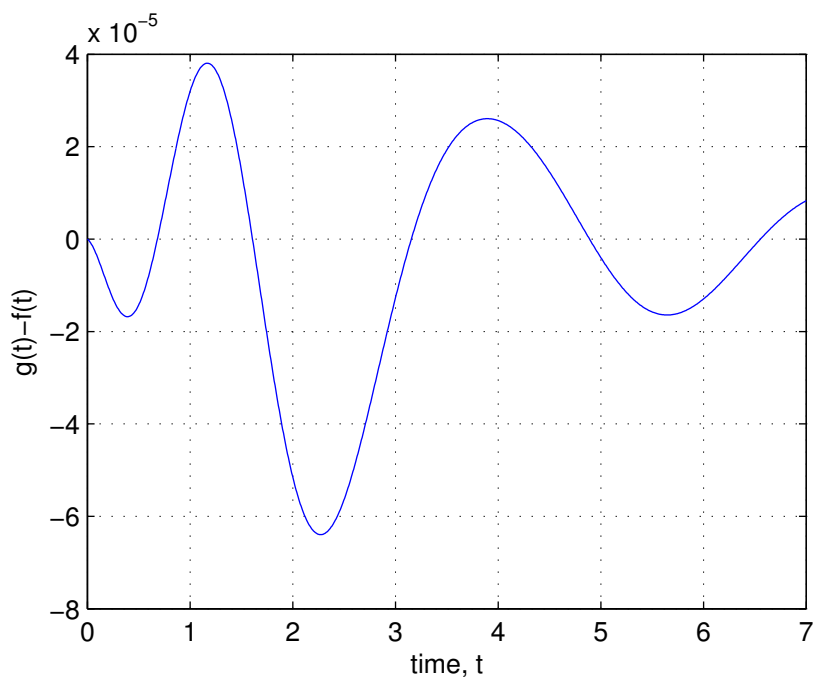
where

$$\mathcal{F} = \left(\frac{1}{2} \alpha \tau + \frac{1}{3} (j - 2k + l) \right) \left(\frac{1}{2} \alpha \tau + \frac{1}{3} (k + j - 2l) \right) + \frac{1}{6}, \quad (8)$$

for $0 \leq l, k \leq N - 1$ and $\mu_{\text{prod}}^2 = \tau^2 (j^2 + k^2 + l^2 - jk - jl - kl)$.

The derivation of (7) and (8) use properties of products of Gaussian functions based on completion of the squares of the exponents.

The linear equations represented by (4) are then solved to obtain the coefficients b_k , which are chosen to minimize $H^2 = \int_{T_1}^{T_2} (g'(t) - \alpha g(t))^2 dt$ subject to the constraint $g(T_1) = f(T_1)$. The deviation between $g(t)$ and $f(t)$ is shown in Figure 6. For $f(t) = e^{-t}$, the accuracy of the approximation is limited by the choice of basis functions near $t = T_1$. Parts of $f(t)$ where rapid change occurs require more basis functions than some other parts. This limitation may force us to use basis functions which are not equally spaced. The error performance of this method, with Gaussian basis functions, is shown in Figure 4.

FIGURE 6: A least squares approximation to $f(t)$.

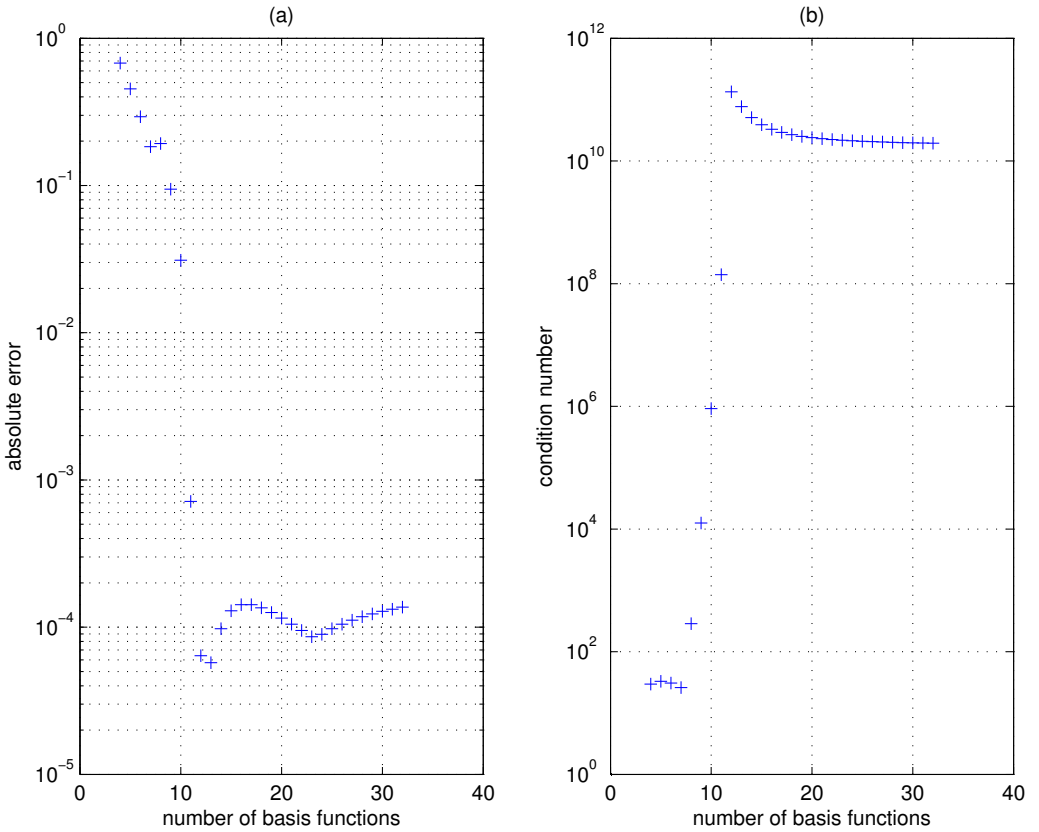


FIGURE 7: Accuracy of Gaussian approximations.

Figure 7(a) shows a plot of the maximum absolute deviation as a function of the number of basis functions. The approximations for $N < 10$ are not particularly accurate, because the exponential function decays rapidly and this aspect of the target function $f(t)$ cannot easily be approximated by a few slowly varying Gaussian functions. Curiously, the algorithm does not converge very rapidly for $N > 10$. The greatest accuracy is obtained for $N = 13$, after which accuracy seems limited. The condition numbers of the underlying equations are shown in Figure 7(b). Large condition numbers usually occur in cases where supposedly different basis functions are actually very similar. A contributing factor to this problem is matching the initial conditions near $t = T_1$. More basis functions may be needed in certain areas, which works against the idea of using simple arrays of equally spaced Gaussian basis functions.

One possible approach to providing more basis functions near the initial condition is to include guard functions slightly outside of the region in which approximation is made. These are to help with the matching of boundary conditions, which strongly affects the accuracy of this method. These functions are still Gaussian and equally spaced with the same properties as the other basis functions. The difference is that the central positions μ_k lie outside the range of the approximation. For $N \leq 8$ there was only one guard basis function at each edge of $[T_1, T_2]$. For $9 \leq N \leq 12$ this was increased to six basis functions near T_1 , keeping one guard basis function near T_2 .

The benefit of more guard basis functions near T_1 is improved accuracy of the solution. However, since these functions are very similar in shape in $[T_1, T_2]$, the resulting linear equation scan is very poorly conditioned, as indicated in Figure 7(b). Thus the use of guard functions is not entirely successful in smoothly matching the initial conditions.

In Figure 7, the coefficients \mathbf{b} are evaluated by inverting a system of linear equations $\mathbf{A}\mathbf{b} = \mathbf{c}$. Accuracy is limited by the condition number of order two $\kappa(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2$. This is shown as a function of the number of basis functions. Equations are well conditioned if the condition numbers are

close to unity, so there are problems with numerical solution when condition numbers of order 10^{10} occur. Thus although we may use equally spaced Gaussian basis functions for special purposes, they are not in general a good choice for solving ODEs.

5 Summary and prospective developments

We examined the use of global basis functions and mean square measure of error. The polynomial basis functions performed well and gave superior accuracy to a Taylor series for $N \leq 12$. This limitation could be possibly be overcome by using multiple precision arithmetic or orthogonal polynomials rather than simple powers t^k . With Gaussian basis functions the approximations are more laborious and less accurate. Convergence for Gaussian basis functions cannot be guaranteed by simply increasing N . The Gaussian basis functions did not give significantly improved accuracy for $N > 12$. There were no values of N for which the Gaussian functions outperformed polynomial basis functions. Apparently *simple is best* in this case.

The Gaussian technique does not appear to be generally efficient, which will often rule out its use. However there are niche problems that benefit from this approach. In particular, a huge literature of applications to quantum chemistry [7, e.g.]. We have previously studied diffusion in inhomogeneous materials in the presence of time varying fields [2, e.g.]. We suspect that Gaussian functions may prove useful for solving diffusion problems of this type.

Diffusion is often modelled by the Fokker–Planck equation, as shown by Gardiner [5, 9] for example,

$$\frac{\partial}{\partial x} (D^{(2)}(t, x) p_x(t, x)) - \frac{\partial}{\partial x} (D^{(1)}(t, x) p(t, x)) - p_t(t, x) = 0. \quad (9)$$

When $D^{(1)} = 0$ and $D^{(2)}$ is constant, (9) reduces to the diffusion equation

$$D^{(2)} \frac{\partial^2 p}{\partial x^2} - \frac{\partial p}{\partial t} = 0. \quad (10)$$

This may be solved using Gaussian probability density functions

$$p(t, x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right), \quad (11)$$

where the variance

$$\sigma^2 = 2D^{(2)}t \quad (12)$$

is a linear function of time. Direct substitution gives

$$D^{(2)} \frac{\partial^2 p}{\partial x^2} = \frac{\partial p}{\partial t} = \frac{-1}{2t} \left(1 - \left(\frac{x}{\sigma}\right)^2\right) p(t, x),$$

which verifies that (11) is a solution to (10).

The solution to these equations has an interesting history and many applications. Fourier developed this solution to the diffusion equation in connection with heat flow. Einstein later applied Fourier's solution to Brownian motion. Fourier's solution is also relevant to propagation of signals in RC transmission lines.

We plan to generate approximate solutions the diffusion equation (10) using arrays of fixed position Gaussian functions, first approximating the initial conditions using sums of Gaussian functions and then determining the effect of diffusion over a short time interval using (11) and (12). However, to simulate the effect of a second time step, it becomes necessary to readjust the widths σ of the Gaussian functions to bring them into proportion with the sampling intervals, that is, satisfying (6). Without this adjustment, the basis functions may cease to be representative of the possible solutions. The resulting equations may become poorly conditioned leading to inaccurate solutions. If we simulate (9) the sample basis function will change variance

and position. The requirement to reassert (1) and (6) becomes even more pressing.

A resolution would be to reconvert the function represented by a set of unequally spaced Gaussian functions of unequal variance to an equivalent function represented by equally spaced Gaussian functions of equal variance. The algorithm would take the following form.

- Initialise the representation of the initial conditions, satisfying both (1) and (6).
- Repeat the following steps for the required length of time
 1. Take the solution forward one time step using (11) and (12).
 2. Reinitialize the representation of the present conditions to satisfy (1) and (6).

We expect that the time evolution step will be almost trivial. The major numerical cost will be in re-initialisation of the solution.

We hope that Gaussian basis functions will prove useful for simulating diffusion problems, especially when fields are time varying and the media inhomogeneous. Using Gaussian functions to solve ODEs is intended as a first step towards solving parabolic PDEs associated with diffusion.

References

- [1] M. Abramowitz and I. A. Stegun, eds. *Handbook of Mathematical Functions*. Dover Publications, New York, 1970. C748
- [2] A. Allison and D. Abbott. The physical basis for Parrondo's games. *Fluctuation and Noise Letters*, 2(4): L327–L341, 2002. C763
- [3] P. W. Atkins. *Physical Chemistry*. Oxford University Press, Oxford, fifth ed., 1994 C752

- [4] C. Calcaterra. Linear combinations with a single variance are dense in \mathcal{L}^2 . In *Proceedings of the World Congress on Engineering, WCE 2008*, vol. II, London, July 2–4 2008. World Congress on Engineering. ISBN 978–988–17012–3–7. **C748**
- [5] C. W. Gardiner. *Handbook of Stochastic Methods*. Springer–Verlag, Berlin, 1983. **C763**
- [6] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, Cambridge, 1966. **C749**
- [7] G. L. Malli, A. B. F. Da Silva and Yasuyuki Ishikawa. Universal Gaussian basis set for relativistic calculations on atoms and molecules. *Chemical Physics Letters* **201** 1–4 37–40, 1993. **C763**
- [8] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, New York, second ed., 1995. **C749**
- [9] H. Risken. *The Fokker–Planck Equation*. Springer–Verlag, Berlin, 1996. **C763**
- [10] M. H. Stone. Applications of the theory of boolean rings to general topology. *Trans. Amer. Math. Soc.*, **41** 375–481, 1937. **C748**
- [11] M. H. Stone. The generalized Weierstrass approximation theorem, part 1. *Mathematics Magazine* **21** 167–184, 1948. **C748**
- [12] M. H. Stone. The generalized Weierstrass approximation theorem, part 2. *Mathematics Magazine* **21** 237–254, 1948. **C748**
- [13] L. N. Trefethen. *Spectral Methods in MATLAB*. SIAM, Philadelphia, 2000. **C749**, **C757**

Author addresses

1. **A. Allison**, School of Electrical and Electronic Engineering, University of Adelaide, South Australia 5005, AUSTRALIA.
<mailto:aallison@eleceng.adelaide.edu.au>
2. **D. Abbott**, School of Electrical and Electronic Engineering, University of Adelaide, South Australia 5005, AUSTRALIA.
3. **C. E. M. Pearce**, School of Mathematical Sciences, University of Adelaide, South Australia 5005, AUSTRALIA.