Spreadsheets in Education (eJSiE)

Volume 9 | Issue 3 Article 1

9-26-2016

Spreadsheet Implementation of Numerical and Analytical Solutions to Some Classical Partial Differential Equations

Mark A. Lau
Universidad del Turabo, mlau@suagm.edu

Spreadsheet Implementation of Numerical and Analytical Solutions to Some Classical Partial Differential Equations

Abstract
This paper presents the implementation of numerical and analytical solutions of some of the classical partial differential equations using Excel spreadsheets. In particular, the heat equation, wave equation, and Laplace's equation are presented herein since these equations have well known analytical solutions. The numerical solutions can be easily obtained once the differential equations are discretized via finite differences and then using cell formulas to implement the resulting recursive algorithms and other iterative methods such as the successive over-relaxation (SOR) method. The graphing capabilities of spreadsheets can be exploited to enhance the visualization of the solutions to these equations. Furthermore, using Visual Basic for Applications (VBA) can greatly facilitate the implementation of the analytical solutions to these equations, and in the process, one obtains Fourier series approximations to functions governing initial and/or boundary conditions.

Keywords
Heat equation, wave equation, Laplace equation, partial differential equations, finite differences, successive over-relaxation (SOR) method

Cover Page Footnote
The author gratefully acknowledges support from the Universidad del Turabo.

1. Introduction

In science and engineering the dynamical behavior of systems in space and time is modeled by ordinary differential equations or partial differential equations. Systems in which the variable of interest (e.g., temperature) depends on more than one independent variable (e.g., location and time) are mathematically modeled by partial differential equations (PDEs). The vast majority of PDEs require the use of computers for their numerical solution. Only a handful of PDEs are amenable to analytical solutions through methods such as separation of variables, characteristics, or change of variables.

This paper presents some of the classical PDEs that appear in numerous science and engineering applications. More specifically, the heat equation, wave equation, and Laplace's equation are presented herein. These equations have well known analytical solutions which are obtainable through the method of separation of variables (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011). Moreover, because of linearity and homogeneity in the boundary conditions, the solutions to these equations naturally give rise to Fourier series.

Electronic spreadsheets have been used to model many PDEs arising from science and engineering. For instance, Arganbright (1985) used *VisiCalc* to create an animated model of the two-dimensional heat flow on a plate by using circular references; Orvis (1997) discusses heat flow, along with a broad range of other science and engineering applications; Neuwirth and Arganbright (2004) present both one- and two-dimensional heat flow problems via animated discrete *Excel* models that display the flow through graphs and the spreadsheet grid display with conditional formatting.

This paper presents the implementation of both numerical and analytical solutions of the heat equation, wave equation, and Laplace's equation using Excel spreadsheets. The presentation is inspired largely on online Excel tutorials used in classroom instruction at the University of British Columbia in Canada (Piccolo, n.d., a,b). In this sense, the author does not claim a great deal of originality; nevertheless the author believes that this paper presents a new perspective insofar as it expands on the work of Piccolo (n.d., a,b) by accommodating slight generalizations whenever feasible and, more important, using Visual Basic for Applications (VBA) to implement the analytical solutions and the Fourier series that represent initial and/or boundary conditions. The spreadsheet implementation of the solution to the Laplace equation presented in this paper draws from standard discretization models and uses the *successive over-relaxation* (SOR) method to iteratively find approximate solutions to the resulting system of linear algebraic equations (Gutierrez, 2009).

The paper is organized as follows. Section 2 presents the one-dimensional heat equation with two illustrative examples (a heat conducting bar with fixed temperatures at both ends, and a heat conducting bar with insulated ends); the numerical solution of the discretized heat equation is implemented using simple cell formulas, and the analytical solution is implemented using VBA. Following the same structure, Sections 3 presents the one-dimensional wave equation along with two examples (a string with clamped ends, and a string with zero-derivative constraints at both ends). Section 4 discusses spreadsheet implementations of numerical and analytical solutions, including the SOR method, to the two-dimensional Laplace equation in Cartesian coordinates; a rectangular, heat conductive plate with Dirichlet boundary conditions is presented as an illustrative example. Finally, concluding remarks are given in Section 5.

2. One-dimensional heat equation

Case 1. Heat conducting bar with fixed temperatures at both ends

Consider a bar of length L whose cross sectional area is negligible compared to its longitudinal dimension. Then, the temperature u of the bar at a distance x from the left end of the bar ($x = 0$) and at time t is governed by the *heat equation* (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011)

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < L, \;\; t > 0 \tag{1a}$$

subject to the boundary conditions

$$u(0,t) = U_0 \quad \text{and} \quad u(L,t) = U_L, \quad \forall t > 0 \tag{1b}$$

and the initial condition

$$u(x,0) = f(x), \quad 0 < x < L. \tag{1c}$$

In (1a)–(1c), c^2 denotes the thermal diffusivity, U_0 the temperature at the left end of the bar, U_L the temperature at the right end of the bar, and $f(x)$ the initial temperature distribution along the bar. In some sense, the heat equation being discussed here represents a slight generalization of the work done by Piccolo (n.d., a), where both ends of the bar are kept at zero temperature.

Numerical solution of (1a)–(1c)

The idea is to obtain a discretized version of the heat equation (1a). This can be achieved by approximating the partial derivatives with difference quotients and establishing the relationships between u at (x,t) and its neighboring values a distance Δx apart and at a time Δt later (Piccolo, n.d., a). In particular, the derivative with respect to time will be approximated with a *forward difference*, i.e.,

$$\frac{\partial u}{\partial t}(x,t) \approx \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} \tag{2}$$

and the derivative with respect to space will be approximated with a *central difference*, i.e.,

$$\frac{\partial^2 u}{\partial x^2}(x,t) \approx \frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{\Delta x^2}. \tag{3}$$

Substituting (2) and (3) into (1a), and rearranging terms, yields the heat equation in discretized form

$$u(x,t+\Delta t) = u(x,t) + \gamma[u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)] \tag{4}$$

where $\gamma = c^2\left(\frac{\Delta t}{\Delta x^2}\right)$.

To propose an algorithm suitable for computer implementation, the spatial interval and the time interval are subdivided into increments of size Δx and Δt, respectively. That is, each sample point in the spatial interval is obtained from

$$x_n = x_{n-1} + \Delta x, \qquad \text{for } n = 1,2,\dots,N, \text{ with } x_0 = 0 \text{ and } x_N = L$$

and each sample time in the time interval $[0,T]$ is obtained from

$$t_k = t_{k-1} + \Delta t, \qquad \text{for } k = 1,2,\dots,M, \text{ with } t_0 = 0 \text{ and } t_M = T$$

where T is the length of the time interval of interest.

With the spatial and time interval partitioning described above, and denoting the value of u at the nth sample point x_n and at the kth sample time t_k as $u(x_n, t_k) = u_n^k$, and noting that

$$u(x_n + \Delta x, t_k) = u(x_{n+1}, t_k) = u_{n+1}^k,$$

$$u(x_n - \Delta x, t_k) = u(x_{n-1}, t_k) = u_{n-1}^k,$$

$$u(x_n, t_k + \Delta t) = u(x_n, t_{k+1}) = u_n^{k+1}$$

the discretized heat equation in (4) can be rewritten in the more compact form

$$u_n^{k+1} = u_n^k + \gamma[u_{n+1}^k - 2u_n^k + u_{n-1}^k]. \tag{5}$$

Equation (5) gives the approximate value of u at point x_n and at time t_{k+1}, which is calculated from the values of u at three adjacent points x_{n+1}, x_n, x_{n-1} at the preceding sample time t_k.

An important consideration in the implementation of any algorithm for numerical computation is convergence. To ensure the convergence of (5), the user must choose Δx and Δt sufficiently small for reasonable resolution and verify that the quantity $\gamma = c^2(\Delta t/\Delta x^2)$ be much less than unity.

EXAMPLE 1. Solve the heat equation

$$\frac{\partial u}{\partial t} = 0.25\frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 1, \ \ t > 0$$

subject to the boundary conditions

$$u(0,t) = 10 \quad \text{and} \quad u(1,t) = 30, \qquad \forall t > 0$$

and the initial condition

$$u(x,0) = f(x) = \begin{cases} 10 + 60x, & 0 < x < 0.5; \\ 50 - 20x, & 0.5 \le x < 1. \end{cases}$$

Here, $c^2 = 0.25$, $L = 1$, $U_0 = 10$, and $U_L = 30$. Choosing $\Delta x = 0.05$ and $\Delta t = 0.005$ will give $\gamma = 0.5 < 1$. To obtain a numerical solution to this example using an Excel spreadsheet, follow these steps:

1) Enter the values of c^2, U_0, U_L, Δx, and Δt. The value of γ can be computed with a cell formula according to $c^2(\Delta t/\Delta x^2)$. The initial setup might look like the portion of the spreadsheet shown in Figure 1. In this case, the value of c^2 has been entered in cell H10, U_0 in D5, U_L in G5, Δx in B10, and Δt in E10. The value of γ is located in cell L10 and contains the formula =H10*(E10/B10^2).

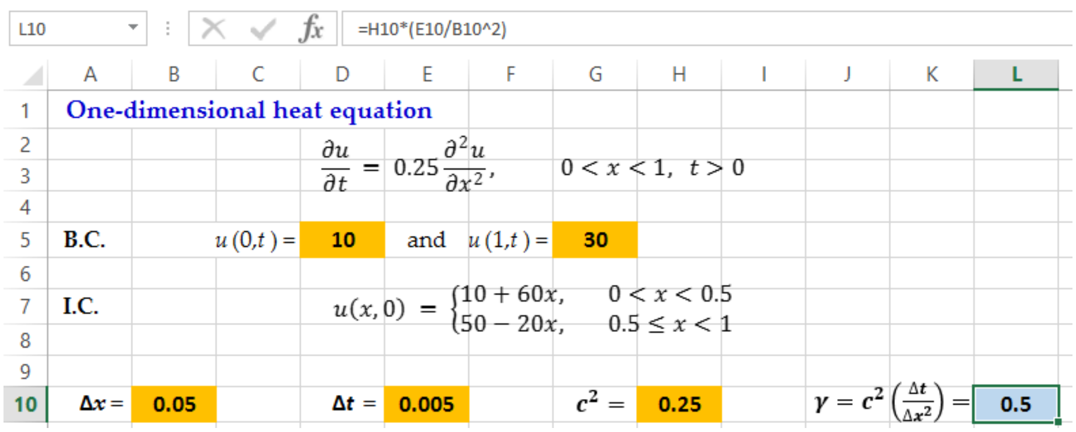


Figure 1: Initial setup for the numerical solution of Example 1.

2) Generate the sample points of the spatial interval. Enter 0 (zero) in cell C13. In cell D13, type the formula =C13+$B$10 (note the absolute reference to cell B10 that contains Δx). Copy the formula in cell D13 onto the cell range E13:W13. The result might resemble that of Figure 2.

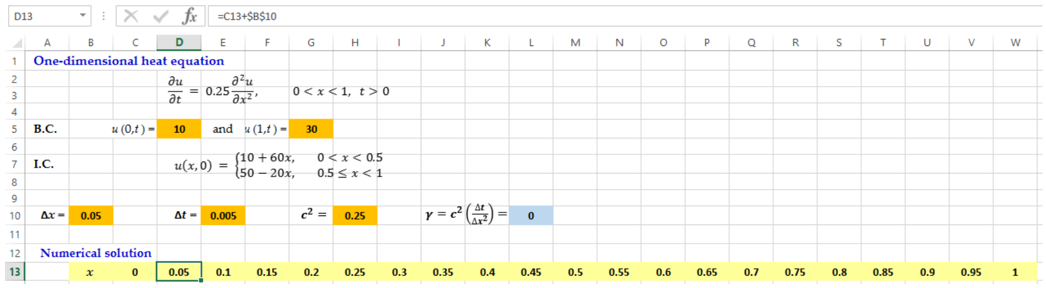


Figure 2: Spreadsheet after constructing the space interval.

3) Generate the sample times of the time interval. Enter 0 (zero) in cell A15. In cell A16, type the formula =A15+$E$10 (note the absolute reference to cell E10 that contains Δt). Copy the formula in cell A16 onto the cell range A17:A670. The result might resemble that of Figure 3. Only the first 20 rows are shown in the figure; cell A670 will display 3.275. The time $t = 3.275$ simply means that the length T of the time interval of interest is 3.275 (the reason for this choice will become apparent in step 6).

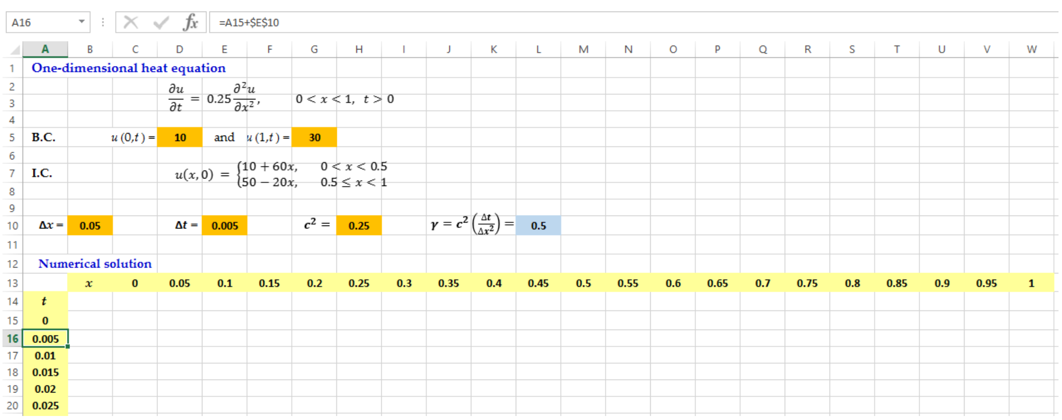


Figure 3: Spreadsheet after constructing the time interval.

4) Impose the boundary conditions located in cells D5 and G5. This is accomplished by typing the formula =$D$5 (note the absolute reference) in cell C15 and copying it onto the cell range C16:C670; this accounts for the (fixed) temperature at the left end of the heat conducting bar. Similarly, typing the formula =$G$5 in cell W15

and then copying it onto the cell range W16:W670 will account for the temperature at the right end of the bar. After completing this step, the spreadsheet will look like Figure 4 (again, only the first 20 rows are shown).

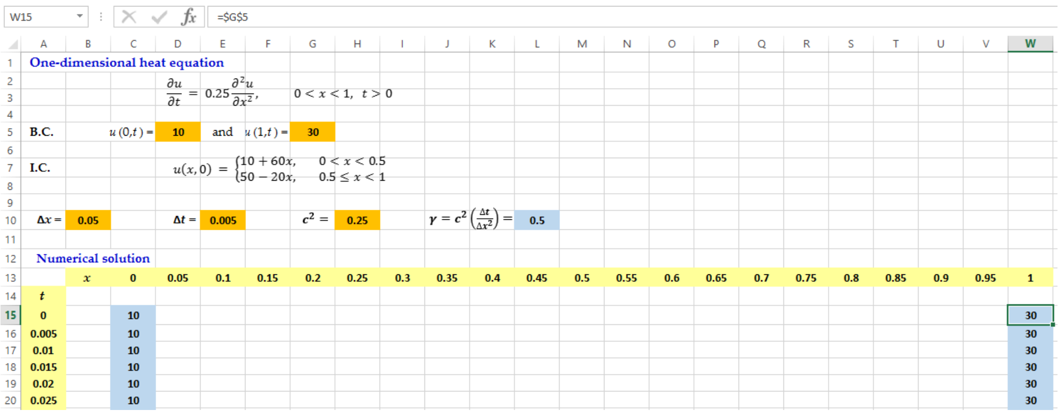


Figure 4: Spreadsheet after imposing the boundary conditions.

5) Impose the initial condition $u(x, 0) = f(x)$. To accomplish this, type the formula =IF(D13<0.5,10+60*D13,50-20*D13) in cell D15, and copy it onto the cell range E15:V15. The result is shown in Figure 5 (only the first 20 rows are displayed).

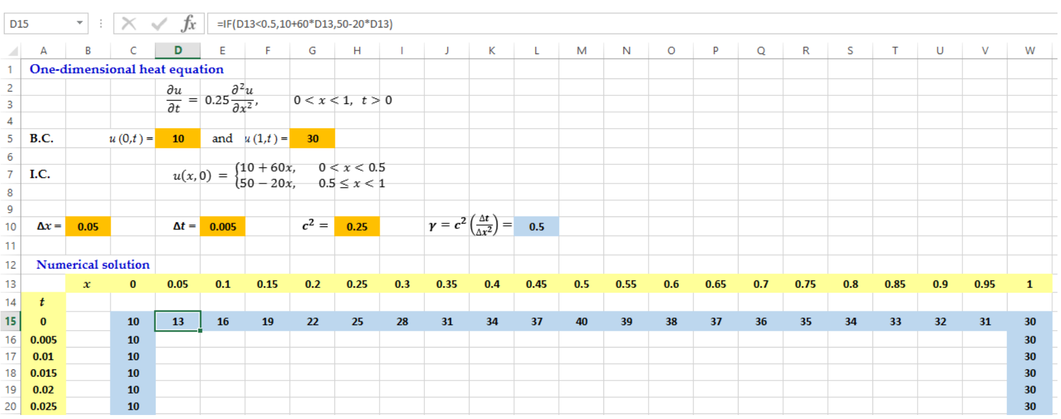
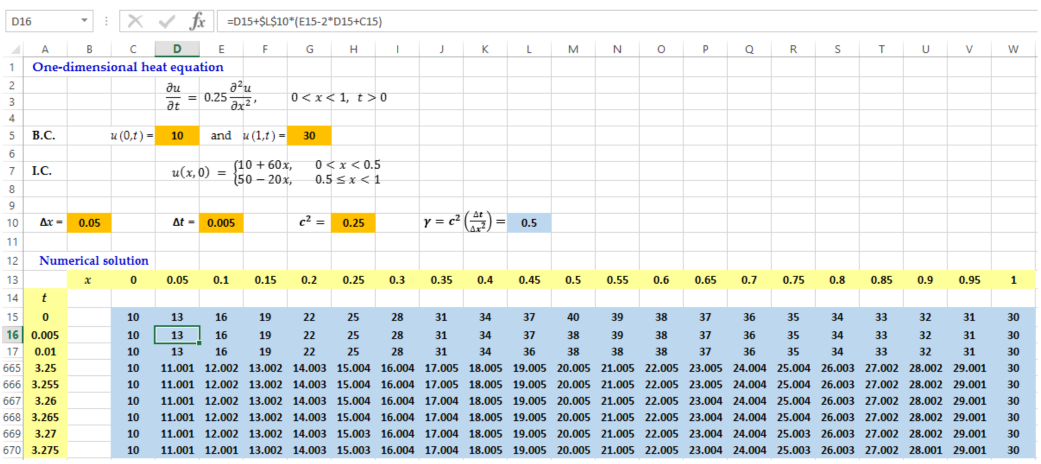


Figure 5: Spreadsheet after imposing the initial condition.

6) Implement the discretized heat equation (5). To do this, type the formula `=D15+$L$10*(E15-2*D15+C15)` in cell `D16` (note the absolute reference to cell `L10` for the value of γ). Copy the formula onto the cell range `D16:V670`. Figure 6 shows the final result (rows 18 through 664 have been hidden to prevent the figure from being too unwieldy). For the choice of $t = 3.275$, it can be seen that the numerical solution has converged to two exact decimal places for all sample points in the x interval.



D16 =D15+$L$10*(E15-2*D15+C15)

One-dimensional heat equation

$\frac{\partial u}{\partial t} = 0.25\frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1, \ t > 0$

B.C. $u(0,t) =$ 10 and $u(1,t) =$ 30

I.C. $u(x,0) = \begin{cases} 10 + 60x, & 0 < x < 0.5 \\ 50 - 20x, & 0.5 \le x < 1 \end{cases}$

$\Delta x =$ 0.05 $\Delta t =$ 0.005 $c^2 =$ 0.25 $\gamma = c^2\left(\frac{\Delta t}{\Delta x^2}\right) =$ 0.5

Numerical solution

	t \ x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
15	0	10	13	16	19	22	25	28	31	34	37	40	39	38	37	36	35	34	33	32	31	30
16	0.005	10	13	16	19	22	25	28	31	34	37	38	39	38	37	36	35	34	33	32	31	30
17	0.01	10	13	16	19	22	25	28	31	34	36	38	38	38	37	36	35	34	33	32	31	30
665	3.25	10	11.001	12.002	13.002	14.003	15.004	16.004	17.005	18.005	19.005	20.005	21.005	22.005	23.005	24.004	25.004	26.003	27.002	28.002	29.001	30
666	3.255	10	11.001	12.002	13.002	14.003	15.004	16.004	17.005	18.005	19.005	20.005	21.005	22.005	23.005	24.004	25.004	26.003	27.002	28.002	29.001	30
667	3.26	10	11.001	12.002	13.002	14.003	15.004	16.004	17.004	18.005	19.005	20.005	21.005	22.005	23.004	24.004	25.004	26.003	27.002	28.002	29.001	30
668	3.265	10	11.001	12.002	13.002	14.003	15.004	16.004	17.004	18.005	19.005	20.005	21.005	22.005	23.004	24.004	25.004	26.003	27.002	28.002	29.001	30
669	3.27	10	11.001	12.002	13.002	14.003	15.003	16.004	17.004	18.005	19.005	20.005	21.005	22.005	23.004	24.004	25.003	26.003	27.002	28.002	29.001	30
670	3.275	10	11.001	12.001	13.002	14.003	15.003	16.004	17.004	18.005	19.005	20.005	21.005	22.005	23.004	24.004	25.003	26.003	27.002	28.001	29.001	30

Figure 6: Numerical solution to Example 1.

The graphing capabilities of Excel can be used to visualize the solution just obtained. For example, Figure 7 shows the temperature distribution along the heat conducting bar at three different instants, namely, $t = 0$, $t = 0.25$, and $t = 3.275$. The graph for $t = 0$ corresponds to the initial condition $u(x, 0) = f(x)$. The Fourier series representation of $f(x)$ will be discussed shortly in the analytical solution of the heat equation.

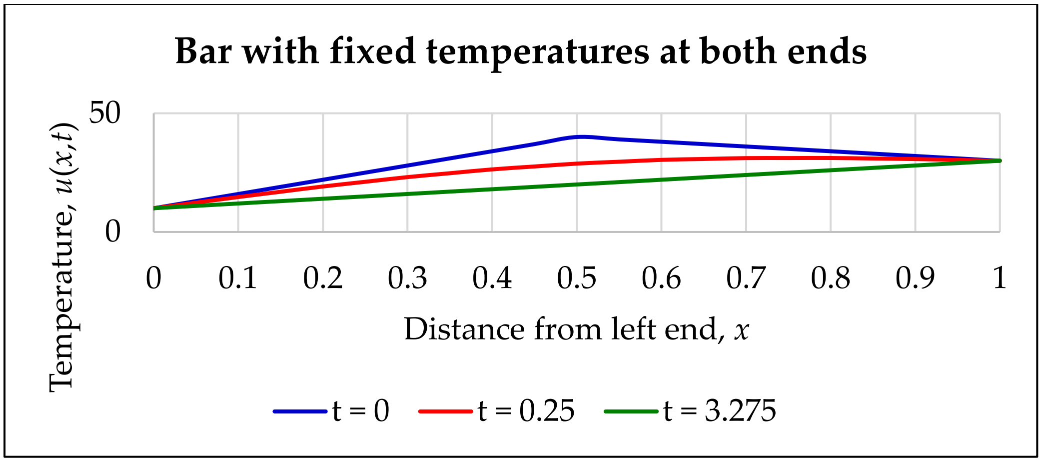


Figure 7: Temperature distribution in heat conducting bar.

Analytical solution of (1a)–(1c)

The analytical solution of (1a)–(1c) is given by (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011)

$$u(x,t) = v(x) + w(x,t) \tag{6a}$$

where

$$v(x) = U_0 + \frac{U_L - U_0}{L}x, \tag{6b}$$

$$w(x,t) = \sum_{n=1}^{\infty} B_n \sin\left(\frac{\pi n x}{L}\right) e^{-\lambda_n^2 t} \qquad \left(\lambda_n = \frac{\pi c n}{L}\right), \tag{6c}$$

$$B_n = \frac{2}{L}\int_0^L [f(x) - v(x)]dx. \tag{6d}$$

In (6a), $v(x)$ represents the steady-state solution and $w(x,t)$ the transient solution to the heat equation problem defined in (1a)–(1c).

To implement the analytical solution (6a)–(6d) on an Excel spreadsheet, a user-defined function can be created using Visual Basic for Applications (VBA). By doing so the user can greatly simplify the typing of cell formulas and streamline the presentation of the spreadsheet; otherwise, auxiliary worksheets would be required to compute the Fourier coefficients B_n and the series solution $w(x,t)$ with somewhat cumbersome cascading cell referencing.

In what follows the key steps for implementing the analytical solution to the problem in Example 1 are given.

1) Launch the Excel VBA editor (accessible from the Developer tab in the ribbon; if not visible, add it by customizing the ribbon from the Excel options of the File menu in Excel 2010 or later). Create the user-defined function heateq1 by typing the code shown in Figure 8. The function requires two input parameters, x and t, which represent a sample point in the space interval and a sample time. The function computes the Fourier coefficients B_n given in (6d) and the series $w(x,t)$ in (6c); these quantities correspond to the variables Bn and S in the code. For this example, the Fourier coefficients were computed manually from Equation (6d) to yield $B_n = \frac{160\sin(0.5\pi n)}{\pi^2 n^2}$, $n = 1, 2, \ldots$

```
Public Const PI = 3.14159265358979, Nmax = 25
Public Function heateq1(x, t)
    c = Sqr(0.25)
    S = 0
    For n = 1 To Nmax
        Bn = 160 * Sin(0.5 * PI * n) / (PI ^ 2 * n ^ 2)
        lambda = PI * c * n
        S = S + Bn * Sin(PI * n * x) * Exp(-lambda ^ 2 * t)
    Next n
    heateq1 = S
End Function
```

Figure 8: VBA code for the user-defined function heateq1.

2) Retaining the same structure of the numerical solution shown in Figure 6, create a new table on another part of the worksheet, preferably maintaining the same row numbers and the values of Δx and Δt used in the numerical solution for easy side-by-side comparison between solutions. For instance, creating the table in the cell range Y13:AU670 will suffice (AA13:AU13 would contain the sample points x_n, Y15:Y670 would contain the sample times t_k, and

AA15 would be the first programmed cell). To populate the table with sample values of the analytical solution, type the formula =$D$5+($G$5-$D$5)*AA$13+heateq1(AA$13,$Y15) in cell AA15. Then copy the formula onto the cell range AA15:AU670. The result is shown in Figure 9 (again, rows 18 through 664 have been hidden for easy viewing).

AA15 | =$D$5+($G$5-$D$5)*AA$13+heateq1(AA$13,$Y15)

	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU
12	Analytical solution																						
13		x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14	t																						
15	0		10	12.99	16.012	18.996	21.991	25.017	27.99	30.989	34.037	36.965	39.688	38.965	38.037	36.989	35.99	35.017	33.991	32.996	32.012	30.99	30
16	0.005		10	13	16	19	22	25	28	30.998	33.966	36.667	38.404	38.667	37.966	36.998	36	35	34	33	32	31	30
17	0.01		10	13	16	19	22	25	27.996	30.966	33.799	36.201	37.743	38.201	37.799	36.966	35.996	35	34	33	32	31	30
665	3.25		10	11.001	12.002	13.002	14.003	15.004	16.004	17.005	18.005	19.005	20.005	21.005	22.005	23.005	24.004	25.004	26.003	27.002	28.002	29.001	30
666	3.255		10	11.001	12.002	13.002	14.003	15.004	16.004	17.005	18.005	19.005	20.005	21.005	22.005	23.005	24.004	25.004	26.003	27.002	28.002	29.001	30
667	3.26		10	11.001	12.002	13.002	14.003	15.004	16.004	17.005	18.005	19.005	20.005	21.005	22.005	23.005	24.004	25.004	26.003	27.002	28.002	29.001	30
668	3.265		10	11.001	12.002	13.002	14.003	15.004	16.004	17.005	18.005	19.005	20.005	21.005	22.005	23.005	24.004	25.004	26.003	27.002	28.002	29.001	30
669	3.27		10	11.001	12.002	13.002	14.003	15.004	16.004	17.005	18.005	19.005	20.005	21.005	22.005	23.005	24.004	25.004	26.003	27.002	28.002	29.001	30
670	3.275		10	11.001	12.002	13.002	14.003	15.004	16.004	17.004	18.005	19.005	20.005	21.005	22.005	23.004	24.004	25.004	26.003	27.002	28.002	29.001	30

Figure 9: Analytical solution of Example 1.

The analytical solution shown in Figure 9 agrees well with the numerical solution given in Figure 6. However, comparing the rows corresponding to $t = 3.275$ (row 670) in both figures, it can be said that the analytical solution converges more slowly than the numerical solution; a closer inspection of row 670 in Figure 9 reveals that not all temperatures at the sample points in the space interval have attained the same level of precision when rounded to two decimal places, as was the case in the numerical solution at this very same iteration step ($t = 3.275$).

In addition, observe that the analytical solution (6a) implemented by the formula =$D$5+($G$5-$D$5)*AA$13+heateq1(AA$13,$Y15) in cell AA15 has two parts: $D$5+($G$5-$D$5)*AA$13 implements the steady-state solution $v(x)$ in (6b), and heateq1(AA$13,$Y15) implements the transient solution $w(x,t)$ in (6c) – these two contributions produce the desired analytical solution (6a). Also notice that the boundary conditions did not have to be copied directly from cells D5 and G5 onto AA15:AA670 (for $x = 0$) and AU15:AU670 (for $x = 1$), as was the case in the numerical solution; likewise, the initial condition $f(x)$ did not have to be programmed directly in the cell range AA15:AU15 (for $t = 0$). The reason that the boundary conditions and the initial condition were not entered directly from the input data is due to the fact that these conditions are already incorporated in the analytical solution (6a)–(6d). That is why a single formula in cell AA15 could be replicated to fill the entire table (AA15:AU670) with sample values of the analytical solution. As Figure 9 shows, the boundary conditions are met exactly; the initial condition, however, is met approximately due to the truncation in the series (6c) when evaluated by the function heateq1. Indeed, the values in the cell range AA15:AU15 of Figure 9 constitute a Fourier series approximation to the initial condition $f(x)$ in the cell range C15:W15 of Figure 6; this comparison is illustrated in Figure 10 and it shows that the Fourier series approximates the initial condition very well.

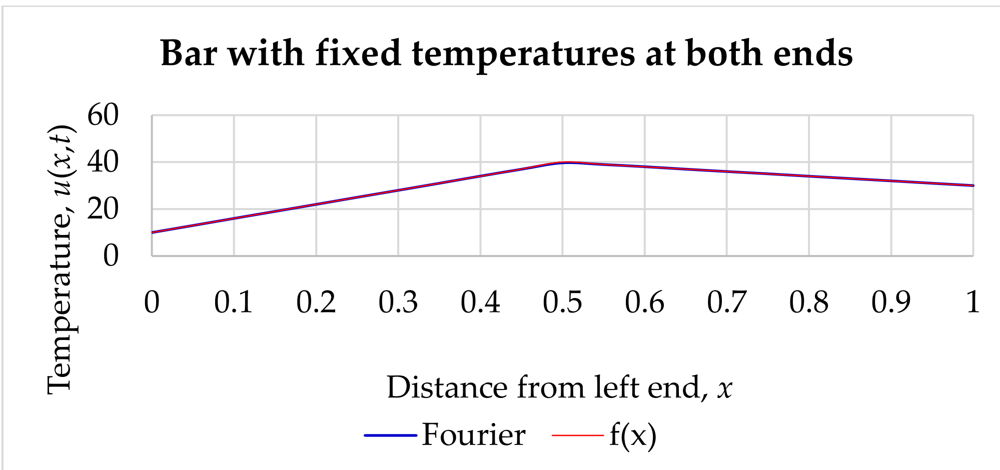


Figure 10: Fourier series approximation to the initial condition $f(x)$ of Example 1.

Caution. A couple of remarks are in order:

- Although the term *analytical solution* is used in the present discussion, it is not meant to imply *exact*. As mentioned before, the series in (6c) had to be truncated to 25 terms (see the constant Nmax in the header of the code shown in Figure 8) for computer evaluation.
- The temperatures in cells D5 and G5 can be changed by the user and the numerical solution can still be correct. This would not be the

case in the analytical solution because the expressions that appear in the VBA code given in Figure 8 were programmed after manual computation of the Fourier coefficients B_n given by Equation (6d). Thus, the formulas in the VBA code are somewhat ad hoc. To make the code more flexible at handling other user-supplied temperatures would require a more parametric style of programming at the expense of readability; in this paper the author opted for clarity of exposition and simple coding.

Case 2. Heat conducting bar with insulated ends

Instead of having fixed temperatures at both ends of the bar, now consider the bar with both ends perfectly insulated (no heat escapes to the external environment). This situation is modeled by (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011)

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < L, \ \ t > 0 \tag{7a}$$

subject to the boundary conditions

$$\frac{\partial u}{\partial x}(0,t) = 0 \quad \text{and} \quad \frac{\partial u}{\partial x}(L,t) = 0, \qquad \forall t > 0 \tag{7b}$$

and the initial condition

$$u(x,0) = f(x), \qquad 0 < x < L. \tag{7c}$$

Compared with Equations (1a)–(1c), the boundary conditions are changed in (7a)–(7c). The problem being presented here is a slight variation of the one proposed by Piccolo (n.d., a), where the left end of the bar is kept at zero temperature and the right end is insulated.

Numerical solution of (7a)–(7c)

Equation (5) is still the key formula for computing sample values of the solution to the heat equation problem (7a)–(7c). At the boundaries, where $x_0 = 0$ and $x_N = L$, evaluation of (5) produces the following equations:

$$u_0^{k+1} = u_0^k + \gamma[u_1^k - 2u_0^k + u_{-1}^k] \tag{8a}$$

and

$$u_N^{k+1} = u_N^k + \gamma\left[u_{N+1}^k - 2u_N^k + u_{N-1}^k\right]. \quad (8b)$$

Unfortunately, the preceding equations require estimates of u_{-1}^k and u_{N+1}^k which fall outside the domain of definition (i.e., $x_{-1}, x_{N+1} \notin [0, L]$). To circumvent this difficulty, the values of u_{-1}^k and u_{N+1}^k will be inferred from the derivatives (*not* actual temperatures) at the boundaries. These derivatives may be approximated by a forward difference or a central difference. Approximating the derivatives by a central difference may prove to be more convenient for spreadsheet implementation as no extra columns would be required to insert the cell formulas for the boundaries. More specifically, the derivatives at $x_0 = 0$ and $x_N = L$ will be approximated by the central differences

$$\frac{\partial u}{\partial x}(x_0, t_k) \approx \frac{u(x_0 + \Delta x, t_k) - u(x_0 - \Delta x, t_k)}{2\Delta x} = \frac{u(x_1, t_k) - u(x_{-1}, t_k)}{2\Delta x} = \frac{u_1^k - u_{-1}^k}{2\Delta x}$$

and

$$\frac{\partial u}{\partial x}(x_N, t_k) \approx \frac{u(x_N + \Delta x, t_k) - u(x_N - \Delta x, t_k)}{2\Delta x} = \frac{u(x_{N+1}, t_k) - u(x_{N-1}, t_k)}{2\Delta x} = \frac{u_{N+1}^k - u_{N-1}^k}{2\Delta x}.$$

Using the preceding central differences and taking into account the given boundary conditions, it can be deduced that

$$\frac{\partial u}{\partial x}(x_0, t_k) = \frac{\partial u}{\partial x}(0, t_k) = 0 \approx \frac{u_1^k - u_{-1}^k}{2\Delta x} \quad \rightarrow \quad u_{-1}^k = u_1^k$$

and

$$\frac{\partial u}{\partial x}(x_N, t_k) = \frac{\partial u}{\partial x}(L, t_k) = 0 \approx \frac{u_{N+1}^k - u_{N-1}^k}{2\Delta x} \quad \rightarrow \quad u_{N+1}^k = u_{N-1}^k.$$

Having determined that $u_{-1}^k = u_1^k$ and $u_{N+1}^k = u_{N-1}^k$, Equations (8a) and (8b) can now be rewritten for the boundaries to obtain

$$u_0^{k+1} = u_0^k + 2\gamma[u_1^k - u_0^k] \quad (9a)$$

and

$$u_N^{k+1} = u_N^k + 2\gamma\left[u_{N-1}^k - u_N^k\right]. \quad (9b)$$

With Equation (5) for the interior points of the space interval and (9a)–(9b) for the boundary points, together with the given initial condition, all elements are in place to construct a spreadsheet model for the numerical solution of (7a)–(7c), as illustrated by the following example.

EXAMPLE 2. Solve the heat equation

$$\frac{\partial u}{\partial t} = 0.25\frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 1,\ t > 0$$

subject to the boundary conditions

$$\frac{\partial u}{\partial x}(0,t) = 0 \quad \text{and} \quad \frac{\partial u}{\partial x}(1,t) = 0, \qquad \forall t > 0$$

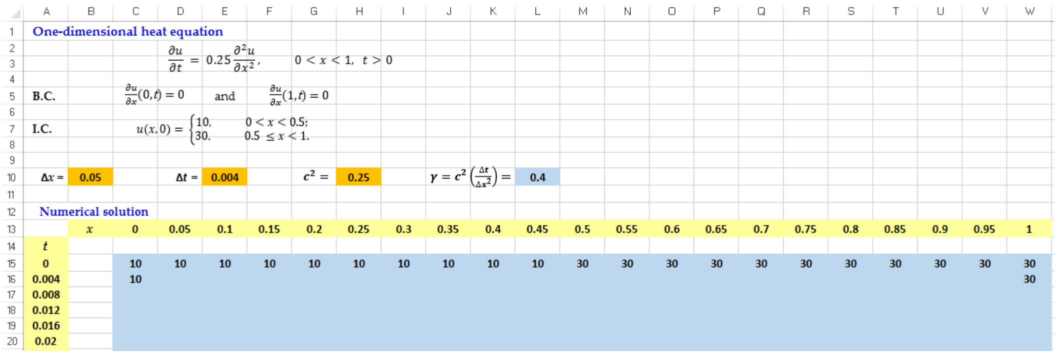
and the initial condition

$$u(x,0) = f(x) = \begin{cases} 10, & 0 < x < 0.5; \\ 30, & 0.5 \le x < 1. \end{cases}$$

The basic steps for finding the numerical solution to this example using a spreadsheet are outlined below.

1) Create a table with the space and time intervals. For instance, the table can span the cell range A13:W1040 (see Figure 11; only the first 20 rows are shown). The range C13:W13 contains the sample points in x with $\Delta x = 0.05$; the range A15:A1040 contains the samples in t with $\Delta t = 0.004$. To enter the initial condition $u(x,0) = f(x)$, type =IF(C13<0.5,10,30) in cell C15 and then copy the formula onto D15:W15.

   To insert the boundary conditions into the spreadsheet, type =C15+2*$L$10*(D15-C15) in cell C16 to code Equation (9a), and in cell W16 type =W15+2*$L$10*(V15-W15) to code Equation (9b). The initial setup is shown in Figure 11.



One-dimensional heat equation

$\frac{\partial u}{\partial t} = 0.25\frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1,\ t > 0$

B.C. $\frac{\partial u}{\partial x}(0,t) = 0$ and $\frac{\partial u}{\partial x}(1,t) = 0$

I.C. $u(x,0) = \begin{cases} 10, & 0 < x < 0.5; \\ 30, & 0.5 \leq x < 1. \end{cases}$

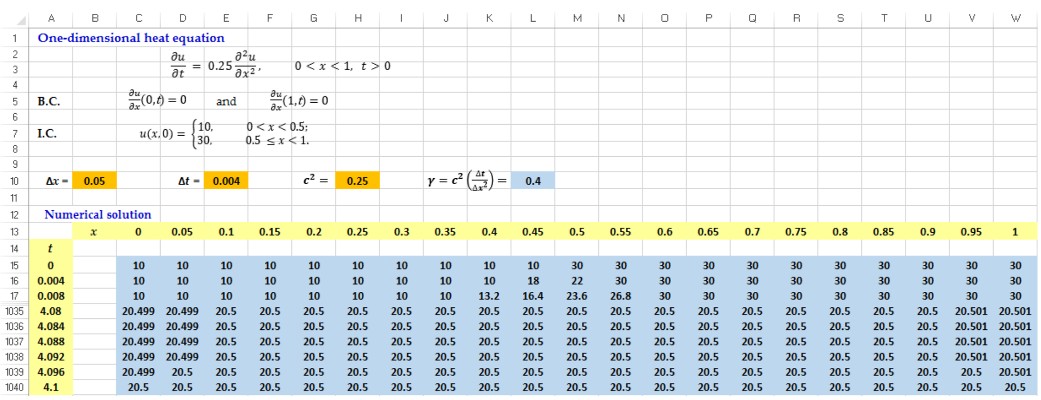
$\Delta x =$ 0.05 $\Delta t =$ 0.004 $c^2 =$ 0.25 $\gamma = c^2\left(\frac{\Delta t}{\Delta x^2}\right) =$ 0.4

Numerical solution

t \ x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
0	10	10	10	10	10	10	10	10	10	10	30	30	30	30	30	30	30	30	30	30	30
0.004	10																				30
0.008																					
0.012																					
0.016																					
0.02																					

Figure 11: Initial setup for the numerical solution of Example 2.

2) Code Equation (5) for the interior points of the space interval. To do this, type =D15+$L$10*(E15-2*D15+C15) in cell D16, and then copy it onto the cell range E16:V16. This will generate the numerical approximation at the next sample time (values in range C16:W16). To propagate the solution in time, copy the cell range C16:W16 onto C17:W1040. The final result is shown in Figure 12 (rows 18 through 1034 have been hidden).



One-dimensional heat equation

$\frac{\partial u}{\partial t} = 0.25\frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1,\ t > 0$

B.C. $\frac{\partial u}{\partial x}(0,t) = 0$ and $\frac{\partial u}{\partial x}(1,t) = 0$

I.C. $u(x,0) = \begin{cases} 10, & 0 < x < 0.5; \\ 30, & 0.5 \leq x < 1. \end{cases}$

$\Delta x =$ 0.05 $\Delta t =$ 0.004 $c^2 =$ 0.25 $\gamma = c^2\left(\frac{\Delta t}{\Delta x^2}\right) =$ 0.4

Numerical solution

Row	t \ x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
15	0	10	10	10	10	10	10	10	10	10	10	30	30	30	30	30	30	30	30	30	30	30
16	0.004	10	10	10	10	10	10	10	10	10	18	22	30	30	30	30	30	30	30	30	30	30
17	0.008	10	10	10	10	10	10	10	10	13.2	16.4	23.6	26.8	30	30	30	30	30	30	30	30	30
1035	4.08	20.499	20.499	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.501	20.501
1036	4.084	20.499	20.499	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.501	20.501
1037	4.088	20.499	20.499	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.501	20.501
1038	4.092	20.499	20.499	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.501	20.501
1039	4.096	20.499	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.501
1040	4.1	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5	20.5

Figure 12: Numerical solution of Example 2.

Observe that the numerical solution converges to a constant temperature of $u = 20.5$ (see row 1040 of the spreadsheet in Figure 12). As will be discussed shortly, the actual steady-state temperature is $u_{ss} = 20$. That is, for the chosen values of Δx and Δt there is a 2.5% approximation error in the steady-state temperature. The reader can try to refine the numerical solution by decreasing the values of Δx and Δt, say $\Delta x = 0.02$ and $\Delta t = 0.0005$ (recall that $\gamma = c^2(\Delta t/\Delta x^2)$ must be much less than unity for convergence); for this new choice of Δx and Δt, the reader would obtain an approximate steady-state temperature of $u = 20.2$ (a 1% approximation error at the expense of having a table with more than 50 columns and 8,000 rows!).

Figure 13 shows temperature distributions in the bar at three different times (strictly speaking, the blue line representing the initial temperature distribution $u(x, 0) = f(x)$ at $t = 0$ should look like a piecewise constant function; however, because of the value of the spatial resolution $\Delta x = 0.05$ and the graphical rendition of the XY Scatter line chart, the jump discontinuity at $x = 0.5$ is not visible since Excel attempts to interpolate smooth lines between points). It can be clearly seen that as time progresses, the temperature in the bar reaches a uniform value (20.5 in this example).

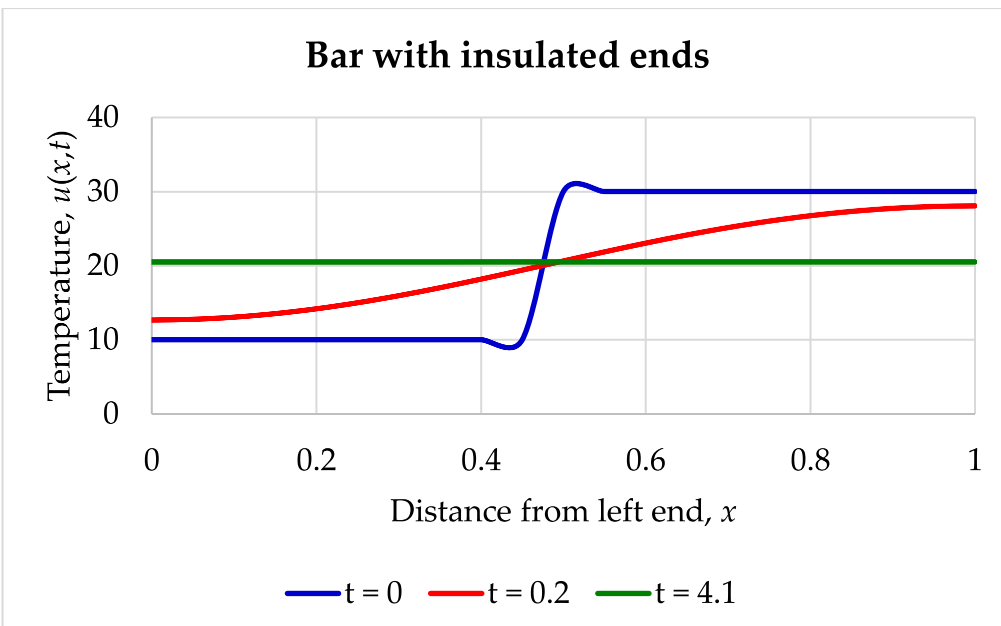


Figure 13: Temperature distribution in heat conducting bar of Example 2.

Analytical solution of (7a)–(7c)

The analytical solution to the heat equation problem (7a)–(7c) is given by (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011)

$$u(x,t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{\pi n x}{L}\right) e^{-\lambda_n^2 t} \qquad \left(\lambda_n = \frac{\pi c n}{L}\right), \tag{10a}$$

$$A_0 = \frac{1}{L}\int_0^L f(x)dx, \tag{10b}$$

$$A_n = \frac{2}{L}\int_0^L f(x)\cos\left(\frac{\pi n x}{L}\right)dx\,, \qquad n = 1, 2, \ldots \tag{10c}$$

Evaluating the Fourier coefficients given by Equations (10b) and (10c) with the given $f(x)$ results in

$$A_0 = 20 \qquad \text{and} \qquad A_n = -\frac{40\sin(0.5\pi n)}{\pi n}, \;\; n = 1, 2, \ldots$$

With these Fourier coefficients, one can proceed to program the analytical solution in the VBA editor of Excel as shown in Figure 14. The figure shows the code for the user-defined function heateq2, which can be appended to the existing code shown in Figure 8.

```
Public Function heateq2(x, t)
    c = Sqr(0.25)
    A0 = 20
    S = 0
    For n = 1 To Nmax
        An = -40 * Sin(0.5 * PI * n) / (PI * n)
        lambda = PI * c * n
        S = S + An * Cos(PI * n * x) * Exp(-lambda ^ 2 * t)
    Next n
    heateq2 = A0 + S
End Function
```

Figure 14: VBA code for the user-defined function heateq2.

Taking a cue from Figure 9, the analytical solution can be obtained as follows:

1) Lay out a table that spans the cell range `Y13:AU1040`. Distribute the sample points x_n along `AA13:AU13`, and the sample times t_k along `Y15:Y1040`.

2) In cell `AA15` type `=heateq2(AA$13,$Y15)` and copy the formula onto `AA15:AU1040`. The result is shown in Figure 15 (rows 18 through 1034 have been hidden). Although the solution is being referred to as "analytical", the values displayed in the figure are just approximations as a result of the truncation of the series in Equation (10a) when coded in VBA.

AA15 f_x =heateq2(AA$13,$Y15)

	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ
12	Analytical solution											
13		*x*	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45
14	*t*											
15	0		9.7555	10.144	10.082	9.7376	10.239	10.013	9.6526	10.495	9.8441	8.9082
16	0.004		9.9997	10	10	9.9997	10	10	9.9997	10.008	10.254	12.635
17	0.008		10	10	10	10	10	10.001	10.016	10.177	11.138	14.292
1035	4.08		19.999	19.999	19.999	20	20	20	20	20	20	20
1036	4.084		19.999	19.999	19.999	20	20	20	20	20	20	20
1037	4.088		19.999	19.999	19.999	20	20	20	20	20	20	20
1038	4.092		19.999	19.999	20	20	20	20	20	20	20	20
1039	4.096		19.999	19.999	20	20	20	20	20	20	20	20
1040	4.1		19.999	19.999	20	20	20	20	20	20	20	20

	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU
12											
13	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14											
15	20	31.092	30.156	29.505	30.347	29.987	29.761	30.262	29.918	29.856	30.244
16	20	27.365	29.746	29.992	30	30	30	30	30	30	30
17	20	25.708	28.862	29.823	29.984	29.999	30	30	30	30	30
1035	20	20	20	20	20	20	20	20	20.001	20.001	20.001
1036	20	20	20	20	20	20	20	20	20.001	20.001	20.001
1037	20	20	20	20	20	20	20	20	20.001	20.001	20.001
1038	20	20	20	20	20	20	20	20	20	20.001	20.001
1039	20	20	20	20	20	20	20	20	20	20.001	20.001
1040	20	20	20	20	20	20	20	20	20	20.001	20.001

Figure 15: Analytical solution of Example 2.

Once again, notice that the initial condition $u(x, 0) = f(x)$ was not programmed directly on the spreadsheet (the cell rage `AA15:AU15` corresponding to $t = t_0 = 0$). The initial condition is already incorporated by the Fourier coefficients A_0 and A_n, as shown by Equations (10b) and (10c). Doing so will allow the user to obtain a Fourier series approximation to $f(x)$, as illustrated in Figure 16. It can be seen that the Gibbs phenomenon is more apparent (see the overshoot at the point of discontinuity $x = 0.5$ in the blue curve; ideally, the red curve representing $f(x)$ should look like a step function but the limitation of the graphical rendering of the `XY scatter` plot of Excel impedes the appropriate display of the jump discontinuity).

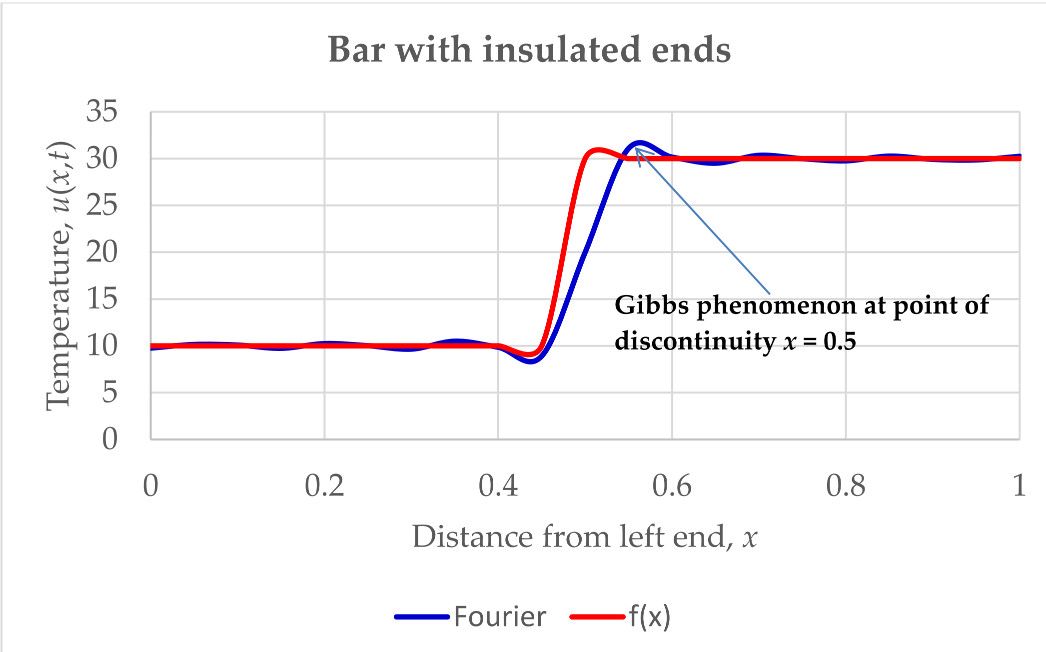


Figure 16: Fourier series approximation to the initial condition $f(x)$ of Example 2.

3. One-dimensional wave equation

Case 1. Taut string clamped at endpoints

The transverse displacement of a stretched vibrating string, clamped at both ends, is governed by the *wave equation* (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011)

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < L, \;\; t > 0 \tag{11a}$$

subject to the boundary conditions

$$u(0,t) = 0 \quad \text{and} \quad u(L,t) = 0, \qquad \forall t > 0 \tag{11b}$$

and the initial conditions

$$u(x,0) = f(x) \quad \text{and} \quad \frac{\partial u}{\partial t}(x,0) = g(x), \qquad 0 < x < L \tag{11c}$$

where

u = (transverse) displacement,
x = distance from the left end of the string,
t = time,
c = wave velocity,
L = length of the string,
$f(x)$ = initial displacement,
$g(x)$ = initial velocity.

Numerical solution of (11a)–(11c)

To obtain a numerical solution, the partial derivatives in Equation (11a) can be approximated by their central differences, namely,

$$\frac{\partial^2 u}{\partial t^2} \approx \frac{u(x,t+\Delta t) - 2u(x,t) + u(x,t-\Delta t)}{\Delta t^2}$$

and

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{\Delta x^2}.$$

Substituting the preceding finite differences into Equation (11a) and rearranging terms will result in

$$u(x,t+\Delta t) = 2u(x,t) - u(x,t-\Delta t) + \eta[u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)] \quad (12)$$

where $\eta = \left(\frac{c\Delta t}{\Delta x}\right)^2$.

Alternatively, rewriting Equation (12) in terms of sample points in space and time yields

$$u(x_n,t_{k+1}) = 2u(x_n,t_k) - u(x_n,t_{k-1}) + \eta[u(x_{n+1},t_k) - 2u(x_n,t_k) + u(x_{n-1},t_k)]$$

or, after rearranging terms,

$$u(x_n,t_{k+1}) = \eta u(x_{n+1},t_k) + 2(1-\eta)u(x_n,t_k) + \eta u(x_{n-1},t_k) - u(x_n,t_{k-1}). \quad (13)$$

Equation (13) gives approximate values to the solution of the wave equation (11a). The boundary conditions in (11b) can be entered directly into a spreadsheet. The same can be said of the initial displacement $f(x)$ in (11c) – this will give the first row of values corresponding to $t_0 = 0$. The initial velocity $g(x)$ in (11c) requires approximation of the derivative at $t = 0$. Using a central difference to approximate this derivative, i.e.,

$$\frac{\partial u}{\partial t} \approx \frac{u(x,t+\Delta t) - u(x,t-\Delta t)}{2\Delta t}$$

which evaluated at $t = 0$ will result in

$$\frac{u(x,\Delta t) - u(x,-\Delta t)}{2\Delta t} = g(x)$$

or

$$u(x,-\Delta t) = u(x,\Delta t) - 2\Delta t g(x).$$

In the preceding equation $\Delta t = t_1$, while $-\Delta t = t_{-1}$. The sample time t_{-1} may be regarded as a fictitious time occurring exactly one step Δt prior to $t_0 = 0$. Thus, in terms of space-time samples, the preceding equation can be written in the alternative form

$$u(x_n, t_{-1}) = u(x_n, t_1) - 2\Delta t g(x_n). \tag{14}$$

Equation (13) is also valid when $t_0 = 0$ (i.e., $k = 0$) resulting in

$$u(x_n, t_1) = \eta u(x_{n+1}, 0) + 2(1-\eta)u(x_n, 0) + \eta u(x_{n-1}, 0) - u(x_n, t_{-1}) \tag{15}$$

and taking into account (14), after rearranging terms, Equation (15) becomes

$$u(x_n, t_1) = \frac{\eta}{2}u(x_{n+1}, 0) + (1-\eta)u(x_n, 0) + \frac{\eta}{2}u(x_{n-1}, 0) + \Delta t g(x_n). \tag{16}$$

Equation (16) produces the second row of values for $t_1 = \Delta t$ by incorporating the initial velocity $g(x)$ from (11c). With the discretized equations (13) and (16) in place, the following example outlines the steps for constructing a spreadsheet model for the numerical solution of (11a)–(11c).

EXAMPLE 3. Solve the wave equation

$$\frac{\partial^2 u}{\partial t^2} = 4\frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 1, \ \ t > 0$$

subject to the boundary conditions

$$u(0,t) = 0 \quad \text{and} \quad u(1,t) = 0, \qquad \forall t > 0$$

and the initial conditions

$$u(x,0) = f(x) = 0.1 - 0.2|x - 0.5| \text{ and } \frac{\partial u}{\partial t}(x,0) = g(x) = x(1-x), \qquad 0 < x < 1.$$

To construct the spreadsheet model, perform the following steps:

1) Lay out a table on the cell range A13:W421 with the samples of the space interval in C13:W13 and the time interval in A15:A421 (see Figure 17). Enter the boundary conditions by typing 0 (zero) in C15:C421 for $x = 0$, and 0 (zero) in W15:W421 for $x = 1$. For the initial displacement $f(x)$, type the formula =0.1-0.2*ABS(D13-0.5) in cell D15 and copy it onto the cell range E15:V15 (the first row for $t_0 = 0$ is now completed). To account for the initial velocity $g(x)$, code Equation (16) by typing the formula =($L$10/2)*E15+(1-$L$10)*D15+($L$10/2)*C15+$E$10*D13*(1-D13) in cell D16 and then copying it onto E16:V16 (the second row for $t_1 = \Delta t = 0.01$ is now completed). After completing all of these preliminary steps, the initial setup should look like the one shown in Figure 17 (only the first 20 rows are displayed).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	One-dimensional wave equation																						
2			$\frac{\partial^2 u}{\partial t^2} = 4\frac{\partial^2 u}{\partial x^2}$,			$0 < x < 1$,	$t > 0$																
3																							
4																							
5	B.C.		$u(0,t) = 0$		and	$u(1,t) = 0$																	
6																							
7	I.C.	$u(x,0) = 0.1 - 0.2\|x - 0.5\|$				and	$\frac{\partial u}{\partial t}(x,0) = x(1-x)$																
8																							
9																							
10	$\Delta x =$	0.05		$\Delta t =$	0.01		$c^2 =$	4		$\eta = \left(\frac{c\Delta t}{\Delta x}\right)^2 =$		0.16											
11																							
12	Numerical solution																						
13		x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14	t																						
15	0		0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.09	0.08	0.07	0.06	0.05	0.04	0.03	0.02	0.01	0
16	0.01		0	0.0105	0.0209	0.0313	0.0416	0.0519	0.0621	0.0723	0.0824	0.0925	0.1009	0.0925	0.0824	0.0723	0.0621	0.0519	0.0416	0.0313	0.0209	0.0105	0
17	0.02		0																				0
18	0.03		0																				0
19	0.04		0																				0
20	0.05		0																				0

Figure 17: Initial setup for the numerical solution of Example 3.

2) To fill in the rest of the table, code Equation (13) by typing the formula =$L$10*E16+2*(1-$L$10)*D16+$L$10*C16-D15 in cell D17, and then copying it onto the cell range D17:V421. The end result is shown in Figure 18 (rows 18 through 415 have been hidden).

D17 f_x =$L$10*E16+2*(1-$L$10)*D16+$L$10*C16-D15

One-dimensional wave equation

$\frac{\partial^2 u}{\partial t^2} = 4\frac{\partial^2 u}{\partial x^2}$, $0 < x < 1$, $t > 0$

B.C. $u(0,t) = 0$ and $u(1,t) = 0$

I.C. $u(x,0) = 0.1 - 0.2|x - 0.5|$ and $\frac{\partial u}{\partial t}(x,0) = x(1-x)$

$\Delta x =$ 0.05 $\Delta t =$ 0.01 $c^2 =$ 4 $\eta = \left(\frac{c\Delta t}{\Delta x}\right)^2 =$ 0.16

Numerical solution

t \ x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
0	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.09	0.08	0.07	0.06	0.05	0.04	0.03	0.02	0.01	0
0.01	0	0.0105	0.0209	0.0313	0.0416	0.0519	0.0621	0.0723	0.0824	0.0925	0.1009	0.0925	0.0824	0.0723	0.0621	0.0519	0.0416	0.0313	0.0209	0.0105	0
0.02	0	0.0109	0.0218	0.0325	0.0432	0.0537	0.0642	0.0745	0.0848	0.0947	0.0991	0.0947	0.0848	0.0745	0.0642	0.0537	0.0432	0.0325	0.0218	0.0109	0
4.01	0	0.0056	0.0141	0.0286	0.0416	0.0559	0.065	0.0753	0.0853	0.0885	0.0876	0.0885	0.0853	0.0753	0.065	0.0559	0.0416	0.0286	0.0141	0.0056	0
4.02	0	0.0064	0.0148	0.0297	0.0427	0.0569	0.0662	0.0761	0.0869	0.0924	0.0911	0.0924	0.0869	0.0761	0.0662	0.0569	0.0427	0.0297	0.0148	0.0064	0
4.03	0	0.0075	0.0165	0.0305	0.0439	0.0572	0.0675	0.0771	0.0877	0.0952	0.0949	0.0952	0.0877	0.0771	0.0675	0.0572	0.0439	0.0305	0.0165	0.0075	0
4.04	0	0.0088	0.0191	0.0312	0.0452	0.057	0.0687	0.0782	0.0879	0.0968	0.0989	0.0968	0.0879	0.0782	0.0687	0.057	0.0452	0.0312	0.0191	0.0088	0
4.05	0	0.0104	0.0219	0.0322	0.0461	0.0568	0.0695	0.0794	0.088	0.0973	0.1022	0.0973	0.088	0.0794	0.0695	0.0568	0.0461	0.0322	0.0219	0.0104	0
4.06	0	0.0121	0.0246	0.0337	0.0465	0.0569	0.0699	0.0803	0.0883	0.0971	0.1039	0.0971	0.0883	0.0803	0.0699	0.0569	0.0465	0.0337	0.0246	0.0121	0

Figure 18: Numerical solution of Example 3.

Figure 19 shows the transverse displacement of the vibrating string at three different times. The oscillatory nature of the string displacement can be discerned from the figure.

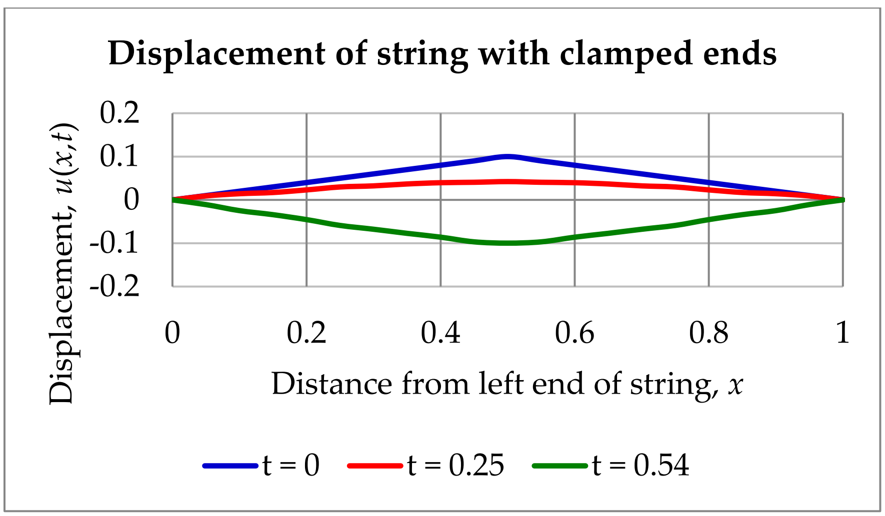


Figure 19: Transverse displacement for the vibrating string of Example 3.

Analytical solution of (11a)–(11c)

The analytical solution to the wave equation (11a)–(11c) is given by (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011)

$$u(x,t) = \sum_{n=1}^{\infty} [B_n \cos(\lambda_n t) + B_n^* \sin(\lambda_n t)] \sin\left(\frac{\pi n x}{L}\right) \qquad \left(\lambda_n = \frac{\pi c n}{L}\right) \tag{17a}$$

$$B_n = \frac{2}{L}\int_0^L f(x) \sin\left(\frac{\pi n x}{L}\right) dx, \qquad n = 1, 2, \ldots \tag{17b}$$

$$B_n^* = \frac{2}{\pi c n}\int_0^L g(x) \sin\left(\frac{\pi n x}{L}\right) dx, \qquad n = 1, 2, \ldots \tag{17c}$$

The bulk of the work is to determine the Fourier coefficients B_n and B_n^* in accordance to Equations (17b) and (17c). For the given initial conditions, namely, $f(x) = 0.1 - 0.2|x - 0.5|$ and $g(x) = x(1 - x)$, it is straightforward to find that

$$B_n = \frac{0.8(-1)^{n+1}}{\pi^2(2n-1)^2} \quad \text{and} \quad B_n^* = \frac{4}{\pi^4(2n-1)^4}, \quad n = 1, 2, \ldots$$

and that the series solution given by Equation (17a) is

$$u(x,t) = \sum_{n=1}^{\infty} \left[\frac{0.8(-1)^{n+1}}{\pi^2(2n-1)^2} \cos(2\pi(2n-1)t) + \frac{4}{\pi^4(2n-1)^4} \sin(2\pi(2n-1)t)\right] \sin(\pi(2n-1)x)$$

The preceding equations will be used to write a VBA code for implementing the analytical solution of Example 3. Figure 20 shows the VBA code for the user-defined function waveeq1.

```
Public Function waveeq1(x, t)
    c = Sqr(4)
    S = 0
    For n = 1 To Nmax
        Bn = (0.8 * (-1) ^ (n + 1)) / (PI ^ 2 * (2 * n - 1) ^ 2)
        Bns = 4 / (PI ^ 4 * (2 * n - 1) ^ 4)
        lambda = PI * c * (2 * n - 1)
        Bncos = Bn * Cos(lambda * t)
        Bnsin = Bns * Sin(lambda * t)
        S = S + (Bncos + Bnsin) * Sin(PI * (2 * n - 1) * x)
    Next n
    waveeq1 = S
End Function
```

Figure 20: VBA code for user-defined function waveeq1.

With the user-defined function in hand, one can proceed to construct a spreadsheet model for the analytical solution to the vibrating string problem of Example 3. For instance, one can create a table that occupies the cell range Y13:AU421 (see Figure 21). Then, distribute the sample points of the space interval along AA13:AU13 and the sample times along Y15:Y421. Finally, type the formula =waveeq1(AA$13,$Y15) in cell AA15 and copy it onto AA15:AU421. Figure 21 shows the end result (rows 18 through 415 have been hidden).

AA15 | =waveeq1(AA$13,$Y15)

	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU
12	Analytical solution																						
13		*x*	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14	*t*																						
15	0		0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.0901	0.0992	0.0901	0.08	0.07	0.06	0.05	0.04	0.03	0.02	0.01	1E-16
16	0.01		0	0.0105	0.0209	0.0313	0.0416	0.0519	0.0621	0.0723	0.0824	0.0924	0.0986	0.0924	0.0824	0.0723	0.0621	0.0519	0.0416	0.0313	0.0209	0.0105	2E-16
17	0.02		0	0.011	0.0218	0.0325	0.0432	0.0538	0.0642	0.0745	0.0848	0.0951	0.097	0.0951	0.0848	0.0745	0.0642	0.0538	0.0432	0.0325	0.0218	0.011	2E-16
416	4.01		0	0.0105	0.0209	0.0313	0.0416	0.0519	0.0621	0.0723	0.0824	0.0924	0.0986	0.0924	0.0824	0.0723	0.0621	0.0519	0.0416	0.0313	0.0209	0.0105	2E-16
417	4.02		0	0.011	0.0218	0.0325	0.0432	0.0538	0.0642	0.0745	0.0848	0.0951	0.097	0.0951	0.0848	0.0745	0.0642	0.0538	0.0432	0.0325	0.0218	0.011	2E-16
418	4.03		0	0.0114	0.0227	0.0338	0.0448	0.0556	0.0663	0.0768	0.0871	0.0955	0.0955	0.0955	0.0871	0.0768	0.0663	0.0556	0.0448	0.0338	0.0227	0.0114	2E-16
419	4.04		0	0.0118	0.0235	0.035	0.0463	0.0574	0.0683	0.079	0.0896	0.0938	0.0939	0.0938	0.0896	0.079	0.0683	0.0574	0.0463	0.035	0.0235	0.0118	2E-16
420	4.05		0	0.0122	0.0243	0.0362	0.0478	0.0592	0.0703	0.0813	0.0914	0.0922	0.0923	0.0922	0.0914	0.0813	0.0703	0.0592	0.0478	0.0362	0.0243	0.0122	2E-16
421	4.06		0	0.0126	0.0251	0.0373	0.0493	0.061	0.0723	0.0833	0.0902	0.0905	0.0907	0.0905	0.0902	0.0833	0.0723	0.061	0.0493	0.0373	0.0251	0.0126	2E-16

Figure 21: Analytical solution of Example 3.

Notice that neither of the boundary conditions (zero) nor the initial conditions has been hard coded on the spreadsheet; these conditions are already accounted for by the series solution in Equation (17a). It can be seen from Figure 21 that the entries in column AU are in the order of 10^{-16}, which may be regarded as zero for the boundary condition at $x = 1$. The rest of the values in the table agree very well with those found in the numerical solution shown in Figure 18.

The analytical solution also provides Fourier series approximations to the initial conditions, as illustrated in Figure 22.

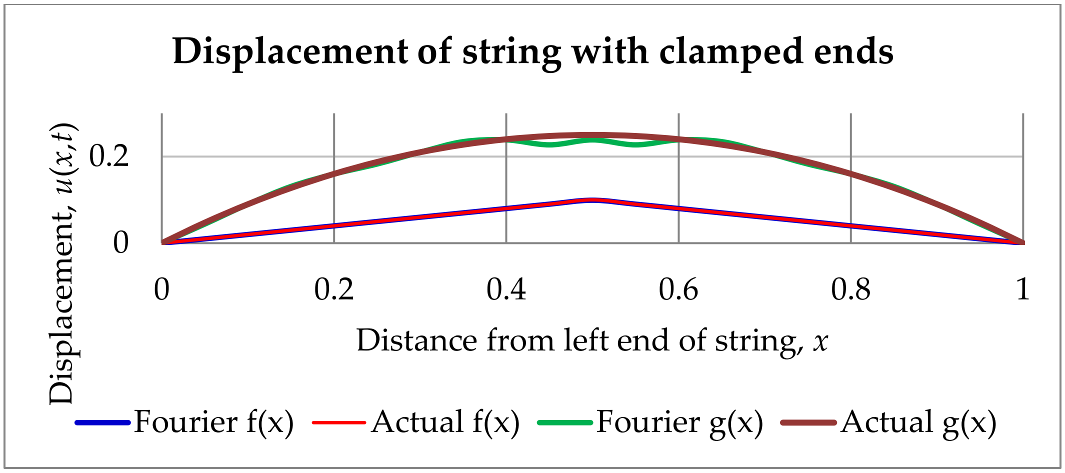


Figure 22: Fourier series approximations to the initial conditions $f(x)$ and $g(x)$ of Example 3.

The initial displacement $f(x)$ is very well approximated by its Fourier series. The initial velocity $g(x)$, however, shows small discrepancies between the actual function and its Fourier series, especially about the midpoint $x = 0.5$ where the derivative changes sign (the values for the green curve were obtained by approximating the derivative by a forward difference). A better approximation to $g(x)$ would have resulted if the user implemented the Fourier series directly, namely,

$$g(x) = \sum_{n=1}^{\infty} B_n^* \lambda_n \sin\left(\frac{\pi n x}{L}\right) \qquad \left(\lambda_n = \frac{\pi c n}{L}\right) \tag{18}$$

which was not pursued here as this would have required additional programming of more formulas in the spreadsheet or another VBA macro. The author opted for exploiting the available data in Figure 21 to find a quick, but reasonable Fourier series approximation to $g(x)$.

Case 2. String with freely sliding ends

Consider a rigid string with its two ends allowed to freely slide in the vertical direction such that the slope of the displacement curve is constrained to be zero. This situation is modeled by

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < L, \ \ t > 0 \tag{19a}$$

subject to the boundary conditions

$$\frac{\partial u}{\partial x}(0, t) = 0 \quad \text{and} \quad \frac{\partial u}{\partial x}(L, t) = 0, \qquad \forall t > 0 \tag{19b}$$

and the initial conditions

$$u(x, 0) = f(x) \quad \text{and} \quad \frac{\partial u}{\partial t}(x, 0) = g(x), \qquad 0 < x < L. \tag{19c}$$

Numerical solution of (19a)–(19c)

The recursive equation (13) remains valid at interior space-time nodal points; the initial displacement $f(x)$ can be coded directly on a spreadsheet, while the initial velocity $g(x)$ can be incorporated via Equation (16). As for the boundary conditions only the spatial derivatives at the endpoints are known (but not the values). To deduce formulas for the derivatives at the boundaries, substitute $x_0 = 0$ and $x_N = L$ in Equation (13) to obtain approximate values of the string displacement at the endpoints, i.e.,

$$u(0, t_{k+1}) = \eta u(x_1, t_k) + 2(1 - \eta)u(0, t_k) + \eta u(x_{-1}, t_k) - u(0, t_{k-1}) \tag{20a}$$

and

$$u(L, t_{k+1}) = \eta u(x_{N+1}, t_k) + 2(1 - \eta)u(L, t_k) + \eta u(x_{N-1}, t_k) - u(L, t_{k-1}). \tag{20b}$$

Notice that x_{-1} in (20a) and x_{N+1} in (20b) are outside the space interval. To circumvent this difficulty consider central difference approximations to the spatial derivatives at the boundaries, namely,

$$0 = \frac{\partial u}{\partial x}(0, t) \approx \frac{u(x_1, t) - u(x_{-1}, t)}{2\Delta x} \quad \rightarrow \quad u(x_{-1}, t) = u(x_1, t) \tag{21a}$$

and

$$0 = \frac{\partial u}{\partial x}(L, t) \approx \frac{u(x_{N+1}, t) - u(x_{N-1}, t)}{2\Delta x} \quad \rightarrow \quad u(x_{N+1}, t) = u(x_{N-1}, t) \tag{21b}$$

which hold for any value of time t and, in particular, $t = t_k$. Substituting (21a) and (21b) into (20a) and (20b), respectively, will result in

$$u(0, t_{k+1}) = 2\eta u(x_1, t_k) + 2(1 - \eta)u(0, t_k) - u(0, t_{k-1}) \tag{22a}$$

and

$$u(L, t_{k+1}) = 2\eta u(x_{N-1}, t_k) + 2(1-\eta)u(L, t_k) - u(L, t_{k-1}). \qquad (22b)$$

Equations (22a) and (22b) give approximate values to the string displacement at the endpoints by enforcing the zero slope condition at the boundaries as required by (19b). Only one more detail needs to be addressed and it is how to reconcile the initial velocity constraint in Equation (16) and the zero slope constraint at the boundaries. More specifically, if one lets $x_n = x_0 = 0$ or $x_n = x_N = L$ in (16) then the sample points x_{-1} and x_{N+1} will appear, but taking into account (21a) and (21b) the equations for the initial velocity at the boundaries will take the form

$$u(0, t_1) = \eta u(x_1, 0) + (1-\eta)u(0,0) + \Delta t g(0), \qquad (23a)$$

$$u(L, t_1) = \eta u(x_{N-1}, 0) + (1-\eta)u(L, 0) + \Delta t g(L). \qquad (23b)$$

Notice that Equations (16), (23a), and (23b) give the values of the displacement at $t = t_1$ (i.e., the second row of the solution table). Equations (13), (16), (22a), (22b), (23a), and (23b) form a complete set from which a spreadsheet model can now be constructed as shown by the following example.

EXAMPLE 4. Solve the wave equation

$$\frac{\partial^2 u}{\partial t^2} = 4\frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 1, \ t > 0$$

subject to the boundary conditions

$$\frac{\partial u}{\partial x}(0, t) = 0 \quad \text{and} \quad \frac{\partial u}{\partial x}(1, t) = 0, \qquad \forall t > 0$$

and the initial conditions

$$u(x, 0) = f(x) = 0.2\cos(2\pi x) \ \text{ and } \ \frac{\partial u}{\partial t}(x, 0) = g(x) = 0.5, \qquad 0 < x < 1.$$

Follow these steps to find the numerical solution to this example.

1) Lay out a table on the cell range A13:W180 with the samples of the space interval in C13:W13 and the time interval in A15:A180 (see Figure 23). Enter the initial displacement $f(x)$ by typing the formula =0.2*COS(2*PI()*C13) in cell C15 and then copying it onto

the cell range D15:W15. To incorporate the initial velocity $g(x)$ at interior points of the space interval, code Equation (16) by typing the formula =($L$10/2)*E15+(1-$L$10)*D15+($L$10/2)*C15+$E$10*0.5 in cell D16 and then copying it onto the cell range E16:V16. To account for the initial velocity $g(x)$ at the boundaries ($x = 0$ and $x = 1$), code Equation (23a) by typing the formula =$L$10*D15+(1-$L$10)*C15+$E$10*0.5 in cell C16, and (23b) by typing the formula =$L$10*V15+(1-$L$10)*W15+$E$10*0.5 in cell W16. The second row for $t_1 = \Delta t = 0.01$ is now completed and the initial setup should look like the one shown in Figure 23 (only the first 20 rows are displayed).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	One-dimensional wave equation																						
2			$\frac{\partial^2 u}{\partial t^2} = 4\frac{\partial^2 u}{\partial x^2}$,			$0 < x < 1$,	$t > 0$																
3																							
4																							
5	B.C.		$\frac{\partial u}{\partial x}(0,t) = 0$		and		$\frac{\partial u}{\partial x}(1,t) = 0$																
6																							
7	I.C.		$u(x,0) = 0.2\cos(2\pi x)$			and		$\frac{\partial u}{\partial t}(x,0) = 0.5$															
8																							
9																							
10	$\Delta x =$	0.05		$\Delta t =$	0.01		$c^2 =$	4		$\eta = \left(\frac{c\Delta t}{\Delta x}\right)^2 =$		0.16											
11																							
12	Numerical solution																						
13		x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14	t																						
15	0		0.2	0.1902	0.1618	0.1176	0.0618	1E-17	-0.062	-0.118	-0.162	-0.19	-0.2	-0.19	-0.162	-0.118	-0.062	1E-16	0.0618	0.1176	0.1618	0.1902	0.2
16	0.01		0.2034	0.1937	0.1655	0.1216	0.0663	0.005	-0.056	-0.112	-0.156	-0.184	-0.193	-0.184	-0.156	-0.112	-0.056	0.005	0.0663	0.1216	0.1655	0.1937	0.203
17	0.02																						
18	0.03																						
19	0.04																						
20	0.05																						

Figure 23: Initial setup for the numerical solution of Example 4.

2) To fill in the rest of the table, code Equation (13) by typing the formula =$L$10*E16+2*(1-$L$10)*D16+$L$10*C16-D15 in cell D17, and then copying it onto the cell range E17:V17. To implement the zero slope constraint at the boundaries, code Equation (22a) by entering the formula =2*$L$10*D16+2*(1-$L$10)*C16-C15 in cell C17, and (22b) by entering =2*$L$10*V16+2*(1-$L$10)*W16-W15 in cell W17. Finally, select the cell range C17:W17 and copy the contents to C18:W180. The end result is shown in Figure 24 (rows 18 through 174 have been hidden).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	One-dimensional wave equation																						
2																							
3			$\frac{\partial^2 u}{\partial t^2} = 4\frac{\partial^2 u}{\partial x^2},$			$0 < x < 1,$		$t > 0$															
4																							
5	B.C.		$\frac{\partial u}{\partial x}(0,t) = 0$		and		$\frac{\partial u}{\partial x}(1,t) = 0$																
6																							
7	I.C.		$u(x,0) = 0.2\cos(2\pi x)$			and		$\frac{\partial u}{\partial t}(x,0) = 0.5$															
8																							
9																							
10	$\Delta x =$	0.05		$\Delta t =$	0.01		$c^2 =$	4		$\eta = \left(\frac{c\Delta t}{\Delta x}\right)^2 =$		0.16											
11																							
12	Numerical solution																						
13		x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14	t																						
15	0		0.2	0.1902	0.1618	0.1176	0.0618	1E-17	-0.062	-0.118	-0.162	-0.19	-0.2	-0.19	-0.162	-0.118	-0.062	1E-16	0.0618	0.1176	0.1618	0.1902	0.2
16	0.01		0.2034	0.1937	0.1655	0.1216	0.0663	0.005	-0.056	-0.112	-0.156	-0.184	-0.193	-0.184	-0.156	-0.112	-0.056	0.005	0.0663	0.1216	0.1655	0.1937	0.2034
17	0.02		0.2038	0.1943	0.1668	0.1239	0.0699	0.01	-0.05	-0.104	-0.147	-0.174	-0.184	-0.174	-0.147	-0.104	-0.05	0.01	0.0699	0.1239	0.1668	0.1943	0.2038
175	1.6		0.8749	0.8712	0.8606	0.844	0.8231	0.8	0.7769	0.756	0.7394	0.7288	0.7251	0.7288	0.7394	0.756	0.7769	0.8	0.8231	0.844	0.8606	0.8712	0.8749
176	1.61		0.8561	0.8536	0.8463	0.835	0.8208	0.805	0.7892	0.775	0.7637	0.7564	0.7539	0.7564	0.7637	0.775	0.7892	0.805	0.8208	0.835	0.8463	0.8536	0.8561
177	1.62		0.8366	0.8353	0.8315	0.8256	0.8182	0.81	0.8018	0.7944	0.7885	0.7847	0.7834	0.7847	0.7885	0.7944	0.8018	0.81	0.8182	0.8256	0.8315	0.8353	0.8366
178	1.63		0.8166	0.8165	0.8163	0.8159	0.8155	0.815	0.8145	0.8141	0.8137	0.8135	0.8134	0.8135	0.8137	0.8141	0.8145	0.815	0.8155	0.8159	0.8163	0.8165	0.8166
179	1.64		0.7966	0.7977	0.8011	0.8062	0.8128	0.82	0.8272	0.8338	0.8389	0.8423	0.8434	0.8423	0.8389	0.8338	0.8272	0.82	0.8128	0.8062	0.8011	0.7977	0.7966
180	1.65		0.777	0.7793	0.7861	0.7968	0.8102	0.825	0.8398	0.8532	0.8639	0.8707	0.873	0.8707	0.8639	0.8532	0.8398	0.825	0.8102	0.7968	0.7861	0.7793	0.777

Figure 24: Numerical solution of Example 4.

The transverse displacement of the vibrating string at three different times is shown in Figure 25. The displacement curves show that the string drifts upwards because of the initial upward push ($g(x) = 0.5$ for $0 < x < 1$); also observe that because of the inflexibility of the string at the endpoints (zero slope constraint at the boundaries), the string tends to adopt a flat profile as time progresses.

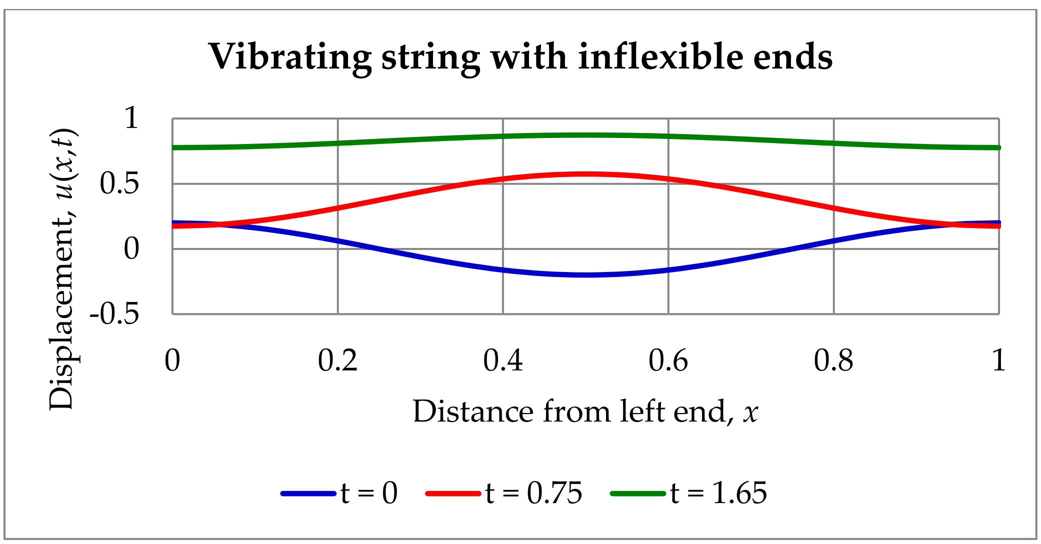


Figure 25: Transverse displacement for the vibrating string of Example 4.

Analytical solution of (19a)–(19c)

The analytical solution to the wave equation problem (19a)–(19c) can be obtained by the method of *separation of variables* (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011). The solution can be shown to be

$$u(x,t) = A_0 + A_0^* t + \sum_{n=1}^{\infty} [A_n \cos(\lambda_n t) + A_n^* \sin(\lambda_n t)] \cos\left(\frac{\pi n x}{L}\right) \qquad \left(\lambda_n = \frac{\pi c n}{L}\right), \tag{24a}$$

$$A_0 = \frac{1}{L}\int_0^L f(x)dx, \tag{24b}$$

$$A_0^* = \frac{1}{L}\int_0^L g(x)dx, \tag{24c}$$

$$A_n = \frac{2}{L}\int_0^L f(x)\cos\left(\frac{\pi n x}{L}\right)dx\,, \qquad n = 1, 2, \ldots \tag{24d}$$

$$A_n^* = \frac{2}{\pi c n}\int_0^L g(x)\cos\left(\frac{\pi n x}{L}\right)dx\,, \qquad n = 1, 2, \ldots \tag{24e}$$

In this example, $c = 2, L = 1, f(x) = 0.2\cos(2\pi x)$, $g(x) = 0.5$. Thus, evaluating the Fourier coefficients in Equations (24b)–(24e) yields $A_0 = 0, A_0^* = 0.5$, $A_2 = 0.2$, $A_n = 0\ (n \neq 2)$, $A_n^* = 0\ (n = 1, 2, \ldots)$ and, from (24a), the transverse displacement is given by

$$u(x,t) = 0.5t + 0.2\cos(2\pi x)\cos(4\pi t), \tag{25}$$

which is simple enough to code directly as a cell formula. Hence, if the solution is to occupy the cell range Y13:AU180 (see Figure 26), then one would type the formula =0.5*$Y15+0.2*COS(2*PI()*AA$13)*COS(4*PI()*$Y15) in cell AA15 and then copy it onto AA15:AU180, producing the final table shown in Figure 26 (rows 18 through 174 have been hidden).

AA15 | fx | =0.5*$Y15+0.2*COS(2*PI()*AA$13)*COS(4*PI()*$Y15)

	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU
12	Analytical solution																						
13		*x*	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14	*t*																						
15	0		0.2	0.1902	0.1618	0.1176	0.0618	1E-17	-0.062	-0.118	-0.162	-0.19	-0.2	-0.19	-0.162	-0.118	-0.062	1E-16	0.0618	0.1176	0.1618	0.1902	0.2
16	0.01		0.2034	0.1937	0.1655	0.1216	0.0663	0.005	-0.056	-0.112	-0.156	-0.184	-0.193	-0.184	-0.156	-0.112	-0.056	0.005	0.0663	0.1216	0.1655	0.1937	0.2034
17	0.02		0.2037	0.1942	0.1667	0.1239	0.0699	0.01	-0.05	-0.104	-0.147	-0.174	-0.184	-0.174	-0.147	-0.104	-0.05	0.01	0.0699	0.1239	0.1667	0.1942	0.2037
175	1.6		0.8618	0.8588	0.85	0.8363	0.8191	0.8	0.7809	0.7637	0.75	0.7412	0.7382	0.7412	0.75	0.7637	0.7809	0.8	0.8191	0.8363	0.85	0.8588	0.8618
176	1.61		0.8425	0.8406	0.8353	0.827	0.8166	0.805	0.7934	0.783	0.7747	0.7694	0.7675	0.7694	0.7747	0.783	0.7934	0.805	0.8166	0.827	0.8353	0.8406	0.8425
177	1.62		0.8226	0.8219	0.8202	0.8174	0.8139	0.81	0.8061	0.8026	0.7998	0.7981	0.7974	0.7981	0.7998	0.8026	0.8061	0.81	0.8139	0.8174	0.8202	0.8219	0.8226
178	1.63		0.8024	0.8031	0.8048	0.8076	0.8111	0.815	0.8189	0.8224	0.8252	0.8269	0.8276	0.8269	0.8252	0.8224	0.8189	0.815	0.8111	0.8076	0.8048	0.8031	0.8024
179	1.64		0.7825	0.7844	0.7897	0.798	0.8084	0.82	0.8316	0.842	0.8503	0.8556	0.8575	0.8556	0.8503	0.842	0.8316	0.82	0.8084	0.798	0.7897	0.7844	0.7825
180	1.65		0.7632	0.7662	0.775	0.7887	0.8059	0.825	0.8441	0.8613	0.875	0.8838	0.8868	0.8838	0.875	0.8613	0.8441	0.825	0.8059	0.7887	0.775	0.7662	0.7632

Figure 26: Analytical solution of Example 4.

Because of the choice of initial displacement for this example there is no difference between $f(x)$ and its Fourier series approximation, as it can be seen by substituting $t = 0$ in Equation (25). However, the initial velocity $g(x) = 0.5$ and its Fourier series approximation will exhibit some differences as shown in Figure 27.

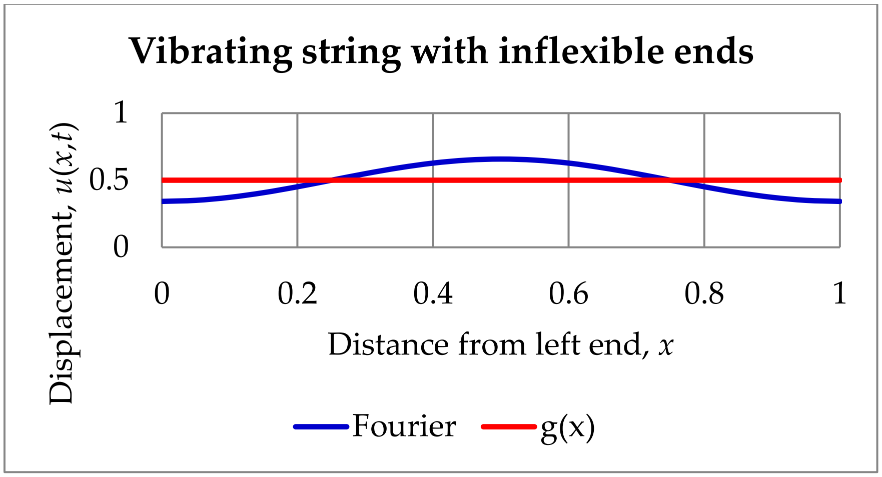


Figure 27: Fourier series approximation to the initial condition $g(x) = 0.5$ of Example 4.

4. Two-dimensional Laplace equation

This section presents the problem of determining the steady-state temperature in a thin, thermally conductive rectangular plate of length a and width b, with the edges of the plate having temperature distributions $f(x), g(x), r(y)$, and $s(y)$ impressed upon them, as shown in Figure 28.

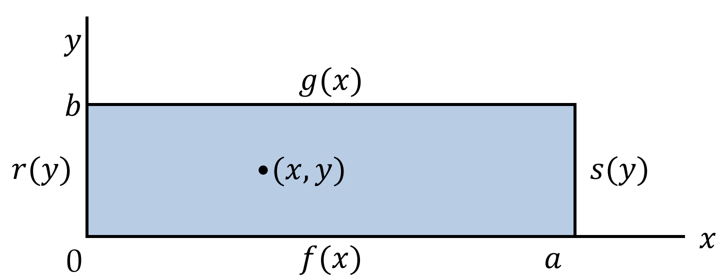


Figure 28: Thermally conductive rectangular plate showing dimensions and boundary conditions.

The steady-state temperature u at an interior point (x, y) of the rectangular plate is governed by the two-dimensional *Laplace equation* (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011)

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \qquad 0 < x < a, \qquad 0 < y < b \tag{26a}$$

subject to the boundary conditions

$$u(x,0) = f(x), \qquad 0 < x < a, \tag{26b}$$

$$u(x,b) = g(x), \qquad 0 < x < a, \tag{26c}$$

$$u(0,y) = r(y), \qquad 0 < y < b, \tag{26d}$$

$$u(a,y) = s(y), \qquad 0 < y < b. \tag{26e}$$

Numerical solution of (26a)–(26e)

To discretize the Laplace equation (26a), the x and y intervals are subdivided to create a rectangular mesh of points $x_i = i\Delta x$ $(i = 0,1,\dots,N)$ and $y_j = j\Delta y$ $(j = 0,1,\dots,M)$, as illustrated in Figure 29. Observe that $x_N = N\Delta x = a$ and $y_M = M\Delta y = b$. Nodes with coordinates of the form $(x_0, y_j), (x_N, y_j), (x_i, y_0)$, and (x_i, y_M) are boundary nodes, while the rest are interior nodes.

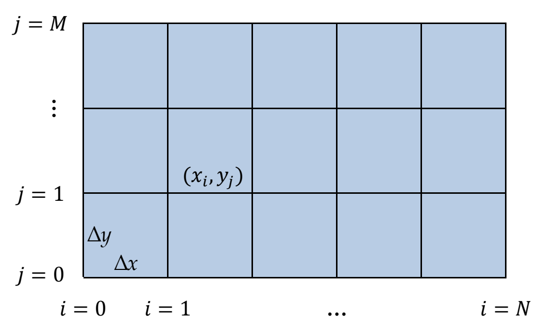


Figure 29: Rectangular mesh of points.

The derivatives at an interior nodal location (x_i, y_j) in Equation (26a) can be approximated by central differences to obtain

$$\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{\Delta x^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{\Delta y^2} = 0. \tag{27}$$

If the temperature at (x_i, y_j) is denoted by $u(x_i, y_j) = u_{i,j}$, and assuming equal increments $\Delta x = \Delta y = h$, then the preceding equation becomes

$$u_{i-1,j} + u_{i,j-1} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1} = 0. \tag{28a}$$

For the boundary nodes, the equations are

$$u_{i,0} = f(x_i), \qquad i = 1, 2, \dots, N-1, \tag{28b}$$

$$u_{i,M} = g(x_i), \qquad i = 1, 2, \dots, N-1, \tag{28c}$$

$$u_{0,j} = r(y_j), \qquad j = 1, 2, \dots, M-1, \tag{28d}$$

$$u_{N,j} = s(y_j), \qquad j = 1, 2, \dots, M-1. \tag{28e}$$

Equations (28a)–(28e) yield a system of $(N-1)(M-1)$ linear algebraic equations whose solution provides the approximate values of the temperatures at the interior nodes. The system of equations is sparse. If N and M are small, the system can be solved directly by Gaussian elimination or matrix inversion. If N and M are large, iterative techniques such as Jacobi, Gauss-Seidel, or relaxation methods are employed. In this paper, an iterative technique referred to as *successive over-relaxation* (SOR) method is illustrated.

Before introducing the SOR method, it is instructive to display the system of linear equations (28a) in the matrix form $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{M} & -\boldsymbol{I} & & \\ -\boldsymbol{I} & \boldsymbol{M} & -\boldsymbol{I} & \\ & -\boldsymbol{I} & \ddots & -\boldsymbol{I} \\ & & -\boldsymbol{I} & \boldsymbol{M} \end{bmatrix}_{[(N-1)(M-1)]\times[(N-1)(M-1)]}, \tag{29a}$$

$$\mathbf{M} = \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & -1 & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}_{(N-1)\times(N-1)}, \qquad \mathbf{I} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}_{(N-1)\times(N-1)}, \tag{29b}$$

$$\mathbf{x}^T = [u_{1,1} \;\; u_{2,1} \;\; \cdots \;\; u_{N-1,1} | \;\; u_{1,2} \;\; u_{2,2} \;\; \cdots \;\; u_{N-1,2} | \;\; \cdots \;\; | u_{1,M-1} \;\; u_{2,M-1} \;\; \cdots \;\; u_{N-1,M-1}], \tag{29c}$$

$$\mathbf{b}^T = [\tilde{\mathbf{b}}_1^T \quad \tilde{\mathbf{b}}_2^T \quad \cdots \quad \tilde{\mathbf{b}}_{M-2}^T \quad \tilde{\mathbf{b}}_{M-1}^T], \tag{29d}$$

$$\tilde{\mathbf{b}}_1^T = [u_{0,1} + u_{1,0} \quad u_{2,0} \quad u_{3,0} \quad \cdots \quad u_{N-2,0} \quad u_{N-1,0} + u_{N,1}], \tag{29e}$$

$$\tilde{\mathbf{b}}_j^T = [u_{0,j} \quad 0 \quad \cdots \quad 0 \quad u_{N,j}] \qquad (j = 2, 3, \dots, M-2), \tag{29f}$$

$$\tilde{\mathbf{b}}_{M-1}^T = [u_{0,M-1} + u_{1,M} \quad u_{2,M} \quad u_{3,M} \quad \cdots \quad u_{N-2,M} \quad u_{N,M-1} + u_{N-1,M}]. \tag{29g}$$

As can be seen from Equation (29a), the coefficient matrix $\mathbf{A}$ is banded. If N and M are not too large, $\mathbf{A}$ and $\mathbf{b}$ can be easily constructed according to (29a)–(29g) and the temperatures at the interior nodes of the mesh grid obtained as $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ using the matrix functions MMULT and MINVERSE of Excel. If N and M are too large then the SOR method can be employed. The SOR method represents an improvement over the Gauss-Seidel method and can be formulated as (Gutierrez, 2009)

$$u_{i,j}^{(k+1)} = u_{i,j}^{(k)} + \omega \frac{u_{i-1,j}^{(k+1)} + u_{i,j-1}^{(k+1)} - 4u_{i,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j+1}^{(k)}}{4}, \quad k = 0,1,\dots \tag{30}$$

which follows from Equation (28a). In Equation (30), the superscripts denote the iteration number, and values with zero superscript correspond to the initial guesses; the quantity ω is the over-relaxation parameter and is such that $1 < \omega < 2$ (when $\omega = 1$ the SOR reduces to Gauss-Seidel). Also observe that because the interior nodes are visited from left to right and from bottom to top, the most recent $k+1$st iterates for $u_{i-1,j}^{(k+1)}$ and $u_{i,j-1}^{(k+1)}$ would be available to compute the next iterate for $u_{i,j}^{(k)}$.

The following example shows how to implement the SOR method for solving a Laplace equation numerically.

EXAMPLE 5. Solve the Laplace equation

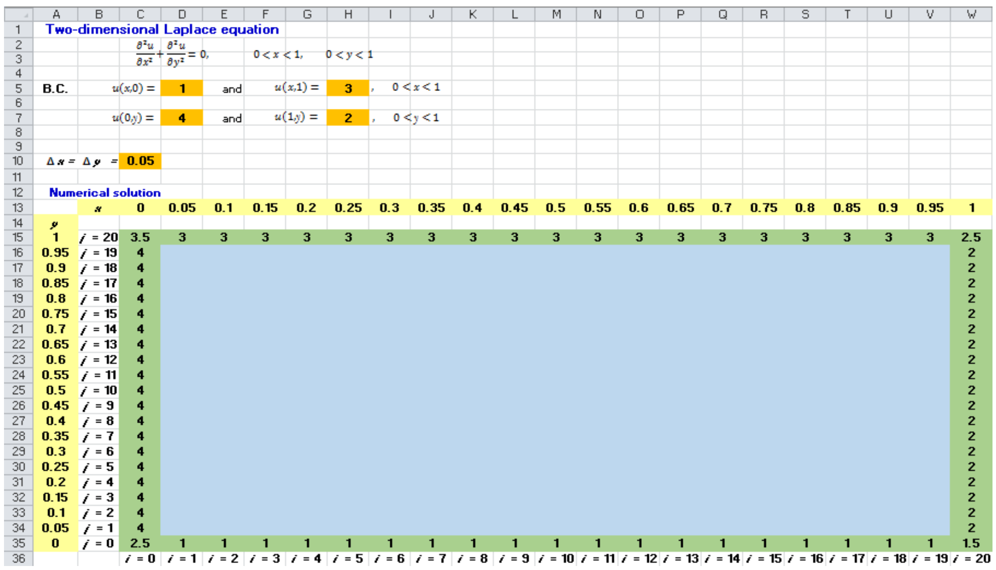
$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \qquad 0 < x < 1, \qquad 0 < y < 1$$

subject to the boundary conditions

$$\begin{aligned} u(x,0) &= 1, & 0 < x < 1, \\ u(x,1) &= 3, & 0 < x < 1, \\ u(0,y) &= 4, & 0 < y < 1, \\ u(1,y) &= 2, & 0 < y < 1. \end{aligned}$$

To solve the Laplace equation numerically, follow these steps:

1) Divide the x and y intervals in steps of size, say $\Delta x = \Delta y = 0.05$ (for a total of $N \times M = 19 \times 19 = 361$ unknown temperatures at the interior nodes in the mesh grid). Write the corresponding coordinates of the (x, y) points thus generated and their associated (i, j) indices for ease of reference. Also enter the boundary conditions, using cell formulas if necessary (in case the boundary conditions are given as functions of x or y). The initial setup may resemble Figure 30.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Two-dimensional Laplace equation																						
2																							
3			$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0,$			$0<x<1,$		$0<y<1$															
4																							
5	B.C.		$u(x,0) =$	1	and		$u(x,1) =$	3	,	$0<x<1$													
6																							
7			$u(0,y) =$	4	and		$u(1,y) =$	2	,	$0<y<1$													
8																							
9																							
10	$\Delta x = \Delta y$	=	0.05																				
11																							
12	Numerical solution																						
13		x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14	y																						
15	1	j = 20	3.5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.5
16	0.95	j = 19	4																				2
17	0.9	j = 18	4																				2
18	0.85	j = 17	4																				2
19	0.8	j = 16	4																				2
20	0.75	j = 15	4																				2
21	0.7	j = 14	4																				2
22	0.65	j = 13	4																				2
23	0.6	j = 12	4																				2
24	0.55	j = 11	4																				2
25	0.5	j = 10	4																				2
26	0.45	j = 9	4																				2
27	0.4	j = 8	4																				2
28	0.35	j = 7	4																				2
29	0.3	j = 6	4																				2
30	0.25	j = 5	4																				2
31	0.2	j = 4	4																				2
32	0.15	j = 3	4																				2
33	0.1	j = 2	4																				2
34	0.05	j = 1	4																				2
35	0	j = 0	2.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.5
36			i = 0	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8	i = 9	i = 10	i = 11	i = 12	i = 13	i = 14	i = 15	i = 16	i = 17	i = 18	i = 19	i = 20

Figure 30: Initial setup for the numerical solution of Example 5.

2) On another section of the spreadsheet, implement the SOR algorithm given in Equation (30) as follows:

   a) Create a column with labels that identify the temperatures at interior nodes. For example, write the temperatures $u_{1,1}$, $u_{2,1}$, …, $u_{19,1}$, … , $u_{1,19}$, $u_{2,19}$, … , $u_{19,19}$ (listing nodal temperatures from left to right and from bottom to top) so as to occupy the cell range A50:A410 as shown in Figure 31 (only the top section of the table is shown in the figure).

D50 =C50+$H$45*($C$34+$D$35-4*C50+C51+C69)/4

	A	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	U
45	Succesive over-relaxation (SOR) method with ω =						1.6												
46																			
47																			
48	Temper-	Iteration (k = 0 is the initial guess)																	
49	ature	k = 0	k = 1	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8	k = 9	k = 10	k = 11	k = 12	k = 13	k = 11	k = 12	k = 13	Diff
50	u_1,1	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	0
51	u_2,1	2.5	1.9	1.924	1.91152	1.9111	1.90924	1.90876	1.90823	1.90798	1.90776	1.90763	1.90752	1.90744	1.90738	1.90733	1.90729	1.90726	-3E-05
52	u_3,1	2.5	1.66	1.6648	1.64502	1.63975	1.63576	1.63372	1.6323	1.63138	1.63071	1.63024	1.62987	1.6296	1.62938	1.62921	1.62907	1.62896	-0.0001
53	u_4,1	2.5	1.564	1.53424	1.50986	1.49776	1.49126	1.487	1.48432	1.48241	1.48107	1.48006	1.4793	1.47871	1.47824	1.47787	1.47756	1.47731	-0.0003
54	u_5,1	2.5	1.5256	1.46512	1.43521	1.41697	1.40731	1.40072	1.39645	1.39337	1.39115	1.38947	1.38818	1.38717	1.38637	1.38572	1.38518	1.38473	-0.0005
55	u_6,1	2.5	1.51024	1.42826	1.39158	1.3686	1.35546	1.34663	1.34061	1.33627	1.33305	1.33061	1.3287	1.32719	1.32598	1.32499	1.32415	1.32339	-0.0008
56	u_7,1	2.5	1.5041	1.40884	1.36536	1.33872	1.32229	1.31134	1.30358	1.29796	1.29371	1.29044	1.28786	1.2858	1.28413	1.28271	1.28144	1.28038	-0.0011
57	u_8,1	2.5	1.50164	1.39881	1.34949	1.31987	1.30066	1.28774	1.27839	1.27151	1.26624	1.26214	1.25888	1.25623	1.25399	1.25201	1.25041	1.24914	-0.0013
58	u_9,1	2.5	1.50066	1.39374	1.33999	1.30785	1.28644	1.27174	1.26099	1.25293	1.2467	1.2418	1.23784	1.23446	1.23152	1.22927	1.22745	1.22574	-0.0017
59	u_10,1	2.5	1.50026	1.39122	1.33438	1.30016	1.27705	1.26082	1.24886	1.23975	1.23265	1.22694	1.22202	1.21788	1.21488	1.21233	1.20999	1.2081	-0.0019
60	u_11,1	2.5	1.5001	1.39	1.33115	1.29525	1.27083	1.25337	1.24037	1.23035	1.22234	1.21536	1.20984	1.20601	1.20254	1.19948	1.19707	1.19502	-0.002
61	u_12,1	2.5	1.50004	1.38941	1.32932	1.29215	1.26673	1.24829	1.23443	1.22332	1.21362	1.20668	1.20196	1.19733	1.19355	1.19056	1.18805	1.18601	-0.002
62	u_13,1	2.5	1.50002	1.38913	1.32831	1.29023	1.26402	1.24483	1.22945	1.21624	1.20822	1.20243	1.19648	1.19203	1.18844	1.1855	1.18312	1.18116	-0.002
63	u_14,1	2.5	1.50001	1.389	1.32776	1.28905	1.26226	1.24045	1.22303	1.21488	1.20759	1.20037	1.19536	1.19119	1.1879	1.18523	1.18306	1.1813	-0.0018
64	u_15,1	2.5	1.5	1.38894	1.32747	1.28834	1.25599	1.23478	1.22808	1.21846	1.21044	1.20493	1.20036	1.19684	1.19399	1.19171	1.18986	1.18836	-0.0015
65	u_16,1	2.5	1.5	1.38891	1.32732	1.27513	1.25474	1.25092	1.23812	1.23019	1.22427	1.21963	1.21606	1.21321	1.21094	1.20911	1.20764	1.20644	-0.0012
66	u_17,1	2.5	1.5	1.3889	1.29524	1.29281	1.29037	1.27489	1.26792	1.26191	1.25766	1.25429	1.25168	1.2496	1.24795	1.24662	1.24554	1.24466	-0.0009
67	u_18,1	2.5	1.5	1.30889	1.3752	1.36506	1.35084	1.34501	1.33986	1.33641	1.33365	1.33157	1.32991	1.3286	1.32756	1.32671	1.32603	1.32547	-0.0006
68	u_19,1	2.5	1.3	1.56489	1.53429	1.52776	1.52314	1.52023	1.51804	1.51643	1.51518	1.51422	1.51346	1.51285	1.51237	1.51198	1.51166	1.5114	-0.0003
69	u_1,2	2.5	3.1	3.076	3.08848	3.0889	3.09076	3.09124	3.09177	3.09202	3.09224	3.09237	3.09248	3.09256	3.09262	3.09267	3.09271	3.09274	3.2E-05
70	u_2,2	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	0

Figure 31: Portion of spreadsheet showing iterations in SOR method.

b) Create a table that contains the iterations resulting from the application of the SOR method. Start by entering the initial guesses $u_{i,j}^{(0)}$ in the cell range C50:C410. In this case, Figure 31 shows all initial guesses of 2.5 (the average of the four temperatures at the boundaries of the square plate), but they could have taken on any values and not necessarily equal to one another for the SOR to converge.

c) Implement the SOR method as given in Equation (30). For example, in cell D50 (first iterate of $u_{1,1}$) type the formula =C50+$H$45*($C$34+$D$35-4*C50+C51+C69)/4, in cell D51 type =C51+$H$45*(D50+$E$35-4*C51+C52+C70)/4 and so on. The user can copy and paste these formulas and edit them accordingly so as to make the appropriate cell references. Observe that the formulas make reference to the over-relation parameter $\omega = 1.6$ stored in cell H45.

d) Add as many columns as needed to continue the SOR iterative process until successive iterations are within a prescribed precision level. Optionally, the user can create an additional column, say U50:U410, that shows the difference between the last two iterations to ascertain convergence.

3) Copy the values from the last iteration of the SOR in the cell range `S50:S410` to fill the cell range `D16:V34` with the internal nodal temperatures. The final result will resemble Figure 32. The temperature at the corners of the plate were manually adjusted to be the average of their closest neighboring nodal temperatures; for instance, the temperature at $(x, y) = (0,0)$ in cell `C35` was adjusted with the formula `=(C34+D35)/2` and so on.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
12	Numerical solution																						
13		x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
14	y																						
15	1	j = 20	3.5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2.5
16	0.95	j = 19	4	3.4886	3.27994	3.17651	3.11505	3.07368	3.04299	3.01879	2.99854	2.98094	2.96501	2.9499	2.93478	2.91861	2.89991	2.87626	2.8432	2.79129	2.69782	2.5	2
17	0.9	j = 18	4	3.67453	3.45459	3.31012	3.2103	3.13651	3.0794	3.03331	2.99435	2.96008	2.92912	2.89976	2.87057	2.83972	2.80474	2.76194	2.70524	2.62413	2.5	2.30218	2
18	0.85	j = 17	4	3.75534	3.55406	3.39855	3.27839	3.1833	3.10499	3.0403	2.98517	2.93602	2.89145	2.8494	2.80798	2.76491	2.71739	2.6615	2.59168	2.5	2.37587	2.20871	2
19	0.8	j = 16	4	3.79356	3.60849	3.45159	3.32082	3.21162	3.11871	3.03791	2.96902	2.90763	2.85108	2.79843	2.74693	2.69454	2.63838	2.57497	2.5	2.40832	2.29476	2.1568	2
20	0.75	j = 15	4	3.81164	3.63595	3.47901	3.3417	3.22257	3.11855	3.0268	2.94456	2.87351	2.80742	2.74595	2.68684	2.62793	2.56656	2.5	2.42503	2.3385	2.23806	2.12374	2
21	0.7	j = 14	4	3.8187	3.64633	3.48785	3.34503	3.21797	3.10501	3.00431	2.91377	2.83155	2.75972	2.69089	2.62672	2.56359	2.5	2.43344	2.36162	2.28261	2.19526	2.10009	2
22	0.65	j = 13	4	3.81884	3.64498	3.48262	3.33385	3.19949	3.07877	2.97047	2.87301	2.78486	2.70455	2.6332	2.56469	2.5	2.43641	2.37207	2.30546	2.23509	2.16028	2.08139	2
23	0.6	j = 12	4	3.81399	3.63467	3.46595	3.31006	3.16823	3.04027	2.92525	2.82198	2.72907	2.64516	2.56913	2.5	2.43531	2.37328	2.31316	2.25307	2.19202	2.12943	2.06522	2
24	0.55	j = 11	4	3.80498	3.61664	3.43894	3.27443	3.12454	2.98948	2.86835	2.76018	2.66358	2.57724	2.5	2.43087	2.3668	2.30911	2.25405	2.20157	2.1506	2.10024	2.0501	2
25	0.5	j = 10	4	3.7919	3.59092	3.40154	3.22669	3.06793	2.92571	2.79898	2.68679	2.58759	2.5	2.42276	2.35484	2.29545	2.24028	2.19258	2.14892	2.10855	2.07088	2.03499	2
26	0.45	j = 9	4	3.77426	3.55666	3.35254	3.1654	2.9969	2.84753	2.71585	2.60063	2.5	2.41241	2.33642	2.27093	2.21514	2.16845	2.12649	2.09237	2.06398	2.03992	2.01906	2
27	0.4	j = 8	4	3.75086	3.51164	3.28926	3.08789	2.90904	2.75274	2.61705	2.5	2.39937	2.31321	2.23982	2.17802	2.12699	2.08623	2.05544	2.03098	2.01483	2.00565	2.00146	2
28	0.35	j = 7	4	3.71962	3.45248	3.20764	2.98994	2.80043	2.63809	2.5	2.38295	2.28415	2.20102	2.13165	2.07475	2.02953	1.99569	1.9732	1.96209	1.9597	1.96669	1.98121	2
29	0.3	j = 6	4	3.67661	3.37279	3.10084	2.8656	2.66639	2.5	2.36191	2.24726	2.15247	2.07429	2.01052	1.95973	1.92123	1.89499	1.88145	1.88129	1.89501	1.9206	1.95701	2
30	0.25	j = 5	4	3.61527	3.26261	2.95853	2.70591	2.5	2.33361	2.19957	2.09096	2.0031	1.93207	1.87546	1.83177	1.80051	1.78203	1.77743	1.78838	1.8167	1.86349	1.92632	2
31	0.2	j = 4	4	3.52269	3.10488	2.76576	2.5	2.29409	2.1344	2.01006	1.91211	1.8346	1.77331	1.72557	1.68994	1.66615	1.65497	1.6583	1.67918	1.72161	1.7897	1.88495	2
32	0.15	j = 3	4	3.37104	2.86894	2.5	2.23424	2.04147	1.89916	1.79236	1.71074	1.64746	1.59846	1.56106	1.53405	1.51738	1.51215	1.52099	1.54841	1.60145	1.68988	1.82349	2
33	0.1	j = 2	4	3.09274	2.5	2.13106	1.89512	1.73739	1.62721	1.54752	1.48836	1.44334	1.40908	1.38336	1.36533	1.35502	1.35367	1.36405	1.39151	1.44594	1.54541	1.72006	2
34	0.05	j = 1	4	2.5	1.90726	1.62896	1.47731	1.38473	1.32339	1.28038	1.24914	1.22574	1.2081	1.19502	1.18601	1.18116	1.1813	1.18836	1.20644	1.24466	1.32547	1.5114	2
35	0	j = 0	2.5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.5

Figure 32: Numerical solution of Example 5.

The temperature distribution in the square plate can be graphically displayed as a filled contour plot as shown in Figure 33.

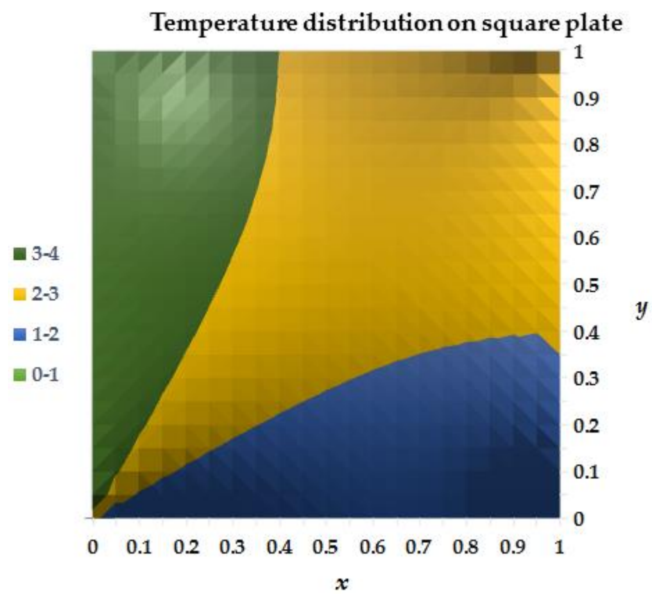


Figure 33: Temperature distribution in heat conducting plate of Example 5.

Analytical solution of (26a)–(26e)

The analytical solution to the Laplace equation (26a) subject to the boundary conditions (26b)–(26e) can be obtained by the method of separation of variables and superposition (Greenberg, 1998; Kreyszig, 2011; O'Neil, 2011) and can be shown to be

$$u(x,y) = \sum_{n=1}^{\infty} \left\{ \left[A_n \sinh\left(\frac{\pi n(b-y)}{a}\right) + B_n \sinh\left(\frac{\pi n y}{a}\right) \right] \sin\left(\frac{\pi n x}{a}\right) + \left[C_n \sinh\left(\frac{\pi n(a-x)}{b}\right) + D_n \sinh\left(\frac{\pi n x}{b}\right) \right] \sin\left(\frac{\pi n y}{b}\right) \right\}, \tag{31a}$$

$$A_n = \frac{2}{a \sinh(\pi b n/a)} \int_0^a f(x) \sin\left(\frac{\pi n x}{a}\right) dx, \qquad n = 1, 2, \ldots \tag{31b}$$

$$B_n = \frac{2}{a \sinh(\pi b n/a)} \int_0^a g(x) \sin\left(\frac{\pi n x}{a}\right) dx, \qquad n = 1, 2, \ldots \tag{31c}$$

$$C_n = \frac{2}{b \sinh(\pi a n/b)} \int_0^b r(y) \sin\left(\frac{\pi n y}{b}\right) dy, \qquad n = 1, 2, \ldots \tag{31d}$$

$$D_n = \frac{2}{b\sinh(\pi a n/b)} \int_0^b s(y) \sin\left(\frac{\pi n y}{b}\right) dy, \qquad n = 1, 2, \ldots \tag{31e}$$

In this example, $a = 1, b = 1, f(x) = 1, g(x) = 3, r(y) = 4$, and $s(y) = 2$. Thus, it can readily be found that the Fourier coefficients are $A_n = \frac{2(1-\cos(\pi n))}{\pi n \sinh(\pi n)}, B_n = \frac{6(1-\cos(\pi n))}{\pi n \sinh(\pi n)}, C_n = \frac{8(1-\cos(\pi n))}{\pi n \sinh(\pi n)}$, and $D_n = \frac{4(1-\cos(\pi n))}{\pi n \sinh(\pi n)}$. With these coefficients in hand, one can code Equation (31a) to evaluate the temperature at an interior point (x, y). The user-defined function lapleq in Figure 34 serves this purpose.

```
Public Function lapleq(x, y)
    S = 0
    For n = 1 To Nmax
        An = 2 * (1 - Cos(PI * n)) / (PI * n * WorksheetFunction.Sinh(PI * n))
        Bn = 3 * An
        Cn = 4 * An
        Dn = 2 * An
        T1 = An * Sin(PI * n * x) * WorksheetFunction.Sinh(PI * n * (1 - y))
        T2 = Bn * Sin(PI * n * x) * WorksheetFunction.Sinh(PI * n * y)
        T3 = Cn * WorksheetFunction.Sinh(PI * n * (1 - x)) * Sin(PI * n * y)
        T4 = Dn * WorksheetFunction.Sinh(PI * n * x) * Sin(PI * n * y)
        S = S + T1 + T2 + T3 + T4
    Next n
    lapleq = S
End Function
```

Figure 34: VBA user-defined function lapleq for solving Laplace's equation.

With the function thus defined, the analytical solution may be implemented in another section of the spreadsheet, say AA15:AU35. One can then enter the formula =lapleq(AB$13,$Y15) in cell AB15 and copy it onto the cell range AA15:AU35. As with the numerical solution, the temperature at the corners of the plate are manually adjusted to match the average of their closest neighboring nodal temperatures; for instance, the temperature at $(x, y) = (0,0)$ in cell AA35 was adjusted with the formula =(AA34+AB35)/2 and so on. The final result is shown in Figure 35.

AB15 =lapleq(AB$13,$Y15)

Analytical solution

y \ x	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
1	3.88215	3.32755	3.04677	2.8516	3.10422	2.99607	2.92824	3.07873	2.97526	2.95672	3.07335	2.95672	2.97526	3.07873	2.92824	2.99607	3.10422	2.8516	3.04677	3.32755	2.77296
0.95	4.43674	3.50058	3.27275	3.1705	3.11511	3.07268	3.04181	3.02017	2.99825	2.98049	2.9665	2.94955	2.93474	2.92053	2.89974	2.87741	2.84759	2.79389	2.70433	2.50823	2.21837
0.9	4.06237	3.68194	3.45619	3.30923	3.20943	3.13651	3.07994	3.03392	2.9949	2.96073	2.92975	2.90041	2.87132	2.84069	2.806	2.76368	2.70757	2.62649	2.49997	2.29458	2.03118
0.85	3.80213	3.75928	3.55943	3.4016	3.28007	3.18446	3.1069	3.04207	2.98636	2.93719	2.89247	2.85035	2.80891	2.76592	2.71852	2.66272	2.5927	2.5	2.37347	2.20286	1.90107
0.8	4.13896	3.80155	3.61549	3.45759	3.32536	3.2148	3.12139	3.04127	2.97138	2.9092	2.8526	2.79954	2.74793	2.69541	2.63913	2.5755	2.5	2.4073	2.29248	2.1555	2.06948
0.75	3.99476	3.81672	3.64413	3.48738	3.34897	3.22811	3.12271	3.03035	2.94872	2.87567	2.80917	2.74724	2.68779	2.62855	2.56693	2.5	2.4245	2.33728	2.23631	2.12206	1.99738
0.7	3.90432	3.82324	3.6558	3.49834	3.35483	3.22594	3.11101	3.00874	2.91751	2.83558	2.76117	2.69242	2.62738	2.56399	2.5	2.43307	2.36087	2.28148	2.19398	2.09848	1.95216
0.65	4.10497	3.82717	3.65574	3.49498	3.34587	3.20969	3.08662	2.97608	2.87699	2.78797	2.70749	2.63386	2.56531	2.5	2.43601	2.37145	2.30459	2.23408	2.15935	2.08168	2.05248
0.6	3.96702	3.82045	3.64641	3.47975	3.32379	3.1802	3.04962	2.93192	2.82635	2.73182	2.64694	2.5702	2.5	2.43469	2.37262	2.31221	2.25207	2.19109	2.12867	2.06448	1.98351
0.55	3.94229	3.81144	3.62897	3.45362	3.28918	3.13756	2.99966	2.87555	2.76465	2.66597	2.57823	2.5	2.4298	2.36614	2.30758	2.25276	2.20046	2.14965	2.09957	2.0493	1.97114
0.5	4.0978	3.80093	3.60341	3.41639	3.24158	3.08106	2.9358	2.80589	2.6907	2.58917	2.5	2.42177	2.35306	2.29251	2.23883	2.19083	2.1474	2.10753	2.07028	2.03552	2.0489
0.45	3.94229	3.7805	3.56865	3.36678	3.17952	3.00913	2.8565	2.72143	2.60304	2.5	2.41083	2.33403	2.26818	2.21203	2.16442	2.12433	2.0908	2.06281	2.03925	2.01836	1.97114
0.4	3.96702	3.75694	3.52283	3.30229	3.10035	2.91927	2.7595	2.62024	2.5	2.39696	2.3093	2.23535	2.17365	2.12301	2.08249	2.05128	2.02862	2.01364	2.00509	2.00097	1.98351
0.35	4.10497	3.72753	3.46251	3.21883	3.00001	2.80788	2.64187	2.5	2.37976	2.27857	2.19411	2.12445	2.06808	2.02392	1.99126	1.96965	1.95873	1.95793	1.96612	1.98204	2.05248
0.3	3.90432	3.68117	3.38185	3.10996	2.87258	2.67016	2.5	2.35813	2.2405	2.1435	2.0642	2.00034	1.95038	1.91338	1.88899	1.87729	1.87861	1.8931	1.92003	1.95641	1.95216
0.25	3.99476	3.62145	3.2713	2.96563	2.70968	2.5	2.32984	2.19212	2.08073	1.99087	1.91894	1.86244	1.8198	1.79031	1.77406	1.77189	1.7852	1.81554	1.86348	1.92679	1.99738
0.2	4.13896	3.53403	3.11364	2.77022	2.5	2.29032	2.12742	1.99999	1.89965	1.82048	1.75842	1.71082	1.67621	1.65413	1.64517	1.65103	1.67464	1.71993	1.79062	1.88798	2.06948
0.15	3.80213	3.38268	2.87669	2.5	2.22978	2.03437	1.89004	1.78117	1.69771	1.63322	1.58361	1.54638	1.52025	1.50502	1.50166	1.51262	1.54241	1.5984	1.69073	1.82626	1.90107
0.1	4.06237	3.11351	2.49997	2.12328	1.88641	1.72869	1.61812	1.53753	1.47716	1.43134	1.39663	1.37101	1.35357	1.34429	1.34418	1.35586	1.38456	1.44053	1.54374	1.72615	2.03118
0.05	4.43674	2.50823	1.88539	1.61407	1.46906	1.37803	1.31705	1.27468	1.24228	1.21835	1.20109	1.18741	1.17877	1.17504	1.17498	1.18275	1.20154	1.23747	1.31696	1.51588	2.21837
0	2.77296	1.10918	1.01559	0.95053	1.03474	0.99869	0.97608	1.02624	0.99175	0.98557	1.02445	0.98557	0.99175	1.02624	0.97608	0.99869	1.03474	0.95053	1.01559	1.10918	1.66378

Figure 35: Analytical solution of Example 5.

Notice how the analytical solution exhibits the Gibbs phenomenon at the boundaries, which are manifested by the fluctuating values about the constant temperatures at the edges of the plate. The temperatures at the interior nodes displayed in Figure 35 compare fairly well with those obtained numerically in Figure 32; the maximum discrepancy between both sets of values is about 0.44 occurring at the left edge of the square plate due to the Gibbs phenomenon.

5. Conclusions

This paper presented some of the classical partial differential equations (viz., the heat equation, wave equation, and Laplace's equation) with illustrative examples that make use of Excel spreadsheets for the implementation of the numerical and analytical solutions to these equations. The basis for the numerical solutions is the discretization of the equations which result in recursive algorithms that can be coded with spreadsheet cell formulas. The equations for which analytical solutions are known were implemented using VBA. Of particular interest is the successive over-relaxation (SOR) method used in the numerical solution to the Laplace equation. The graphing capabilities of Excel were exploited to enhance the visualization of the solutions to these equations by displaying their behavior as functions and Fourier series approximations to initial and/or boundary conditions.

References

Arganbright, D. (1985). Heat flow on a plate. In *Mathematical applications of electronic spreadsheets*. McGraw-Hill. pp. 152–155.

Greenberg, M. D. (1998). Advanced engineering mathematics. 2nd ed. Prentice Hall.

Gutierrez, E. (2009). Steady state heat conduction in 2D and 3D: Numerical methods.
Online: http://www.ewp.rpi.edu/hartford/~ernesto/F2009/CHT/Notes/ch04.html
Accessed: September 18, 2016.

Kreyszig, E. (2011). Advanced engineering mathematics. 10th ed. John Wiley & Sons.

Neuwirth, E., & Arganbright, D. (2004). Heat propagation. In *The active modeler: Mathematical modeling with Microsoft Excel*. Cengage Learning. pp. 141–154.

O'Neil, P. V. (2011). Advanced engineering mathematics. 7th ed. Cengage Learning.

Orvis, W. J. (1997). Applying spreadsheets in physics and electronic engineering. In *Spreadsheets in science and engineering*, Filby, G. (ed.) Springer. pp. 37-86.

Piccolo, C. (n.d. a). Using Excel spreadsheets to solve a 1D heat equation.
Online: http://www.math.ubc.ca/~costanza/HeatEqTutorial.html
Accessed: September 18, 2016.

— (n.d. b). Using Excel spreadsheets to solve a 1D wave equation.
Online: http://www.math.ubc.ca/~costanza/WaveTutorial.html
Accessed: September 18, 2016.