Les fonctions et portions de scripts utilisent la syntaxe de Python 3.4 et scilab 5.5.2.

Q1. La taille en octets d'un cube de données avant compression est :

$$t_o = \frac{(64 \times 64 \times 352 \times 12)}{8} = 2162688\, octets$$

On note ici que le sujet utilise le méga-octet (10^6 octets) et non le mébioctet (2^{20} octets).

Q2. Le taux de compression est donc :

$$\tau = \frac{2162688 - 1000000}{2162688} \approx 0{,}538$$

Q3. Pour l'exemple donné, l'entropie de Shannon vaut :

$$H_{exemple} = -(2 * 0.3 * \log_2(0.3)\ + 4 * 0.1 * \log_2(0.1)\ ) \approx 2{,}37$$

La longueur moyenne de codage de l'exemple est donc légèrement supérieure à l'entropie calculée. On est au taux de compression maximal, puisqu'un codage de la même chaine avec 23 bits donnerait une valeur inférieure à l'entropie de Shannon.

Q4. Voir script
Q5. Voir script
Q6. Voir script
Q7. Un bloc de données est constitué de pixels codés sur 12 bits. On peut donc exprimer le taux de compression limite en fonction de l'entropie de Shannon (voir script pour le code), exprimée en bits par pixel :

$$\tau = \frac{12 - H(bloc\_image)}{12}$$

Q8. Voir script
Q9. Voir script
Q10. Voir script
Q11. Complexité

Dans la fonction `pretraitement12`, on parcourt deux boucles simples de m itérations, on a donc une complexité linéaire en $\mathcal{O}(m)$.

Dans la fonction `pretraitement34`, on parcourt une boucle simple de n itérations, là-aussi on a une complexité linéaire en $\mathcal{O}(n)$.

Dans la fonction `pretraitement5`, on parcourt deux boucles imbriquées de n et m itérations, on a donc une complexité quadratique en $\mathcal{O}(n^2)$, en posant $n \equiv m$.

Q12. Voir script
Q13. Voir script

Q14. Code de Rice pour $p = 3$

δ_k	Quotien par $2^p = 8$	Codage unaire	Reste	Codage binaire	Codage complet
4	0	0	4	100	0-100
10	1	10	2	010	10-010
18	2	110	2	010	110-010
18	2	110	2	010	110-010
6	0	0	6	110	0-110
1	0	0	1	001	0-001
3	0	0	3	011	0-011

On notera que les codages des valeurs 4 à 7 du tableau 4 donné en exemple pour $p = 2$ sont inutiles, puisque le reste de la division entière par 4 ne peuvent pas prendre ces valeurs.

Q15. Voir script

Q16. Voir script. On notera que les candidats, en Python, ne doivent pas nécessairement prendre en compte le fait que la fonction `bin` renvoie un résultat commençant par `'0b'`.

Q17. Gravitation

Si on répond en s'appuyant sur le cours de SI, on applique le théorème de la résultante dynamique appliqué à la sonde. Le bilan des actions mécaniques est alors :

- Action du Soleil : $-G.\frac{m.m_s}{\|\vec{r}(t)\|^2}.\frac{\vec{r}(t)}{\|\vec{r}(t)\|}$, le vecteur $\frac{\vec{r}(t)}{\|\vec{r}(t)\|}$ étant unitaire et orienté depuis le centre du soleil.
- Action de Vénus : $-G.\frac{m.m_v}{\|\vec{r}(t)-\vec{r_v}(t)\|^2}.\frac{\vec{r}(t)-\vec{r_v}(t)}{\|\vec{r}(t)-\vec{r_v}(t)\|}$, le vecteur $\frac{\vec{r}(t)-\vec{r_v}(t)}{\|\vec{r}(t)-\vec{r_v}(t)\|}$ étant unitaire et orienté depuis le centre de Vénus.

La résultante dynamique, dans le repère héliocentrique considéré galiléen, pour la sonde considérée ponctuelle est $m.\frac{d^2\vec{r}(t)}{dt^2}$.

On en déduit l'expression proposée en simplifiant par m.

Q18. Fonctions

On en déduit directement :

$$S_x(x(t), y(t), t) = -G \cdot m_s \cdot \frac{x(t)}{(x(t)^2 + y(t)^2)^{\frac{3}{2}}} - G \cdot m_v \cdot \frac{x(t) - x_v(t)}{\left(\left(x(t) - x_v(t)\right)^2 + \left(y(t) - y_v(t)\right)^2\right)^{\frac{3}{2}}}$$

$$S_y(x(t), y(t), t) = -G \cdot m_s \cdot \frac{y(t)}{(x(t)^2 + y(t)^2)^{\frac{3}{2}}} - G \cdot m_v \cdot \frac{y(t) - y_v(t)}{\left(\left(x(t) - x_v(t)\right)^2 + \left(y(t) - y_v(t)\right)^2\right)^{\frac{3}{2}}}$$

On remplace $x_v(t)$ et $y_v(t)$ par leurs expressions en fonction du temps :

$$S_x(x(t), y(t), t) = -G \cdot m_s \cdot \frac{x(t)}{(x(t)^2 + y(t)^2)^{\frac{3}{2}}} - G \cdot m_v \cdot \frac{x(t) - r_v.\cos(\Omega.t + \Phi)}{((x(t) - r_v.\cos(\Omega.t + \Phi))^2 + (y(t) - r_v.\sin(\Omega.t + \Phi))^2)^{\frac{3}{2}}}$$

$$S_y(x(t),y(t),t) = -G \cdot m_s \cdot \frac{y(t)}{(x(t)^2+y(t)^2)^{\frac{3}{2}}} - G \cdot m_v \cdot \frac{y(t) - r_v.\sin(\Omega.t+\Phi)}{((x(t)-r_v.\cos(\Omega.t+\Phi))^2 + (y(t)-r_v.\sin(\Omega.t+\Phi))^2)^{\frac{3}{2}}}$$

Voir script pour le codage.

Q19. Euler explicite 1

On note que :

$$\frac{d(v_x(t))}{dt} = S_x(x(t),y(t),t)$$

$$\frac{d\left(v_y(t)\right)}{dt} = S_y(x(t),y(t),t)$$

En application directe du cours :

$$u_{i+1} = u_i + \Delta t * S_x(x(t),y(t),t)$$

$$v_{i+1} = v_i + \Delta t * S_y(x(t),y(t),t)$$

Q20. Euler explicite 2

On note également que :

$$v_x(t) = \frac{d(x(t))}{dt}$$

$$v_y(t) = \frac{d(y(t))}{dt}$$

D'où les expressions :

$$x_{i+1} = x_i + \Delta t * u_i$$

$$y_{i+1} = y_i + \Delta t * v_i$$

Q21. Voir script

Q22. Voir script

La quantité de mémoire à allouer pour ce calcul est :

$$qm = 4 * 8 * 30 * 24 * 3600 = 103680000\ octets = 103.68\ Mo$$

Le trajet de la sonde ayant duré un peu moins de sept ans, le calcul semble faisable du point de vue de la mémoire utilisée.