



Projet DEPHI: Modernisation de l'interface du TP Zygo

Abdallah Mrabti

Annexe

01

Introduction

02

Problématique

03

Démarche

04

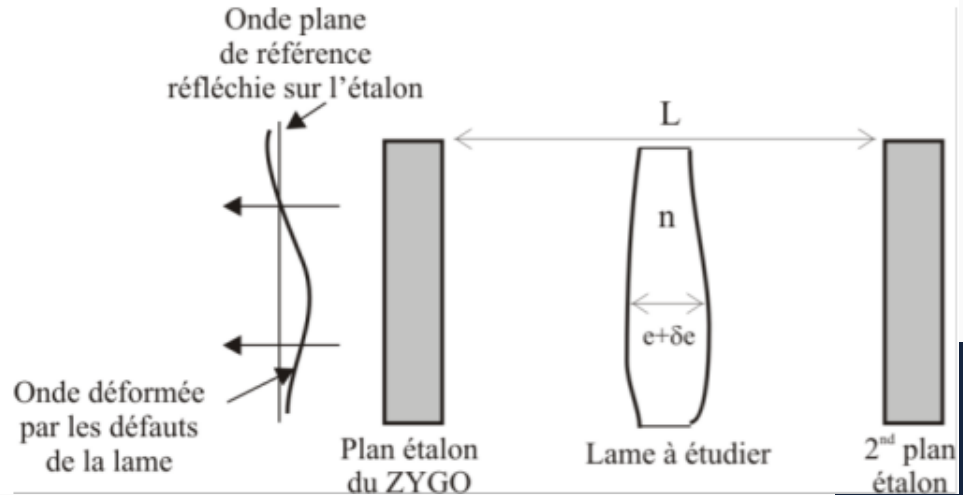
**Planification et
Organisation**

05

Conclusion

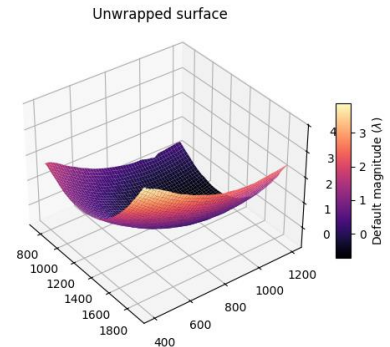
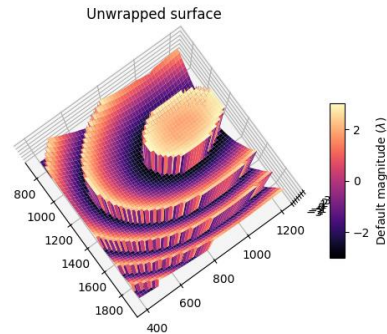
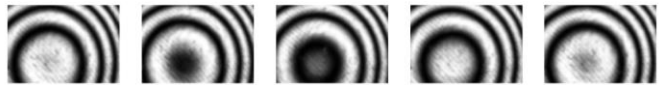
Introduction: Interféromètre de Zygo

L'interféromètre de Zygo est un instrument de mesure de haute précision utilisant l'interférométrie pour analyser la topographie et la rugosité des surfaces avec une résolution nanométrique.



Introduction : Algorithme de Hariharan

L'algorithme à 5 images (de Hariharan) utilisé dans le programme Zygo_GUI pour le Zygo permet de mesurer la phase entre chaque image, avec un décalage de phase de $\pi/2$ entre chaque image, ce qui permet de déterminer les défauts de surface.

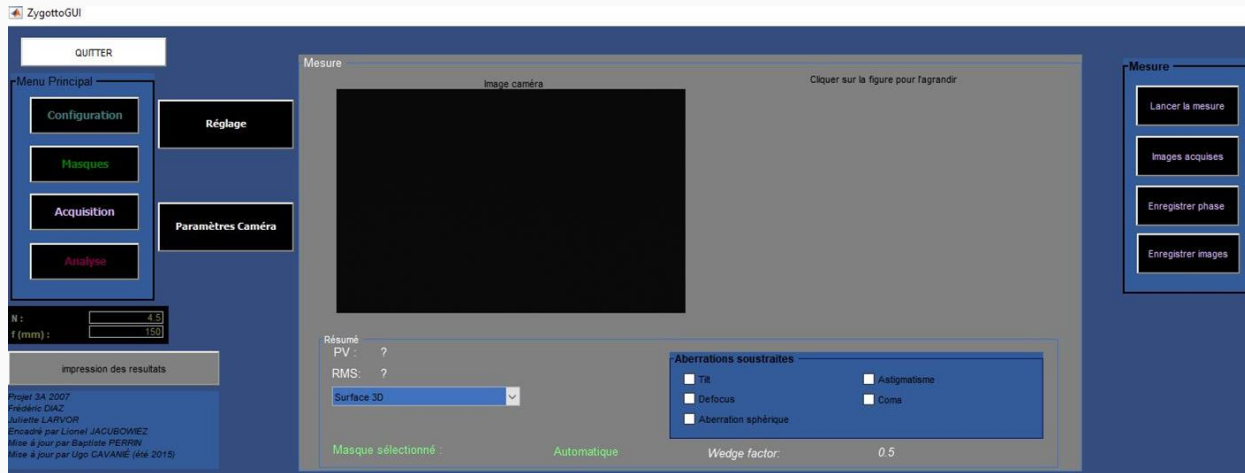


Objectifs

Passage de Matlab
vers Python

Améliorer
l'interface
graphique

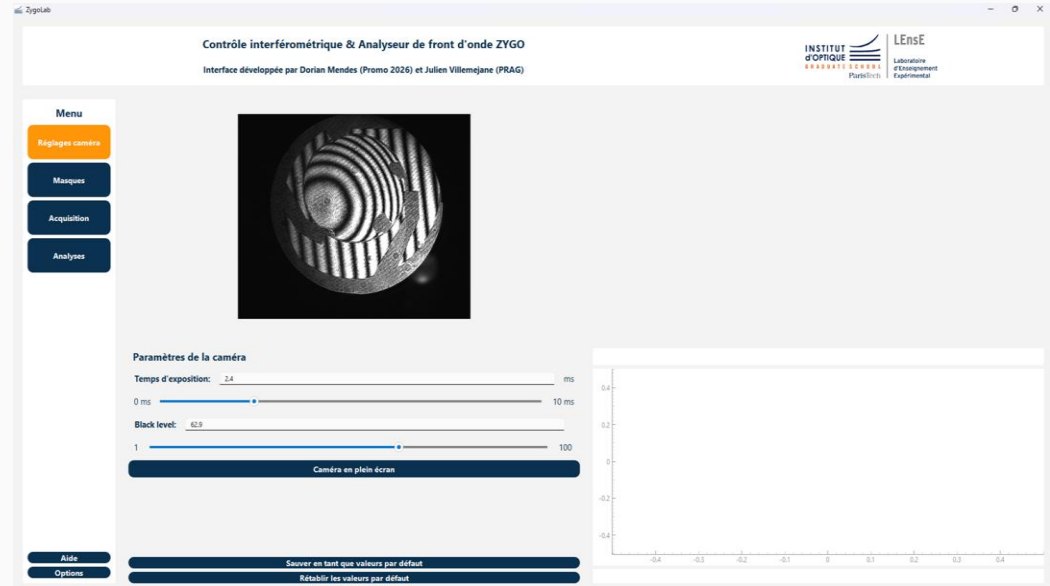
Optimiser le
logiciel



Démarche

Nous disposons déjà d'une première version de l'application que je peux utiliser comme modèle, mais elle est imparfaite.

Elle comporte quelques bugs et des calculs erronés, ainsi qu'une latence non négligeable.



Contrôle interférométrique & Analyseur de front d'onde ZYGO

Interface développée par Dorian Mendes (Promo 2026) et Julien Villemejeane (PRAG)

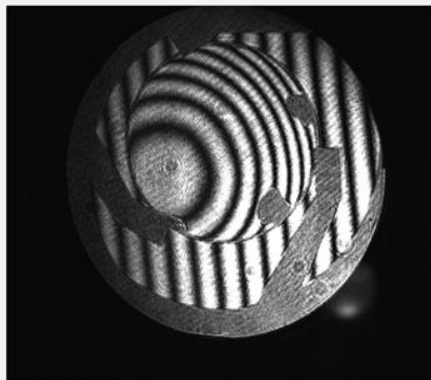
Menu

Réglages caméra

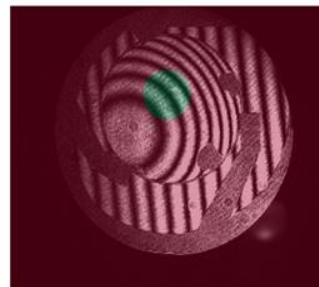
Masques

Acquisition

Analyses



Visualisation de la zone sélectionnée



La zone sélectionnée est en vert.

Gestion des masques

Circulaire

Supprimer le masque

Rectangulaire

Supprimer tous les masques

Polygonal

Appliquer le masque

Masque sélectionné

Inverser le masque

Inverser la fusion des masques

Aide

Options

Contrôle interférométrique & Analyseur de front d'onde ZYGO

Interface développée par Dorian Mendes (Promo 2026) et Julien Villemejeane (PRAG)

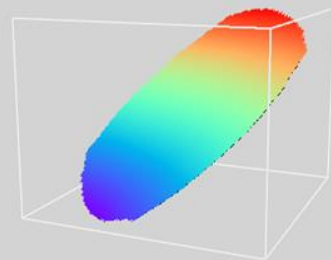
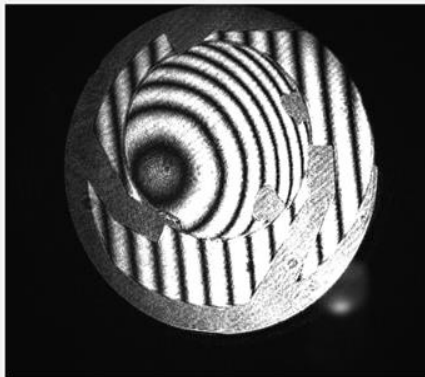
Menu

Réglages caméra

Masques

Acquisition

Analyses



Acquisition

Wedge factor

Acquisition unique

Acquisition répétée

Acquisition en cours...

Soustraire les défauts

- Retirer le tilt
 Retirer l'astigmatisme
 Retirer le defocus
 Retirer la coma
 Retirer l'aberration sphérique

Aide

Options

Voir et enregistrer les images

Enregistrer la phase

Résultats

	1	2	3	4	5	Moyenne
PV (λ)	1.5796	nan	nan	nan	nan	1.5796
RMS (λ)	0.401	nan	nan	nan	nan	0.401

Contrôles interférométriques | Analyses poussées

Interface développée par Dorian Mendes (Promo 2026) et Julien Villemejeane (PRAG)

Menu

Focale (mm)

f-number

Zernike / Seidel

PSF

MTF

Spot diagram

Coefficients de Zernike

Les coefficients sont donnés en λ .

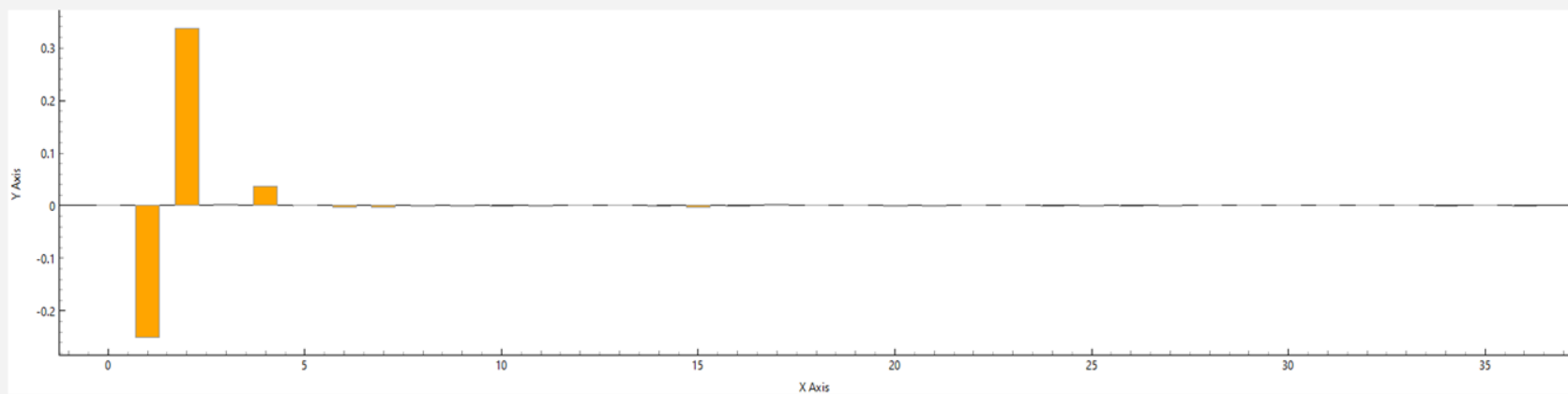
C1-C9	-0.2503	0.3375	0.0034	0.0369	0.0015	-0.0021	-0.0032	-0.0001	-0.0014
C10-C18	-0.0008	-0.0012	0.0008	0.0003	-0.0001	-0.0021	-0.0009	0.002	0.0002
C19-C27	0.0006	-0.0016	-0.0016	0.0003	0.0008	-0.0009	-0.0014	-0.0008	-0.0001
C28-C36	0.0006	0.0004	0.0003	-0.0	-0.0	0.0003	-0.0007	0.0006	-0.001

Signification des coefficients

Coefficients de Seidel

Les coefficients sont donnés en λ .

	Tilt	Defocus	Astigmatism	Coma	Sph. Ab
Amplitude	0.3375	0.0738	0.0051	0.0095	-0.0072
Angle	0.5843	°	-26.5502	-178.9315	°



Points techniques difficiles

- La première version contient des milliers de lignes de codes
- PyQt6: une bibliothèque Python permettant de créer des interfaces graphiques utilisateur (GUI)

acquisition_menu_widget.py	11/7/2024 6:43 PM	PY File	30 KB
analysis_menu_widget.py	11/7/2024 6:43 PM	PY File	7 KB
bar_chart_widget.py	11/7/2024 6:43 PM	PY File	7 KB
camera_settings_widget.py	11/7/2024 6:43 PM	PY File	9 KB
combobox_bloc.py	11/7/2024 6:43 PM	PY File	4 KB
contour_plot.py	11/7/2024 6:43 PM	PY File	10 KB
display_zernike_widget.py	11/7/2024 6:43 PM	PY File	9 KB
imshow_pyqtgraph.py	11/7/2024 6:43 PM	PY File	11 KB
lineedit_bloc.py	11/7/2024 6:43 PM	PY File	4 KB
main_menu_widget.py	11/7/2024 6:43 PM	PY File	8 KB
mask_selectors.py	11/7/2024 6:43 PM	PY File	23 KB
masks_menu_widget.py	11/7/2024 6:43 PM	PY File	26 KB
mtf_view.py	11/7/2024 6:43 PM	PY File	18 KB
options_menu_widget.py	11/7/2024 6:43 PM	PY File	5 KB
piezo_calibration_widget.py	11/7/2024 6:43 PM	PY File	10 KB
psf_view.py	11/7/2024 6:43 PM	PY File	19 KB
results_menu_widget.py	11/7/2024 6:43 PM	PY File	6 KB
slider_bloc.py	11/7/2024 6:43 PM	PY File	8 KB
table_from_numpy.py	11/7/2024 6:43 PM	PY File	4 KB
title_widget.py	11/7/2024 6:43 PM	PY File	4 KB
x_y_chart_widget.py	11/7/2024 6:43 PM	PY File	13 KB
x_y_z_chart_widget.py	11/7/2024 6:43 PM	PY File	9 KB

```
268     def show_analysis_window_maximized(self):
269         try:
270             self.analysis_window.showMaximized()
271         except Exception as e:
272             print(f'Exception - analysis_app_open (e)')
273
274     if __name__ == '__main__':
275         from PyQt6.QtWidgets import QApplication
276
277         class PyMainWindow(QMainWindow):
278             def __init__(self):
279                 super().__init__()
280
281                 # Define window title
282                 self.setWindowTitle(translate("ZygoLab"))
283                 self.setWindowIcon(QIcon('assets/IOAS-LENSf-icon.jpg'))
284                 self.setGeometry(50, 50, 700, 700)
285
286                 self.central_widget = ZygoLabApp()
287                 self.setCentralWidget(self.central_widget)
288
289             def closeEvent(self, event):
290                 """
291                 closeEvent redefinition, use when the user clicks
292                 on the red cross to close the window
293                 """
294                 reply = QMessageBox.question(self, 'Quit', 'Do you really want to close?',
295                                             QMessageBox.StandardButton.Yes | QMessageBox.StandardButton.No,
296                                             QMessageBox.StandardButton.No)
297
298                 if reply == QMessageBox.StandardButton.Yes:
299                     if self.central_widget.camera is not None:
300                         self.central_widget.camera_widget.disconnect()
301                     event.accept()
302                 else:
303                     event.ignore()"""
304
305                 print("Ouverture en cours...")
306                 app = QApplication(sys.argv)
307                 main = PyMainWindow()
308                 main.showMaximized()
309                 sys.exit(app.exec())
310
```

Planification

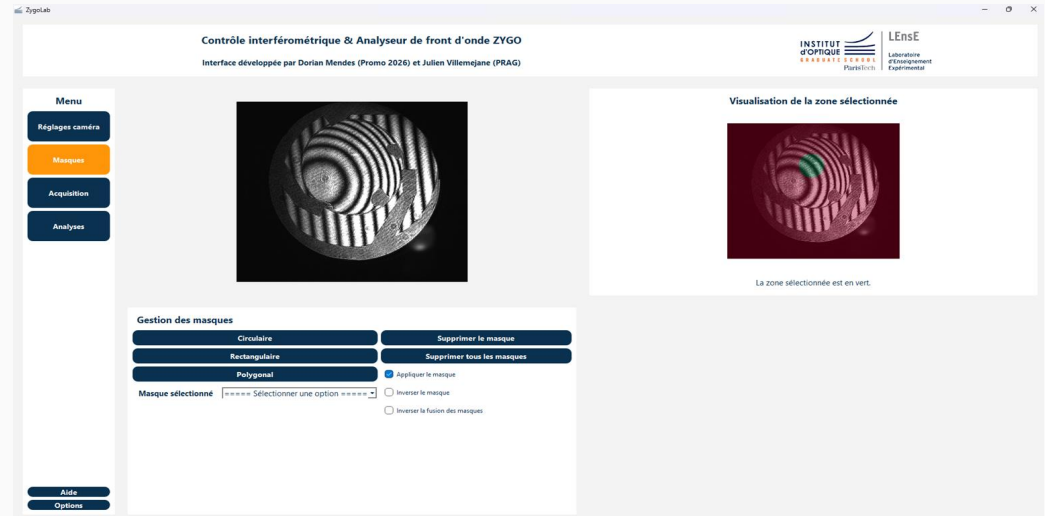
Tester les procédés de calcul sur des données pré-enregistrées

Revoir la gestion des masques et corriger les bugs.

Revoir le menu principal: Acquisitions intermédiaires, possibilité de configurer le piezo et la caméra...



Revoir l'affichage 3D: essayer de résoudre les problèmes de latence.

Découper les étapes de calculs et d'acquisition, essayer de séparer les calculs du fichier principale.





Compétence à acquérir

- Utilisation du PyQt6 et les interfaces graphiques en générale
 - Un projet assez proche du monde du travail
 - Réalisation de bibliothèques de fonctionnalités communes à l'ensemble du LEnsE
 - Méthodes d'optimisation des programmes
 - Autonomie
- 
- 

Conclusion

Pendant cette semaine, j'ai appris à connaître PyQt6 et j'ai assimilé le code de la première version de l'application GUI.

Après avoir discuté avec mes collègues, je peux également les aider à développer des interfaces pour leurs projets.

Ce projet d'interfaçage sur le Zygo permettra de simplifier et d'automatiser la mesure des surfaces, en intégrant des outils comme PyQt6 pour améliorer l'interface graphique et de moderniser le TP pour les générations suivantes.
