6. CONCLUSION

For a furnace whose operating temperature is above 1000^0C, it is essential to develop radiation heat transfer model to understand the performance of the furnace. Obtaining radiation heat transfer model using zonal method requires DEA values for each and every zone. Such difficult task for rectilinear box shape furnace could be achieved by applying vector algebra and reduction integration scheme to general equations for the complete enclosure and simplify the resulting equations by using Simpson's rule. Resulting computer program can handle rectilinear box shapes with any dimension with any transmission factor. Depend upon the accuracy requirement and the time constraint, the user is provided with the privilege to alter the zone size.

The validation is done by using basic form of DEA (Viewfactor) and comparing the result with mathematically proven equation. Ways of reducing the error percentage is discussed in order to achieve better results is suggested for future work.

Extension of this program to fine tune for garbage inputs from users, optimum amount of zones for reasonable accuracy, reducing time consumption has been suggested for further improvements.

7. REFERENCES

[1] UNEP, "Thermal equipment : Furnaces and refractories," 2006. [Online]. Available: www.energyefficiencies.org. [Accessed 10 6 2014].

[2] P. Mullinger, B. Jenkins, "Industrial and Process Furnaces," in *Industrial and Process Furnaces*, 1st ed., Burlington, Elsevier, 2008, pp. 7 - 20, 112 - 120.

[3] J. H. Leinhard, "A heat transfer textbook," in *A heat transfer textbook*, Massachusetts, Phlogiston, 2008, p. 526.

[4] S. Strommer, M. Niederer, A. Steinboeck, A. Kugi, "A mathematical model of a direct fired continuous strip annealing furnace," *International journal of heat and mass transfer*, pp. 382 - 384, 2013.

[5] M. Modest, "Radiative heat transfer," in *Radiative heat transfer*, 2nd Edition ed., Califonia, Elsevier Science, 2003, pp. 1 - 5, 167 - 172, 642 - 648.

[6] F.P. Incropera, D.P. Dewitt, T.L. Bergman, A.S. Lavine, "Fundamentals of heat and mass transfer," in *Fundamentals of heat and mass transfer*, 6th Edition ed., John Wiley & Sons, 2007, pp. 762 - 825.

[7] H.C. Hottel, R.B. Egbert, *Transactions of American institute of chemical engineers.*, pp. 38, 531 - 565, 1942.

[8] H.C. Hottel, F. Sarofim, Radiative transfer, New York: MacGraw-Hill, 1967.

[9] A.K. Mehrotra, K.Karan, L.A. Behie, "Estimate gas emissivities for equipments and process deisgns," *Chemical Engineering Progress*, pp. 72 - 73, September 1995.

[10] H.C. Hottel, E.S. Cohen, "Radiation heat exchange in a gas filled enclosure: Allowance for non uniformity of gas temperature," *AICHE journal*, vol. 4, pp. 3 - 14, 1958.

[11] H.Erkku, Radiant Heat Exchange in Gas-Filled Slabs and Cylinders, Cambridge: Massachusets Institute of Technology, 1959, pp. 100 - 113.

[12] W. Tian, W.K.S. Chiu, "Calculation of Direct Exchange Area for Nonuniform Zones Using a Reduced Integration Scheme," *Journal of Heat Transfer*, vol.

125, pp. 839 - 842, October 2003.

[13] B. Simmons. [Online]. Available: http://www.mathwords.com/s/simpsons_rule.html. [Accessed 1 April 2015].

[14] *Matlab for Windows,* 2009.

[15] J. Howell. [Online]. Available: http://www.thermalradiation.net/calc/sectionc/C-15.html. [Accessed 5 February 2015].

[16] "Heat transfer radiation," 2004. [Online]. Available: http://yjresources.files.wordpress.com/2009/05/5-2-view-factor-tables-and-graphs.pdf. [Accessed 22 07 2013].

8. APPENDICES

Appendix 1: Emissivity values of different gases at different temperature and pressures

[2]

Table 1. Summary of database for the emissivity of pure gases.

	Number of Data Points	Emissivity (ε)		Temperature *(T)*, K		*pL*, kPa•m	
		Maximum	Minimum	Maximum	Minimum	Maximum	Minimum
CO_2	939	2,788	274	154	0.031	0.23	3.0×10^{-3}
H_2O	1,034	2,776	278	616	0.15	0.66	7.0×10^{-3}
CO	425	1,333	334	62	0.12	0.12	5.0×10^{-3}
CH_4	447	2,088	287	62	0.31	0.24	1.0×10^{-2}
SO_2	394	1,948	396	62	0.062	0.46	3.0×10^{-3}
NH_3	300	1,446	274	62	0.022	0.72	2.0×10^{-3}

Table 2. Summary of database for the emissivity correction factors C_{H_2O}, C_{CO_2}, and C_{SO}.

	Number of Data Points	Pressure *(P)*, kPa		*pL*, kPa•m		Correction Factor	
		Maximum	Minimum	Maximum	Minimum	Maximum	Minimum
C_{H_2O}	151	122	0	309	0	1.7	0.1
C_{CO_2}	190	507	5	77	0	1.7	0.3
C_{SO}	288	N.A.	N.A.	62	3	0.08	0.0

Appendix 2: Sample DEA calculation for bottom to left surface

```
function Output = perpendicular_Bo_L(length,width,height,n1,n2,n3,kT)

% Obtaining the user inputs

count = 0;

stepX = width/n1;
stepY = length/n2;
stepZ = height/n3;

% Defining the step size

start_x = 0;
start_y = 0;
start_z = 0;




for j = 1:n2
for i = 1:n1
pt_y(j).pt_x(i).z_0 = [ start_x, start_y, start_z;start_x, start_y+stepY, start_z;start_x+stepX, start_y+stepY, start_z;start_x+stepX,
start_y, start_z ];


      A = pt_y(j).pt_x(i).z_0; % calculating pointts of surface zones for bottom surface Z = 0.
start_x = start_x+stepX;
```

```
end
start_x = 0;
start_y = start_y+stepY;
end


start_x = 0;
start_y = 0;
start_z = 0;


%
for k = 1:n3
for j = 1:n2
pt_z(k).pt_y(j).x_0 = [ start_x, start_y, start_z;start_x, start_y+stepY, start_z;start_x, start_y+stepY, start_z+stepZ;start_x, start_y,
start_z+stepZ ];



      B = pt_z(k).pt_y(j).x_0; % calculating pointts of surface zones for left surface
start_y = start_y+stepY;

end
start_y = 0;
start_z = start_z+stepZ;
end
```

```
% Following procedure uptosheetnum 1 was used to open an excel file to
% record the values


currentfile = mfilename('fullpath');
 [pathstr,name,ext] = fileparts(currentfile);
 S = fullfile(pathstr,'surface2surface.xls')
addpath(pathstr)
file = S;
 Excel = actxserver('Excel.Application');
 Workbooks = Excel.Workbooks;
Excel.Visible=1;
 Workbook=Workbooks.Open(file)
sheetnum=8;


%
for j = 1:n2

for i = 1:n1
x0 = pt_y(j).pt_x(i).z_0(1,1);
x1 = pt_y(j).pt_x(i).z_0(3,1);
        y0 = pt_y(j).pt_x(i).z_0(1,2);
        y1 = pt_y(j).pt_x(i).z_0(2,2);
        z0 = pt_y(j).pt_x(i).z_0(1,3);
        z1 = pt_y(j).pt_x(i).z_0(3,3);

       A=  [x0 x1, y0 y1];
```

```
for k = 1:n3
for l = 1:n2

          ep0= pt_z(k).pt_y(l).x_0(1,2);
          ep1= pt_z(k).pt_y(l).x_0(2,2);
          ne0= pt_z(k).pt_y(l).x_0(1,3);
          ne1= pt_z(k).pt_y(l).x_0(3,3);
          th0= pt_z(k).pt_y(l).x_0(1,1);
          th1= pt_z(k).pt_y(l).x_0(3,1);

        B=   [ep0 ep1, ne0 ne1];

         [Coordinate, Area_Factor] = f(x0,x1,y0,y1,z0,z1,ep0,ep1,ne0,ne1,th0,th1,stepX,stepY,stepZ,kT)

%  Recording the direct exchange area values that has been obtained


count = count+1;
        rn1   = 12+count;
        rn2   = 12+count;
        range1 = sprintf('B%d:M%d',rn1,rn1);
        range2 = sprintf('O%d:O%d',rn2,rn2);
        Sheets = Excel.ActiveWorkBook.Sheets;
        sheet1 = get(Sheets, 'Item', sheetnum);
invoke(sheet1, 'Activate');
Activesheet = Excel.Activesheet;
ActivesheetRange = get(Activesheet,'Range',range1);
set(ActivesheetRange, 'Value', Coordinate);
        Range = get(Activesheet,'Range',range1);
```

```
out = Range.value;
ActivesheetRange = get(Activesheet,'Range',range2);
set(ActivesheetRange, 'Value',  Area_Factor);
        Range = get(Activesheet,'Range',range2);
out = Range.value;


end

end


end

end

% Saving the opened work book

invoke(Workbook,'Save')
invoke(Excel,'Quit');
delete(Excel);
clearExcel;



function [Coordinate, Area_Factor] = f(x0,x1,y0,y1,z0,z1,ep0,ep1,ne0,ne1,th0,th1,stepX,stepY,stepZ,kT)

symsxyz
```

```
% Lower Limits
xi1 = x0; xj1 = th0 ;
yi1 = y0; yj1 = ep0 ;
zi1 = z0; zj1 = ne0 ;
dxi = stepX; dxj = 0 ;
dyi = stepY; dyj = stepY ;
dzi = 0; dzj = stepZ ;

 F = @(x,y,z)(((stepY+y).*exp(sqrt((xi1+x).^2+(yi1-yj1+y).^2+(zj1+z).^2).*(-kT)).*(xi1+x).*(zj1+z))./((((xi1+x).^2+(yi1-
yj1+y).^2+(zj1+z).^2).^2).*pi));

 G = @(x,y,z)(((stepY-y).*exp(sqrt((xi1+x).^2+(yi1-yj1+y).^2+(zj1+z).^2).*(-kT)).*(xi1+x).*(zj1+z))./((((xi1+x).^2+(yi1-
yj1+y).^2+(zj1+z).^2).^2).*pi));

 Q = triplequad(F,0,dxi,-dyi,0,0,dzj);
 R = triplequad(G,0,dxi,0,dyj,0,dzj);
Coordinate = [x0 x1 y0 y1 z0 z1 th0 th1 ep0 ep1 ne0 ne1];
Area_Factor = Q+R;
```

Appendix3: Sample program which uses Simpson's rule to perform numerical integration

```
% function [Coordinate, Area_Factor] = f(x0,x1,y0,y1,z0,z1,ep0,ep1,ne0,ne1,th0,th1,stepX,stepY,stepZ,kT)
%
%  n=8;
%
%
% i=0;
% j=0;
% k=0;
% l=0;
%
%
%
%
% syms x y z
%
% xi1 = x0; xj1 = th0 ;
% yi1 = y0; yj1 = ep0 ;
% zi1 = z0; zj1 = ne0 ;
% dxi = stepX; dxj = 0 ;
% dyi = stepY; dyj = stepY ;
% dzi = 0; dzj = stepZ ;
%
%
% Base_Function_1 = (stepY+y)*exp(sqrt((xi1+x)^2+(yj1-yi1+y)^2+(zj1+z)^2)*(-kT))*(xi1+x)*(zj1+z)/((((xi1+x)^2+(yi1-
yj1+y)^2+(zj1+z)^2)^2)*pi);
```

```
% Base_Function_2 = (stepY-y)*exp(sqrt((xi1+x)^2+(yj1-yi1+y)^2+(zj1+z)^2)*(-kT))*(xi1+x)*(zj1+z)/((((xi1+x)^2+(yi1-
yj1+y)^2+(zj1+z)^2)^2)*pi);
%
% for j = 0:(dzj)/n:dzj
%         r = r+1;
%         Z(r) = j;
%         z = Z(r);
%         Base1F1(r) = eval(Base_Function_1);
%         Base1F2(r) = eval(Base_Function_2);
%
%     end
%
%      First_Int_1 = (dzj)*(Base1F1(1)+ 4*Base1F1(2)+ 2*Base1F1(3) + 4*Base1F1(4)+ 2*Base1F1(5)+ 4*Base1F1(6)+
2*Base1F1(7)+ 4*Base1F1(8)+Base1F1(9))/(3*n);
%      First_Int_2 = (dzj)*(Base1F2(1)+ 4*Base1F2(2)+ 2*Base1F2(3) + 4*Base1F2(4)+ 2*Base1F2(5)+ 4*Base1F2(6)+
2*Base1F2(7)+ 4*Base1F2(8)+Base1F2(9))/(3*n);
%
%
%      for i = -dyi:(dyi)/n:0
%         s = s+1;
%         Yi(s) = i;
%         y = Yi(s);
%         Base2F1(s) = eval(First_Int_1);
%      end
%
%     Second_Int_1 = (dyi)*(Base2F1(1)+ 4*Base2F1(2) + 2*Base2F1(3) + 4*Base2F1(4)+ 2*Base2F1(5)+ 4*Base2F1(6)+
2*Base2F1(7)+ 4*Base2F1(8)+Base2F1(9))/(3*n);
%
%
```

```
%     for k = 0:(dyj)/n:dyj
%        q = q+1;
%        Yk(q) = k;
%        y = Yk(q);
%        Base2F2(q) = eval(First_Int_2);
%
%     end
%
%     Second_Int_2 = (dyj)*(Base2F2(1)+ 4*Base2F2(2) + 2*Base2F2(3) + 4*Base2F2(4)+ 2*Base2F2(5)+ 4*Base2F2(6)+
2*Base2F2(7)+ 4*Base2F2(8)+Base2F2(9))/(3*n);
%
%
%
%    for l = 0:(dxi)/n:dxi
%        p = p+1;
%        Xi(p) = l;
%        x = Xi(p);
%        Base3F1(p) = eval(Second_Int_1);
%        Base3F2(p) = eval(Second_Int_2);
%    end
%
%     Third_Int_1 = (dxi)*(Base3F1(1)+ 4*Base3F1(2) + 2*Base3F1(3) + 4*Base3F1(4)+ 2*Base3F1(5)+ 4*Base3F1(6)+
2*Base3F1(7)+ 4*Base3F1(8)+Base3F1(9))/(3*n);
%     Third_Int_2 = (dxi)*(Base3F2(1)+ 4*Base3F2(2) + 2*Base3F2(3) + 4*Base3F2(4)+ 2*Base3F2(5)+ 4*Base3F2(6)+
2*Base3F2(7)+ 4*Base3F2(8)+Base3F2(9))/(3*n);


%      Coordinate = [x0 x1 y0 y1 z0 z1 th0 th1 ep0 ep1 ne0 ne1];
%     Area_Factor = Third_Int_1+Third_Int_2;
```

Appendix 4: Sample DEA calculation for bottom to top surface

X0	X1	Y0	Y1	EP0	EP1	NE0	NE1	V.F From other ways	V.F from software calculation	Error %
0	1	0	1	0	1	0	1	0.0686	0.0686	0.00
0	1	0	1	1	2	0	1	0.0481	0.0481	0.00
0	1	0	1	0	1	1	2	0.0481	0.0481	0.00
0	1	0	1	1	2	1	2	0.0351	0.0351	0.00
1	2	0	1	0	1	0	1	0.0481	0.0481	0.00
1	2	0	1	1	2	0	1	0.0686	0.0686	0.00
1	2	0	1	0	1	1	2	0.0351	0.0351	0.00
1	2	0	1	1	2	1	2	0.0481	0.0481	0.00
0	1	1	2	0	1	0	1	0.0481	0.0481	0.00
0	1	1	2	1	2	0	1	0.0351	0.0351	0.00
0	1	1	2	0	1	1	2	0.0686	0.0686	0.00
0	1	1	2	1	2	1	2	0.0481	0.0481	0.00
1	2	1	2	0	1	0	1	0.0351	0.0351	0.00
1	2	1	2	1	2	0	1	0.0481	0.0481	0.00
1	2	1	2	0	1	1	2	0.0481	0.0481	0.00
1	2	1	2	1	2	1	2	0.0686	0.0686	0.00

Appendix 5: Graphical representation of view factor for simple geometries

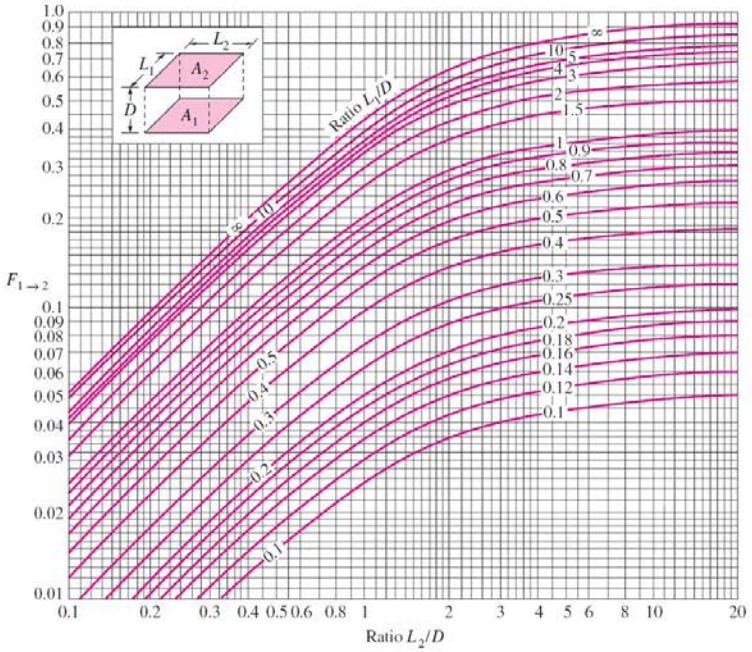

FIGURE 12–5
View factor between two aligned parallel rectangles of equal size.

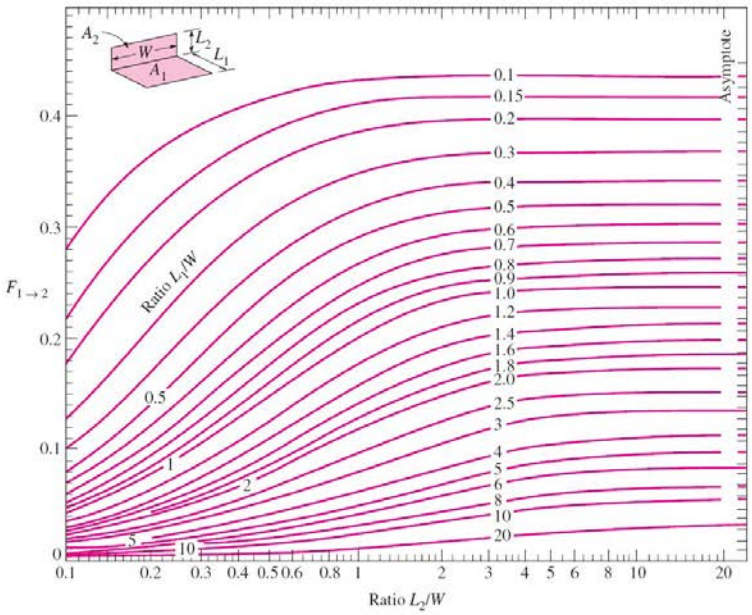

FIGURE 12–6
View factor between two perpendicular rectangles with a common edge.

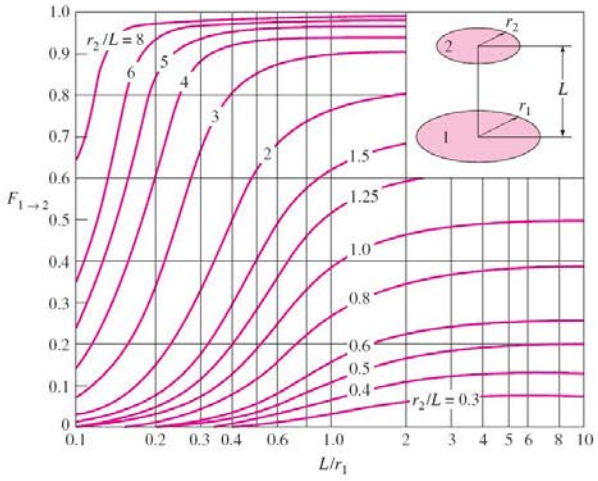

FIGURE 12–7
View factor between two coaxial parallel disks.

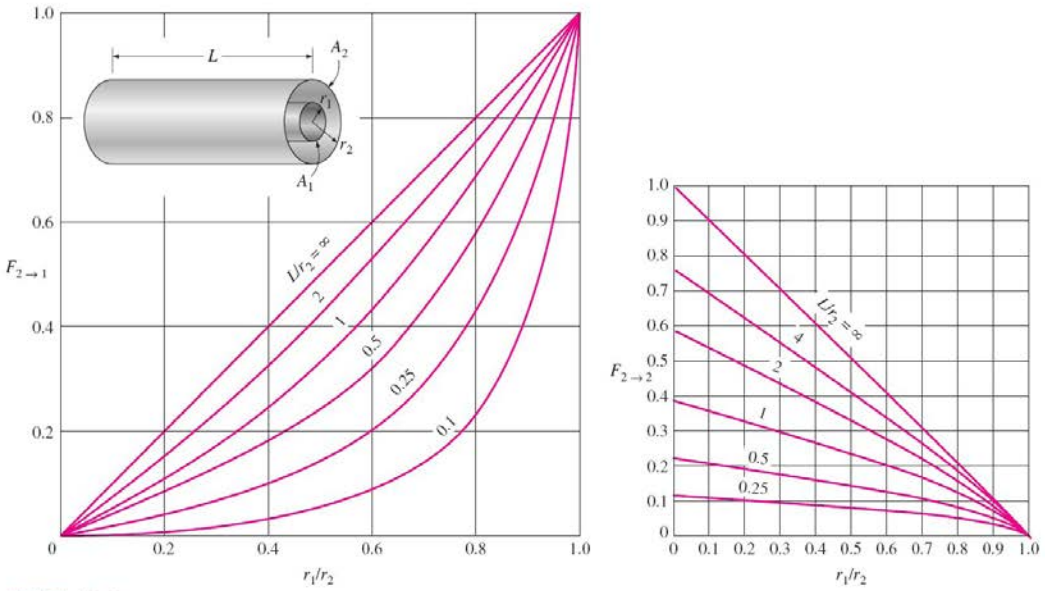

FIGURE 12–8
View factors for two concentric cylinders of finite length: (*a*) outer cylinder to inner cylinder; (*b*) outer cylinder to itself.

Source :[16]