Inna WIlliams

##

> with(student); with(MTM) :

[D, *Diff*, *Doubleint*, *Int*, *Limit*, *Lineint*, *Product*, *Sum*, *Tripleint*, *changevar*, *completesquare*, *distance*, *equate*, *integrand*, *intercept*, *intparts*, *leftbox*, *leftsum*, *makeproc*, *middlebox*, *middlesum*, *midpoint*, *powsubs*, *rightbox*, *rightsum*, *showtangent*, *simpson*, *slope*, *summand*, *trapezoid*] **(1)**

##

Problem 1

Consider the parametric curve c(t) = (t cos(t); t sin(t)),

$$\left(\int_a^b (\cos(t) - t\sin(t))^2 + (\sin(t) + t\cos(t))^2\right)^{\frac{1}{2}}$$ **(a spiral). Recall**

from calculus that the arclength formula of a parametric curve c(t) = (x(t); y(t)) with a <= t <= b

##

(a)

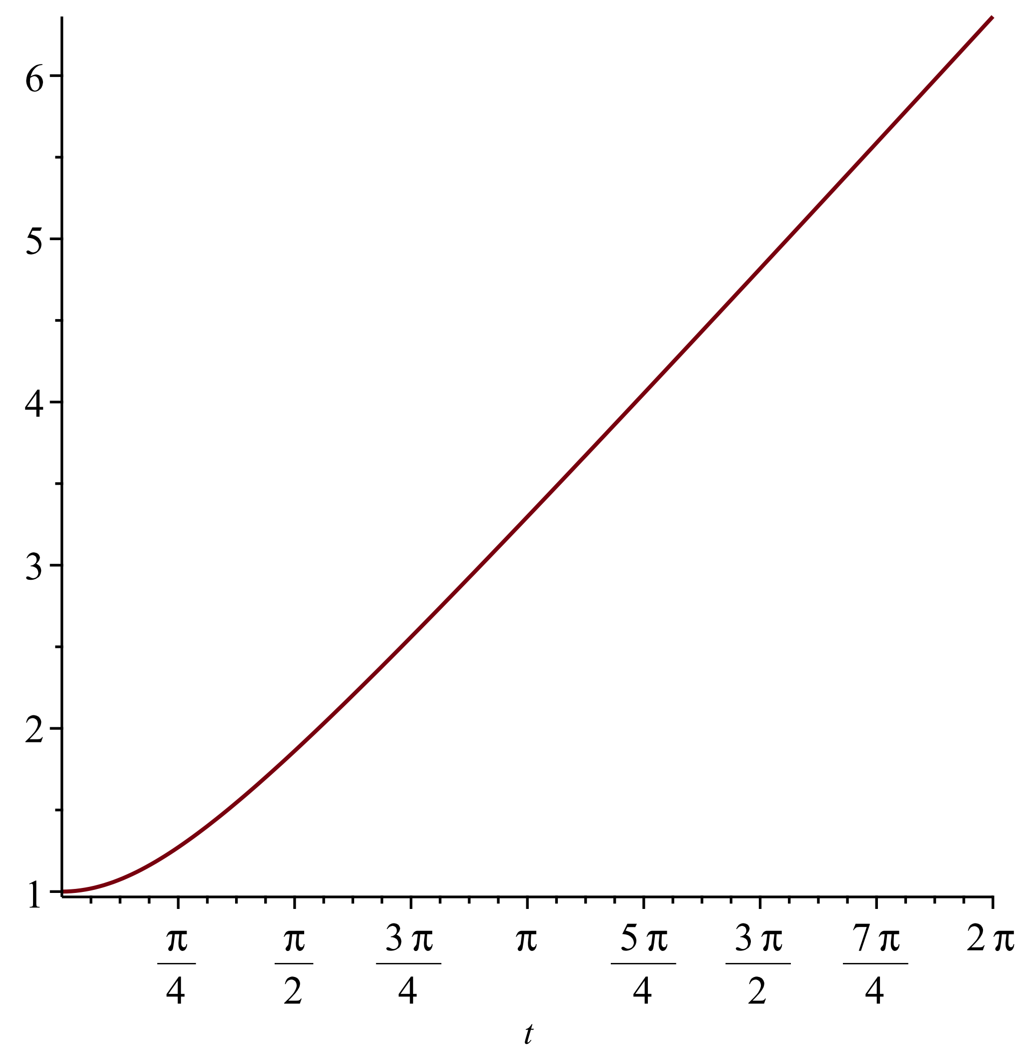
Plot the curve using Maple.

##

> $a := 0 : b := 2\cdot\pi : f := t \rightarrow \left((\cos(t) - t\sin(t))^2 + (\sin(t) + t\cos(t))^2 \right)^{\frac{1}{2}}$

$$f := t \mapsto \sqrt{(\cos(t) - t\sin(t))^2 + (\sin(t) + t\cos(t))^2} \quad \textbf{(2)}$$

> $p1 := plot(f(t), t = a..b) : display(p1)$



> $theoretical\_value := \mathbf{evalf}\left(\int_a^b f(t)\,\mathrm{d}t\right)$

$$theoretical\_value := 21.25629415 \tag{3}$$

##

(b) Calculate by hand the arclength of the spiral.

Calculated by hand (written attached)

$$\int (\sec(u))^3 du = \pi \cdot \left(1 + 4 \cdot \pi^2\right)^{\frac{1}{2}} + \frac{1}{2} \ln\left(2 \cdot \pi + \left(1 + 4 \cdot \pi^2\right)^{\frac{1}{2}}\right)$$

##

> $calculated\_byhand := evalf\left(\pi \cdot \left(1 + 4 \cdot \pi^2\right)^{\frac{1}{2}} + \frac{1}{2} \ln\left(2 \cdot \pi + \left(1 + 4 \cdot \pi^2\right)^{\frac{1}{2}}\right)\right)$

$$calculated\_byhand := 21.25629415 \tag{4}$$

Answer:

***calculated_byhand* := 21.25629415**

***theoretical_value* := 21.25629415**

##

(c)
Use Simpson's rule to approximate the integral. Find the error bound using the appropriate error formula, and compare it to the actual error.
Attached also calculated by hand
###

```
> approximation := proc (f, m, y0, ym, type)
    if type ='sim' then
          print('Simpson');
           approximation := simpson(f(x), x=y0..ym, 2*m);
          error_bound := evalf((ym-y0)^5*max(abs((D(D(D(D(f))))) (y0)),
        abs((D(D(D(D(f))))) (ym)))/(2880*m^4));
          min := theoretical_value - error_bound;
          max := theoretical_value + error_bound;
    elif type ='trap' then
        print("Trapezoid");
        approximation := trapezoid(f(x), x=y0..ym, m);
        error_bound := evalf((ym-y0)^3*max(abs((D(D(f))) (y0)), abs((D(D(f))) (ym)))/(12*m
        ^2));
        min := theoretical_value - error_bound;
        max := theoretical_value + error_bound;
    end if;
      approximation_value := evalf(approximation);
      appoximation_error := abs(theoretical_value-approximation_value);
      printf("\n     m       value       error       error_bound               error_bounds\n");
      printf(" ---------------------------------------------------------------------------------------------\n");
      printf(" %1.10f  %10.10f  %10.10f  %10.10f [  %10.10f ,  %10.10f  ]\n", m,
        approximation_value, appoximation_error, error_bound, min, max);

    end proc:
```

Check if the same what find by hand for m=1 Simpson's rule

```
> m := 1 : approximation(f, m, a, b, 'sim') :
                                  Simpson

        m             value             error           error_bound
                error_bounds
   ------------------------------------------------------------
 ---------------------------------
   1.0000000000  21.5198032500   0.2635091000   10.2006561700  [
  11.0556379800  ,   31.4569503200   ]
>
```

For m=1 we can see that error boud =10.2006561700 > then actual error = 0.263535

```
>
>
```

Answer:
appximated_value = 21.25629529

theoretical_value =21.25629415
actual_error = 0.26350910 < 10.20065617 = error_bound
theoretical_value − error_bounds < appximated_value < theoretical_value + error_bounds
approximated value is between error bounds
11.05563798 < 21.25629529 < 31.45695032

##
(d)
Use Maple or other computer software to approximate the given integral by using the composite trapezoidal rule with 10 subintervals. Compare the approximation with the actual value of the integral.
##

> $m := 10 : approximation(f, m, a, b, 'trap');$

```
                              "Trapezoid"

       m           value            error          error_bound
                error_bounds
  --------------------------------------------------------------
----------------------------------
  10.0000000000  21.2887805800   0.0324864300   0.2067085113  [
 21.0495856400  ,   21.4630026600   ]
```

##
Answer:
appximated_value = 21.2887805800
theoretical_value := 21.25629415
actual_error = 0.032486430 $<$ 0.2067085113 = error_bound
theoretical_value − error_bounds appximated_value $<$ theoretical_value + error_bounds
approximated value is between error bounds
21.04958564 $<$ 21.2887805800 $<$ 21.4630026600

##
Problem 2
##

Consider the following nonlinear system of 3 equations with 3 unknowns:

$f := (x, y, z) \mapsto x^2 + y + z - 37$

##

(a)

Apply two steps of Newton's method (by hand) starting with initial guess (1; 1; 1) written attached

$$f := (x, y, z) \rightarrow x^2 + y + z - 37$$
$$g := (x, y, z) \rightarrow x - y^2 - z - 5$$
$$k := (x, y, z) \rightarrow x + y + z - 3$$

##

(b) Change the NewtonMD Maple file (written for 2 equations with 2 unknowns) in order to find a numerical solution for the above system with six correct decimal places starting from (1:0; 1:0; 1:0).

##

Newton Method

```
> restart;
> with(plots, implicitplot) : with(plots) : with(LinearAlgebra) : with(VectorCalculus) :
> newtonMD := proc(X0, Y0, Z0, TOL, N, x, y, z)
      local i, s, x0, y0, z0, sol, A;
      x0 := X0; y0 := Y0; z0 := Z0; sol := ⟨x0, y0, z0⟩;
      i := 1;
      while i ≤ N do
        A := Jacobian(F, [x,y, z] = [x0, y0, z0]);
        if Determinant(A) = 0 then
            printf("Non-invertible Jacobian. Method failed");
            break;
         else
         s := LinearSolve(A, subs([x = x0, y = y0, z = z0], -F));
         sol := sol + s;
         x0 := DotProduct(sol, ⟨1, 0, 0⟩); y0 := DotProduct(sol, ⟨0, 1, 0⟩); z0 := DotProduct(sol, ⟨0, 0, 1⟩);
         printf("Step %d: x=%.8g, y=%.8g, z=%.8g \n", i, x0, y0, z0);
         if evalf(Norm(s)) < TOL then
           printf("Number of iterations needed: %d", i); return([x0, y0, z0]);
           break;
          end if;
        i := i + 1;
        end if;
     end do;
     printf("The method failed after %d iterations.\n", N);
     return( );
  end proc:
```

$> f1 := x^2 + y + z - 37; f2 := x - y^2 - z - 5; f3 := x + y + z - 3;$

$$f1 := x^2 + y + z - 37$$
$$f2 := -y^2 + x - z - 5$$
$$f3 := x + y + z - 3 \quad (5)$$

$> F := \langle f1, f2, f3 \rangle;$

$$F := (x^2 + y + z - 37)e_x + (-y^2 + x - z - 5)e_y + (x + y + z - 3)e_z \quad (6)$$

$> sol := newtonMD(1.0, 1.0, 1.0, 0.5 \cdot 10^{-6}, 100, x, y, z)$

```
Step 1: x=35, y=63, z=-95
```

```
Step 2: x=18.246377, y=31.979942, z=-47.226319
Step 3: x=10.338174, y=16.445282, z=-23.783456
Step 4: x=7.1597556, y=8.6786427, z=-12.838398
Step 5: x=6.4012935, y=4.8982104, z=-8.2995039
Step 6: x=6.3525529, y=3.2624145, z=-6.6149674
Step 7: x=6.35235, y=2.7780133, z=-6.1303632
Step 8: x=6.35235, y=2.7265113, z=-6.0788612
Step 9: x=6.35235, y=2.7259156, z=-6.0782656
Step 10: x=6.35235, y=2.7259155, z=-6.0782655
Number of iterations needed: 10
```

$$sol := [6.35234995535981, 2.72591552191893, -6.07826547727875] \quad (7)$$

>

Check if the solution is correct

> $f := (x, y, z) \rightarrow x^2 + y + z - 37$

$$f := (x, y, z) \mapsto x^2 + y + z - 37 \quad (8)$$

> $g := (x, y, z) \rightarrow x - y^2 - z - 5$

$$g := (x, y, z) \mapsto x + (-y^2) + (-z) - 5 \quad (9)$$

> $k := (x, y, z) \rightarrow x + y + z - 3;$

$$k := (x, y, z) \mapsto x + y + z - 3 \quad (10)$$

> $f(sol[1], sol[2], sol[3]); g(sol[1], sol[2], sol[3]); k(sol[1], sol[2], sol[3])$

$$0.$$
$$-6.21724893790088\ 10^{-15}$$
$$-8.88178419700125\ 10^{-16} \quad (11)$$

Answer
Newton Method converged after 10 intereations
solution = [6.35234995535981, 2.72591552191893, −6.07826547727875]

##

Problem 3

##

The table below shows how seven friends rated six movies from 1 to 5 (low to high), with 0 meaning \not viewed/no rating".

##

```
> restart( ) :
> with(LinearAlgebra) : with(plots) : interface(rtablesize = 15) :
> A := Matrix([[1, 0, 2, 4, 5], [5, 4, 4, 2, 0], [2, 2, 0, 5, 3], [5, 3, 4, 0, 2], [5, 4, 0, 2, 2], [0, 2, 3, 4, 0], [4, 0, 5, 1, 1]])
```

$$A := \begin{bmatrix} 1 & 0 & 2 & 4 & 5 \\ 5 & 4 & 4 & 2 & 0 \\ 2 & 2 & 0 & 5 & 3 \\ 5 & 3 & 4 & 0 & 2 \\ 5 & 4 & 0 & 2 & 2 \\ 0 & 2 & 3 & 4 & 0 \\ 4 & 0 & 5 & 1 & 1 \end{bmatrix} \quad (12)$$

\>

****

```
> U, S, Vt := SingularValues(A, output = ['U','S','Vt']) :
> Sm2 := DiagonalMatrix(S[1..2], 2, 2);
```

$$Sm2 := \begin{bmatrix} 15.0154674945463 & 0. \\ 0. & 7.21541522130896 \end{bmatrix} \quad (13)$$

```
> Um2 := SubMatrix(U, 1..7, 1..2);
```

$$Um2 := \begin{bmatrix} -0.311391548098427 & 0.588473218374909 \\ -0.485991620343302 & -0.321003851879510 \\ -0.330102422290805 & 0.544795306481915 \\ -0.446301804770097 & -0.361055696634442 \\ -0.404058030824304 & -0.0137140305343523 \\ -0.253294034085323 & 0.225519366390527 \\ -0.362317234766473 & -0.269157197649266 \end{bmatrix} \quad (14)$$

```
> Vtm2 := SubMatrix(Vt, 1..2, 1..5);
```

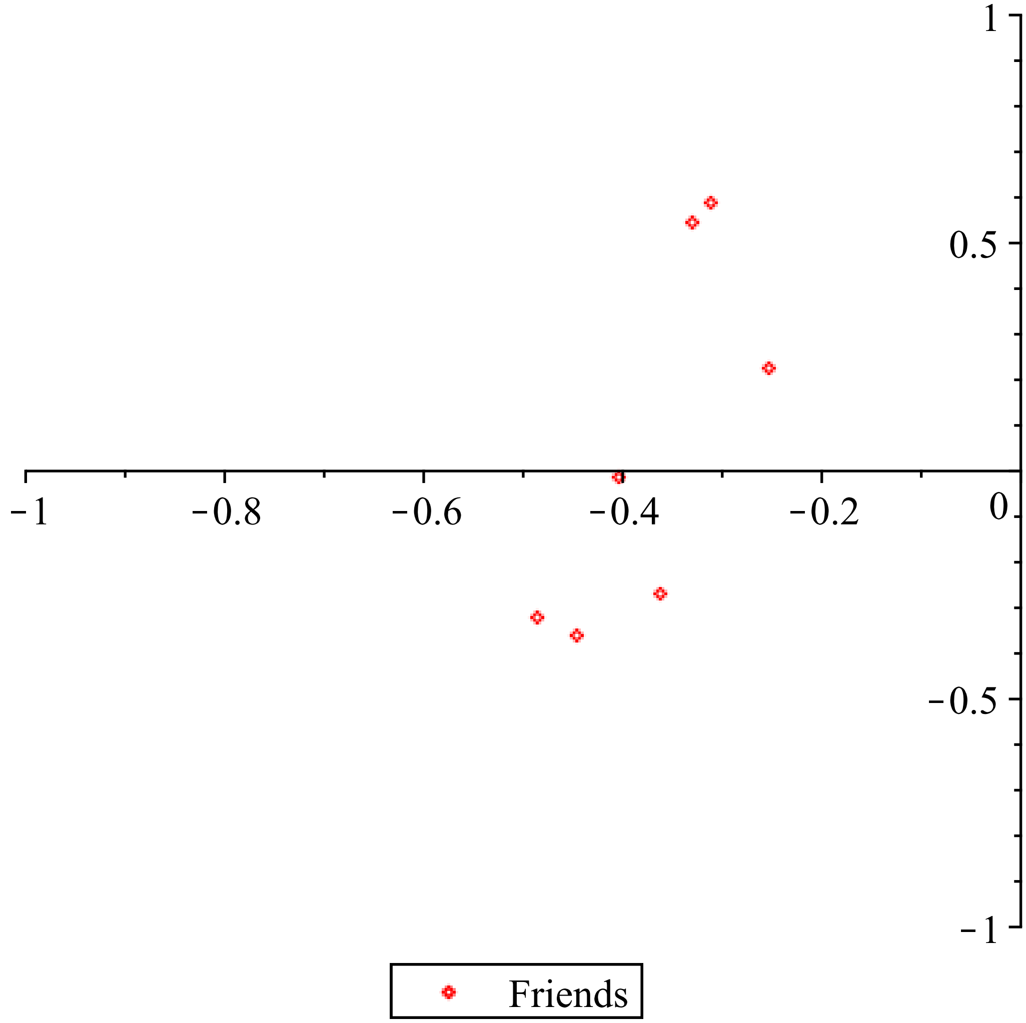
$Vtm2 := [[-0.606216397507475, -0.403976561764470, -0.461086214948247, -0.403029141749247, -0.307036854829483], [-0.398789642209649, -0.122156417445093, -0.307746620017512, 0.698692140974220, 0.493118587108556]]$ (15)

****

(a)

Use the rows of the U72 matrix to associate 2D coordinates to the seven viewers and plot them in the plane. Is there a natural way to separate them into two distinct groups? Comment on your classification.
****

> *plot_friends* := *pointplot*(*Um2*, *legend* = "Friends", *color* = *red*, *axes* = *normal*, *view* = [-1 ..0, -1 ..1]) : *display*(*plot_friends*)



Two distinct groups are :

Group_ 1 : People {Anna, Ghris, Jane} => who avoring Romantic Movies = {Love Story, Titanic}

Group_ 2 : *People*{*Bob*, Dan, Ella, George} ⇒ who favoring Scientific Fiction Movies = {Matrix, Alien, Star Wars}

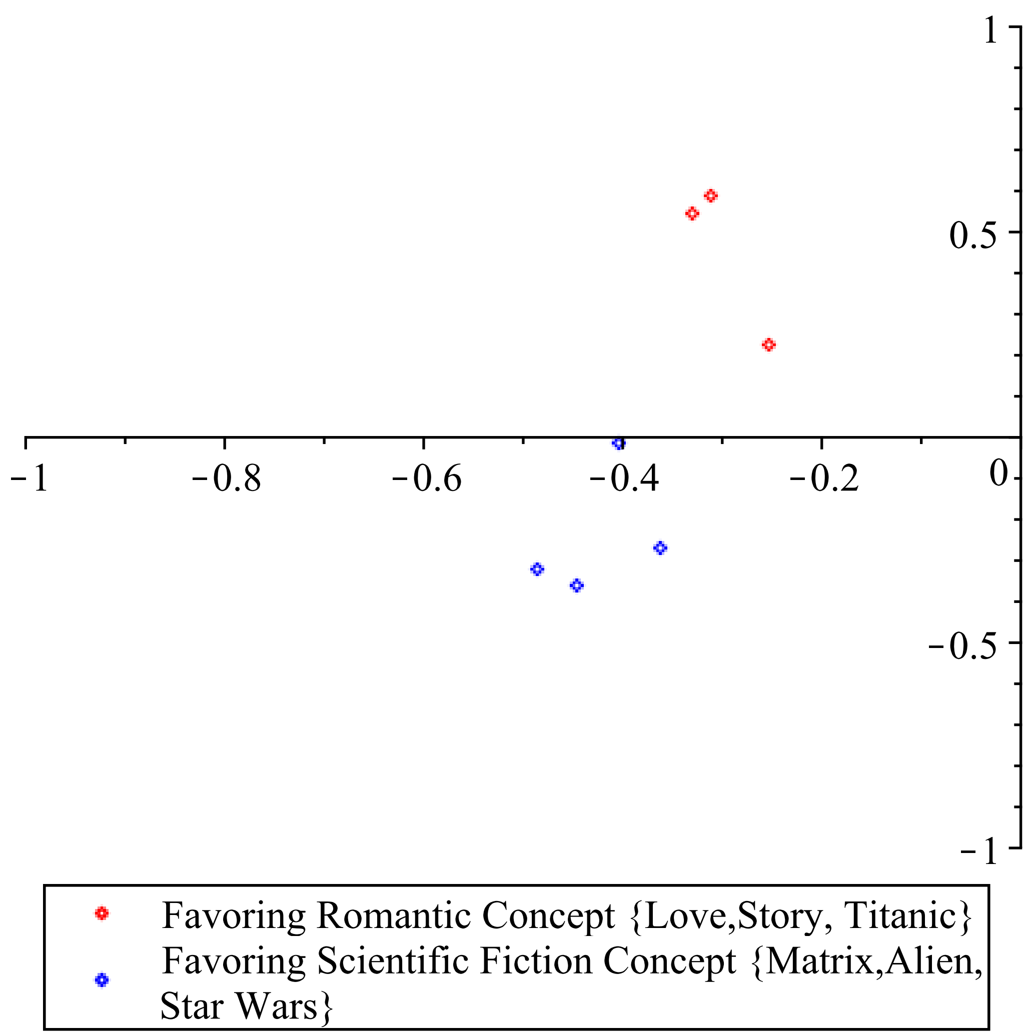
> *Group_1* := ⟨*Um2*[1], *Um2*[3], *Um2*[6]⟩; *Group_2* := ⟨*Um2*[2], *Um2*[4], *Um2*[5], *Um2*[7]⟩;

$$Group\_1 := \begin{bmatrix} -0.311391548098427 & 0.588473218374909 \\ -0.330102422290805 & 0.544795306481915 \\ -0.253294034085323 & 0.225519366390527 \end{bmatrix}$$

$$Group\_2 := \begin{bmatrix} -0.485991620343302 & -0.321003851879510 \\ -0.446301804770097 & -0.361055696634442 \\ -0.404058030824304 & -0.0137140305343523 \\ -0.362317234766473 & -0.269157197649266 \end{bmatrix} \quad (16)$$

> *plot_group_1* := *pointplot*(*Group_1*, *legend* = "Favoring Romantic Concept {Love,Story, Titanic}", *color* = *red*, *axes* = *normal*, *view* = [-1 ..0, -1 ..1]) :

> *plot_group_2* := *pointplot*(*Group_2*, *legend* = "Favoring Scientific Fiction Concept {Matrix,Alien,Star Wars}", *color* = *blue*, *axes* = *normal*, *view* = [-1 ..0, -1 ..1]) : *display*(*plot_group_1*, *plot_group_2*)



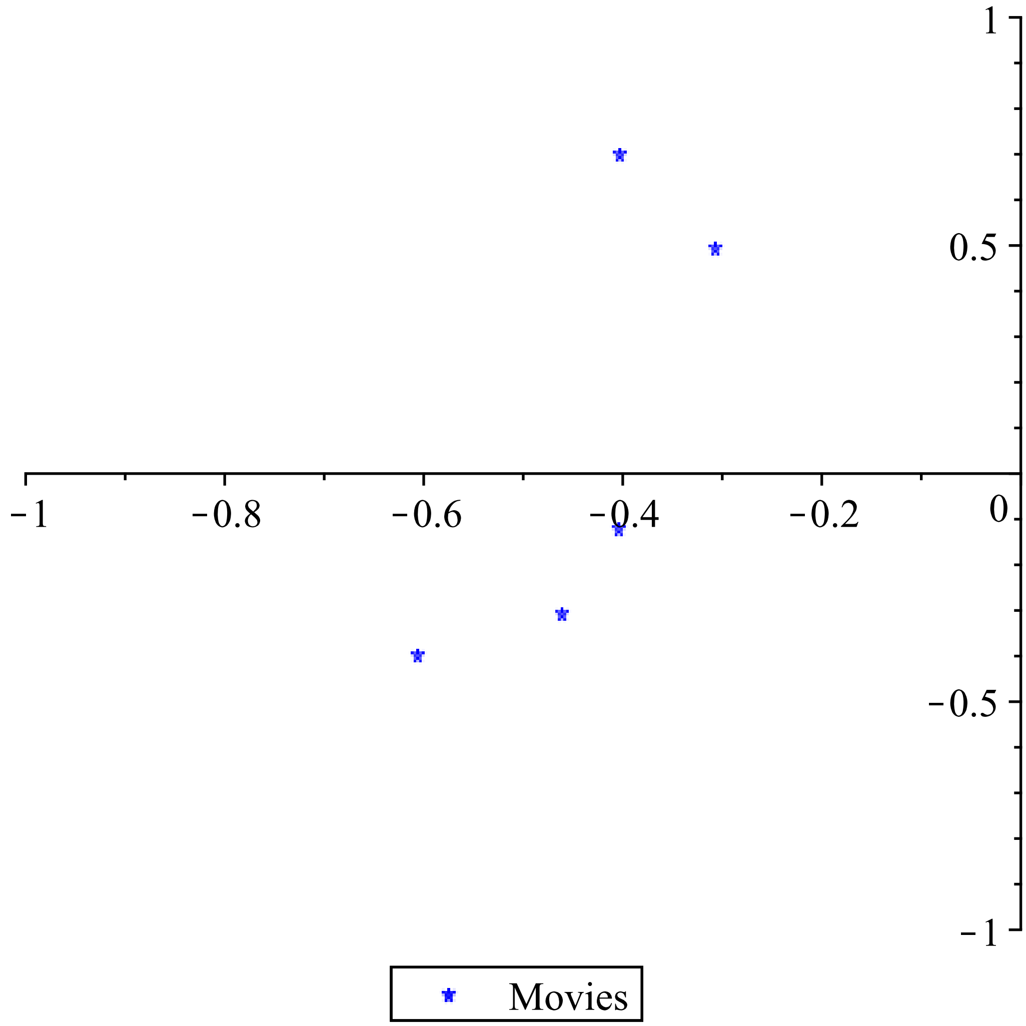
>

****

(b)
Use the columns of the V t
2x5 matrix to associate 2D coordinates to the five movies
and plot them in the plane. Is there a natural way to separate them into two
distinct groups? Comment on your classifcation.

****

> $plot\_movies := pointplot(Vtm2, legend = \text{"Movies"}, color = blue, symbol = asterisk, axes = normal, view = [-1..0, -1..1]) : display(plot\_movies);$



Two distinct groups are :

Group_ 11 : {Romantic Movies} => (fcolumns 1,2 3) Of Matrix Vtm2 that corresponds to movies

Group_ 21 : {Scientific Fiction Movies} ⇒ (*column* 4 and 5)*Of Matrix Vtm2 that corresponds* to *movies*

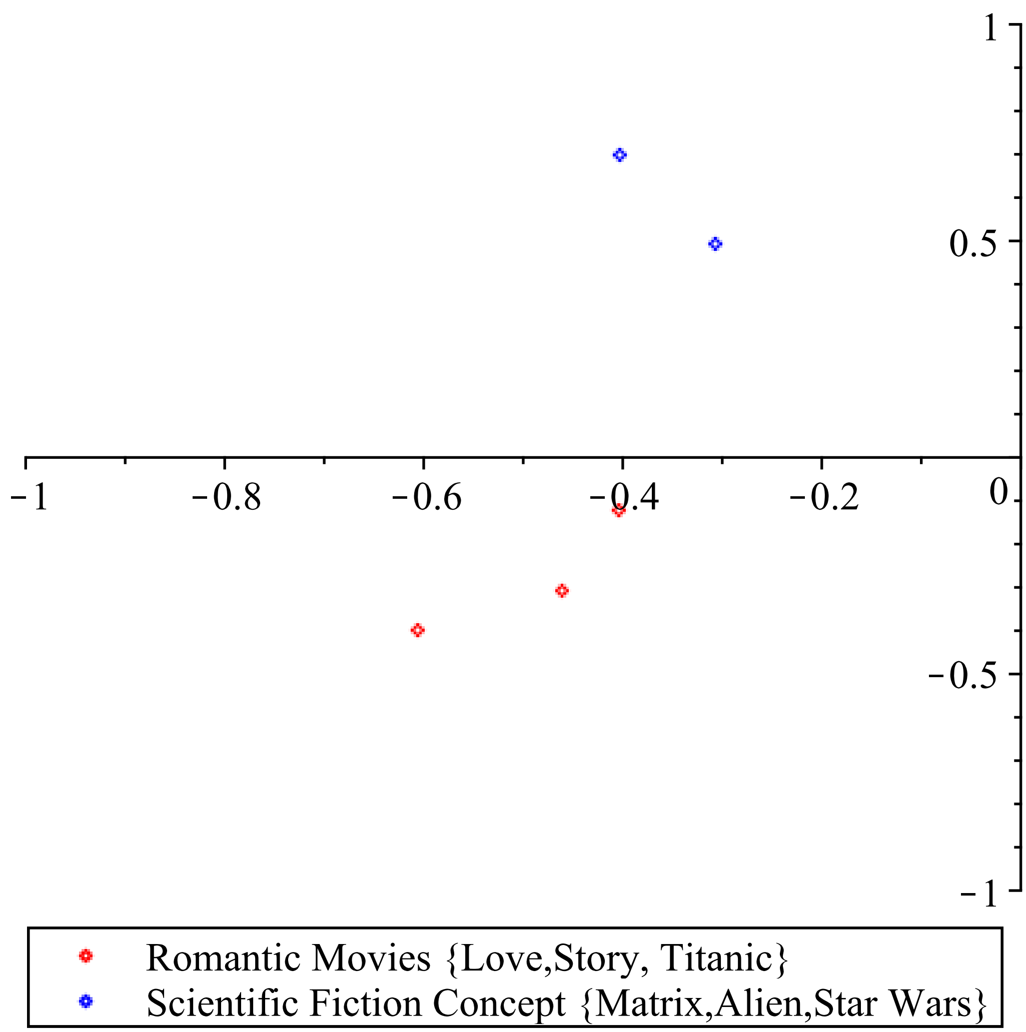
> *Group_11* := *Transpose*(*Vtm2*[1 ..2, 1 ..3]); *Group_21* := *Vtm2*[1 ..2, 4 ..5] : *Group_21* := *Transpose*(*Group_21*)

$$Group\_11 := \begin{bmatrix} -0.606216397507475 & -0.398789642209649 \\ -0.403976561764470 & -0.122156417445093 \\ -0.461086214948247 & -0.307746620017512 \end{bmatrix}$$

$$Group\_21 := \begin{bmatrix} -0.403029141749247 & 0.698692140974220 \\ -0.307036854829483 & 0.493118587108556 \end{bmatrix} \tag{17}$$

> *plot_group_1* := *pointplot*(*Group_11*, *legend* = "Romantic Movies {Love,Story, Titanic}", *color* = *red*, *axes* = *normal*, *view* = [-1 ..0, -1 ..1]) :

> *Group_21*; *plot_group_2* := *pointplot*(*Group_21* , *legend* = "Scientific Fiction Concept {Matrix,Alien,Star Wars}", *color* = *blue*, *axes* = *normal*, *view* = [-1 ..0, -1 ..1]) : *display*(*plot_group_1*, *plot_group_2*)



>

>

****

(c)
Use the rank-2 approximation U7x2 xS2x2 x V t 2x5 to replace the 0 entries with some plausible ratings (round to the nearest integer).
Comment on your fndings.

****

> $with(Statistics):\ with(MTM):\quad Arank\_2 := \mathbf{Um2 \cdot Sm2 \cdot Vtm2};$

$Arank\_2 := [[1.14118757418508, 1.37018328411107, 0.849179708773200, 4.85114095399369, 3.52942933845284], [5.34746534212371, 3.23091085066496, 4.07752192908811, 1.32276736056304, 1.09841862157956], [1.43718585938518, 1.52217963506797, 1.07571070576085, 4.74417719644499, 3.46028368904649], [5.10143042342456, 3.02545858755875, 3.89166837296832, 0.880662131446946, 0.772929908021624], [3.71744896974014, 2.46306204311356, 2.82791777568304, 2.37608897683825, 1.81403423276071], [1.65672316782842, 1.33768064322319, 1.25289208265198, 2.66977509410184, 1.97017236019536], [4.07251881723880, 2.43501665280574, 3.10614507281256, 0.835708003705398, 0.712715631073474]]$ **(18)**

>

$Arank\_2\_rounded := Matrix(7, 5, (i, j) \to \mathrm{round}(Arank\_2[i][j]));$

$$Arank\_2\_rounded := \begin{bmatrix} 1 & 1 & 1 & 5 & 4 \\ 5 & 3 & 4 & 1 & 1 \\ 1 & 2 & 1 & 5 & 3 \\ 5 & 3 & 4 & 1 & 1 \\ 4 & 2 & 3 & 2 & 2 \\ 2 & 1 & 1 & 3 & 2 \\ 4 & 2 & 3 & 1 & 1 \end{bmatrix} \quad \textbf{(19)}$$

> $Energy\_Retained := \mathrm{round}\left(\frac{Norm(Sm2, Frobenius)}{Norm(S, Frobenius)} \cdot 100\right);\ Energy\_Loss := 100 - Energy\_Retained;$

$$Energy\_Retained := 93$$
$$Energy\_Loss := 7 \quad \textbf{(20)}$$

> $Error := \mathrm{round}(evalf((Norm(A - Arank\_2\_rounded, Frobenius))))$

$$Error := 7 \quad \textbf{(21)}$$

If we take rank 2 approximation we will retain 93 percent of energy

which is a good approximation of the properties of the original matrix.
Frobenius of the Approximation Error of the rounded Arank_2 is 7
and equal to the Energy Loss was calculated before. This tells us that
rounded matrix has the same error as Arank_2 and therefore the properties
of the original matrix will be preserved in Approximated *Arank_2_rounded*
Matrix. In Arank_2_rounded we can see 2 revieled clusters better then
in original A Matrix. Also we can see that rank of the matrix revieled

We can say that the data has 2 distinct clusters:
Cluster_1 => one direction => people faivoring Romantic Concept

Cluster_2 => second direction => people faivoring Scientific Fiction Concept

$$\begin{bmatrix} 1 & 1 & 1 & 5 & 4 \\ 5 & 3 & 4 & 1 & 1 \\ 1 & 2 & 1 & 5 & 3 \\ 5 & 3 & 4 & 1 & 1 \\ 4 & 2 & 3 & 2 & 2 \\ 2 & 1 & 1 & 3 & 2 \\ 4 & 2 & 3 & 1 & 1 \end{bmatrix} \tag{22}$$

> $Cluster\_1 := Arank\_2\_rounded[1], Arank\_2\_rounded[3], Arank\_2\_rounded[6]$

$$Cluster\_1 := \begin{bmatrix} 1 & 1 & 1 & 5 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 1 & 5 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 1 & 1 & 3 & 2 \end{bmatrix} \tag{23}$$

> $Cluster\_2 := Arank\_2\_rounded[2], Arank\_2\_rounded[4], Arank\_2\_rounded[5], Arank\_2\_rounded[7]$

$$Cluster\_2 := \begin{bmatrix} 5 & 3 & 4 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 5 & 3 & 4 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 4 & 2 & 3 & 2 & 2 \end{bmatrix}, \begin{bmatrix} 4 & 2 & 3 & 1 & 1 \end{bmatrix} \tag{24}$$

We can identify people who belong to the same cluster or concept.
We can perform queries based on someone rating of those movies
and define what goup of people they belong with good accuracy
Example:
If Person X gives the following rating

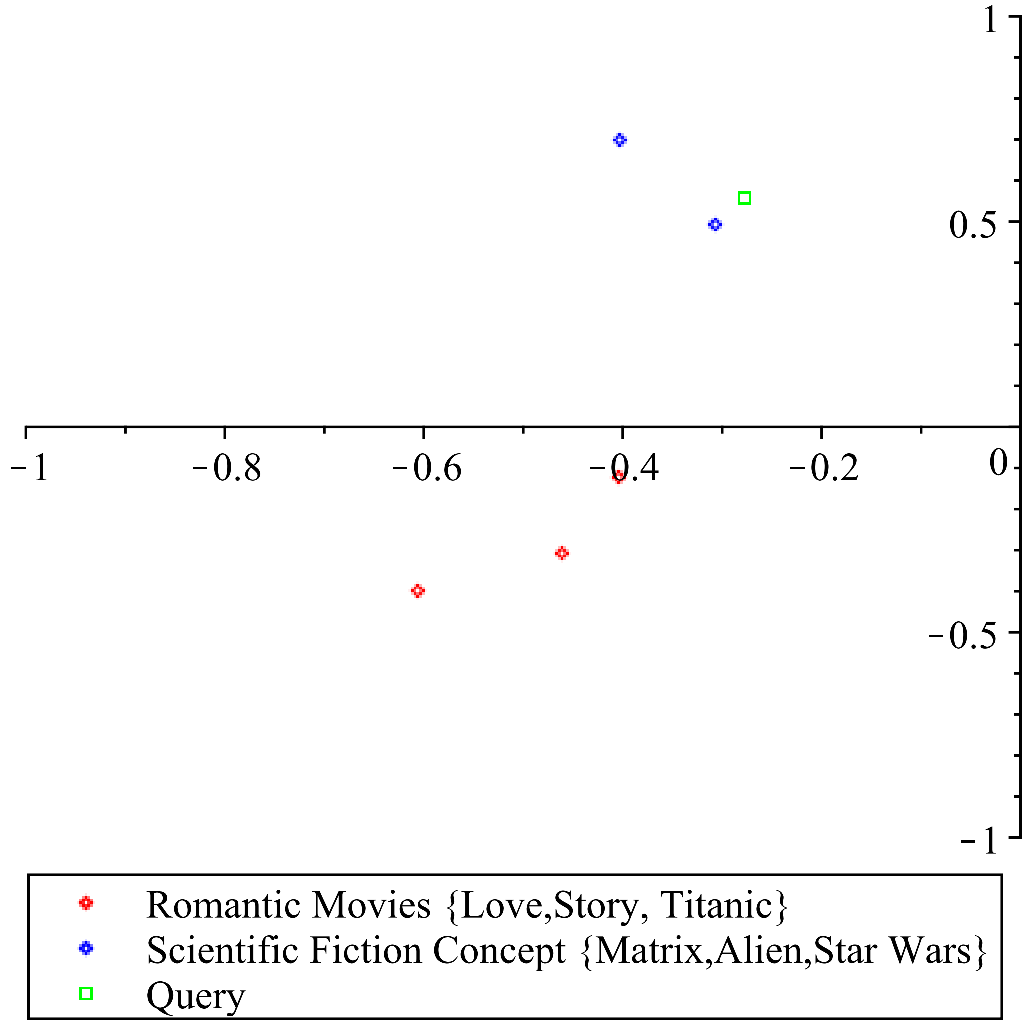
> $q := \langle 0, 1, 2, 4, 4 \rangle;$

$$q := \begin{bmatrix} 0 \\ 1 \\ 2 \\ 4 \\ 4 \end{bmatrix} \tag{25}$$

> $qcoord := Transpose(q).(Transpose(Vtm2).MatrixInverse(Sm2));$

$$qcoord := \begin{bmatrix} -0.277474742593872 & 0.558470043823753 \end{bmatrix} \tag{26}$$

> $plot3 := pointplot(qcoord, symbol = box, legend = \text{"Query"}, color = \text{"green"}, axes = normal) : display(plot\_group\_1, plot\_group\_2, plot3)$



When we plot the 2 clusters and the query element we can see that the element of the query belog to the cluster of the group 2=> Romantic movies

Computing cosine distance

> $person1 := Row(Um2, 1); test1 := \frac{qcoord \cdot person1}{Norm(qcoord, 2) \cdot Norm(person1, 2)}$

$$person1 := \begin{bmatrix} -0.311391548098427 & 0.588473218374909 \end{bmatrix}$$

$$test1 := 0.999673024813634 \quad (27)$$

> $person2 := Row(Um2, 2); test2 := \frac{qcoord \cdot person2}{Norm(qcoord, 2) \cdot Norm(person2, 2)}$

$$person2 := \begin{bmatrix} -0.485991620343302 & -0.321003851879510 \end{bmatrix}$$

$$test2 := -0.122300551618460 \quad (28)$$

> $person3 := Row(Um2, 3); test3 := \frac{qcoord \cdot person3}{Norm(qcoord, 2) \cdot Norm(person3, 2)}$

$$person3 := \begin{bmatrix} -0.330102422290805 & 0.544795306481915 \end{bmatrix}$$

$$test3 := 0.996504357307137 \quad (29)$$

> $person4 := Row(Um2, 4); test4 := \frac{qcoord \cdot person4}{Norm(qcoord, 2) \cdot Norm(person4, 2)}$

$$person4 := \begin{bmatrix} -0.446301804770097 & -0.361055696634442 \end{bmatrix}$$

$$test4 := -0.217330252522953 \quad (30)$$

> $person5 := Row(Um2, 5); test5 := \frac{qcoord \cdot person5}{Norm(qcoord, 2) \cdot Norm(person5, 2)}$

$$person5 := \begin{bmatrix} -0.404058030824304 & -0.0137140305343523 \end{bmatrix}$$

$$test5 := 0.414319640985456 \quad (31)$$

> $person6 := Row(Um2, 6); test6 := \frac{qcoord \cdot person6}{Norm(qcoord, 2) \cdot Norm(person6, 2)}$

$$person6 := \begin{bmatrix} -0.253294034085323 & 0.225519366390527 \end{bmatrix}$$

$$test6 := 0.927839775538363 \quad (32)$$

> $person7 := Row(Um2, 7); test4 := \frac{qcoord \cdot person7}{Norm(qcoord, 2) \cdot Norm(person7, 2)}$

$$person7 := \begin{bmatrix} -0.362317234766473 & -0.269157197649266 \end{bmatrix}$$

$$test4 := -0.176868647588548 \quad (33)$$

We can see that the person X belong to the group 1 cluster [Person1, Person3,Person6]
Using Arank_2_rounded approximation we can predict using the score calculated to what space person X would be mapped to.

> $\boldsymbol{Rating := Transpose(q).Transpose(Arank\_2\_rounded)}$

$$Rating := \begin{bmatrix} 39 & 19 & 36 & 19 & 24 & 23 & 16 \end{bmatrix} \quad (34)$$

The Highest Score is for Person1 and Person3
The Rank 2 approximation matrix classifies the person X the most closest to person1 and person3 who belong to the ones who rated high Romantic Movies.
We can see that in this case rank 2 is very good 2 dimentional fit (2 dimentional least square approximation of the original Matrix A) and can be used to predict to what of the 2 spaces the unknown person are coled to based on this person ratings

##

Problem 4

##

The table below lists the closing Dow Jones Industrial Averages the first day the market was open for the months from March 2013 to February 2014.

##

(b)

Use Maple to construct the order 12 trigonometric interpolation polynomial (in reduced form). (Use time unit=1 year, so the interval is [0,1])

##

```
> restart; with(DiscreteTransforms) : with(LinearAlgebra) :
  with(plots) : with(MTM) : interface(rtablesize = 12); V := Vector(12, [14090, 14573, 14701,
      15254, 14975, 15628, 14834, 15193, 15616, 16009, 16441, 15373])
```

$$10$$

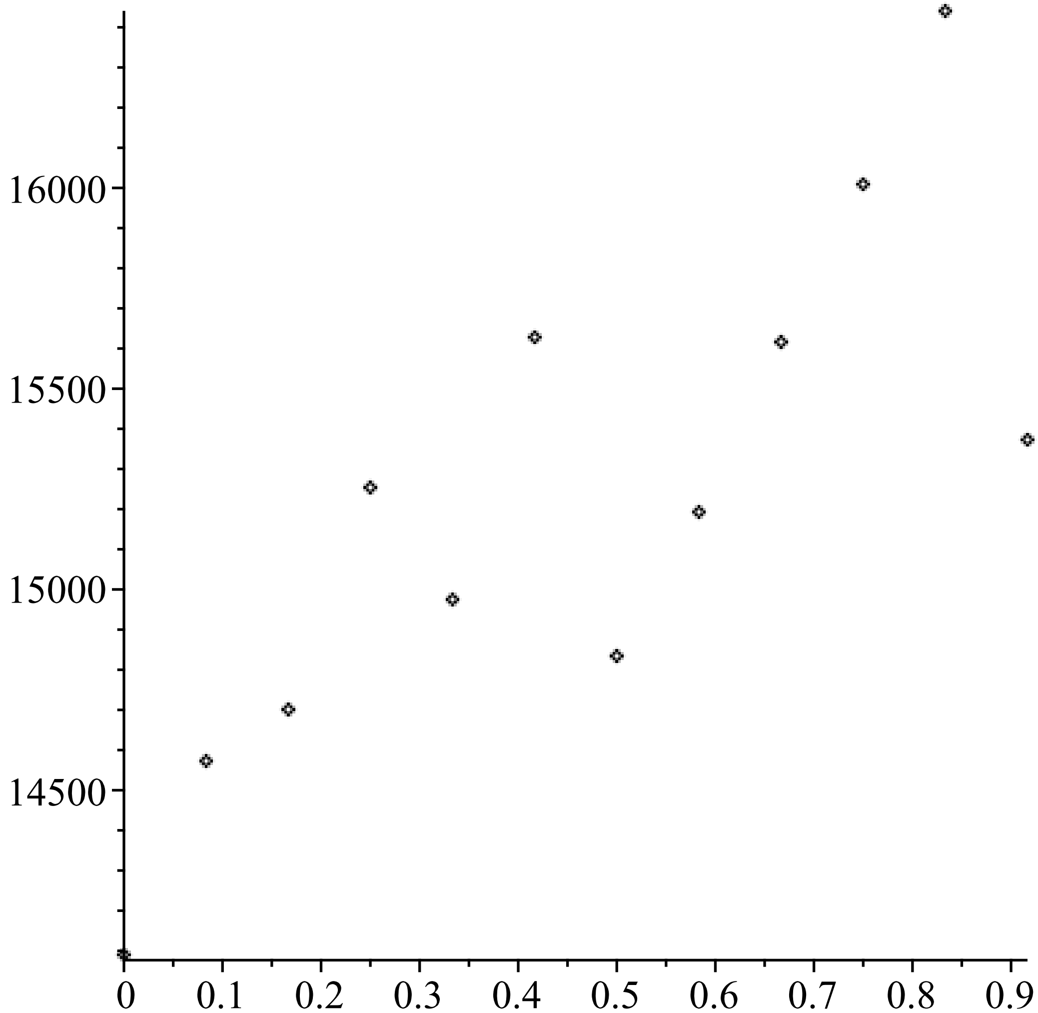
$$V := \begin{bmatrix} 14090 \\ 14573 \\ 14701 \\ 15254 \\ 14975 \\ 15628 \\ 14834 \\ 15193 \\ 15616 \\ 16009 \\ 16441 \\ 15373 \end{bmatrix} \quad \textbf{(35)}$$

>

> $rule := i \to \frac{(i-1)}{12} : T := Vector(12, rule);$

$$T := \begin{bmatrix} 0 \\ \frac{1}{12} \\ \frac{1}{6} \\ \frac{1}{4} \\ \frac{1}{3} \\ \frac{5}{12} \\ \frac{1}{2} \\ \frac{7}{12} \\ \frac{2}{3} \\ \frac{3}{4} \\ \frac{5}{6} \\ \frac{11}{12} \end{bmatrix} \tag{36}$$

> $dataplot1 := pointplot(T, V) : display(dataplot1);$



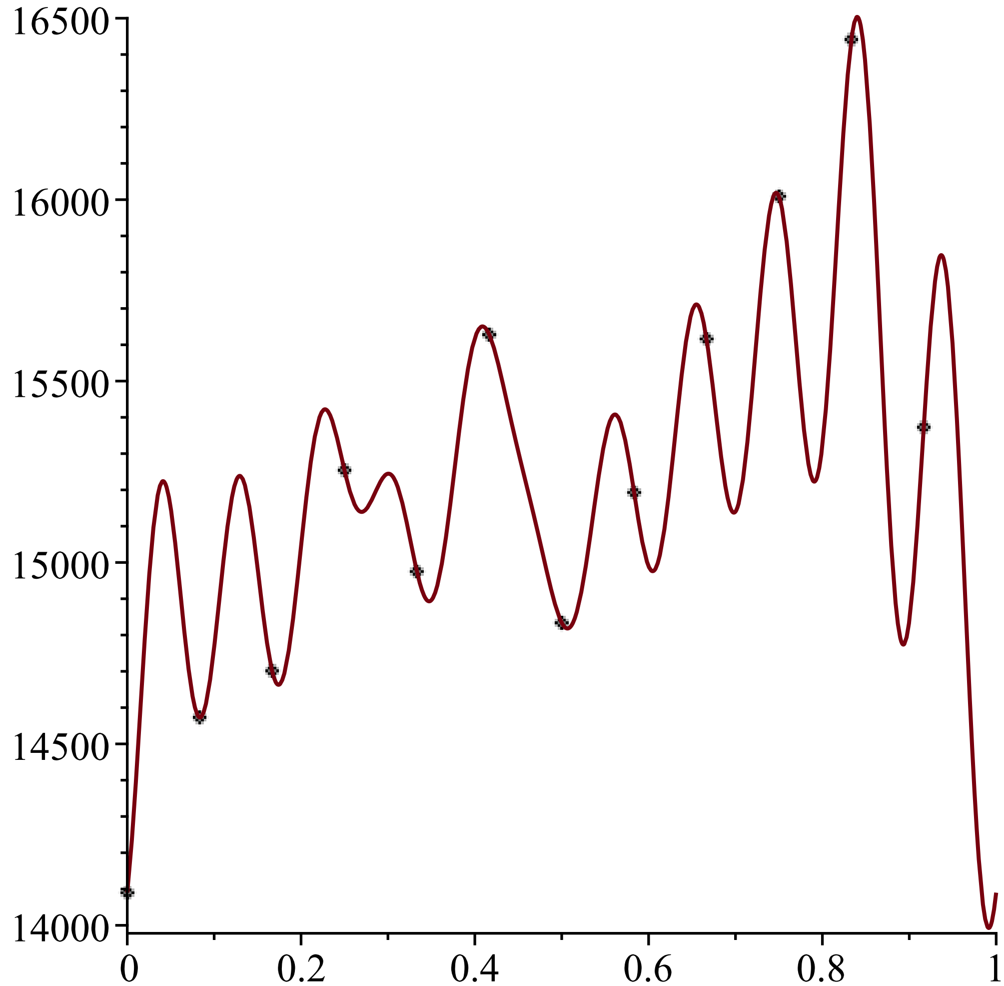
> $Z := \mathit{FourierTransform}(V)$

$$Z := \begin{bmatrix} 52737.1943137226 + 0.\,\mathrm{I} \\ -353.994300557670 + 865.882938682637\,\mathrm{I} \\ -814.641229826562 + 583.500000000000\,\mathrm{I} \\ -373.834299300283 - 112.583302491977\,\mathrm{I} \\ -306.861668074286 + 34.0000000000001\,\mathrm{I} \\ 83.5056994423302 - 324.617061317363\,\mathrm{I} \\ -396.350959798678 + 0.\,\mathrm{I} \\ 83.5056994423302 + 324.617061317363\,\mathrm{I} \\ -306.861668074286 - 33.9999999999999\,\mathrm{I} \\ -373.834299300283 + 112.583302491977\,\mathrm{I} \\ -814.641229826562 - 583.500000000000\,\mathrm{I} \\ -353.994300557670 - 865.882938682637\,\mathrm{I} \end{bmatrix} \tag{37}$$

```
> P := t → evalf( 1/sqrt(12) · add(Re(Z[k])·cos(2·(k−1)·π·t) − Im(Z[k])·sin(2·(k−1)·π·t), k = 1..12) );
```

$$P := t \mapsto evalf\left(\frac{add\left(\Re(Z_k)\cos((2k-2)\pi t) - \Im(Z_k)\sin((2k-2)\pi t), k=1..12\right)}{\sqrt{12}}\right) \quad \textbf{(38)}$$

```
> plot1 := plot(P, 0..1) : display(dataplot1, plot1)
```




```
> a := Re(Z) : b := Im(Z) :
> n := 12 : a := 2· Re(Z[1 .. n/2 + 1]) : b := 2· Im(Z[1 .. n/2 + 1]) : a[1] := Re(Z[1]) :
      a[7] := Re(Z[7]) : a, b
```

(39)

$$\left[\begin{array}{c} 52737.1943137226 \\ -707.988601115340 \\ -1629.28245965312 \\ -747.668598600565 \\ -613.723336148572 \\ 167.011398884660 \\ -396.350959798678 \end{array}\right], \left[\begin{array}{c} 0. \\ 1731.76587736527 \\ 1167.00000000000 \\ -225.166604983954 \\ 68.0000000000002 \\ -649.234122634726 \\ 0. \end{array}\right] \tag{39}$$

> $P := (k, t) \to \left(a[k]\cdot\cos(2\cdot\pi\cdot(k-1)\cdot t) - b[k]\cdot\sin(2\cdot\pi\cdot(k-1)\cdot t)\right)$; $P7 := t \to a[7]\cdot\cos(n\cdot t)$;

$$P := (k, t) \mapsto a_k \cos(2\,\pi\,(k-1)\,t) - b_k \sin(2\,\pi\,(k-1)\,t)$$
$$P7 := t \mapsto a_7 \cos(n\,t) \tag{40}$$

> $P12\_t := t \to 12^{-\frac{1}{2}}\cdot(P(1, t) + P(2, t) + P(3, t) + P(4, t) + P(5, t) + P(6, t) + P7(t))$:

> $Check\_If\_Correct := Vector(12, (i) \to evalf(P12\_t(T[i])))$;

$$Check\_If\_Correct := \left[\begin{array}{c} 14089.9999994397 \\ 14396.7637432203 \\ 14863.0307991408 \\ 15252.8549736027 \\ 15164.2043896686 \\ 15481.1276505055 \\ 14838.5571818465 \\ 14992.3243491856 \\ 15747.0642943576 \\ 15998.8318195229 \\ 16651.4204330095 \\ 15258.0769585272 \end{array}\right] \tag{41}$$

Answer:
order 12 trigonometric interpolation polynomial:

> $\mathbf{P12 := evalf\left(12^{-\frac{1}{2}}\cdot(P(1,t) + P(2,t) + P(3,t) + P(4,t) + P(5,t) + P(6,t) + P7(t))\right)}$

$$\begin{aligned} P12 := {} & 15223.9166654523 - 204.378704702261\cos(6.283185308\,t) \\ & - 499.917747695248\sin(6.283185308\,t) - 470.333333295817\cos(12.56637062\,t) \\ & - 336.883882045275\sin(12.56637062\,t) - 215.833333316117\cos(18.84955592\,t) \\ & + 64.9999999948152\sin(18.84955592\,t) - 177.166666652535\cos(25.13274123\,t) \\ & - 19.6299091508815\sin(25.13274123\,t) + 48.2120380480516\cos(31.41592654\,t) \\ & + 187.417747720175\sin(31.41592654\,t) - 114.416666657540\cos(12.\,t) \end{aligned} \tag{42}$$

>

##
(c)
Use Maple to construct the order 4 least square trigonometric approximation (in reduced form). Compare the estimated value at 09/13 with the actual value given.
##

> $Pdeg4 := t \rightarrow evalf\left(\frac{1}{\mathrm{sqrt}(12)} \cdot add\left(\mathrm{Re}(Z[k]) \cdot \cos\left(2 \cdot (k-1) \cdot \pi \cdot t\right) - \mathrm{Im}(Z[k]) \cdot \sin\left(2 \cdot (k-1) \cdot \pi \cdot t\right), k = 1..4\right)\right):$

> $IndustrialAverage\_09\_13 := Pdeg4\left(\frac{6}{12}\right);$

$$IndustrialAverage\_09\_13 := 15198.8560228091 \quad \textbf{(43)}$$

> $Actual\_Error := \mathrm{abs}(V[7] - IndustrialAverage\_09\_13); Relative\_Error\_\% := \frac{Actual\_Error}{V[7]} \cdot 100;$

$$Actual\_Error := 364.856022809076$$
$$Relative\_Error\_\% := 2.45959298105080 \quad \textbf{(44)}$$

##
Answer:

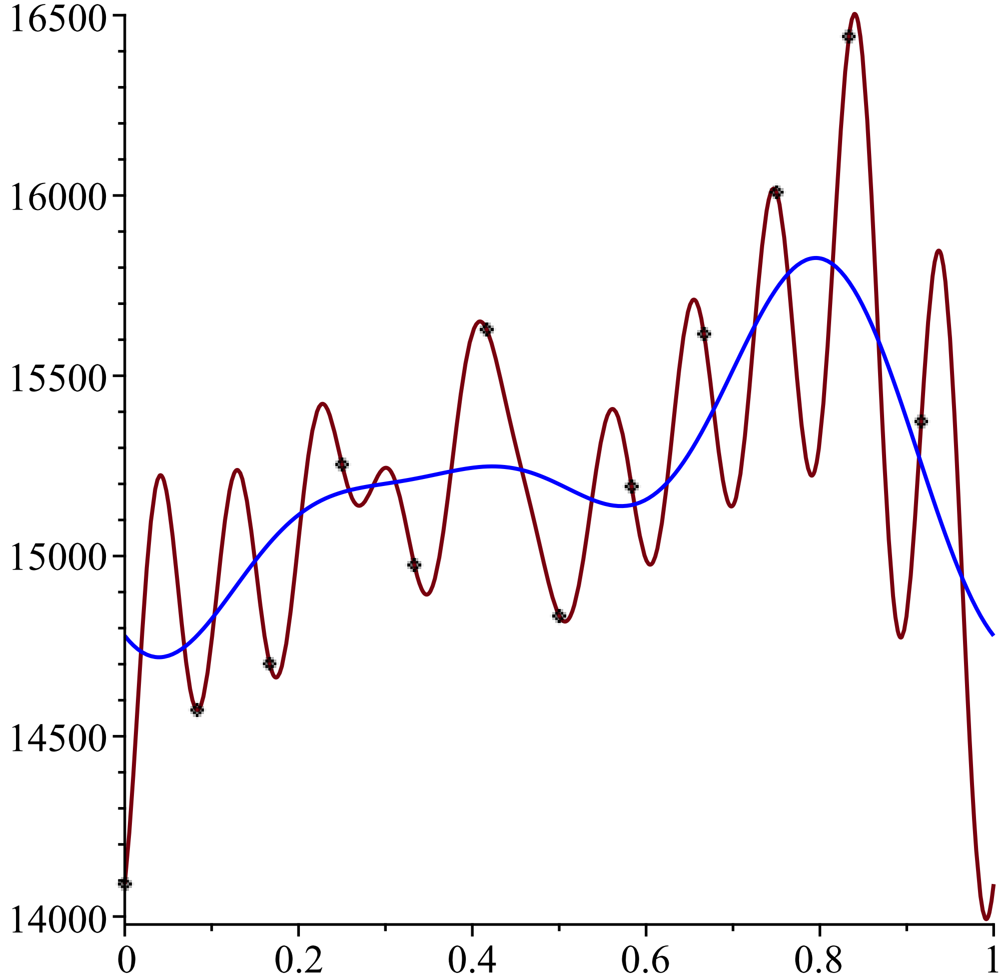
$\boldsymbol{IndustrialAverage\_09\_13 := 15198.86}$
$\boldsymbol{Actual\_Error := 364.86}$
$\boldsymbol{Relative\_Error := 2.46\ \%}$

###

(d) On one plot show the data, the graph from part (b) and the one from part (c)

###

> $plot2 := plot(Pdeg4, 0..1, color = blue) : display(plot1, plot2, dataplot1)$



>

##

>