Felipe Montero Concha
20147611-0

c) Sabemos que Epsilon machine entrega la información de un límite superior para saber que cuando uno realiza $1+\varepsilon_m$ sea mayor a 1, es decir define el numero más pequeño que al sumarlo el computador no redondea.

Epsilon para esta representación de 32 bits es:

$$\varepsilon_{mach\ single\ precision} = 2^{-23}$$

Para saber el k:

$$k \simeq \sqrt{\frac{1}{\varepsilon_{mach}}} \qquad k \simeq 2896,309376$$

d) Al realizar el primer algoritmo se comienza con un numero muy grande en comparación a los ultimos numeros de la serie ($1 \ll k^2$), por lo tanto la precisión tiene una mayor tolerancia desde el comienzo. En cambio en el segundo, hay numeros tan pequeños (en exponente y mantisa), por lo tanto para la representación comienza sumando ceros hasta que 32 bits logra evitar perdida de precisión y dar una suma distinta a cero.

Se puede concluir que para conseguir un error más chico (se de aumentar) que 10^{-6} se de aumentar en 3 ordenes de magnitud el k originalmente buscado junto con encontrar que al sumar numeros de mayor magnitud se mantiene precisión sumando numeros más pequeños, precisión que se pierde en el primero.