MEMS 1060: Homework #6

Due on March 11, 2021 at 6:00pm

Professor Sammak Th 6:00PM

Made using LaTeX/Inkscape. Source files available upon request.

Shane Riley

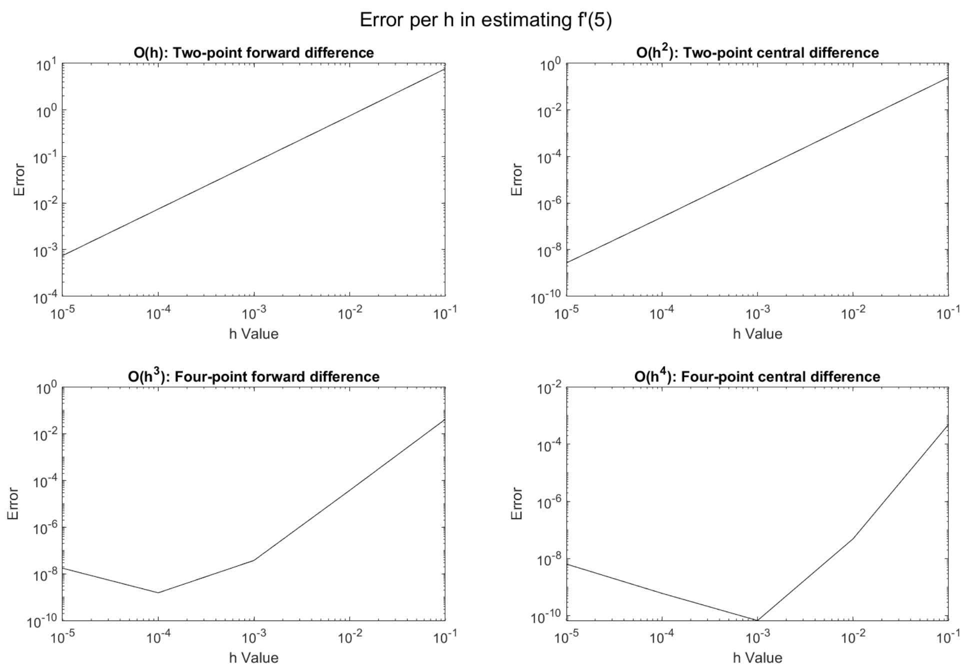
Table 1: Schemes chosen to model f'(x)

Order	Num Points	Type	Formula
$O(h)$	2	Forward	$\frac{f(x+h)-f(x)}{h}$
$O(h^2)$	2	Central	$\frac{f(x+h)-f(x-h)}{2h}$
$O(h^3)$	4	Forward	$\frac{-11f(x)+18f(x+h)-9f(x+2h)+2f(x+3h)}{6h}$
$O(h^4)$	4	Central	$\frac{f(x-2h)-8f(x-h)+8f(x+h)-f(x+2h)}{6h^4}$

Problem 1

Evaluate $f(x) = e^x$ at $x = 5$ where $h = [10^{-1}10^{-2}10^{-3}10^{-4}10^{-5}]$ using four finite difference approximations of varying schemes and orders of accuracy. Plot the errors as a function of h as loglog and explain the slope of the errors from each approximation.

Figure 1: Error per h.



Solution

4 schemes are selected at random to model $f'(x)$ at various orders of accuracy.
The error plotted is just the absolute difference between the true value $f'(a)$ for $a = 5$ and $f(x) = e^x$ for different values of h.

First, we notice that with increasing order of accuracy the error decreases. That is expected. In both two-point schemes, we see that for the entire interval of h the error decreases with decreasing h. This indicates that truncation error is dominant for our values of h, and that we could decrease error further by decreasing h. For the four-point schemes, however, we see an h value within the interval that minimizes error. When

Problem 1 continued on next page. . .

the error begins increasing with decreasing h (to the left of the minimum), the round-off error becomes dominant. Additionally, the minimum value of error seen on these intervals is about the lowest we should expect to get using the scheme.
See the plots and commentary.

Problem 2

BONUS QUESTION: Temperature per length data is provided (evenly-spaced) for a rod-shaped fin. Determine $Q_x(x)$, where Q_x is the heat flux through the fin. Use an $O(h^2)$ scheme. Thermal conductivity is $k = 240[W/m - k]$.

Solution

Since the data for $T(x)$ is evenly spaced, we can simply say that h is equal to the spacing, which is $0.01[m]$. We will implement a 2 point central differencing scheme (same as in Problem 1), using the following formula:

$$\frac{dT}{dx} = \frac{T(x_{i+1}) - T(x_{i-1})}{2h} + O(h^2)$$

Applying the formula for Q_x yields the following approximation using the data we are provided:

$$\begin{aligned} h &= 0.01[m] \\ Q_x &= -k * \frac{dT}{dx} \\ Q_x(x) &\approx -k * \frac{T(x_{i+1}) - T(x_{i-1})}{2h} \end{aligned}$$

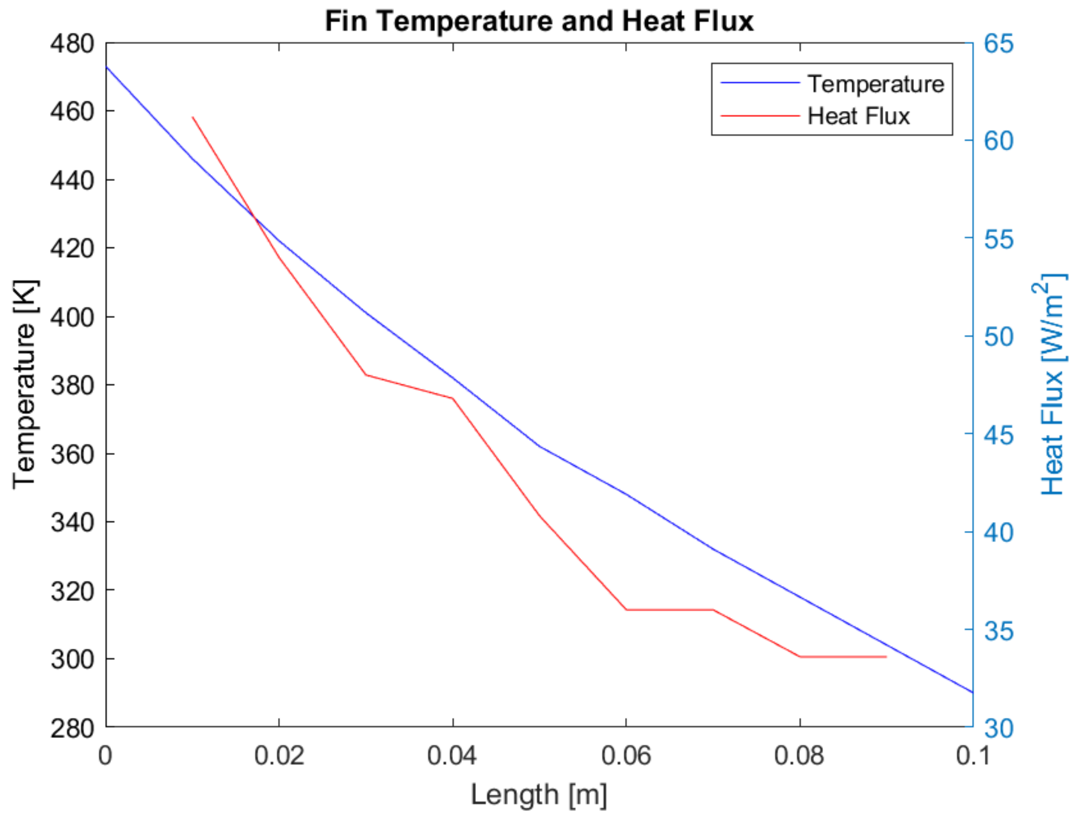
The solution is implemented in MATLAB, and the solution is plotted. Since the approximation requires a point before and after x, the endpoints do not have values. We could implement forward and backward difference schemes of the same order of accuracy to get approximations for the endpoints, but I will spare the work to do so.
See the plots and commentary.

```
%% MEMS 1060 Homework 6
% Author: Shane Riley
% Date: 3/7/2021
format long

%% Problem 1
% Implement four finite difference schemes ranging 1st to 4th order to approximate f'(a). h =
% 10^{-1 through -5). Plot the error per h on a loglog. Write your
% conclusions
% f(x) = exp(x), x = a = 5

f = @(x) exp(x);
df = @(x) exp(x);
a = 5;
tf = f(a);
tdf = df(a);
```

Figure 2: Fin Temperature and Heat Flux.




```
% 1st order:
% Two-point forward differnence O(h)
order1 = @(x,h) (f(x+h) - f(x))/h;

% 2nd order:
% Two-point central difference O(h^2)
order2 = @(x,h) (f(x+h) - f(x-h))/(2*h);

% 3rd order:
% Four point forward difference O(h^3)
order3 = @(x,h) ( -11*f(x) + 18*f(x+h) - 9*f(x+2*h) + 2*f(x+3*h))/(6*h);

% 4th order:
% Four point central difference O(h^4)
order4 = @(x,h) (f(x-2*h) - 8*f(x-h) + 8*f(x+h) - f(x+2*h))/(12*h);


% error
error = @(a, t) abs(a - t);


```

```
% Run the schemes
hvalues = 10.^(-1:-1:-5);

order1errors = zeros(1,length(hvalues));
order2errors = zeros(1,length(hvalues));
order3errors = zeros(1,length(hvalues));
order4errors = zeros(1,length(hvalues));

for i=1:length(hvalues)
    h = hvalues(i);
    order1errors(i) = error(order1(a, h), tdf);
    order2errors(i) = error(order2(a, h), tdf);
    order3errors(i) = error(order3(a, h), tdf);
    order4errors(i) = error(order4(a, h), tdf);
end

% Plot the results

subplot(2,2,1);
loglog(hvalues, order1errors, 'k-');
title("O(h): Two-point forward difference");
xlabel("h Value");
ylabel("Error");
subplot(2,2,2);
loglog(hvalues, order2errors, 'k-');
title("O(h^2): Two-point central difference");
xlabel("h Value");
ylabel("Error");
subplot(2,2,3);
loglog(hvalues, order3errors, 'k-');
title("O(h^3): Four-point forward difference");
xlabel("h Value");
ylabel("Error");
subplot(2,2,4);
loglog(hvalues, order4errors, 'k-');
title("O(h^4): Four-point central difference");
xlabel("h Value");
ylabel("Error");
sgtitle("Error per h in estimating f'(5)");
print("images/figure1",'-dpng');

%% Problem 2
% Given T and x vectors, find the Q_x vector using a O(h^2) scheme

T = [...
    473
    446
    422
    401
```

```
    382
    362
    348
    332
    318
    304
    290
    ]';  % [K]
h = 0.01;  % [m]
xT = 0:h:0.10;  % [m]
k = 240;  % 240 [W/m-K]

% Scheme
Qformula = @(tbefore, tafter) (-k/2*h) * (tafter - tbefore);
xQ = [];
Q = [];
% Run scheme
for i=1:length(xT)

    % Skip if at an endpoint
    if (i == 1); continue; end
    if (i == length(xT)); continue; end
    xQ(end+1) = xT(i);
    Q(end+1) = Qformula(T(i-1), T(i+1));
end

% Plot
figure(2);

xlabel("Length [m]");
plot(xT, T, 'b-');
ylabel("Temperature [K]");

yyaxis right
plot(xQ, Q, 'r-');
ylabel("Heat Flux [W/m^2]");
title("Fin Temperature and Heat Flux");
legend("Temperature", "Heat Flux");
print("images/figure2",'-dpng');
```