

The Fourier series uses the trigonometric functions as an orthogonal basis set. The trigonometric form of the Fourier series is given by the equation:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cdot \text{Cos}(\omega_0 kt) + \sum_{k=1}^{\infty} b_k \cdot \text{Sin}(\omega_0 kt) \quad 2.4$$

where

$$a_k = \frac{2}{T} \cdot \int_T f(t) \cdot \text{Cos}(\omega_0 kt) dt \quad \text{and} \quad b_k = \frac{2}{T} \cdot \int_T f(t) \cdot \text{Sin}(\omega_0 kt) dt$$

Example 2.1 Find the Fourier series for the square wave shown below.

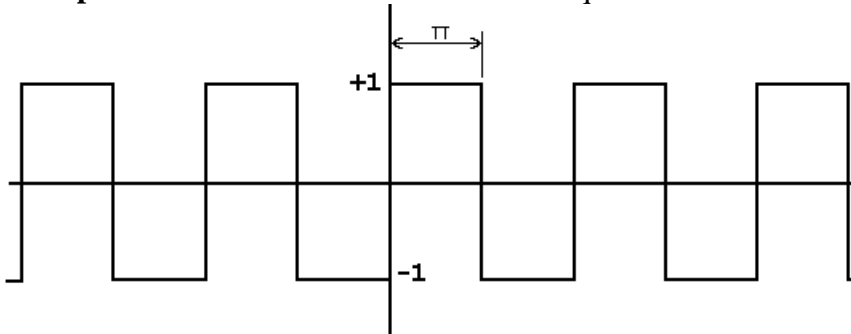


Figure E2.1-1

A square wave with a period of 2π .

Solution:

To find the coefficients a_k and b_k , we need to integrate over one period. For this problem we will take the period from 0 to 2π . The equation for $f(t)$ is:

$$f(t) = \begin{cases} +1 & 0 \leq t \leq \pi \\ -1 & \pi \leq t \leq 2\pi \end{cases}$$

The equations for a_k and b_k can be evaluated as:

$$a_k = \frac{1}{\pi} \int_0^{\pi} (+1) \text{Cos}(kt) dt + \frac{1}{\pi} \int_{\pi}^{2\pi} (-1) \text{Cos}(kt) dt = 0$$

$$b_k = \frac{1}{\pi} \int_0^{\pi} (+1) \text{Sin}(kt) dt + \frac{1}{\pi} \int_{\pi}^{2\pi} (-1) \text{Sin}(kt) dt = \begin{cases} 0 & k \text{ even} \\ \frac{4}{k\pi} & k \text{ odd} \end{cases}$$

We can then write $f(t)$ as a Fourier series.

$$f(t) = \sum_{\substack{k=1 \\ k \text{ odd}}}^{\infty} \frac{4}{k\pi} \text{Sin}(kt)$$

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
/*This program calculates and displays a square wave or a saw tooth wave constructed
 * by summing sinusoids using the Fourier series.
 */
namespace FourierSeriesSquareWave
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private const float xWLDMIN = (float)(-2*Math.PI); //Min and max world x values
        private const float xWLDMAX = (float)(2*Math.PI);
        private const float yWLDMIN = -1.5f; //Min and max world y values
        private const float yWLDMAX = 1.5f;
        private const int xPOINTS = 200; //Number of points in x
        //fourierData holds the data for the plot in screen coordinates
        private float [,] fourierData = new float[xPOINTS, 2];
        private int terms; //number of terms to use
        private bool dataValid = false; //true when the data is valid
        /// <summary>
        /// btnPlot
        /// Calculates the data for the Fourier series of square wave and stores
        /// it in the float array fourierData
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnPlot_Click(object sender, EventArgs e)
        {
            float y, t, tIncr;
            int i, k;
            terms = (int)nudTerms.Value;
            tIncr = (xWLDMAX - xWLDMIN)/xPOINTS;
            t = xWLDMIN;
            for(i=0;i<xPOINTS;i++)
            {
                y = 0;
                if(rdoSquare.Checked)
                {
                    for(k=1;k<=2*terms;k+=2)
                        y += Fourier(t, k);
                }
                else
                {
                    for(k=1;k<=terms;k++)
                        y += Fourier(t, k);
                }
                TranslateXY(t, y, out fourierData[i, 0], out fourierData[i, 1]);
                t += tIncr;
            }
            dataValid = true;
            pnlPlot.Invalidate();
        }

        /// <summary>
        /// Paint event for the pnlPlot
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void pnlPlot_Paint(object sender, PaintEventArgs e)
        {
            float x, y, xOld, yOld;
            int i;

```

```

Font fontTNR14 = new Font("TimesNewRoman", 14);
Brush blkBrush = new SolidBrush(Color.Black);
Graphics grphPanel = e.Graphics;
Pen blkPen = new Pen(Color.Black);
grphPanel.DrawRectangle(blkPen, 0, 0, pnlPlot.Width-1, pnlPlot.Height-1);
if(dataValid)
    {xOld = fourierData[0,0];
    yOld = fourierData[0,1];
    for(i=1;i<=fourierData.GetUpperBound(0);i++)
        {x = fourierData[i,0];
        y = fourierData[i,1];
        grphPanel.DrawLine(blkPen, xOld, yOld, x, y);
        xOld = x; yOld = y;
        }
    }
DrawAxis(grphPanel);
DrawTicMarks(grphPanel);
}
/// <summary>
/// Fourier
/// Calculates one term of the Fourier series for a square wave.
/// </summary>
/// <param name="t">a float time variable</param>
/// <param name="k">an int term number</param>
/// <returns></returns>
private float Fourier(float t, int k)
    {if(rdoSquare.Checked)
        return (float)((4/(k*Math.PI))*(Math.Sin(k*t)));
    else
        return (float)(.75*((Math.Pow(-1,k+1)/k)*(Math.Sin(k*t))));
    }
/// <summary>
/// TranslateXY
/// Translates x and y from world coordinates to screen coordinates
/// </summary>
/// <param name="xw">x in the world</param>
/// <param name="yw">y in the world</param>
/// <param name="xs">out parameter x on the screen</param>
/// <param name="ys">out parameter y on the screen</param>
private void TranslateXY(float xw, float yw, out float xs, out float ys)
    {float xOFFSET = pnlPlot.Width/2;
    float yOFFSET = pnlPlot.Height/2;
    xs = xw*pnlPlot.Size.Width/(xWLDMAX - xWLDMIN) + xOFFSET;
    ys = -yw*pnlPlot.Size.Height/(yWLDMAX - yWLDMIN) + yOFFSET;
    }
/// <summary>
/// DrawTicMarks
/// Puts tic marks on the x and y axis. Axis is in blue (just for fun)
/// The tic mark increment is 1 unit.
/// </summary>
/// <param name="grphPanel">A graphics object on which to draw</param>
private void DrawTicMarks(Graphics grphPanel)
    {float x1, y1, x2, y2;
    int ticIncr = 1;
    int x, y;
    Pen bluPen = new Pen(Color.Blue);
    for(x= (int)xWLDMIN;x<=xWLDMAX;x+=ticIncr) //along x axis
        {TranslateXY(x, -.05f, out x1, out y1);
        TranslateXY(x, 0.05f, out x2, out y2);
        grphPanel.DrawLine(bluPen, x1, y1, x2, y2);
        }
    for(y= (int)(yWLDMIN);y<=yWLDMAX;y+=ticIncr) //along y axis
        {TranslateXY(-.05f, y, out x1, out y1);
        TranslateXY(0.05f, y, out x2, out y2);
    }
}

```

```
        grphPanel.DrawLine(bluePen, x1, y1, x2, y2);
    }
}
/// <summary>
/// DrawAxis
/// Draws an x and y axis on the screen
/// </summary>
/// <param name="grphPanel">A graphics object on which to draw</param>
private void DrawAxis(Graphics grphPanel)
{float x1, y1, x2, y2;
  Pen blkPen = new Pen(Color.Black);
  TranslateXY(xWLDMIN, 0, out x1, out y1); //x axis
  TranslateXY(xWLDMAX, 0, out x2, out y2);
  grphPanel.DrawLine(blkPen, x1, y1, x2, y2);
  TranslateXY(0, yWLDMIN, out x1, out y1); //y axis
  TranslateXY(0, yWLDMAX, out x2, out y2);
  grphPanel.DrawLine(blkPen, x1, y1, x2, y2);
}
}
```