Temperature Distribution Produced in a Thin Rectangular Plate Heated by a Laser Beam

Paul Hudlow

January 24, 2012
Contents

1 Introduction 2

2 Methodology 3
   2.1 Problem Model ........................................... 3
   2.2 Solution of the Differential Equation ...................... 3
   2.3 Grid Size Considerations .................................. 4

3 Results and Discussion 5
   3.1 Solution Form ............................................. 5
   3.2 Solution Accuracy .......................................... 5
   3.3 Impact of Radiation ......................................... 6

4 Conclusions 8

5 Code Listing 10
Chapter 1

Introduction

With continuing advances in laser technology and an expanding range of applications for it there comes the necessity to better understand how lasers interact with objects and materials, particularly in the area of heat transfer. This report explores some characteristics of laser-induced heat transfer by examining particular case which is relevent to many applications.

The problem posed is that of a thin rectangular plate being heating by a narrow laser beam. For the purposes of analysis the beam is placed in the center of the plate and fixed temperature boundary conditions are used on the edges. The steady state temperature distribution is found using an iterative finite difference method.

This report will begin by discussing the methodology used to model the problem and solve the relevent differential equation. The next chapter will present the results of the analysis and explain the prominent features of the temperature distributions produced. Conclusions will then be made about the results as well as the methods chosen for solving the problem. Finally the references and a code listing will be given to facilitate detailed review.
Chapter 2

Methodology

2.1 Problem Model

In order to properly model the heat transfer across the plate the heat equation is used with the addition of a heat source term (from the laser) and a radiation term.

\[ \nabla \cdot (k \nabla T) + \dot{q} - \varepsilon \sigma (T^4 - T_{amb}^4) = \rho C \frac{\partial T}{\partial t} \]

Since the steady state solution is sought the time derivative term may be set to zero.

\[ \nabla \cdot (k \nabla T) + \dot{q} - \varepsilon \sigma (T^4 - T_{amb}^4) = 0 \]

The differential term is expanding recognizing that the conductivity, \( k \) is a function of temperature.

\[
\frac{\partial k}{\partial x} \frac{\partial T}{\partial x} + k \frac{\partial^2 T}{\partial x^2} + \frac{\partial k}{\partial y} \frac{\partial T}{\partial y} + k \frac{\partial^2 T}{\partial y^2} + \dot{q} - \varepsilon \sigma (T^4 - T_{amb}^4) = 0
\]

At this point it is necessary to specify the relation between the conductivity, \( k \) and the temperature. For the purpose of this analysis a linear relationship is assumed.

\[ k = AT + B \]

\[ \Rightarrow A \left( \frac{\partial T}{\partial x} \right)^2 + k \frac{\partial^2 T}{\partial x^2} + A \left( \frac{\partial T}{\partial y} \right)^2 + k \frac{\partial^2 T}{\partial y^2} + \dot{q} - \varepsilon \sigma (T^4 - T_{amb}^4) = 0 \]

The laser beam is assumed to be Gaussian and will be modeled by the following equation.

\[ \dot{q} = \dot{q}_0 e^{-2r^2/w^2} \]

Where \( \dot{q}_0 \) is the laser intensity in the center of the beam, \( r \) is the radial distance from the beam’s center and \( w \) is the nominal radius of the beam.

2.2 Solution of the Differential Equation

A second-order central differencing scheme will be used to discretize the spacial derivatives. The scheme is shown below.

\[
\frac{\partial T}{\partial x} = \frac{T_{i+1,j} - T_{i-1,j}}{2\Delta x}, \quad \frac{\partial T}{\partial y} = \frac{T_{i,j+1} - T_{i,j-1}}{2\Delta y}
\]
\[
\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2}, \quad \frac{\partial^2 T}{\partial y^2} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta y^2}
\]

These differencing formulas are substituted into the differential equation and the resulting algebraic system of equations are solved with successive substitution. The following manipulation of the equation is used.

\[
T_{i,j} = \left[ C_{1,x} + C_{2,x} + C_{1,y} + C_{2,y} + \dot{q} - \varepsilon\sigma(T_{i,j}^4 - T_{amb}^4) \right]
\]

where

\[
C_{1,x} = A \left( \frac{T_{i+1,j} - T_{i-1,j}}{2\Delta x} \right)^2, \quad C_{1,y} = A \left( \frac{T_{i,j+1} - T_{i,j-1}}{2\Delta y} \right)^2
\]

\[
C_{2,x} = k \left( \frac{T_{i+1,j} + T_{i-1,j}}{2\Delta x} \right), \quad C_{2,y} = k \left( \frac{T_{i,j+1} + T_{i,j-1}}{2\Delta y} \right)
\]

Because of the inherently stable nature of the conduction terms and because the radiation term typically does not dominate the solution it is possible to get reasonable convergence with this method.

### 2.3 Grid Size Considerations

Sizing the grid for this particular solution presented two unique challenges. First, the method of successive substitution converges extremely slowly for fine grids. Thus it was impractical to use a very fine grid through the entire domain and solution process. Second, the very narrow laser beam requires a fine grid to reasonably resolve it. In order ensure rapid convergence the process was started with a coarse grid and the grid was repeatedly enlarged using guess values from the coarser solutions until a very fine resolution could be reached. During the early solutions with poor grid resolution the total power of the laser beam was place on a single node in the center of the grid. Once the grid became large enough the actual model for the laser was implemented.
Chapter 3

Results and Discussion

3.1 Solution Form

A typical temperature distribution is shown in Figure 3.1. The form of the solution is much as expected. Clearly where the laser strikes the plate the temperature increases very sharply and levels out toward the edge of the plate. Still there is some significant slope near the edges where the heat continues to be dissipated. This basic form of the solution holds for any case in which the laser beam is very small compared to the size of the plate. As the beam becomes larger visible rounding or smoothing of the sharp profile occurs near the center.

Figure 3.1: Temperature Distribution

3.2 Solution Accuracy

Since the exact solution to this problem is not available the accuracy can only be estimated by how it changes with iterations, grid changes, or other variations purely of the solver. Figure 3.2 shows a typical distribution of the changes after each iteration as the solution begins to converge. Clearly the solution near the center of the plate where the temperature
changes rapidly in space does not converge as quickly as the rest of the domain. This is not surprising considering that the grid does not get finer near the center where the solution is inherently more difficult to resolve.

Varying the grid size, guess values and iterations created some significant changes (one the order of 1%) in the apparently converged solutions for several configurations. This also implies that superior grid characteristics and possibly solution methods could improve the quality of the solutions significantly.

### 3.3 Impact of Radiation

While radiation was not a dominant factor in the basic form of the solution it did play a significant role when the conduction coefficient was low or the plate size was high. Figure 3.3 shows the temperature distribution with prominent radiation cooling.
Figure 3.4 shows the temperature distribution of the exact same configuration but not considering radiation at all. Comparing the two distributions it is evident that the radiation reduces the temperature and causes the heat to be even more concentrated toward the center since it does not have to all conduct to the edges of the plate. Still conduction dominates the behavior of the temperature profile overall.
Chapter 4

Conclusions

Since meaningful dimensional results were not obtained it is difficult to draw many quantative conclusions about the physics of the problem. However, the results show approximately the expected behavior for the physical situation being studied. Still it is obvious that the methods used could be improved to produce a more consistent solution that one might have greater confidence in.

The basic numerical method was easy to implement and relatively computationally light as well but the accuracy of its results may be somewhat questionable. The method is also not terribly versatile and would not be suited to problems more dominated by radiation cooling. The grid was perhaps the largest weakness of the analysis since it was not specially fitted to the problem. A more carefully generated grid better tailored to the model probably would have yielded superior results. Overall the results of the simulation were helpful in visualizing the physical solution but could still be improved.
Bibliography


<?php
    set_time_limit(180);
    // Set up problem parameters
    $A = 0.0314; // Conductivity constant
    $B = 15.2; // Conductivity constant
    $D = 0.521; // Specific heat constant
    $E = 441; // Specific heat constant
    $rho = 7900; // Density
    $eps = 0.75; // Emissivity
    $sig = 5.67e−8; // Stefan–Boltzmann constant
    $q0 = 1e8; // Laser intensity
    $w = 0.001; // Beam radius
    $Ta = 10; // Ambient temperature
    $pi = 3.1415926536;

    // Set up grid parameters
    $X = 0.1; // (m)
    $Y = 0.1; // (m)
    $I = 11;
    $J = 11;
    $dx = $X/($I−1);
    $dy = $Y/($J−1);

    // Set up solution matrix
    $g = fopen("test",'r');
    $T = guess(rdmat($g),3,3,$I,$J);
    ////$T = rdmat($g);
    fclose($g);

    // Prepare to iterate
    $To = $T;
    $itm = 200;
    $err = 1;
    $q = laser($q0,$w,$I,$J,$X,$Y);
    $e = zero($I,$J);
for ($it = 0; $it < $itm; $it++) { // Iterate algorithm
    if ($err < 1e−2 && $it < $itm - 100 && $I < 150) { // Increase grid size on the fly
        $Iadd = 20;
        $Jadd = 20;
        $I += $Iadd;
        $J += $Jadd;
        $dx = $X/($I−1);
        $dy = $Y/($J−1);
        $q = laser ($q0 , $w , $I , $J , $X , $Y);
        $e = zero ($I , $J);
        $T = guess ($T , $I−$Jadd , $I−$Jadd , $I , $J);
    }
    $err = 0;
    // Apply explicit expression to every grid point
    for ($i = 1; $i < $I−1; $i++){
        for ($j = 1; $j < $J−1; $j++){
            $k = $A*$_T[$i][ $j ] + $B;
            $rad = $eps*$sig*(pow(($_T[$i][$j]+273.15) ,4) − pow(($Ta +273.15) ,4));
            $cond = $A*(pow(($_T[$i+1][$j] − $_T[$i−1][$j])/(2*$dx) ,2) + pow(($_T[$i][$j+1] − $_T[$i][$j−1])/(2*$dy) ,2)) + $k *(($_T[$i+1][$j] + $_T[$i−1][$j])/(pow($dx ,2)) + ($_T[$i][$j +1] + $_T[$i][$j−1])/(pow($dy ,2)));
            $den = 2*$k*(1/pow($dx ,2) + 1/pow($dy ,2));
            $_T[$i][$j] = ($cond + $q[$i][$j])/$den;
            $err += abs($_T[$i][$j] − $To[$i][$j]);
            $e[$i][$j] = abs($_T[$i][$j] − $To[$i][$j]);
        }
    }
    $To = $_T;
    $err = $err/($I*$J);
}
$new = fopen(’guess.dat’, ’w+’);
$e = fopen(’q.dat’, ’w+’);
reemat($e, $I , $J , $new);
function zero($I , $J){ // Create zero matrix or vector
    if ($I == 1){
        for ($i = 0; $i < $I; $i++){
            $A[$i] = 0;
        }
    } else {
        for ($i = 0; $i < $I; $i++){
            for ($j = 0; $j < $J; $j++){
                $A[$i][$j] = 0;
            }
        }
    }
function laser ($q0, $w, $I, $J, $X, $Y) { // Set up laser heat distribution
    $pi = 3.1415926536;
    $dx = $X / ($I - 1);
    $dy = $Y / ($J - 1);
    if ($dx < $w && $dy < $w) {
        for ($i = 0; $i < $I; $i++) {
            for ($j = 0; $j < $J; $j++) {
                $x = $dx * $i;
                $y = $dy * $j;
                $d = sqrt (pow(($x - $X / 2), 2) + pow(($y - $Y / 2), 2));
                $q[$i][$j] = $q0 * exp(-2 * pow($d, 2) / pow($w, 2));
            }
        }
    } else {
        $q = zero($I, $J);
        $Ic = round(($I - 1)/2);
        $Jc = round(($J - 1)/2);
        $q[$Ic][$Jc] = $q0 * ($pi * $w * $w) / (2 * $dx * $dy);
        echo 'Warning: No laser definition.';
    }
    return $q;
}

function guess ($g, $Ig, $Jg, $I, $J) { // Move guess values to larger grid
    $ig = 0;
    $jg = 0;
    for ($i = 0; $i < $I; $i++) {
        $jg = 0;
        for ($j = 0; $j < $J; $j++) {
            $F[$i][$j] = $g[round($ig)][round($jg)];
            $jg += ($Jg - 1) / ($J - 1);
        }
        $ig += ($Ig - 1) / ($I - 1);
    }
    return $F;
}

function recmat ($F, $I, $J, $file) { // Write matrix to file
    for ($i = 0; $i < $I; $i++) {
        for ($j = 0; $j < $J; $j++) {
            $line = $i . " ". $j . " ". $F[$i][$j]."\n";
            fwrite ($file, $line);
        }
    }
fwrite($file,'\n');
}

function rdmat($file){  // Read data file into matrix
    while (($i = fgetn($file)) !== false && ($j = fgetn($file))
        !== false && ($n = fgetn($file)) !== false)
    {
        $F[$i][$j] = $n;
    }
    return $F;
}

function fgetn($file){  // Get next number from data file
    $n = 0;
    $a = 0;
    $o = 0;
    while($a < 3 && ($c = fgetc($file)) !== false)
    {
        if (($c > 0 && $c < 10) || $c === "0")
        {
            $n = $n * 10 + $c;
            switch($a){
                case 1:
                    break;
                case 0:
                    $a = 1;
                    break;
                case 2:
                    $o++;
                    break;
            }
        } else if ($c === ".")
        {
            $a++;
        } else if ($a > 0)
        {
            $a = 3;
        }
    }
    if ($a == 0)
    {
        return false;
    } else {
        return $n/pow(10,$o);
    }
}

function lsm($I,$J,$A){  // prints the values of a matrix in an html table
    echo '\n<table >';
    for ($i = 0; $i < $I; $i++)
    {
        echo '\n<tr >';
        for ($j = 0; $j < $J; $j++)
        {
            echo '\n<td >$A[$i][$j]$</td>
        }
        echo '\n</tr >';
    }
    echo '\n</table >';
echo "<td width="40px">\n\t<center>\n\t	$A[$i][$j].\n\t</center>\n\t</td>";