

Simulating The Cooling Of Medical Ct-Scanners: Part 1: FormulationAlireza Vahdati^{1*}, Fakhroddin Dadjoo², Fakhroddin Alimoradi², Ali Falavand³, Mehdi Vahdati¹¹ Mechanical group, Naein Branch, Islamic Azad University, Naein,² Engineering Group, Chaloos Branch, Islamic Azad University, Chaloos, Iran³ Engineering Group, Ahvaz Branch, Islamic Azad University, Ahvaz, IranCorresponding author email: vahdati@naeiniau.ac.ir

Abstract: CT-scanners are used as a non-invasive detecting method in medical applications as a powerful radiological device. Many patients are used CT-scanners for detecting disease. However CT-scanners required cooling due to the huge heat generated during the operation. In this work we modeled a convection-diffusion cooling process of CT-scan devices. A flow in a rectangular body with symmetry, constant heat flux and constant temperature boundary conditions has been considered. The governing equation has been discretized based on the finite volume method and has been solved using a fully implicit method. This paper is mostly concerned with the problem description and the results presents in part 2 of this paper.

[Vahdati A, Dadjoo F, Alimoradi F, Falavand A, Vahdati M. **Simulating the Cooling of Medical CT-Scanners: Part 1: Formulation.** *Life Sci J* 2012;9(1):1029-1034] (ISSN:1097-8135). <http://www.lifesciencesite.com>. 147

Keywords: CT-scanner, CFD simulation, Heat transfer

1. Introduction

Computed tomography (CT) creates two-dimensional images from body structures (S. Hassan and G. Hassan, 2011). Figure 1(a-c) shows the CT imaging system, the gantry and a typical produced image. As seen in figure 1(a), computed tomography imaging system primarily includes the gantry and patient table. The gantry (figure 1(b)) is a moveable frame that contains the x-ray imaging system. The load of heat can lead decreasing the CT-scanner life. When a CT-scanner x-ray tube reaches a maximum heat level, it is necessary to cool it using a cooling fluid.

CT scanners must possess the high amount of heat that a tube can store without any operational damage. The system must absorb high heat levels generated during the high speed rotation of the anode and the bombardment of electrons upon the anode surface (Reddinger, 1997; Ahmadi and Marghmaleki, 2011). The x-ray tubes heat capacity is expressed in heat units. Modern CT scanners bear a heat capacity of 3.5-5 million heat units (Reddinger, 1997). Many CT x-ray tubes utilize a combination of oil and air cooling systems to eliminate heat and maintain continuous operational capabilities (Reddinger, 1997).

In recent decades, CFD, is known as the powerful method that uses numerical methods to simulate problems involving fluid motion by solving the equations (<http://www.ehow.com>; Biao *et al.* 2012). Heat transfer analysis of CT-scanner using Computational Fluid Dynamics is an important process to understand the thermal history of any component or cooling fluid in the CT-scanner. Heat transfer analysis plays a vital role in determining,

monitoring, and controlling thermal loads across a CT-scanner. Heat transfer analysis is used in several industries such as engineering and medical sciences (McKenna *et al.*, 1999; Kuipers *et al.*, 1992; Owoyemi *et al.*, 2007; Behjat *et al.*, 2008; Piechowski, 1999). For example thermal analysis is used in areas where there is consistent interaction between fluid and component surfaces of CT-scanner. Some main applications of heat transfer analysis include cooling systems of CT-scanners, MRI devices and insulation etc. CFD thermal analysis allows an automated, multi-dimensional design optimization system to ensure accurate heat transfer simulation (www.cfdoutsourcingservices.com).

In this work, a Computational Fluid Dynamics simulation is done, to analyze the cooling system dynamic variation of temperature, during passing a cold fluid above a hot plate in a CT-scanner. The parametric characteristics of the problem, together with the domain and applied grid are the items discussed in this paper. The next part of this article is mostly dedicated to the results of this project on cooling CT-scanner surface.

2. Problem formulation

The governing equation is the conservation of energy, in an unsteady-state form as below:

$$\rho C_p \left(\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} \right) = \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) \quad (1)$$

The parameters of the problem in the formula are all in SI units. FVM discretization is used to convert the PDEs to a set of ODEs. As it's a time dependent

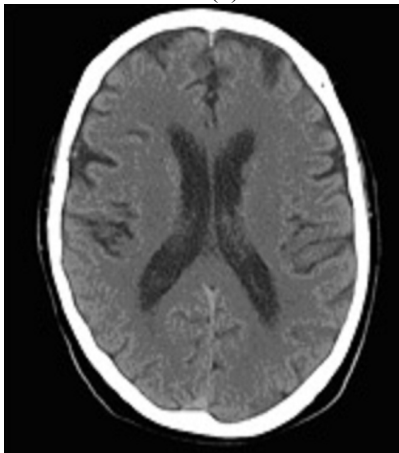
problem, FDM time discretization is also used. The written code is based on implicit formulas. Convergence of the solution is guaranteed, but more memory resource is required rather than explicit method. Table 1 includes specified parameters.



(a)



(b)



(c)

Figure 1. (a) the CT system, (b) the CT gantry and (c) a typical CT image.

Table 1. Specific parameters of the problem

Length	100
Width	50
Initial Temperature	20
Maximum time	500
number of mesh, x direction	30
number of mesh, y direction	15
time step	5
K1	{ $K_x=K_1*(1-\exp(-1000/T))$ }
K2	{ $K_y=K_2*T*(x+y)$ }
Rho	500
Cp	10
T down	200
T left	120
H	200
T	300
Dp/dx	0.05
Heat generation rate	1

3. Geometry of the model

The geometry and boundary conditions applied for the problem of cooling CT-scanner anode surface is shown schematically in Figure 2 and 3. The values of L, W, T up, T left, T_∞ , h can be considered as input data.

Regarding the initial conditions of this problem, the following figure shows the state of the fluid at time 0. As shown, the fluid body of the fluid is at 20°C before we start to pass it over the hot plate.

4. Grid

A simple rectangular grid is used to simulate the problem. Figure 4. includes a schematic picture of the grid used in this study.

5. The in-house coding

The formulation and boundary conditions described above are all programmed in MATLAB 2006a. The following section of this paper includes the code we used. This code is based on the finite volume discretization and the schemes required for consistency, convergence and stability of the solution.

6. Discussion

CFD has many benefits to companies and to researchers. Some argue that using a computer to simulate cooling flow in CT-scanners using CFD is cheaper and easier than building the required machinery, tooling, and equipment to manufacture a test part of CT-scanner and then test it physically. CFD also produces accurate results and data for every point of time and thus can be used to completely track even a single particle of a fluid from point A to point B.

One of the major issues of relying on CFD-produced data is verification and validation. Before a new CFD code or results can be trusted, it must first be verified that the codes are calculating the right equations, and then the results must be compared with physical, real-world tests in order to validate the accuracy of the calculations. Because of this, CFD will not be used as a sole source of data for the time being. Another major hurdle of using CFD is the limitation due to computing power. Supercomputers are getting more and more powerful every year, but real complex CFD problems, such as problems with moving boundaries and many different flows, require

computing power that, while it is possible today, is not fast enough to solve in due time.

Current research topics that use CFD is the thermal analysis of heat flux and temperature distribution through the heat transfer along a cold fluid, passing over a hot plate. This is the case that occurs during cooling CT-Scanner operation. The code we used is presented in this article. A comprehensive assessment on the results and their feedback on the criteria we used to solve this problem will be in the follow up paper.

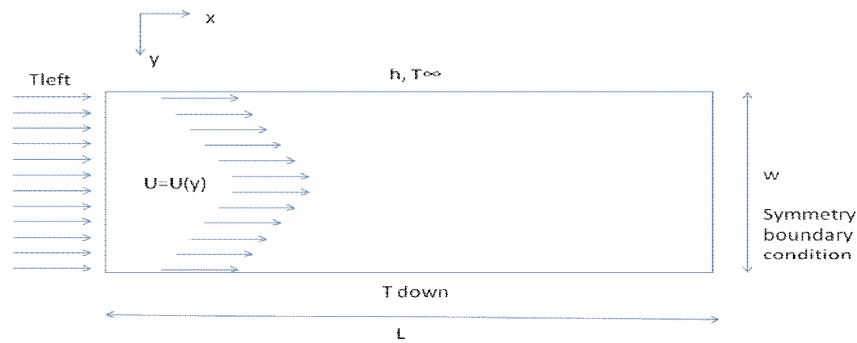


Figure 2. Boundary conditions

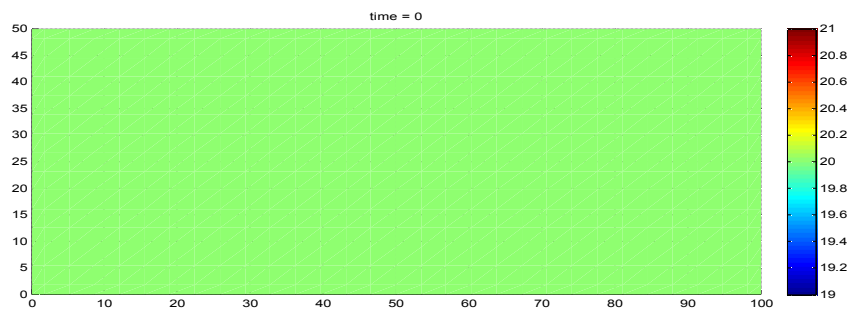


Figure 3. Domain at time=0

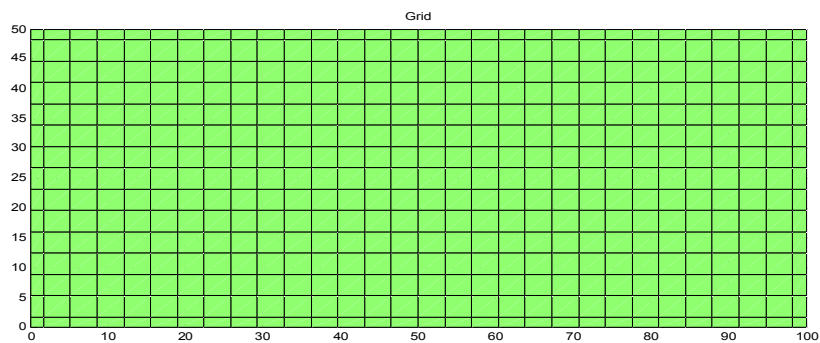


Figure 4. A schematic picture of the grid

7. Conclusions

While CT-scanners required cooling due to the huge heat generated during the operation, less work has been conducted on modeling a convection-diffusion cooling process of CT-scan devices. The cooling flow in a medical CT-scanner with constant heat flux and constant temperature boundary conditions has been considered in this work. Velocity profile comes from a one dimensional flow between parallel planes of CT-scanner. The governing equation has been discretized based on the finite volume method and has been solved using a fully implicit method. This paper is mostly concerned with the CT-scanner problem description and the results presents in part 2 of this paper. The computer code has been presented in appendix.

Corresponding Author:

Alireza Vahdati

Naein Branch, Islamic Azad University, Naein, Iran

E-mail: vahdati@naeiniau.ac.ir

References

- Ahmadi M, Marghmaleki I S. Numerical simulation of turbulent flow in channels with three-dimensional blocks; Life Science Journal, 2011; 8(4): 511-516.
- Behjat Y, Shahhosseini S, Hashemabadi S H. CFD modeling of hydrodynamic and heat transfer in fluidized bed reactor; Int. Communications Heat Mass Transfer 2008; 35 (3):357-368.
- Biao H, Wei W , Xixu W, Jue W, Meng Y. Numerical Simulation to Get Flow Pattern in Modified Carotid Artery Bifurcation Model Using PIV; Life Science Journal, 2012; 9(3): 1296-1301.
- Hassan M S, Hassan M G. Role of Multislice CT in Assessment of Carotid Stenosis; Life Science Journal, 2011; 8(4): 753-756
- <http://www.cfdoutsourcingservices.com>
- <http://www.ehow.com>
- Kuipers J A M, Prins W, Van Swaaij W P M. Numerical calculation of wall-to-bed heat transfer coefficients in gas-fluidized beds; AIChE Journal 1992, 38 (7):1079-1091.
- McKenna T F, Spitz R, Cokljat D. Heat transfer from catalysts with computational fluid dynamics; AIChE Journal 1999; 45 (11):2392-2410.
- Owoyemi O, Mazzei L, Lettieri P. CFD modeling of binary-fluidized suspensions and investigation of role of particle-particle drag on mixing and segregation; AIChE Journal 2007; 53 (8):1924-1940.
- Piechowski M. Heat and mass transfer model of a Ground Heat Exchanger: theoretical development; International Journal of Energy Research 1999; 23 (7):571-588.
- Reddinger W L. CT Instrumentation & Physics, OutSource Inc., 1997.

Appendix: MATLAB Code

```

clc
clear
n=1;
while n==1
    prompt = {' LENGTH :','WIDTH:','initial
temperature','max time','number of mesh, x
direction','number of mesh, y direction','time step'};
    dlg_title = 'Input';
    num_lines = 1;
    def = {'100','50','20','500','30','15','5'};
    answer = inputdlg(prompt,dlg_title,num_lines,def);
    if size(answer)==0
        button = questdlg('Would you like to close this
simulation program?',...
'simulation','yes','no','no');
        if strcmp(button,'yes')
            return;
        end;
    else
        n=0;
    end;
end;
L=str2num(cell2mat(answer(1)));
W=str2num(cell2mat(answer(2)));
T0=str2num(cell2mat(answer(3)));
t=str2num(cell2mat(answer(4)));
delta_x=L/(str2num(cell2mat(answer(5)))-1);
delta_y=W/(str2num(cell2mat(answer(6)))-1);
delta_t=str2num(cell2mat(answer(7)));
n=1;
while n==1
    prompt = {'K1 (Kx=K1*(1-exp(-1000/T))','K2
Ky=K2*T*(x+y))','Rho','Cp','T down','T left','h','T
inf','dp/dx','Heat generation'};
    dlg_title = 'Input';
    num_lines = 1;
    def =
{'600','1','500','10','200','120','200','300','0.05','1'};
    answer = inputdlg(prompt,dlg_title,num_lines,def);
    if size(answer)==0
        button = questdlg('Would you like to close this
simulation program?',...
'simulation','yes','no','no');
        if strcmp(button,'yes')
            return;
        end;
    else
        n=0;
    end;
end;
K1=str2num(cell2mat(answer(1)));
K2=str2num(cell2mat(answer(2)));

```

```

Rho=str2num(cell2mat(answer(3)));
Cp=str2num(cell2mat(answer(4)));
Tdown=str2num(cell2mat(answer(5)));
Tleft=str2num(cell2mat(answer(6)));
h=str2num(cell2mat(answer(7)));
Tinf=str2num(cell2mat(answer(8)));
dpdx=str2num(cell2mat(answer(9)));
Q=str2num(cell2mat(answer(10)));
x=[0,delta_x/2:delta_x:L-delta_x/2,L];
y=[0,delta_y/2:delta_y:W-delta_y/2,W];
[n,s1]=size(x);
[n,s2]=size(y);
[x,y]=meshgrid(x,y); %node locate
tt=zeros(s2,s1);
T0=T0+tt;
axes1 = axes(...
'CameraPosition',[2.54 41.94 916],...
'CameraUpVector',[1 0 0],...
'DataAspectRatio',[50 50 1]);
axis(axes1,[0 L 0 W]);
grid(axes1,'on');
hold(axes1,'all');
surf1 = surf(x,y,tt,'Parent',axes1);
pause(1)
delete(surf1);
delete(axes1);
axes1 = axes(...
'CameraPosition',[3*L/4 W/2 1000],...
'CameraUpVector',[1 0 0],...
'XGrid','off',...
'YGrid','off',...
'ZGrid','off',...
'DataAspectRatio',[50 50 1]);
axis(axes1,[0 L 0 W]);
grid(axes1,'on');
hold(axes1,'all');
surf1
surf(x,y,T0,'Parent',axes1,'FaceColor','interp',...
'EdgeColor','none','FaceLighting','phong');
colorbar1 = colorbar('EastOutside',...
'XLim',[-0.5 1.5]);
pause(1)
e=[];
for m=[1:s1-2]
for n=[2:s2-1]
% P, W, E, N, S
d=[s2*m+n,s2*(m-
1)+n,s2*(m+1)+n,s2*m+n+1,s2*m+n-1];
e=[e;d];
end
end
TT=T0;
T0=zeros(s1*s2,s1*s2);
for i=1:s1
for j=1:s2
T0(s2*(i-1)+j,s2*(i-1)+j)=TT(j,i);
end
end
KX=inline('K*(1-exp(-1000/T))','K','T'); % K1
KY=inline('K*T*(x+y)','K','T','x','y'); % K2
U=zeros(s2,s1);
for i=1:s2
for j=1:s1
Kx(i,j)=KX(K1,TT(i,j)); % Kx(i,j)=K1;
Ky(i,j)=KY(K2,TT(i,j),x(i,j),y(i,j)); %
Ky(i,j)=K2;
yy=y(i,j)-y(1,j);
U(i,j)=dpdx*yy/2*(W-yy);
end
end
for ti=delta_t:delta_t:t
A=zeros(s1*s2,s1*s2); B=zeros(s1*s2,1);
j=0;
for m=1:s1
for n=1:s2
i=(m-1)*s2+n;
a=zeros(1,5); b=zeros(1); % a=[P W E N S]
if x(n,m)==0
a(1,1)=1; b=Tleft; p=0;
else if x(n,m)==L
a(1,1)=1; a(1,2)=-1; b=0; p=0;
else if y(n,m)==0
a(1,1)=1; b=Tdown; p=0;
else if y(n,m)==W
Dy=y(n,m)-y(n-1,m);
Kyn=(Ky(n-1,m)+Ky(n,m))/2;
a(1,1)=-Kyn/Dy-h;
a(1,5)=Kyn/Dy; b=-h*Tinf; p=0;
else
j=j+1;
Dxw=x(e(j,1))-x(e(j,2));
Dxe=x(e(j,3))-x(e(j,1));
Dyn=y(e(j,1))-y(e(j,4));
Dys=y(e(j,5))-y(e(j,1));
Dx=(Dxw+Dxe)/2; Dy=(Dyn+Dys)/2;
if m>2
Kxe=6/8*Kx(n,m)+3/8*Kx(n,m+1)-
1/8*Kx(n,m-1);
Kxw=6/8*Kx(n,m-1)+3/8*Kx(n,m)-
1/8*Kx(n,m-2);
else
Kxe=(Kx(n,m)+Kx(n,m+1))/2;
Kxw=(Kx(n,m)+Kx(n,m-1))/2;
end
Kxn=(Kx(n-1,m)+Kx(n,m))/2;
Kxs=(Kx(n+1,m)+Kx(n,m))/2;
if m>2
Kye=6/8*Ky(n,m)+3/8*Ky(n,m+1)-
1/8*Ky(n,m-1); %
Kyw=6/8*Ky(n,m-1)+3/8*Ky(n,m)-
1/8*Ky(n,m-2); %
else

```

```

    Kye=(Ky(n,m)+Ky(n,m+1))/2;
    Kyw=(Ky(n,m)+Ky(n,m-1))/2;
    end
    Kyn=(Ky(n-1,m)+Ky(n,m))/2;
    Kys=(Ky(n+1,m)+Ky(n,m))/2;
    Ue=(U(n,m)+U(n,m+1))/2;
    Uw=(U(n,m)+U(n,m-1))/2;
    Un=(U(n-1,m)+U(n,m))/2;
    Us=(U(n+1,m)+U(n,m))/2;
    a(1,3)=a(1,3)+Kxe*Dy/Dxe;
    a(1,1)=a(1,1)-Kxe*Dy/Dxe;
    a(1,1)=a(1,1)-Kxw*Dy/Dxw;
    a(1,2)=a(1,2)+Kxw*Dy/Dxw;
    a(1,4)=a(1,4)+Kyn*Dx/Dyn;
    a(1,1)=a(1,1)-Kyn*Dx/Dyn;
    a(1,1)=a(1,1)-Kys*Dx/Dys;
    a(1,5)=a(1,5)+Kys*Dx/Dys;
    a(1,3)=a(1,3)+Rho*Cp*U(n,m)/2/Dx;
    a(1,2)=a(1,2)-Rho*Cp*U(n,m)/2/Dx;
    b=Q*Dx*Dy; p=1;
    end
    end
    end
    if p==0
    A(i,i)=a(1,1); B(i)=b;
    if m==s1
        A(i,i-s2)=a(1,2);
    end
    if n==s2 & m>1 & m<s1
        A(i,i-1)=a(1,5);
    end
    else
    A(i,i)=(Rho*Cp*Dx*Dy/delta_t)-a(1,1);
    A(i,e(j,2))=-a(1,2); A(i,e(j,3))=-a(1,3);
    A(i,e(j,4))=-a(1,4); A(i,e(j,5))=-a(1,5);
    B(i)=b+(Rho*Cp*Dx*Dy/delta_t)*T0(i,i);
    end
    end
    end
    T=zeros(s2,s1);
    X=inv(A)*B;
    for i=1:s1
        for j=1:s2
            T(j,i)=X(s2*(i-1)+j);
        end
    end
    end
    T0=T;
    TT=T0;
    T0=zeros(s1*s2,s1*s2);
    for i=1:s1
        for j=1:s2
            T0(s2*(i-1)+j,s2*(i-1)+j)=TT(j,i);
        end
    end
    delete(surf1);
    delete(axes1);
    ttt=num2str(t);
    axes1 = axes(...
        'CameraPosition',[3*L/4 W/2 1000],...
        'CameraUpVector',[1 0 0],...
        'XGrid','off',...
        'YGrid','off',...
        'ZGrid','off',...
        'DataAspectRatio',[50 50 1]);
    axis(axes1,[0 L 0 W]);
    grid(axes1,'on');
    hold(axes1,'all');
    surf1 =
    surf(x,y,T,'Parent',axes1,'FaceColor','interp',...
        'EdgeColor','none','FaceLighting','phong');
    colorbar1 = colorbar('peer',...
    axes1,'EastOutside',...
    'Box','on',...
    'XLim',[-0.5 1.5],...
    'YLim',[20,200]);
    pause(1)
    T
    ti
    for i=1:s2
        for j=1:s1
            Kx(i,j)=KX(K1,T(i,j)); % Kx(i,j)=K1;
            Ky(i,j)=KY(K2,T(i,j),x(i,j),y(i,j)); %
            Ky(i,j)=K2;
            yy=y(i,j)-y(1,j);
            U(i,j)=dpdx*yy/2*(W-yy);
        end
    end
    end
    end

```

2/12/2012