6.1.4. A surprise: ill-conditioning

Stimulated by the accuracy achieved with $q = 3$, we compute the approximation with $q = 9$. We solve the linear algebraic system in two ways: first exactly using a symbolic manipulation package and then approximately using Gaussian elimination on a computer that uses roughly 16 digits. In general, the systems that come from the discretization of a differential equation are too large to be solved exactly and we are forced to solve them numerically with Gaussian elimination for example.

We obtain the following coefficients ξ_i in the two computations:

exact coefficients	approximate coefficients
.14068...	152.72...
.48864...	−3432.6...
.71125...	32163.2...
.86937...	−157267.8...
.98878...	441485.8...
1.0827...	−737459.5...
1.1588...	723830.3...
1.2219...	−385203.7...
1.2751...	85733.4...

We notice the huge difference, which makes the approximately computed U worthless. We shall now see that the difficulty is related to the fact that the system of equations (6.4) is *ill-conditioned* and this problem is exacerbated by using the standard polynomial basis $\{t^i\}_{i=0}^q$.

Problem 6.5. If access to a symbolic manipulation program and to numerical software for solving linear algebraic systems is handy, then compare the coefficients of U computed exactly and approximately for $q = 1, 2, ...$ until significant differences are found.

It is not so surprising that solving a system of equations $A\xi = b$, which is theoretically equivalent to inverting A, is sensitive to errors in the coefficients of A and b. The errors result from the fact that the computer stores only a finite number of digits of real numbers. This sensitivity is easily demonstrated in the solution of the 1×1 "system" of equations $ax = 1$ corresponding to computing the inverse $x = 1/a$ of a given real number $a \neq 0$. In Fig. 6.3, we plot the inverses of two numbers a_1 and a_2 computed from two approximations $\tilde{a}_1$ and $\tilde{a}_2$ of the same accuracy. We see that the corresponding errors in the approximations $\tilde{x} = 1/\tilde{a}_i$ of the exact values $x = 1/a_i$ vary greatly in the two cases, since

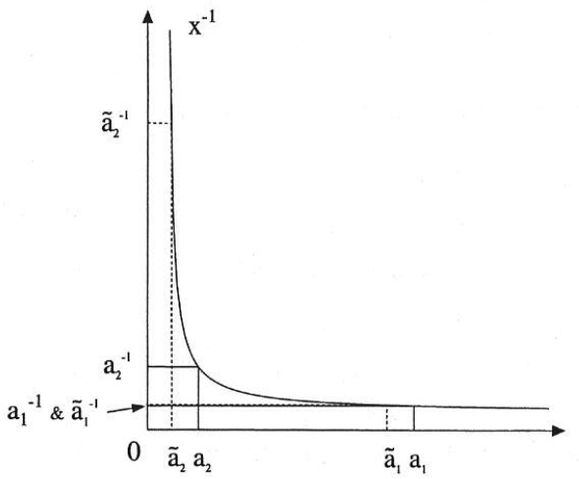

Figure 6.3: The sensitivity of the solution of $ax = 1$ to errors in a.

$1/a_i - 1/\tilde{a}_i = (\tilde{a}_i - a_i)/(a_i\tilde{a}_i)$. The closer a_i is to zero the more sensitive is the solution $1/a_i$ to errors in a_i. This expresses that computing $1/a$ is *ill-conditioned* when a is close to zero.

In general, the solution of $Ax = b$ is sensitive to errors in the entries of A when A is "close" to being non-invertible. Recall that a matrix is non-invertible if one row (or column) is a linear combination of the other rows (or columns). In the example of computing the coefficients of the Galerkin approximation with $q = 9$ above, we can see that there might be a problem if we look at the coefficient matrix A:

0.167	0.417	0.550	0.633	0.690	0.732	0.764	0.789	0.809
0.0833	0.300	0.433	0.524	0.589	0.639	0.678	0.709	0.735
0.0500	0.233	0.357	0.446	0.514	0.567	0.609	0.644	0.673
0.0333	0.190	0.304	0.389	0.456	0.509	0.553	0.590	0.621
0.0238	0.161	0.264	0.344	0.409	0.462	0.506	0.544	0.576
0.0179	0.139	0.233	0.309	0.371	0.423	0.467	0.505	0.538
0.0139	0.122	0.209	0.280	0.340	0.390	0.433	0.471	0.504
0.0111	0.109	0.190	0.256	0.313	0.360	0.404	0.441	0.474
0.00909	0.0985	0.173	0.236	0.290	0.338	0.379	0.415	0.447

which is nearly singular since the entries in some rows and columns are quite close. On reflection, this is not surprising because the last two rows are given by $\int_0^1 R(U,t)t^8\,dt$ and $\int_0^1 R(U,t)t^9\,dt$, respectively, and t^8 and t^9 look very similar on $[0,1]$. We plot the two basis functions in Fig. 6.4.