

BME 1 - integration

The given coordinate list contains the vertices of a building's survey. Use these to determine the floor area of the building. The maximum score available in the task is 15 in total.

a) Load the given data (x, y coordinates) and display the vertices and sides of the building in a closed shape. (2p)

Coordinate list of building vertices: bme_K.txt

b) Fit an interpolating function to the building's vertices, ensuring that the interpolation is linear. Draw the interpolating function in the figure as well. Using interpolation, resample the surveyed points along the boundary line to have 50 points that are evenly distributed and determine the area with a trapezoidal rule based on the new vertices. Also, show the new vertices in the previous figure. (6p)

c) What would be the area calculated in the previous point if we interpolated the boundary line based on the built-in spline approximation? Also, show the applied new functions and the new points in the previous figure. Add a legend on the figure for easier interpretation. (4p)

d) What is the building's perimeter assuming straight lines between the newly created points in cases **b)** and **c)**? (3p)

```
clear all; close all
% a.) load and plot (2p)
xy = load('bme_K.txt');
x = xy(:,1); y = xy(:,2);
x(end+1,1)=x(1,1); y(end+1,1)=y(1,1);
figure
plot(x,y,'r*'); hold on
% b.) linear interpolation, area with trapezoid method
dx=diff(x); % x coord. differences
dy=diff(y); % y coord. differences
u=[0; sqrt(dx.^2+dy.^2)]; % distance between points
s=cumsum(u); % distances from the beginning -> approximating arc length
meth = 'linear';
xslin=@(p) interp1(s,x,p,meth);
yslin=@(p) interp1(s,y,p,meth);
% plot
hold on;
ezplot(xslin,yslin,[s(1),s(end)]);
% Area calculation using 50 points
p_i=linspace(0, s(end), 50)';
xi=xslin(p_i);
yi=yslin(p_i);
plot(xi,yi,'bx')
xi1=xi([2:end,1]); % x_(i+1) vector
yi1=yi([2:end,1]); % y_(i+1) vector
Tp2=sum((yi1+yi).*(xi1-xi)/2)
```

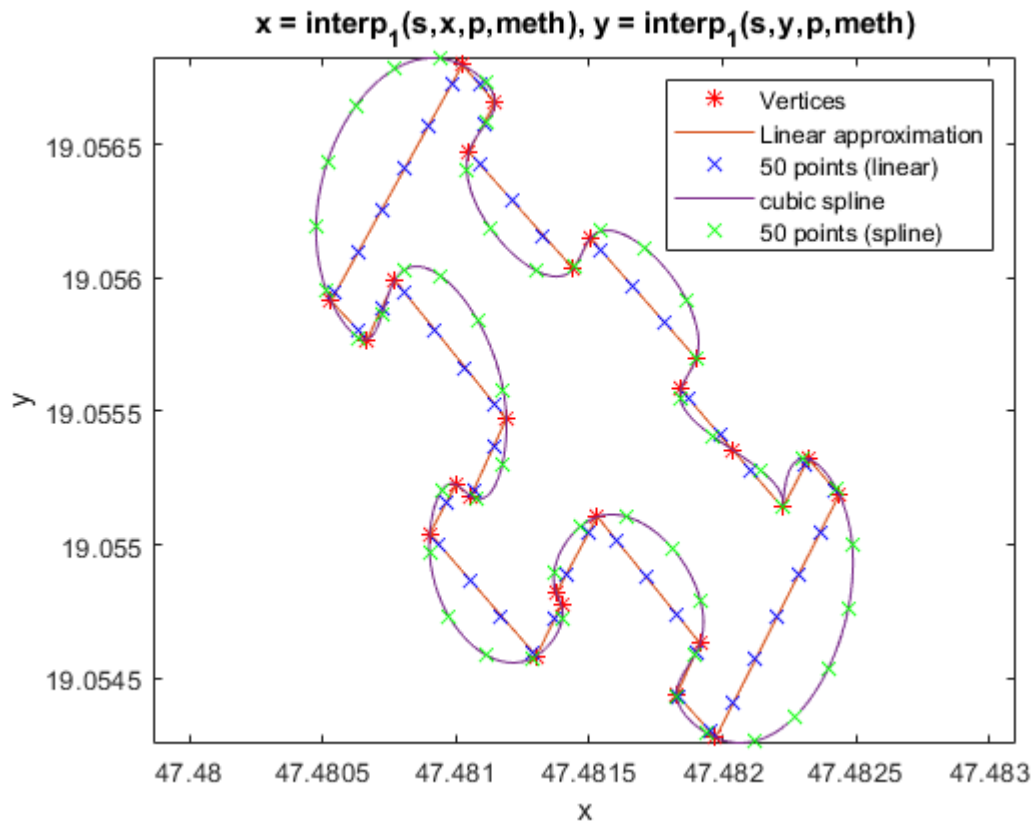
```
Tp2 =
    1.6916e-06
```

```
% c.) interpolation and area with cubic spline
```

```

meth = 'spline';
xs=@(p) interp1(s,x,p,meth);
ys=@(p) interp1(s,y,p,meth);
% plot
hold on;
ezplot(xs,ys,[s(1),s(end)]);
p_i=linspace(0, s(end), 50)';
xis=xs(p_i);
yis=ys(p_i);
plot(xis,yis,'gx')
legend('Vertices','Linear approximation',...
      '50 points (linear)','cubic spline', '50 points (spline)')

```



```

xi1=xis([2:end,1]); % x_(i+1) vector
yi1=yis([2:end,1]); % y_(i+1) vector
Tp3=sum((yi1+yis).*(xi1-xis)/2)

```

```

Tp3 =
    2.0139e-06

```

```

% d.) Perimeter
% közelítő:
sum(u) % vagy s(end)

```

```

ans =
    0.0088066

```

```

% lin:
dxi=diff(xi); % x coord. differences
dyi=diff(yi); % y coord. differences

```

```
us=[0; sqrt(dxi.^2+dyi.^2)]; % distance between points
sum(us)
```

```
ans =
    0.0080748
```

```
% spline:
dxi=diff(xis); % x coord. differences
dyi=diff(yis); % y coord. differences
us=[0; sqrt(dxi.^2+dyi.^2)]; % distance between points
sum(us)
```

```
ans =
    0.0097696
```

Intregation 2.

Trapezoid

The given coordinate list contains the vertices of a building's survey. Use these to determine the floor area of the building. The maximum score available in the task is 15 in total. **a)** Load the given data (x, y coordinates):

```
xy = [1,1;2,1;2,2;1,2]
```

Display the vertices and sides of the building in a closed shape. (2p)

b) Fit an interpolating function to the building's vertices, ensuring that the interpolation is linear. Draw the interpolating function in the figure as well. Using interpolation, resample the surveyed points along the boundary line to have 40 points that are evenly distributed and determine the area with a trapezoidal rule based on the new vertices. Also, show the new vertices in the previous figure. (6p)

c) What would be the area calculated in the previous point if we interpolated the boundary line based on the built-in 'pchip' approximation? Also, show the applied new functions and the new points in the previous figure. Add a legend on the figure for easier interpretation. (4p)

d) What is the building's perimeter assuming straight lines between the newly created points in cases **b)** and **c)**? (3p)

```
% a.)
clear all; close all
%xy = [1.5,1;3.5,1;3,2;2,2];
% LP: Unfortunately, here wrong coordinates were placed on the worksheet,
% instead of the one in the previous row, % a unit-sided square coordinates
% were given, which was not designed that way, but the task can also be solved.
xy = [1,1;2,1;2,2;1,2];
x = xy(:,1); y = xy(:,2);
x(end+1,1)=x(1,1);
y(end+1,1)=y(1,1);
figure; plot(x,y,'r*'); hold on
% b.)
dx=diff(x); % x coord. differences
dy=diff(y); % y coord. differences
u=[0; sqrt(dx.^2+dy.^2)]; % distance between points
```

```

s=cumsum(u); % distances from the beginning -> approximating arc length
meth = 'linear';
xs=@(p) interp1(s,x,p,meth);
ys=@(p) interp1(s,y,p,meth);
% Plot
hold on; ezplot(xs,ys,[s(1),s(end)]);
% 40 points
pi=linspace(0, s(end), 40)';
xi=xs(pi); yi=ys(pi);
plot(xi,yi,'bx')
xi1=xi([2:end,1]); % x_(i+1) vector
yi1=yi([2:end,1]); % y_(i+1) vector
Tp2=abs(sum((yi1+yi).*(xi1-xi)/2))

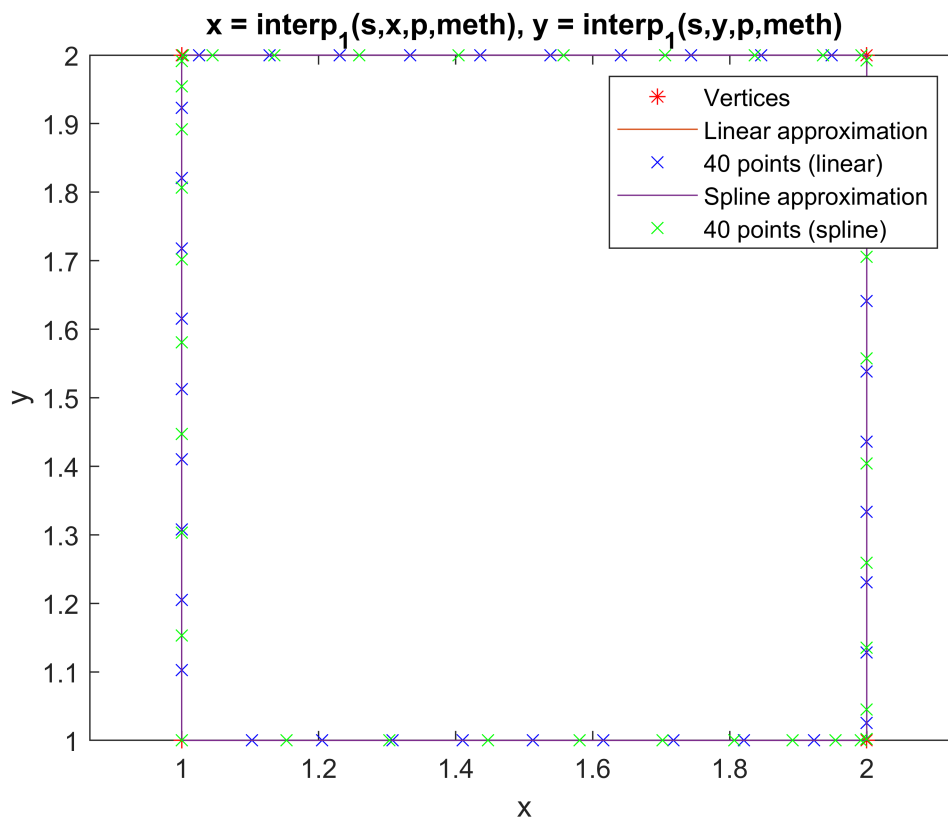
```

Tp2 =
0.99671

```

% c.)
meth = 'pchip';
xs=@(p) interp1(s,x,p,meth);
ys=@(p) interp1(s,y,p,meth);
% Plot
hold on; ezplot(xs,ys,[s(1),s(end)]);
p_i=linspace(0, s(end), 40)';
xis=xs(p_i);
yis=ys(p_i);
plot(xis,yis,'gx')
legend('Vertices','Linear approximation',...
      '40 points (linear)','Spline approximation', '40 points (spline)')

```



```
xxi1=xis([2:end,1]); % x_(i+1) vector
yi1=yis([2:end,1]); % y_(i+1) vector
Tp3=abs(sum((yi1+yis).*(xxi1-xis)/2))
```

```
Tp3 =
    0.99995
```

```
% d.) Perimeter
sum(u) % or s(end)
```

```
ans =
    4
```

```
% lin:
dxi=diff(xi); % x coord. differences
dyi=diff(yi); % y coord. differences
us=[0; sqrt(dxi.^2+dyi.^2)]; % distance between points
sum(us)
```

```
ans =
    3.927
```

```
% spline:
dxi=diff(xis); % x coord. differences
dyi=diff(yis); % y coord. differences
us=[0; sqrt(dxi.^2+dyi.^2)]; % distance between points
sum(us)
```

```
ans =
    3.9921
```

Differential equation 1.

We want to model the number of expected diseases in the population if, based on previous experience, there are initially 12,000 inhabitants in a given community, and 105 of these are found to be infectious after random testing. If the population acts with sufficient caution, the incidence rate based on experience in similar communities, summing up newly diagnosed diseases every week, is 2%.

That is, the following differential equation describes the increase:

$$\frac{d}{dt} \text{contagious_number} = \text{incidence_rate} * \text{contagious_number}$$

- How does the number of infections change after three months, broken down by week? Solve the differential equation and show the process in the figure as well. (6p)
- Fit an interpolating function to the obtained numerical result, which determines the exact number of infected as a function of time, and plot it in the previous figure. (3p)
- How does the task so far change if we consider the **combined** effect of the followings:

Many people in a given settlement are unwilling to accept and follow even the minimum precautions, so the infection rate jumps seven times of what was previously measured!

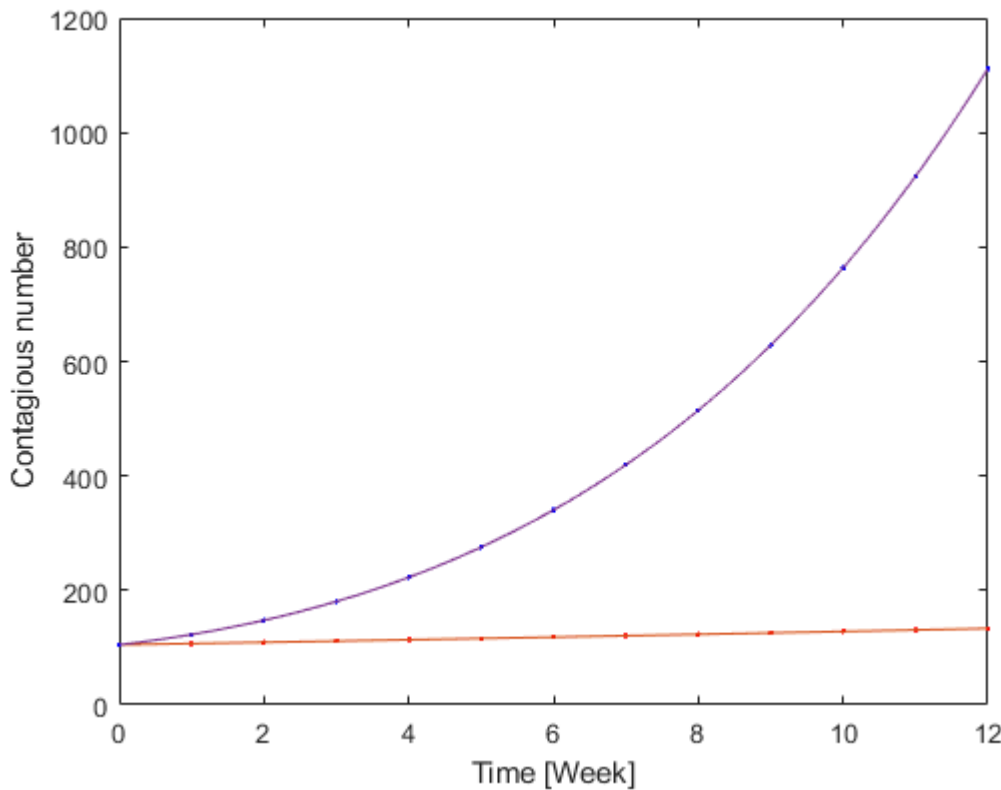
- Due to the overcrowding of social homes from the neighboring settlement, more residents are sent to the local free municipal flats every week, but unfortunately, an average of 4 new residents tend to be contagious.

$$\frac{d}{dt} \text{contagious_number} = \text{incidence_rate} * \text{contagious_number} + 4 * t$$

Solve problem parts a) and b) considering the modifications and show the results in the previous figure. (6p)

$$\text{ddtcontagious_number} = \text{incidence_rate} * \text{contagious_number} + 4 * t$$

```
% a.)
clear all;close all;
contagious_number=105;
incidence_rate=0.02;
dfdt=@(t,contagious_number) incidence_rate*contagious_number;
[t,f] =ode45(dfdt,[0:1:12],contagious_number);
figure;
plot(t,f,'r.')
xlabel('Time [Week]')
ylabel('Contagious number')
% b.)
fbet=@(u)interp1(t,f,u,'pchip');
hold on; fplot(fbet,[0,12])
% c.)
contagious_number=105;
incidence_rate=0.02*7;
dfdt=@(t,contagious_number) incidence_rate*contagious_number+4*t;
[t2,bet2] =ode45(dfdt,[0:1:12],contagious_number);
plot(t2,bet2,'b.')
fbet2=@(u)interp1(t2,bet2,u,'pchip');
hold on; fplot(fbet2,[0,12])
```



Differential equation 2.

We want to model the number of expected diseases in the population if, based on previous experience, there are initially 12,000 inhabitants in a given community, and 103 of these are found to be infectious after random testing. If the population acts with sufficient caution, the incidence rate based on experience in similar communities, summing up newly diagnosed diseases daily, is 1%.

That is, the following differential equation describes the increase:

$$\frac{d}{dt} \text{contagious_number} = \text{incidence_rate} * \text{contagious_number}$$

$$\text{contagious_number} = \text{incidence_rate} * \text{contagious_number}$$

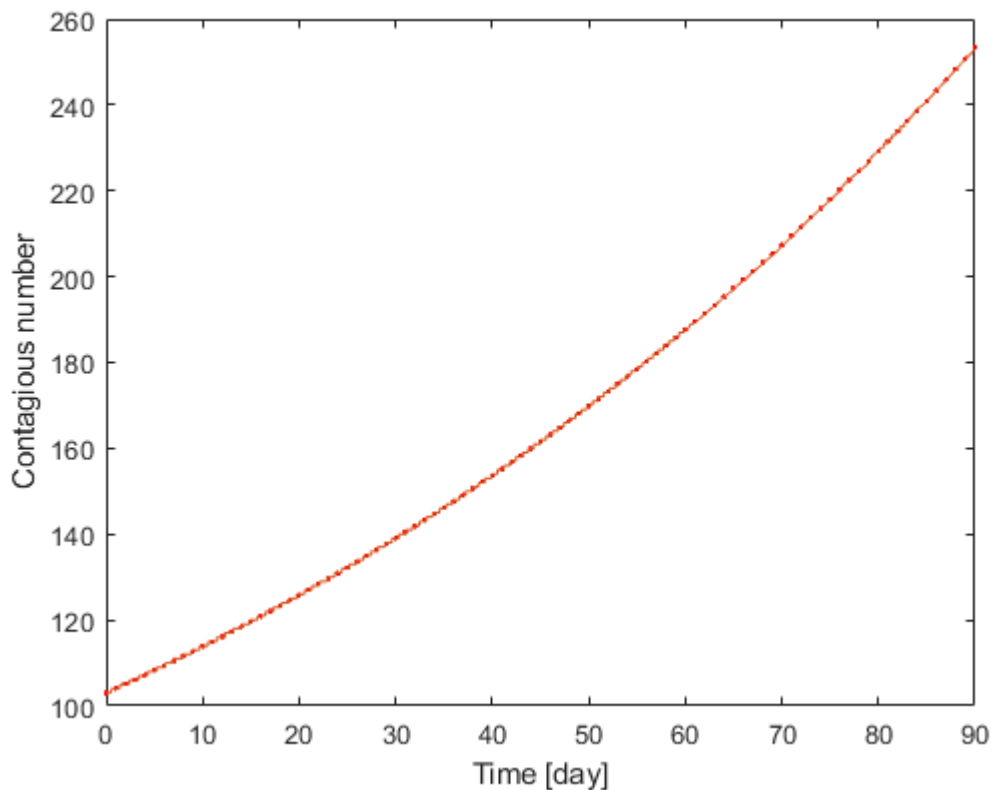
- a)** How will the number of infections in the first three months of this year change daily if the infected data above were measured at the beginning of the year? Solve the differential equation and show the process in the figure as well. (6p)
- b)** Fit an interpolating function to the obtained numerical result, which determines the exact number of infected as a function of time, and plot it in the previous figure. (3p)
- c)** Assuming that there is no change in either the population's habits or the nature of the disease, on what day is it expected that a third of the people will be infected in the given settlement? Show the number of infections in this way on a new chart. (6p)

% a.)

```

clear all;close all;
contagious_number=103;
incidence_rate=0.01;
dfdt=@(t,contagious_number) incidence_rate*contagious_number;
[t,f]=ode45(dfdt,[0:1:90],contagious_number);
figure;
plot(t,f,'r.')
xlabel('Time [day]')
ylabel('Contagious number')
% b.)
fbet=@(u)round(interp1(t,f,u,'pchip'));
hold on; fplot(fbet,[0,90])

```



```

% c.)
population = 12000;
[t,f]=ode45(dfdt,[0:1:500],contagious_number);
fbet=@(u)round(interp1(t,f,u,'pchip'));
figure; hold on; fplot(fbet,[0,500])
F=@(u)fbet(u)-population/3;
fplot(fbet,[0,500])
hold on;
T=fzero(F,500)

```

T =
365.93

```

plot(T,fbet(T),'k*')
fplot('4000',[0,500])

```


Warning: fplot will not accept character vector or string inputs in a future release. Use `fplot(@(t)4000*ones(size(t)))` instead.

