

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет «МЭИ»

Институт: ИРЭ

Кафедра: Радиотехнических систем

Специальность:

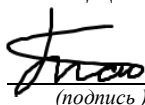
11.05.01 Радиоэлектронные системы и  
комплексы

**ОТЧЕТ по практике**

**Наименование  
практики:**

Производственная практика: научно-  
исследовательская работа

**СТУДЕНТ**



(подпись)

/ Тасканов В.Е.

(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

<b>ГЛАВА 1 ДОБАВЛЕНИЕ РАСЧЕТА КООРДИНАТ НС .....</b>	<b>5</b>
1.1. Алгоритм расчета для ГНСС GPS .....	6
1.1.2. Алгоритм расчета координат .....	8
1.3. Алгоритм расчета для ГНСС ГЛОНАСС .....	10
1.3.2. Алгоритм расчета координат .....	12
1.4. Алгоритм расчета ионосферной погрешности .....	15
<b>ГЛАВА 2 РЕАЛИЗАЦИЯ АЛГОРИТМОВ В ПРОГРАММЕ .....</b>	<b>18</b>
2.1. Скачивание файла .....	18
2.2. Обработка файла .....	18
2.2.1. Обработка файла GPS .....	18
2.2.1. Обработка файла ГЛОНАСС .....	18
2.3. Расчет координат .....	19
2.4. Расчет времени .....	19
2.3. Изменение интерфейса программы .....	20
2.4. Необходимые файлы для сборки проекта .....	21
2.5. Профилирование проекта .....	24
<b>ГЛАВА 3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ .....</b>	<b>26</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>32</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>32</b>
<b>ПРИЛОЖЕНИЕ .....</b>	<b>33</b>

## 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 возвышения 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 возвышения 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,
- Обработываем файл,
- Рассчитываем координаты

Файл будет скачивать с сервера «информационно-аналитического центра координатно-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 – *agr* ( 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 – agr, содержатся альманахи, записанные в виде строк:

```

МССJ_200108 – Блокнот
Файл Правка Формат Вид Справка
07 01 2020 00871
1 0 2087 405504 09 01 2020 59904.000 -0.25749207E-03 -0.10913936E-10 -0.24811015E-08
-0.22440493E+00 0.31148988E+00 0.24107289E+00 0.92692375E-02 0.51535981E+04 0.40129888E+00
07 01 2020 00872
2 0 2087 405504 09 01 2020 59904.000 -0.38242340E-03 -0.72759576E-11 -0.25465852E-08
-0.24777687E+00 0.30480462E+00 -0.53421688E+00 0.19700527E-01 0.51535654E+04 0.49287593E+00
07 01 2020 00873
3 0 2087 405504 09 01 2020 59904.000 -0.68664551E-04 -0.72759576E-11 -0.24156179E-08
0.10692251E+00 0.30714494E+00 0.24356914E+00 0.26187897E-02 0.51535933E+04 0.32618165E-01
07 01 2020 00874
4 1 2087 405504 09 01 2020 59904.000 -0.31471252E-04 -0.36379788E-11 -0.25320332E-08
0.45012069E+00 0.30539781E+00 -0.86515725E+00 0.45394897E-03 0.51536631E+04 0.80713654E+00
07 01 2020 00875
5 0 2087 405504 09 01 2020 59904.000 -0.57220459E-05 0.00000000E+00 -0.24483597E-08
0.97382665E-01 0.30297548E+00 0.25501275E+00 0.57921410E-02 0.51535308E+04 -0.68326879E+00
07 01 2020 00876
6 0 2087 405504 09 01 2020 59904.000 -0.17738342E-03 -0.10913936E-10 -0.24847395E-08
-0.22703457E+00 0.31136209E+00 -0.37527728E+00 0.17843246E-02 0.51536895E+04 0.50334632E+00
07 01 2020 00877
7 0 2087 405504 09 01 2020 59904.000 -0.18882751E-03 -0.72759576E-11 -0.24629117E-08
0.77744925E+00 0.30363160E+00 -0.77044582E+00 0.13245106E-01 0.51536509E+04 0.17412102E+00
07 01 2020 00878
8 0 2087 405504 09 01 2020 59904.000 -0.20027161E-04 0.00000000E+00 -0.26266207E-08
-0.56470180E+00 0.30871850E+00 -0.68337798E-01 0.49986839E-02 0.51535586E+04 -0.83613443E+00
07 01 2020 00879
9 0 2087 405504 09 01 2020 59904.000 -0.12874603E-03 -0.72759576E-11 -0.25574991E-08
0.43520999E+00 0.30307466E+00 0.54175544E+00 0.17147064E-02 0.51534614E+04 -0.74238789E+00
07 01 2020 00880
10 0 2087 405504 09 01 2020 59904.000 -0.20313263E-03 -0.14551915E-10 -0.23974280E-08
0.10593116E+00 0.30713922E+00 -0.86190462E+00 0.52723885E-02 0.51536206E+04 -0.24406230E+00
07 01 2020 00881

```

Рисунок 1 – Пример скаченного файла с расширением agr

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/с, $\dot{\Delta\Omega}^{****}$

Строка 3

1	Om0 - долгота узла, полуциклы, $\Omega_{0-n}$
2	I - наклонение, полуциклы, $i_{0-n}$
3	w - аргумент перигея, полуциклы, $\omega_n$
4	E – эксцентриситет, $e_n$

5	SQRT(A) - корень из большой полуоси, $m \cdot 0.5$ , $\sqrt{A_0}$
6	$M_0$ - средняя аномалия, полуциклы, $M_{0-n}$

### 1.1.2. Алгоритм расчета координат

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

- 1.1.2.1. Определим время, отсчитываемое от опорной эпохи эфемерид:

$$t_k = t - t_{oc}$$

- 1.1.2.2. Определим среднее движение:

$$n_0 = \sqrt{\frac{\mu}{A_0^3}}$$

- 1.1.2.3. Определим скорректированное среднее движение:

$$n_A = n_0 + \Delta n$$

- 1.1.2.4. Определим среднюю аномалию:

$$M_k = M_0 + n_A \cdot t_k$$

- 1.1.2.5. Решим уравнение Кеплера минимум 3-мя итерациями и определим  $E_k$ :

$$M_k = E_k - e_n \cdot \sin(E_k) \Rightarrow E_k = M_k + e_n \cdot \sin(E_k)$$

- 1.1.2.6. Определим истинную аномалию:

$$v_k = \arctg \left( \frac{\sqrt{1 - e_n^2} \sin(E_k)}{(\cos(E_k) - e_n)} \right)$$

- 1.1.2.7. Определим скорректированный радиус орбиты спутника:

$$A_k = A_0 + \left( \dot{A} \right) t_k$$



$$r_k = A_k (1 - e_n \cos(E_k)) + \cancel{\delta r_k}$$

1.1.2.8. Определим аргумент широты:

$$\Phi_k = v_k + \omega$$

$$u_k = \Phi_k + \cancel{\delta u_k}$$

1.1.2.9. Определим координаты НС в орбитальной плоскости:

$$\begin{cases} x'_k = r_k \cdot \cos(u_k) \\ y'_k = r_k \cdot \sin(u_k) \end{cases}$$

1.1.2.10. Определим скорректированную долготу восходящего узла  $\Omega_k$  определяется из соотношения:

$$\dot{\Omega} = \dot{\Omega}_{REF} + \cancel{\Delta \dot{\Omega}}$$

$$\Omega_k = \Omega_{0-n} + \left( \dot{\Omega} - \dot{\Omega}_e \right) t_{oe}$$

1.1.2.11. Определим скорректированное наклонение орбиты спутника

$$i_k = i_{0-n} + (\cancel{i_{0-n} - DOT}) t_k + \cancel{\delta i_k}$$

1.1.2.12. Определим координаты НС в геоцентрической системе координат:

$$\begin{cases} x_k = x'_k \cos \Omega_k - y'_k \cos i_k \sin \Omega_k \\ y_k = x'_k \sin \Omega_k + y'_k \cos i_k \cos \Omega_k \\ z_k = y'_k \sin i_k \end{cases}$$

### 1.3. Алгоритм расчета для ГНСС ГЛОНАСС

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

```

01 01 2020 75600
1 1 1 31 12 2019 0.517487500E+04 0.000000000E+00 0.000000000E+00 -0.534057617E-04
0.4236565E+00 0.6714821E-02 0.1851501E+00 0.3643036E-03 -0.2656244E+04 0.8544922E-03
01 01 2020 75601
2 -4 1 31 12 2019 0.111991563E+05 0.000000000E+00 0.000000000E+00 -0.404357910E-03
0.2906666E+00 0.9313583E-02 -0.7553711E+00 0.1609802E-02 -0.2656010E+04 0.1037598E-02
01 01 2020 75602
3 5 1 31 12 2019 0.162281563E+05 0.000000000E+00 0.000000000E+00 -0.381469727E-05
0.1726351E+00 0.8380890E-02 -0.7248840E+00 0.1530647E-02 -0.2656043E+04 0.1037598E-02
06 01 2020 1803
4 6 1 05 01 2020 0.351999375E+05 0.000000000E+00 0.000000000E+00 -0.343322754E-04
-0.2965450E+00 0.1013565E-01 -0.5179749E+00 0.6799698E-03 -0.2656115E+04 0.1708984E-02
01 01 2020 75604
5 1 1 31 12 2019 0.262059375E+05 0.000000000E+00 0.000000000E+00 -0.419616699E-04
-0.5873966E-01 0.1023102E-01 -0.9271851E+00 0.4472733E-03 -0.2656137E+04 0.1098633E-02
01 01 2020 75605
6 -4 1 31 12 2019 0.317489688E+05 0.000000000E+00 0.000000000E+00 -0.167846680E-03
-0.1937952E+00 0.6618500E-02 0.7283020E+00 0.7905960E-03 -0.2656008E+04 0.1220703E-02
01 01 2020 75606
7 5 1 31 12 2019 0.366605000E+05 0.000000000E+00 0.000000000E+00 0.305175781E-04
-0.3013716E+00 0.8426666E-02 -0.7244873E+00 0.1291275E-02 -0.2656018E+04 0.1159668E-02
01 01 2020 75607
8 6 1 31 12 2019 0.920250000E+03 0.000000000E+00 0.000000000E+00 -0.114440918E-04
0.5282898E+00 0.8465767E-02 -0.5527344E+00 0.1614571E-02 -0.2656029E+04 0.9155273E-03
06 01 2020 1808
9 -2 1 05 01 2020 0.193098437E+05 0.000000000E+00 0.000000000E+00 -0.106811523E-03
0.7387104E+00 0.5420685E-02 -0.8858643E+00 0.1379013E-02 -0.2655211E+04 -0.1892090E-02
01 01 2020 75609
10 -5 1 31 12 2019 0.119666250E+05 0.000000000E+00 0.000000000E+00 0.534057617E-04
0.9348469E+00 0.9264946E-02 -0.4678345E-01 0.1546860E-02 -0.2689875E+04 -0.6103516E-04
01 01 2020 75610
  
```

Рисунок 2 – 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 наклона, 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:

1.3.2.1. Определяется интервал прогноза в секундах:

$$\Delta t_{\text{пр}} = \Delta N_A \cdot 86400 + (t_i - t_{\lambda_A}),$$

$$\Delta N_A = \begin{cases} N - N_A - \left\langle \left\langle \frac{N - N_A}{1461} \right\rangle \right\rangle \cdot 1461 & \text{если } N_4 \neq 27, \\ N - N_A - \left\langle \left\langle \frac{N - N_A}{1460} \right\rangle \right\rangle \cdot 1460 & \text{если } N_4 = 27; \end{cases}$$

Где:

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

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

$\langle \langle x \rangle \rangle$  – вычисление целого, ближайшего к  $x$ .

1.3.2.2. Рассчитывается количество целых витков  $W$  на интервале прогноза:

$$W = \left\langle \frac{\Delta t_{\text{пр}}}{T_{\text{ср}} + \Delta T_A} \right\rangle,$$

где  $\langle x \rangle$  выделение целой части  $x$ ;

1.3.2.3. Определяется текущее наклонение:

$$i = \left( \frac{i_{\text{ф}}}{180^\circ} + \Delta i_A \right) \cdot \pi \text{ рад},$$

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

$$T_{\text{др}} = T_{\text{ср}} + \Delta T_A + (2W + 1) \cdot \Delta \dot{T}_A,$$

$$n = 2\pi / T_{\text{др}},$$

1.3.2.5. Методом последовательных приближений  $m = 0, 1, 2 \dots$  рассчитывается большая полуось орбиты  $a$ :

$$a^{(m+1)} = \sqrt[3]{\left(\frac{T_{\text{оск}}^{(m)}}{2\pi}\right)^2 \cdot GM};$$

$$p^{(m+1)} = a^{(m+1)}(1 - (\varepsilon_A)^2);$$

$$T_{\text{оск}}^{(m+1)} = \frac{T_{\text{др}}}{1 - \frac{3}{2} \cdot J_2^0 \left(\frac{a_e}{p^{(m+1)}}\right)^2 \left[ \left(2 - \frac{5}{2} \cdot \sin^2 i\right) \cdot \frac{(1 - (\varepsilon_A)^2)^{3/2}}{(1 + \varepsilon_A \cdot \cos(\omega_A \pi))^2} + \frac{(1 + \varepsilon_A \cdot \cos(\omega_A \pi))^3}{1 - (\varepsilon_A)^2} \right]},$$

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

$$\lambda = \lambda_A \cdot \pi - \left\{ \omega_3 + \frac{3}{2} J_2^0 \cdot n \cdot \left(\frac{a_e}{p}\right)^2 \cos i \right\} \Delta t_{\text{пр}};$$

$$\omega = \omega_A \cdot \pi - \frac{3}{4} J_2^0 n \left(\frac{a_e}{p}\right)^2 (1 - 5 \cos^2 i) \cdot \Delta t_{\text{пр}},$$

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

$$L_1 = \omega + E_0 - \varepsilon_A \sin E_0,$$

$$E_0 = -2 \cdot a \tan \left( \sqrt{\frac{1 - \varepsilon_A}{1 + \varepsilon_A}} \cdot \tan \frac{\omega}{2} \right).$$

Где

1.3.2.8. Определяется текущее значение средней долготы НКА:

$$L = L_1 + n(\Delta t_{\text{np}} - (T_{\text{cp}} + \Delta T_A)W - \Delta \dot{T}_A W^2).$$

1.3.2.9. Определяется эксцентрическая аномалия путем решения уравнения Кеплера

$$L - \omega = E - \varepsilon \cdot \sin E.$$

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

$$E^{(m+1)} = L - \omega + \varepsilon \cdot \sin E^{(m)},$$

1.3.2.10. Вычисляются истинная аномалия и аргумент широты НКА  $u$ :

$$v = 2 \arctan \left( \sqrt{\frac{1 + \varepsilon_A}{1 - \varepsilon_A}} \tan \frac{E}{2} \right);$$

$$u = v + \omega.$$

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

$$p = a(1 - (\varepsilon_A)^2);$$

$$r = \frac{p}{1 + \varepsilon_A \cos u};$$

$$x(t_i) = r(\cos \lambda \cos u - \sin \lambda \sin u \cos i);$$

$$y(t_i) = r(\sin \lambda \cos u + \cos \lambda \sin u \cos i);$$

$$z(t_i) = r \sin u \sin i.$$

#### 1.4. Алгоритм расчета ионосферной погрешности

Воспользуемся алгоритмом расчета из ИКД GPS

Модель коррекции модели ионосферы

$$T_{iono} = \begin{cases} F \cdot \left[ 5 \cdot 10^{-9} + AMP \cdot \left( 1 - \frac{x^2}{2} + \frac{x^4}{24} \right) \right], |x| < 1.57 \\ F \cdot [5 \cdot 10^{-9}], |x| \geq 1.57 \end{cases}, [сек]$$

Определим  $AMP$

$$AMP = \begin{cases} \sum_{n=0}^3 a_n \phi_m^n, AMP \geq 0 \\ \text{Если } AMP < 0, AMP = 0 \end{cases}, [сек]$$

Где  $a_n$  - коэффициенты кубического уравнения, представляющие амплитуду вертикальной задержки

Определим фазу

$$x = \frac{2\pi(t - 50400)}{PER}, [рад]$$

Определим  $PER$

$$PER = \begin{cases} \sum_{n=0}^3 \beta_n \phi_m^n, PER \geq 72.000 \\ \text{Если } PER < 72.000, PER = 72.000 \end{cases}, [сек]$$

Где  $\beta_n$  - коэффициенты кубического уравнения, представляющие период модели

Определим коэффициент наклона

$$F = 1 + 16[0.53 - E]^3$$

Где  $E$  - угол возвышения между пользователем и спутником

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

$$\phi_m = \phi_i + 0.064 \cos(\lambda_i - 1.617), [\text{полуцикл}]$$

Определим геодезическая долгота земной проекции точки пересечения ионосферы

$$\lambda_i = \lambda_u + \frac{\psi \sin A}{\cos \phi_i}, [\text{полуцикл}]$$

Определим геодезическая широта земной проекции точки пересечения ионосферы

$$\phi_i = \left\{ \begin{array}{l} \phi_u + \cos A, |\phi_i| \leq 0.416 \\ \text{Если } \phi_i > 0.416, \text{ тогда } \phi_i = \phi_i + 0.416 \\ \text{Если } \phi_i < -0.416, \text{ тогда } \phi_i = \phi_i - 0.416 \end{array} \right\}, [\text{полуциклы}]$$

$$\phi_i = \left\{ \begin{array}{l} \phi_u + \psi \cos A, \quad |\phi_i| \leq 0.416 \\ \text{if } \phi_i > +0.416, \text{ then } \phi_i = +0.416 \\ \text{if } \phi_i < -0.416, \text{ then } \phi_i = -0.416 \end{array} \right\} \quad (\text{semi-circles})$$

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

$$\psi = \frac{0.0137}{E + 0.11} - 0.022, [\text{полуциклы}]$$

Определим локальное время:



$$t = 4.32 \cdot 10^4 \cdot \lambda_i + time, [сек]$$

Где  $t \in 0 \leq t < 86400$ ,

*time* - вычисленное системное время приемника

Значения коэффициентов  $a_n, \beta_n$  берется из файла, который будем скачивать с сервера «информационно-аналитического центра координатно-временного и навигационного обеспечения, по адресу: «ftp://ftp.glonass-ias.ru/MCC/BRDC/», расширение файла 21n

```

BRDC0010 (1) - Блокнот
Файл  Правка  Формат  Вид  Справка
2.10      N: GPS NAV DATA      RINEX VERSION / TYPE
EPHEMCHK 2.10      MCC D.ZALETAEV      03-Jan-21 20:00      PGM / RUN BY / DATE
A new version of accumulating navigation files by IANC      COMMENT
7.4510D-09 -1.4900D-08 -5.9600D-08 1.1920D-07      ION ALPHA
9.0110D+04 -6.5540D+04 -1.3110D+05 4.5880D+05      ION BETA
-3.725290298460D-09-1.065814103640D-14 61440      2139 DELTA-UTC: A0,A1,T,W
18      LEAP SECONDS
END OF HEADER
1 21 1 1 0 0 0.0 7.875198498368D-04-5.911715561524D-12 0.00000000000D+00
5.10000000000D+01-7.57500000000D+01 4.207318108830D-09-1.021277967070D+00
-3.935769200325D-06 1.022381347138D-02 1.428648829460D-06 5.153692249298D+03
4.32000000000D+05 2.216547727585D-07-8.086751269421D-01 2.309679985046D-07
9.827440567889D-01 3.61937500000D+02 8.220205737136D-01-8.233557246422D-09
-3.185846989029D-10 1.00000000000D+00 2.13800000000D+03 0.00000000000D+00
2.00000000000D+00 0.00000000000D+00 5.122274160385D-09 5.10000000000D+01
4.24818000000D+05 4.00000000000D+00
2 21 1 1 0 0 0.0-5.608978681266D-04-4.320099833421D-12 0.00000000000D+00
8.30000000000D+01-6.36875000000D+01 4.862702551080D-09-7.825898141899D-01
-3.134831786156D-06 2.031416934915D-02 1.531094312668D-06 5.153582004547D+03
4.32000000000D+05-2.365559339523D-07-8.894724859850D-01 1.154839992523D-07
9.615078420210D-01 3.55593750000D+02-1.567745133186D+00-8.812509933554D-09
-1.464346710204D-10 1.00000000000D+00 2.13800000000D+03 0.00000000000D+00
2.00000000000D+00 0.00000000000D+00-1.769512891769D-08 8.30000000000D+01
4.24818000000D+05 4.00000000000D+00
3 21 1 1 0 0 0.0-4.453537985682D-05-9.890754881781D-12 0.00000000000D+00
5.10000000000D+01-1.09875000000D+02 3.918020344131D-09-2.198381241254D+00
-5.602836608887D-06 3.261363017373D-03 1.010671257973D-05 5.153687782288D+03
4.32000000000D+05-2.980232238770D-08 2.298830274365D-01-7.264316082001D-08
9.68224906020D-01 1.89781250000D+02 8.808689872262D-01-7.586744589793D-09
1.292910997790D-10 1.00000000000D+00 2.13800000000D+03 0.00000000000D+00
2.00000000000D+00 0.00000000000D+00 1.862645149231D-09 5.10000000000D+01
4.24818000000D+05 4.00000000000D+00
4 21 01 01 00 0.0-1.690830104053D-04-3.069544618484D-12 0.00000000000D+00
8.00000000000D+01 8.21875000000D+00 4.643050388609D-09 7.181579851113D-01
4.731118679047D-07 1.009354135022D-03 7.683411240578D-06 5.153595529556D+03
4.32000000000D+05 2.793967723846D-08 1.309591313346D+00 1.676380634308D-08
9.598456408522D-01 2.30781250000D+02-3.072245845474D+00-7.931044265774D-09
5.975248629220D-10 1.00000000000D+00 2.13800000000D+03 0.00000000000D+00
2.00000000000D+00 0.00000000000D+00-4.190951585770D-09 8.00000000000D+01
4.24740000000D+05 0.00000000000D+00
5 21 01 01 00 0.0-2.980604767799D-05-1.023181539495D-12 0.00000000000D+00
9.80000000000D+01-1.07406250000D+02 4.315893864515D-09 1.814955427561D+00
-5.638226866722D-06 6.027893046848D-03 1.027062535286D-05 5.153797132492D+03
4.32000000000D+05-7.636845111847D-08 1.950148724094D-01-6.332993507385D-08
9.551620486418D-01 1.81500000000D+02 8.897933829174D-01-7.873184770801D-09
6.857428513918D-11 1.00000000000D+00 2.13800000000D+03 0.00000000000D+00
2.00000000000D+00 0.00000000000D+00-1.117587089539D-08 9.80000000000D+01
4.24740000000D+05 0.00000000000D+00

```

Рисунок 3 – 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 ранее 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.C»

#### **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.C»

## 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

The screenshot shows a window titled "Data app" with a close button (X) in the top right corner. Inside the window, there is a dropdown menu at the top left showing "Glonass". Below it is a table with 24 rows and 3 columns. The first column contains numbers 1 through 24. The second column is labeled "SISRE, m" and the third column is labeled "SISVE, mm/s". To the right of the table is a panel with several input fields and buttons. At the top right of the panel is a button labeled "Загрузить". Below it is a label "Время прогнозирования по UTC(+3)" followed by a date picker showing "07.06.2021" and a time picker showing "0:05:09". Below these are three labels: "Введите значение высоты в метрах:", "Введите значение В в градусах:", and "Введите значение L в градусах:", each followed by a text input field containing the number "0". At the bottom of the panel is a button labeled "Обработка" and a label "Значения СКО:".

	SISRE, m	SISVE, mm/s
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

Рисунок 4 – 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 фильтрация 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 «C», а приложение 1 на языке «C++», 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.wxh”, код описан в приложение 14 – это файл описания графического пользовательского интерфейса для плагина wxSmith, путь к файлу: “...\data\ wxsmith\datadiaslog.wxh”.

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 «C», а приложение 1 на языке «C++», 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 встроенной утилиты – Gprof.

Получим результаты



Flat profile:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self ms/call	total 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::ContainsHWNDR(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<doub
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>
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, 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*, unsig
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*)

Рисунок 5 – 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:*

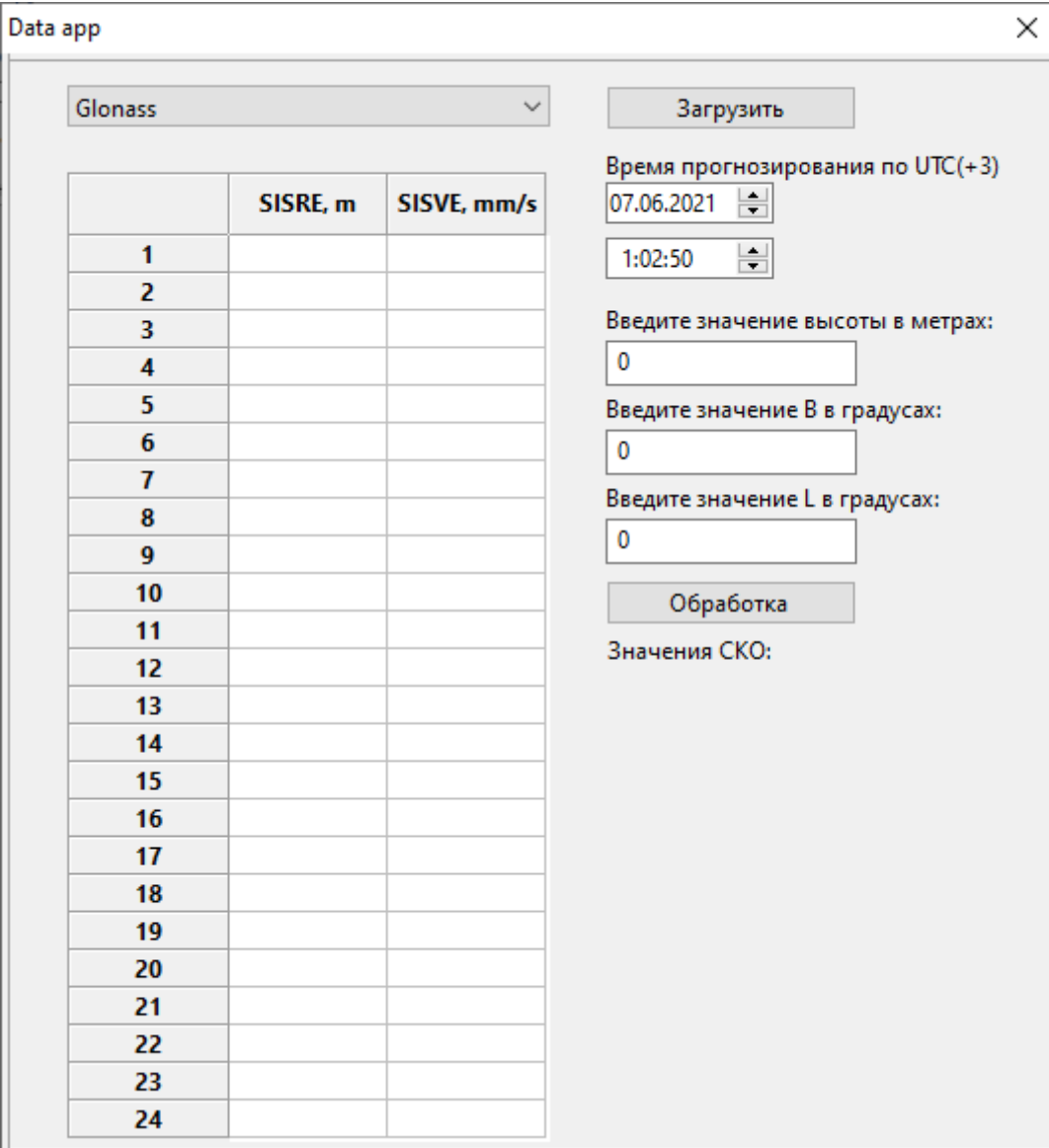
Для запуска программы необходимо иметь windows 7/10 и подключенное устройство к интернету.

### *Инструкция:*

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

Если программа не запустится отключите антивирус.

У вас появится диалоговое окно:



The screenshot shows a window titled "Data app" with a close button (X) in the top right corner. Inside the window, there is a dropdown menu at the top left showing "Glonass". Below it is a table with 24 rows and 3 columns. The first column contains numbers 1 through 24. The second column is labeled "SISRE, m" and the third column is labeled "SISVE, mm/s". To the right of the table, there is a "Загрузить" (Load) button. Below it, there are two date/time pickers: "Время прогнозирования по UTC(+3)" with a date of "07.06.2021" and a time of "1:02:50". Below these are three input fields for "Введите значение высоты в метрах:" (0), "Введите значение В в градусах:" (0), and "Введите значение L в градусах:" (0). Below the input fields is an "Обработка" (Processing) button. At the bottom right, there is a label "Значения СКО:".

	SISRE, m	SISVE, mm/s
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

Рисунок 3.1. Интерфейс программы

2. Выберите необходимую вам НС:

Data app

Glonass

Glonass

**GPS**

Galileo

Beidou

QZSS

2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

Загрузить

Время прогнозирования по UTC(+3)  
07.06.2021  
1:02:50

Введите значение высоты в метрах:  
0

Введите значение В в градусах:  
0

Введите значение L в градусах:  
0

Обработка

Значения СКО:

Рисунок 3.2. Интерфейс выбора НС

### 3. Нажмите кнопку «Загрузить»

**Data app**

Гlonass

	SISRE, m	SISVE, mm/s
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

**Загрузить**

Время прогнозирования по UTC(+3)  
07.06.2021  
1:02:50

Введите значение высоты в метрах:  
0

Введите значение В в градусах:  
0

Введите значение L в градусах:  
0

**Обработка**

Значения СКО:

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.

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

Data app

Glomass

Загрузить

	SISRE, m	SISVE, mm/s
1	0.427	0.508
2	1.205	0.650
3	0.863	0.511
4	0.851	0.569
5	0.682	0.597
6	0.000	0.000
7	0.686	0.469
8	2.104	0.506
9	2.751	1.216
10	0.000	0.000
11	0.000	0.000
12	1.390	0.753
13	1.451	0.657
14	0.914	0.540
15	0.519	0.562
16	1.392	0.580
17	0.440	0.701
18	1.481	0.551
19	1.023	0.591
20	1.190	0.768
21	0.849	0.636
22	1.324	0.796
23	0.000	0.000
24	0.690	0.534

Введите значение высоты:

Введите значение В:

Введите значение L:

Обработка

СКО для x:

СКО для y:

СКО для z:

СКО для D:

Рисунок 3.4. Полученные значения

5. Далее вводим значения:

Data app

Glomass

	SISRE, m	SISVE, mm/s
1	1.647	0.604
2	1.488	0.764
3	0.696	0.642
4	0.618	0.647
5	0.438	0.571
6	0.000	0.000
7	1.364	0.600
8	1.760	0.638
9	1.522	0.853
10	0.000	0.000
11	0.000	0.000
12	0.837	0.733
13	1.695	0.714
14	0.802	0.633
15	0.933	0.709
16	1.664	0.680
17	0.896	0.737
18	0.974	0.635
19	1.029	0.782
20	0.868	0.819
21	1.534	0.641
22	1.183	0.618
23	0.000	0.000
24	1.499	0.639

Загрузить

Время прогнозирования по UTC(+3)  
07.06.2021

1:07:05

Введите значение высоты в метрах:  
200

Введите значение В в градусах:  
37.611680

Введите значение L в градусах:  
55.819715

Обработка

Значения СКО:

Рисунок 3.5. Ввод значений в формы

6. Нажимаем кнопку обработка:

The screenshot shows a software window titled "Data app" with a close button (X) in the top right corner. Inside the window, there is a dropdown menu set to "Glonass". Below it is a table with 24 rows and 3 columns. The first column contains numbers 1 through 24. The second column is labeled "SISRE, m" and the third is labeled "SISVE, mm/s". To the right of the table, there is a "Загрузить" (Load) button. Below this button, there are input fields for "Время прогнозирования по UTC(+3)" (Prediction time in UTC(+3)) with the value "07.06.2021", and a time field with "1:07:39". Further down, there are input fields for "Введите значение высоты в метрах:" (Enter height value in meters) with "200", "Введите значение В в градусах:" (Enter value B in degrees) with "37.611680", and "Введите значение L в градусах:" (Enter value L in degrees) with "55.819715". Below these is an "Обработка" (Processing) button. At the bottom right, there is a section titled "Значение СКО:" (SD value) with the following text: "СКО для x: 1.478 м", "СКО для y: 1.681 м", "СКО для z: 1.665 м", and "СКО для D: 1.480, м СКО: 2.790 м".

	SISRE, m	SISVE, mm/s
1	1.647	0.604
2	1.488	0.764
3	0.696	0.642
4	0.618	0.647
5	0.438	0.571
6	0.000	0.000
7	1.364	0.600
8	1.760	0.638
9	1.522	0.853
10	0.000	0.000
11	0.000	0.000
12	0.837	0.733
13	1.695	0.714
14	0.802	0.633
15	0.933	0.709
16	1.664	0.680
17	0.896	0.737
18	0.974	0.635
19	1.029	0.782
20	0.868	0.819
21	1.534	0.641
22	1.183	0.618
23	0.000	0.000
24	1.499	0.639

Рисунок 3.6. Полученные значения

Получаем значения СКО.

7. 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 затянувшегося 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 «информационно-аналитического центра координатно-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, __gnu_cxx::__promote<int, std::__is_integer<double>::__value>::__type, std::pow<double, int>(double, int)>::__type>::__type
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)
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,
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_Ke
yEx()
0.00          0.04          0.00          4          0.00          0.00
wxLongToLongHashMap_wxImplementation_HashTable::wxLongToLongHashMap_wxImplementatio
n_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::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::char_traits<char>,
std::allocator<char> > const&

```

```

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::Op<arma::Op<arma::Op<arma::Glue<arma::Glue<arma::Ma
t<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<doub
le>, 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::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.

