Definir la matriz A y el vector b:
Esto define el sistema de ecuaciones lineales. Aquí, A es la matriz de coeficientes y b es el vector de términos independientes.

```
octave:1> A=[2 3 1; 4 6.0001 2; 3 1 2]
A =

   2.0000   3.0000   1.0000
   4.0000   6.0001   2.0000
   3.0000   1.0000   2.0000

octave:2> b=[1;2;3]
b =

   1
   2
   3
```

Cálculo del determinante:
El determinante det_A se calcula usando la función det(). Si el determinante es cercano a cero, la matriz es casi singular, lo que indica un sistema mal condicionado.

```
octave:3> det(A)
ans = 1.0000e-04
```

Matriz identidad:
La matriz identidad I se usa para entender mejor el comportamiento de la matriz A. Aunque no es usada directamente aquí, en otros casos se compara con la matriz inversa.

```
octave:4> I=eye(3)
I =

Diagonal Matrix

   1   0   0
   0   1   0
   0   0   1
```

Resolución del sistema:
Se resuelve el sistema con x = A\b, que usa la factorización de la matriz para encontrar la solución.

```
octave:12> disp(x)
  -1.0000e+00
  -2.0589e-12
   3.0000e+00
```

Cálculo de la norma:

Se usa la función norm() para calcular la norma de la matriz A y su inversa. Esto es necesario para calcular el número de condición.

```
octave:5> norm_A=norm(A)
norm_A = 8.9034
```

Número de condición:

El número de condición se calcula como cond_A = norm_A * norm_A_inv. Si este número es grande, indica que el sistema es mal condicionado. Los sistemas bien condicionados tienen números de condición cercanos a 1.

```
octave:9> cond_A=norm_A*norm_A_inv
cond_A = 1.7242e+06
```

Comparación de los resultados:

Podemos observar el cambio no tan grande en números enteros pero si en decimales, mas q todo en segunda fila que pasa de 0 a -2.05, en cambio las de mas filas no sufren un gran cambio.

```
octave:12> disp(x)
  -1.0000e+00
  -2.0589e-12
   3.0000e+00
octave:13> X=inv(A)*b
X =

  -1
   0
   3
```

Conclusión:

Lo que observaste está bien. Aunque las diferencias numéricas pueden parecer pequeñas (especialmente en la segunda fila), estas diferencias pueden ser críticas en sistemas mal condicionados. Este ejemplo muestra cómo incluso un pequeño error de redondeo puede llevar a variaciones significativas en la solución cuando se trabaja con sistemas de este tipo.

En resumen, A\b es más confiable que el uso de inv(A)*b para resolver sistemas lineales, especialmente cuando el sistema es mal condicionado.

Grafica

```
[x_vals, y_vals] = meshgrid(linspace(-10, 10, 100), linspace(-10, 10, 100));
% Definir las ecuaciones de los planos
z1 = (b(1) - A(1,1) * x_vals - A(1,2) * y_vals) / A(1,3);
z2 = (b(2) - A(2,1) * x_vals - A(2,2) * y_vals) / A(2,3);
z3 = (b(3) - A(3,1) * x_vals - A(3,2) * y_vals) / A(3,3);
% Crear una figura 3D
figure;
hold on;
```

```
% Graficar los planos
surf(x_vals, y_vals, z1, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'r');
surf(x_vals, y_vals, z2, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'g');
surf(x_vals, y_vals, z3, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'b');

% Etiquetas de los ejes
xlabel('x');
ylabel('y');
zlabel('z');

% Título del gráfico
title('Sistema mal condicionado 3x3 - Planos');

% Configurar el gráfico
grid on;
axis equal;
view(3);
hold off;
```