Project 2

Sverre Wehn Noremsaune & Frida Marie Engøy Westby
(Dated: September 26, 2021)

https://github.uio.no/comPhys/FYS3150/tree/project2/project2

PROBLEM 1 - SECOND ORDER DIFFERENTIAL EQUATION

In this problem we are looking at:

$$\gamma \frac{d^2 u(x)}{dx^2} = -F u(x) \tag{1}$$

and

$$\frac{d^2 u(\hat{x})}{dx^2} = -\lambda u(\hat{x}) \tag{2}$$

We know from the assignment that $\hat{x} \equiv x/L \Rightarrow x = \hat{x}L$ and $\lambda = \frac{FL^2}{\gamma}$, and by appling this, we can see that (2) is the scaled version of (1):

$$\gamma \frac{d^2 u(x)}{dx^2} = -F u(x)$$
$$\frac{d^2 u(x)}{dx^2} = -\frac{F u(x)}{\gamma}$$

Put in $x = \hat{x}L$:

$$\frac{d^2 u(\hat{x}L)}{d(\hat{x}L)^2} = -\frac{F u(\hat{x}L)}{\gamma}$$
$$\frac{d^2 u(\hat{x})L}{d(\hat{x})^2 L^2} = -\frac{F u(\hat{x})L}{\gamma}$$

Then we multiplies both sides with L:

$$\frac{d^2 u(\hat{x})L^2}{d(\hat{x})^2 L^2} = -\frac{F u(\hat{x})L^2}{\gamma}$$

Now we can use $\lambda = \frac{FL^2}{\gamma}$ and we easily see that we get (2). $\blacksquare$

PROBLEM 2 - JACOBI'S ROTATION ALGORITHM

From the assignment we have:

- $\vec{v}_i$ is a set of orthonormal basis vectors: $\vec{v}_i^T \cdot \vec{v}_i = \delta_{ij}$
- $\mathbf{U}$ is an orthogonal transformation matrix: $\mathbf{U}^T = \mathbf{U}^{-1}$

From this we can show that tansformations with $\mathbf{U}$ preserves orthonormality, i.e. that $\vec{w}_i = \mathbf{U}\vec{v}_i$ will also be an orthonormal set:

$$\vec{w}_i^T \cdot \vec{w}_j = (\mathbf{U}\vec{v}_i)^T(\mathbf{U}\vec{v}_j) = \vec{v}_i^T \mathbf{U}^T \mathbf{U}\vec{v}_j = \vec{v}_i^T \mathbf{U}^{-1}\mathbf{U}\vec{v}_j = \vec{v}_i^T \mathbf{I}\vec{v}_j = \vec{v}_i^T \cdot \vec{v}_j = \delta_{ij} \quad \blacksquare$$

PROBLEM 3 - THE TRIDIAGONAL MATRIX A

See 'triag.cpp' for code and 'test_triag.cpp' for test.

PROBLEM 4 - LARGEST OFF-DIAGONAL ELEMENT

Problem a

See 'max_offdiag_symmetric.cpp'.

Problem b

See 'test_max_offdiag_symmetric.cpp'.

PROBLEM 5 - IMPLEMENTATION OF JACOBI'S ROTATION

Problem a

See 'jacobi_eigensolver.cpp'.

Problem b

See 'test_jacobi_eigensolver.cpp'.

PROBLEM 6 - TRANSFORMATIONS

Problem a

See 'estimate_rotations.cpp' and FIG 1.

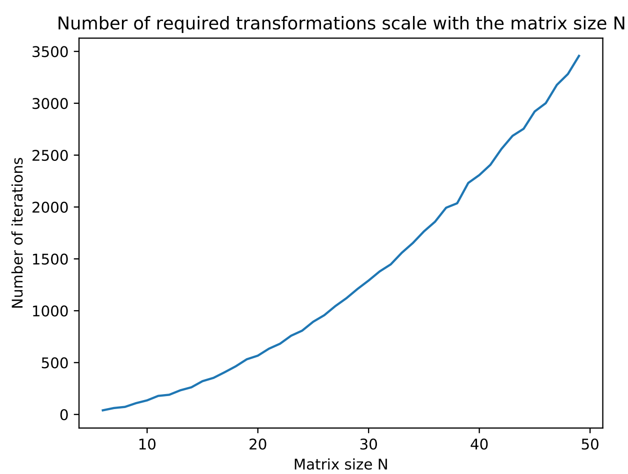

FIG. 1. Number of required transformations scale with the tridiagonal matrix of size N

Problem b

The scaling behavior we expect to see if $\mathbf{A}$ was a dense matrix is that insted of having a $\mathcal{O}(n^2)$ we have $\mathcal{O}(n^3)$.

PROBLEM 7 - EIGENVALUE PROBLEM

Problem a

Problem b