Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Национальный исследовательский университет «МЭИ»

|  |  |  |  |
| --- | --- | --- | --- |
| Институт: | *ИРЭ* | Кафедра: | *Радиотехнических систем* |
| Специальность: | | *11.05.01 Радиоэлектронные системы и комплексы* | |

**ОТЧЕТ по практике**

|  |  |
| --- | --- |
| **Наименование практики:** | Производственная практика: научно-исследовательская работа |

**СТУДЕНТ**



|  |  |
| --- | --- |
|  | / Тасканов В.Е. / |
| *(подпись )* | (*Фамилия и инициалы*) |

|  |  |
| --- | --- |
| Группа | ЭР-15-16 |
|  | *(номер учебной группы)* |

**ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ПРАКТИКЕ**

|  |
| --- |
|  |
| *(отлично, хорошо, удовлетворительно, неудовлетворительно, зачтено, не зачтено)* |

|  |  |
| --- | --- |
|  | / Куликов Р.С. / |
| *(подпись )* | (*Фамилия и инициалы члена комиссии*) |

|  |  |
| --- | --- |
|  | / Шатилов А.Ю. / |
| *(подпись )* | (*Фамилия и инициалы члена комиссии*) |

**Москва**

**2021**

СПИСОК ИСПОЛНИТЕЛЕЙ

студент Тасканов В.Е.

**Содержание**

[**ГЛАВА 1 ДОБАВЛЕНИЕ РАСЧЕТА КООРДИНАТ НС** 5](#_Toc73991732)

[1.1. Алгоритм расчета для ГНСС GPS 6](#_Toc73991733)

[1.1.2. Алгоритм расчета координат 8](#_Toc73991734)

[1.3. Алгоритм расчета для ГНСС ГЛОНАСС 10](#_Toc73991735)

[1.3.2. Алгоритм расчета координат 12](#_Toc73991736)

[1.4. Алгоритм расчета ионосферной погрешности 15](#_Toc73991737)

[**ГЛАВА 2 РЕАЛИЗАЦИЯ АЛГОРИТМОВ В ПРОГРАММЕ** 18](#_Toc73991738)

[2.1. Скачивание файла 18](#_Toc73991739)

[2.2. Обработка файла 18](#_Toc73991740)

[2.2.1. Обработка файла GPS 18](#_Toc73991741)

[2.2.1. Обработка файла ГЛОНАСС 18](#_Toc73991742)

[2.3. Расчет координат 19](#_Toc73991743)

[2.4. Расчет времени 19](#_Toc73991744)

[2.3. Изменение интерфейса программы 20](#_Toc73991745)

[2.4. Необходимые файлы для сборки проекта 21](#_Toc73991746)

[2.5. Профилирование проекта 24](#_Toc73991747)

[**ГЛАВА 3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ** 26](#_Toc73991748)

[**ЗАКЛЮЧЕНИЕ** 32](#_Toc73991749)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 32](#_Toc73991750)

[**ПРИЛОЖЕНИЕ** 33](#_Toc73991751)

Введение

Спутниковые радионавигационные системы (СРНС) являются самыми точными системами по определению координат потребителя. Они стали важной частью в различных сферах нашей жизни. Наиболее распространенными являются системы ГЛОНАСС (Россия), GPS (США), Galileo (Евросоюз).

В 10 семестре стояла **цель работы** – откорректировать предыдущую программу, добавить функцию расчета координат НКА по данным альманахов систем ГЛОНАСС и GPS, добавить функцию расчета ионосферных погрешностей по углу возвышении спутника и текущему календарному времени года, и с учетом всех изменений произвести оценку координаты потребителя с учетом ошибок SISRE.

В рамках данной цели решаются следующие задачи:

1. Изменения предыдущего алгоритма определения координат НКА
2. Теоретическое изучение вносимых погрешностей от ионосферы
3. Добавления алгоритма расчета ионосферных погрешностей по углу возвышении спутника и текущему календарному времени года
4. Нахождения оценки координаты потребителя с учетом ошибок SISRE.

# **ГЛАВА 1 ДОБАВЛЕНИЕ РАСЧЕТА КООРДИНАТ НС**

Вкоторые рассчитываются по данным альманахам соответствующих группировок для созвездий ГЛОНАСС и GPS.

Алгоритм расчета координат в программе будет следующий:

* Скачиваем файл с данными альманаха,
* Обрабатываем файл,
* Рассчитываем координаты

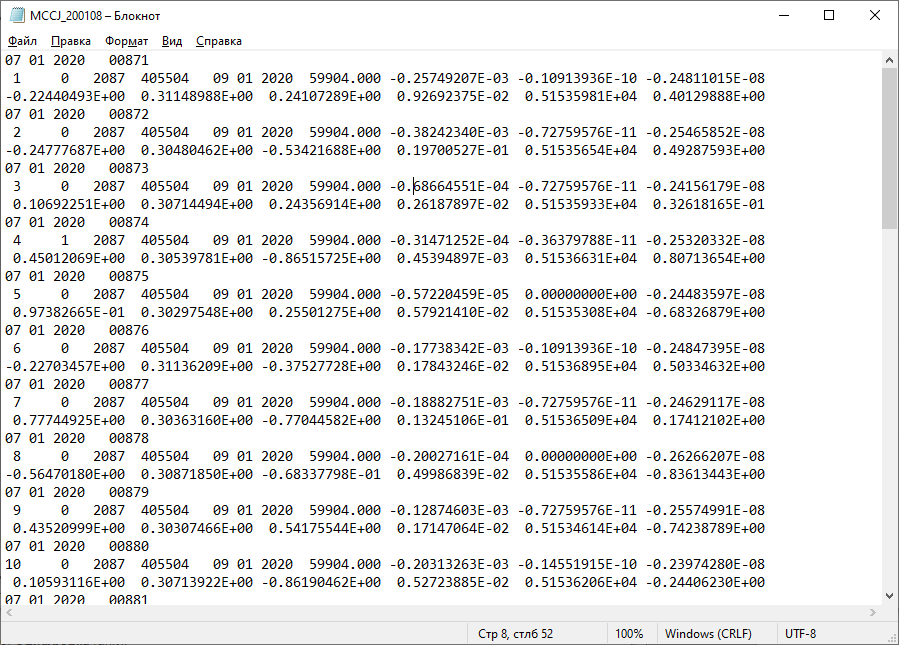
Файл будет скачивать с сервера «инфомационно-аналитечкского центра коррдинатно-временного и навигационного обеспечения, по адресу: «ftp://ftp.glonass-iac.ru/MCC/ALMANAC/», где далее следует выбор года и даты нужного альманаха. На сервере содержится два файла с разными расширениями – agp ( для ГНСС GPS) и agl (для ГНСС ГЛОНАСС).

Скачав файл необходимо его оцифровать (перенести нужные данные в программу для реализации последующих алгоритмов).

Оцифровка файлов для GPS и ГЛОНАСС значительно отличается, поэтому разделим их на разные функции.

## 1.1. Алгоритм расчета для ГНСС GPS

В файле с расширением – agp, содержатся альманахи, записанные в виде строк:



1. Пример скаченного файла с расширением agp

Где, строка 1, соответствует:

|  |  |
| --- | --- |
| 1 | Число получения альманаха |
| 2 | месяц получения альманаха |
| 3 | год получения альманаха |
| 4 | время получения альманаха от начала суток, с UTC |

Строка 2

|  |  |
| --- | --- |
| 1 | номер PRN |
| 2 | обобщенный признак здоровья (0 - здоров) |
| 3 | неделя GPS (альманаха) (номер недели полный) |
| 4 | время недели GPS, с (альманаха) (количество секунд от начала недели) |
| 5 | число |
| 6 | месяц |
| 7 | год |
| 8 | время альманаха, с |
| 9 | поправка времени КА GPS относительно системного времени, с, |
| 10 | скорость поправки времени КА GPS относительно системного времени, с/с |
| 11 | Om0 - скорость долготы узла, полуциклы/c, |

Строка 3

|  |  |
| --- | --- |
| 1 | Om0 - долгота узла, полуциклы, |
| 2 | I - наклонение, полуциклы, |
| 3 | w - аргумент перигея, полуциклы, |
| 4 | E – эксцентриситет, |
| 5 | SQRT(A) - корень из большой полуоси, м\*\*0.5, sqrt(A0) |
| 6 | M0 - средняя аномалия, полуциклы, |

1.1.2. Алгоритм расчета координат

Далее полученные значения подставляются в алгоритм расчета координат, который возьмем из ИКД GPS:

1. Определим время, отсчитываемое от опорной эпохи эфемерид:



1. Определим среднее движение:



1. Определим скорректированное среднее движение:



1. Определим среднюю аномалию:



1. Решим уравнение Кеплера минимум 3-мя итерациями и определим :



1. Определим истинную аномалию:



1. Определим скорректированный радиус орбиты спутника:





1. Определим аргумент широты:





1. Определим координаты НС в орбитальной плоскости:



1. Определим скорректированную долготу восходящего узла  определяется из соотношения:





1. Определим скорректированное наклонение орбиты спутника

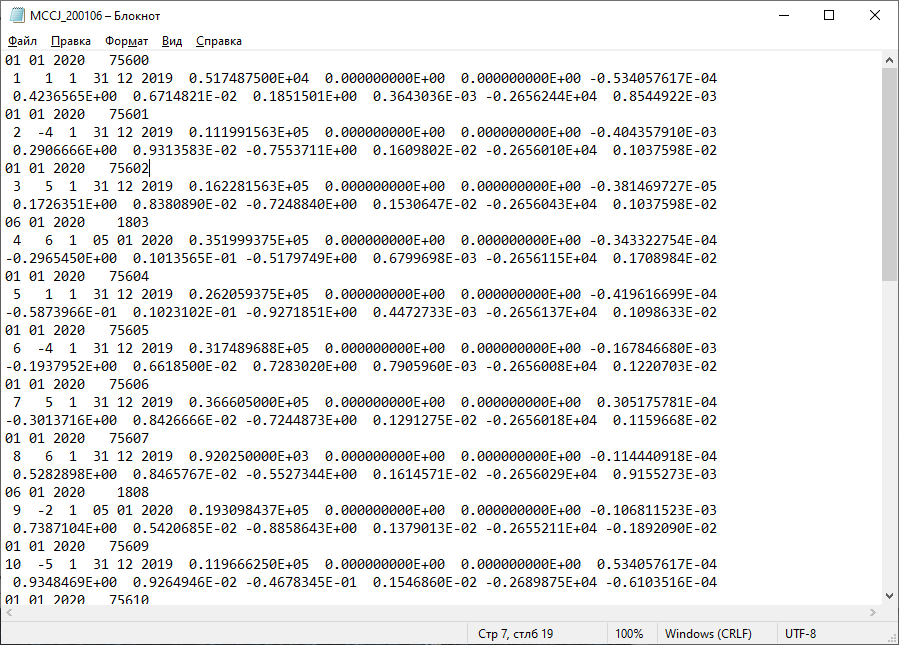


1. Определим координаты НС в геоцентрической системе координат:



1.3. Алгоритм расчета для ГНСС ГЛОНАСС

В файле с расширением – agl, содержатся альманахи, записанные в виде строк:



1. Пример скаченного файла с расширением agl

Где, строка 1, соответствует:

|  |  |
| --- | --- |
| 1 | Число получения альманаха |
| 2 | месяц получения альманаха |
| 3 | год получения альманаха |
| 4 | время получения альманаха от начала суток, с UTC |

Строка 2

|  |  |
| --- | --- |
| 1 | номер КА в группировке |
| 2 | номер частотного слота (-7 - 24) |
| 3 | признак здоровья по альманаху (0 - 1) |
| 4 | число |
| 5 | месяц |
| 6 | год |
| 7 | время прохождения первого узла, на которое все дано, с |
| 8 | поправка ГЛОНАСС-UTC, с |
| 9 | поправка GPS-ГЛОНАСС, с |
| 10 | поправка времени КА ГЛОНАСС относительно системного времени, с |

Строка 3

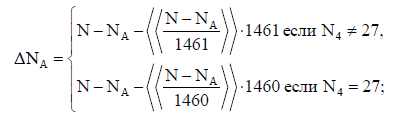
|  |  |
| --- | --- |
| 1 | Lam - долгота узла, полуциклы |
| 2 | dI - коррекция наклонения, полуциклы |
| 3 | w - аргумент перигея, полуциклы |
| 4 | E - эксцентриситет |
| 5 | dT - поправка к драконическому периоду, с |
| 6 | dTT - поправка к драконическому периоду, с/виток |

1.3.2. Алгоритм расчета координат

Далее полученные значения подставляются в алгоритм расчета координат, который возьмем из ИКД ГЛОНАСС:

1. Определяется интервал прогноза в секундах:



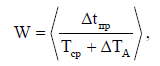
Где: 

N – календарный номер суток внутри четырехлетнего периода, начиная с високосного года, на которых находится заданный момент времени ti в секундах по шкале МДВ;

NA – календарный номер суток по шкале МДВ внутри четырехлетнего интервала, передаваемый НКА в составе неоперативной информации;

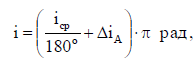
 – вычисление целого, ближайшего к x.

1. Рассчитывается количество целых витков W на интервале прогноза:

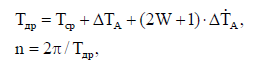


где  выделение целой части x;

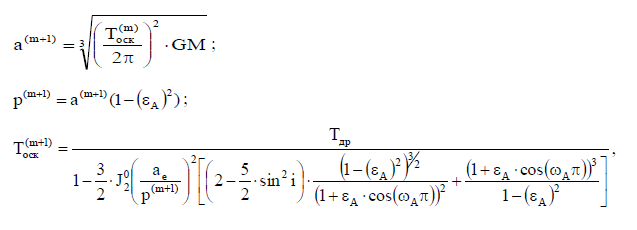
1. Определяется текущее наклонение:



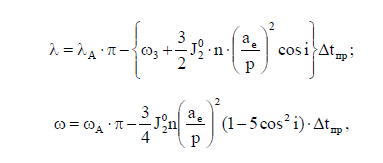
1. Определяются средний драконический период на витке W+1 и среднее движение:



1. Методом последовательных приближений m = 0, 1, 2… рассчитывается большая полуось орбиты a:



1. Определяются текущие значения долготы восходящего узла орбиты и аргумента перигея с учетом их векового движения под влиянием сжатия Земли:



1. Рассчитывается значение средней долготы на момент прохождения текущего восходящего узла:



Где 

1. Определяется текущее значение средней долготы НКА:



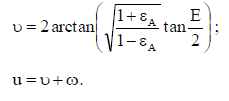
1. Определяется эксцентрическая аномалия путем решения уравнения Кеплера



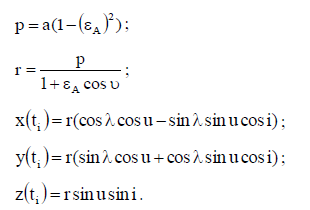
Как правило, используется схема последовательных приближений m = 0, 1, 2, и т.д.:



1. Вычисляются истинная аномалия и аргумент широты НКА u:



1. Рассчитываются координаты центра масс НКА в геоцентрической прямоугольной пространственной системе координат:



## 1.4. Алгоритм расчета ионосферной погрешности

Воспользуемся алгоритмом расчета из ИКД GPS

Модель коррекции модели ионосферы



Определим 



Где  - коэффициенты кубического уравнения, представляющие амплитуду вертикальной задержки

Определим фазу



Определим 



Где  - коэффициенты кубического уравнения, представляющие период модели

Определим коэффициент наклона



Где  - угол возвышения между пользователем и спутником

Определим геомагнитную широту земной проекции точки пересечения ионосферы (средняя высота ионосферы, предполагаемая 350 км)

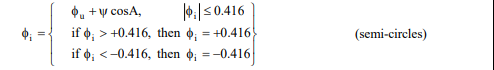


Определим геодезическая долгота земной проекции точки пересечения ионосферы



Определим геодезическая широта земной проекции точки пересечения ионосферы





Определим центральный угол Земли между положением пользователя и проекцией на землю точки пересечения ионосферы



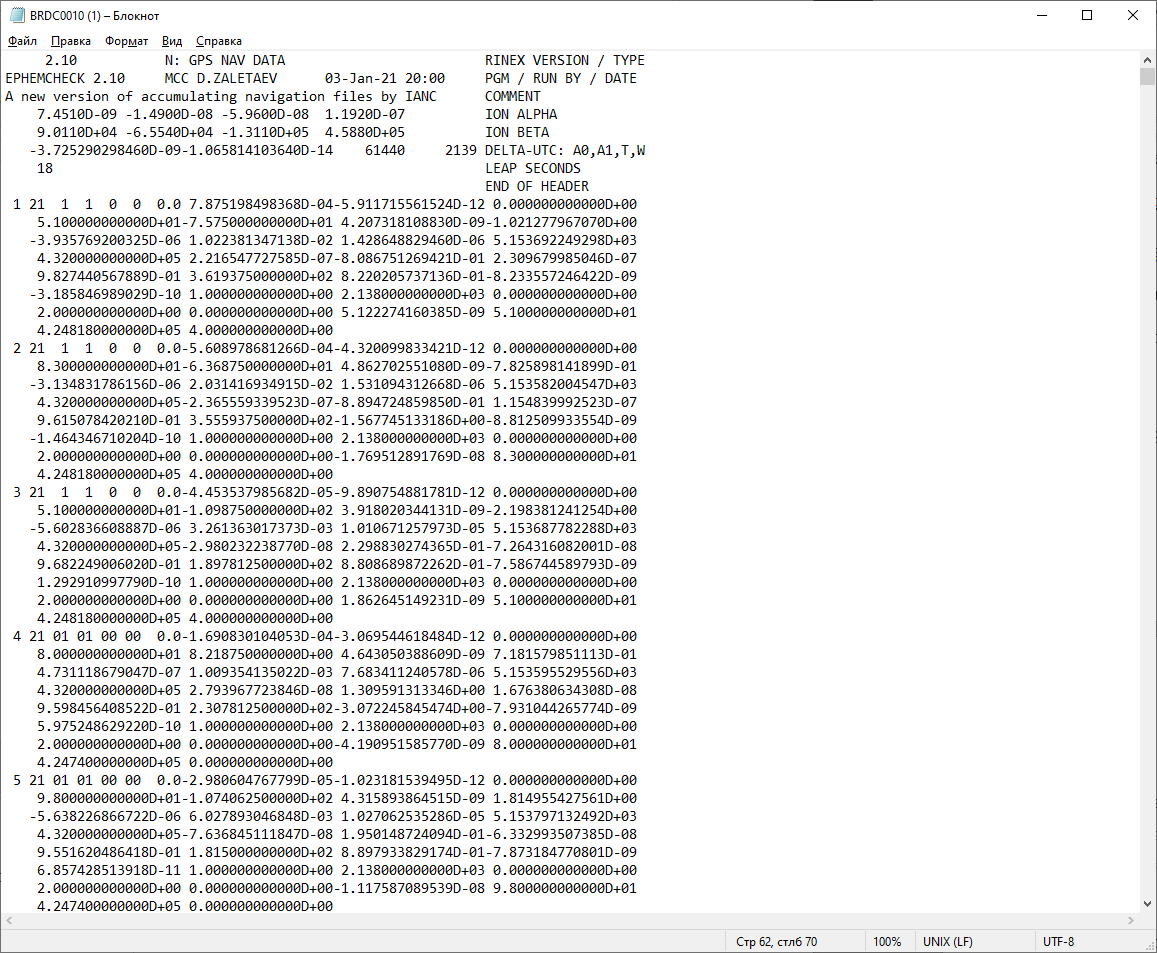
Определим локальное время:



Где ,

 - вычисленное системное время приемника

Значения коэффициентов ,берется из файла, который будем скачивать с сервера «инфомационно-аналитечкского центра коррдинатно-временного и навигационного обеспечения, по адресу: «ftp://ftp.glonass-iac.ru/MCC/BRDC/», расширение файла 21n



1. Пример скаченного файла с расширением 21n

Необходимые нам коэффициенты расположены на 4 и 5 строчках.

# **ГЛАВА 2 РЕАЛИЗАЦИЯ АЛГОРИТМОВ В ПРОГРАММЕ**

2.1. Скачивание файла

Для скачивания файлов модернизируем раннее созданный алгоритм «download» и для удобства последующих вызовов перенесем его в отдельный файл функции, который назовем: «FTPdownload», на вход которой подается разные пути и названия файла.

Функция содержит заголовочный файл – «FTPdownload.h», в котором хранятся применяемые классы и методы, а также файл с кодом реализации скачивания файла – «FTPdownload.CPP»

## 2.2. Обработка файла

Для обработки файлов также создадим отдельные функции, для ГЛОНАСС – «parserGLNS», а для GPS – «parserGPS»

2.2.1. Обработка файла GPS

Алгоритм обработки файла строится на методе «fscanf», которая обрабатывает последовательно каждое заданное значение, далее переносим полученные значения в массив значений «almanax\_GPS».

Функция содержит заголовочный файл – «parserGPS.h», в котором хранятся применяемые классы и методы, а также файл с кодом реализации обработки файла – «parserGPS.С»

### 2.2.1. Обработка файла ГЛОНАСС

Алгоритм обработки файла строится на методе «fscanf», которая обрабатывает последовательно каждое заданное значение, далее переносим полученные значения в массив значений «almanax\_GLNS».

Функция содержит заголовочный файл – «parserGLNS.h», в котором хранятся применяемые классы и методы, а также файл с кодом реализации обработки файла – «parserGLNS.С»

## 2.3. Расчет координат

Расчет координат для ГЛОНАСС и GPS выведем также в отдельные функции.

Для ГЛОНАСС функция принимает название – «ephemeridsGLNS», расчет соответствует формулам из п.1.3.2;

Функция содержит заголовочный файл – «ephemeridsGLNS.h», в котором хранятся применяемые классы и методы, а также файл с кодом реализации обработки файла – «ephemeridsGLNS.cpp»

Для GPS функция имеет название – «ephemerids», расчет соответствует формулам из п.1.1.2;

Функция содержит заголовочный файл – «ephemerids.h», в котором хранятся применяемые классы и методы, а также файл с кодом реализации обработки файла – «ephemerids.cpp»

## 2.4. Расчет времени

В процессе расчета координат возникнет проблема – получения времени расчета на которое нужно спрогнозировать координаты.

Для этого запишем класс – «timeCalc», в котором будет производиться перерасчет времени в нужный формат для трех ГНСС – ГЛОНАСС, GPS и GALILEO.

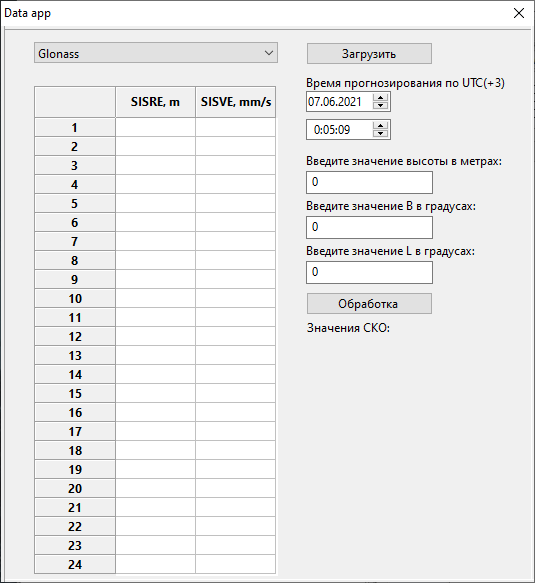
Для создания класса необходимо подать начальные значения: число, месяц, год, часы, минуты, секунды и миллисекунды.

Далее начальные значения преобразуются в секунды, с помощью встроенной библиотеки «ctime», а также подсчитывается количество поправок ко времени, для расчета в системе GPS и GALILEO.

В классе имеется три функции расчета времени:

* «timeGLNS» - для расчета времени в системе ГЛОНАСС,
* «timeGPS» - для расчета времени в системе GPS,
* «timeGLL» - для расчета времени в системе GALILEO.

2.3. Изменение интерфейса программы



1. Изменённый интерфейс программы

Добавил две новых формы:

* Ввод даты

Ввод даты производится с помощью встроенной формы в библиотеку wxWidgets – wxDatePickerCtrl, который реализован в виде небольшого окна, показывающего текущую дату, элемент управления можно редактировать с помощью клавиатуры, а также с помощью мышки

* Ввод времени

Ввод даты производится с помощью встроенной формы в библиотеку wxWidgets – wxTimePicerCtrl, который реализован в виде небольшого окна, показывающего текущее время, элемент управления можно редактировать с помощью клавиатуры, а также с помощью мышки

2.4. Необходимые файлы для сборки проекта

К отчету прикреплены 10 приложений, в которых содержатся основные файлы кода программы, необходимые для сборки проекта. Так файл: “dataMain.cpp”, код описан в приложение 3, содержит в себе основной алгоритм настройки окна приложения, с помощью библиотек wxWidgets, также содержит алгоритм скачивания файла с сервера и заполнения таблицы данными SISRE и SISVE, путь к файлу: “…\data\dataMain.cpp”.

Файл: “parser.c”, код описан в приложение 4, содержит в себе обработку скаченного файла с сервера, с помощью алгоритма из приложения 1 [2]. Под обработкой подразумевается фильтрование нужной нам информации – значения SISRE и SISVE для определенного спутника, путь к файлу: “…\data\parser.c”.

Файл: “parser.h”, код описан в приложение 5, содержит в себе обработчик массива SISerr, для использования этого массива в приложение 1 [2], данный обработчик необходим, так как приложение 2 написано на языке «С», а приложение 1 на языке «С++», путь к файлу: “…\data\parser.h”.

Файл: “data.cbp”, код описан в приложение 6 – это необходимый файл для сборки проекта, в котором прописан используемый компилятор, библиотеки, а также все необходимые заголовочные файлы, путь к файлу: “…\data \data.cbp”.

Файл: “dataMain.h”, код описан в приложение 7 – это заголовочный файл, в котором хранятся применяемые классы и методы, путь к файлу: “…\data\dataMain.h”.

Файл “xyz2enu.cpp”, код описан в приложение 8, в файле реализована функция перевода из геодезической системы координат в топоцентрические координаты (ENU), путь к файлу: “…\data \xyz2enu.cpp”

Файл “xyz2enu.h”, код описан в приложение 9 - это заголовочный файл, в котором объявляются применяемые классы и методы для функции перевода из геодезической системы координат в топоцентрические координаты (ENU) : “…\data \xyz2enu.h”

Файл “ephemeridsGLNS.cpp”, код описан в приложение 10, в файле реализована функция описание движения спутников ГЛОНАСС по орбитам и нахождения координат спутников ГЛОНАСС в определенный момент, путь к файлу: “…\data \ephemeridsGLNS.cpp ”

Файл “ ephemeridsGLNS.h”, код описан в приложение 11 - это заголовочный файл, в котором объявляются применяемые классы и методы для функции “ephemerids”, путь к файлу: “…\data \ ephemeridsGLNS.h ”

Файл “angle.cpp”, код описан в приложение 12, в файле реализована функция расчета видимости спутников, путь к файлу: “…\data \angle.cpp ”

Файл “ angle.h ”, код описан в приложение 13 - это заголовочный файл, в котором объявляются применяемые классы и методы расчета видимости спутников, путь к файлу: “…\data \ angle.h ”

Файл: “datadiaslog.wxs”, код описан в приложение 14 – это файл описания графического пользовательского интерфейса для плагина wxSmith, путь к файлу: “…\data\ wxsmith\datadiaslog.wxs”.

Файл “timeCalc.cpp”, код описан в приложение 15, в файле реализован класс перевода времени для трех систем СРНС, путь к файлу: “…\data \timeCalc.cpp ”

Файл “timeCalc.h”, код описан в приложение 16 - это заголовочный файл, в котором объявляются применяемые классы и методы для класса перевода времени для трех систем СРНС : “…\data \timeCalc.h”

Файл “FTPdownload.cpp”, код описан в приложение 17, в файле реализована функция скачивания файлов с сервера, путь к файлу: “…\data \FTPdownload.cpp ”

Файл “ FTPdownload.h ”, код описан в приложение 18 - это заголовочный файл, в котором объявляются применяемые классы и методы скачивания файлов с сервера, путь к файлу: “…\data \ FTPdownload.h ”

Файл: “parserGLNS.c”, код описан в приложение 19, содержит в себе обработку скаченного файла с сервера. Под обработкой подразумевается фильтрование нужной нам информации – значения SISRE и SISVE для определенного спутника, путь к файлу: “…\data\parserGLNS.c”.

Файл: “ parserGLNS.h”, код описан в приложение 20, содержит в себе обработчик массива SISerr, данный обработчик необходим, так как приложение 2 написано на языке «С», а приложение 1 на языке «С++», путь к файлу: “…\data\parserGLNS.h”.

Файл: “parserGPS.c”, код описан в приложение 21, содержит в себе обработку скаченного файла с сервера. Под обработкой подразумевается фильтрование нужной нам информации – значения SISRE и SISVE для определенного спутника, путь к файлу: “…\data\parserGPS.c”.

Файл: “ parserGPS.h”, код описан в приложение 22, содержит в себе обработчик массива SISerr, данный обработчик необходим, так как приложение 2 написано на языке «С», а приложение 1 на языке «С++», путь к файлу: “…\data\parserGPS.h”.

Файл “ephemerids.cpp”, код описан в приложение 10, в файле реализована функция описание движения спутников GPS по орбитам и нахождения координат спутников GPS в определенный момент, путь к файлу: “…\data \ephemerids.cpp ”

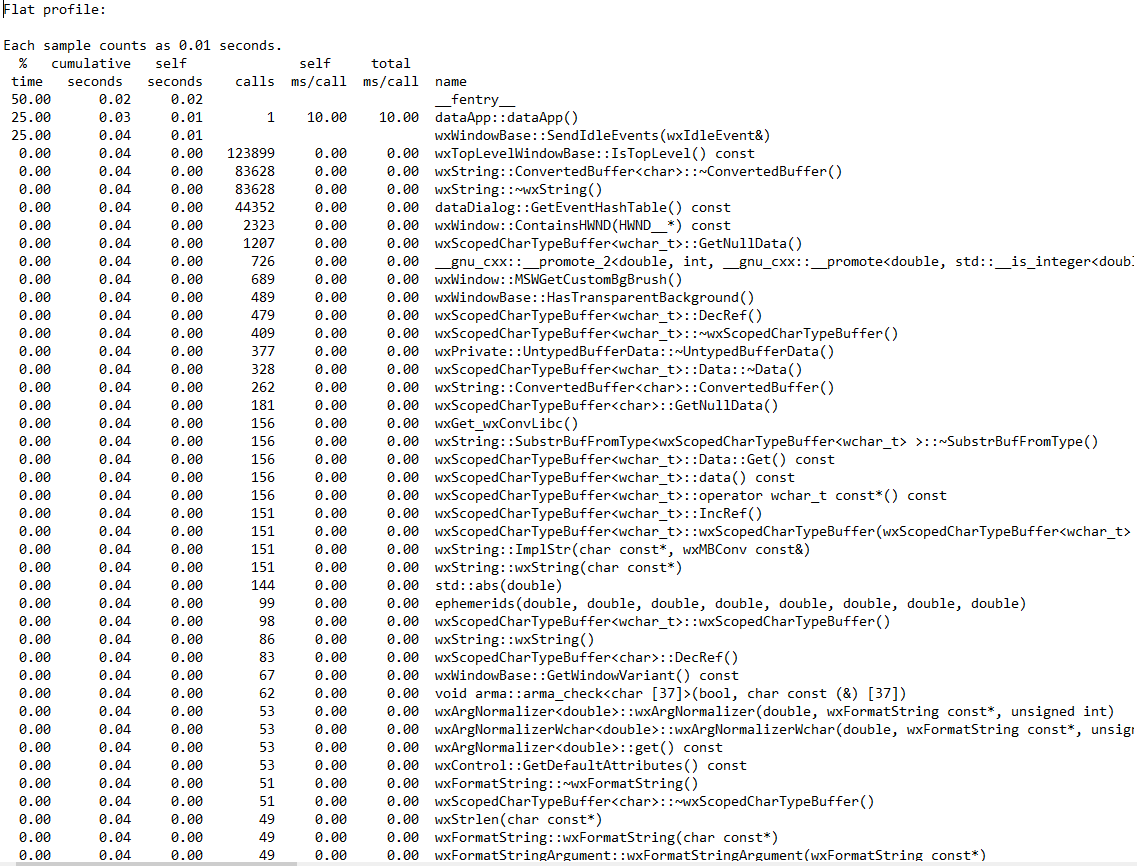
Файл “ephemerids.h”, код описан в приложение 11 - это заголовочный файл, в котором объявляются применяемые классы и методы для функции “ephemerids”, путь к файлу: “…\data \ ephemerids.h ”

2.5. Профилирование проекта

Произведём профилирование для того, чтобы изучить, где программа расходует свое время и какие функции вызывались, пока программа исполнялась. С полученными результатами сможем узнать какая часть кода выполняется медленнее.

Профилирование будем производить с помощью встроенный утилиты – Gprof.

Получим результаты



1. Результаты профилирования программы

Полные логи полученные от утилиты Gprof в приложение 2.

По полученным результатам можно увидеть, что основная трата времени уходит на графический интерфейс программы, следовательно, можно сделать вывод, что пока что дальнейшие доработки по ускорению кода функций не нужны.

**ГЛАВА 3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

*Минимальные требования:*

Для запуска программы необходимо иметь windows 7/10 и подключенное устройство к интернету.

*Инструкция:*

1. Запустите программу “Data.exe” от имени администратора

Если программа не запустится отключите антивирус.

У вас появится диалоговое окно:

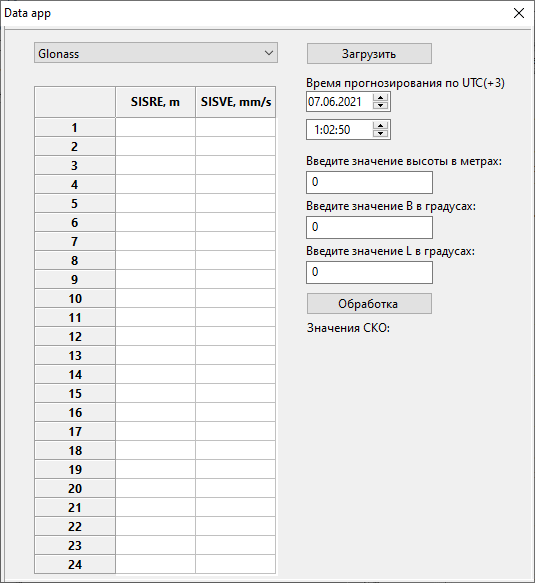


Рисунок 3.1. Интерфейс программы

1. Выберите необходимую вам НС:

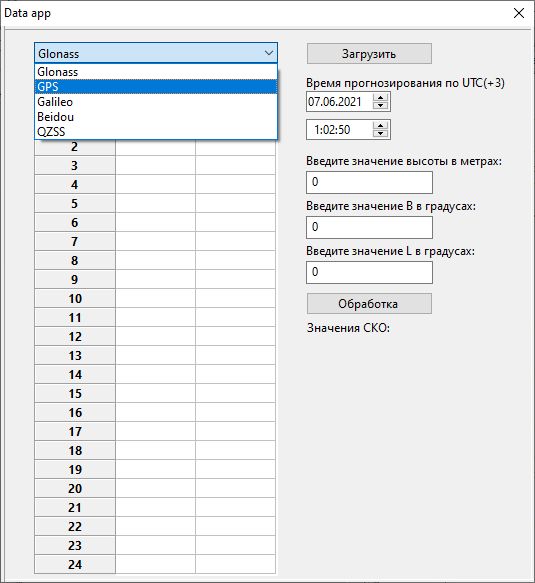


Рисунок 3.2. Интерфейс выбора НС

1. Нажмите кнопку «Загрузить»

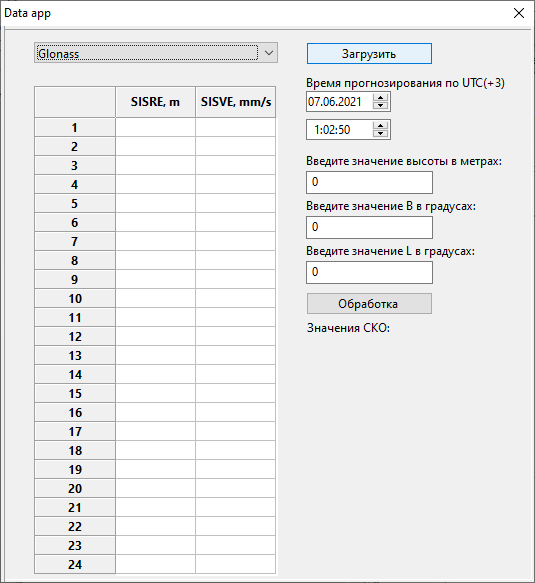


Рисунок 3.3. Интерфейс выбранной НС

Получили значения SISRE и SISVE для каждого спутника, если

значения равны 0.00, то данный спутник отсутствует.

Если при загрузке возникла ошибка существует два варианта решения ее:

* Отключите антивирус,
* Включите брандмауэр.

1. Для того, чтобы скачать данные для других НС, перейдите к п. 2 инструкции.

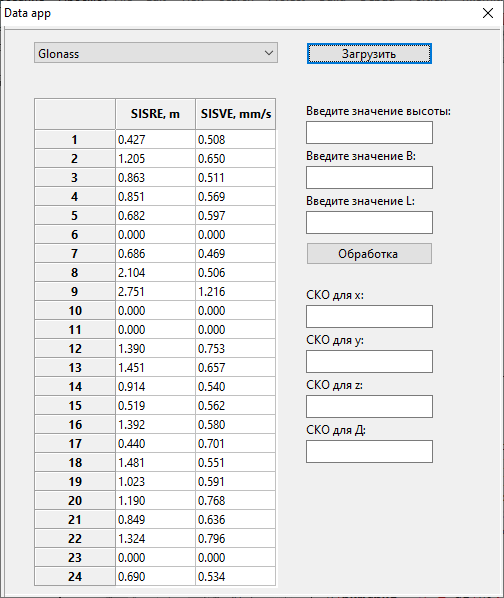


Рисунок 3.4. Полученные значения

1. Далее вводим значения:

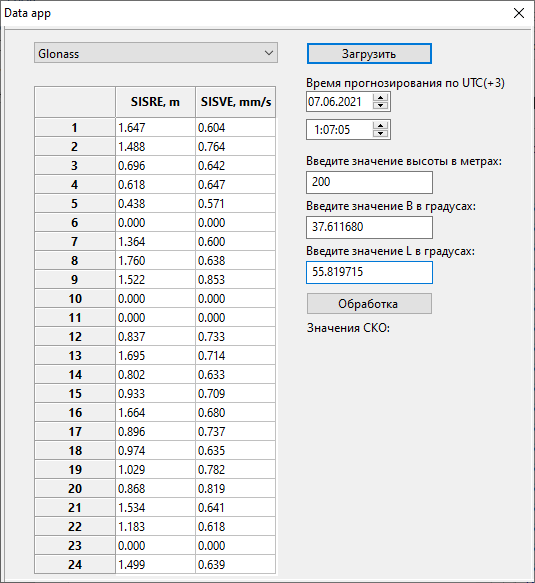


Рисунок 3.5. Ввод значений в формы

1. Нажимаем кнопку обработка:

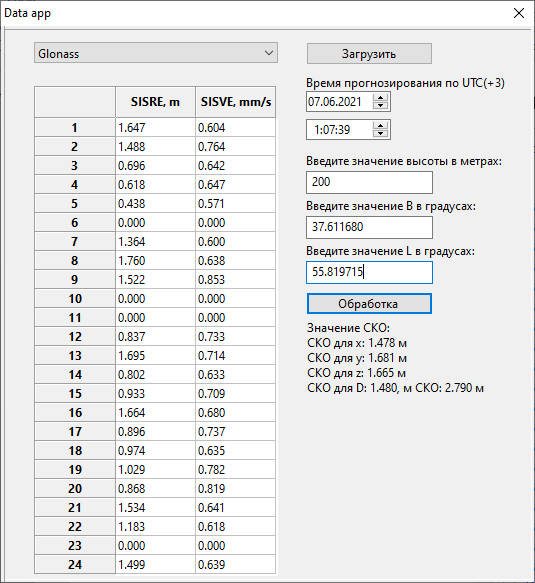


Рисунок 3.6. Полученные значения

Получаем значения СКО.

1. Чтобы закрыть программу нажмите крестик в диалоговом окне

**ЗАКЛЮЧЕНИЕ**

Получил значения СКО для x,y,z - координаты и погрешности временной шкалы, которые рассчитываются по данным альманахам соответствующих группировок для созвездий ГЛОНАСС и GPS.

Не все задачи удалось выполнить из-за затеявшегося изменения функции расчета координат НКА по данным альманахов, так как для изменения ее пришлось написать и переписать ряд функций: функции обработки файлов с альманахами, перерасчет времени, алгоритм скачивания файлов с сервера.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1]. Сервер «инфомационно-аналитечкского центра коррдинатно-временного и навигационного обеспечения «ftp://glonass-iac.ru» »

[2]. Отчет НИР за 9 семестр

**ПРИЛОЖЕНИЕ**

Приложение 1

Приложение 2

|  |
| --- |
| Flat profile:  Each sample counts as 0.01 seconds.  % cumulative self self total  time seconds seconds calls ms/call ms/call name  50.00 0.02 0.02 \_\_fentry\_\_  25.00 0.03 0.01 1 10.00 10.00 dataApp::dataApp()  25.00 0.04 0.01 wxWindowBase::SendIdleEvents(wxIdleEvent&)  0.00 0.04 0.00 123899 0.00 0.00 wxTopLevelWindowBase::IsTopLevel() const  0.00 0.04 0.00 83628 0.00 0.00 wxString::ConvertedBuffer<char>::~ConvertedBuffer()  0.00 0.04 0.00 83628 0.00 0.00 wxString::~wxString()  0.00 0.04 0.00 44352 0.00 0.00 dataDialog::GetEventHashTable() const  0.00 0.04 0.00 2323 0.00 0.00 wxWindow::ContainsHWND(HWND\_\_\*) const  0.00 0.04 0.00 1207 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::GetNullData()  0.00 0.04 0.00 726 0.00 0.00 \_\_gnu\_cxx::\_\_promote\_2<double, int, \_\_gnu\_cxx::\_\_promote<double, std::\_\_is\_integer<double>::\_\_value>::\_\_type, \_\_gnu\_cxx::\_\_promote<int, std::\_\_is\_integer<int>::\_\_value>::\_\_type>::\_\_type std::pow<double, int>(double, int)  0.00 0.04 0.00 689 0.00 0.00 wxWindow::MSWGetCustomBgBrush()  0.00 0.04 0.00 489 0.00 0.00 wxWindowBase::HasTransparentBackground()  0.00 0.04 0.00 479 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::DecRef()  0.00 0.04 0.00 409 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::~wxScopedCharTypeBuffer()  0.00 0.04 0.00 377 0.00 0.00 wxPrivate::UntypedBufferData::~UntypedBufferData()  0.00 0.04 0.00 328 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::Data::~Data()  0.00 0.04 0.00 262 0.00 0.00 wxString::ConvertedBuffer<char>::ConvertedBuffer()  0.00 0.04 0.00 181 0.00 0.00 wxScopedCharTypeBuffer<char>::GetNullData()  0.00 0.04 0.00 156 0.00 0.00 wxGet\_wxConvLibc()  0.00 0.04 0.00 156 0.00 0.00 wxString::SubstrBufFromType<wxScopedCharTypeBuffer<wchar\_t> >::~SubstrBufFromType()  0.00 0.04 0.00 156 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::Data::Get() const  0.00 0.04 0.00 156 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::data() const  0.00 0.04 0.00 156 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::operator wchar\_t const\*() const  0.00 0.04 0.00 151 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::IncRef()  0.00 0.04 0.00 151 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::wxScopedCharTypeBuffer(wxScopedCharTypeBuffer<wchar\_t> const&)  0.00 0.04 0.00 151 0.00 0.00 wxString::ImplStr(char const\*, wxMBConv const&)  0.00 0.04 0.00 151 0.00 0.00 wxString::wxString(char const\*)  0.00 0.04 0.00 144 0.00 0.00 std::abs(double)  0.00 0.04 0.00 99 0.00 0.00 ephemerids(double, double, double, double, double, double, double, double)  0.00 0.04 0.00 98 0.00 0.00 wxScopedCharTypeBuffer<wchar\_t>::wxScopedCharTypeBuffer()  0.00 0.04 0.00 86 0.00 0.00 wxString::wxString()  0.00 0.04 0.00 83 0.00 0.00 wxScopedCharTypeBuffer<char>::DecRef()  0.00 0.04 0.00 67 0.00 0.00 wxWindowBase::GetWindowVariant() const  0.00 0.04 0.00 62 0.00 0.00 void arma::arma\_check<char [37]>(bool, char const (&) [37])  0.00 0.04 0.00 53 0.00 0.00 wxArgNormalizer<double>::wxArgNormalizer(double, wxFormatString const\*, unsigned int)  0.00 0.04 0.00 53 0.00 0.00 wxArgNormalizerWchar<double>::wxArgNormalizerWchar(double, wxFormatString const\*, unsigned int)  0.00 0.04 0.00 53 0.00 0.00 wxArgNormalizer<double>::get() const  0.00 0.04 0.00 53 0.00 0.00 wxControl::GetDefaultAttributes() const  0.00 0.04 0.00 51 0.00 0.00 wxFormatString::~wxFormatString()  0.00 0.04 0.00 51 0.00 0.00 wxScopedCharTypeBuffer<char>::~wxScopedCharTypeBuffer()  0.00 0.04 0.00 49 0.00 0.00 wxStrlen(char const\*)  0.00 0.04 0.00 49 0.00 0.00 wxFormatString::wxFormatString(char const\*)  0.00 0.04 0.00 49 0.00 0.00 wxFormatStringArgument::wxFormatStringArgument(wxFormatString const\*)  0.00 0.04 0.00 49 0.00 0.00 wxScopedCharTypeBuffer<char>::CreateNonOwned(char const\*, unsigned long long)  0.00 0.04 0.00 49 0.00 0.00 wxScopedCharTypeBuffer<char>::Data::Data(char\*, unsigned long long, wxPrivate::UntypedBufferData::Kind)  0.00 0.04 0.00 49 0.00 0.00 wxScopedCharTypeBuffer<char>::Data::~Data()  0.00 0.04 0.00 49 0.00 0.00 wxScopedCharTypeBuffer<char>::wxScopedCharTypeBuffer()  0.00 0.04 0.00 49 0.00 0.00 wxFormatStringArgumentFinder<wxFormatString const&>::find(wxFormatString const&)  0.00 0.04 0.00 49 0.00 0.00 wxPrivate::UntypedBufferData::UntypedBufferData(void\*, unsigned long long, wxPrivate::UntypedBufferData::Kind)  0.00 0.04 0.00 49 0.00 0.00 wxFormatString::operator wchar\_t const\*() const  0.00 0.04 0.00 49 0.00 0.00 wxFormatStringArgument::operator wxFormatString const\*() const  0.00 0.04 0.00 48 0.00 0.00 wxString wxString::Format<double>(wxFormatString const&, double)  0.00 0.04 0.00 34 0.00 0.00 timeCalc::timeCalc(int, int, int, int, int, int, int)  0.00 0.04 0.00 34 0.00 0.00 timeCalc::~timeCalc()  0.00 0.04 0.00 33 0.00 0.00 ephemeridsGLNS(double, double, double, double, double, double, double, double, double, double, double)  0.00 0.04 0.00 30 0.00 0.00 wxObject::wxObject()  0.00 0.04 0.00 28 0.00 0.00 wxColourBase::wxColourBase()  0.00 0.04 0.00 28 0.00 0.00 wxColour::wxColour()  0.00 0.04 0.00 28 0.00 0.00 wxObjectEventFunctor::GetEvtHandler() const  0.00 0.04 0.00 25 0.00 0.00 std::vector<int, std::allocator<int> >::size() const  0.00 0.04 0.00 24 0.00 0.00 angle(double\*, double\*, double, double)  0.00 0.04 0.00 24 0.00 0.00 xyz2enu(double, double, double\*)  0.00 0.04 0.00 24 0.00 0.00 timeCalc::timeGLNS()  0.00 0.04 0.00 24 0.00 0.00 \_\_gnu\_cxx::\_\_promote\_2<double, int, \_\_gnu\_cxx::\_\_promote<double, std::\_\_is\_integer<double>::\_\_value>::\_\_type, \_\_gnu\_cxx::\_\_promote<int, std::\_\_is\_integer<int>::\_\_value>::\_\_type>::\_\_type std::fmod<double, int>(double, int)  0.00 0.04 0.00 23 0.00 0.00 int const& std::forward<int const&>(std::remove\_reference<int const&>::type&)  0.00 0.04 0.00 21 0.00 0.00 wxGetTranslation(wxString const&, wxString const&, wxString const&)  0.00 0.04 0.00 20 0.00 0.00 wxString::ImplStr(wchar\_t const\*)  0.00 0.04 0.00 20 0.00 0.00 wxString::wxString(wchar\_t const\*)  0.00 0.04 0.00 19 0.00 0.00 wxPoint::wxPoint()  0.00 0.04 0.00 18 0.00 0.00 wxPoint::wxPoint(int, int)  0.00 0.04 0.00 17 0.00 0.00 wxSize::wxSize()  0.00 0.04 0.00 17 0.00 0.00 wxWindow::wxWindow()  0.00 0.04 0.00 15 0.00 0.00 \_\_gnu\_cxx::new\_allocator<int>::max\_size() const  0.00 0.04 0.00 15 0.00 0.00 int\* std::\_\_niter\_base<int\*>(int\*)  0.00 0.04 0.00 14 0.00 0.00 wxArrayLong::wxArrayLong()  0.00 0.04 0.00 14 0.00 0.00 wxBaseArray<long, wxSortedArray\_SortFunction<long> >::wxBaseArray()  0.00 0.04 0.00 14 0.00 0.00 wxEvtHandler::SetNextHandler(wxEvtHandler\*)  0.00 0.04 0.00 14 0.00 0.00 wxControlBase::wxControlBase()  0.00 0.04 0.00 14 0.00 0.00 wxSize::wxSize(int, int)  0.00 0.04 0.00 14 0.00 0.00 wxVector<long>::wxVector()  0.00 0.04 0.00 14 0.00 0.00 wxControl::wxControl()  0.00 0.04 0.00 14 0.00 0.00 wxWindowBase::GetDefaultAttributes() const  0.00 0.04 0.00 13 0.00 0.00 wxArrayInt::wxArrayInt()  0.00 0.04 0.00 13 0.00 0.00 wxBaseArray<int, wxSortedArray\_SortFunction<int> >::wxBaseArray()  0.00 0.04 0.00 13 0.00 0.00 wxVector<int>::wxVector()  0.00 0.04 0.00 12 0.00 0.00 wxWindowBase::DoGetBestClientSize() const  0.00 0.04 0.00 11 0.00 0.00 wxWindowBase::ShouldInheritColours() const  0.00 0.04 0.00 11 0.00 0.00 std::\_Vector\_base<int, std::allocator<int> >::\_M\_get\_Tp\_allocator()  0.00 0.04 0.00 10 0.00 0.00 wxEvtHandler::OnDynamicBind(wxDynamicEventTableEntry&)  0.00 0.04 0.00 10 0.00 0.00 wxGridCellCoords::wxGridCellCoords()  0.00 0.04 0.00 10 0.00 0.00 wxString::length() const  0.00 0.04 0.00 10 0.00 0.00 std::\_Vector\_base<int, std::allocator<int> >::\_M\_get\_Tp\_allocator() const  0.00 0.04 0.00 10 0.00 0.00 std::move\_iterator<int\*>::base() const  0.00 0.04 0.00 10 0.00 0.00 std::vector<int, std::allocator<int> >::max\_size() const  0.00 0.04 0.00 10 0.00 0.00 std::move\_iterator<int\*>::move\_iterator(int\*)  0.00 0.04 0.00 10 0.00 0.00 std::allocator\_traits<std::allocator<int> >::max\_size(std::allocator<int> const&)  0.00 0.04 0.00 10 0.00 0.00 decltype (\_\_miter\_base(({parm#1}.base)())) std::\_\_miter\_base<int\*>(std::move\_iterator<int\*>)  0.00 0.04 0.00 10 0.00 0.00 int\* std::\_\_miter\_base<int\*>(int\*)  0.00 0.04 0.00 10 0.00 0.00 std::move\_iterator<int\*> std::\_\_make\_move\_if\_noexcept\_iterator<int, std::move\_iterator<int\*> >(int\*)  0.00 0.04 0.00 9 0.00 0.00 arma::Base\_eval\_Mat<double, arma::Mat<double> >::Base\_eval\_Mat()  0.00 0.04 0.00 9 0.00 0.00 arma::Base\_extra\_yes<double, arma::Mat<double> >::Base\_extra\_yes()  0.00 0.04 0.00 9 0.00 0.00 arma::Base\_trans\_default<arma::Mat<double> >::Base\_trans\_default()  0.00 0.04 0.00 9 0.00 0.00 arma::Mat<double>::~Mat()  0.00 0.04 0.00 9 0.00 0.00 arma::Base<double, arma::Mat<double> >::Base()  0.00 0.04 0.00 9 0.00 0.00 void \_\_gnu\_cxx::new\_allocator<int>::construct<int, int const&>(int\*, int const&)  0.00 0.04 0.00 9 0.00 0.00 wxControlBase::ShouldInheritColours() const  0.00 0.04 0.00 9 0.00 0.00 void std::allocator\_traits<std::allocator<int> >::construct<int, int const&>(std::allocator<int>&, int\*, int const&)  0.00 0.04 0.00 9 0.00 0.00 std::vector<int, std::allocator<int> >::push\_back(int const&)  0.00 0.04 0.00 9 0.00 0.00 operator new(unsigned long long, void\*)  0.00 0.04 0.00 8 0.00 0.00 std::literals::string\_literals::operator"" s[abi:cxx11](char const\*, unsigned long long)  0.00 0.04 0.00 7 0.00 0.00 wxNewEventFunctor(int const&, void (wxEvtHandler::\*)(wxEvent&), wxEvtHandler\*)  0.00 0.04 0.00 7 0.00 0.00 wxColourBase::~wxColourBase()  0.00 0.04 0.00 7 0.00 0.00 wxEvtHandler::Connect(int, int, void (wxEvtHandler::\*)(wxEvent&), wxObject\*, wxEvtHandler\*)  0.00 0.04 0.00 7 0.00 0.00 wxEvtHandler::Connect(int, int, int, void (wxEvtHandler::\*)(wxEvent&), wxObject\*, wxEvtHandler\*)  0.00 0.04 0.00 7 0.00 0.00 wxEventFunctor::wxEventFunctor()  0.00 0.04 0.00 7 0.00 0.00 wxObjectEventFunctor::wxObjectEventFunctor(void (wxEvtHandler::\*)(wxEvent&), wxEvtHandler\*)  0.00 0.04 0.00 7 0.00 0.00 wxObjectEventFunctor::~wxObjectEventFunctor()  0.00 0.04 0.00 7 0.00 0.00 wxObjectEventFunctor::~wxObjectEventFunctor()  0.00 0.04 0.00 7 0.00 0.00 wxColour::~wxColour()  0.00 0.04 0.00 7 0.00 0.00 wxObject::~wxObject()  0.00 0.04 0.00 7 0.00 0.00 wxAnyButtonBase::GetDefaultBorder() const  0.00 0.04 0.00 7 0.00 0.00 std::remove\_reference<std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >&>::type&& std::move<std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >&>(std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >&)  0.00 0.04 0.00 6 0.00 0.00 void arma::arma\_check<char [74]>(bool, char const (&) [74])  0.00 0.04 0.00 6 0.00 0.00 void arma::arma\_check<char\*>(bool, char\* const&)  0.00 0.04 0.00 6 0.00 0.00 arma::Mat<double>::init\_warm(unsigned long long, unsigned long long)  0.00 0.04 0.00 6 0.00 0.00 wxDateTimePickerCtrl::GetDefaultBorder() const  0.00 0.04 0.00 6 0.00 0.00 wxObjectEventFunctor::GetEvtMethod() const  0.00 0.04 0.00 6 0.00 0.00 wxWindow::MSWFindItem(long, HWND\_\_\*) const  0.00 0.04 0.00 6 0.00 0.00 void std::\_Destroy\_aux<true>::\_\_destroy<int\*>(int\*, int\*)  0.00 0.04 0.00 6 0.00 0.00 std::\_Vector\_base<int, std::allocator<int> >::\_M\_deallocate(int\*, unsigned long long)  0.00 0.04 0.00 6 0.00 0.00 void std::\_Destroy<int\*>(int\*, int\*)  0.00 0.04 0.00 6 0.00 0.00 void std::\_Destroy<int\*, int>(int\*, int\*, std::allocator<int>&)  0.00 0.04 0.00 5 0.00 0.00 wxStaticText::wxStaticText(wxWindow\*, int, wxString const&, wxPoint const&, wxSize const&, long, wxString const&)  0.00 0.04 0.00 5 0.00 0.00 wxItemContainer::AppendItems(wxArrayStringsAdapter const&)  0.00 0.04 0.00 5 0.00 0.00 wxItemContainer::AppendItems(wxArrayStringsAdapter const&, void\*\*, wxClientDataType)  0.00 0.04 0.00 5 0.00 0.00 wxItemContainer::DoAppendItems(wxArrayStringsAdapter const&, void\*\*, wxClientDataType)  0.00 0.04 0.00 5 0.00 0.00 wxItemContainer::Append(wxString const&)  0.00 0.04 0.00 5 0.00 0.00 wxStaticTextBase::wxStaticTextBase()  0.00 0.04 0.00 5 0.00 0.00 wxArrayStringsAdapter::wxArrayStringsAdapter(wxString const&)  0.00 0.04 0.00 5 0.00 0.00 void arma::arma\_check<char [53]>(bool, char const (&) [53])  0.00 0.04 0.00 5 0.00 0.00 void arma::arma\_check\_bad\_alloc<char [39]>(bool, char const (&) [39])  0.00 0.04 0.00 5 0.00 0.00 double\* arma::memory::acquire<double>(unsigned long long)  0.00 0.04 0.00 5 0.00 0.00 dataApp::OnInit()  0.00 0.04 0.00 5 0.00 0.00 wxString::assign(char const\*, unsigned long long)  0.00 0.04 0.00 5 0.00 0.00 wxString::ImplStr(char const\*, unsigned long long, wxMBConv const&)  0.00 0.04 0.00 5 0.00 0.00 wxString::wxString(std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > const&)  0.00 0.04 0.00 5 0.00 0.00 \_\_gnu\_cxx::new\_allocator<int>::deallocate(int\*, unsigned long long)  0.00 0.04 0.00 5 0.00 0.00 \_\_gnu\_cxx::new\_allocator<int>::allocate(unsigned long long, void const\*)  0.00 0.04 0.00 5 0.00 0.00 bool \_\_gnu\_cxx::\_\_is\_null\_pointer<char>(char\*)  0.00 0.04 0.00 5 0.00 0.00 wxArrayStringsAdapter::IsEmpty() const  0.00 0.04 0.00 5 0.00 0.00 wxArrayStringsAdapter::GetCount() const  0.00 0.04 0.00 5 0.00 0.00 wxString::IsSameAs(wxString const&, bool) const  0.00 0.04 0.00 5 0.00 0.00 std::vector<int, std::allocator<int> >::\_M\_check\_len(unsigned long long, char const\*) const  0.00 0.04 0.00 5 0.00 0.00 int\* std::\_\_copy\_move<true, true, std::random\_access\_iterator\_tag>::\_\_copy\_m<int>(int const\*, int const\*, int\*)  0.00 0.04 0.00 5 0.00 0.00 std::\_Vector\_base<int, std::allocator<int> >::\_M\_allocate(unsigned long long)  0.00 0.04 0.00 5 0.00 0.00 std::numeric\_limits<unsigned long long>::max()  0.00 0.04 0.00 5 0.00 0.00 std::allocator\_traits<std::allocator<int> >::deallocate(std::allocator<int>&, int\*, unsigned long long)  0.00 0.04 0.00 5 0.00 0.00 std::allocator\_traits<std::allocator<int> >::allocate(std::allocator<int>&, unsigned long long)  0.00 0.04 0.00 5 0.00 0.00 int\* std::\_\_uninitialized\_copy<true>::\_\_uninit\_copy<std::move\_iterator<int\*>, int\*>(std::move\_iterator<int\*>, std::move\_iterator<int\*>, int\*)  0.00 0.04 0.00 5 0.00 0.00 void std::vector<int, std::allocator<int> >::\_M\_emplace\_back\_aux<int const&>(int const&)  0.00 0.04 0.00 5 0.00 0.00 void std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >::\_M\_construct<char\*>(char\*, char\*, std::forward\_iterator\_tag)  0.00 0.04 0.00 5 0.00 0.00 std::iterator\_traits<char\*>::difference\_type std::\_\_distance<char\*>(char\*, char\*, std::random\_access\_iterator\_tag)  0.00 0.04 0.00 5 0.00 0.00 int\* std::\_\_copy\_move\_a<true, int\*, int\*>(int\*, int\*, int\*)  0.00 0.04 0.00 5 0.00 0.00 int\* std::\_\_copy\_move\_a2<true, int\*, int\*>(int\*, int\*, int\*)  0.00 0.04 0.00 5 0.00 0.00 int\* std::uninitialized\_copy<std::move\_iterator<int\*>, int\*>(std::move\_iterator<int\*>, std::move\_iterator<int\*>, int\*)  0.00 0.04 0.00 5 0.00 0.00 std::iterator\_traits<char\*>::iterator\_category std::\_\_iterator\_category<char\*>(char\* const&)  0.00 0.04 0.00 5 0.00 0.00 int\* std::\_\_uninitialized\_copy\_a<std::move\_iterator<int\*>, int\*, int>(std::move\_iterator<int\*>, std::move\_iterator<int\*>, int\*, std::allocator<int>&)  0.00 0.04 0.00 5 0.00 0.00 int\* std::\_\_uninitialized\_move\_if\_noexcept\_a<int\*, int\*, std::allocator<int> >(int\*, int\*, int\*, std::allocator<int>&)  0.00 0.04 0.00 5 0.00 0.00 unsigned long long const& std::max<unsigned long long>(unsigned long long const&, unsigned long long const&)  0.00 0.04 0.00 5 0.00 0.00 int\* std::copy<std::move\_iterator<int\*>, int\*>(std::move\_iterator<int\*>, std::move\_iterator<int\*>, int\*)  0.00 0.04 0.00 5 0.00 0.00 std::iterator\_traits<char\*>::difference\_type std::distance<char\*>(char\*, char\*)  0.00 0.04 0.00 5 0.00 0.00 operator==(wxString const&, wxString const&)  0.00 0.04 0.00 4 0.00 0.00 wxSashEdge::wxSashEdge()  0.00 0.04 0.00 4 0.00 0.00 wxIntegerHash::wxIntegerHash()  0.00 0.04 0.00 4 0.00 0.00 wxIntegerEqual::wxIntegerEqual()  0.00 0.04 0.00 4 0.00 0.00 \_wxHashTableBase2::AllocTable(unsigned long long)  0.00 0.04 0.00 4 0.00 0.00 \_wxHashTableBase2::\_wxHashTableBase2()  0.00 0.04 0.00 4 0.00 0.00 wxLongToLongHashMap::wxLongToLongHashMap(unsigned long long, wxIntegerHash, wxIntegerEqual)  0.00 0.04 0.00 4 0.00 0.00 wxLongToLongHashMap\_wxImplementation\_KeyEx::wxLongToLongHashMap\_wxImplementation\_KeyEx()  0.00 0.04 0.00 4 0.00 0.00 wxLongToLongHashMap\_wxImplementation\_HashTable::wxLongToLongHashMap\_wxImplementation\_HashTable(unsigned long long, wxIntegerHash const&, wxIntegerEqual const&, wxLongToLongHashMap\_wxImplementation\_KeyEx const&)  0.00 0.04 0.00 4 0.00 0.00 arma::Mat<double>::set\_size(unsigned long long, unsigned long long)  0.00 0.04 0.00 4 0.00 0.00 arma::Mat<double>::Mat()  0.00 0.04 0.00 4 0.00 0.00 std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > \_\_gnu\_cxx::\_\_to\_xstring<std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >, char>(int (\*)(char\*, unsigned long long, char const\*, char\*), unsigned long long, char const\*, ...)  0.00 0.04 0.00 4 0.00 0.00 wxEventTypeTag<wxCommandEvent>::operator int const&() const  0.00 0.04 0.00 4 0.00 0.00 void std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >::\_M\_construct<char\*>(char\*, char\*)  0.00 0.04 0.00 4 0.00 0.00 void std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >::\_M\_construct\_aux<char\*>(char\*, char\*, std::\_\_false\_type)  0.00 0.04 0.00 4 0.00 0.00 std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >::basic\_string<char\*, void>(char\*, char\*, std::allocator<char> const&)  0.00 0.04 0.00 4 0.00 0.00 std::\_\_cxx11::to\_string(int)  0.00 0.04 0.00 3 0.00 0.00 dataDialog::OnTextCtrl1Text1(wxCommandEvent&)  0.00 0.04 0.00 3 0.00 0.00 wxDateTime::wxDateTime()  0.00 0.04 0.00 3 0.00 0.00 wxTextAttr::wxTextAttr()  0.00 0.04 0.00 3 0.00 0.00 wxTextCtrl::wxTextCtrl(wxWindow\*, int, wxString const&, wxPoint const&, wxSize const&, long, wxValidator const&, wxString const&)  0.00 0.04 0.00 3 0.00 0.00 wxAnyButton::wxAnyButton()  0.00 0.04 0.00 3 0.00 0.00 wxButtonBase::wxButtonBase()  0.00 0.04 0.00 3 0.00 0.00 wxTextAreaBase::wxTextAreaBase()  0.00 0.04 0.00 3 0.00 0.00 wxTextCtrlBase::wxTextCtrlBase()  0.00 0.04 0.00 3 0.00 0.00 wxAnyButtonBase::wxAnyButtonBase()  0.00 0.04 0.00 3 0.00 0.00 wxLongLongNative::wxLongLongNative(long long)  0.00 0.04 0.00 3 0.00 0.00 wxButton::Init()  0.00 0.04 0.00 3 0.00 0.00 wxButton::wxButton(wxWindow\*, int, wxString const&, wxPoint const&, wxSize const&, long, wxValidator const&, wxString const&)  0.00 0.04 0.00 3 0.00 0.00 wxAnyButtonBase::ShouldInheritColours() const  0.00 0.04 0.00 3 0.00 0.00 std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > std::operator+<char, std::char\_traits<char>, std::allocator<char> >(std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >&&, std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > const&)  0.00 0.04 0.00 3 0.00 0.00 std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > std::operator+<char, std::char\_traits<char>, std::allocator<char> >(std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >&&, std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >&&)  0.00 0.04 0.00 2 0.00 0.00 download(char const\*, char const\*, char const\*, char const\*, char const\*)  0.00 0.04 0.00 2 0.00 0.00 wxFontBase::wxFontBase()  0.00 0.04 0.00 2 0.00 0.00 wxScrolled<wxWindow>::wxScrolled()  0.00 0.04 0.00 2 0.00 0.00 wxGDIObject::wxGDIObject()  0.00 0.04 0.00 2 0.00 0.00 wxScrolledT\_Helper::wxScrolledT\_Helper()  0.00 0.04 0.00 2 0.00 0.00 wxDateTimePickerCtrl::wxDateTimePickerCtrl()  0.00 0.04 0.00 2 0.00 0.00 wxDateTimePickerCtrlBase::wxDateTimePickerCtrlBase()  0.00 0.04 0.00 2 0.00 0.00 void arma::arma\_check<char [41]>(bool, char const (&) [41])  0.00 0.04 0.00 2 0.00 0.00 void arma::arma\_check<char const\*>(bool, char const\* const&)  0.00 0.04 0.00 2 0.00 0.00 arma::Mat<double>::zeros()  0.00 0.04 0.00 2 0.00 0.00 arma::Mat<double>::init\_cold()  0.00 0.04 0.00 2 0.00 0.00 arma::unwrap<arma::Mat<double> >::unwrap(arma::Mat<double> const&)  0.00 0.04 0.00 2 0.00 0.00 void arma::arrayops::fill\_zeros<double>(double\*, unsigned long long)  0.00 0.04 0.00 2 0.00 0.00 void arma::arrayops::inplace\_set<double>(double\*, double, unsigned long long)  0.00 0.04 0.00 2 0.00 0.00 wxFont::wxFont()  0.00 0.04 0.00 2 0.00 0.00 wxGrid::wxGrid(wxWindow\*, int, wxPoint const&, wxSize const&, long, wxString const&)  0.00 0.04 0.00 2 0.00 0.00 wxString::Cmp(char const\*) const  0.00 0.04 0.00 2 0.00 0.00 wxString::Cmp(wxString const&) const  0.00 0.04 0.00 2 0.00 0.00 operator==(wxString const&, char const\*)  0.00 0.04 0.00 1 0.00 10.00 wxCreateApp()  0.00 0.04 0.00 1 0.00 0.00 dataDialog::OnButton2Click(wxCommandEvent&)  0.00 0.04 0.00 1 0.00 0.00 dataDialog::OnButton1Click1(wxCommandEvent&)  0.00 0.04 0.00 1 0.00 0.00 dataDialog::Gridd(char const\*)  0.00 0.04 0.00 1 0.00 0.00 dataDialog::OnInit(wxInitDialogEvent&)  0.00 0.04 0.00 1 0.00 0.00 dataDialog::dataDialog(wxWindow\*, int)  0.00 0.04 0.00 1 0.00 0.00 dataDialog::~dataDialog()  0.00 0.04 0.00 1 0.00 0.00 wxDateTime::Now()  0.00 0.04 0.00 1 0.00 0.00 wxDateTime::wxDateTime(tm const&)  0.00 0.04 0.00 1 0.00 0.00 wxTextFile::~wxTextFile()  0.00 0.04 0.00 1 0.00 0.00 wxChoiceBase::wxChoiceBase()  0.00 0.04 0.00 1 0.00 0.00 wxSashWindow::wxSashWindow(wxWindow\*, int, wxPoint const&, wxSize const&, long, wxString const&)  0.00 0.04 0.00 1 0.00 0.00 wxWindowBase::SetClientSize(wxSize const&)  0.00 0.04 0.00 1 0.00 0.00 wxItemContainer::wxItemContainer()  0.00 0.04 0.00 1 0.00 0.00 wxDatePickerCtrl::wxDatePickerCtrl(wxWindow\*, int, wxDateTime const&, wxPoint const&, wxSize const&, long, wxValidator const&, wxString const&)  0.00 0.04 0.00 1 0.00 0.00 wxLongLongNative::wxLongLongNative()  0.00 0.04 0.00 1 0.00 0.00 wxTimePickerCtrl::Create(wxWindow\*, int, wxDateTime const&, wxPoint const&, wxSize const&, long, wxValidator const&, wxString const&)  0.00 0.04 0.00 1 0.00 0.00 wxTimePickerCtrl::wxTimePickerCtrl(wxWindow\*, int, wxDateTime const&, wxPoint const&, wxSize const&, long, wxValidator const&, wxString const&)  0.00 0.04 0.00 1 0.00 0.00 wxWindowWithItems<wxControl, wxItemContainer>::wxWindowWithItems()  0.00 0.04 0.00 1 0.00 0.00 wxControlWithItems::wxControlWithItems()  0.00 0.04 0.00 1 0.00 0.00 wxNavigationEnabled<wxTopLevelWindow>::AddChild(wxWindowBase\*)  0.00 0.04 0.00 1 0.00 0.00 wxDatePickerCtrlBase::wxDatePickerCtrlBase()  0.00 0.04 0.00 1 0.00 0.00 wxTimePickerCtrlBase::wxTimePickerCtrlBase()  0.00 0.04 0.00 1 0.00 0.00 wxControlWithItemsBase::wxControlWithItemsBase()  0.00 0.04 0.00 1 0.00 0.00 wxItemContainerImmutable::wxItemContainerImmutable()  0.00 0.04 0.00 1 0.00 0.00 void arma::arma\_check<char [63]>(bool, char const (&) [63])  0.00 0.04 0.00 1 0.00 0.00 void arma::glue\_times::apply<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double> >(arma::Mat<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>::elem\_type>&, arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> const&)  0.00 0.04 0.00 1 0.00 0.00 void arma::glue\_times::apply<double, false, false, false, arma::Mat<double>, arma::Mat<double> >(arma::Mat<double>&, arma::Mat<double> const&, arma::Mat<double> const&, double)  0.00 0.04 0.00 1 0.00 0.00 arma::xtrans\_mat<double, true>::xtrans\_mat(arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::xtrans\_mat<double, true>::~xtrans\_mat()  0.00 0.04 0.00 1 0.00 0.00 arma::Glue\_traits<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times, true>::Glue\_traits()  0.00 0.04 0.00 1 0.00 0.00 arma::Glue\_traits<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times, true>::Glue\_traits()  0.00 0.04 0.00 1 0.00 0.00 arma::quasi\_unwrap<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::quasi\_unwrap(arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::quasi\_unwrap<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::~quasi\_unwrap()  0.00 0.04 0.00 1 0.00 0.00 arma::strip\_diagmat<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::strip\_diagmat(arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_eval\_expr<double, arma::xtrans\_mat<double, true> >::Base\_eval\_expr()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_eval\_expr<double, arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >::Base\_eval\_expr()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_eval\_expr<double, arma::Op<arma::Mat<double>, arma::op\_inv> >::Base\_eval\_expr()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_eval\_expr<double, arma::Op<arma::Mat<double>, arma::op\_htrans> >::Base\_eval\_expr()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_eval\_expr<double, arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> >::Base\_eval\_expr()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_eval\_expr<double, arma::eOp<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt> >::Base\_eval\_expr()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_eval\_expr<double, arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::Base\_eval\_expr()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_eval\_expr<double, arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times> >::Base\_eval\_expr()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_extra\_yes<double, arma::xtrans\_mat<double, true> >::Base\_extra\_yes()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_extra\_yes<double, arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >::Base\_extra\_yes()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_extra\_yes<double, arma::Op<arma::Mat<double>, arma::op\_inv> >::Base\_extra\_yes()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_extra\_yes<double, arma::Op<arma::Mat<double>, arma::op\_htrans> >::Base\_extra\_yes()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_extra\_yes<double, arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> >::Base\_extra\_yes()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_extra\_yes<double, arma::eOp<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt> >::Base\_extra\_yes()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_extra\_yes<double, arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::Base\_extra\_yes()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_extra\_yes<double, arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times> >::Base\_extra\_yes()  0.00 0.04 0.00 1 0.00 0.00 arma::unwrap\_default<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> >::unwrap\_default(arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::unwrap\_default<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> >::~unwrap\_default()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_trans\_default<arma::xtrans\_mat<double, true> >::Base\_trans\_default()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_trans\_default<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >::Base\_trans\_default()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_trans\_default<arma::Op<arma::Mat<double>, arma::op\_inv> >::Base\_trans\_default()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_trans\_default<arma::Op<arma::Mat<double>, arma::op\_htrans> >::Base\_trans\_default()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_trans\_default<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> >::Base\_trans\_default()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_trans\_default<arma::eOp<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt> >::Base\_trans\_default()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_trans\_default<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::Base\_trans\_default()  0.00 0.04 0.00 1 0.00 0.00 arma::Base\_trans\_default<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times> >::Base\_trans\_default()  0.00 0.04 0.00 1 0.00 0.00 void arma::glue\_times\_redirect<3ull>::apply<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::Mat<double> >(arma::Mat<arma::Mat<double>::elem\_type>&, arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Proxy\_xtrans\_default<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >::Proxy\_xtrans\_default(arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Proxy\_xtrans\_default<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >::~Proxy\_xtrans\_default()  0.00 0.04 0.00 1 0.00 0.00 void arma::arma\_assert\_mul\_size<double, double>(arma::Mat<double> const&, arma::Mat<double> const&, char const\*)  0.00 0.04 0.00 1 0.00 0.00 arma::partial\_unwrap\_check<arma::Mat<double> >::partial\_unwrap\_check(arma::Mat<double> const&, arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::partial\_unwrap\_check<arma::Mat<double> >::~partial\_unwrap\_check()  0.00 0.04 0.00 1 0.00 0.00 arma::quasi\_unwrap\_default<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::quasi\_unwrap\_default(arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::quasi\_unwrap\_default<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::~quasi\_unwrap\_default()  0.00 0.04 0.00 1 0.00 0.00 void arma::arma\_assert\_blas\_size<arma::Mat<double> >(arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 void arma::arma\_assert\_blas\_size<arma::Mat<double>, arma::Mat<double> >(arma::Mat<double> const&, arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 void arma::arma\_assert\_trans\_mul\_size<false, false>(unsigned long long, unsigned long long, unsigned long long, unsigned long long, char const\*)  0.00 0.04 0.00 1 0.00 0.00 void arma::glue\_times\_redirect3\_helper<true>::apply<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::Mat<double> >(arma::Mat<arma::Mat<double>::elem\_type>&, arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>::Op(arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>::~Op()  0.00 0.04 0.00 1 0.00 0.00 arma::Op<arma::Mat<double>, arma::op\_inv>::Op(arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Op<arma::Mat<double>, arma::op\_inv>::~Op()  0.00 0.04 0.00 1 0.00 0.00 arma::Op<arma::Mat<double>, arma::op\_htrans>::Op(arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Op<arma::Mat<double>, arma::op\_htrans>::~Op()  0.00 0.04 0.00 1 0.00 0.00 arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>::Op(arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>::~Op()  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>::zeros(unsigned long long, unsigned long long)  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>::Mat(arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>::Mat(unsigned long long, unsigned long long)  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>::Mat<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>(arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>::Mat<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>(arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>::Mat<arma::Mat<double>, arma::op\_htrans>(arma::Op<arma::Mat<double>, arma::op\_htrans> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>::operator=(arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>& arma::Mat<double>::operator=<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt>(arma::eOp<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::eOp<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt>::eOp(arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::eOp<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt>::~eOp()  0.00 0.04 0.00 1 0.00 0.00 arma::Base<double, arma::xtrans\_mat<double, true> >::Base()  0.00 0.04 0.00 1 0.00 0.00 arma::Base<double, arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >::Base()  0.00 0.04 0.00 1 0.00 0.00 arma::Base<double, arma::Op<arma::Mat<double>, arma::op\_inv> >::Base()  0.00 0.04 0.00 1 0.00 0.00 arma::Base<double, arma::Op<arma::Mat<double>, arma::op\_htrans> >::Base()  0.00 0.04 0.00 1 0.00 0.00 arma::Base<double, arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> >::Base()  0.00 0.04 0.00 1 0.00 0.00 arma::Base<double, arma::eOp<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt> >::Base()  0.00 0.04 0.00 1 0.00 0.00 arma::Base<double, arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >::Base()  0.00 0.04 0.00 1 0.00 0.00 arma::Base<double, arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times> >::Base()  0.00 0.04 0.00 1 0.00 0.00 void arma::blas::gemm<double>(char const\*, char const\*, int const\*, int const\*, int const\*, double const\*, double const\*, int const\*, double const\*, int const\*, double const\*, double\*, int const\*)  0.00 0.04 0.00 1 0.00 0.00 void arma::gemm<false, false, false, false>::apply\_blas\_type<double, arma::Mat<double>, arma::Mat<double> >(arma::Mat<double>&, arma::Mat<double> const&, arma::Mat<double> const&, double, double)  0.00 0.04 0.00 1 0.00 0.00 arma::Proxy<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >::Proxy(arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::Proxy<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >::~Proxy()  0.00 0.04 0.00 1 0.00 0.00 arma::Proxy<arma::Mat<double> >::Proxy(arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 double arma::auxlib::det\_tinymat<double>(arma::Mat<double> const&, unsigned long long)  0.00 0.04 0.00 1 0.00 0.00 bool arma::auxlib::solve\_square\_fast<arma::Mat<double> >(arma::Mat<arma::Mat<double>::elem\_type>&, arma::Mat<arma::Mat<double>::elem\_type>&, arma::Base<arma::Mat<double>::elem\_type, arma::Mat<double> > const&)  0.00 0.04 0.00 1 0.00 0.00 bool arma::auxlib::inv\_tiny<double>(arma::Mat<double>&, arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 void arma::lapack::gesv<double>(int\*, int\*, double\*, int\*, int\*, double\*, int\*, int\*)  0.00 0.04 0.00 1 0.00 0.00 void arma::op\_inv::apply\_noalias<double>(arma::Mat<double>&, arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 void arma::op\_inv::apply<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times> >(arma::Mat<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>::elem\_type>&, arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> const&)  0.00 0.04 0.00 1 0.00 0.00 double arma::op\_max::direct\_max<double>(double const\*, unsigned long long)  0.00 0.04 0.00 1 0.00 0.00 arma::unwrap<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> >::unwrap(arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::unwrap<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv> >::~unwrap()  0.00 0.04 0.00 1 0.00 0.00 void arma::eop\_core<arma::eop\_sqrt>::apply<arma::Mat<double>, arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans> >(arma::Mat<double>&, arma::eOp<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans>, arma::eop\_sqrt> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::podarray<int>::init\_cold(unsigned long long)  0.00 0.04 0.00 1 0.00 0.00 arma::podarray<int>::~podarray()  0.00 0.04 0.00 1 0.00 0.00 arma::Op\_traits<arma::Op<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv>, arma::op\_htrans, true>::Op\_traits()  0.00 0.04 0.00 1 0.00 0.00 arma::Op\_traits<arma::Mat<double>, arma::op\_inv, true>::Op\_traits()  0.00 0.04 0.00 1 0.00 0.00 arma::Op\_traits<arma::Mat<double>, arma::op\_htrans, true>::Op\_traits()  0.00 0.04 0.00 1 0.00 0.00 arma::Op\_traits<arma::Glue<arma::Glue<arma::Mat<double>, arma::Op<arma::Mat<double>, arma::op\_inv>, arma::glue\_times>, arma::Mat<double>, arma::glue\_times>, arma::op\_inv, true>::Op\_traits()  0.00 0.04 0.00 1 0.00 0.00 void arma::op\_htrans::apply<arma::Mat<double> >(arma::Mat<arma::Mat<double>::elem\_type>&, arma::Op<arma::Mat<double>, arma::op\_htrans> const&, arma::arma\_not\_cx<arma::Mat<double>::elem\_type>::result const\*)  0.00 0.04 0.00 1 0.00 0.00 void arma::op\_strans::apply\_proxy<arma::Mat<double> >(arma::Mat<arma::Mat<double>::elem\_type>&, arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 void arma::op\_strans::apply\_mat\_noalias<double, arma::Mat<double> >(arma::Mat<double>&, arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 void arma::op\_strans::apply\_mat<double, arma::Mat<double> >(arma::Mat<double>&, arma::Mat<double> const&)  0.00 0.04 0.00 1 0.00 0.00 arma::strip\_inv<arma::Op<arma::Mat<double>, arma::op\_inv> >::strip\_inv(arma::Op<arma::Mat<double>, arma::op\_inv> const&)  0.00 0.04 0.00 1 0.00 0.00 wxFile::~wxFile()  0.00 0.04 0.00 1 0.00 0.00 dataApp::~dataApp()  0.00 0.04 0.00 1 0.00 0.00 dataApp::~dataApp()  0.00 0.04 0.00 1 0.00 0.00 wxChoice::Init()  0.00 0.04 0.00 1 0.00 0.00 wxChoice::wxChoice(wxWindow\*, int, wxPoint const&, wxSize const&, int, wxString const\*, long, wxValidator const&, wxString const&)  0.00 0.04 0.00 1 0.00 0.00 wxDialog::wxDialog()  0.00 0.04 0.00 1 0.00 0.00 int wxString::Printf<double, double, double, double, double>(wxFormatString const&, double, double, double, double, double)  0.00 0.04 0.00 1 0.00 0.00 \_\_gnu\_cxx::new\_allocator<int>::new\_allocator()  0.00 0.04 0.00 1 0.00 0.00 \_\_gnu\_cxx::new\_allocator<int>::~new\_allocator()  0.00 0.04 0.00 1 0.00 0.00 wxAppBase::SetTopWindow(wxWindow\*)  0.00 0.04 0.00 1 0.00 0.00 wxDateTime::GetDay(wxDateTime::TimeZone const&) const  0.00 0.04 0.00 1 0.00 0.00 wxDateTime::GetYear(wxDateTime::TimeZone const&) const  0.00 0.04 0.00 1 0.00 0.00 wxDateTime::GetMonth(wxDateTime::TimeZone const&) const  0.00 0.04 0.00 1 0.00 0.00 wxWindowBase::HasFlag(int) const  0.00 0.04 0.00 1 0.00 0.00 wxEventTypeTag<wxDateEvent>::operator int const&() const  0.00 0.04 0.00 1 0.00 0.00 wxEventTypeTag<wxSashEvent>::operator int const&() const  0.00 0.04 0.00 1 0.00 0.00 wxEventTypeTag<wxInitDialogEvent>::operator int const&() const  0.00 0.04 0.00 1 0.00 0.00 wxTimePickerCtrlBase::GetTime(int\*, int\*, int\*) const  0.00 0.04 0.00 1 0.00 0.00 wxControlWithItemsBase::ShouldInheritColours() const  0.00 0.04 0.00 1 0.00 0.00 arma::Mat<double>::max() const  0.00 0.04 0.00 1 0.00 0.00 std::allocator<int>::allocator()  0.00 0.04 0.00 1 0.00 0.00 std::allocator<int>::~allocator()  0.00 0.04 0.00 1 0.00 0.00 std::char\_traits<char>::length(char const\*)  0.00 0.04 0.00 1 0.00 0.00 std::\_Vector\_base<int, std::allocator<int> >::\_Vector\_impl::\_Vector\_impl()  0.00 0.04 0.00 1 0.00 0.00 std::\_Vector\_base<int, std::allocator<int> >::\_Vector\_impl::~\_Vector\_impl()  0.00 0.04 0.00 1 0.00 0.00 std::\_Vector\_base<int, std::allocator<int> >::\_Vector\_base()  0.00 0.04 0.00 1 0.00 0.00 std::\_Vector\_base<int, std::allocator<int> >::~\_Vector\_base()  0.00 0.04 0.00 1 0.00 0.00 std::numeric\_limits<double>::epsilon()  0.00 0.04 0.00 1 0.00 0.00 std::numeric\_limits<double>::infinity()  0.00 0.04 0.00 1 0.00 0.00 std::vector<int, std::allocator<int> >::vector()  0.00 0.04 0.00 1 0.00 0.00 std::vector<int, std::allocator<int> >::~vector()  0.00 0.04 0.00 1 0.00 0.00 std::operator|(std::\_Ios\_Openmode, std::\_Ios\_Openmode)  0.00 0.04 0.00 1 0.00 0.00 std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > std::operator+<char, std::char\_traits<char>, std::allocator<char> >(std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> >&&, char const\*)  0.00 0.04 0.00 1 0.00 0.00 std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > std::operator+<char, std::char\_traits<char>, std::allocator<char> >(char const\*, std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > const&)  0.00 0.04 0.00 1 0.00 0.00 std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > std::operator+<char, std::char\_traits<char>, std::allocator<char> >(std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > const&, std::\_\_cxx11::basic\_string<char, std::char\_traits<char>, std::allocator<char> > const&)  0.00 0.04 0.00 1 0.00 0.00 parse  0.00 0.04 0.00 1 0.00 0.00 parseGLNS  % the percentage of the total running time of the  time program used by this function.  cumulative a running sum of the number of seconds accounted  seconds for by this function and those listed above it.  self the number of seconds accounted for by this  seconds function alone. This is the major sort for this  listing.  calls the number of times this function was invoked, if  this function is profiled, else blank.  self the average number of milliseconds spent in this  ms/call function per call, if this function is profiled,  else blank.  total the average number of milliseconds spent in this  ms/call function and its descendents per call, if this  function is profiled, else blank.  name the name of the function. This is the minor sort  for this listing. The index shows the location of  the function in the gprof listing. If the index is  in parenthesis it shows where it would appear in  the gprof listing if it were to be printed. |