University of Zurich - HS 2013

Computational Science I
Exercise notes: Matrices

Tobias Grubenmann

October 13, 2013

Exercise 1

First, we need to get the equations for the resistor-cube problem:

$$
\begin{aligned}
\frac{V_0 - V_1}{R_{01}} + \frac{V_0 - V_3}{R_{03}} + \frac{V_0 - V_5}{R_{05}} + I_0 &= 0 \\
\frac{V_1 - V_0}{R_{01}} + \frac{V_1 - V_2}{R_{12}} + \frac{V_1 - V_6}{R_{16}} &= 0 \\
\frac{V_2 - V_1}{R_{12}} + \frac{V_2 - V_3}{R_{23}} + \frac{V_2 - V_7}{R_{27}} &= 0 \\
\frac{V_3 - V_0}{R_{03}} + \frac{V_3 - V_2}{R_{23}} + \frac{V_3 - V_4}{R_{34}} &= 0 \\
\frac{V_4 - V_3}{R_{34}} + \frac{V_4 - V_5}{R_{45}} + \frac{V_4 - V_7}{R_{47}} &= 0 \\
\frac{V_5 - V_0}{R_{05}} + \frac{V_5 - V_4}{R_{45}} + \frac{V_5 - V_6}{R_{56}} &= 0 \\
\frac{V_6 - V_1}{R_{16}} + \frac{V_6 - V_5}{R_{56}} + \frac{V_6 - V_7}{R_{67}} &= 0 \\
\frac{V_7 - V_2}{R_{27}} + \frac{V_7 - V_4}{R_{47}} + \frac{V_7 - V_6}{R_{67}} + I_7 &= 0
\end{aligned}
$$

From this equations we get the following matrix equation:

$$\begin{pmatrix} (\frac{1}{R_{01}}+\frac{1}{R_{03}}+\frac{1}{R_{05}}) & -\frac{1}{R_{01}} & 0 & -\frac{1}{R_{03}} & 0 & -\frac{1}{R_{05}} & 0 & 0 & 1 & 0 \\ & & & & \vdots & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ I_0 \\ I_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V_0 \\ V_7 \end{pmatrix}$$

Now we can implement a Python script that evaluates the total resistance between the edges 0 and 7 given the values for the resistors as well as the voltages V_0 and V_7:

```
from scipy.linalg import solve

def getTotalResistance(R01, R03, R05, R12, R16, R23, R27,
   R34, R45, R47, R56, R67, V0, V7):

    # create matrices
    A = [[1/R01+1/R03+1/R05, -1/R01, 0, -1/R03, 0, -1/R05,
       0, 0, 1, 0],
         [-1/R01, 1/R01+1/R12+1/R16, -1/R12, 0, 0, 0, -1/R16
            , 0, 0, 0],
         [0, -1/R12, 1/R12+1/R23+1/R27, -1/R23, 0, 0, 0, -1/
            R27, 0, 0],
         [-1/R03, 0, -1/R23, 1/R03+1/R23+1/R34, -1/R34, 0,
            0, 0, 0, 0],
         [0, 0, 0, -1/R34, 1/R34+1/R45+1/R47, -1/R45, 0, -1/
            R47, 0, 0],
         [-1/R05, 0, 0, 0, -1/R45, 1/R05+1/R45+1/R56, -1/R56
            , 0, 0, 0],
         [0, -1/R16, 0, 0, 0, -1/R56, 1/R16+1/R56+1/R67, -1/
            R67, 0, 0],
         [0, 0, -1/R27, 0, -1/R47, 0, -1/R67, 1/R27+1/R47+1/
            R67, 0, 1],
         [1, 0, 0, 0, 0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]]
    B = [[0], [0], [0], [0], [0], [0], [0], [0], [V0], [V7
       ]]

    X = solve(matrix(A), matrix(B))
```

```
    return ((V7-V0)/(X[8]))
```

With the function getTotalResistance we can now calculate the total resistance of a cube where all resistors have 1Ω and $V_0 = 0V$, $V_1 = 1V$. We get the a total resistance of 0.833333Ω.

Exercise 2

The function fft recursively calculates the fast Fourier transform of a function f. The function testFFT calculates the FFT for the function $f = \frac{1}{1+x^2}$.

```
from numpy import pi, arange, exp, concatenate
from pylab import subplot, plot, show
import sys

def testFFT():

    N = 64
    L = 8

    dx = 2.*L/N
    x = (arange(N) - N/2) * dx
    k = 2 * pi/(N * dx) * (arange(N) - N/2)
    f = 1/(1 + x*x)

    F = fft(f)

    F = concatenate((F[N/2:N], F[0:N/2]))

    subplot(212)
    plot(x, f)
    subplot(211)
    plot(k, F.real, color = "blue")
    plot(k, F.imag, color = "magenta")


def fft(f):

    N = len(f)

    if N%2 == 0:
```

```
        # split sum

        F_even = fft(f[::2])
        F_odd = fft(f[1::2])

        # get sums together

        w = exp(-2 * pi * 1j * arange(N)/N)

        F = concatenate(([F_even + w[:N/2] * F_odd,
                                F_even + w[N/2:] * F_odd]))

    else:

        if N > 1:
            print("Error: N must be a power of 2.")
            sys.exit(0)
        else:
            F = [f[0]]

    return F
```

The function gives the following output:

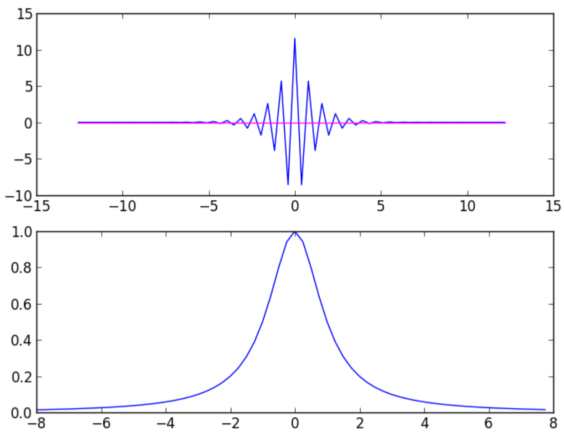

Figure 1: FFT of $f = \frac{1}{1+x^2}$ with $N = 64$.