Расчет H, СКО и Dn

Матрица **H:**

|  |  |
| --- | --- |
|  | (1.1) |

где:

|  |  |
| --- | --- |
|  | (1.2) |

Матрица **:**

|  |  |
| --- | --- |
|  | (1.3) |

где n зависит от количества видимых спутников

Выведем значение СКО:



Тогда СКО примет вид:

|  |  |
| --- | --- |
|  | (1.4) |

Для работы с матрицами воспользуемся встроенной библиотекой: “Armadillo”.

Объявим матрицу : “mat Dn;”

Чтобы не появились аномальные числа выделим необходимый нам массив данных и приравняем его к нулю: “Dn.zeros(inum, inum);”

Матрица Dn приняла вид: 

Заполним диагональ матрицы значениями: ( из п. 1.2.1).

Найдем обратную матрицу “Dn” и запишем ее в матрицу “Dninv”.

Так как матрица квадратная, воспользуемся функцией “inv”.

Функция Y = inv(A) вычисляет матрицу, обратную квадратной матрице A. В случаях, когда матрица A плохо масштабирована или близка к вырожденной, выдаются сообщения.

Следовательно, “Dninv = inv(Dn)”;

Объявим матрицу H: “mat H(inum, 4); H.zeros();” создается матрицу размером (inum, 4).

…

Теперь найдем матрицу **СКО:** “mat sko = Dninv \*H;”

**Библиотека “Armadillo”**

Armadillo - это программная библиотека линейной алгебры с открытым исходным кодом для языка программирования C++ . Он нацелен на обеспечение эффективных и оптимизированных базовых вычислений. Синтаксис намеренно похож на Matlab.

Поддерживаются целые числа, числа с плавающей точкой и комплексные числа, а также подмножество тригонометрических и статистических функций. Различные разложения матриц обеспечиваются за счет необязательной интеграции с LAPACK или одной из его высокопроизводительных замен (например, многопоточных библиотек MKL или ACML).

Подход с отложенной оценкой используется (во время компиляции) для объединения нескольких операций в одну и уменьшения (или устранения) потребности во временных. Это достигается с помощью рекурсивных шаблонов и шаблонного метапрограммирования.