Datorövning 4: Linjära ekvationssystem

För denna uppgift, genomför följande importer: `import numpy as np`, `from scipy import linalg as LA`, `from sympy import Matrix`, `import time`

Vi börjar med att titta på 2 olika sätt att lösa ett linjärt ekvationssystem $A\mathbf{x} = \mathbf{y}$.

- Det första sättet är Gausseliminering. Detta finns implementerat för objekt av klassen `Matrix` och kan beräknas genom att använda `T.rref()` där `T` är totalmatrisen $(A\ \mathbf{y})$.
- Det andra sättet fungerar om A är en inverterbar matris, och går ut på att beräkna inversen till A och multiplicera med den på lämpligt sätt. Inversen kan beräknas med `LA.inv(A)`.

Gå tillbaka till Lab 3 bland introduktionsbladen. I avsnitt 6 finns beskrivet hur man använder `rref()` för objekt av klassen `Matrix`.

(a) Använd `rref()` för att lösa ekvationssystemet $A\mathbf{x} = \mathbf{0}$ där

$$A = \begin{pmatrix} 1 & 2 & 1 & -1 & 2 \\ 3 & 4 & 5 & 2 & 0 \\ 2 & 2 & 1 & 0 & 2 \end{pmatrix}.$$

Paketet `sympy`, som definierar klassen `Matrix`, är avsett för symbolisk matematik och `rref()` är därför inte särskilt snabb.

(b) Bilda en 100×100-matris A och en 100-vektor $\mathbf{y}$ med slumptal. Använd sedan `rref()` för att lösa $A\mathbf{x} = \mathbf{y}$. Hur lång tid tar det? Hur varierar tiden det tar om man använder slumptal från `np.random.rand` jämfört med `np.random.randint`? Tiden kan mätas med

```
t=time.time(); ...[kommandon]...; print(Time taken:", time.time()-t)
```

Skriv även kod som verifierar att svaret du beräknar faktiskt är en lösning, inom rimlig numerisk tolerans.

(c) Gör nu samma som i föregående del men försök lösa $A\mathbf{x} = \mathbf{y}$ genom att först beräkna inversen A^{-1} med med `LA.inv(A)`. Hur lång tid tar det?

Inlämning:
Sista inlämning är den 5 mars klockan 23:59. Lämna in Python-kod som enkelt kan köras i exempelvis Spyder. Inlämningen görs i Canvas. Där finns även instruktioner för hur inlämningen skall göras. Läs dessa noggrannt.