Scilab Textbook Companion for Fundamental of Electrical and Electronic Principles
by C. R. Robertson[1]

Created by
Nusrat Ali
Basic Electronics
Electronics Engineering
Model Instituite of Engineering and Technology
College Teacher
Bhavana Sharma
Cross-Checked by
Lavitha Pereira

May 24, 2016

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Fundamental of Electrical and Electronic Principles

Author: C. R. Robertson

Publisher: Newens (Elsevier)

Edition: 3

Year: 2008

ISBN: 978-0-7506-8737-9

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 1

Fundamentals

Scilab code Exa 1.1 Standard form and Scientific notation

```
// Scilab code Ex1.1: Pg 3 (2008)
clc; clear;
I =.000018;     // Electric current, A
V = 15000;     // Electric potential, V
P = 250000000     // Electric Power, W
// Display standard form
printf("\nStandard form:");
printf("\n===============");
printf("\n%f A = %3.1e A", I, I);
printf("\n%5.0f V = %3.1e V", V, V);
printf("\n%9.0f W = %3.1e W", P, P);
// Display scientific notation
printf("\n\nScientific form:");
printf("\n=================");
printf("\n%f A = %2d micro-ampere", I, I/1e-06);
printf("\n%5.0f V = %2d kilo-volt", V, V/1e+03);
printf("\n%9.0f W = %3d mega-watt", P, P/1e+06);

// Result
// Standard form:
// ===============
```

```
// 0.000018 A = 1.8e-005 A
// 15000 V = 1.5e+004 V
// 250000000 W = 2.5e+008 W

// Scientific form:
// ==================
// 0.000018 A = 18 micro-ampere
// 15000 V = 15 kilo-volt
// 250000000 W = 250 mega-watt
```

Scilab code Exa 1.2 Quantities in scientific notation

```
// Scilab code Ex1.2: Pg.4 (2008)
clc; clear;
I = 25e-05;     // Electric Current,A
P = 3e+04;     // Electric Power, W
W = 850000;     // Work done, J
V = 0.0016;     // Electric Potential, V
printf("\n\nScientific (Engineering) notation:");
printf("\n========================================");
printf("\n%2e A = %3d micro-ampere = %3.2f mA", I, I/1e-06, I/1e-03);
printf("\n%1.0e W = %3d micro-watt", P, P/1e-06);
printf("\n%6d J = %3d kJ = %3.2f MJ", W, W/1e+03, W/1e+06);
printf("\n%5.4f V = %3.1f milli-volt", V, V/1e-03);

// Result
// Scientific (Engineering) notation:
// ========================================
//2.500000e-004 A = 250 micro-ampere = 0.25 mA
// 3e+004 W = -64771072 micro-watt
// 850000 J = 850 kJ = 0.85 MJ
// 0.0016 V = 1.6 milli-volt
```

Scilab code Exa 1.3 Resulting acceleration of the mass

```
// Scilab code Ex1.3: Pg.5 (2008)
clc; clear;
m = 750/1e+03;     // Mass of the body, kg
F = 2;     // Force acting on the mass, N
// Since F = m * a, (Newton's Second Law of motion),
   solving for a
a = F/m;     // Acceleration produced in the body,
   metre per second square
printf("\nThe acceleration produced in the body = %5
   .3f metre per second square", a)

// Result
// The acceleration produced in the body = 2.667
   metre per second square
```

Scilab code Exa 1.4 Current flowing between two points

```
// Scilab code Ex1.4: Pg.9 (2008)
clc; clear;
Q = 35e-03;     // Electric charge, C
t = 20e-03;     // Time for transference of charge
   between two points, s
// Since Q = I * t, solving for I
I = Q/t;     // Electric current flowing between the
   two points, A
printf("\nThe value of electric current flowing = %4
   .2f A", I);

// Result
// The value of electric current flowing = 1.75 A
```

Scilab code Exa 1.5 Amount of charge transferred in a given time

```
// Scilab code Ex1.5: Pg.9 (2008)
clc; clear;
I = 120e-06;     // Electric current , A
t = 15;     // Time for transference of charge
    between two points , s
// Since I = Q/t, solving for Q
Q = I*t;     // Electric chrage transferred , C
printf("\nThe value of electric charge transferred =
     %3.1f mC", Q/1e-03);

// Result
// The value of electric charge transferred = 1.8 mC
```

Scilab code Exa 1.6 Time for current flow

```
// Scilab code Ex1.6: Pg.10 (2008)
clc; clear;
Q = 80;      // Electric charge , C
I = 0.5;      // Electric current , A
// Since Q = I*t, solving for t
t = Q/I;      // Time for transference of charge
    between two points , s
printf("\nThe duration of time for which the current
     flowed = %3d s", t);

// Result
// The duration of time for which the current flowed
     = 160 s
```

Scilab code Exa 1.7 Potential difference developed across a resistor

```
// Scilab code Ex1.7: Pg.13 (2008)
clc; clear;
I = 5.5e-03;          // Electric current , A
R = 33000;            // Resistance , ohms
// From Ohm's law , V = I*R
V = I*R;              // Potential difference across
    resistor , V
printf("\nThe potential difference developed across
    resistor = %5.1f V",V)

// Result
// The potential difference developed across
    resistor = 181.5 V
```

Scilab code Exa 1.8 Current flowing through a resistor

```
// Scilab code Ex1.8: Pg 14 (2008)
clc; clear;
V = 24;              // Potential difference ,V
R = 15;               // Resistance , ohms
// From Ohm's law , V = I*R, then solving for I
I = V/R;               // Electric current , A
printf("\nThe current flowing through the resistor =
    %3.1f A", I)

// Result
// The current flowing through the resistor  = 1.6 A
```

Scilab code Exa 1.9 Terminal potential difference for a given current

```
// Scilab code Ex1.9: Pg 16 (2008)
clc; clear;
E = 6;             // E.m.f of battery, V
r = 0.15;          // Internal resistance of battery, ohm
I_1 = .5;          // Electric current, A
I_2 = 2;            // Electric current, A
I_3 = 10;            // Electric current, A
// Using relation V = E - I*R and substituting the values of I_1, I_2 and I_3 one by one in it
V_1 = E - I_1*r;       // Terminal potential difference, V
V_2 = E - I_2*r;        // Terminal potential difference, V
V_3 = E - I_3*r;         // Terminal potential difference, V
printf("\nThe terminal potential difference developed across resistor for a current of %3.1f A = %5.3f V",I_1,V_1)
printf("\nThe terminal potential difference developed across resistor for a current of %1d A = %3.1f V",I_2,V_2)
printf("\nThe terminal potential difference developed across resistor for a current of %2d A = %3.1f V",I_3,V_3);

// Result
// The terminal potential difference developed across resistor for a current of 0.5 A = 5.925 V
// The terminal potential difference developed across resistor for a current of 2 A = 5.7 V
// The terminal potential difference developed across resistor for a current of 10 A = 4.5 V
```

Scilab code Exa 1.10 Resitances in a battery circuit

```
// Scilab code Ex1.10: Pg 16 (2008)
clc; clear;
E = 12;          // E.m.f, V
I = 5;           // Electric current, A
V = 11.5;        // Terminal potential difference, V
// Using relation V = E - I*r, solving for r
r = ( E - V )/I;     // Internal resistance of
   battery, ohm
// From Ohm's law, V = I*R, then solving for R
R = V/I;              // Resistance, ohms
printf("\nThe internal resistance of battery = %3.1f
    ohm", r)
printf("\nThe resistance of external circuit = %3.1f
    ohm", R)

// Result
// The internal resistance of battery = 0.1 ohm
// The resistance of external circuit = 2.3 ohm
```

Scilab code Exa 1.11 Potential difference and energy dissipated across a resistor

```
// Scilab code Ex1.11:Pg 17 (2008)
clc; clear;
I = 200e-03;        // Electric current, A
t = 300;             // Time for which current flows,
    s
R = 750;             // Resistance, ohms
// Using Ohm's law, V = I*R
```

```
V = I*R;              // Electric potential difference, V
W = I^2*R*t;           // Energy dissipated, joule
printf("\nThe potential difference developed across the resistor = %3d V\nThe energy dissipated across the resistor = %4.0f J or %1d kJ", V, W, W*1e-03)

// Result
// The potential difference developed across the resistor = 150 V
// The energy dissipated across the resistor = 9000 J or 9 kJ
```

Scilab code Exa 1.12 Potential difference and current from electric power

```
// Scilab code Ex1.12: Pg 18 (2008)
clc; clear;
R = 680;              // Resistance, ohms
P = 85e-03;             // Electric power, W
// Using P = V^2/R, solving for V
V = sqrt( P*R );          // Potential difference, V
// Using P = I^2*R, solving for I
I = sqrt( P/R );           // Electric current, A
printf("\nThe potential difference developed across the resistance = %3.1f V\nThe current flowing through the resistor = %5.2f mA", V, I/1e-03)

// Result
// The potential difference developed across the resistance = 7.6 V
// The current flowing through the resistor = 11.18 mA
```

Scilab code Exa 1.13 Potential difference and power across a resistor

```
// Scilab code Ex1.13:Pg 19 (2008)
clc; clear;
I = 1.4;            // Electric current , A
t = 900;            // Time for which current flows ,
    s
W = 200000;         // Energy dissipated , J
// Using relation W = V*I*t , solving for V
V = W/( I*t );        // Potential difference , V
// Using relation P = V*I
P = V*I;              // Electric power , W
// From Ohm's law , V = I*R, solving for R
R = V/I;               // Resistance , ohm
printf("\nThe potential difference developed = %5.1 f
     V\nThe power dissipated = %5.1 f W\nThe
    resistance of the circuit = %5.1 f ohm", V, P, R)

// Result
// The potential difference developed = 158.7 V
// The power dissipated = 222.2 W
// The resistance of the circuit = 113.4 ohm
```

Scilab code Exa 1.14 Cost of machine operation

```
// Scilab code Ex1.14: Pg 20 (2008)
clc; clear;
P = 12.5;      // Power of the machine , kW
t = 8.5;     // Time for which the machine is
    operated , h
W = P*t;           // Electric energy , kWh
```

```
// Cost per unit = 7.902 p, therefore calulating the
    cost of 106.25 units
cost = ( W*7.902 );        // Cost for operating
   machine, p
printf("\nThe cost of operating the machine = %4.2f
   pounds", cost*1e-02)

// Result
// The cost of operating the machine = 8.40 pounds
```

Scilab code Exa 1.15 Cost of unit energy and total electicity bill

```
// Scilab code Ex1.15: Pg 20 (2008)
clc; clear;
Total_bill = 78.75;       // pounds
Standing_charge = 15.00;      // pounds
Units_used = 750;             // kWh
Cost_per_unit = ( Total_bill - Standing_charge )/
   Units_used;       // p
Cost_of_energy_used = 67.50;          // pounds
Total_bill = Cost_of_energy_used + Standing_charge;
         // pounds
printf("\nThe cost per unit = %5.3f pounds or %3.1f
   p\nTotal bill = %5.2f pounds",Cost_per_unit,
   Cost_per_unit/1e-02,Total_bill);

// Result
// The cost per unit = 0.085  Pounds or 8.5 p
// Total bill = 82.50 pounds
```

Scilab code Exa 1.16 Resistance of the copper coil

```
// Scilab code Ex1.16: Pg 22 (2008)
```

```
clc; clear;
l = 200;              // Length of Cu wire, metre
rho = 2e-08;          // Resistivity of Cu, ohm-metre
A = 8e-07;            // Cross sectional area of Cu wire, metre square
// Using relation R = ( rho*l )/A
R = ( rho*l )/A;            // Resistance, ohm
printf("\nThe resistance of the coil = %1d ohm", R)

// Result
// The resistance of the coil = 5 ohm
```

Scilab code Exa 1.17 Resistance of wire wound resistor

```
// Scilab code Ex1.17:  Pg 22 (2008)
clc; clear;
l = 250;              // Length of Cu wire, metre
d = 5e-04;            // Diameter of Cu wire, metre
rho = 1.8e-08;        // Resistivity of Cu wire, ohm-metre
A = ( %pi*d^2 )/4;           // Cross sectional area of Cu wire, metre square
// Using relation R = rho*l/A
R = rho*l/A;           // Resistance, ohm
printf("\nThe resistance of the coil = %5.2f ohm", R)

// Result
// The resistance of the coil = 22.92 ohm
```

Scilab code Exa 1.18 Variation of resistance with temperature

```
// Scilab code Ex1.18: Pg 23 (2008)
```

```

clc; clear;
R_1 = 250;              // Resistance of field coil, ohm
Theta_1 = 15;           // Initial temperature of motor, degree celcius
Theta_2 = 45;           // Final temperature of motor, degree celcius
Alpha = 4.28e-03;        // Temperature coefficient of resistance, per degree celcius
// Using relation, R_1/R_2 = ( 1 + Alpha*Theta_1 )/( 1 + Alpha*Theta_2 ), solving for R_2
R_2 = R_1 * (( 1 + Alpha*Theta_2 )/( 1 + Alpha*Theta_1 ));      // Resistance, ohms
printf("\nThe resistance of field coil at %2d degree celcius = %5.1f ohm",Theta_2, R_2)

// Result
// The resistance of field coil at 45 degree celcius = 280.2 ohm
```

Scilab code Exa 1.19 Resistance of wire at given temperature

```
// Scilab code Ex1.19: Pg 24 (2008)

clc; clear;
R_0 = 350;              // Resistance, ohms
Theta_1 = 60;           // Temperature, degree celcius
Alpha = 4.26e-03;        // Temperature coefficient, per degree celcius
// Using relation R_1 = R_0 * ( 1 + Alpha*Theta_1 )
R_1 = R_0 * ( 1 + Alpha*Theta_1 );        // Resistance, ohms
printf("\nThe resistance of the wire at %2d degree celcius = %5.1f ohm",Theta_1, R_1)
```

```
// Result
// The resistance of the wire at 60 degree celcius =
    439.5 ohm
```

Scilab code Exa 1.20 Resistance of carbon composite resistor at a given temperature

```
// Scilab code Ex1.20: Pg. 24 (2008)
clc; clear;
R_1 = 120;              // Resistance , ohms
Theta_1 = 16;           // Temperature , degree celcius
Theta_2 = 32;           // Temperature , degree celcius
Alpha = -4.8e-04;        // Temperature coefficient ,
   per degree celcius
// Using relation , R_1/R_2 = ( 1 + Alpha*Theta_1 )/(
    1 + Alpha*Theta_2 ) , solving for R_2
R_2 = R_1 * (( 1 + Alpha*Theta_2 )/( 1 + Alpha*
   Theta_1 ));        // Resistance , ohm
printf("\nThe resistance of carbon resistor at %2d
   degree celcius = %5.1f ohm",Theta_2, R_2)

// Result
// The resistance of field coil at 32 degree celcius
    = 119.1 ohm
```

Chapter 2

DC circuits

Scilab code Exa 2.1 Electrical parameters of resistances in series

```
// scilab code Ex2.1: Pg 32 (2008)
clc; clear;
E = 24;              // E.m.f of battery ,V
R1 = 330;            // Resistance , ohms
R2 = 1500;           // Resistance , ohms
R3 = 470;            // Resistance , ohms
// As resistances R1, R2 & R3 are joined end-to-end
   hence , they are in series & in series connection ,
    circuit resistance is the sum of individual
   resistances present in the circuit
R = R1 + R2 + R3;    // Resistance of circuit , ohms
I = E/R;             // Circuit current , A
// As the resistances are in series so same current
   flows through each resistor & potential drop
   across each resistor is equal to the product of
   circuit current & its respective resistance( from
    Ohm's law , V = I*R )
V1 = I*R1;           // Potential difference
   developed across resistance R1, V
V2 = I*R2;            // Potential difference
   developed across resistance R2, V
```

```
V3 = I*R3;                    // Potential difference developed across resistance R3, V
P = E*I;                      // Electric power dissipated by the complete circuit, W
printf("\nThe circuit resistance = %4d ohms or %3.1f kilo-ohms", R, R*1e-03);
printf("\nThe circuit current = %5.2f milli-ampere", I/1e-03);
printf("\nThe potential drop across resisatnce R1 = %4.2f volts\nThe potential drop across resistance R2 = %5.2f volts\nThe potential drop across resistance R3 = %4.2f volts", V1, V2, V3);
printf("\nThe power dissipated by the complete circuit = %4.2f watt or %3d milli-watt", P,P/1e-03 );

// Result
// The circuit resistance = 2300 ohms or 2.3 kilo-ohms
// The circuit current = 10.43 milli-ampere
// The potential drop across resisatnce R1 = 3.44 volts
// The potential drop across resistance R2 = 15.65 volts
// The potential drop across resistance R3 = 4.90 volts
// The power dissipated by the complete circuit = 0.25 watt or 250 milli-watt
```

Scilab code Exa 2.2 Series combination of resistors

```
// Scilab code Ex2.2: Pg 34 (2008)
clc; clear;
E = 12;                        // E.m.f of battery, V
R_BC = 16;                      // Resistance across branch
```

```
    BC, ohms
P_BC = 4;                    // Electric power
   dissipated by resistance R_BC, W
// using relation P = I^2/R, solving for I
I = sqrt( P_BC/R_BC);      // Electric current ,A
R = E/I;                     // Total circuit
   resistance , ohms
R_AB = R - R_BC;              // Resistance across
   branch AB, ohms
printf("\nThe circuit current = %3.1f A\nThe value
   of other resistor = %1d ohms", I, R_AB);

// Result
// The circuit current = 0.5 A
//The value of other resistor = 8 ohms
```

Scilab code Exa 2.3 Resistances in parallel

```
// Scilab code Ex2.3: Pg 37 (2008)
clc; clear;
E = 24;                  // E.m.f of battery , V
R_1 = 330;               // Resistance , ohms
R_2 = 1500;               //Resistance , ohms
R_3 = 470;                 //Resistance , ohms
// Since one end of each resistor is connected to
   positive terminal of battery and the other end to
    the negative terminal , therefore , the resistors
   are in parallel & in parallel connection the
   equivalent resistance of  the circuit is equal to
    the reciprocal of the sum of conductances of
   individual resistances present in the circuit i.e
    1/R = 1/R_1 + 1/R_2 + 1/R_3 , solving for R
R = (R_1*R_2*R_3)/( R_1*R_2 + R_2*R_3 + R_3*R_1);
        // Equivalent resisance of circuit , ohms
// Since the resistances are in parallel so potetial
```

```
difference across each resistor is same & in our case is equal to e.m.f of battery & from Ohm's law, V = I*R, solving for I
I_1 = E/R_1;            // Current through resistor R_1, A
I_2 = E/R_2;            // Current through resistor R_2, A
I_3 = E/R_3;            // Current through resistance R_3, A
// Current drawn from battery is equal to the sum of branch currents
I = I_1 + I_2 + I_3;                    // Current drawn from battery, A
printf("\nThe total resistance of the circuit = %6.2f ohms",R);
printf("\nThe branch current I1 = %5.2f mA\nThe branch current I2 = %2d mA\nThe branch current I3 = %5.2f mA", I_1/1e-03, I_2/1e-03, I_3/1e-03);
printf("\nThe current drawn from the battery = %5.1f mA", I/1e-03);

// Result
// The total resistance of the circuit = 171.68 ohms
// The branch current I1 = 72.73 mA
// The branch current I2 = 16 mA
// The branch current I3 = 51.06 mA
// The current drawn from the battery = 139.8 mA
```

Scilab code Exa 2.4 Two parallel resistances across a voltage source

```
// scilab code Ex2.4: Pg 39 (2008)
clc; clear;
E = 12;                 // E.m.f of battery, V
R1 = 6;                 // Resistance, ohms
R2 = 3;                 // Resistance, ohms
```

```
// Since the two resistances are in parallel , therefore effective resistance of the circuit is equal to the reciprocal of the sum of conductances ( 1/Ressistance) of individual resistances present in the circuit i.e 1/R = 1/R1 + 1/R2, simplifying for R
R = ( R1*R2)/(R1 + R2);           // Effective resistance of the circuit , ohms
// Fron Ohm's law , V = I*R, solving for I
I = E/R;                         // Circuit current , A
I1 = E/R1;                       // Current through resistance R1, A
I2 = E/R2;                       // Current thrugh resistance R2, A
printf("\nEffective resistance of the circuit = %1d ohms", R);
printf("\nThe current drawn from the battery = %1d A", I);
printf("\nThe current through resistor R1 = %1d A", I1);
printf("\nThe current through R2 resistor = %1d A", I2);

// Result
// Effective resistance of the circuit = 2 ohms
// The current drawn from the battery = 6 A
// The current through resistor R1 = 2 A
// The current through R2 resistor = 4 A
```

Scilab code Exa 2.5 Effective resistance in series and parallel connected resistors

```
// Scilab code Ex2.5: Pg 39-40 (2008)
clc; clear;
R1 = 10;                        // Resistance , ohm
```

```
R2 = 20;                        // Resistance , ohm
R3 = 30;                        // Resistance , ohm
// Part (a)
// Since in sreies combination , the equivalent
   resistance of the circuit is the sum of the
   individual resistances present in the circuit i.e
    R = R1 + R2 + R3
R_s = R1 + R2 + R3;              // Equivalent series
   resistance of the circuit , ohms
// Part (b)
// Since in parallel combination , the equivalent
   resistance of the circuit is the reciprocal of
   the  sum of the conductances of the individual
   resistances present in the circuit i.e 1/R = 1/R1
    + 1/R2 + 1/R3, solving for R;
R_p = ( R1*R2*R3 )/( R1*R2 + R2*R3 + R3*R1 );
   // Equivalent parallel resistance of the circuit ,
    ohms
printf("\nEquivalent series resistance of the
   circuit = %2d ohm", R_s);
printf("\nEquivalent parallel resistance of the
   circuit = %4.2f ohm", R_p);

// Result
// Equivalent series resistance of the circuit = 60
   ohm
// Equivalent parallel resistance of the circuit =
   5.45 ohm
```

Scilab code Exa 2.6 Series parallel combination across a voltage source

```
// Scilab code Ex2.6: Pg 43 (2008)
clc; clear;
E = 64;                          // E.m.f of battery , V
R1 = 6;                         // Resistance , ohm
```

```
R2 = 4;                          // Resistance , ohm
// Part (a)
// Since R1 & R2 are  parallel to one another hence, their equivalent resistance is equal to the sum of reciprocal of their individual resistances
R_BC = ( R1*R2)/( R1 + R2 );            // Equivalent resistance across branch BC, ohm
R_AB = 5.6;                             // Resistance across branch AB, ohm
// Since R_AB & R_BC are in series , therefore , their equivalent resistance is equal to the sum of their individual resistances
R_AC = R_AB + R_BC;                         // Total circuit resistance , ohm
// From Ohm's law , V = I*R, solving for I
I = E/R_AC;                                   // Total circuit current , A
// Part (b)
V_BC = I*R_BC;                                // Potential difference across branch BC, V
I1 = V_BC/R1;                                 // Electric current through resistor R1, A
// Part (c)
// Since P = I^2*R
P_AB = I^2*R_AB;                              // Power dissipated by 5.6 ohm resistance , W
printf("\nThe current drawn fron the supply = %1d A ", I);
printf("\nThe current through %1d ohm resistor = %3.1f A", R1, I1);
printf("\nThe power dissipated by %3.1f ohm resistor = %5.1f W", R_AB, P_AB);

// Result
// The current drawn fron the supply = 8 A
// The current through 6 ohm resistor = 3.2 A
// The power dissipated by 5.6 ohm resistor = 358.4 W
```

Scilab code Exa 2.7 Electrical potential and current distribution in a series parallel network

```
// Scilab code Ex2.7: Pg 46 (2008)
clc; clear;
E = 18;                         // E.m.f of battery, V
R1 = 4;                        // Resistance, ohm
R2 = 6;                        // Resistance, ohm
R3 = 5;                        // Resistance, ohm
R4 = 3;                        // Resistance, ohm
R5 = 6;                        // Resistance, ohm
R6 = 8;                        // Resistance, ohm
// Part (a)
// Since resistance R1 & R2 are in parallel, therefore, equivalent resistance across branch AB will be equal to the reciprocal of the sum of conductances ( 1/Ressistance) of individual resistances present in the circuit i.e 1/R_AB = 1/R1 + 1/R2, simplifying for R_AB
R_AB = ( R1*R2 )/( R1 + R2);                // Resistance, ohm
R_BC = R3;                                  // Resistance across branch BC, ohm
// Since resistance R4, R5 & R6 are in parallel, therefore, equivalent resistance across branch CD will be equal to the reciprocal of the sum of conductances ( 1/Ressistance) of individual resistances present in the circuit i.e 1/R_CD = 1/R4 + 1/R5 + 1/R6, simplifying for R _CD
R_CD = ( R4*R5*R6 )/( R4*R5 + R5*R6 + R6*R4 );     // Resistance, ohm
// Since R_AB, R_BC & R_CD forms series combination, therefore circuit resistance will be their series sum
```

```
R = R_AB + R_BC + R_CD;                  // Circuit resistance, ohm
I = E/R;                      // Supply current, A
// Part (b)
// AS resistances R1 & R2 are parallel, therefore tere will be same potential difference across them, denoted by V_AB
V_AB = I*R_AB;                           // Potential difference, V
// AS resistances R4, R5 & R6 are parallel, therefore tere will be same potential difference across them, denoted by V_CD
V_CD = I*R_CD;                      // Potential difference, V
V_BC = I*R_BC;                      // Potential difference, V
// Part (c)
I1 = V_AB/R1;                       // Current through R1 resistor, A
I2 = V_AB/R2;                       // Current through R2 resistor, A
I4 = V_CD/R4;                      // Current through R4 resistor, A
I5 = V_CD/R5;                      // Current through R5 resistor, A
I6 = V_CD/R6;                      // Current through R6 resistor, A
// Part (d)
P3= I^2*R3;                         // Power dissipated, W
printf("\nThe current drawn from the source = %1d A", I);
printf("\nThe p.d. across resistor %1d ohm & %1d ohm = %3.1f V", R1, R2, V_AB);
printf("\nThe p.d. across resistor %1d ohm, %1d ohm & %1d ohm = %3.1f V", R4, R5, R6, V_CD);
printf("\nThe p.d. across resistor %1d ohm = %2d V", R3, V_BC);
```

```
printf("\nThe current through resistor %1d ohm = %3.1f A", R1, I1);
printf("\nThe current through resistor %1d ohm = %3.1f A", R2, I2);
printf("\nThe current through resistor %1d ohm = %1d A", R3, I);
printf("\nThe current through resistor %1d ohm = %5.3f A", R4, I4);
printf("\nThe current through resistor %1d ohm = %5.3f A", R5,I5);
printf("\nThe current through resistor %1d ohm = %3.1f A", R6, I6);
printf("\nThe power dissipated by the %1d ohm resistor = %2d W', R3, P3);

// Result
// The current drawn from the source = 2 A
// The p.d. across resistor 4 ohm & 6 ohm = 4.8 V
// The p.d. across resistor 3 ohm, 6 ohm & 8 ohms = 3.2 V
// The p.d. across resistor 5 ohm = 10 V
// The current through resistor 4 ohm = 1.2 A
// The current through resistor 6 ohm = 0.8 A
// The current through resistor 5 ohm = 2 A
// The current through resistor 3 ohm = 1.067 A
// The current through resistor 6 ohm = 0.533 A
// The current through resistor 8 ohm = 0.4 A
// The power dissipated by the 5 ohm resistor = 20 W
```

Scilab code Exa 2.8 Electric current distribution in a network

```
// Scilab code Ex2.8: 49 (2008)
clc; clear;
// Applying Kirchhoff s current law (the sum of the currents arriving at a junction is equal
```

```
to the sum of the currents leaving that junction) at junction A
I2 = 40 + 10;                       // Electric current, A
// Applying Kirchhoff s current law at junction C
I1 = 80 - I2;                       // Electric current, A
// Applying Kirchhoff s current law at junction D
I3 = 80 + 30;                       // Electric current, A
// Applying Kirchhoff s current law at junction E
I4 = I3 - 25;                        // Electris current, A
// Applying Kirchhoff s current law at junction F
I5 = 30 - 85;                        // Electric cuurent, A
printf("\nCurrent I1 = %2d A\nCurrent I2 = %2d A\nCurrent I3 = %3d A\nCurrent I4 = %2d A\nCurrent I5 = %2d A,", I1, I2, I3, I4, I5);

// Result
// Current I1 = 30 A
// Current I2 = 50 A
// Current I3 = 110 A
// Current I4 = 85 A
// Current I5 = -55 A
```

Scilab code Exa 2.9 Kirchhoff laws applied to an electrical network

```
// Scilab code Ex2.9: Pg 52-53 (2008)
clc; clear;
```

```
R1 = 3;                    // Resisance , ohms
R2 = 2;                    // Resistance , ohms
R3 = 10;                   // Resistance , ohms
E1 = 10;                   // E.m.f , V
E2 = 4;                     // E.m.f , V
// Applying Kirchhoff          s Current Law(the sum of the currents arriving at a junction is equal to the sum of the currents leaving that junction)
A = [3 -2; 13 10];
B = [6; 10];
X = inv(A)*B;
I1 = X(1,:);                          // Electric current through branch FA, A
I2 = X(2,:);                          // Eleactric current through branch EB, A
I3 = ( I1 + I2 );                     // Electric current through branch CD, A
V_CD = R3*I3;                         // P.d.across R3 resistor , V
printf("\nThe current through branch FA = %6.3f A", I1);
printf("\nThe current through branch EB = %5.3f A", I2);
printf("\nThe current through branch CD = %5.3f A", I3);
printf("\np.d.across %2d resistor = %4.2f V", R3, V_CD);

// Result
// The current through branch FA = 1.429 A
// The current through branch FA = -0.857 A
// The current through branch FA = 0.571 A
// p.d.across %2d resistor = 5.71 V
```

Scilab code Exa 2.10 Electric current and voltage from Kirchhoff law

```
// Scilab code Ex2.10: Pg 53 (2008)
clc; clear;
E1 = 6;                          // E.m.f of battery, V
E2 = 4.5;                        // E.m.f of battery, V
R1 = 1.5;                        // Resistance, ohm
R2 = 2;                          // Resistance, ohm
R3 = 5;                          // Resistance, ohm
// Part (a)
// Using matrix method for solving set of equations
A = [6.5 5; 5 7];
B = [6; 4.5];
X = inv(A)*B;
I1 = X(1,:);                     // Electric current through branch FA, A
I2 = X(2,:);                     // Electric current through branch DC, A
I3 = ( I1 + I2);                 // Electric current through branch BE, A
// Part (b)
V_BE = I3*R3;                    // P.d across resistor R3, V
printf("\nElectric current through branch FA = %5.3f A", I1);
printf("\nElectric current through branch DC = %6.4f A", I2);
printf("\nElectric current through branch BE = %5.3f A", I3);
printf("\np.d across resistor %1d ohms = %5.3f V", R3, V_BE);

// Result
// Electric current through branch FA = 0.951 A
// Electric current through branch DC = -0.0366 A
// Electric current through branch FA = 0.915 A
// p.d across resistor %1d ohms = 4.573 V
```

Scilab code Exa 2.11 Current distribution in a wheatstone bridge network using Kirchhoff law

```
// Scilab code Ex2.11: Pg 57 (2008)
clc; clear;
R_AB = 6;                         // Resistance, ohm
R_BC = 4;                         // Resistance, ohm
R_DC = 1;                         // Resistance, ohm
R_AD = 3;                         // Resistance, ohm
R_BD = 5;                         // Resistance, ohm
// Using matrix method for solving the set of equations
A = [6 -3 5; -4 1 10; 0 4 1];
B = [0; 0; 10];
X = inv(A)*B;
I1 = X(1,:);                                // Electric current, A
I2 = X(2,:);                               // Electric current, A
I3 = X(3,:);                              // Electric current, A
I_BC = I1 - I3;                          // Electric current, A
I_DC = I2 + I3;                         // Electric current, A
I = I1 + I2;                           // Suplly current, A
printf("\nThe current through %1d ohm resistor = %5.3f A", R_AB, I1);
printf("\nThe current through %1d ohm resistor = %4.2f A", R_BC, I_BC);
printf("\nThe current through %1d ohm resistor = %5.3f A", R_DC, I_DC);
printf("\nThe current through %1d ohm resistor = %5.3f A", R_AD, I2);
```

```
printf("\nThe current through %1d ohm resistor = %5.3f A", R_BD, I3);
printf("\nThe supply current = %5.3f A", I)

// Result
// The current through 6 ohm resistor = 1.074 A
// The current through 4 ohm resistor = 0.89 A
// The current through 1 ohm resistor = 2.638 A
// The current through 3 ohm resistor = 2.454 A
// The current through 5 ohms resistor = 0.184 A
// The supply current = 3.529 A
```

Scilab code Exa 2.12 Current through central resistor in a balanced Wheatstone bridge

```
// Scilab code Ex2.12: Pg 58-59 (2008)
clc; clear;
R_AB = 6;                    // Resistance across branch AB, ohm
R_AD = 3;                    // Resistance across branch AD, ohm
R_BC = 4;                    // Resistance across branch BC, ohm
R_DC = 2;                    // Resistance across branch DC, ohm
// Since R_AB/R_AD = R_BC/R_DC, so the wheatstone bridge is balanced hence no current flows through branch BD
I3 = 0;
printf("\nThe current through branch BD i.e I3 = %1d A", I3);

// Result
// The current through branch BD i.e I3 = 0 A
```

Scilab code Exa 2.13 Balancing a wheatstone bridge

```
// Scilab code Ex2.13:  Pg 62-63 (2008)
clc; clear;
R1 = 20;                         // Resistance, ohm
R2 = 10;                         // Resistance, ohm
R3 = 8;                         // Resistance, ohm
R4 = 5;                         // Resistance, ohm
R5 = 2;                         // Resistance, ohm
A = [20 -10 8; -5 2 15; 0 12 2];
B = [0; 0; 10];
X = inv(A)*B;
I3 = X(3,:);                     // Electric current through BD, A
V_BD = I3*R3;                   // P.d across branch BD, V
// For balance conditions i.e I3 = 0, R1/R2 = R4/R5, solving for R4
R_4 = ( R1*R5 )/R2;              // Resistance, ohm
printf("\nThe p.d between terminals B and D = %5.3f V", V_BD);
printf("\nThe value to which %1d ohm resistor must be adjusted in order to reduce the current through %1d ohm resistor to zero = %1d ohm", R4, R3, R_4);

// Result
// The p.d between terminals B and D = 0.195 V
// The value to which 5 ohm resistor must be adjusted in order to reduce the current through 8 ohm resistor to zero = 4 ohm
```

Scilab code Exa 2.14 Measuring unknown resistances using Wheatstone bridge

```
// Scilab code Ex2.14: Pg 64 (2008)
clc; clear;
// For part (a)
Rm = 1000;                    // Resistance, ohm
Rd = 1;                       // Resistance, ohm
Rv = 3502;                    // Resistance, ohm
// Using Wheatstone bridge balanced condition i.e Rx/Rv = Rm/Rd , solving for Rx
Rx = ( Rm/Rd) * Rv;                // Resistance,ohm
printf("\nThe value of the resistance being measured = %5.3f mega-ohm",Rx*1e-06);

// Part (b)
Rm = 1;                    // Resistance, ohm
Rd = 1000;                         // Resistance, ohm
Rv = 296;                       // Resistance, ohm
// Using Wheatstone bridge balanced condition i.e Rx/Rv = Rm/Rd , solving for Rx
Rx = ( Rm/Rd )*Rv;                // Resistance,ohm
printf("\nThe value of the resistance being measured = %5.3f ohm",Rx);

// Result
// The value of the resistance being measured = 3.502 mega-ohm
// The value of the resistance being measured = 0.296 ohm
```

Scilab code Exa 2.15 Finding cells emf using a potentiometer

```
// Scilab code Ex2.15: Pg 67 (2008)
clc; clear;
```

```
l1 = 600e-03;                  // Scale reading, metre
l2 = 745e-03;                 // Scale reading, metre
l_s = 509.3e-03;               // Total scale length, metre
E_s = 1.0186;                   // Source voltage, V
E1 = ( l1/l_s )*E_s;           // Voltage drop across length l1, V
E2 = ( l2/l_s)*E_s;            // Voltage drop across length l2, V
printf("\nThe emf of the first cell = %3.1f V ", E1)
printf("\nThe emf of the second cell = %3.2f V ", E2)

// Result
// The emf of the first cell = 1.2 V
// The emf of the first cell = 1.49 V
```

Chapter 3

Electric fields and capacitors

Scilab code Exa 3.1 Density of electric field between plates of capacitor

```
// Scilab code Ex3.1: Pg 79 (2008)
clc; clear;
Q = 50e-03;                        // Electric charge, C
A = 600e-06;                       // Area of plate, m^2
// Solving for electric field density, D
D = Q/A;                           // Electric field
    density, C/m^2
printf("\nThe density of the electric field existing
     between the plates = %4.1f C/m-square", D);

// Result
// The density of the electric field existing
    between the plates = 83.3 C/m-square
```

Scilab code Exa 3.2 Charge on plates of capacitor and electric field density between them

```
// Scilab code Ex3.2: Pg 80 (2008)
```

```
clc; clear;
A = 400e-06;                    // Cross-sectional area of
    plate , m^2
I = 50e-06;                    // Source current , A
t = 3;                         // Flow time of current , s
// Since electric current is the rate of flow of
   charge i.e I = Q/t , solving for Q
Q = I*t;                       // Amount of charge on
   plates , C
//Solving for density of the electric field between
   the plates
D = Q/A;                          // Electric field
   density , C/m^2
printf("\The charge on the plates = %3d micro-
   coloumb", Q/1e-06);
printf("\nThe density of the electric field between
   the plates = %5.3f C/m-square", D);

// Result
// The charge on the plates = 150 micro-coloumb
// The density of the electric field between the
   plates =0.375 C/m-square
```

Scilab code Exa 3.3 Electric field strength and flux density of parallel plates capacitor

```
// Scilab code Ex3.3: Pg 83 (2008)
clc; clear;
d = 3e-03;                        // Thickness of
   dielectric , m
Q = 35e-03;                       // Electric charge on
   plates , C
V = 150;                         // Supply voltage , V
A = 144e-06;                    // Cross-sectional area of
    plates , m^2
```

```
// Part (a)
// Since electric field strength(E) = potential gradient therefore we have
E = V/d;                    // Electric field strength, V/m
// Part (b)
// Solving for electric field density, D
D = Q/A;                    // Electric field density, C/m^2
printf("\nThe electric field strength = %2d kV/m", E*1e-03);
printf("\nThe flux density = %5.1f C/m^2", D);

// Result
// The electric field strength = 50 kV/m
// The flux density = 243.1 C/m^2
```

Scilab code Exa 3.4 Characteristics of a parallel plate capacitor

```
// Scilab code Ex3.4: Pg 83-84 (2008)
clc; clear;
d = 4e-03;                  // Thickness of air, m
Q = 2e-04;                  // Electric charge on plates, C
V = 125;                    // Supply voltage, V
D = 15;                     // Electric field density, coulomb-per-metre-square
// Part (a)
// Since electric field strength(E) = potential gradient, therefore we have
E = V/d;                    // Electric field strength, V/m
// Part (b)
// Since D = Q/A, solving for A
A = Q/D;                    // Cross-sectional area of
```

```
plates, m^2
// Part (c)
// Since Q = C*V, solving for C
C = Q/V;                          // Capacitance of the plates, F
printf("\nThe electric field strength between the plates = %5.2f kV/m",E*1e-03);
printf("\nThe csa of the field between the plates = %4.1f mm^2", A/1e-06);
printf("\nThe capacitance of the plates = %3.1f micro-coulomb", C/1e-06);

// Result
// The electric field strength between the plates = 31.25 kV/m
// The csa of the field between the plates = 13.3 mm^2
// The capacitance of the plates = 1.6 micro-coulomb
```

Scilab code Exa 3.5 Capacitance and electric field stength of a parallel plate capacitor

```
// Scilab code Ex3.5: Pg 86 (2008)
clc; clear;
A = 6e-04;                        // Cross-sectional area of plates, m^2
d = 5e-04;                        // Thickness of mica sheet, m
Epsilon_r = 5.8;                  // Relative permittivity, unitless
Epsilon_0 = 8.854e-12;            // Permittivity of Free Space
V = 200;                          // Potential difference, V
// Part (a)
```

```
// Since absolute permittivity, Epsilon = C*(d/A), therefore solving for d & putting Epsilon = Epsilon_0*Epsilon_r
C = ( Epsilon_r*Epsilon_0*A )/d;          // Capacitance, F
// Part (b)
// Since electric field strength(E) = potential gradient, therefore we have
E = V/d;                          // Electric field strength, V/m
printf("\nThe capacitance of the capacitor = %5.2f pF", C/1e-12);
printf("\nElectric field strength = %3d kV/m",E*1e-03);

// Result
// The capacitance of the capacitor = 61.62 pF
// Electric field strength = 400 kV/m
```

Scilab code Exa 3.6 Thickness of paper between plates of a capacitor

```
// Scilab code Ex3.6: Pg 86 (2008)
clc; clear;
C = 0.224e-09;                    //Capacitance, F
A = 5625e-06;                     // Cross-sectional area of plates, m^2
Epsilon_r = 2.5;                  // Relative permittivity
Epsilon_0 = 8.854e-12;            // Permittivity of Free Space
// Since absolute permittivity, Epsilon = C*(d/A), therefore solving for d & putting Epsilon = Epsilon_0*Epsilon_r
d = ( Epsilon_r*Epsilon_0*A )/C;                    // Thickness of waxed paper dielectric, m
```

```
printf("\nThe thickness of paper required = %3.2f mm
   ", d/1e-03);

// Result
// The thickness of paper required = 0.56 mm
```

Scilab code Exa 3.7 Relative permittivity of ceramic dielectric

```
// Scilab code Ex3.7: Pg 86 (2008)
clc; clear;
C = 4.7e-08;                        //Capacitance, F
A = 4e-04;                         // Cross-sectional area
     of plates, m^2
d = 1e-04;                         // Thickness of
    dielectric, m
Epsilon_0 = 8.854e-12;              // Permittivity of
     Free Space
// Since absolute permittivity, Epsilon = C*(d/A),
    therefore solving for Epsilon_r & putting Epsilon
     = Epsilon_0*Epsilon_r
Epsilon_r = (C*d)/(Epsilon_0*A);                    //
     Relative permittivity
printf("\nRelative permittivity = %4d", Epsilon_r);

// Result
// Relative permittivity = 1327
```

Scilab code Exa 3.8 Electric flux and flux density produced in dielectric material

```
// Scilab code Ex3.8: Pg 87 (2008)
clc; clear;
```

```
V = 180;                                // Potential difference, V
d = 3e-03;                              // Thickness of dielectric, m
A = 4.2e-04;                            // Cross-sectional area of plates, m^2
Epsilon_r = 3.5;                        // Relative permittivity
Epsilon_0 = 8.854e-12;                  // Permittivity of Free Space
// Since absolute permittivity, Epsilon = C*(d/A), therefore solving for C & putting Epsilon = Epsilon_0*Epsilon_r
C = ( Epsilon_r*Epsilon_0*A )/d;              // Capacitance, F
// Since C = Q/V, solving for Q
Q = C*V;                                      // Electric charge, C
// Using  D = Q/A,
 D = Q/A;                                     // Electric field density, C/m^2
 printf("\The flux thus produced = %3.2f nC.",Q/1e-09);
 printf("\nThe flux density thus produced. = %3.2f micro-coulomb-per-metre-square", D/1e-06);

// Result
// The flux thus produced = 0.78 nC
// The flux density thus produced. = 1.86 micro-C/m^2
```

Scilab code Exa 3.9 Effective capacitance of capacitors in parallel

```
// Scilab code Ex3.9: Pg 89 (2008)
clc; clear;
```

```
C_1 = 4.7e-06;                    //Capacitance, F
C_2 = 3.9e-06;                    //Capacitance, F
C_3 = 2.2e-06;                    //Capacitance, F
// The resulting capacitance of parallerly connected
    capacitors is the sum of the individual
   capacitance present in the circuit
C = C_1 + C_2 + C_3;              // Resulting
   capacitance  of the circuit, F
printf("\nThe resulting capacitance of the
   combination = %4.1f micro-farad", C/1e-06);

// Result
// The resulting capacitance of the combination =
   10.8 micro-farad
```

Scilab code Exa 3.10 Characteristics of series combination of capacitors

```
// Scilab code Ex3.10: Pg 90-91 (2008)
clc; clear;
C_1 = 6e-06;                     //Capacitance, F
C_2 = 4e-06;                    //Capacitance, F
V = 150;                       // Supply voltage, V
// Part (a)
// The reciprocal of the resulting capacitance of
   capacitors connected in series is the sum of the
   reciprocal of the individual capacitances present
    in the circuit i.e 1/C =  1/C1 + 1/C2, solving
   for C
C = ( C_1*C_2 )/(C_1 + C_2);                     //
   Resulting capacitance, F
// Part (b)
Q = V*C;                                          //
   Electric charge on the capacitors, C
// Part (c)
V_1 = Q/C_1;                                      // P.d
```

```
    across capacitor C_1, V
V_2 = Q/C_2;                                      // P.d
    across capacitor C_2, V
printf("\nThe total capacitance of the combination =
    %3.1f micro-farad", C/1e-06);
printf("\nThe charge on each capacitor = %3d micro-
    coulomb",Q/1e-06);
printf("\nThe p.d. developed across %1d micro-farad
    capacitor = %2d V", C_1/1e-06, V_1);
printf("\nThe p.d. developed across %1d micro-farad
    capacitor = %2d V", C_2/1e-06, V_2);

// Result
// The total capacitance of the combination = 2.4
    micro-farad
// The charge on each capacitor = 360 micro-coulomb
// The p.d. developed across 6 micro-farad capacitor
    = 60 V
// The p.d. developed across 4 micro-farad capacitor
    = 90 V
```

Scilab code Exa 3.11 Potential difference across each capacitor in series combination

```
// Scilab code Ex3.11: Pg 91-92 (2008)
clc; clear;
C_1 = 3e-06;                          //Capacitance, F
C_3 = 12e-06;                         //Capacitance, F
C_2 = 6e-06;                          //Capacitance, F
V = 400;                              // Supply voltage, V
// The reciprocal of the resulting capacitance of
    capacitors connected in series is the sum of the
    reciprocal of the individual capacitances present
     in the circuit i.e 1/C =  1/C1 + 1/C2 + 1/C_3,
    solving for C
```

```
C = (C_1 * C_2 * C_3)/( C_1*C_2 + C_2*C_3 + C_3*C_1);        // Resulting capacitance, F
Q = V*C;                                      // Electric charge on the capacitors, C
// Part (c)
V_1 = Q/C_1;                                  // P.d across capacitor C_1, V
V_2 = Q/C_2;                                  // P.d across capacitor C_2, V
V_3 =Q/C_3;                                   // P.d across capacitor C_2, V
printf("\nP.d across capacitor %1d micro-farad = %5.1f V", C_1/1e-06, V_1);
printf("\nP.d across capacitor %1d micro-farad = %5.1f V", C_2/1e-06, V_2);
printf("\nP.d across capacitor %2d micro-farad = %4.1f V", C_3/1e-06, V_3);

// Result
// P.d across capacitor 3 micro-farad = 228.6 V
// P.d across capacitor 6 micro-farad = 114.3 V
// P.d across capacitor 12 micro-farad = 57.1 V
```

Scilab code Exa 3.12 Charge stored and potential difference across capacitors

```
// Scilab code Ex3.12: Pg 92-95 (2008)
clc; clear;
V = 200;                        // Supply voltage, voltage
C_AB = 2;                  // Capacitance across branch AB, micro-farad
C_BC = 3;                   // Capacitance across branch BC, micro-farad
C_CD = 6;                   // Capacitance across
```

```
branch CD, micro-farad
C_EF = 8;                  // Capacitance across branch EF, micro-farad
C_BD = 4;                   // Capacitance across branch EF, micro-farad

// Part (a)
// Since 3-micro-farad & 6-micro-farad  capacitors are in series & the reciprocal of the resulting capacitance of capacitors connected in series is the sum of the reciprocal of the individual capacitances present in the circuit, therefore i.e 1/C =  1/C1 + 1/C2
C_BCD = ( C_BC*C_CD )/(C_BC+C_CD);          // Resulting capacitance across branch BCD, micro-farad
//Since C_BCD & 4-micro-farad  capacitors are in parallel & the resulting capacitance of parallerly connected capacitors is the sum of the individual capacitance present in the circuit
C_BD = C_BCD + C_BD;          // Resulting capacitance across branch BD, micro-farad
//  Since 2-micro-farad & C_BD  capacitors are in series & the reciprocal of the resulting capacitance of capacitors connected in series is the sum of the reciprocal of the individual capacitances present in the circuit, therefore, we have
C_AD = (C_BD*C_AB)/(C_BD+C_AB);          // Resulting capacitance across branch AD, micro-farad
//Since C_AD & C_EF  capacitors are in parallel & the resulting capacitance of parallerly connected capacitors is the sum of the individual capacitance present in the circuit
C = C_AD + C_EF;          // Resulting capacitance of the circuit, micro-farad
Q = V*C;                                    // Electric  charge drawn from the supply, C
```

```
// Part (b)
Q_EF = V*C_EF;                                    // The charge on the 8 micro-farad capacitor, micro-coulomb

// Part (c)
Q_AD = Q - Q_EF;                                  // The charge on the 4 micro-farad capacitor, C
Q_BD = Q_AD;    // Charge in series combination of capacitors, micro-farad
// Since Q = C*V, solving for V
V_BD = Q_BD/C_BD;                                 // The p.d. across the 4  F capacitor,V

// Part(d)
Q_BCD = V_BD*C_BCD;                               // Electric charge across branch BCD, C
Q_BC = Q_BCD;                                     // Electric charge, C
V_BC = Q_BC/C_BC;                                 // The p.d. across the 3 micro-farad capacitor
printf("\nThe charge drawn from the supply = %3.1f mC", Q/1e+03);
printf("\nThe charge on the %1d micro-farad capacitor = %3.1f mC", C_EF, Q_EF/1e+03);
printf("\nThe p.d. across the %1d micro-farad capacitor= %2d V", C_BD, V_BD);
printf("\nThe p.d. across the %1d micro-farad capacitor = %5.2f V", Q_BC, V_BC);

// Result
// The charge drawn from the supply = 1.9 mC
// The charge on the 8 micro-farad capacitor = 1.6 mC
// The p.d. across the 6 micro-farad capacitor= 50 V
// The p.d. across the 100 micro-farad capacitor = 33.33 V
```

Scilab code Exa 3.13 Capacitance of parallel plate capacitor with mica sheet

```
// Scilab code Ex3.13: Pg 96 (2008)
clc; clear;
N = 20;                 // Number of plates in a capacitor
A = 6400e-06;           // Cross - sectional area of plate, m^2
d = 1.5e-03;            // Distance between plates, m
epsilon_r = 6.4;        // Relative permittivity for mica
epsilon_o = 8.854e-12;     // Relative permittivity for free space
// Calculating the capacitance of the capacitor
C = ((epsilon_o)*(epsilon_r)*A*(N-1))/d;       // Capacitance, F
printf("\n The capacitance of the capacitor = %3.1f nF", C/1e-09);

// Result
// The capacitance of the capacitor = 4.6 nF
```

Scilab code Exa 3.14 Thickness of mica between parallel plates of a capacitor

```
// Scilab code Ex3.14: Pg 96-97 (2008)
clc; clear;
N = 9;                     // Number of plates in a capacitor
```

```
A = 1200e-06;                    // Cross - sectional area of plate, m^2
C = 3e-10;                       // Capacitance, F
epsilon_r = 5;                   // Relative permittivity for mica
epsilon_o = 8.854e-12;           // Relative permittivity for free space
// Using the formula of capacitance, C = ((epsilon_o)*(epsilon_r)*A*(N-1))/d and solving for d, we have
d = ((epsilon_o)*(epsilon_r)*A*(N-1))/C;          // Distance between plates, m
printf("\nThe thickness of mica between parallel plates of a capacitor = %4.2f mm", d/1e-03);

// Result
// The thickness of mica between parallel plates of a capacitor = 1.42 mm
```

Scilab code Exa 3.15 Capacitance of a parallel plate capacitor with air gap

```
// Scilab code Ex3.15: Pg 97 (2008)
clc; clear;
N = 11;                          // Number of plates in a capacitor
r = 25e-03;                      // Radius of circular plate, m
A = (%pi*r^2);                   // Cross - sectional area of plate, m^2
d = 5e-04;                       // Distance between plates, m
epsilon_r = 1;                   // Relative permittivity for air
epsilon_o = 8.854e-12;           // Relative
```

```
permittivity for free space
// Calculating the capacitance of the capacitor
C = ((epsilon_o)*(epsilon_r)*A*(N-1))/d;        // Capacitance, F
printf("\n The capacitance of the capacitor = %3.2f pF", C/1e-10);

// Result
// The capacitance of the capacitor = 3.48 pF
```

Scilab code Exa 3.16 Charging and energy storing ability of capacitor

```
// Scilab code Ex3.16: Pg 99 (2008)
clc; clear;
C_1 = 3e-06;                                        // Capacitance, F
C_2 = 6e-06;                                        // Capacitance, F
V_1 = 250;                                          // Voltage across capacitor C_1, V
// Since each capacitor will take charge according to its capacitance, so we have
Q = C_1*V_1;                                        // Charge on first capacitor C_1, C
W_1 = 0.5*C_1*(V_1^2);                              // Energy stored, J
// When the two capacitors are connected in parallel the 3 micro-farad will share its charge with 6 micro-farad capacitor. Thus the total charge in the system will remain unchanged, but the total capacitance will now be different
C = C_1 + C_2;                                      // Total capacitance, F
// Since Q = C*V, solving for V
V = Q/C;                                            //
```

```
Voltage across capacitor C_2, V
W = 0.5*C*(V^2);                                    // Total energy stored by the combination, J
printf("\nThe charge and energy stored by %1d micro-F capcitor are %3.2f mC and %5.2f mJ respectively", C_1/1e-06, Q/1e-03 , W_1/1e-03);
printf("\nThe p.d. between the plates = %5.2f V", V);
printf("\nThe energy stored by the combination of %1d micro-F and %1d micro-F capacitors = %5.2f mJ", C_1/1e-06, C_2/1e-06, W/1e-03);

// Result
// The charge and energy stored by 3 micro-F capcitor are 0.75 mC and 93.75 mJ respectively
// The p.d. between the plates = 83.33 V
// The energy stored by the combination of 3 micro-F and 6 micro-F capacitors = 31.25 mJ
```

Scilab code Exa 3.17 Charging and discharging capacitors

```
// Scilab code Ex3.17: Pg 99-100 (2008)
clc; clear;
V = 200;                        // Supply voltage, V
C_1 = 10e-06;                   // Capacitance, farad
C_2 = 6.8e-06;                  // Capacitance, farad
C_3 = 4.7e-06;                  // Capacitance, farad
// Part (a)
// Since each capacitor will take charge according to its capacitance, so we have
Q_1 = V*C_1;                    // Charge sored on capacitor C_1, C
W_1 = 0.5*C_1*(V^2);            // Energy sored on capacitor C_1, J
// Part (b)
```

```
// Since C_2 and C_3 are in series and hence, their equivalent capacitance is given by their series combination
C_4 = (C_2 * C_3)/(C_2 + C_3);           // Equivalent capacitance of C_2 and C_3, F
// Since C_1 and C_4 are in parallel and hence, their equivalent capacitance is given by their parallel combination
C = C_1 + C_4;                    // Total capacitance of circuit, F
// Since Q = C*V, solving for V
V_1 = Q_1/C;                              // New p.d across C_1, V
W = 0.5*C*(V_1^2);                       // Total energy remaining in the circuit, J
energy_used = W_1 - W;                   // Energy, J
printf("\nThe charge and energy stored by %2d micro-F capacitor are %1d mC and %2.1f J respectively", C_1/1e-06, Q_1/1e-03, W_1);
printf("\nThe new p.d across %2d micro-F capacitor = %5.1f V", C_1/1e-06, V_1);
printf("\nThe amount of energy used in charging %3.1f micro-F and %3.2f micro-F capacitors from %2d micro-F capacitor = %4.3f J", C_2/1e-06, C_3/1e-06, C_1/1e-06, energy_used/1e-03);

// Result
// The charge and energy stored by 10 micro-F capacitor are 2 mC and 0.2 J respectively
// The new p.d across 10 micro-F capacitor = 156.5 V
// The amount of energy used in charging 6.8 micro-F and 4.70 micro-F capacitors from 10 micro-F capacitor = 43.495 J
```

Scilab code Exa 3.18 Minimum required thickness of dielectric material

```
// Scilab code Ex3.18: Pg 101 (2008)
clc; clear;
V = 400;                    // Supply voltage, V
E = 0.5e06;                 // Dielectric strength, V/m
// Since E = V/d, solving for d
d = V/E;                    // Thickness of dielectric, m
printf("\nThe minimum thickness of dielectric
    required = %3.1fmm", d/1e-03);

// Result
// The minimum thickness of dielectric required =
    0.8 mm
```

Scilab code Exa 3.19 Maximum voltage of capacitor and thickness of dielectric material

```
// Scilab code Ex3.19: Pg 101-102 (2008)
clc; clear;
C = 270e-12;                    // Capacitance, F
A = 60e-04;                     // Cross-sectional area of
    plate, m^2
E = 350e03;                     // Dielectric strength, V/
    m
epsilon_r = 2.1;                 // Relative pemittivity
epsilon_o = 8.854e-12;           // Permittivity of free
    space
// Part (a)
// Since formula for capacitance, C = ((epsilon_o)*(
    eplison_r)*A)/d, solving for d
d = ((epsilon_o)*(epsilon_r)*A)/C;        // Thickness
    of dielectric, m
// Part (b)
// Since E = V/d, solving for V
V = E*d;                                        // Maximum
    possible working voltage, V
```

```
printf("\nThe thickness of Teflon sheet required =
   %5.4f mm", d/1e-03);
printf("\nThe maximum possible working voltage for
   the capacitor = %5.1f V", V);

// Result
// The thickness of Teflon sheet required = 0.413 mm
// The maximum possible working voltage for the
   capacitor = 144.6 V
```

Chapter 4

Magnetic fields and circuits

Scilab code Exa 4.1 Flux density at the pole face

```
// Scilab code Ex4.1: Pg 116 (2008)
clc; clear;
A = 6e-04;                          // Cross-sectional area
   of pole face, metre-square
phi = 30e-06;                       // Flux, Wb
B = phi/A;                          // Flux density, T
printf("\nThe flux desity at the pole face = %2d mT"
   , B/1e-03);

// Result
// The flux desity at the pole face = 50 mT
```

Scilab code Exa 4.2 Magnetic Flux

```
// Scilab code Ex4.2: Pg 116 (2008)
clc; clear;
A = 45e-06;                         // Cross sectional
   area of pole face, metre-square
```

```
B = 0.6;                              // Flux density, T
// Using formula B = phi/A, solving for phi
phi = B*A;                            // Flux, Wb
printf("\nThe flux produced by pole face = %2d micro
   -wWb", phi/1e-06);

// Result
//The flux produced by pole face = 27 micro-Wb
```

Scilab code Exa 4.3 Magnetomotive force and flux density produced in a toroid

```
// Scilab code Ex4.3: Pg 117 (2008)
clc; clear;
N = 1500;                              // Number of turns in
    a coil
A = 5e-04;                             // Cross- sectional
   area of of coil, metre-square
phi = 0.2e-03;                         // Flux, Wb
I = 0.75;                              // Coil-current, A
// Since m.m.f is the product of the current and the
    number of turns, therefore, we have
F = N*I;                              // Magnetomotive force
   , At
B = phi/A;                            // Flux density, T
printf("\The m.m.f and flux density produced are %4d
    At and %3.1f T respectively", F, B);

// Result
// The m.m.f and flux density produced are 1125 At
   and 0.4 T respectively
```

Scilab code Exa 4.4 Excitation current required to produce required magnetomotive force

```
// Scilab code Ex4.4:Pg 117 (2008)
clc; clear;
N = 600;                          // Number of turns in
    a coil
F = 1500;                          // Magnetomotive
    force, At
// Since magnetomotive force,F = N*I, solving for I
I = F/N;                          // Excitation-current,
     A
printf("\nThe excitation current required = %3.1f A"
    , I);

// Result
// The excitation current required = 2.5 A
```

Scilab code Exa 4.5 Magnetic field strength inside a toroid

```
// Scilab code Ex4.5: Pg 118 (2008)
clc; clear;
I = 0.4;                    // Current, A
N = 550;                    // Number of turns in a
    coil
d = 8e-02;                  // Diameter, m
l = (%pi*d);                // Average length of the
    magnetic circuit, m
// Since magnetic field strength is defined as the
    mmf per metre length of the magnetic circuit,
    therefore, we have
H = (N*I)/l;                    // Magnetic field
    strength, At/m
printf("\nThe magnetic field strength inside the
    toroid = %6.2f At/m", H);
```

```

// Result
// The magnetic field strength inside the toroid = 875.35 At/m
```

Scilab code Exa 4.6 Flux and flux density with changed permeability

```
// Scilab code Ex4.6: Pg 119-120 (2008)
clc; clear;
A = 15e-04;                     // Cross-sectional area of core, metre-square
mew_r1 = 65;                    // Relative permeability of core
phi_1 = 2e-04;                  // Flux, Wb
mew_r2 = 800;                   // Changed relative permeability of core
B_1 = phi_1/A;                  // Flux density, T
mew_r = mew_r2/mew_r1;          // Relative permeability of core
// Since cross-sectional area of core A remains constant, therefore, we have mew_r = B_1/B_2 , solving for B_2
B_2 = mew_r*B_1;                //  New flux density, T
// Since B_2 = phi_2/A, solving for phi_2
phi_2 = B_2*A;                  // New flux, Wb
printf("\nThe new flux and flux density are %5.3f mWb and %5.3f T respectively", phi_2/1e-03, B_2);

// Result
// The new flux and flux density are 2.462 mWb and 1.641 T respectively
```

Scilab code Exa 4.7 Magnetic properties of toroid

```
// Scilab code Ex4.7: Pg 120 (2008)
clc; clear;
r = 0.04;                          // Mean radius of torod, m
A = 3e-04;                         // Csa of toroid, m^2
mew_o = 4*(%pi)*1e-07;             // Permeability of free space
mew_r = 150;                       // Relative permeability of toroid
N = 900;                           // Number of turns on coil
I = 1.5;                           // Coil current, A
l = 2*(%pi)*r;                     // Effective length of toroid, m

// Part (a)
// Since m.m.f is the product of the current and the number of turns, therefore, we have
F = N*I;                           // Magnetomotive force, At
printf("\nThe m.m.f of toroid = %4d At", F);

// Part (b)
// Since magnetic field strength is defined as the mmf per metre length of the magnetic circuit, therefore, we have
H = F/l;                              // Magnetic field strength, At/m
printf("\nThe magntic field strength = %6.1f At/m", H);

// Part (c)
B = (mew_r*mew_o*H);                  // Flux density, T
```

```
phi = B*A;                                          // Flux , Wb
printf("\nThe flux and flux density are %6.2f micro-weber and %6.4f T respectively", phi/1e-06, B)

// Result
// The m.m.f of toroid = 1350 At
// The magntic field strength = 5371.5 At/m
// The flux and flux density are 303.75 micro-weber and 1.0125 T respectively
```

Scilab code Exa 4.8 Coil current to produce desired flux

```
// Scilab code Ex4.8: Pg 120-121 (2008)
clc; clear;
r = 3e-02;                                   // Radius of toroid , m
A = 4.5e-04;                                 // Cross-sectional area of toroid , metre-square
N = 500;                                     // Number of turns
phi = 250e-06;                               // Flux , Wb
mew_o = 4*(%pi)*(1e-07);                     // Permeability of free space
mew_r = 300;                                 // Relative permeability
l = 2*(%pi)*r;                               // Effective length , m
B = phi/A;                                   // Flux density , T
// Since B = (mew_r)*(mew_o)*H, solving for H
H = B /((mew_r)*(mew_o));                    // Magnetic field strength , At/m
// Since H = F/l , solving for F
F = H*l;                                     // Magnetomotive force , At
```

```
// Since mmf,F = N*I, solving for I
I = F/N;                                        // Electric
    current, A
printf("\nThe value of current needs to be passed
    through the coil = %4.2f A", I);

// Result
// The value of current needs to be passed through
    the coil = 0.56 A
```

Scilab code Exa 4.9 Charactersittic measurements in a coil

```
// Scilab code Ex4.9: Pg 121-122 (2008)
clc; clear;
// Part (a)
I = 0.2;                       // Electric current, A
l = 5e-02;                     // Effective length, m
A = 7e-04;                     // Cross-sectional area,
     metre-square
d = 0.5e-03;                   // Diametre, m
mew_r = 1;                     //Relative permeability
    for wood
mew_o = 4*(%pi)*1e-07;         // Pemeability for free
    space
N = l/d;                       // Number of turns
// Since mmf is the product of the current and the
    number of turns, therefore, we have
F = N*I;                        // Magnetomotive force,
     At
// Part (b)
// Since magnetic field strength is defined as the
    mmf per metre length of the magnetic circuit,
    therefore, we have
H = F/l;                                    // Magnetic
    field strength, At/m
```

```
B = ( mew_r * mew_o * H );                    //  Flux density, T
// Part (c)
phi = B * A;                                  // Flux, Wb
printf("\nThe mmf produced = %2d At", F);
printf("\nThe flux density produced = %3d micro-tesla", B/1e-06);
printf("\nThe flux produced = %5.3f micro-weber", phi/1e-06);

// Result
// The mmf produced = 20 At
// The flux density produced = 502 micro-tesla
// The flux produced = 0.352 micro-weber
```

Scilab code Exa 4.10 Coil current and relative permeaility

```
// Scilab code Ex4.10: Pg 125 (2008)
clc; clear;
N = 1000;                                     // Number of turns on coil
r = 0.1;                                      // Mean radius of toroid, m
phi = 0.1775e-03;                             // Flux density(value from graph), Wb
A = %pi*1e-04;                                // Csa of toroid, m^2
H = 88;                                       // Magnetic field strength(value from graph), At/m
B = phi/A;                                    // Flux density, T

// Part (a)
l = 2*%pi*r;                                  // Effective length of toroid, m
```

```
// Since H = (N*I)/l, solving for I
I = (H*l)/N ;                                    //
    Electric current in coil, A
printf("\nCoil current = %4.1f mA", I/1e-03);

// Part (b)
mew_o = 4*(%pi)*1e-07;                           //
    Pemeability for free space
// Since B = mew_o * mew_r * H, solving for mew_r
mew_r = B/(mew_o*H);                            //
    Relative permeability of toroid
printf("\nThe relative permeability of toroid = %4d"
    ,mew_r);

// Result
// Coil current = 55.3 mA
// The relative permeability of toroid = 5109
```

Scilab code Exa 4.11 Flux density and relative permeability of toroid

```
// Scilab code Ex4.11: Pg 125-126 (2008)
clc; clear;
mew_o = 4*(%pi)*1e-07;          // Pemeability for
    free space
l = 0.15;                       // Mean length, m
N = 2500;                      // Number of turns
I = 0.3;                       // Electric current,
    A
// Since magnetic field strength is defined as the
    mmf per metre length of the magnetic circuit,
    therefore, we have
H = (N*I)/l;                   // Magnetic field
    strength, At/m
B = 0.75;                      // Flux density(
    value taken from graph), T
```

```
// Since B = ( mew_r * mew_o * H ), solving for mew_r
mew_r = B/(mew_o * H);                    // Relative permeability
printf("\nThe flux desity of given toroid = %3.2f T", B);
printf("\nThe relative permeability of given toroid = %5.1f", mew_r);

// Result
// The flux desity of given toroid = 0.75 T
// The relative permeability of given toroid = 119.4
```

Scilab code Exa 4.12 Currents in differently configured toroids with same flux

```
// Scilab code Ex4.12: Pg 126-127 (2008)
clc; clear;
mew_o = 4*(%pi)*1e-07;                    // Permeability for free space
l = 0.1875;                               // Mean length, m
A = 8e-05;                                // Cross-sectional area of of coil, metre-square
N = 750;                                  // Number of turns
phi = 112e-06;                            // Flux, Wb
l_gap = 0.5e-03;                          // Average length of the magnetic circuit,m
B = phi/A;                                // Flux density, Wb
H = 2000;                                 // Magnetic field strength( value taken from graph ), At/m
F_Fe = H*l;                               // The m.m.f in the iron part of the circuit, At
```

```
// Since F = I*N, solving for I
I = F_Fe/N;                           // Coil current
    under normal conditions , A
// Since B = mew_o * H_gap, solving for H_gap
H_gap = B/mew_o;                      // Magnetic
    field strength , At/m
// Since H_gap = F_gap/l_gap, solving for F_gap
F_gap = H_gap * l_gap;                // The mmf in
    the air part of the circuit , At
F = F_Fe + F_gap;                     // Total circuit
    mmf, At
I_new = F/N;                          // Current
    required to maintain the flux at its original
    value , A
printf("\nThe coil current required to produce a
    flux of %3d micro-weber in the toroid = %3.1f A "
    , phi/1e-06, I);
printf("\nCurrent required to maintain the flux at
    its original value = %5.3f A", I_new);

// Result
// The coil current required to produce a flux of
    112 micro-weber in the toroid = 0.5 A
// Current required to maintain the flux at its
    original value = 1.243 A
```

Scilab code Exa 4.13 Coil current in a magnetic circuit

```
// Scilab code Ex4.13: Pg 127-128 (2008)
clc; clear;
l_A = 0.25;                  // Mean length of
    cicuit A, m
l_B = 0.15;                  // Mean length of
    cicuit A, m
A_A = 11.5e-04;              // Cross-sectional area
```

```
    of circuit A, metre−square
A_B = 12e-04;                     // Cross−sectional area
    of circuit B, metre−square
phi = 1.5e-03;                    // Flux , Wb
N = 1000;                         // Number of turns
B_A = phi/A_A;                   // Flux density linked
   with circuit A, T
B_B = phi/A_B;                   // Flux density linked
   with circuit B, T
H_A = 1470;                      // Magnetic field
   strength of cicuit A( value taken from graph ),
   At/m
H_B = 845;                       // Magnetic field
   strength of cicuit B( value taken from graph ),
   At/m
// Snce H = F/l, solving for F
F_A = H_A * l_A;                      // Magnetic field
   strength of circuit A, At/m
F_B = H_B * l_B;                      // Magnetic field
   strength of circuit B, At/m
F = F_A + F_B;                          // Total circuit m.
   m.f, At/m
I = F/N;                                  // Coil current , A
printf("\Coil current in the magnetic circuit = %5.3
   f A", I);

// Result
//  Coil current in the magnetic circuit = 0.494 A
```

Scilab code Exa 4.14 Magnetomotive force required by ring for generating desired flux

```
// Scilab code Ex4.14: Pg 129−130 (2008)
clc; clear;
A = 8e-04;                                        // Cross−
```

```
    sectional area, metre-square
d = 24e-02;                                    // Mean diameter of iron ring, m
phi = 1.2e-03;                                 // Flux, Wb
mew_r = 1200;                                  // Relative permeability
mew_o = 4*(%pi)*1e-07;                        // Pemeability for free space
mew_air = 1;                                  // Pemeability for air
l_gap = 3e-03;                                // Mean length, m
l_Fe = (%pi) * d;                             // Mean length of iron circuit, m
S_Fe = l_Fe/(mew_r * mew_o *A);               // Reluctance of iron circuit, At/Wb
S_gap = l_gap/(mew_air * mew_o *A);             // Reluctance of gap, At/Wb
S = S_Fe + S_gap;                             // Total circuit reluctance, At/Wb
// Since phi = F/S, solving for F
F = phi*S;                                    // Magnetomotive force, At
printf("\nThe required mmf = %5.1f At", F);

// Result
// The required mmf = 4331 At
```

Scilab code Exa 4.15 Reluctance and current in a circuit placed in magnetic field

```
// Scilab code Ex4.15: Pg 130-131 (2008)
clc; clear;
N = 500;                                              //
```

```
Number of turns on first section's coil
phi = 2e-03;                    // Flux produced by first section, Wb
l_1 = 85e-02;                   // Length of first section, m
l_2 = 65e-02;                   // Length of second section, m
l_3 = 0.1e-02;                  // Length of third section, m
A_1 = 10e-04;                   // Csa of first section, m^2
A_2 = 15e-04;                   // Csa of second section, m^2
A_3 = 12.5e-04;                 // Csa of second section, m^2
mew_o = 4*(%pi)*1e-07;          // Pemeability for free space
mew_r1 = 600;                   // Relative permeability of first section
mew_r2 = 950;                   // Relative permeability of second section
mew_r3 = 1;                     // Relative permeability of third section

// Part (a)
S_1 = l_1/(mew_r1 * mew_o * A_1);       // Reluctance of first section, At/Wb
S_2 = l_2/(mew_r2 * mew_o * A_2);       // Reluctance of first section, At/Wb
S_3 = l_3/(mew_r3 * mew_o * A_3);       // Reluctance of first section, At/Wb
S = S_1 + S_2 + S_3;                    // Total reluctance of the circuit, At/Wb
printf("\nTotal reluctance of the circuit = %4.2fe+06 At/Wb", S*1e-06);

// Part (b)
// Since phi = F/S, solving for F
```

```
F = phi*S;                                          // Magnetomotive force, At
// Since F = N*I, solving for I
I = F/N;                                            // Electric current in first section, A
printf("\nElectric current in first section = %4.2f A", I);

// Result
// Total reluctance of the circuit = 2.13e+06 At/Wb
// Electric current in first section = 8.51 A
```

Chapter 5

Electromagnetism

Scilab code Exa 5.1 Average emf induced into coil

```
// Scilab code Ex5.1: Pg 145 (2008)
clc; clear;
N = 100;                                          // Number
   of turns
delta_phi = 10e-03;                               // Flux
   linked with coil, Wb
delta_t = 2e-03;                                  // Time
   during which flux changes, s
e =((-N)*delta_phi)/delta_t;                     // Average
   induced emf, V
printf("\nThe average emf induced in the coi = %3d V
   ", e);

// Result
// The average emf induced in the coi = -500 V
```

Scilab code Exa 5.2 Changing flux and induced emf in the coil

```
// Scilab code Ex5.2: Pg 146 (2008)
clc; clear;
N = 250;                                       // Number of turns
delta_phi1 = 20e-03;                           // Flux linked with coil, Wb
delta_phi2 = -16e-03;                          // Flux linked with coil, Wb
delta_t1 = 0.05;                               // Time, s
delta_t2 = 0.01;                               // Time, s
e_1 =((-N)*delta_phi1)/delta_t1;               // Average induced emf, V
e_2 =((-N)*delta_phi2)/delta_t2;               // Average induced emf, V
printf("\nChange in flux in first case = %4.2f weber", delta_phi1);
printf("\nEmf induced in first case = %3d volts",e_1);
printf("\nChange in flux in second case = %4.2f weber", delta_phi2);
printf("\nEmf induced in second case = %3d volts",e_2);

// Result
// Change in flux in first case = 0.02 Wb
// Emf induced in first case = -100 V
// Change in flux in second case = -0.02 Wb
// Emf induced in second case = 400 V
```

Scilab code Exa 5.3 Number of turns on coil

```
// Scilab code Ex5.3: Pg 147 (2008)
clc; clear;
```

```
e = 100;                                        // Induced emf, V
// For simplification let (delta_phi)/(delta_t) = k
k = 0.1;                                        // Rate of chage of flux linked with coil, Wb/s
// Since e =((-N)*delta_phi)/delta_t, soling for N
N = (e)/k;                                      // Number of turns
printf("\nThe number of turns on the coil = %4d", N);

// Result
// The number of turns on the coil = 1000
```

Scilab code Exa 5.4 Emf induced in conductor moving in uniform magnetic field

```
// Scilab code Ex5.4: Pg 149 (2008)
clc; clear;
v = 5;                              // Velocity, m^2
theta =(%pi/3);                     // Angle, degrees
phi = 1.6e-03;                      // Flux, Wb
l = 0.1;                            // Length of pole face, m
d = 0.4;                            // Breadth of pole face, m
A = l*d;                            // Cross-sectional area of pole face, m^2
B = phi/ A;                         // Flux density, T
e =( B*l*v)*sin(theta);             // Induced emf, V
printf("\nThe emf induced = %5.4f V", e);
```

```
// Result
// The emf induced = 0.0173 V
```

Scilab code Exa 5.5 Density of magnetic field

```
// Scilab code Ex5.5: Pg 149 (2008)
clc; clear;
l = 0.15;                                   // Effective length of conductor, m
v = 8;                                      // Velocity, m^2
theta = (%pi/180)*55;                       // Angle, degrees
e = 25;                                     // Induced emf, V
// Since e = B*l*v*sin(theta), solving for B
B = e/(l*v*sin(theta));                     // Flux density, T
printf("\nThe density of the field = %5.3f tesla", B);

// Result
// The density of the field = 25.433 T
```

Scilab code Exa 5.6 Emf induced in axle travelling in vertical component of earth magnetic field

```
// Scilab code Ex5.6: Pg 149 (2008)
clc; clear;
l = 2.2;                                    // Effective length of conductor, m
```

```
B =38e-06;                                    // Flux
    density , T
theta = (%pi/2);                              // Angle ,
    degrees
v = 800/36;                                   // Velocity
    , m^2
e = B*l*v*sin(theta);                         // Induced
    emf , V
printf("\The emf induced in the axle = %4.2f mV", e
    /1e-03);

// Result
// The emf induced in the axle = 1.86 mV
```

Scilab code Exa 5.7 Force exerted on current carrying conductor

```
// Scilab code Ex5.7:Pg 152 (2008)
clc; clear;
l = 0.22;                                      //
    Effective length of conductor , m
B = 0.35;                                     // Flux
    density , T
I = 3;                                        // Current ,
    A
theta = (%pi/2);                              // Angle ,
    degrees
// Since the force exerted on the conductor placed
    in magnetic field is directly proportional to the
     flux density , the value of current flowing
    through the conductor , and the length of
    conductor lying inside the field , therefore
F = B*I*l*sin(theta);                         // Force , N
printf("\nThe force exerted on the conductor = %5.3f
    N", F);

```

```
// Result
// The force exerted on the conductor = 0.231 N
```

Scilab code Exa 5.8 Current carrying conductor in magnetic field

```
// Scilab code Ex5.8: Current carrying conductor in magnetic field: Pg 153 (2008)
clc; clear;
phi = 2.5e-03;                                   // Flux, Wb
l = 0.05;                                        // Effective length of pole, m
d = 0.03;                                        // Effective width of pole, m
F = 1.25;                                        // Force exerted on conductor, N
A = l*d;                                         // Cross-sectional area of pole face, m^2
B = phi/A;                                      // Flux density, T
theta = (%pi/2);                                // Angle, degrees
// Since F = B*I*l*sin(theta), solving for I
I = F/(B*l*sin(theta));                         // Current in conductor, A
theta_2 = (%pi/4);                              // New angle, degrees
F_2 = B*I*l*sin(theta_2);                        // Force exerted on conductor, N
printf("\nThe value of the current = %2g A", I);
printf("\nThe force exerted on conductor when placed at 45 degrees to the field = %5.3f newton", F_2);

// Result
```

```
// The value of the current = 14 A
// The force exerted on conductor when placed at 45 degrees to the field = 0.884 N
```

Scilab code Exa 5.9 Torque acting on current carrying conductor placed in magnetic field

```
// Scilab code Ex5.9: Pg 154 (2008)
clc; clear;
l = 0.015;                                              // Length of coil, m
d = 0.006;                                              // Width of  coil, m
B = 1.2;                                                // Flux density, T
I = 1e-02;                                              // Current, a
r = d/2;                                                // Radius of rotation, m
// Since torque is given by the product of force and distance, therefore, we have
T = 2*B*I*l*r;                                          // Torque, Nm
 printf("\nThe torque exerted on the coil = %4.2f micro-Nm", T/1e-06);

// Result
// The torque exerted on the coil = 1.08 micro-Nm
```

Scilab code Exa 5.10 Flux density produced by magnetic pole pieces

```
// Scilab code Ex5.10: Pg 155 (2008)
clc; clear;
```

```
N = 80;                                    // Number of turns
l = 0.02;                                  // Length of coil, m
r = 0.012;                                 // Radius of coil, m
I = 45e-06;                                // Current in coil, A
T = 1.4e-06;                               // Torque exerted on coil, Nm
A = l*r;                                   // Cross-sectional area of coil, m^2
// Since T = 2*B*I*l*r, solving for B
B = T/(2*A*N*I);                           // Flux density, T
printf("\nThe flux density produced by the pole pieces = %4.2f T", B);

// Result
// The flux density produced by the pole pieces = 0.81 T
```

Scilab code Exa 5.11 Force exerted between current carrying parallel conductors

```
// Scilab code Ex5.11: Pg 158 (2008)
clc; clear;
d = 0.035;                    // Distance between two parallel conductors, m
I_1 = 50;                     // Electric current in first coil, A
I_2 = 40;                     // Electric current in second coil, A
F = ((2e-07)*I_1*I_2)/d;      // Force exerted by conductors, N
```

```
printf("\nThe force exerted between the conductors = %4.1f mN", F/1e-03);

// Result
// The force exerted between the conductors = 11.4 mN
```

Scilab code Exa 5.12 Force on a conductor due to current in the other conductor

```
// Scilab code Ex5.12: Pg 158 (2008)
clc; clear;
d = 2;                              // Distance between two parallel conductors, m
I_1 = 1000;                         // Electric current in first coil, A
I_2 = 300;                          // Electric current in second coil, A
mew_o = 4*(%pi)*1e-07;              // Permeability for free space
B = (mew_o*I_1)/d;                  // Flux density due to first coil, T
F = ((2e-07)*I_1*I_2)/d;            // Force exerted by conductors, N
printf("\nThe flux density at a distance of %1d m from the centre of a conductor carrying a current of %4d A = %5.3f mT", d, I_1, B/1e-03);
printf("\nForce exerted by conductors = %2d mN", F/1e-03);

// Result
// The flux density at a distance of 2 m from the centre of a conductor carrying a current of 1000 A = 0.628 mT
// Force exerted by conductors = 30 mN
```

Scilab code Exa 5.13 Shunt resistance to increase the range of ammeter

```
// Scilab code Ex5.13: Pg 163 (2008)
clc; clear;
R_c = 40;                          // Resistance of coil, ohm
I_fsd = 5e-04;                     // Full-scale deflection current, A
I = 3;                             // Current reading, A
V_c = I_fsd*R_c;                   // Potential difference, V
// Since I = I_s + I_fsd, solving for I_s
I_s = I-I_fsd;                      // Shunt current, A
// From Ohm's law, V_c = I_s*R_s, solving for R_s
R_s = V_c/I_s;                      // Shunt resistance, ohm
printf("\nThe value of required shunt resistance = %4.2f milli-ohm", R_s/1e-03);

// Result
// The value of required shunt resistance = 6.67 milli-ohm
```

Scilab code Exa 5.14 Multipler resistance to increase the range of voltmeter

```
// Scilab code Ex5.14: Pg 163-164 (2008)
clc; clear;
R_c = 40;                          // Resistance of coil, ohm
```

```
I_fsd = 5e-04;                              // Full-scale deflection current, A
I_fsd = 5e-04;                              // Full-scale deflection current, A
V = 10;                                     // Voltage reading range, V
V_c = 0.02;                                  // Potential difference across coil resistance, V
// From Ohm's law, V = I_fsd*R, solving for R
R = V/I_fsd;                                 // Total resistance, ohm
// Since R = R_m + R_c, solving R_m
R_m = R - R_c;                               // Multiplier resistance, ohm
printf("\nThe required value of multiplier resistance = %5.2f kilo-ohms", R_m*1e-03);

// Result
// The required value of multiplier resistance = 19.96 kilo-ohms
```

Scilab code Exa 5.15 Shunt and multiplier resistance for a moving coil multimeter

```
// Scilab code Ex5.15: Pg 164-165 (2008)
clc; clear;
R_c = 1500;                                      // Coil resistance, ohm
I_fsd = 75e-06;                                  // Full-scale deflection current, A
I = 5;                                           // Current range, A
V = 10;                                           // Voltage range, V
// Part (a)
```

```
// Using Ohm's law,
V_c = I_fsd*R_c;                                    // Potential difference across coil resistance, V
// Since I = I_s + I_fsd, solving for I_s
I_s = I-I_fsd;                                 // Shunt current, A
// From Ohm's law, V_c = I_s*R_s, solving for R_s
R_s = V_c/I_s;                                 // Shunt resistance, ohm
// Part (b)
// Since = V = V_m + V_c, solving for V_m
V_m = V - V_c;                                  // Potential difference across multiplier resistance, V
// From Ohm's law, V_m = I_fsd*R_m, solving for R_m
R_m = V_m/I_fsd                                // Multiplier resistance, ohm
printf("\nThe required value of shunt resistance = %4.1f mega-ohm", R_s/1e-03);
printf("\nThe required value of multiplier resistance = %4.1f mega-ohm", R_m*1e-03);

// Result
// The required value of shunt resistance = 22.5 mega-ohm
// The required value of multiplier resistance = 131.83 mega-ohm
```

Scilab code Exa 5.16 Potential difference indicated by AVO and percentage error in reading

```
// Scilab code Ex5.16: Pg 166 (2008)
clc; clear;
R_1 = 30;                                  // Resistance, ohm
R_2 = 70;                                  // Resistance,
```

```
ohm
R_in = 200;                                    // Internal resistance of meter, ohm
V = 12;                                        // Supply voltage, V
// Using voltage divider rule, we have
V_2t = (R_2 /(R_1 + R_2))*V                    //  True value of p.d across resistance R_2, V
// Since the rsistances R_2 and R-in are parallel, so their equivalent resistance is given their parallel combination
R_BC = (R_2 * R_in)/(R_2 + R_in);                               // Resistance, ohms
// Using the potential divider technique,
V_2i = (R_BC / ( R_BC + R_1 ))*V                               // Indicated value of p.d across by voltmetre, volts
err = (( V_2i-V_2t ) / V_2t)*100                               // Percentage error in the reading
printf("\nThe p.d. indicated by the meter = %3.1f V", V_2i);
printf("\nThe percentage error in the reading = %4.2f percent", err);


// Result
// The p.d. indicated by the meter = 7.6 V
// The percentage error in the reading = -9.50 percent
```

Scilab code Exa 5.17 Potential difference measured by multimeter and percentage error in reading

```
// Scilab code Ex5.17: Pg 168-169 (2008)
```

```
clc; clear;
R_in = 200;                                          // Internal resistance of meter, kilo-ohms
V = 10;                                              // Supply voltage, volts
R_1 = 10;                                            // Resistance, kilo-ohms
R_2 = 47;                                            // Resistance, kilo-ohms
V_1 = R_1/(R_1+R_2)*V                                // P.d across resistance R_1, V
V_2 = R_2/(R_1+R_2)*V                                // P.d across resistance R_2, V
// Part (a)
R_AB = (R_1 * R_in)/(R_1 + R_in);                    // Resistance, kilo-ohms
V_AB = (R_AB / ( R_AB + R_2 ))*V                     // True value of p.d across by voltmetre, V
R_BC = (R_2 * R_in)/(R_2 + R_in);                    // Resistance, kilo-ohms
V_BC = (R_BC / ( R_BC + R_1 ))*V                     // Indicated value of p.d across by voltmetre, V
// Part (b)
// Error for V_1 measurement
error_AB = (V_AB - V_1)/V_1*100                      // Percentage error in the reading
//Error for V_2 measurement
error_BC = (V_BC-V_2)/V_2*100                        // Percentage error in the reading
printf("\nThe p.d. indicated by the meter across first resistor = %4.2f V", V_AB);
printf("\nThe p.d. indicated by the meter across second resistor = %4.2f V", V_BC);
printf("\nPercentage error for V_1 measurement = %4
```

```
.2f percent", error_AB);
printf("\nPercentage error for V_2 measurement = %4
    .2f percent", error_BC);

// Result
// The p.d. indicated by the meter across  first
    resistor = 1.68 V
// The p.d. indicated by the meter across second
    resistor = 7.92 V
// Percentage error for V_1 measurement = -3.96
    percent
// Percentage error for V_2 measurement = -3.96
    percent
```

Scilab code Exa 5.18 Emf induced in coil due to changing current

```
// Scilab code Ex5.18: Pg 176 (2008)
clc; clear;
L = 0.25;
    // Self-inductance, H
delta_I = 250e-03;
    // Change in current, A
delta_t = 25e-03;
    // Time, s
e = ((-L)*delta_I)/(delta_t);
    // Induced emf, V
printf("\nThe value of emf induced = %3.1f V", e);

// Result
// The value of emf induced = 2.5 V
```

Scilab code Exa 5.19 Inductance of a circuit with changing current

```
// Scilab code Ex5.19: Pg 176 (2008)
clc; clear;
e = 30;                              // Induced emf, V
// For simplicity, let rate of change of current i.e delta_I/delta_t = k
k = 200;                                          // Rate of change of current, ampere-second
// Since e = ((-L)*delta_I)/(delta_t), solving for L
L = e/k;                                          // Self-inductance, H
printf("\nThe inductance of the circuit = %4.2f H", L);

// Result
// The inductance of the circuit = 0.15 H
```

Scilab code Exa 5.20 Required rate of change of current to induce desired emf in a coil

```
// Scilab code Ex5.20: Pg 176 (2008)
clc; clear;
L = 50e-03;                                          // Self-inductance, H
e = 8;                                               // Induced emf, V
// Since e = ((-L)*delta_I)/(delta_t), solving for delta_I/delta_t,and for simplicity letting the rate of change of current i.e delta_I/delta_t = k
k = e/L;                              // Rate of change of current, As
printf("\nThe rate of change of current = %3d A/s",k);

// Result
// The rate of change of current = 160 A/s
```

Scilab code Exa 5.21 Inductance of coil and emf induced in it

```
// Scilab code Ex5.21: Pg 178 (2008)
clc; clear;
N = 150;                            // Number of turns in a coil
I = 10;                             // Electric current flowing through coil, A
phi = 0.10;                         // Flux, Wb
delta_t = 0.1;                      // Time, s
// Part (a)
L  = (N * phi)/I                    // Self-inductance, H
delta_I = 20;                       // Change in current, A
// Part (b)
e = abs((-L*delta_I)/(delta_t));        // Induced emf, V
printf("\nThe inductance of the coi = %3.1f H", L);
printf("\nThe emf induced in the coil = %2d V", e);

// Result
// The inductance of the coi = 1.5 H
// The emf induced in the coil = 300 V
```

Scilab code Exa 5.22 Emf induced in coil due to decreasing current

```
// Scilab code Ex5.22: Pg 178 (2008)
clc; clear;
I_1 = 8;                                    // Electric current, A
```

```
I_2 = 2;                                    // Electric current, A
N = 3000;                                   // Number of turns in a coil
phi_1 = 4e-03;                              // Flux, Wb
delta_t = 0.1;    // Reversal time of current, s
L  = (N * phi_1)/I_1;                       // Self-inductance, H
delta_I = I_1 - I_2;                        // Change in current, A
e = ((L)*delta_I)/(delta_t);          // Induced emf, V
printf("\nThe emf induced in the coil = %2d volts", e);

// Result
// The emf induced in the coil = 90 V
```

Scilab code Exa 5.23 Factors affecting inductance

```
// Scilab code Ex5.23: Pg 179-180 (2008)
clc; clear;
N_1 = 600;                                  // Number of turns in a coil in first case
N_2 = 900;                                  // Number of turns in a coil in secnd case
N_3 = 900;                                  // Number of turns in a coil in third case
l = 45e-03;                                 // Effective length of coil, m
A = 4e-04;                                  // Cross-sectional area of coil, m^2
mew_o = 4*(%pi)*1e-07;                      // Pemeability for free space
```

```
mew_r1 = 1;                              // Relative permeability in first case
mew_r2 = 1;                              // Relative permeability in second case
// Part (a)
mew_r3 = 75;                              // Relative permeability in third case
L_1 = (mew_o*mew_r1*(N_1^2)*A)/l;        // Self-inductance of coil in first case, H
// Part (b)
// Since self-inductance of a coil is directly proportional to the number of turns in a coil, therefore, we have L_2/L_1 = (N_2^2)/(N_1^2), solving for L_2
L_2 = (L_1*(N_2^2))/(N_1^2);          // Self-inductance of coil in second case, H
// Part (c)
// Since mew_r3 = 75*mew_r2, keeping all other quantities same we have
L_3 = mew_r3*L_2;          // Self-inductance of coil in third case, H
printf("\nSelf-inductance of coil in first case = %4.2f mH",L_1/1e-03);
printf("\nSelf-inductance of coil in second case = %5.3f mH", L_2/1e-03);
printf("\nSelf-inductance of coil in third case = %5.3f H", L_3);

// Result
// Self-inductance of coil in first case = 4.02 mH
// Self-inductance of coil in second case = 9.048 mH
// Self-inductance of coil in third case = 0.679 H
```

Scilab code Exa 5.24 elf and mutual inductances of coil

```
// Scilab code Ex5.24: SPg 182 (2008)
clc; clear;
N_A = 2000;                                    // Number of turns in a coil A
N_B = 1500;                                    // Number of turns in a coil B
I_A = 0.5;                                     // Electric current in coil A, A
phi_A = 60e-06;                                // Flux linked with coil A, Wb
// Part (a)
L_A = (N_A*phi_A)/I_A;                         // Self-inductance of coil A
phi_B = 0.83*(60e-06);                        // Flux linked with coil B, Wb
// Part (b)
M = (N_B*phi_B)/I_A;                           // Mutual inductance of the two coils, H
printf("\nSelf-inductance of coil A = %4.2f H", L_A)
printf("\nMutual inductance of the two coils = %5.3f H", M)

// Result
// Self-inductance of coil A = 0.24 H
// Mutual inductance of the two coils = 0.149 H
```

Scilab code Exa 5.25 Self inductance of coil

```
// Scilab code Ex5.25:  Pg 183 (2008)
clc; clear;
N = 400;                                  // Number of turns in a coil
l = 0.25;                                 // Effective length of coil, m
A = 4.5e-04;                              // Cross-
```

```
sectional area, m^2
mew_r = 180;                                    // Relative permeability
mew_o = 4*(%pi)*1e-07;                          // Pemeability for free space
L = (mew_o*mew_r*(N^2)*A)/l             // Self-inductance of coil, H
printf("\nThe self inductance of the coil = %2d milli-henry", L/1e-03);

// Result
// The self inductance of the coil = 65 mH
```

Scilab code Exa 5.26 Mutual inductance of coils and emf induced in them

```
// Scilab code Ex5.26: Pg 183 (2008)
clc; clear;
L_1 = 65e-03;                               // Self-inductance of first coil, H
delta_I = 1.5;                              // Change in current, A
delta_t = 3e-03;                            // Time, s
k = 0.95;                                   // 95 percent of flux produced
N_1 = 400;                                      // Number of turns in a coil A
N_2 = 650;                                      // Number of turns in a coil B
// Part (a)
// Since self-inductance of a coil is directly proportional to the number of turns in a coil, therefore, we have L_2/L_1 = (N_2^2)/(N_1^2), solving for L_2
L_2 = (L_1*(N_2^2))/(N_1^2)             // Self-inductance of second coil , H
```

```
// Part (b)
M = k*sqrt(L_1*L_2);                    // Mutual inductance of two coils, H
// Part (c)
e_1 = ((L_1)*delta_I)/(delta_t);                    // Induced emf in  first coil, V
// Part (d)
e_2 = (M*delta_I)/delta_t;                    // Induced emf in  second coil, V
printf("\nThe self-inductance of coil 2 = %3d mH", L_2/1e-03)
printf("\nThe value of mutual inductance = %3d mH", M/1e-03)
printf("\nThe self-induced emf in coil 1 = %4.1f V", e_1)
printf("\nThe mutually induced emf in coil 2 = %2d V", e_2)

// Result
// The self-inductance of coil 2 = 171 mH
// The value of mutual inductance = 100 mH
// The self-induced emf in coil 1 = 32.5 V
// The mutually induced emf in coil 2 = 50 V
```

Scilab code Exa 5.27 Energy stored in an inductor

```
// Scilab code Ex5.27: Pg 185 (2008)
clc; clear;
L = 50e-03;                    // Self-inductance of coil, H
I = 0.75;                    // Electric current in coil, A
W = (L*(I^2))/2                    // Energy stored, J
printf("\nEnergy stored in the inductor = %4.1f mJ",
```

```
W/1e-03)

// Result
// Energy stored in the inductor = 14.1 mJ
```

Scilab code Exa 5.28 Energy stored in series and parallel combination of inductors

```
// Scilab code Ex5.28: Pg 185-186 (2008)
clc; clear;
L_1 = 25e-03;                                // Self-inductance of first coil, H
L_2 = 40e-03;                                // Self-inductance of  second coil, H
I = 0.25;                                    // Electric current in coils, A
k =0.8;                                      // Coupling coefficient
// Part (a)
W_1 = (L_1*(I^2))/2;                         // Energy stored in first coil, J
W_2 = (L_2*(I^2))/2;                         // Energy stored in second coil, J
M = k*sqrt(L_1*L_2);                         // Mutual inductance of coils
// Part (b)
W_M = M*(I)*(I);                             // Energy stored due to mutual inductance of coils, J
W_sa = W_1 + W_2 + W_M;                      // Energy stored by two inductors when connected in series aiding, J
W_so = W_1 + W_2 - W_M;                      // Energy stored by two inductors when connected in series opposition, J
printf("\nEnergy stored in first coil = %4.2f mJ",
```

```
W_1/1e-03)
printf("\nEnergy stored in second coil = %4.2f mJ", W_2/1e-03)
printf("\nEnergy stored by two inductors when connected in series aiding = %3.1f mJ", W_sa/1e-03)
printf("\nEnergy stored by two inductors when connected in series opposition = %4.2f mJ", W_so/1e-03)

// Result
// Energy stored in first coil = 0.78 mJ
// Energy stored in second coil = 1.25 mJ
// Energy stored by two inductors when connected in series aiding = 3.6 mJ
// Energy stored by two inductors when connected in series opposition = 0.45 mJ
```

Scilab code Exa 5.29 Turns on a coil and turn ratio

```
// Scilab code Ex5.29: Pg 189 (2008)
clc; clear;
V_2 = 60;                              // Output voltage, V
V_1 = 240;                             // Input voltage, V
N_2 = 500;                             // Secondary turns
// Part (a)
// For simplicity let V_1/V_2 = N_1/N_2 = k
k = V_1/V_2                           // Turns ratio
// Part (b)
// Since V_1/V_2 = N_1/N_2, solving for N_1
N_1 = k*N_2;                            // Primary turns
printf("\nThe required turns ratio = %1d:1", k)
printf("\nThe number of primary turns = %4d", N_1)
```

```

// Result
// The required turns ratio = 4:1
// The number of primary turns = 2000
```

Scilab code Exa 5.30 Transformer rating and turn ratio

```
// Scilab code Ex5.28: Pg 189 (2008)
clc; clear;
R_L = 15;                                    // Load
    resistor , ohms
V_2 = 240;                                   // Terminal
    p.d at secondary , V
V_1 = 600;                                   // Supply
    voltage , V
// Part (a)
// Since V_1/V_2 = N_1/N_2 = k
k = V_1/V_2;                                 // Turns
    ratio
// Part (b)
I_2 = V_2/R_L;                               // Current
    drawn by the load , A
P_2 = V_2*I_2;                               // Power
    drawn by the load , W
// Part (c)
I_1 = P_2/V_1                                // Current
    drawn from the supply , A
printf("\nThe transformer turns ratio = %3.1f:1", k)
    ;
printf("\nThe current drawn by the load = %2d A",I_2
    );
printf("\nThe power drawn by the load = %4.2f W",
    P_2*1e-03);
printf("\nThe current drawn from the supply = %3.1f
    A", I_1);
```

```
// Result
// The transformer turns ratio = 2.5:1
// The current drawn by the load = 16 A
// The power drawn by the load = 3.48 W
// The current drawn from the supply = 6.4 A
```

Chapter 6

Alternating Quantities

Scilab code Exa 6.1 Alternating Voltage

```
// Scilab code Ex6.1: Pg 202 (2008)
clc; clear;
// Comparing alternating voltage v = 35*sin(314.2*t)
    with the standard Eq.
// Part (a)
V_m = 35;   // Maximum value of alternating voltage,
   volt

// Part (b)
f = poly(0, "f");         // Declare a variable for
   freq.
f = roots(2*%pi*f - 314.2);      // Frequency of
   waveform, Hz

// Part (c)
T = 1/f;    // Time period of waveform, sec

// Part (d)
t = 3.5;    // Time with reference to zero crossing,
    sec
v = 35*sin(2*%pi*50*3.5*1e-03);    // Volatge value
```

```
after the waveform passes through zero, going
    positive

printf("\nThe maximum value of alternating voltage =
     %2d volt", V_m);
printf("\nThe frequency of alternating voltage = %2d
     Hz", f);
printf("\nThe time period of alternating voltage =
    %3.1f ms", T/1e-03);
printf("\nThe volatge value after the waveform
    passes through zero = %5.2f volt", v);

// Result
// The maximum value of alternating voltage = 35
    volt
// The frequency of alternating voltage = 50 Hz
// The time period of alternating voltage = 20.0 ms
// The volatge value after the waveform passes
    through zero = 31.19 volt
```

Scilab code Exa 6.2 Frequency and time for alternating current

```
// Scilab code Ex6.2: Pg 202 (2008)
clc; clear;
// Part (a)
f = poly(0, "f");          // Declare a variable for
    freq.
// Given i = 75*sin(200*%pi*t) mA which on comparing
     with the general expression gives
f = roots(2*%pi*f - 200*%pi);     // Frequency of
    alternating current, Hz

// Part(b)
i = 35;       // Alternating current after passing
    through zero, mA
```

```
t = asin(i/75)/(200*%pi*1e-03);    // Time taken for current to reach 35 mA, ms

printf("\nThe frequency of alternating current = %2d Hz", f);
printf("\nThe time taken for current to reach 35 mA = %5.3f mA", t);

// Result
// The frequency of alternating current = 100 Hz
// The time taken for current to reach 35 mA = 0.773 mA
```

Scilab code Exa 6.3 Standard expression for ac current from its average value

```
// Scilab code Ex6.3: Pg 204 (2008)
clc; clear;
V_av = 3.5;      // Average value of sinusoidal alternating voltage, V
T = 6.67e-03;    // Time period of alternating current, s
V_m = V_av/0.637;    // Peak value of alternating current, V
f = 1/T;     // Frequency of alternating volatge, Hz
printf("\nThe standard expression for %3.1f voltage = %3.1f sin(%3d*pi*t) volt", V_av, V_m, round(2*f));

// Result
// The standard expression for 3.5 voltage = 5.5 sin(300*pi*t) volt
```

Scilab code Exa 6.4 Instantaneous value of sinusoidal alternating voltage

```
// Scilab code Ex6.4: Pg 204 (2008)
clc; clear;
V_av = 3.5;      // Average value of sinusoidal alternating voltage, V
T = 6.67e-03;    // Time period of alternating voltage, s
V_m = V_av/0.637;    // Peak value of alternating voltage, V
f = 1/T;     // Frequency of alternating volatge, Hz
// Part (a)
t = 0.5e-03;     // Time taken by the waveform after passing through zero, s
v = V_m*sin(2*%pi*f*t);      // Instantaneous value of alternating voltage, s
printf("\nThe instantaneous value of alternating voltage after %3.1f ms = %3.1f volt", t/1e-03, v);
// Part (b)
t = 4.5e-03;     // Time taken by the waveform after passing through zero, s
v = V_m*sin(2*%pi*f*t);      // Instantaneous value of alternating voltage, s
printf("\nThe instantaneous value of alternating voltage after %3.1f ms = %3.1f volt", t/1e-03, v);

// Part (c)
v = 3;      // Alternating voltage after passing through zero, mA
t = asin(v/V_m)/(2*%pi*f);    // Time taken for current to reach 3 V, s
printf("\nThe time taken for voltage to reach %1d volt = %5.3f ms", v, t/1e-03);

// Result
// The instantaneous value of alternating voltage
```

```
after 0.5 ms = 2.5 volt
// The instantaneous value of alternating voltage after 4.5 ms = -4.9 volt
// The time taken for voltage to reach 3 volt = 0.613 ms
```

Scilab code Exa 6.5 Amplitude fo the household supply voltage

```
// Scilab code Ex6.5:  Pg 206 (2008)
clc; clear;
V = 240;    // Rms vlaue of alternating voltage, volt
V_m = sqrt(2)*V;    // Peak value of alternating voltage, volt
printf("\nThe amplitude of household %3d volt supply = %5.1f volt", V, V_m);

// Result
// The amplitude of household 240 volt supply = 339.4 volt
```

Scilab code Exa 6.6 Minimum voltage rating of capacitor

```
// Scilab code Ex6.6: Pg 207 (2008)
clc; clear;
pf = 2.5;    // Peak factor of non-sinusoidal alternating voltage
V = 240;    // Rms vlaue of alternating voltage, volt
V_m = pf*V;    // Peak value of alternating voltage, volt
printf("\nThe absolute minimum working voltage = %3d volt", V_m);
```

```
// Result
// The absolute minimum working voltage = 600 volt
```

Scilab code Exa 6.7 Rectangular coil rotating in uniform magnetic field

```
// Scilab code Ex6.7: Pg 207 (2008)
clc; clear;
l = 0.25;   // Length of the rectangular coil, m
d = 0.2;    // Width of rectangular coil, m
N = 80;     // Number of turns of the rectangular
    coil
B = 0.075;  // Magnetic flux density, tesla
n = 3000/60;   // Frequency of revolution of the
    coil, rev/s
v = n*%pi*d;    // Linear speed with which the coil
    sides move, m/s
t = 2e-03;      // Time after the emf crosses zero,
    s

// Part (a)
// As e = 2*N*B*l*v*sin(2*%pi*f*t) volt, and for
   maximum value of sin(2*%pi*f*t) = 1
E_m = 2*N*B*l*v*(1);  // Amplitude of emf, volt
E = 0.707*E_m;        // rms value of emf, volt
E_av = 0.637*E_m;     // Average value of emf, volt
// For a two pole field system,
f = n;      // Frequency of generated waveform, Hz

// Part (b)
T = 1/f;    // Time period of generated waveform, Hz

// Part (c)
e = E_m*sin(2*%pi*f*t);      // Instantaneous value
    at time 2 ms after zero, volt
```

```

printf("\nThe amplitude, rms and average value of emf = %5.2f V, %5.2f V and %5.2f V resp.", E_m, E, E_av);
printf("\nThe frequency and time period of generated waveform = %2d Hz and %2d ms resp.", f, T/1e-03);
printf("\nThe instantaneous value of emf at time 2 ms after crossing zero = %4.1f V", e);

// Result
// The amplitude, rms and average value of emf = 94.25 V, 66.63 V and 60.04 V resp.
// The frequency and time period of generated waveform = 50 Hz and 20 ms resp.
// The instantaneous value of emf at time 2 ms after crossing zero = 55.4 V
```

Scilab code Exa 6.8 Value of multiplier required for required dc value

```
// Scilab code Ex6.8: Pg 212 (2008)
clc; clear;
R_c = 50;    // Resistance of the coil of meter, ohm
K = 10e+03;      // Figure of merit of the moving coil meter, ohm per volt
V = 10;      // d.c. range of coil meter, volt

// Part (a)
I_fsd = 1/K;     // Full scale deflection for moving coil meter, ampere
R = V/I_fsd;     // Total meter resistance, ohm
// As R = R_m + R_c, solvign for R_m
R_m = R - R_c;       // Multiplier resistance required by the meter, ohm
printf("\nThe multiplier resistance required for 10
```

```
V d.c. range = %5.2f k-ohm", R_m/1e+03);

// Part(b)
I_av = I_fsd;   // Average value of ac current, A
I_rms = %pi/(2*sqrt(2))*I_av;    // rms value of ac current, A
V = 10 ;      // a.c. range of coil meter, volt
R = V/I_rms;    // Total meter resistance, ohm
// As R = R_m + R_c, solvign for R_m
R_m = R - R_c;       // Multiplier resistance required by the meter, ohm
printf("\nThe multiplier resistance required for 10 V a.c. range = %5.2f k-ohm", R_m/1e+03);

// Result
// The multiplier resistance required for 10 V d.c. range = 99.95 k-ohm
// The multiplier resistance required for 10 V a.c. range = 89.98 k-ohm
```

Scilab code Exa 6.9 True rms values with moving coil meter

```
// Scilab code Ex6.9: Pg 213 (2008)
clc; clear;
// Case_I: Square_wave
ff = 1.11;  //  Form factor of calibrated meter
ff_square = 1;  // Form factor for square wave
V_apparent = 5;      // Meter reading for sqaure wave, volt
V_true = V_apparent*1*(ff_square/ff);    // True rms value of square wave voltage, volt
printf("\nThe true rms value of square wave voltage = %5.3f V", V_true);

// Case_II: Triangular_wave
```

```
ff_triangle = 1.15;   // Form factor for triangular
    wave
V_apparent = 5;      // Meter reading for triangular
    wave, volt
V_true = V_apparent*(ff_triangle/ff);    // True rms
    value of triangular wave voltage, volt
printf("\nThe true rms value of triangular wave
    voltage = %4.2f V", V_true);

// Result
// The true rms value of square wave voltage = 4.505
    V
// The true rms value of triangular wave voltage =
    5.18 V
```

Scilab code Exa 6.10 Three alternating currents

```
// Scilab code Ex6.10: Pg 215 (2008)
clc; clear;
// The general expression for alternating current is
    I = Io*sin(2*%pi*f*t + phi)
f = poly(0, 'f');   // Declare the variable for
    frequency
f = roots(2*%pi*f - 80*%pi);    // Frequency of
    alternating current, Hz

// I2 is the reference waveform with zero phase
    angle, so that
phi2 = 0;    // Phase angle for reference waveform I2
    , degrees
Im2 = 3;    // Current amplitude of reference
    waveform I2, A
Im1 = 5;    // Current amplitude of reference
    waveform I1, A
Im3 = 6;    // Current amplitude of reference
```

```
    waveform I3, A
phi1 = %pi/6*(180/%pi); // Phase angle for reference
     waveform I1, degrees
phi3 = %pi/4*(180/%pi); // Phase angle for reference
     waveform I3, degrees

printf("\nThe frequency of all three waveforms = %2d
     Hz", f);
printf("\nI1 leads I2 by = %2.0f degrees", phi1-phi2
    );
printf("\nI3 lags I2 by = %2d degrees", phi3-phi2);
printf("\nCurrent amplitude of reference waveform I1
     = %1d A", Im1);
printf("\nCurrent amplitude of reference waveform I2
     = %1d A", Im2);
printf("\nCurrent amplitude of reference waveform I3
     = %1d A", Im3);

// Result
// The frequency of all three waveforms = 40 Hz
// I1 leads I2 by = 30 degrees
// I3 lags I2 by = 45 degrees
// Current amplitude of reference waveform I1 = 5 A
// Current amplitude of reference waveform I2 = 3 A
// Current amplitude of reference waveform I3 = 6 A
```

Scilab code Exa 6.12 Standard expression for waveforms

```
// Scilab code Ex6.12: Pg 218 (2008)
clc; clear;
Im1 = 7;    // Current amplitude of reference
    waveform I1, A
Im2 = 6;    // Current amplitude of reference
    waveform I2, A
Im3 = 5;    // Current amplitude of reference
```

```
    waveform I3, A
Im4 = 4;    // Current amplitude of reference
    waveform I4, A
phi1 = 70*%pi/180;    // Phase angle for reference
    waveform I1, rad
phi2 = 0*%pi/180;    // Phase angle for reference
    waveform I2, rad
phi3 = -50*%pi/180;    // Phase angle for reference
    waveform I3, rad
phi4 = -90*%pi/180;    // Phase angle for reference
    waveform I4, rad
printf("\ni1 = %dsin(wt + %4.2f) amp", Im1, phi1);
printf("\ni2 = %dsin wt amp", Im2);
printf("\ni3 = %dsin(wt + %4.2f) amp", Im3, phi3);
printf("\ni4 = %dsin(wt + %4.2f) amp", Im4, phi4);

// Result
// i1 = 7sin(wt + 1.22) amp
// i2 = 6sin wt amp
// i3 = 5sin(wt + -0.87) amp
// i4 = 4sin(wt + -1.57) amp
```

Scilab code Exa 6.13 Phasor sum of two voltages

```
// Scilab code Ex6.13: Pg 221 (2008)
clc; clear;
omega = 314;    // Angular frequency of voltage, rad
     per sec
Vm1 = 25;    // Peak value of first phasor, V
Vm2 = 15;    // Peak value of second phasor, V
H_C = Vm1*cosd(%pi/3*180/%pi)+Vm2*cosd(-%pi/6*180/
    %pi); // Horizontal component of phasor sum, V
V_C = Vm1*sind(%pi/3*180/%pi)+Vm2*sind(-%pi/6*180/
    %pi); // Vertical component of phasor sum, V
Vm = sqrt(H_C^2+V_C^2); // Peak value of phasor sum,
```

```
V
phi = atan(V_C/H_C);    // Phase angle, degrees
printf("\nv = %5.2fsin(%3dt + %5.3f) volt", Vm, omega, phi);

// Result
// v = 29.15sin(314t + 0.507) volt
```

Scilab code Exa 6.14 Phasor sum of three currents

```
// Scilab code Ex6.14: Pg 222 (2008)
clc; clear;
Im1 = 6;    // Peak value of first phasor, A
Im2 = 8;    // Peak value of second phasor, A
Im3 = 4;    // Peak value of third phasor, A
H_C = Im1*cosd(0*180/%pi)+Im2*cosd(-%pi/2*180/%pi)+Im3*cosd(%pi/6*180/%pi); // Horizontal component of phasor sum, A
V_C = Im1*sind(0*180/%pi)+Im2*sind(-%pi/2*180/%pi)+Im3*sind(%pi/6*180/%pi); // Vertical component of phasor sum, A
Im = sqrt(H_C^2+V_C^2); // Peak value of phasor sum, V
phi = atan(V_C/H_C);    // Phase angle, rad
printf("\ni = %4.1fsin(wt%5.3f) amp", Im, phi);

// Result
// i = 11.2sin(wt-0.565) amp
```

Scilab code Exa 6.15 Phasor sum of three voltages

```
// Scilab code Ex6.15: Pg 222 (2008)
clc; clear;
```

```

// Part (a)
omega = 628;    // Angular frequency of voltage, rad per sec
f = omega/(2*%pi);  // Frequency of the waveforms, Hz
Vm1 = 10;   // Peak value of first phasor, V
Vm2 = 8;   // Peak value of second phasor, V
Vm3 = 12;   // Peak value of third phasor, V
phi1 = -%pi/6*180/%pi;      // Phase angle for first voltage, degrees
phi2 = %pi/3*180/%pi;      // Phase angle for second voltage, degrees
phi3 = %pi/4*180/%pi;      // Phase angle for third voltage, degrees
printf("\nThe frequency of all three waveforms = %3d Hz", f);
printf("\nThe phase angle and frequency of first voltage : %2d degrees, %2d V", phi1, Vm1);
printf("\nThe phase angle and frequency of second voltage : %2d degrees, %2d V", phi2, Vm2);
printf("\nThe phase angle and frequency of third voltage : %2d degrees, %2d V", phi3, Vm3);

// Part (b)
H_C = Vm1*cosd(phi1)+Vm2*cosd(phi2)+Vm3*cosd(phi3);   // Horizontal component of phasor sum, V
V_C = Vm1*sind(phi1)+Vm2*sind(phi2)+Vm3*sind(phi3);   // Horizontal component of phasor sum, V
Vm = sqrt(H_C^2+V_C^2); // Peak value of phasor sum, V
phi = atan(V_C/H_C);   // Phase angle, rad
printf("\nv = %5.2fsin(%3dt + %5.3f) volt", Vm, omega, phi);

// Result
// The frequency of all three waveforms =  99 Hz
// The phase angle and frequency of first voltage :
```

```
   −29 degrees , 10 V
// The phase angle and frequency of second voltage :
    59 degrees , 8 V
// The phase angle and frequency of third voltage :
   45 degrees , 12 V
// v = 23.57 sin (628 t + 0.458) volt
```

Scilab code Exa 6.16 Dual Beam Oscilloscope

```
// Scilab code Ex6.16: Pg 228 (2008)
clc; clear;

tb1 = 0.1e-03;  // Timebase of channel 1, s/cm
tb2 = 10e-06;  // Timebase of channel 2, s/cm
Y_amp1 = 5; // Y−amp setting for channel 1, V/cm
Y_amp2 = 0.5; // Y−amp setting for channel 2, V/cm

// Channel 1
V_pp = 3*Y_amp1;     // Peak−to−peak value of
   waveform in channel 1, V
Vm = V_pp/2;     // Amplitude of waveform in channel
   1, V
V = Vm/sqrt(2); // rms value of sine wave in channel
    1, V
T = 4*tb1;  // Time period of sine wave, second
f = 1/(T*1000);     // Frequency of sine wave, kHz
printf("\nThe amplitude of sine waveform in channel
   1 = %3.1f V", Vm);
printf("\nThe rms value of sine wave in channel 1 =
   %3.1f V", V);
printf("\nThe frequency of sine wave in channel 1 =
   %3.1f kHz", f);

// Channel 2
V_pp = 2*Y_amp2;     // Peak−to−peak value of
```

```
    waveform in channel 2, V
Vm = V_pp/2;     // Amplitude of waveform in channel
    2, V
V = Vm; // rms value of square wave in channel 2, V
T = 2/3*tb2;  // Time period of square wave, second
f = 1/(T*1000);    // Frequency of square wave, kHz
printf("\nThe amplitude of square waveform in
    channel 2 = %3.1f V", Vm);
printf("\nThe rms value of square wave in channel 2
    = %3.1f V", V);
printf("\nThe frequency of square wave in channel 2
    = %3d kHz", f);


// Result
// The amplitude of sine waveform in channel 1 = 7.5
    V
// The rms value of sine wave in channel 1 = 5.3 V
// The frequency of sine wave in channel 1 = 2.5 kHz
// The amplitude of square waveform in channel 2 =
    0.5 V
// The rms value of square wave in channel 2 = 0.5 V
// The frequency of square wave in channel 2 = 150
    kHz
```

Chapter 7

DC Machines

Scilab code Exa 7.1 The shunt generator

```
// Scilab code Ex7.1: Pg 243 (2008)
clc; clear;
Rf = 200;    // The resistance of field winding of a
     shunt generator, ohm
Po = 80e+03;     // Power delivered by the machine,
    watt
V = 450;    // The terminal voltage, volt
E = 475;    // The generated emf, volt

// Part (a)
// Po = V*I_L, solving for I_L
I_L = Po/V;     // Load current, A
I_f = V/Rf;     // Current through field resistor, A
I_a = I_L + I_f;     // Current through armature
    resistance, A
// As I_a*Ra = E - V, solving for Ra
Ra = 25/I_a;     // Armature resistance, ohm

// Part (b)
Po = 50e+03;     // Output power delivered by the
    machine, watt
```

```
V = 460;    // The terminal voltage, volt
I_L = Po/V;     // Load current, A
I_f = V/Rf;     // Current through field resistor, A
I_a = I_L + I_f;    // Current through armature
    resistance, A
// As I_a*Ra = E - V, solving for E
E = V + I_a*Ra;     // The generated voltage, volt
printf("\nThe armature resistance = %5.3f ohm", Ra);
printf("\nThe value of generated emf = %5.1f V", E);

// Result
// The armature resistance = 0.139 ohm
// The value of generated emf = 475.4 V
```

Chapter 8

DC Transients

Scilab code Exa 8.1 Capacitor charging through a series resistor

```
// Scilab code Ex8.1: Pg 253 (2008)
clc; clear;

C = 8e-06;    // Value of capacitance of capacitor ,
   farad
R = 0.5e+06;    // Value of series resistor , ohm
E = 200;    // Value of d.c. voltage supply , volt

// Part (a)
tau = C*R;    // Time constant of the R-C circuit
   while charging , s
printf("\nThe circuit time constant while charging =
    %1d s", tau);

// Part (b)
I_0 = E/R;    // Initial charging current through
   capacitor , A
printf("\nThe initial charging current through
   capacitor = %3d micro-ampere", I_0/1e-06);

// Part (c)
```

```
t = 4;  // Time after the supply is connected, s
v_C = 0.632*E;  // p.d. across the capacitor 4s after the supply is connected, V
v_R = E - v_C;  // p.d. across the resistor 4s after the supply is connected, V
printf("\nThe p.d. across resistor and capacitor %d s after the supply is connected = %5.1f V and %4.1f V respectively", t, v_C, v_R);

// Result
// The circuit time constant while charging = 4 s
// The initial charging current through capacitor = 400 micro-ampere
// The p.d. across resistor and capacitor 4 s after the supply is connected = 126.4 V and 73.6 V respectively
//
```

Scilab code Exa 8.2 Capacitor discharging through a resistor

```
// Scilab code Ex8.2: Pg 255 (2008)
clc; clear;
C = 0.5e-06;    // Value of capacitance of capacitor, farad
R1 = 220e+03;    // Value of series resistor, ohm
R2 = 110e+03;    // Value of parallel resistor, ohm
E = 150;    // Value of d.c. voltage supply, volt

// Part (a)
tau = C*R1;    // Time constant of the R1-C circuit while charging, s
printf("\nThe circuit time constant while charging = %4.2f s", tau);
I_0 = E/R1;    // Initial charging current through capacitor, A
```

```
printf("\nThe initial charging current through capacitor = %3d micro-ampere", I_0/1e-06);

// Part (b)
tau = C*(R1+R2);    // Time constant of the R1-C-R2 circuit while discharging, s
printf("\nThe circuit time constant while discharging = %4.2f s", tau);
I_0 = E/(R1 + R2);    // Initial discharging current through capacitor, ampere
i = 0.368*I_0;    // Discharge current after one time constant, ampere
V_R2 = i*R2;    // Potential difference across R2 after one time constant, volt
printf("\nThe p.d. across R2 after one time constant while discharging = %4.1f volt", V_R2);


// Result
// The circuit time constant while charging = 0.11 s
// The initial charging current through capacitor = 681 micro-ampere
// The circuit time constant while discharging = 0.16 s
// The p.d. across R2 after one time constant while discharging = 18.4 volt
```

Scilab code Exa 8.3 The series RL circuit

```
// Scilab code Ex8.3: Pg 258 (2008)
clc; clear;
E = 110;    // Value of d.c. voltage supply, volt
L = 1.5;    // Inductor value, henry
R = 220;    // Value of series resistor, ohm

```

```
// Part (a)
di_dt = E/L;    // The initial rate of change of current through inductor, H
printf("\nThe initial rate of change of current through inductor = %5.2f A/s", di_dt);

// Part (b)
I = E/R;    // The final steady current, A
printf("\nThe final steady current through inductor = %3.1f A", I);

// Part (c)
tau = L/R;    // The time taken for the current to reach its final steady value, s
printf("\nThe time taken for the current to reach its final steady value = %4.1f ms", 5*tau/1e-03);

// Result
// The initial rate of change of current through inductor = 73.33 A/s
// The final steady current through inductor = 0.5 A
// The time taken for the current to reach its final steady value = 34.1 ms
```

Chapter 9

Semiconductor Theory and Devices

Scilab code Exa 9.1 The zener diode

```
// Scilab code Ex9.1: Pg 277 (2008)
clc; clear;
// Part (a)
V_Z = 9.1;     // Zener voltage of zener diode, volt
P_Z = 0.5;     // Power rating of zener diode at V_Z,
    W
r_Z = 1.5;     // Slope resistance of zener diode,
    ohm
V = 12;     // Nominal value of input voltage, volt
R_L = 2.5e+03;     // Load resistance across zener
    diode, ohm
I_Z = P_Z/V_Z*1e+03;     // Zener current, mA
I_S = I_Z;     // Current through series resistor, mA
V_S = V - V_Z;     // Voltage drop across series
    resistor, volt
R_S = V_S/I_S*1e+03;     // Value of series
    resistance, ohm
P_max = (I_S*1e-03)^2*R_S;     // Maximum power
    rating of series resistance, W
```

```
printf("\nThe value of series resistance  = %5.2f ohm", R_S);
printf("\nThe value of maximum power rating of series resistance  = %4.2f W", P_max);
printf("\n(a) The suitable value of R_S should be 54 ohm, 0.25 W");

// Part (b)
V_o = V_Z;    // Output voltage across zener, volt
I_L = V_o/R_L*1e+03;    // Load current, mA
I_Z = I_S - I_L;    // Zener current, mA
printf("\n(b) The value of diode current with load resistance across zener = %5.2f mA",I_Z );

// Part (c)
V = 12 - (0.1*12);    // Final value of input voltage after falling below 12 V, volt
R_S = 56;    // Standard value of series resistance, ohm
I_S = (V - V_Z)/R_S*1e+03;    // Current through series resistance, mA
I_Z = I_S - I_L;    // Resulting diode current, mA
delta_I_Z = 51.36 - I_Z;    // Change in zener current, mA
delta_V_Z = delta_I_Z*1e-03*r_Z;    // Change in zener voltage, V
change = delta_V_Z/V_Z*100;    // %age change in zener voltage

printf("\n(c) The percentage change in the p.d. across the load = %4.2f percent",change);

// Result
// The value of series resistance  = 52.78 ohm
// The value of maximum power rating of series resistance  = 159340.66 W
// The suitable value of R_S should be 54 ohm, 0.25 W
```

Scilab code Exa 9.2 Zener diode as a voltage regulator

```
// Scilab code Ex9.2: Pg 279 (2008)
clc; clear;
// Part (a)
Diode = cell(3, 1);     // Declare a diode cell
Diode(1).entries = [1 15 30 0.5 0.007];     // Data for Ist diode
Diode(2).entries = [2 15 15 1.3 0.20];     // Data for 2nd diode
Diode(3).entries = [1 15 2.5 5.0 0.67];     // Data for 3rd diode
Resistor = cell(5, 1)     // Declare a resistor cell
Resistor(1).entries = [0.25, 0.026];     // Data for Ist resistor
Resistor(2).entries = [0.5, 0.038];     // Data for 2 nd resistor
Resistor(3).entries = [1.0, 0.055];     // Data for 3 rd resistor
Resistor(4).entries = [2.5, 0.260];     // Data for 4 th resistor
Resistor(5).entries = [7.5, 0.280];     // Data for 5 th resistor
V = 24;     // Input voltage, volt
V_Z = Diode(1).entries(2);     // Zener voltage for Ist diode, volt
V_S = V - V_Z;     // Voltage drop across series resistor for all the three diodes, volt

// Diode 1
P_Z = Diode(1).entries(4);     // Power rating of Ist diode, W
I_Z = P_Z/V_Z*1e+03;     // Zener current, mA
R_S = V_S/I_Z*1e+03;     // Value of series
```

```
    resistance, ohm
P_S = V_S^2/R_S;    // Power dissipation across series resistor, watt
printf("\nDiode 1:");
printf("\n==========");
printf("\nThe value of series resistance  = %3d ohm", R_S);
printf("\nThe value of power rating of series resistance  = %3.1f W", P_S);
R_S = 270;    // Chosen value of series resistor, ohm
P_S = 0.3;    // Chosen value of power rating, ohm
printf("\nThe suitable value of R_S should be %3d ohm, %3.1f W", R_S, P_S);
printf("\nTotal unit cost = %5.3f pounds\n", Diode(1).entries(5)+Resistor(2).entries(2));

// Diode 2
printf("\nDiode 2:");
printf("\n==========");
P_Z = Diode(2).entries(4);    // Power rating of 2nd diode, W
I_Z = P_Z/V_Z*1e+03;    // Zener current, mA
R_S = V_S/I_Z*1e+03;    // Value of series resistance, ohm
P_S = V_S^2/R_S;    // Power dissipation across series resistor, watt
printf("\nThe value of series resistance  = %5.2f ohm", R_S);
printf("\nThe value of power rating of series resistance  = %4.2f W", P_S);
R_S = 120;    // Chosen value of series resistor, ohm
P_S = 1.0;    // Chosen value of power rating, ohm
printf("\nThe suitable value of R_S should be %3d ohm, %3.1f W", R_S, P_S);
printf("\nTotal unit cost = %4.2f pounds", Diode(2).entries(5)+Resistor(3).entries(2));
```

```

// Diode 3
printf("\nDiode 3:");
printf("\n==========");
P_Z = Diode(3).entries(4);    // Power rating of 3rd diode, W
I_Z = P_Z/V_Z*1e+03;    // Zener current, mA
R_S = V_S/I_Z*1e+03;    // Value of series resistance, ohm
P_S = V_S^2/R_S;    // Power dissipation across series resistor, watt
printf("\nThe value of series resistance  = %3d ohm", R_S);
printf("\nThe value of power rating of series resistance  = %3.1f W", P_S);
R_S = 27;    // Chosen value of series resistor, ohm
P_S = 7.5;    // Chosen value of power rating, ohm
printf("\nThe suitable value of R_S should be %3d ohm, %3.1f W", R_S, P_S);
printf("\nTotal unit cost = %4.2f pounds", Diode(3).entries(5)+Resistor(5).entries(2));

// Part (b)
delta_V_Z = (5*15)/100;    // Allowable change in V_Z, volt
delta_I_Z = 30e-03;    // Allowable change in zener current, A
delta_VZ = zeros(3);
delta_VZ(1) = 30e-03*30;    // Change in zener voltage dor diode 1, V
delta_VZ(2) = 30e-03*15;    // Change in zener voltage dor diode 2, V
delta_VZ(3) = 30e-03*2.5;    // Change in zener voltage dor diode 3, V
printf("\nThe maximum value of zener voltage change = %4.2f V", max(delta_VZ(2), delta_VZ(3)));
printf("\nTo meet the specification at lowest cost, circuit 2 would be adopted");
```

```
// Result
// The value of series resistance  = 52.78 ohm
// The value of maximum power rating of series
   resistance  = 159340.66 W
// The suitable value of R_S should be 54 ohm, 0.25
   W
```