requirements enginering

matrix class ← pure virtual

Sparse

three lists as before

write an index method

banded

dense

- Just a data pointer
- with basic index and set functions

int dimensions (x andy)

int size

map info_map

holds String to detail keys for info map

Solve (b, method)

act as Solve method Switchboard

Solve (b, method, non-zero_elements, number_non-zero_elements)

through some kind of pre calculation we know which outputs will be non-zero

can see how fost the operation is on a boolean matrix

quick

use boolean dense matrix to find non-zero outputs

slow

use some other method to find non-zero elenents

virtual must be set per matrix class

Set (x, y, v) = 0

Set the value of an element (will be more complex for dense or banded matrixs)

index (x, y) = 0

used for default unoptemised methods

pivot_row (r_1, r_2) = 0

will need to be different for each matrix type

can be general then overwritten if needed

Print matrix()

used the above index as default

info()

give vairous details Such as Sparcity

throughout use

Shared_ptrs

So we can use weak_ptrs on the matrix and not worry about memory or multi-deleting of the array

Solve methods

→ gaussian_Sentinal

→ LU decom

→ Some form of iterative (maybe Jacobi)

because of the index method these can be written for all types bascally

then optimised by identifying