WirnikApp Documentation

Wydanie 1.0.0

Sławomir Polański

June 23 2016

Spis treści

WirnikApp jest aplikacją wykorzystującą Metodę Elementów Skończonych w celu określenia stanu naprężeń, oraz przemieszczenia wirnika promieniowego w czasie jego pracy. Program posiada własny interfejs graficzny, w którym możliwe jest zdefiniowanie geometrii, a także ustalenie ilości obrotów wirnika w jednostce czasu.

Główną zaletą skryptu jest fakt, iż komunikuje się on z programami GMSH, oraz Calculix. Dzięki zastosowaniu tak wydajnego generatora siatki i rzetelnego solver'a, otrzymane rezultaty są satysfakcjonujące przy jednocześnie krótkim czasie obliczeń.

Zawartość:

ROZDZIAŁ 1

Wprowadzenie

Do poprawnego uruchomienia aplikacji należy posiadać następujące oprogramowanie:

- GMSH 2.10 lub nowsze
- Calculix 2.7 lub nowsze

W przypadku programu Calculix nie stwierdzono wpływu starszej wersji na otrzymywane wyniki. Jeśli chodzi o aplikację GMSH problem pojawia się przy użyciu wersji 2.8 lub starszej, w której ustawienia domyślne generowania siatki są różne od tych zastosowanych w nowszej wersji przez co jakość dyskretyzacji jest niższa.

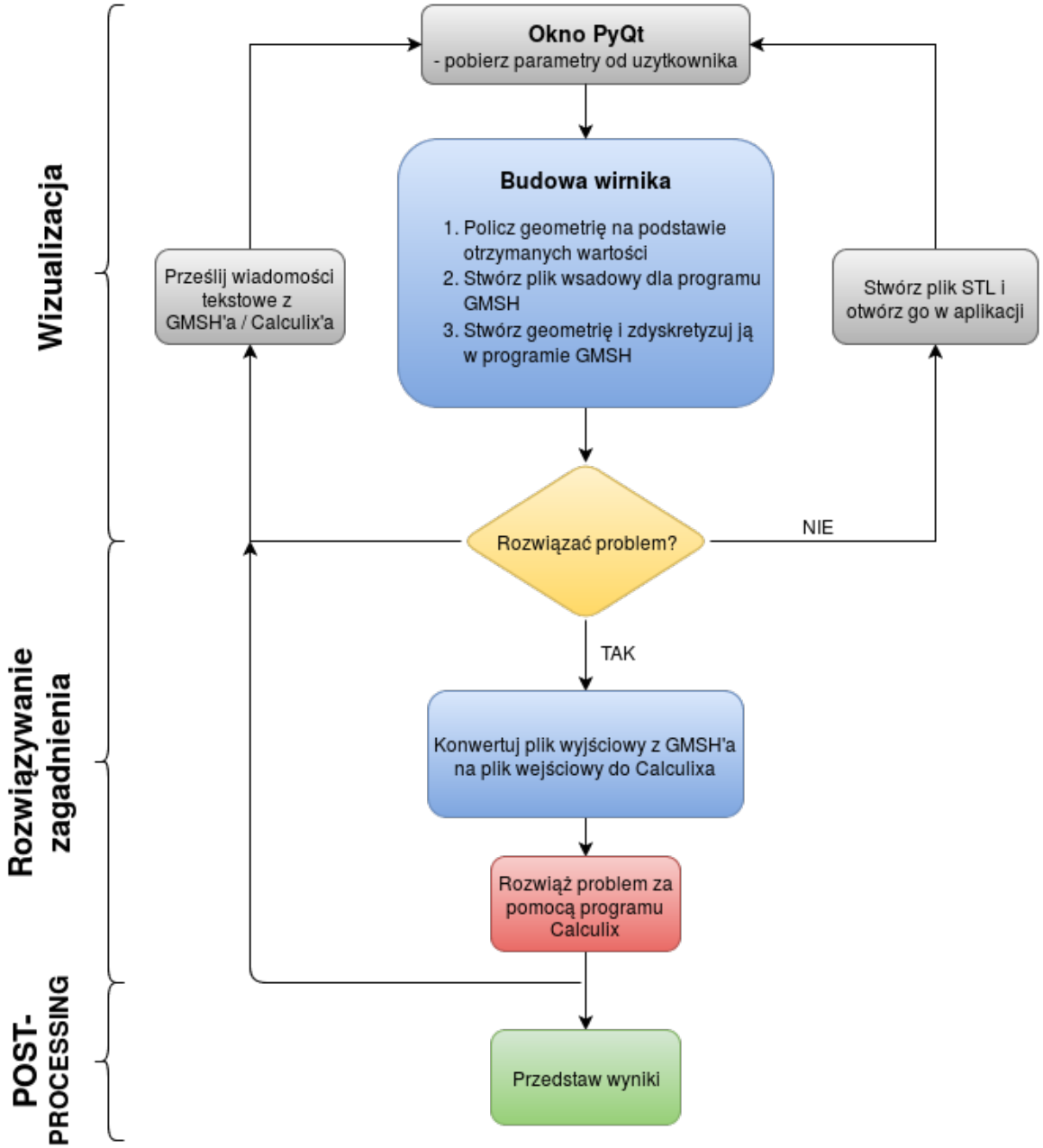
Aplikacja została napisana przy użyciu języka programowania jakim jest Python z wykorzystaniem następujących modułów:

- Subprocess
- NumPy/SciPy
- Python-Vtk
- PyQt4

W przypadku próby uruchomienia skryptu pod systemem Linux, instalacja repozytoriów polega na pobraniu powyższych repozytoriów. Uruchomienie WirnikApp na systemie Windows sprawia więcej kłopotów, gdyż instalacja modułu 'Python-Vtk' jest bardziej skomplikowana. Najłatwiejszym sposobem na ominięcie tej niedogodności jest instalacja gotowego zestawu oprogramowania Python(x,y). Pakiet ten zawiera wszystkie wymienione moduły, oraz wiele innych poszerzających zastosowanie Python'a.

Program WirnikApp uruchamiany jest poprzez otwarcie lini komend w folderze głównym aplikacji oraz wpisanie komendy *python wirnikApp*. W momencie pokazania się menu głównego użytkownik ma możliwość zdefiniowania geometrii wirnika, jakości siatki elementów skończonych czy rodzaju materiału. Jeśli WirnikApp został uruchomiony pod systemem Linux wtedy pozostawienie wartości domyślnych w rubrykach zawierających lokalizacje GMSHa i Calculixa będzie funkcjonowało poprawnie. Jest tak, dlatego iż lokalizacja zainstalowanych programów w systemie Linux jest dostępna bezpośrednio po wpisaniu nazwy programu do linii komand. Podczas próby uruchomienia aplikacji pod systemem Windows, lokalizacja preprocessor'a i solver'a musi zostać zdefiniowana przed uruchomieniem polecania wizualizacji lub obliczeń.

W celu wizualizacji geometrii wirnika stworzonego w oparciu o zdefiniowane uprzednio dane należy kliknąć przycisk 'Stwórz!'. Jak tylko przycisk zostanie naciśnięty,



Rys. 1.1: Diagram przedstawiający w jaki sposób działa aplikacja.

ROZDZIAŁ 2

Opis struktury programu

Program jest wykonywany według schematu przedstawionego poniżej.

Opis użytych funkcji i klas

Niniejsza strona zawiera opis funkcji użytych w aplikacji WirnikApp. Uwagę zwrócono przede wszystkim na rdzeń aplikacji napisany przez autora, nie przywiązując większej uwagi do menu graficznego aplikacji stworzonego w PyQt.

3.1 wirnikApp.py

3.2 messenger.py

Przykład kodu zawartego wykorzystanego w pliku ‘messenger.py’:

```
def main(dane,frt):
    from senderClass import Sender
    # Stworz gońca
    Goniec = Sender()
    # Pobierz dane z GUI
    Goniec.pobierzDane(dane)
    # Testuj dane
    Goniec.testujDane()
#============================================================================
#   Wyślij parametry do funkcji obliczających położenie punktów geometrii
#============================================================================
    import objectGeometry
    objectGeometry.obliczPotrzebneParametry(Goniec)
#============================================================================
#   Stwórz geometrię na podstawie obliczonych parametrów
#============================================================================
    import gmshInputFile
    gmshInputFile.przygotujPlik(Goniec,frt)
#============================================================================
#   Wyślij geometrię do programu GMSH
#============================================================================
    from senderClass import SenderBrain
    # Przekaż wiadomości z gońca do obiektu wykonującego operacje na programach zewnętrznych
    Brain = SenderBrain(Goniec)
    Brain.utworzGeometrie(pokazGmsh=False)
    Brain.dyskretyzujGeometrie()
#============================================================================
#   Instrukcja warunkowa która pozwala na rozróżnienie sygnału wysłanego w celu
#   wizualizacji wyników od sygnału z prośbą o rozpoczęcie symulacji
#============================================================================
    if frt == 'stl':
```

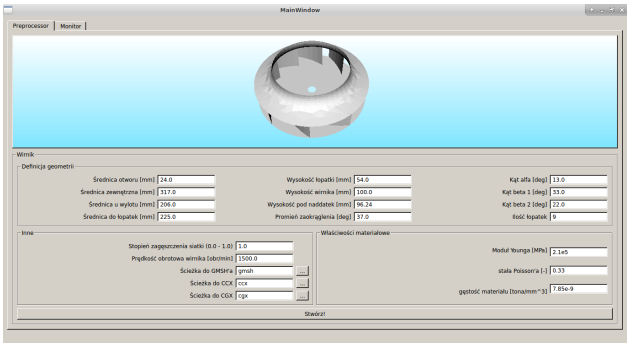
```
        # Wizualizuj obiekt
        Brain.wizualizacjaObiektu()
    if frt == 'inp':
        # Usun pozostalosci po poprzedniej symulacji jeśli istnieją
        Brain.przygotujSymulacje()

        import calculix
        # Przystosuj siatke z pliku wsadowego do użycia w Calulixie
        calculix.konwertujSiatke(Goniec)
        # Stwórz plik wsadowy do Calculixa
        calculix.stworzPlikWsadowy(Goniec)
        # Rozwiąż problem przy użyciu Calculixa i zaprezentuj wyniki
        Brain.rozwiazProblem(pokazWyniki=True)
    return True
```

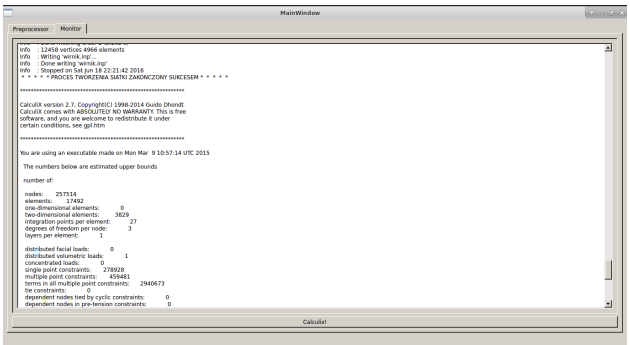
3.3 senderClass.py

Galeria

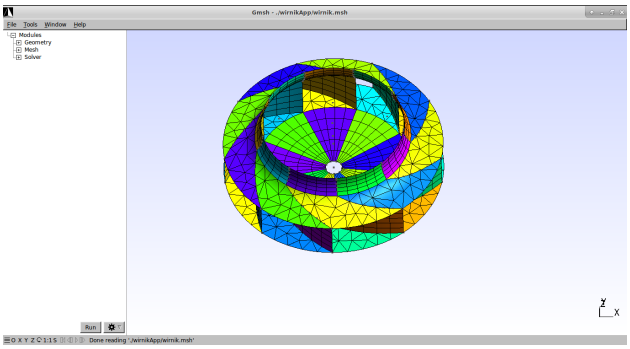
W niniejszej galerii zaprezentowany został interfejs graficzny programu, oraz wyniki uzyskanie dzięki wykorzystaniu GMSH'a i Calculix'a.



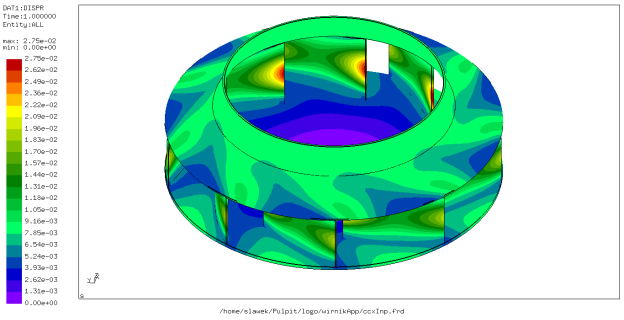
Rys. 4.1: Menu główne aplikacji WirnikApp



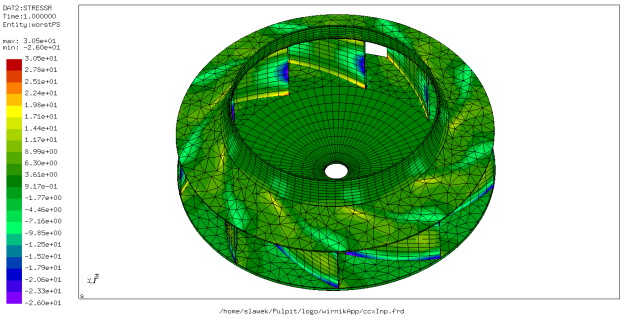
Rys. 4.2: Monitor postępu zawarty w aplikacji



Rys. 4.3: Siatka elementów skończonych w GMSH'u



Rys. 4.4: Przemieszczenia wirnika - Calculix



Rys. 4.5: Stan naprężeń wirnika - Calculix

ROZDZIAŁ 5

Pobierz!

Jeśli chcesz pobrać aplikację, możesz ją znaleźć na moim GitHubie.

https://github.com/spolanski/WirnikApp

Jeśli chciałbyś uzyskać więcej informacji na temat programu lub chciałbyś pomóc go rozwijać, napisz do mnie na polanski.slawomir@gmail.com