Spectral computation with Laguerre functions

Nathaniel Jewell
The University of Adelaide, Australia

December 11, 2009

1 Properties of Laguerre functions

The Laguerre polynomials are defined as the set of polynomials which are orthonormal over the real half-line $[0, \infty)$ with respect to the weight function $w(x) = e^{-x}$, ie

$$\int_0^\infty L_m(x)\, L_n(x)\, e^{-x}\, \mathrm{d}x \;=\; \delta_{mn}, \qquad m, n \geq 0, \tag{1}$$

and which satisfy the boundary conditions

$$L_n(0) = 1, \quad L_n'(0) = -n \qquad \text{for each } n. \tag{2}$$

The Laguerre functions $\{\widehat{L}_n(x)\}$ are simply the Laguerre polynomials rescaled by $\sqrt{w}$, ie

$$\widehat{L}_n(x) \;=\; L_n(x)\, e^{-x/2}, \qquad n = 0, 1, 2, \dots, \tag{3}$$

and thus are orthonormal:

$$\int_0^\infty \widehat{L}_m(x)\, \widehat{L}_n(x)\, \mathrm{d}x \;=\; \delta_{mn}. \tag{4}$$

Laguerre polynomials and functions are discussed in numerous reference texts, including Kreyszig (1978, pp. 184–8), Weisstein (2003, pp. 1675–9), and Abramowitz and Stegun (1972, Chapter 22). Yet in practice they are seldom used for numerical approximation. Shen (2000) states that this lack of enthusiasm can be traced to a history of poor implementation of Laguerre methods, leading to disappointing results. In most cases authors have chosen to approximate functions on the real half-line by Laguerre *polynomials* rather than *functions*:

> Pioneer work on Laguerre approximation [has been done by several authors]. ... However, to the best of our knowledge, most of the previous numerical results used Laguerre polynomials, which, from a theoretical point of view, provide meaningful results only inside small intervals ... and ... are not suitable for practical computations, due to the extremely ill-conditioned behaviors of the Laguerre polynomials and of the Laguerre–Gauss–Radau quadrature formula.

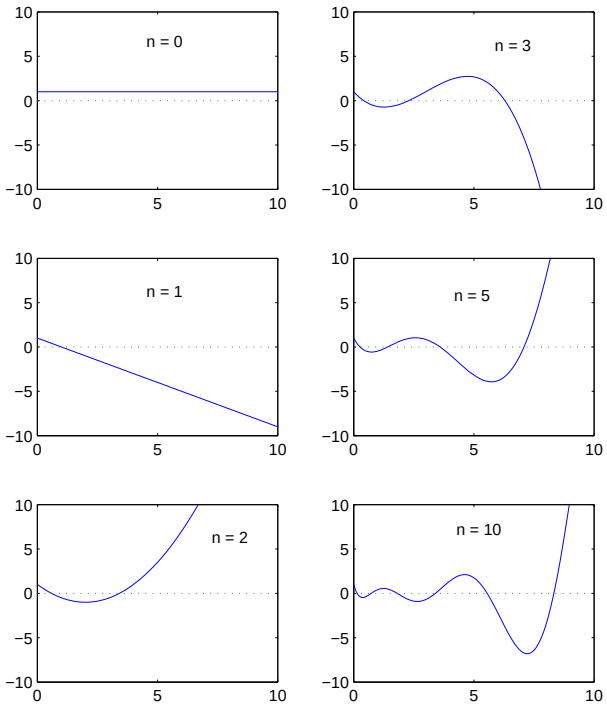

Figure 1: The Laguerre polynomials $\{L_n(x)\}$ of orders $n = 0, 1, 2, 3, 5, 10$.

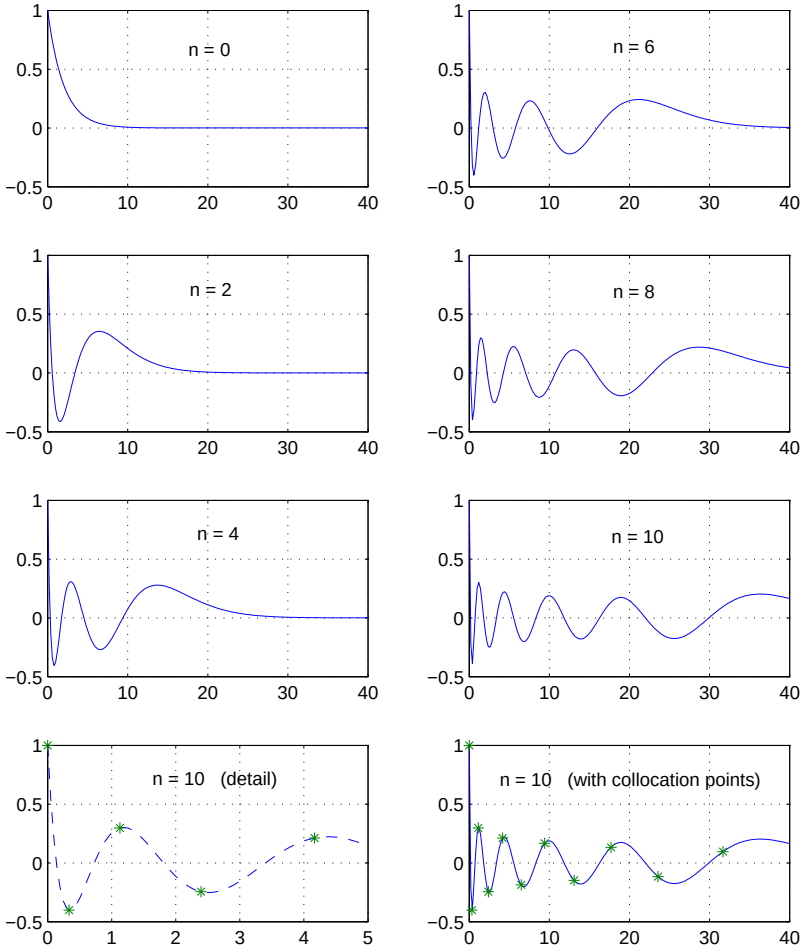

Figure 2: The (unscaled) Laguerre functions $\{\widehat{L}_n(x)\}$ of orders $n = 0, 2, 4, 6, 8, 10$. They are orthonormal over the semi-infinite domain $0 < x < \infty$. The bottom-left and bottom-right panels show $\widehat{L}_{10}(x)$ superimposed with the (unscaled) Laguerre collocation points of order $N = 10$ (see §3).

Figures 1 and 2 plot the first few Laguerre polynomials and Laguerre functions respectively. For numerical work we normally scale them by some factor $\kappa > 0$ (typically much larger than unity). We therefore define the *rescaled Laguerre functions* $\{\widetilde{L}_n(x)\}$ by

$$\widetilde{L}_n(x) \equiv \widetilde{L}_n(x;\kappa) = \widehat{L}_n(\kappa x), \qquad n = 0, 1, 2, \ldots . \tag{5}$$

These rescaled functions are obviously orthogonal but not orthonormal:

$$\int_0^\infty \widetilde{L}_m(x)\, \widetilde{L}_n(x)\, \mathrm{d}x = \kappa^{-1}\delta_{mn}. \tag{6}$$

Since the unscaled Laguerre functions carry a weighting factor of $\sqrt{w} \equiv e^{-x/2}$, one might expect them to decay rapidly to zero for large x. This turns out to be true only in an asymptotic sense. To make this statement more precise, consider the zeros $\{\, z_{mn} \,\}_{m=1}^{n}$ of L_n and $\widehat{L}_n$. These zeros are strictly positive, the largest being $z_{nn} = O(n)$. Consequently, L_n and $\widehat{L}_n$ oscillate over a sub-domain I of width $O(n)$. In fact, L_n oscillates within an envelope of magnitude $O(e^{x/2})$ until its leading term x^n dominates:

$$L_n(x) = O(e^{+x/2}) \qquad \text{for} \qquad 0 < x < (z_{nn} + O(1)), \tag{7a}$$

$$L_n(x) = O(x^n) \qquad \text{for} \qquad x > (z_{nn} + O(1)). \tag{7b}$$

Consequently, $\widehat{L}_n$ is non-negligible over a width of $O(n)$:

$$\widehat{L}_n(x) = O(1) \qquad \text{for} \qquad 0 \le x \lesssim \frac{1}{n}\,, \tag{8a}$$

$$\widehat{L}_n(x) = O\left(n^{-\frac{1}{2}}\right) \qquad \text{for} \qquad \frac{1}{n} \lesssim x < (z_{nn} + O(1)), \tag{8b}$$

$$\widehat{L}_n(x) = O\left(\exp\left[-\tfrac{1}{2}(x - z_{nn})\right]\right) \qquad \text{for} \qquad x > (z_{nn} + O(1)). \tag{8c}$$

The contrasting behaviours of L_n and $\widehat{L}_n$ are illustrated in Figure 3 for the case $n = 10$. Note that the quantity n^{-1} in the domain of (8a) is a consequence of the boundary condition (2), while the factor of $n^{-\frac{1}{2}}$ in (8b) reflects the orthonormality of the Laguerre functions.

We close this introductory section by discussing the practical evaluation of Laguerre functions. Firstly, we note that the nth Laguerre *polynomial* is a solution to the order-n Laguerre differential equation

$$xL_n'' + (1 - x)L_n' + nL_n = 0. \tag{9}$$

It may be computed either from the explicit formula

$$L_n(x) = \sum_{m=0}^{n} \alpha_{mn} x^m \tag{10a}$$

$$\alpha_{mn} = (-1)^m \frac{n!}{(n-m)!\,(m!)^2} \tag{10b}$$

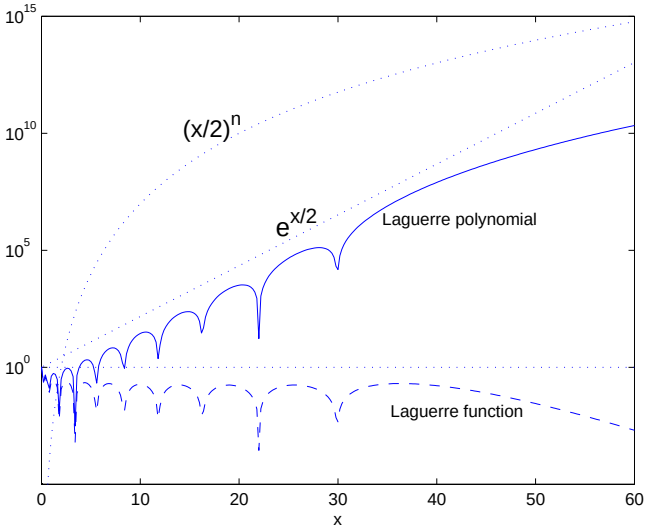

Figure 3: Semilog plot of the Laguerre polynomial $|L_{10}(x)|$ and the corresponding Laguerre function $|\widehat{L}_{10}(x)|$.

or from the following recursion:

$$L_0 = 1, \tag{11a}$$
$$L_1 = 1 - x, \tag{11b}$$
$$(n+1)L_{n+1} = ((2n+1) - x)L_n - nL_{n-1} \qquad (n > 1). \tag{11c}$$

Likewise, derivatives can be computed either explicitly (by differentiating (10a)) or from either of the following recursion formulae:

$$xL_n'(x) = n(L_n(x) - L_{n-1}(x)) \qquad (n > 0), \tag{12a}$$
$$L_n' = L_{n-1}' - L_{n-1} \qquad (n > 0). \tag{12b}$$

Hence,

$$L_0' = 0, \qquad L_n' = -\sum_{m=0}^{n-1} L_m \qquad (n > 0), \tag{13}$$

and

$$\widehat{L}_0' = -\tfrac{1}{2}\widehat{L}_0, \qquad \widehat{L}_n' = -\left[\sum_{m=0}^{n-1} \widehat{L}_m + \tfrac{1}{2}\widehat{L}_n\right] \qquad (n > 0). \tag{14}$$

We are now ready to construct a simple Laguerre spectral scheme from first principles. In §2 we will solve three ordinary differential equations using a Laguerre Galerkin scheme, prior to constructing a Laguerre pseudospectral scheme in §3.

2 A simple Laguerre–Galerkin scheme

The building block of the Laguerre Galerkin scheme is its differentiation matrix

$$\mathbf{D} \equiv \mathbf{D}_N^{\mathrm{LG}} \tag{15}$$

where N is the Laguerre spectral order. This square matrix of size $N+1$ is defined as follows. Let $f(x)$ be some function on $[0,\infty)$ with the property that $f(x) \to 0$ as $x \to \infty$. Let $\widetilde{f}$ denote an approximation of f by some linear combination of scaled Laguerre functions of order up to N, ie

$$f(x) \approx \widetilde{f}(x) = \sum_{n=0}^{N} a_{n+1} \widetilde{L}_n(x). \tag{16}$$

Similarly, let f' be approximated by

$$f'(x) \approx \widetilde{f}'(x) \equiv \sum_{n=0}^{N} b_{n+1} \widetilde{L}_n(x). \tag{17}$$

Then $\mathbf{D}$ is defined as the matrix which relates $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{b} = \mathbf{D}\mathbf{a}. \tag{18}$$

Fortunately, $\mathbf{D}$ is very easy to compute. Property (14) implies that $\mathbf{D}$ is an upper-triangular matrix. Moreover, the matrix has only two distinct non-zero entries, namely $(-\kappa/2)$ on the main diagonal and $(-\kappa)$ for the above-diagonal elements.

Let us now solve three test problems using a Laguerre Galerkin scheme. Test Problem 1 is the second-order ODE

$$\mathcal{L}_1[f] \equiv f'' - f = 0 \qquad \text{subject to} \qquad f(0) = 1, \tag{19}$$

whose exact solution is

$$f_1(x) = e^{-x}. \tag{20}$$

(The Laguerre scheme will automatically reject the alternative solution $\hat{f}_1(x) = e^{+x}$, since it does not decay asymptotically to zero.) Problem 2 is also second-order, although both of its fundamental solutions decay asymptotically to zero:

$$\mathcal{L}_2[f] \equiv f'' + 2f' + 5f = 0 \qquad \text{subject to} \qquad f(0) = 1, \quad f'(0) = 0. \tag{21}$$

Its exact solution is

$$f_2(x) = e^{-x} \left(\cos 2x + \tfrac{1}{2} \sin 2x \right). \tag{22}$$

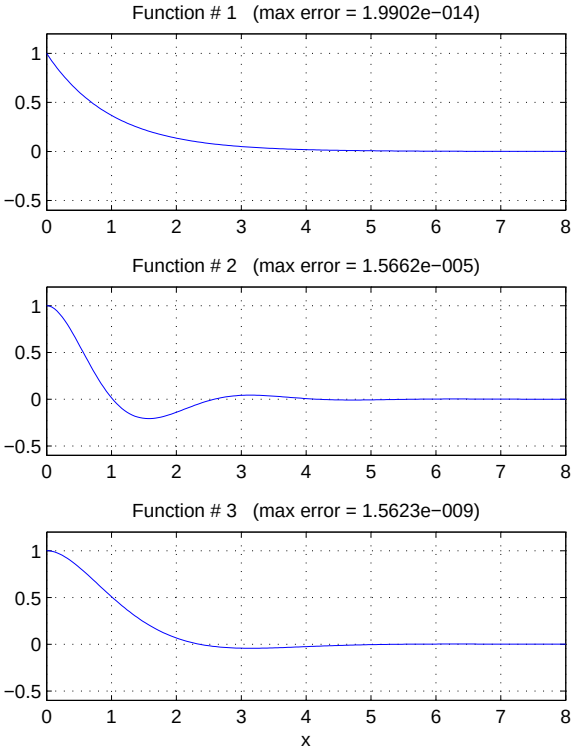

Figure 4: Results of the Laguerre Galerkin spectral scheme for the three test problems (19), (21) and (23), whose exact solutions are given by (20), (22) and (24) respectively. This figure was generated by the function `test_LG` listed in Figures 12 and 13.

Problem 3 is a fourth-order ODE, two of whose fundamental solutions decay asymptotically to zero:

$$\mathcal{L}_3[f] \equiv f^{(4)} + 4f = 0 \qquad \text{subject to} \qquad f(0) = 1, \quad f'(0) = 0. \tag{23}$$

Its exact solution is

$$f_3(x) = e^{-x}(\cos x + \sin x). \tag{24}$$

We solve Problems 1–3 using the Matlab program `test_LG` listed at the end of this section (Figure 12). Figure 4 presents the output of this program for spectral order $N = 20$ and scale factor $\kappa = N/\rho$ with $\rho = 6$. For Problem 1, the numerical solution is accurate to machine precision; for Problems 2 and 3, the maximum numerical error is of order $O(10^{-5})$ and $O(10^{-9})$ respectively.

The Laguerre–Galerkin Matlab code is largely self-explanatory. The square matrix `R` corresponds to the *residual* of the Laguerre approximation,

namely

$$R(x) \;=\; \mathcal{L}[\widetilde{f}] \;\equiv\; \sum_{n=0}^{N} r_{n+1}\widetilde{L}_n(x). \tag{25}$$

The code

```
rhs = zeros(N+1,1);
```

attempts to set $r_n = 0$ for each n. Subsequently, however, the condition $r_N = 0$ is sacrificed in favour of the boundary condition $f(0) = 1$. This is implemented by the code

```
rhs(end) = 1;    R(end,:) = 1;
```

where the unit values for `R` reflect the fact that $\widetilde{L}_n(0) = 1$ for each n. Similarly, the condition $r_{N-1} = 0$ is sacrificed in favour of the Dirichlet boundary condition, via the code

```
rhs(end-1) = 0;    R(end-1,:) = sum(D);
```

where `sum` corresponds to setting $x = 0$ in (17).

3 A simple Laguerre pseudospectral scheme

The Laguerre pseudospectral scheme is designed to deliver the advantages of its Galerkin counterpart, while simultaneously rendering the Laguerre functions invisible. This scheme replaces the Galerkin vector $\mathbf{a}$ of spectral coefficients by a working vector $\mathbf{f}$ representing $\widetilde{f}(x)$ sampled at N+1 collocation points. Internally, the computation is still based on Laguerre functions, but these are now subsumed into the differentiation matrix.

As in the Galerkin case, the building block of the Laguerre pseudospectral scheme is its differentiation matrix

$$\mathbf{D} \equiv \mathbf{D}_N^{\mathrm{LPs}}, \tag{26}$$

a square matrix of size N+1 defined by

$$\widetilde{f}'(\mathbf{x}) \;=\; \mathbf{D}\,\widetilde{f}(\mathbf{x}) \tag{27}$$

where $\mathbf{x}$ denotes the N+1 scaled collocation points. Unlike its Galerkin counterpart $\mathbf{D}_N^{\mathrm{LG}}$ (which is upper triangular), the pseudospectral differentiation matrix is full, and has no obvious symmetry pattern. It is easy to use, but somewhat tedious to derive.

The unscaled Laguerre collocation points

$$\mathbf{x} \;\equiv\; \{\, x_n \equiv x_{nN} \,\}_{n=0}^{N}. \tag{28}$$

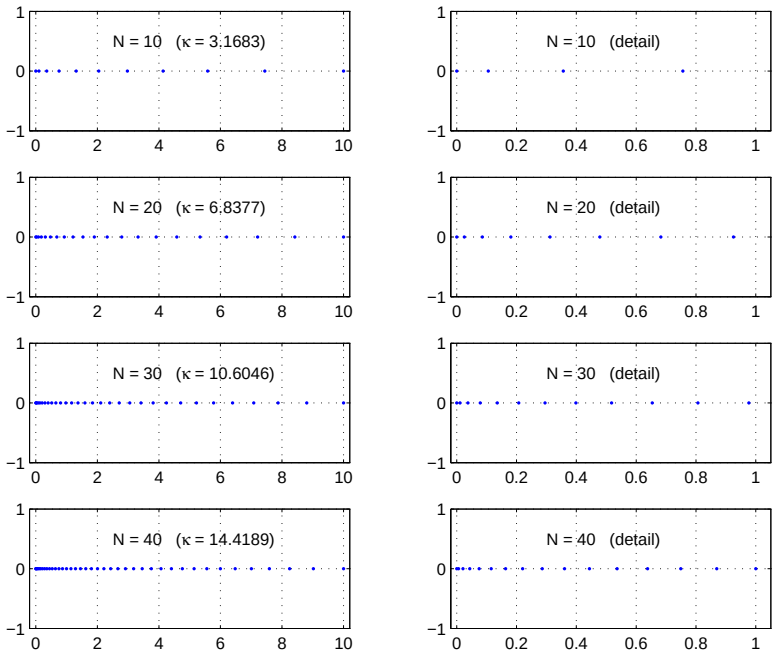

Figure 5: The rescaled Laguerre collocation points $\{\,\widetilde{x}_n\,\}_{n=0}^{N}$ for $N = 10, 20, 30$ and 40. In each case, the scale factor $\kappa_N = O(N)$ is chosen so that the collocation points span the domain $[0, x_{\max}]$ for $x_{\max} = 10$. This results in quadratic clustering near $x = 0$, meaning that the first few points are of magnitude $O\left((n/N)^2\right)$.

could in principle be chosen in arbitrary fashion (within reasonable limits). However, following convention, we define the leading collocation point by

$$x_0 = 0 \tag{29}$$

and the subsequent N collocation points by

$$L'_{N+1}(x_n) = 0 \qquad \text{for} \qquad 0 < n \leq N, \tag{30}$$

or equivalently, by

$$L_{N+1}(x) \;=\; L_N(x). \tag{31}$$

These collocation points are readily computed by applying a numerical root-finding routine to the function $g_N(x) \equiv L_{N+1}(x) - L_N(x)$. They are interleaved with the zeros of L_N:

$$(x_0 \equiv 0) < \left(z_{1N} \approx \frac{1}{N}\right) < x_1 < z_{2N} < x_2 < \cdots < z_{NN} < x_N\,, \tag{32}$$

as illustrated for $N = 10$ in the bottom-left and bottom-right panels of Figure 2. Consequently, the span of the unscaled collocation points is approximately linear in N:

$$x_n = O(N) \qquad \text{for} \qquad 1 \ll n \leq N. \tag{33}$$

In practice, it would be computationally wasteful to place collocation points at great distances from $x = 0$. It is therefore preferable to use the rescaled Laguerre collocation points defined by

$$\{\, \widetilde{x}_n \,\}_{n=0}^{N} \qquad \text{where} \qquad \widetilde{x}_n = x_n/\kappa \tag{34}$$

and

$$\kappa \equiv \kappa_N = O(N). \tag{35}$$

This is illustrated by the lower-left and lower-right panels of Figure 5, wherein each set of scaled collocation points spans the sub-domain $[0, 10]$. The policy (35) also implies that the scaled collocation points are clustered quadratically near $x = 0$, ie

$$\frac{\widetilde{x}_n}{x_{\max}} = O\left(\left(\frac{n}{N}\right)^2\right) \qquad \text{for} \quad 0 \leq n \ll N, \tag{36}$$

since the fundamental Laguerre boundary condition (2) implies that

$$x_n = O\left(N^{-1}\right) \qquad \text{for} \quad 0 < n \ll N. \tag{37}$$

This quadratic clustering also occurs in the Chebyshev and Legendre pseudospectral schemes, where it provides numerical stability by suppressing the well-known Runge phenomenon encountered in high-order polynomial approximation (see, for example, Trefethen (2000, Chapter 5) and Fornberg (1996, Chapter 3)).

The next step in computing the differentiation matrix $\mathbf{D}$ is to define the unscaled Laguerre interpolating functions $\{\mathcal{L}_{nN}\}_{n=0}^{N}$ by

$$\mathcal{L}_{nN}(x) = P_{nN}(x) \exp\left[-\tfrac{1}{2}(x - x_n)\right], \qquad 0 \leq n \leq N, \tag{38}$$

where the functions $\{P_{nN}(x)\}$ are the degree-N Lagrange interpolating polynomials associated with the unscaled collocation points. Similarly, we define rescaled Laguerre interpolating functions $\{\widetilde{\mathcal{L}}_{nN}\}_{n=0}^{N}$ by

$$\widetilde{\mathcal{L}}_{nN}(x) = \mathcal{L}_{nN}(\kappa x), \qquad 0 \leq n \leq N, \tag{39}$$

as shown in Figure 6 for $N = 10$. By definition, the Laguerre interpolants possess the delta-function property

$$\widetilde{\mathcal{L}}_n(\widetilde{x}_m) = \mathcal{L}_n(x_m) = P_n(x_m) = \delta_{mn} \qquad \text{for} \qquad 0 \leq m, n \leq N. \tag{40}$$

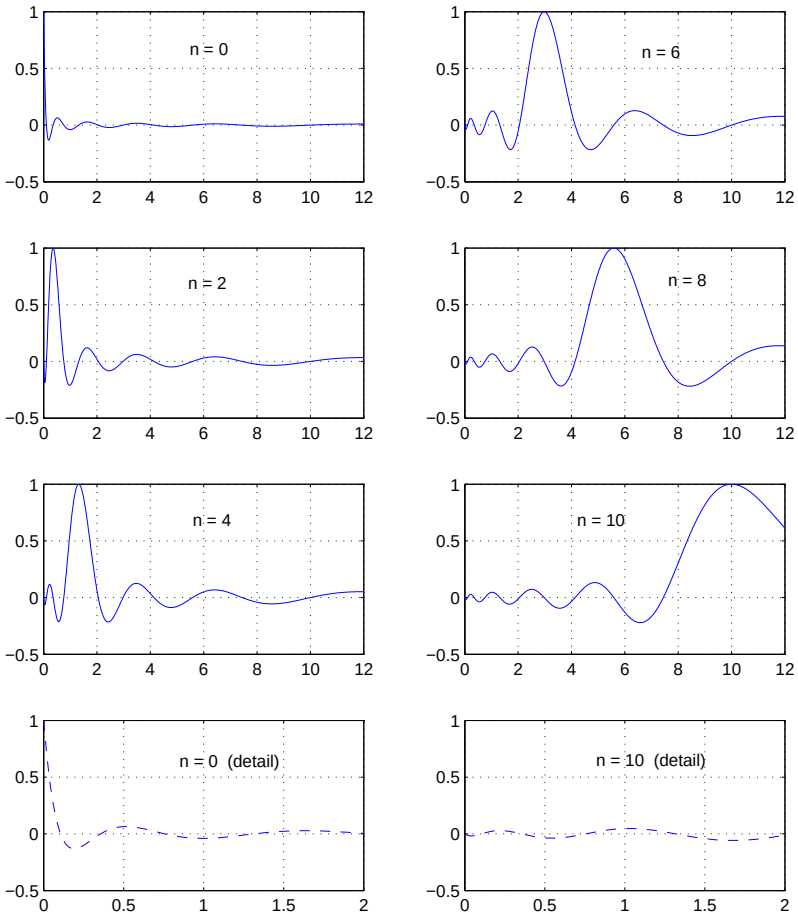

Figure 6: The rescaled Laguerre interpolating functions $\{\,\widetilde{\mathcal{L}}_{nN}(x)\,\}_{n=0}^{N}$ defined by (43), shown for the case $N = 10$ and $n = 0, 2, 4, 6, 8, 10$. For each n, $\mathcal{L}_n$ takes the value 1 at the nth collocation point $\widetilde{x}_n$ and 0 at each of the other collocation points. These interpolants constitute a set of non-orthogonal basis functions for the order-N Laguerre pseudospectral scheme.

It therefore follows that $\mathbf{D} \equiv [D_{mn}]$ is given by

$$D_{mn} \;=\; \widetilde{\mathcal{L}}_n'(\widetilde{x}_m), \qquad 0 \le m, n \le N. \tag{41}$$

To evaluate (41), we need an explicit formula for each polynomial $P_n(x)$. The classical Lagrange formula expresses P_n as a product of N factors, each of which is linear in x. However, following Shen (2000), we prefer the more concise formulae[1]

$$P_{nN}(x) \;=\; \frac{L_N(x) - L_{N+1}(x)}{(x - x_n)\, L_{N+1}(x_n)}\,, \tag{43a}$$

$$\mathcal{L}_{nN}(x) \;=\; \frac{\widehat{L}_N(x) - \widehat{L}_{N+1}(x)}{(x - x_n)\, \widehat{L}_{N+1}(x_n)}\,, \tag{43b}$$

$$\widetilde{\mathcal{L}}_{nN}(x) \;=\; \frac{\widetilde{L}_N(x) - \widetilde{L}_{N+1}(x)}{\kappa\,(x - \widetilde{x}_n)\, \widetilde{L}_{N+1}(\widetilde{x}_n)}\,. \tag{43c}$$

Differentiating the system (43), we obtain

$$P_{nN}'(x_m) \;=\; \begin{cases} [\,L_{N+1}(x_m)\,] \,/\, [\,(x_m - x_n)\, L_{N+1}(x_n)\,] & \text{if } m \ne n, \\ \frac{1}{2} & \text{if } m = n \ne 0, \\ -\frac{1}{2} N & \text{if } m = n = 0, \end{cases} \tag{44a}$$

$$\mathcal{L}_n'(x_m) \;=\; \begin{cases} [\,\widehat{L}_{N+1}(x_m)\,] \,/\, [\,(x_m - x_n)\, \widehat{L}_{N+1}(x_n)\,] & \text{if } m \ne n, \\ 0 & \text{if } m = n \ne 0, \\ -\frac{1}{2}(N+1) & \text{if } m = n = 0, \end{cases} \tag{44b}$$

$$D_{mn} \;=\; \begin{cases} [\,\widetilde{L}_{N+1}(\widetilde{x}_m)\,] \,/\, [\,(\widetilde{x}_m - \widetilde{x}_n)\, \widetilde{L}_{N+1}(\widetilde{x}_n)\,] & \text{if } m \ne n, \\ 0 & \text{if } m = n \ne 0, \\ -\frac{1}{2}\kappa(N+1) & \text{if } m = n = 0. \end{cases} \tag{44c}$$

Note that $\mathbf{D}$ has non-zero row-sums, and that its diagonal entries are zero (except in the case $\widetilde{x}_0 \equiv 0$). Neither property holds for conventional pseudospectral or finite-difference differentiation matrices. These diagonal entries indicate that the nth collocation point $\widetilde{x}_n$ maximizes the nth interpolant $\widetilde{\mathcal{L}}_n(x)$ (see Figure 6). The non-zero row sums indicate that $\mathbf{D}$ cannot differentiate the constant function $f(x) = c \ne 0$ (or any other function which does not decay asymptotically to zero). This should come as no surprise, since Laguerre functions themselves decay to zero!

[1] To show that $P_n(x_n) = 1$ for each n, apply l'Hôpital's Rule to (43a), together with the recursion (12b):

$$P_{nN}(x_n) \;=\; \frac{L_N'(x_n) - L_{N+1}'(x_n)}{L_{N+1}(x_n)} \;=\; \frac{L_N(x_n)}{L_N(x_n)} \;=\; 1. \tag{42}$$

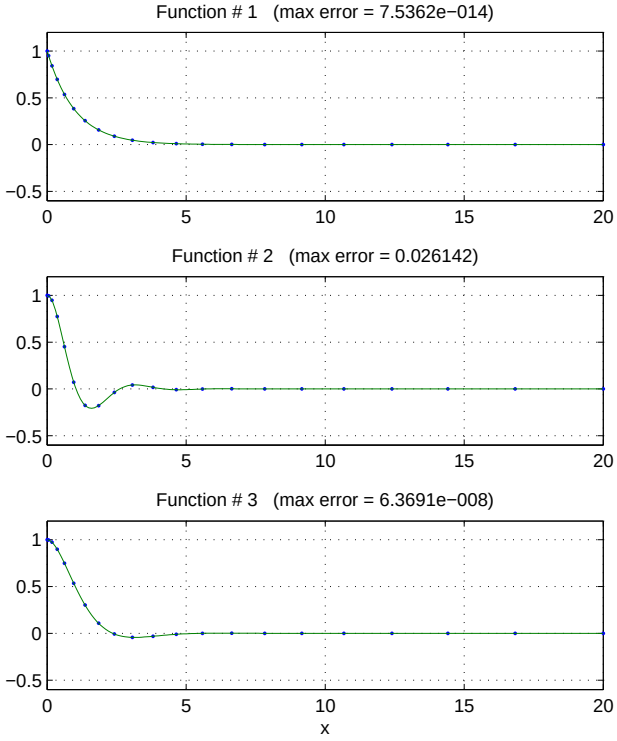

Figure 7: The three test problems from §2, but computed using the Laguerre pseudospectral scheme of §3. This figure was generated by the function `test_LPs` listed in Figure 14.

The Matlab program `test_LPs` listed in Figure 14 uses the above Laguerre pseudospectral scheme to solve and plot the three test problems of §2, as shown in Figure 7 for parameter values of $N = 20$ and $x_{\max} = 20$. For Problem 1, the computed function values are accurate to within machine precision; for Problem 3 they are accurate to seven digits; while for Problem 2, even two-figure accuracy is elusive. The issue of numerical convergence is explored further in Figure 9, which plots maximum numerical error as a function of spectral order N (left-hand axis) and domain size $x_{\max}$ (right-hand axis). In general, the approximation error decreases exponentially with N and $x_{\max}$ until machine precision is attained. When N is fairly small, however, numerical error may actually *increase* as $x_{\max}$ increases, indicating that the collocation-point spacing is too sparse. Overall, numerical convergence is very rapid for Problem 1 (ten-figure accuracy at $N = 10$–15), slow for Problem 2 (five-figure accuracy at $N = 40$–50), and fairly rapid for Problem 3 (ten-figure accuracy at $N = 25$–30).

Figure 10 illustrates the numerical convergence of the Galerkin scheme

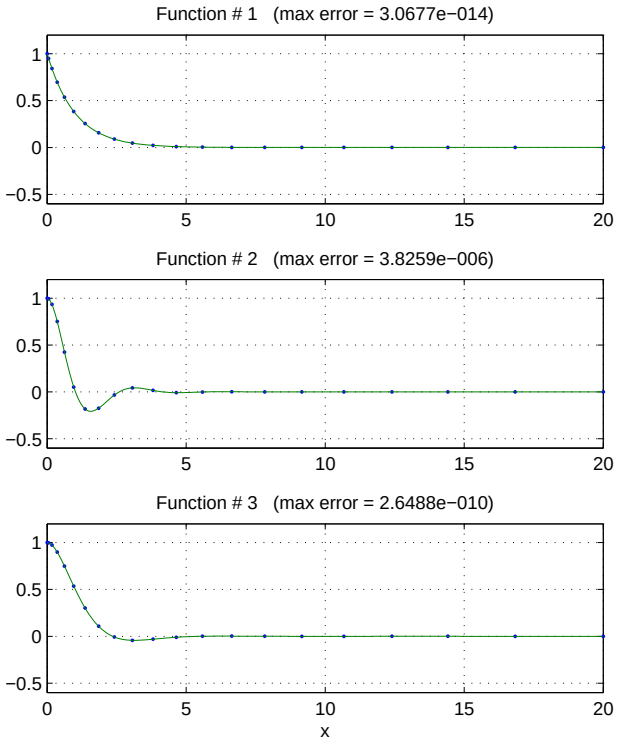

Figure 8: The three test problems from §2, but computed using the Laguerre collocation scheme of §4. This scheme is numerically equivalent to the Laguerre pseudospectral scheme, differing only in style. For each of the test cases, the collocation points span the interval $[0, 20]$, although the error statistic is computed over the reduced domain $[0, 10]$. This figure was generated by the function `test_Lcol` listed in Figure 15.

in N and κ. (For consistency with Figure 9, we have selected κ values equivalent to $x_{\max} = 5, 10, 15, 20, 25$ and 30 respectively.) It is clear that the Galerkin and pseudospectral schemes have generally similar convergence characteristics, although the Galerkin scheme is unmistakeably superior for Problem 2 and mildly superior for Problems 1 and 3.

4 A simple Laguerre collocation scheme

The Laguerre collocation scheme is a hybrid of the Galerkin and pseudospectral schemes. Ideally, it combines the flexibility of the pseudospectral scheme with the numerical robustness of the Galerkin scheme. As in the Galerkin case, the working variable for the collocation scheme is the vector $\mathbf{a}$ of spectral coefficients of $\widetilde{f}(x)$ (rather than the vector $\mathbf{f}$ of collocation-point values).

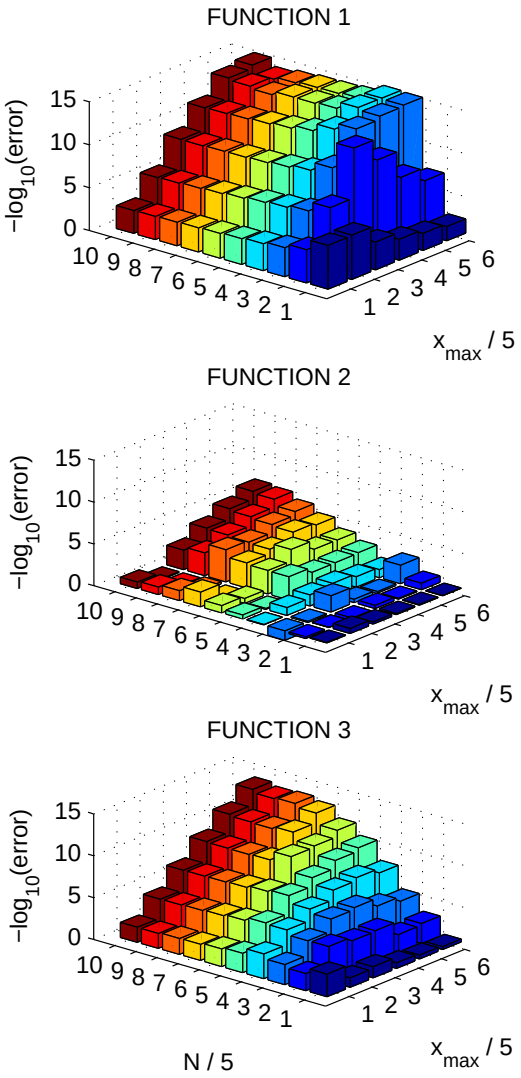

Figure 9: Maximum numerical error, as a function of spectral order N and computational sub-domain $[0, x_{\max}]$, when using a Laguerre pseudospectral scheme to compute the three test functions of §2. The left-hand axis in each figure corresponds to $N = 5, 10, ..., 50$, while the right-hand axis corresponds to $x_{\max} = 5, 10, ..., 30$.

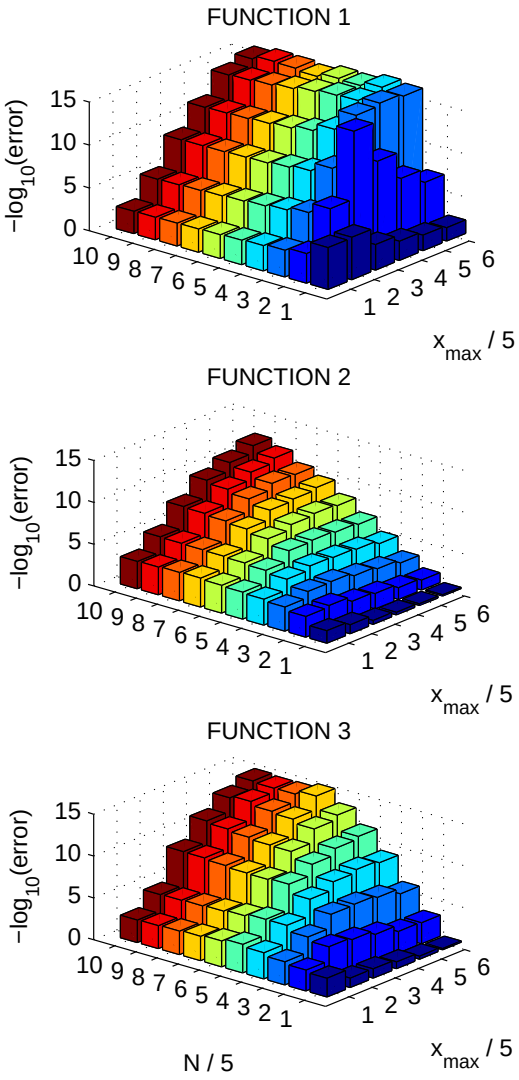

Figure 10: Maximum numerical error as a function of N and $x_{\max}$ for the Laguerre Galerkin scheme.

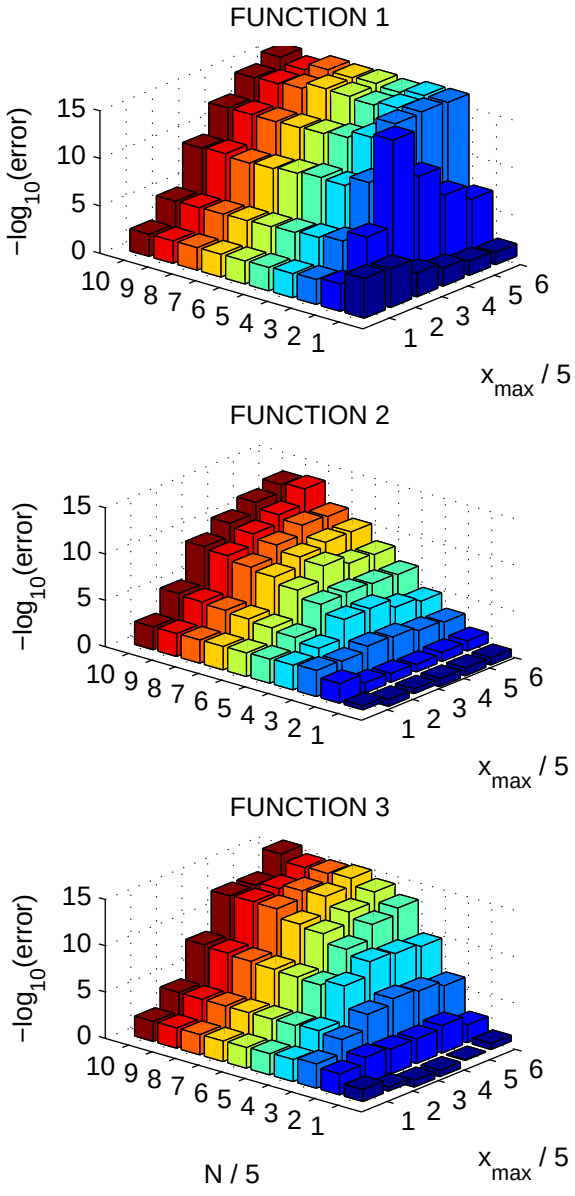

Figure 11: Numerical convergence of the Laguerre collocation scheme as a function of N and $x_{\max}$. As in previous figures, the left-hand axis corresponds to $N = 5, 10, ..., 50$, while the right-hand axis corresponds to $x_{\max} = 5, 10, ..., 30$.

As in the pseudospectral case, however, the collocation scheme attempts to zero the residual $R(x)$ at each of the $N+1$ collocation points (rather than attempting to zero the Laguerre coefficients $\{r_n\}$ of the residual).

The Matlab program `test_Lcol` listed in Figure 15 uses a Laguerre collocation scheme to solve the three test problems of §2 and §3. Figure 8 is the output of this program for spectral order $N = 20$ and scale factor κ equivalent to $x_{\max} = 20$. Figure 11 shows that the Galerkin and collocation schemes are virtually equivalent in accuracy and efficiency. In particular, the Laguerre collocation scheme out-performs its pseudospectral counterpart, even though the two schemes are notionally equivalent.

5 Bibliography

- Shen, J. Stable and efficient spectral methods in unbounded domains using Laguerre functions. *SIAM J. Num. Anal.* **38** (4), 1113–1133 (2000).
- Trefethen, L.N. *Spectral Methods in Matlab*, SIAM Publications (2000).
- Fornberg, B. *A Practical Guide to Pseudospectral Methods*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press (1996).
- Abramowitz, M. and Stegun, I. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, Dover Publications (1964).
- Kreyszig, E. *Introductory Functional Analysis with Applications*, John Wiley & Sons (1989).
- Weisstein, E. *CRC Concise Encyclopedia of Mathematics*, 2nd ed., CRC Press (Chapman & Hall) (2003).

```
function test_LG(N,rho)

  kappa = N/rho;
  xmax  = 8;                  % For plotting
  x     = xmax*(0:0.005:1)';  %   purposes only
  L     = LFuncs(x,N,kappa);
  D     = LG(N,kappa);
  I     = eye(N+1);
  rhs   = zeros(N+1,1);

  for i = 1:3

      switch i
          case 1,  exact = exp(-x);
                   R     = D^2 - I;
          case 2,  exact = exp(-x) .* (cos(2*x) + sin(2*x)/2);
                   R     = D^2 + 2*D + 5*I;
          case 3,  exact = exp(-x) .* (cos(x) + sin(x));
                   R     = D^4 + 4*I;
      end

      R(end,:) = 1;   rhs(end) = 1;

      if i > 1,  R(end-1,:) = sum(D);  rhs(end-1) = 0;   end

      a     = R\rhs;
      f     = L*a;
      err   = f - exact;
      emax  = max(abs(err));

      subplot(3,1,i)
      plot(x,f),  axis([0 xmax -0.6 1.2]),  grid on
      title(['Function # ', int2str(i), ...
             '   (max error = ', num2str(emax), ')' ])
  end

  xlabel x
```

Figure 12: The Matlab program used to generate Figure 4.

```
function D = LG(N,k)

  One = ones(N+1);
  Eye = eye(N+1);
  D   = kappa * (Eye/2 - triu(One));


function Lf = LFuncs(x,N,k)

% Lf = LFuncs(x,N,k)
%
% Scaled Laguerre functions of order 0 to N, evaluated at x.
% (The default scaling is k==1.)
% Lf is of size (Nx by N+1), where Nx is the length of x.

  if nargin < 3,  k = 1;  end

  xx = k*x;   Lf = diag(exp(-xx/2)) * LPolys(xx,N);


function L = LPolys(x,N)

% L = LPolys(x,N)
%
% Unscaled Laguerre polynomials of order 0 to N, evaluated at x.
% L is of size (Nx by N+1), where Nx is the length of x.

  x = x(:);  Nx = length(x);  L = zeros(Nx,N+1);

  f0 = ones(size(x));  L(:,1) = f0;   if N==0,  return, end;
  f1 = 1 - x;          L(:,2) = f1;   if N==1,  return, end;

  for n = 2:N
      f2 = ((2*n - 1 - x).*f1 - (n-1)*f0)/n;   L(:,n+1) = f2;
      f0 = f1;  f1 = f2;
  end
```

Figure 13: The auxiliary functions `LG` and `LFuncs` called by the Matlab program `test_LG` listed on the previous page.

```
function test_LPs(N,xmax)

  [D,x,kappa] = LPs(N,xmax);

  xx  = xmax*(0:0.005:1)';
  C   = interpL(x,xx,kappa);
  I   = eye(N+1);
  rhs = zeros(N+1,1);

  for i = 1:3

      switch i
          case 1,  exact = exp(-xx);
                   R     = D^2 - I;
          case 2,  exact = exp(-xx) .* (cos(2*xx) + sin(2*xx)/2);
                   R     = D^2 + 2*D + 5*I;
          case 3,  exact = exp(-xx) .* (cos(xx) + sin(xx));
                   R     = D^4 + 4*I;
      end

      R(1,:) = 0;   R(1,1) = 1;  rhs(1) = 1;

      if i > 1,  R(2,:) = D(1,:);   rhs(2) = 0;  end

      f    = R\rhs;
      ff   = C*f;
      err  = ff - exact;
      emax = max(abs(err(xx < 10)));

      subplot(3,1,i)
      plot(x,f,'.',xx,ff,'-'),  axis([0 xmax -0.6 1.2]),  grid on
      title(['Function # ', int2str(i), ...
             '   (max error = ', num2str(emax), ')' ])
  end

  xlabel x
```

Figure 14: The Matlab function used to generate Figure 7. The auxiliary functions `LPs` and `interpL` are omitted for clarity.

```
function test_Lcol(N,xmax)

[D,k,x] = LG2(N,xmax);
  xx    = xmax*(0:0.005:1)';
  Lx    = LFuncs(x,N,k);
  Lxx   = LFuncs(xx,N,k);
  I     = eye(N+1);
  rhs   = zeros(N+1,1);

  for i = 1:3

      switch i
          case 1,   exact = exp(-xx);
                    R     = Lx*(D^2 - I);
          case 2,   exact = exp(-xx) .* (cos(2*xx) + sin(2*xx)/2);
                    R     = Lx*(D^2 + 2*D + 5*I);
          case 3,   exact = exp(-xx) .* (cos(xx) + sin(xx));
                    R     = Lx*(D^4 + 4*I);
      end

      R(end,:) = 1;   rhs(end) = 1;

      if i > 1,  R(end-1,:) = sum(D);  rhs(end-1) = 0;   end

      a     = R\rhs;
      f     = Lx*a;
      ff    = Lxx*a;
      err   = ff - exact;
      emax  = max(abs(err(xx < 10)));

      subplot(3,1,i)
      plot(x,f,'.',xx,ff,'-'),  axis([0 xmax -0.6 1.2]),  grid on
      title(['Function # ', int2str(i), ...
             '   (max error = ', num2str(emax), ')' ])
  end

  xlabel x
```

Figure 15: The program used to generate Figure 8. (The auxiliary function `LG2` is a variant of `LG` in §2, with the scale factor κ determined implicitly by $x_{\max}$.)