Studencki Iteracyjny Model Przepływu
Specyfikacja wymagań projektu

15/12/2003

1. Wstęp.

Jednym z najbardziej użytecznych zastosowań komputerów jest modelowanie świata rzeczywistego. Gdyby nie modelowania komputerowe nie byłoby lotów kosmicznych, monumentalnych budowli, ekonomicznych samochodów i wielu innych osiągnięć cywilizacji. Modelowanie komputerowe nie jest jednak zadaniem łatwym. Pomimo olbrzymich mocy obliczeniowych stosowanych obecnie w tej dziedzinie, niezmiernie istotne jest oszczędne zaprojektowanie modelu. Rzeczywistość fizyczna przejawia niebywały poziom skomplikowania, w wyniku czego nawet modelowanie tak prostej sytuacji jak ruch trzech ciał w polu grawitacyjnym przysparza wielu trudności. W modelowaniu kluczowe znaczenie ma opis matematyczny zjawiska. Opisy te są już w znacznej mierze gotowe, jest to dorobek fizyki. Najczęściej układy fizyczne są opisywane układami równań różniczkowych cząstkowych (zazwyczaj są to układy nieliniowe i nie posiadające ogólnych rozwiązań analitycznych). Modelowanie komputerowe polega na dyskretyzacji równania opisującego zagadnienie i rozwiązaniu go metodami numerycznymi. Stąd przy opracowywaniu modeli należy kontrolować ich stabilność oraz błąd stosowanych schematów numerycznych. Jednym z najczęściej stosowanych modeli numerycznych jest model przepływu fluidu (fluidem może być ciecz lub gaz), znanym pod nazwą CFD (Computational Fluid Dynamics). Wspomniane wyżej modele wykorzystywane są na przykład: w aerodynamice (przy projektowaniu samolotów, samochodów, promow kosmicznych) hydrodynamice (projektowanie zaawansowanych układów hydraulicznych), prognozowaniu pogody, modelowniu fal uderzeniowych, skutków eksplozji atomowych itd. Przepływ ściśliwego fluidu jest opisywany równaniem Naviera-Stokes'a, które najprawdopodobniej nie posiada ogólnego analitycznego rozwiązania przy żadnych warunkach początkowych. Równanie to jesteśmy w stanie zdyskretyzować i rozwiązać numerycznie.

W naszym projekcie SIMP będziemy modelować przepłw nieściśliwego fluidu w zamkniętej domenie na regularnej siatce kartezjańskiej, z prostym modelem konwekcji. Założenie nieściśliwości fluidu (co jest równoważne zerowej dywergencji pola prędkości) upraszcza równanie Naviera-Stokes'a i sprowadza je do równania Eulera. Uproszenie to nie zmniejsza użyteczności modelu. Zarówno ciecz jak i gaz w niewielkich skalach objętościowych oraz przy założeniu, że prędkość przepływu jest znacznie mniejsza od prędkości dźwięku w ośrodku, można uważać za nieściśliwe. Model taki oczywiście nie obejmuje fal uderzeniowych, do których modelowania potrzebne jest założenie ściśliwości. W naszym modelu zakładamy, że gęstość modelowanego medium jest stała i wynosi 1.

2 Wprowadzenie

2.1 Cele przedsięwzięcia

Pakiet SIMP - Studencki Iteracyjny Model Przepływu, ma zostać wykonany jako projekt studencki w ramach programowania zespołowego na Wydziale Matematyki i Informatyki Uniwersytetu Mikołaja Kopernika w Toruniu.

Celem projektu jest realizacja numerycznego modelu przepływu nieściśliwego fluidu w domenie kartezjańskiej wraz z prostym modelowaniem zjawiska konwekcji. Podstawą matematyczną modelu jest uproszczone równanie Naviera-Stokesa, które w tym przypadku sprowadza się do równania Eulera.

W tworzonym pakiecie powinien znajdować się:

- SIMP:Konstruktor - program generujący domenę obliczeniową,
- SIMP:Iterator - program, który rozwiązuje problem modelowania,
- SIMP:Prezenter - program prezentujący wyniki w postaci wykresów 3d.

2.2 Systemy zewnętrzne

Tworzona aplikacja współpracowała będzie z użytkownikiem, który będzie miał możliwość:

1. zaprojektowania parametrów modelu takich jak: gęstość siatki, wstawienie warunków brzegowych oraz wygenerowanie domeny obliczeniowej,
2. wyprodukowania za pomocą SIMP:Iteratora danych na temat stanu modelu,
3. wizualizacji danych wyprodukowanych przez SIMP:Iteratora za pomocą modułu zwanego SIMP:Prezenterem.

Ponadto, w przypadku korzystania przez użytkownika z kilku maszyn do różnych modułów projektu, wymagana jest metoda przesyłania danych między maszynami (e-mail, ftp, scp, www, ...). Ze względu na różnorodność sytuacji oraz konfiguracji sprzętowych i czasoprzestrzennych, w jakich mogą być używane moduły projektu, jak też założenie o potencjalnej możliwości tworzenia nowych modułów przez użytkownika końcowego, zdecydowaliśmy się pozostawić ten wybór użytkownikowi.

2.3 Ogólny opis wymagań

Pakiet powinien składać sie z 3 modułów:

1. moduł SIMP:Konstruktor: przed wykonaniem modelowania pozwala pozwoli ustalić gestość siatki oraz dodać warunki brzegowe. Aplikacja ta napisana będzie w języku java (ze względu na przenośność i prostotę).Do zobrazowania domeny obliczeniowej wykorzystane będą biblioteki: java3d i VisAD.

2. moduł SIMP:Iterator to centralny punkt pakietu. Program napisany w języku C++, ze względu na wymaganą wydajność, będzie wykorzystywał procedury biblioteki SLATEC. Wynikiem działania modułu będą dane opisujące stan modelu, zapisane w pliku tekstowym. Wizualizacja danych możliwa będzie dzięki kolejnemu modułowi SIMP:Prezenterowi.

3. moduł SIMP:Prezenter: wizualizuje dane wygenerowane przez moduł SIMP:Iterator. Forma wizualizacji będzie przejrzysta i przyjemna dla uzytkownika. Do tego celu wykorzystamy szeroko stosowaną do takich celów bibliotekę VisAD dostępną do javy.

Wszystkie części projektu będą korzystały jedynie z elementów dostępnych na licencji GPL lub udostępnianych bezpłatnie w zasobach internetowych.

3. Wymagania funkcjonalne

3.1 Moduł SIMP:Konstruktor

Nazwa funkcji: Nowy plik
Opis: Pozwala stworzyć nowy, pusty model.
Dane wejściowe:
Źródło danych wejściowych:
Wynik: W pamięci powstaje nowy model (SIMPEnvironment) o domyślnych parametrach i pustym zestawie obiektów.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne: Istniejący model zostaje wymazany z pamięci.
Powód: Umożlienie użytkownikowi rozpoczęcie konstrukcji od nowa

Nazwa funkcji: Otwórz plik
Opis: Funkcja pozwala wczytać plik z opisem sceny/modelu albo utworzony za pomocą SIMP:Konstruktora, albo wprowadzony ręcznie przez

użytkownika.
Dane wejściowe: Opis sceny
Źródło danych wejściowych: Plik w formacie XML
Wynik: Wczytanie opisu sceny.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne: Istniejący model zostaje wymazany z pamięci.
Powód: Umożliwienie użytkownikowi wczytanie uprzednio zapisanej pracy.

Nazwa funkcji: Zapisz plik
Opis: Funkcja pozwala zapisać stan sceny do postaci XML w celu dalszej obróbki w czasie następnej sesji lub przekazania pliku do SIMP:Iteratora
Dane wejściowe: Opis sceny znajdujący się w pamięci.
Źródło danych wejściowych: Formularz
Wynik: Powstaje plik z opisem sceny.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi przechowywania opisów sceny.

Nazwa funkcji: Dodaj obiekt
Opis: Pozwala tworzyć nowe elementy sceny.
Dane wejściowe:
Źródło danych wejściowych:
Wynik: W wewnętrznej reprezentacji modelu powstaje nowy obiekt o domyślnych parametrach, SIMP:Konstruktor przygotowuje się do zmiany parametrów obiektu.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne: Tracone są niezachowane parametry ostatnio przetwarzanego obiektu.
Powód: Umożliwienie użytkownikowi tworzenia komponentów modelu i lokalnych warunków początkowych.

Nazwa funkcji: Usuń obiekt
Opis: Pozwala usuwać wybrany obiekt ze sceny.
Dane wejściowe: numer/nazwa usuwanego obiektu
Źródło danych wejściowych: formularz
Wynik: Zostaje usunięty wybrany obiekt.
Warunek wstępny: Obiekt przeznaczony do usunięcia musi uprzednio istnieć.

Warunek końcowy:
Efekty uboczne: Graficzna reprezentacja sceny zostaje uaktualniona, formularz pokazuje parametry obiektu sąsiadującego z usuniętym.
Powód: Pozwala użytkownikowi zrezygnować z wcześniej utworzonego obiektu.

Nazwa funkcji: Duplikuj obiekt
Opis: Tworzy obiekt o identycznych właściwościach jak ostatnio wybrany.
Dane wejściowe: parametry ostatnio wybranego obiektu
Źródło danych wejściowych: reprezentacja wewnętrzna, formularz
Wynik: Zostaje utworzony nowy obiekt, nieznacznie przesunięty wobec poprzednika, SIMP:Konstruktor przygotowuje się do zmiany parametrów nowego obiektu.
Warunek wstępny: Zbiór obiektów w modelu musi być niepusty.
Warunek końcowy:
Efekty uboczne: Graficzna reprezentacja sceny zostaje uaktualniona.
Powód: Ułatwia użytkownikowi utworzenie większej ilości obiektów o podobnych właściwościach przez wielokrotną duplikację i modyfikację parametrów.

Nazwa funkcji: Zastosuj parametry
Opis: SIMP:Konstruktor opiera się na ciągłej modyfikacji parametrów obiektów w formularzu, funkcja „Zastosuj" pobiera dane z formularza i uaktualnia wewnętrzną reprezentację modelu.
Dane wejściowe: parametry obiektu
Źródło danych wejściowych: formularz parametrów obiektu
Wynik: Zostaje uaktualniona wewnętrzna reprezentacja wybranego modelu.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne: Graficzna reprezentacja sceny zostaje uaktualniona.
Powód: Umożliwienie użytkownikowi zatwierdzenia nowych parametrów i obejrzenia ich w postaci graficznej reprezentacji obiektu.

Nazwa funkcji: Anuluj parametry
Opis: Odwrotność funkcji „Zastosuj"
Dane wejściowe: parametry obiektu
Źródło danych wejściowych: wewnętrzna reprezentacja w pamięci
Wynik: Formularz zostaje wypełniony
Warunek wstępny:
Warunek końcowy:

Efekty uboczne: Zostają utracone niezatwierdzone parametry w formularzu.
Powód: Umożliwienie użytkownikowi powrót do ostatnio zatwierdzonych parametrów wybranego obiektu.

Nazwa funkcji: Przerysuj
Opis: Wymusza wprowadzenie do pamięci parametrów sceny z formularza i przerysowanie graficznej reprezentacji modelu.
Dane wejściowe: parametry sceny
Źródło danych wejściowych: formularz
Wynik: Zostaje uaktualniona graficzna reprezentacja sceny.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi natychmiastowego obejrzenia sceny w przypadku zmiany parametrów, przede wszystkim rozmiaru.

Nazwa funkcji: Włącz/wyłącz podgląd uproszczony
Opis: Funkcja włącza/wyłącza tryb wizualizacji o zmniejszonej (dwukrotnie) dokładności.
Dane wejściowe:
Źródło danych wejściowych:
Wynik: Wynik zmiany staje się widoczny dopiero po wybraniu funkcij „Przerysuj”
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Uproszczony podgląd przyspiesza aktualizację graficznej reprezentacji sceny, co zwiększa wydajność pracy przy modelach o dużej dokładności.

Nazwa funkcji: Wybierz obiekt
Opis: Zaznacza ustalony obiekt jako aktywny. Filozofia SIMP:Konstruktora obejmuje przetwarzanie jednego obiektu. Funkcja ta pozwala wybrać obiekt, który będzie podlegał modyfikacji.
Dane wejściowe: numer/nazwa obiektu
Źródło danych wejściowych: formularz (lista obiektów)
Wynik: Formularz z parametrami obiektu przyjmuje wartości parametrów nowo wybranego obiektu, w reprezentacji graficznej obiekt zostaje zaznaczony jako aktywny
Warunek wstępny: Zbiór obiektów musi być niepusty
Warunek końcowy:

Efekty uboczne: Tracone są niezatwierdzone wartości w formularzu dotyczące poprzednio wybranego obiektu.
Powód: Pozwala użytkownikowi wybrać obiekt, który chce modyfikować.

Nazwa funkcji: Zmień kolor
Opis: Każdy obiekt w SIMP:Konstruktorze może mieć jeden z kilku predefiniowanych kolorów. Kolory służą identyfikacji wizualnej obiektów i nie mają wpływu na działanie modelu.
Dane wejściowe: numer/nazwa koloru
Źródło danych wejściowych: formularz
Wynik: Zostaje zmieniony kolor obiektu w wewnętrznej reprezentacji modelu, reprezentacja graficzna zostaje uaktualniona.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwia użytkownikowi ustalenie koloru obiektu, w celu łatwiejszej identyfikacji wizualnej w reprezentacji graficznej modelu.

Nazwa funkcji:
Opis:
Dane wejściowe:
Źródło danych wejściowych:
Wynik:
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód:

3.2 Moduł SIMP:Iterator

Nazwa funkcji: Wczytaj opis modelu
Opis: Tworzy model na podstawie opisu
Dane wejściowe: opis modelu
Źródło danych wejściowych: plik XML
Wynik: Zostaje utworzona w pamięci siatka dyskretyzująca problem opisany w pliku. Wartości w węzłach stają się wymaganymi warunkami brzegowymi.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne: Parametry metod numerycznych zostają ustalone na zadane w opisie sceny.

Powód: Pozwala przetworzyć opis sceny w postaci pliku XML na dyskretny model podlegający obróbce.

Nazwa funkcji: Wczytaj model dyskretny
Opis: Pozwala wczytać uprzednio zapisany stan siatki.
Dane wejściowe: stan siatki
Źródło danych wejściowych: plik tekstowy
Wynik: Zostaje utworzona w pamięci siatka zgodnie z uprzednio wczytanym opisem, wartości początkowe ustawione są na wczytane z pliku tekstowego.
Warunek wstępny: Uprzednio musi być wczytany opis modelu.
Warunek końcowy:
Efekty uboczne:
Powód: Pozwala dokonać kolejnej iteracji uprzednio przetwarzanego modelu. Szczególnie przydatne jest to przy badaniu kolejnych etapów rozwoju zjawiska.

Nazwa funkcji: Iteruj
Opis: Dokonuje obliczeń zgodnie z parametrami
Dane wejściowe: siatka
Źródło danych wejściowych: wewnętrzna reprezentacja
Wynik: Dane w modelu dyskretnym są uaktualniane zgodnie z wynikiem obliczeń numerycznych.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Pozwala dokonać obliczeń zgodnie z parametrami wczytanymi z opisu sceny.

Nazwa funkcji: Uruchom model
Opis: Wykonuje wymaganą dla modelu ilość iteracji
Dane wejściowe: wymagana ilość iteracji
Źródło danych wejściowych: plik XML
Wynik: Wykonywana jest funkcja „Iteruj" określoną ilość razy
Warunek wstępny:
Warunek końcowy:
Efekty uboczne: Wyniki poszczególnych iteracji są tracone na rzecz kolejnych.
Powód: Pozwala wykonać obliczenia na tym samym modelu określoną ilość razy.

Nazwa funkcji: Zapisz stan

Opis: Zapisuje aktualny stan siatki
Dane wejściowe: siatka
Źródło danych wejściowych: wewnętrzna reprezentacja, obliczenia numeryczne
Wynik: Tworzony jest plik na dysku, zawierający dane numeryczne opisujące siatkę.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Pozwala zapisać model w postaci umożliwiającej wizualizację i ewentualne kolejne iteracje.

Nazwa funkcji:
Opis:
Dane wejściowe:
Źródło danych wejściowych:
Wynik:
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód:

3.3 Moduł SIMP:Prezenter

Nazwa funkcji: Wczytaj plik
Opis: Wczytuje plik w formacie txt, będący wynikiem działania SIMP:-Iteratora.
Dane wejściowe: Plik txt zawierający informacje na temat stanu modelu.
Źródło danych wejściowych: plik
Wynik: Wizualizacja danych zamieszczonych w pliku.
Warunek wstępny: Dane w pliku zapisane są w ustalonym formacie, by uniknąć wieloznaczności interpretacji.
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi wizualizacji wyprodukowanych danych przez SIMP:Iteratora na temat modelu.

Nazwa funkcji: Zapisz wykres
Opis: Pozwala zapisać wizualizowany wykres.
Dane wejściowe: Wizualizowane dane.
Źródło danych wejściowych: SIMP:Prezenter

Wynik: Plik z zapisanym wykresem.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi zapisu zwizualizowanych danych.

Nazwa funkcji: Ustal dokładność wczytywanych danych
Opis: Pozwala zinterpolować wczytywane dane do siatki posiadającej inne wymiarach niż pierwotne.
Dane wejściowe:
Źródło danych wejściowych: plik
Wynik: Ustalona zostaje dokładność wczytywanych danych.
Warunek wstępny: Wczytany plik z danymi odnośnie stanu modelu
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytnikom aplikacji uzyskania porządanej dokładności prezentowanych danych.

Nazwa funkcji: Połącz/rozłącz okna
Opis: SIMP:Prezenter ma dwa, widoczne jednocześnie, okna w których można oglądać dane na temat modelu. Funkcja pozwala odpowiednio sterować obydwoma oknami jednocześnie (połącz okna), badź każdym oddzielnie (rozłącz okna). Pod pojęciem sterowania rozumiemy wszelkie rotacje modelu.
Dane wejściowe: Źródło danych wejściowych: użytkownik
Wynik: Połączenie lub rozdzielenie okien.
Warunek wstępny: Wczytany zostal plik txt z danymi (wynik działania SIMP:Iteratora).
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi szczegółowej analizy wyników otrzymanych dla modelu.

Nazwa funkcji: Ustaw opcje rysowania danych.
Opis: Funkcja pozwala ustawić opcje rysowania danych. Użytkownik ma możliwość zmiany (włączenia bądź wyłączenia) następujących opcji: antialiasing, lepsza jakość wizualizacji, pełen ekran, połaczenie okien.
Dane wejściowe:
Źródło danych wejściowych: odpowiedni formularz (combo box)
Wynik: Ustawione zostają opcje rysowania danych.
Warunek wstępny: Wczytany plik txt zawierający dane odnośnie

stanu modelu.
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie kontroli przez użytkownika sposobu wizualizacji danych.

Nazwa funkcji: Rysuj wykres
Opis: Rysyje wybrany wykres. Użytkownik ma możliwość wyboru pomiędzy następującymi wykresami: temperatury, ciśnienia, pola prędkości.
Dane wejściowe:
Źródło danych wejściowych: odpowiedni formularz (combo box)
Wynik: Narysowany zostaje wybrany wykres.
Warunek wstępny: Wczytany plik txt z danymi odnośnie stanu modelu.
Warunek końcowy:
Efekty uboczne:
Powód: Pozwala użytkownikowi obejrzeć wyniki na temat modelu wyprodukowane przez SIMP:Iteratora.

Nazwa funkcji: Ustaw opcje wykresu temperatury
Opis: Funkcja umożliwia wybór prezentowanych danych:

1. pola temperatury
2. gradientu pola temperatury
3. modułu gradientu pola temperatury
4. laplasjanu pola temperatury

Dane wejściowe:
Źrodło danych wejściowych: formularz (combo box)
Wynik: Prezentacja wybranych danych.
Warunek wstępny: Wybrany został do wizualizacji wykres temperatury.
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi szczegółowej obserwacji uzyskanych wyników odnośnie modelowanej rzeczywistości.

Nazwa funkcji: Ustaw opcje wykresu ciśnienia
Opis: Funkcja umożliwia wybór prezentowanych danych:

1. pola ciśnienia

2. gradientu pola ciśnienia
3. modułu gradientu pola ciśnienia
4. laplasjanu pola ciśnienia

Dane wejściowe:
Źródło danych wejściowych: formularz (combo box)
Wynik: Prezentacja wybranych danych.
Warunek wstępny: Wybrany został do wizualizacji wykres ciśnienia.
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi szczegółowej obserwacji uzyskanych wyników odnośnie modelowanej rzeczywistości.

Nazwa funkcji: Ustaw opcje wykresu pola prędkosci
Opis: Funkcja umożliwia wybór prezentowanych danych:

1. pola predkości
2. modułu pola prędkości
3. dywergencji pola prędkości
4. rotacji pola prędkości
5. modułu rotacji pola prędkości

Dane wejściowe:
Źródło danych wejściowych: formularz (combo box)
Wynik: Prezentacja wybranych danych.
Warunek wstępny: Wybrany został do wizualizacji wykres pola prędkości.
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi szczegółowej obserwcji uzyskanych wyników odnośnie modelowanej rzeczywistości.

Nazwa funkcji: Zmień rozkład kolorów
Opis: Pozwala zmienić rozkład kolorów wyświetlanego wykresu, wybierając jeden z sześciu dostępnych zestawów.
Dane wejściowe:
Źródło danych wejściowych: formularz (lista przycisków)
Wynik: Zmiana rozkładu kolorów.
Warunek wstępny:
Warunek końcowy:

Efekty uboczne:
Powód: Umożliwienie użytkownikowi zmiany rozkładu kolorów w rysowanym wykresie.

Nazwa funkcji: Ustaw zestawy kanału alpha
Opis: Pozwala zmienić rozkład kanału alpha wyświetlanego wykresu, wybierając z czterech dostępnych zestawów.
Dane wejściowe:
Źródło danych wejściowych: formularz (lista przycisków)
Wynik: Zmiana rozkładu kanału alpha.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi zmiany rozkładu kolorów w rysowanym wykresie.

Nazwa funkcji: Zmień mapę kolorów wykresu
Opis: Pozwala zmienić mapę kolorów wyświetlanego wykresu.
Dane wejściowe:
Źródło danych wejściowych: formularz
Wynik: Zmiana mapy kolorów.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi zmiany rozkładu kolorów w rysowanym wykresie.

Nazwa funkcji: Ustaw zakres danych
Opis: Ustawia zakres wizualizowanych danych, rysując jedynie te dane, których wartości należą do zadanego przedziału. Istnieje mozliwość ustawienia zakresu danych dla każdej osi (X, Y, Z).
Dane wejściowe:
Źródło dddanych wejściowych: formularz (suwak)
Wynik: Zmiana zakresu prezentowanych danych.
Warunek wstępny:
Warunek końcowy:
Efekty uboczne:
Powód: Umożliwienie użytkownikowi szczegółowej analizy prezentowanych wyników.

4. Wymagania niefunkcjonalne.

Stawiamy na wieloplatformowość i łatwość ewentualnej rozbudowy projektu o kolejne moduły, więc wszędzie, gdzie to możliwe i praktyczne, stosujemy przenośne technologie i formaty zapisu. Format opisu sceny powinien być oparty o XML, z kodowaniem polskich znaków, które będzie czytelne na różnych platformach i w różnych językach programowania. Format zapisu stanu modelu będzie tekstowy ze względu na oszczędność miejsca, przenośność i, w przeciwieństwie do formatu opisu sceny, brak potrzeby bezpośredniego oglądania przez użytkownika.

Moduły posiadające graficzny interfejs użytkownika powinny być napisane w języku Java. W miarę możliwości interfejsy tych modułów powinny być ujednolicone, zarówno pod względem ogólnego wyglądu, obsługi i użytych metod wizualizacji.

Wizualizacja powinna umożliwiać oglądanie modelu w postaci kolorowego trójwymiarowego wykresu gęstości (*density plot*), jak też pola wektorowego. Ze względu na powyższe założenie, moduł konstrukcyjny będzie korzystał z tej samej metody wizualizacji.

W całym projekcie decydujemy się korzystać z wolnodostępnych technologii, co dotyczy użytych bibliotek, kompilatorów, systemu wizualizacji i fragmentów kodu. Jeśli jest to możliwe, korzystamy z oprogramowania i kodu na licencji GPL. Decyzja ta podyktowana jest naukowym charakterem projektu, różnorodnością dostępnych na otwartej licencji narzędzi oraz łatwością dopasowania otwartych bibliotek programistycznych do potrzb projektu.