0.1 Método da Eliminação Gaussiana

Divide-se a 1ª equação por a_{11}, para tornar unitário o 1º coeficiente. Multiplica-se esta equação por $-a_{22}$, e soma-se à primeira; resultado será a nova 2ª equação; repete-se este processo para todas as linhas. Repete-se todos os passes utilizando os elementos da diagonal.

C++: (útil definir rowop como no Maxima

```
void rowop(vector<vector<double>> &matrix, size_t i, size_t j, double value) {
    for(size_t k = 0; k < matrix[0].size(); k++) {
        matrix[i][k] -= value * matrix[j][k];
    }
}

...

    for(size_t i = 0; i < matrix.size(); i++) {
        rowop(matrix, i, i, 1 - 1/matrix[i][i]);
        for(size_t j = 0; j < matrix.size(); j++) {
            if(i != j)
                rowop(matrix, j, i, matrix[j][i]);
        }
    }
```

O Erro no Método de Gauss:

- **Estabilidade Externa** (potenciais erros dos coeficientes e dos termos constantes): $A.\delta x = \delta b - \delta A.x$
- **Estabilidade Interna** (erros de arredondamento no decorrer do cálculo): $A.\delta = b - A.x_0 = \epsilon$ (ϵ = coluna dos resíduos)

Mínimização dos erros:

- **Pivotagem Parcial** (eliminar coluna com a equação com maior coeficiente nessa coluna)
- **Pivotagem Total** (eliminar com a equação não tratada com maior coeficiente)
- **Escalagem de Linhas**
- **Escalagem de Colunas**

0.2 Método de Khaletsky

Dado um sistema $A.x = b$, representa-se A por um produto $L.U$, $A = L.U$, e calcula-se x através dos sistems $L.y = b$ e $U.x = y$

Coeficientes LU:

- $l_{i,1} = a_{i,1}$
- $l_{i,j} = a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}.u_{k,j} (i \geq j)$
- $u_{1,i} = \frac{a_{1,i}}{l_{1,1}}$
- $u_{i,j} = \frac{a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}.u_{k,j}}{l_{i,i}} (i < j)$

0.3 Método de Gauss-Jacobi

$$x_i^{(k)} = \frac{1}{a_{i,i}}[- \sum_{j=1, j \neq i}^{n} a_{i,j} x_j^{(k-1)} + b_i], (1 \leq i \leq n)$$

0.4 Método de Gauss-Seidel

$$x_i^{(k)} = \frac{1}{a_{i,i}}[- \sum_{j=1, j<i}^{n} a_{i,j} x_j^{(k)} - \sum_{j=1, j>i}^{n} a_{i,j} x_j^{(k-1)} + b_i], (1 \leq i \leq n)$$