

Correction Examen d' Informatique / Modélisation (1h30)

2025-2026

**AF 1: Être capable de formuler un problème sous forme d'un modèle.
Savoir faire appel aux outils numériques adaptés au problème.**

Question 1: Donner la solution à cette equation différentielle, $R(x)$, pour le cas stationnaire (Considérer une hauteur de matériel meuble R_0 et une pente nulle en $x=0$). Quelle est la forme des profils topographiques attendus pour des versants stables ? /1

Dans le cas stationnaire:

$$\kappa \frac{\partial^2 R}{\partial x^2} + \rho_r W = 0$$

En intégrant 2 fois, on obtient:

$$R(x) = -\frac{\rho_r W}{2K} x^2 + c$$

Avec c une constante.

La forme des profils topographiques à l'équilibre est donc parabolique.

Question 2: Ecrire sous forme matricielle le problème inverse vous permettant d'estimer le taux de production de matériel meuble W ? Comment est-il possible de résoudre ce problème inverse ? /2

Sous forme matricielle le problème inverse s'écrit:

$$\begin{Bmatrix} z_1 \\ z_2 \\ \dots \\ z_N \end{Bmatrix} = \begin{Bmatrix} dx_1^2 & 1 \\ dx_2^2 & 1 \\ \dots & \dots \\ dx_N^2 & 1 \end{Bmatrix} \begin{Bmatrix} W \\ c \end{Bmatrix}$$

Avec N , le nombre de point de mesure et $d = \frac{\rho_r}{2K}$

Il est possible de résoudre ce problème linéaire avec la solution des moindres carrés.

Question 3: Combien de points en x , N_x , comporte votre maillage en fonction de x_{max} et dx ? Décrire une implémentation du maillage 1D de la distance horizontale. Quelle est la distance en mètre à l'indice 20 ? /1

```
x = np.arange(0, xmax + dx, dx) # maillage 1D en x
Nx = len(x) = int(xmax/dx) + 1 = 501 # nombre de point en x
x[20] = 20*dx = 20
```

Question 4: Décrire une implémentation du maillage 2D de la hauteur de régolithe, $R(t, x)$, dans le langage Python. /1

```
R = np.zeros((Nt + 1, Nx+1)) # maillage 2D de la hauteur de regolithe
```

Question 5: Quel est le temps à l'indice N_t ? En supposant une production de regolithe nulle ($W=0$), calculer l'ordre de grandeur du temps de diffusion sur l'ensemble du versant pour une diffusivité, $\kappa = 0.05 \text{ m}^2.\text{yr}^{-1}$. Qu'en concluez-vous ? /2

$t_{\max} = t[N_t] = N_t * dt = 10.000 * 10 \text{ ans} = 0.1 \text{ Ma}$
 $t_{\text{car}} = x_{\max}^2 / K_c = 5 \text{ Ma}$

$t_{\max} \ll t_{\text{car}}$, il faut donc augmenter le nombre d'iteration pour parcourir l'ensemble du versant. Il est possible aussi d'augmenter le pas de temps dt mais en faisant attention à ne pas dépasser le critère de stabilité, ie. $dt_{\max} = dx^2 / (2.0 * K) = 10 \text{ ans}$.

Partie 2: Analyse d'image

Question 6: Vous avez ouvert l'image avec *gdal* et enregistré les valeurs d'amplitude de l'image dans un tableau *numpy* appelé *amp*. Décrire succinctement une méthode pour 1) améliorer les contrastes de l'image et 2) binariser l'image en appliquant un seuillage sur l'eau. /3

1) On remarque sur l'histogramme de l'amplitude que les valeurs sont généralement très faibles, la dynamique de l'image est donc très comprimée et l'image apparaît sombre. L'amélioration des contrastes peut se faire en convertissant en décibels les valeurs, i.e. en prenant le logarithme de l'amplitude: $\text{amp} = \log_{10}(\text{amp})$. Rq: il est aussi possible d'effectuer un étirement linéaire comme nous l'avons fait en TD.

2) Le seuillage se fait ensuite très simplement avec une opération booléenne: $(\text{amp} > \text{seuil}).\text{astype}(\text{np.uint8})$ avec $\text{seuil} = 2$. Attention: Si vous avez précédemment effectué un étirement linéaire, la valeur de seuil sera différente.

AF 2: Être capable de développer des programmes structurés et robustes en Python, intégrant des bibliothèques scientifiques, des interfaces graphiques et des visualisations avancées.

Question 7: On considère le tableau NumPy suivant: /1

```
A = np.array([
    [12, 15, 18, 20],
    [10, 14, 16, 19],
    [ 8, 11, 13, 17]
])
```

Que renvoie :

- `np.shape(A)`: (3, 4)
- `A[:, 3]`: [20, 19, 17]
- `A[-1, 1]`: 11

Question 8: Charger les deux dernières colonnes du fichier texte suivant avec la bibliothèque Numpy: /2

`lon, lat = np.loadtxt(fichier, delimiter=' ', usecols=(1,2), skiprows=1, unpack=True)`

Question 9: Compléter le programme Python de la page suivante permettant d'afficher une interface graphique telle que le montre la figure ci-contre pour la visualisation de données raster au format TIFF. Le programme utilise la bibliothèque GDAL pour la lecture des données et Matplotlib pour l'affichage graphique, intégrés dans une interface PyQt5.

```
# -----
# Fonctions
# -----
def charger_fichier():

    path, _ = QFileDialog.getOpenFileName(
        None,
        "Ouvrir un fichier",
        "",
        "Fichiers TIFF (*.tif *.tiff)"
    )

    ds = gdal.Open(path)

    filename = os.path.basename(path)
    # Mise à jour de la zone de texte
    edit_filename.setText(filename) /0.5

    # Mise à jour de la valeur maximale du numéro de bande autorisé
    edit_band.setRange(1, ds.RasterCount) /0.5
    edit_band.setEnabled(True)

    # Mise à jour metadonnées de l'image
    info = (
        f"Taille (X,Y) : {ds.RasterXSize} x {ds.RasterYSize}\n" /0.5
        f"Nb de bandes : {ds.RasterCount}\n"
        f"Résolution pixel (m) : {ds.GetGeoTransform()[1]}\n" /0.5
    )
    log_meta.setText(info)

def afficher_bande():

    num_band = edit_band.value()
    cmap = cmap_selector.currentText()

    band = ds.GetRasterBand(num_band)

    # Extraire la bande du raster dans un tableau NumPy
    array = band.ReadAsArray() /0.5

    figure.clear()
    ax1 = figure.add_subplot(121)
    ax2 = figure.add_subplot(122)

    # Affichage de l'image
    cax = ax1.imshow(array, cmap=cmap, /0.5
                     vmin=10, vmax=100)
    ax1.set_title(f"Bande {num_band} ({cmap})")
    figure.colorbar(cax, ax=ax1, label="DN")

    # Histogramme
    ax2.hist(array.flatten(), bins=256, /0.5
             color='black', alpha=0.7)
    ax2.set_title(f"Histogramme - Bande {num_band}")
    ax2.set_xlabel("Valeurs DN")
    ax2.set_ylabel("Fréquence")

    # Ajustement automatique de la mise en page
```

```

figure.tight_layout()
canvas.draw()

def create_button(libelle, method):
    """Création d'un bouton et association de l'action au clic"""
    button = QPushButton(libelle) /0.5
    button.clicked.connect(method) /0.5
    return button

# -----
# Programme principal
# -----

app = QApplication([])

# Creation d'une fenêtre principale
window = QWidget() /0.5
window.setWindowTitle("Exam Modélisation")
window.setGeometry(100, 100, 700, 500)

# Boutons
b_charger = create_button("Charger fichier", charger_fichier)
b_afficher = create_button("Afficher bande", afficher_bande)
b_quitter = create_button("Quitter", qApp.quit)

# Fichier
label_filename = QLabel("Nom du fichier de données:")
label_filename.setAlignment(Qt.AlignCenter)
edit_filename = QLineEdit()
edit_filename.setReadOnly(True)

# Bande
label_band = QLabel("Numéro de bande:")
label_band.setAlignment(Qt.AlignCenter)
edit_band = QSpinBox()
edit_band.setEnabled(False)

# Palette de couleurs
label_cmap = QLabel("Palette de couleurs:")
label_cmap.setAlignment(Qt.AlignCenter)
cmap_selector = QComboBox()
cmap_selector.addItem(
    ['viridis', 'plasma', 'coolwarm', 'gray', 'inferno', 'magma']
)

# choix des vmin / vmax palette de couleur
label_vmin = QLabel("vmin :")
edit_vmin = QSpinBox()
edit_vmin.setRange(-10000, 10000)
edit_vmin.setValue(10)

label_vmax = QLabel("vmax :")
edit_vmax = QSpinBox()
edit_vmax.setRange(-10000, 10000)
edit_vmax.setValue(100)

# Métadonnées
log_meta = QTextEdit()
log_meta.setReadOnly(True)

# Figure Matplotlib
figure = plt.figure(figsize=(8, 4))
canvas = FigureCanvas(figure)

# Organisation des widgets

```

```

layout = QGridLayout()

layout.addWidget(b_charger, 0, 0, 1, 2)

layout.addWidget(label_filename, 1, 0, 1, 1)

layout.addWidget(edit_filename, 1, 1, 1, 1)
layout.addWidget(label_band, 2, 0, 1, 1)

layout.addWidget(edit_band, 2, 1, 1, 1)

layout.addWidget(label_cmap, 3, 0, 1, 1)

layout.addWidget(cmap_selector, 3, 1, 1, 1)

layout.addWidget(label_vmin, 4, 0, 1, 1)

layout.addWidget(edit_vmin, 4, 1, 1, 1)

layout.addWidget(label_vmax, 5, 0, 1, 1)

layout.addWidget(edit_vmax, 5, 1, 1, 1)

layout.addWidget(b_afficher, 6, 0, 1, 2)

layout.addWidget(log_meta, 7, 0, 2, 2)

layout.addWidget(canvas, 9, 0, 2, 2)

layout.addWidget(b_quitter, 11, 0, 1, 2)

window.setLayout(layout)
window.show()
app.exec_()

```

/2 (retirer 1 pt si
label/edit vmin/vmax sont
faux ou recopie TD en POO ...)