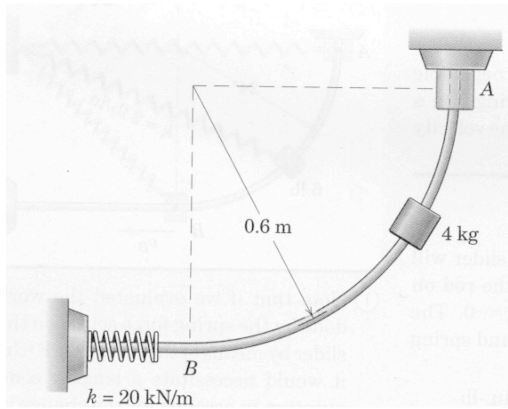
Session S5
Unité APP 2

Principes de dynamique et méthodes numériques

Évaluation formative solutionnaire
GEN441, GEL450 et GIF590

Département de génie électrique et de génie informatique
Faculté de génie
Université de Sherbrooke

Problème no 1
Le bloc de 4 kg représenté à la figure ci-dessous est relâché d'un état de repos au point A et glisse sans frottement, par la suite, sur une tige en arc de cercle (dans un plan vertical). À l'aide du théorème de l'énergie, trouver la vitesse du bloc au point B et l'écrasement maximal du ressort.



Solution

Théorème de l'énergie : $U'_{1\to 2} = \Delta T + \Delta V_g + \Delta V_e$

Entre A et B : $U'_{A\to B} = T_B - T_A + \Delta V_{gB} - \Delta V_{gA}$ Pas de ressort impliqué entre A et B

Avec : $U'_{A\to B} = 0,\ T_A = 0,\ T_B = \frac{1}{2}m{v_B}^2,\ V_{gA} = 0,\ V_{gB} = -0.6mg$

$$0 = \frac{1}{2}m{v_B}^2 - 0.6mg \Rightarrow \frac{1}{2}m{v_B}^2 = 0.6mg$$

$$v_B = \sqrt{1.2g}$$

Vitesse au point B : $v_B = 3.43\ m/s$

Théorème de l'énergie: $U'_{1\to 2} = \Delta T + \Delta V_g + \Delta V_e$

Entre A et C : $U'_{A\to C} = T_C - T_A + \Delta V_{gC} - \Delta V_{gA} + \Delta V_{eC} - \Delta V_{eA}$ Écrasement max du ressort au point C

Avec : $U'_{A\to C} = 0,\ T_A = T_C = 0,\ V_{gA} = 0,\ V_{gC} = -0.6mg, V_{eA} = 0,\ V_{eC} = \frac{1}{2}k{x_C}^2,\ k =$ 20kN

$$0 = -0.6mg + \frac{1}{2}k{x_C}^2 \Rightarrow 0.6mg = \frac{1}{2}k{x_C}^2$$

$$x_C = \sqrt{\frac{1.2mg}{k}}$$

Écrasement du ressort : $x_C = 0.0485\ m$

Autre solution possible :

T. de l'énergie : $U'_{1\to 2} = \Delta T + \Delta V_g + \Delta V_e$

Entre B et C : $U'_{A\to C} = T_C - T_B + \Delta V_{gC} - \Delta V_{gB} + \Delta V_{eC} - \Delta V_{eB}$ Écrasement max du ressort au point C

Avec : $U'_{A\to C} = 0,\ T_B = 3.43\frac{m}{s},\ T_C = 0,\ V_{gB} = V_{gC} = 0, V_{eA} = 0,\ V_{eC} = \frac{1}{2}k{x_C}^2,\ k = 20\text{kN}$

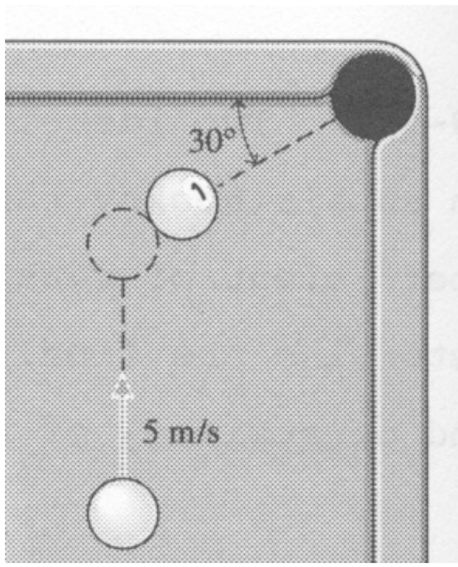
$$0 = -\frac{1}{2}m{v_B}^2 + \frac{1}{2}k{x_C}^2 \Rightarrow m{v_B}^2 = k{x_C}^2$$

$$x_C = \sqrt{\frac{m{v_B}^2}{k}}$$

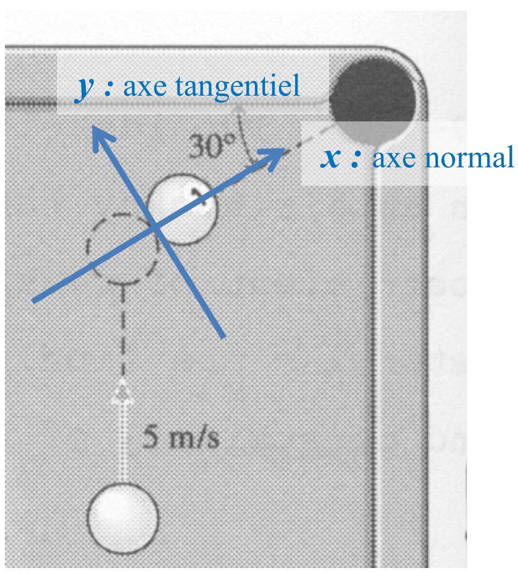
Écrasement du ressort : $x_C = 0.0485\ m$

Problème no 2

Soit le coup de billard représenté à la figure ci-dessous. La boule no 1 (vitesse initiale nulle) est envoyée dans la poche du coin par la boule blanche (vitesse initiale : 5 m/s). Si le coefficient de restitution, *e*, vaut 0.95, trouver le module du vecteur vitesse de la boule blanche tout juste après la collision.



Solution



Balle blanche = B, Balle #1 = #1

$v_{\#\ Initiale} = 5 \Rightarrow v_{B\ Initiale\ x} = 5\ sin30 = 2.5,\ \ v_{B\ Initiale\ y} = 5cos30 = 4.33$

$v_{\#1\ Initiale} = 0 \Rightarrow v_{\#1\ Initiale\ x} = 0,\ \ v_{\#1\ Initiale\ y} = 0$

Selon l'axe tangentiel *(en y)*:

$$v_{B\ Initiale\ y} = v_{B\ Final\ y}$$
$$v_{B\ Final\ y} = 5cos30 = 4.33$$

$$v_{\#1\ Initiale\ y} = v_{\#1\ Final\ y}$$
$$v_{\#1\ Final\ y} = 0$$

Selon l'axe normal *(en x)* :

Système à deux équations avec deux inconnues

Équation 1 :

$$m_B\, v_{B\ Initiale\ x} + m_{\#1}\, v_{\#1\ Initiale\ x} = m_B\, v_{B\ Final\ x} + m_{\#1}\, v_{\#1\ Final\ x}$$

En divisant par $m_B = m_{\#1}$

$$v_{B\ Initiale\ x} + v_{\#1\ Initiale\ x} = v_{B\ Final\ x} + v_{\#1\ Final\ x}$$
$$2.5 + 0 = v_{B\ Final\ x} + v_{\#1\ Final\ x}$$
$$\boxed{2.5 - v_{B\ Final\ x}} = v_{\#1\ Final\ x}$$

Équation 2 :

$$e = \frac{v_{\#1\ Final\ x} - v_{B\ Final\ x}}{v_{B\ Initiale\ x} - v_{\#1\ Initiale\ x}}$$
$$0.95 = \frac{v_{\#1\ Final\ x} - v_{B\ Final\ x}}{2.5 - 0}$$

Équation 1 dans équation 2 :

$$0.95 = \frac{\boxed{2.5 - v_{B\ Final\ x}} - v_{B\ Final\ x}}{2.5}$$
$$v_{B\ Final\ x} \frac{-2.5 * 0.95 - 2.5}{2} = 0{,}0625\ m/s$$
$$v_{\#1\ Final\ x} = 2.5 - v_{B\ Final\ x} = 2.5 - 0{,}0625 = 2{,}4375\ m/s$$

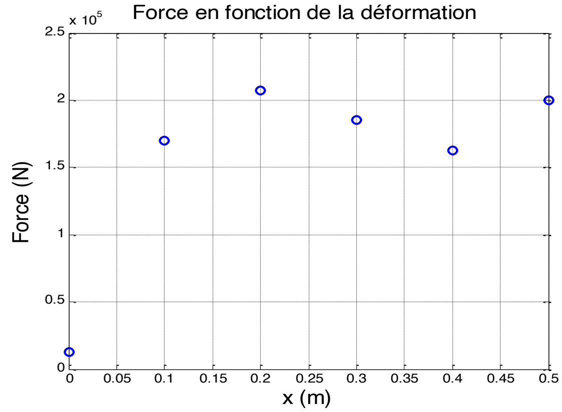
Module de la vitesse finale de la balle blanche :

$$v_{B\ Final} = \sqrt{{v_{B\ Final\ x}}^2 + {v_{B\ Final\ y}}^2} = \sqrt{0.0625^2 + 4.33^2} = 4.33\ m/s$$

Problème no 3

Dans un test de collision automobile contre un mur, on vérifie l'efficacité de l'amortissement du choc en mesurant la force d'impact $F(x)$ en fonction de la déformation x. On suppose une collision plastique.

x (m)	0.0	0.1	0.2	0.3	0.4	0.5
F (kN)	12.5	170	207.5	185	162.5	200



La masse de la voiture est de 1687.5 kg. La vitesse de l'automobile est nulle à la fin de course à $x_f = 0.5$ m. Des mesures très précises de la force de déformation sont indiquées dans le tableau ci-dessus. Utiliser une approximation ou une interpolation polynomiale de la forme $F(x) = \sum_{m=1}^{M} a_m \, x^{m-1}$ pour trouver une expression analytique de la force $F(x)$ en fonction de la distance de déformation x. Faire varier le nombre de coefficients M de sa valeur d'interpolation à une valeur d'approximation adaptée au problème pour trouver la meilleure valeur de M.

(a) Quel est le nombre optimal de coefficients M ?

(b) Quelle méthode entre l'interpolation et l'approximation est la meilleure à prendre dans ce problème? Calculer les erreurs RMS et les corrélations dans chaque cas pour supporter vos conclusions.

(c) Le travail de $F(x)$ fait pendant la collision est de 84,375 kJ.

  a. Où est allée cette énergie?

  b. Calculer la vitesse de l'automobile juste avant l'impact.

(d) Au lieu de se déformer de façon plastique, supposons maintenant que le parechoc est un ressort linéaire de rigidité k. Si la déformation élastique du parechoc ne doit pas excéder 15 cm avec cette même vitesse initiale, quelle devrait être la valeur de la rigidité k du parechoc?

(e) Si maintenant l'impact avec le mur (qui est infiniment plus massif que la voiture) est une collision avec un coefficient de restitution $e = 0.2$, quelle serait la vitesse de la voiture après l'impact? Quelle serait la perte d'énergie pendant l'impact. Où est allée cette énergie?

Mise à jour JdeL 17 mai 2014

Problème no 3

Interpolation vs approximation polynomiale :

- Il y a $N = 6$ points (x_i, y_i) de mesures, donc :
  - pour interpolation, on utilise un polynôme d'ordre 5 avec $M = N = 6$ coefficients à déterminer
  - pour une approximation adaptée, on utilise un polynôme d'ordre 2 avec $M = N - 3 = 3$ coefficients
  - on fait donc des approximations d'ordre 2, 3, 4, 5 avec $M = 3, 4, 5, 6$ coefficients.
- Code MATLAB

```
% Formatif S4-E14
% JdeL 17 mai 2014

% Interpolation et approximation polynômiale

% y = a1 + a2 x + a3 x^2 + a4 x^3 + a5 x^4 + a6 x^5

% Fonctions de base évaluées aux points de mesures

  phi(:,1) = ones(size(xi));
  phi(:,2) = xi;
  phi(:,3) = xi.^2;
  phi(:,4) = xi.^3;
  phi(:,5) = xi.^4;
  phi(:,6) = xi.^5;

% Matrices P pour les approximations de M=3 à M=6 coefficients

  P2 = [phi(:,1), phi(:,2), phi(:,3)];
  P3 = [phi(:,1), phi(:,2), phi(:,3), phi(:,4)];
  P4 = [phi(:,1), phi(:,2), phi(:,3), phi(:,4), phi(:,5)];
  P5 = [phi(:,1), phi(:,2), phi(:,3), phi(:,4), phi(:,5), phi(:,6)];

% Calcul des coefficients

  coef2 = pinv(P2)*yi;
  coef3 = pinv(P3)*yi;
  coef4 = pinv(P4)*yi;
  coef5 = pinv(P5)*yi;

% Dans le cas de l'interpolation, la pseudo-inverse et l'inverse doivent
% donner le même résultat. On vérifie.

  coef5_chk = inv(P5)*yi;

  disp(' ')
  disp('Interpolation et approximation polynomiale')

  disp(' ')
  disp(['Coefficients ordre 2 (M=3) = ', num2str(coef2')])
  disp(['Coefficients ordre 3 (M=4) = ', num2str(coef3')])
  disp(['Coefficients ordre 4 (M=5) = ', num2str(coef4')])
  disp(['Coefficients ordre 5 (M=6) = ', num2str(coef5')])
  disp(['Coefficients ordre 5 (M=6) = ', num2str(coef5_chk')])
```

- Résultats sur MATLAB (toujours en rouge)

Interpolation et approximation polynomiale

Coefficients ordre 2 (M=3) = 42500	**1005000**	**-1500000**			
Coefficients ordre 3 (M=4) = 12500	**2375000**	**-9000000**	**10000000**		
Coefficients ordre 4 (M=5) = 12500	**2375000**	**-9000000**	**10000000**	**1.4901161194e-08**	
Coefficients ordre 5 (M=6) = 12500	**2375000**	**-9000000**	**10000000**	**2.5033950806e-06**	**-1.9967556e-06**
Coefficients ordre 5 (M=6) = 12500	**2375000**	**-9000000**	**10000000**	**3.8743019104e-07**	**-6.8545341e-07**

- On voit que pour $M = 4, 5, 6$, les coefficients sont essentiellement les mêmes.
- On calcule les erreurs quadratiques, RMS et la corrélation.

```
% Calcul des polynômes aux points de mesures pour calculer l'erreur
% Toujours aligner les lignes de code => truc pour trouver des erreurs

  y2 =                                           ((coef2(3)).*xi+coef2(2)).*xi+coef2(1);
  y3 =                               (((coef3(4)).*xi+coef3(3)).*xi+coef3(2)).*xi+coef3(1);
  y4 =                   ((((coef4(5)).*xi+coef4(4)).*xi+coef4(3)).*xi+coef4(2)).*xi+coef4(1);
  y5 = (((((coef5(6)).*xi+coef5(5)).*xi+coef5(4)).*xi+coef5(3)).*xi+coef5(2)).*xi+coef5(1);

  ym = mean(yi);  % Moyenne des données pour le calcul de la corrélation

% Calcul des erreurs quadratique et RMS et de la corrélation

  E2 = (y2-yi)'*(y2-yi);
  err_rms2 = sqrt( mean((y2-yi).*(y2-yi)) );
  R2 = (y2-ym)'*(y2-ym)/((yi-ym)'*(yi-ym));

  disp(' ')
  disp(['Erreur quadratique ordre 2 (M=3)= ', num2str(E2)])
  disp(['Erreur RMS ordre 2 (M=3)= ', num2str(err_rms2)])
  disp(['Corrélation ordre 2 (M=3)= ', num2str(R2)])

% Etc… pour E3, R3, E4, R4… (pas mis ici pour réduire l'espace)

  E5 = (y5-yi)'*(y5-yi);
  err_rms5 = sqrt( mean((y5-yi).*(y5-yi)) );
  R5 = (y5-ym)'*(y5-ym)/((yi-ym)'*(yi-ym));

  disp(' ')
  disp(['Erreur quadratique ordre 5 (M=6)= ', num2str(E5)])
  disp(['Erreur RMS ordre 5 (M=6)= ', num2str(err_rms5)])
  disp(['Corrélation ordre 5 (M=6)= ', num2str(R5)])
  disp(' ')

  disp(' ')
 disp(['Approximation optimale F(x) = ', num2str(coef3(4)), ' x^3 + (', num2str(coef3(3)), ')
                    x^2 + (', num2str(coef3(2)), ')', ' x + (', num2str(coef3(1)), ')'])
```

- Résultats sur MATLAB

Erreur quadratique ordre 2 (M=3)= 6480000000
Erreur RMS ordre 2 (M=3)= 32863.3535
Corrélation ordre 2 (M=3)= 0.75323

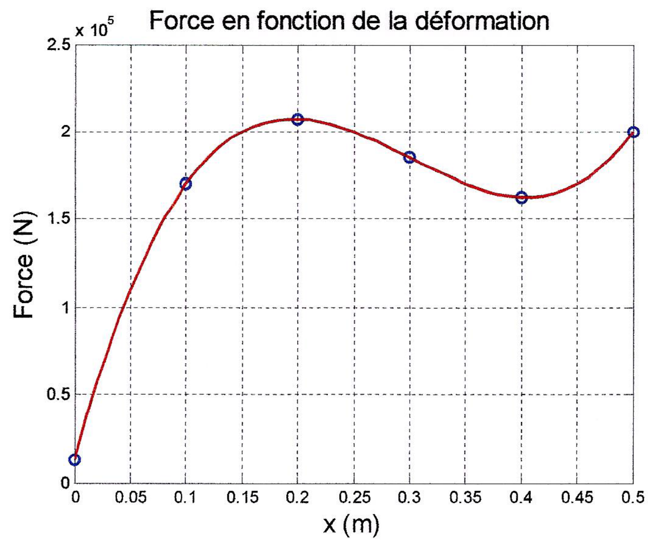
Erreur quadratique ordre 3 (M=4)= 4.8244e-19
Erreur RMS ordre 3 (M=4)= 2.8356e-10
Corrélation ordre 3 (M=4)= 1

Erreur quadratique ordre 4 (M=5)= 1.8183e-17
Erreur RMS ordre 4 (M=5)= 1.7408e-09
Corrélation ordre 4 (M=5)= 1

Erreur quadratique ordre 5 (M=6)= 8.3395e-17
Erreur RMS ordre 5 (M=6)= 3.7282e-09
Corrélation ordre 5 (M=6)= 1

- Avec $M = N - 3 = 3$, l'erreur RMS est grande, la corrélation est médiocre, l'approximation est mauvaise.
- Les autres approximations sont bonnes. Le graphique avec M=3 démontre qu'il n'y a pas de « wiggle » :

```
  a = 0.0;
  b = 0.5;
  figure(1)
  plot(xi,yi,'bo','LineWidth',2), grid
  xlabel('x (m)', 'Fontsize',15)
  ylabel('Force (N)', 'Fontsize',15)
  title('Force en fonction de la déformation ', 'Fontsize',15)
% Calcul du polynôme optimal (M=4) avec beaucoup de points pour graphique
  xx = [a:0.01:b]';
  yy = (((coef3(4)).*xx+coef3(3)).*xx+coef3(2)).*xx+coef3(1);
  hold on
  plot(xx, yy, 'r', 'LineWidth',2)
```



Réponses :

(a) Quel est le nombre optimal de coefficients M ?

M = 4 coefficients

(b) Quelle méthode entre l'interpolation et l'approximation est la meilleure à prendre dans ce problème? Calculer les erreurs RMS et les corrélations dans chaque cas pour supporter vos conclusions.

L'interpolation (M = 6) et les approximations avec M = 4, 5 donnent les même résultats mais les 5e et 6e coefficients des approximations M =5, 6 sont relativement négligeables. Vu que les mesures sont annoncées comme étant très précises, le polynôme avec M=3 est rejeté. Le polynôme optimal est donc avec M = 4 coefficients (équation cubique d'ordre 3) :

$$F(x) = 10000000\,x^3 + (-9000000)\,x^2 + (2375000)\,x + (12500)$$

Intégrale analytique et calcul du travail :

- L'intégrale analytique donne ($a = 0, b = 0.5$):

$$\int_a^b F(x)dx = [10000000\, b^4/4 - (9000000)\, b^3/3 + (2375000)\, b^2/2 + (12500)b] - [10000000\, a^4/4 - (9000000)\, a^3/3 + (2375000)\, a^2/2 + (12500)a]$$

- Code MATLAB:

```
% Intégration analytique pour trouver le travail

  Int_exacte = coef3(1)*b + coef3(2)*b^2/2 + coef3(3)*b^3/3 + coef3(4)*b^4/4 -...
               (coef3(1)*a + coef3(2)*a^2/2 + coef3(3)*a^3/3 + coef3(4)*a^4/4);
  disp(' ')
  disp(['Travail fait pendant la collision = ', num2str(Int_exacte), ' Joules'])
```

- Résultats sur MATLAB
  Travail fait pendant la collision = 84375 Joules

Réponses :

(c) Par intégration analytique de la fonction $F(x)$, calculer le travail fait pendant la collision.
Travail fait pendant la collision = 84375 Joules
(d) Où est allée cette énergie?
Perdue en chaleur dans le métal tordu de la voiture.

Calcul de la vitesse initiale :

Théorème de l'énergie cinétique : $\frac{1}{2}mv_i^2 = U'_{1,2} = 84375\text{ J} \Rightarrow v_i = \sqrt{2U'_{1,2}/m}$ où $U'_{1,2}$ est le travail de déformation.

- Code MATLAB:

```
  U    = Int_exacte;
  vini = sqrt(2*U/m);
  vkmh = vini*3.6;
  disp(' ')
  disp(['Vitesse initiale = ', num2str(vkmh), ' km/h'])
```

- Résultats sur MATLAB
  Vitesse initiale = 36 km/h

Réponse :

(e) Calculer la vitesse de l'automobile juste avant l'impact.
Vitesse initiale avant impact = 36 km/h

Questions complémentaires :

Déformation élastique du parechoc

Théorème de l'énergie mécanique : $\frac{1}{2}mv_i^2 = U'_{1,2} = 84375\text{ J} = V_{rf} = \frac{1}{2}k\Delta x_f^2 \Rightarrow k = 2U'_{1,2}/\Delta x_f^2$ où l'énergie cinétique initiale est égale au travail fait pendant la déformation de la voiture, $U'_{1,2}$.

- Code MATLAB:

```
  deform = 0.25;
  k = 2*U/(deform^2);
```

```
disp(' ')
disp(['Rigidité du parechoc-ressort (déformation de ', num2str(deform),' m) = ',
      num2str(k/1000), ' kN/m'])
```

- Résultats sur MATLAB
  Rigidité du parechoc-ressort (déformation de 0.15 m) = 7500 kN/m

Réponse :

(f) Au lieu de se déformer de façon plastique, supposons maintenant que le parechoc est un ressort linéaire de rigidité *k*. Si la déformation élastique du parechoc ne doit pas excéder 15 cm avec cette même vitesse initiale, quelle devrait être la valeur de la rigidité *k* du parechoc?
Rigidité du parechoc-ressort (déformation de 0.15 m) = 7500 kN/m

Collision semi-élastique avec restitution de $\boldsymbol{e = 0.2}$

Coefficient de restitution : $e = \frac{v'_{mur} - v'_{auto}}{v_{auto} - v_{mur}} = \frac{0 - v'_{auto}}{v_{auto} - 0} \Rightarrow v'_{auto} = -ev_{auto} = -(0.2)(10) = -2m/s = -7.2\ km/h$

Énergie initiale : $\frac{1}{2}mv_i^2 = U'_{1,2} = 84375\ \text{J}$ Énergie finale : $\frac{1}{2}mv_f^2 = \frac{1}{2}m(ev_i)^2 = 3375\ \text{J} \Rightarrow \Delta E = 81\ kJ$

- Code MATLAB:

```
cores = 0.2;
vfin  = -vini*cores;
disp(' ')
disp(['Vitesse après impact (restitution de ', num2str(cores),') = ',num2str(vfin*3.6), ' km/h'])

Eini = 0.5*m*vini^2;
Efin = 0.5*m*vfin^2;
deltaE = Eini-Efin;
disp(['Perte d''énergie (restitution de ', num2str(cores),') = ', num2str(deltaE/1000), ' kJ'])
disp(' ')
```

- Résultats sur MATLAB
  Vitesse après impact (restitution de 0.2) = -7.2 km/h
  Perte d'énergie (restitution de 0.2) = 81 kJ

Réponses :

(g) Si maintenant l'impact avec le mur (qui est infiniment plus massif que la voiture) est une collision avec un coefficient de restitution e = 0.2, quelle serait la vitesse de la voiture après l'impact? Quelle serait la perte d'énergie pendant l'impact. Où est allée cette énergie?
Vitesse après impact (restitution de 0.2) = -7.2 km/h
Perte d'énergie (restitution de 0.2) = 81 kJ

Problème 4:

$$m\frac{d^2x}{dt^2} = -K_1 x - K_2 \sin(x)$$

(a) Équation d'équilibre : $x = x_e$ t.q. $\frac{d^2x}{dt^2} = 0$

donc,
$$0 = -K_1 x_e - K_2 \sin(x_e)$$
$$K_1 x_e = -K_2 \sin(x_e)$$
$$\frac{\sin(x_e)}{x_e} = \frac{-K_1}{K_2}$$

(b) Linéarisée l'équation:

Posons $f(x) = m\frac{d^2x}{dt^2} = -K_1 x - K_2 \sin(x)$

Taylor: $f(x_0 + \Delta x) = f(x_0) + \Delta x \left.\frac{df(x)}{dx}\right|_{x=x_0}$

$$f(x_0 + \Delta x) = -K_1 x_0 - K_2 \sin(x_0) + \Delta x\left[-K_1 - K_2 \cos(x_0)\right]$$

$$m\frac{d^2x}{dt^2} = f(x_0 + \Delta x) = -\left[K_1 + K_2\cos(x_0)\right]\Delta x - \left[K_1 x_0 + K_2 \sin(x_0)\right]$$

Nous savons que $\Delta x = x - x_e$, donc

$$\frac{d^2\Delta x}{dt^2} = \frac{d^2x}{dt^2} - \frac{d^2x_e}{dt^2}$$ puisque $x_e = cste$, nous obtenons

$$\frac{d^2\Delta x}{dt^2} = \frac{d^2x}{dt^2}$$ car $\frac{d^2x_e}{dt^2} = 0$

si $x_0 = x_e$

$$m\frac{d^2\Delta x}{dt^2} = -\left[K_1 + K_2\cos(x_e)\right]\Delta x - \left[K_1 x_e + K_2 \sin(x_e)\right]$$

au point (a), nous savons que $K_1 x_e = -K_2 \sin(x_e)$, donc

$$m\frac{d^2\Delta x}{dt^2} = -\left[K_1 + K_2\cos(x_e)\right]\Delta x - \underbrace{\left[K_1 x_e - K_1 x_e\right]}_{\to 0}$$

$$m\frac{d^2\Delta t}{dt^2} = -\left[K_1 + K_2\cos(x_e)\right]\Delta x$$

(c) Alors, $K_{eq} = K_1 + K_2\cos(x_e)$