Com compilar i executar:

Els programes han estat fets des de la terminal de Linux mitjançant l'editor de textos vim.

Com he programat la funció $f(x,y) = \frac{y^2}{sin^2(x)}$ en un programa de nom **f.c** fora dels programes principals, l'he compilat per separat:

→ gcc -c -ansi -Wall f.c

Posteriorment he compilat els 5 programes dels apartats **(a), (b), (c), (d), (e)** amb la funció f.c, que la he definit dins els 5 programes. Per a compilar els 5 mètodes numèrics demanats, anomenaré programa.c com a terme genèric a l'arxiu que generaliza els programes ***euler_method.c, euler_method_with_richardson.c, adams_bashfort.c, runge_kutta.c, runge_kutta_method_with_richardson.c.***

→ gcc -c -ansi -Wall programa.c
→ gcc programa.o f.o -o programa.exe -lm
→ ./programa.exe

Finalment compilo el programa de l'apartat **(f)** de nom ***gnuplot.c***.

→ gcc -c -ansi -Wall gnuplot.c
→ gcc gnuplot.o -o gnuplot.exe -lm
→ ./gnuplot.exe

Observacions:

- Les dades que setejem segons l'enunciat no les demano per pantalla ja que no varien en el transcurs de la pràctica.

- Entenc que per als programes on s'aplica extrapolació de Richardson és amb m = 1 passos, és a dir, que per al punt x, calculo dues aproximacions de y pel mètode en qüestió pels valors de h = h0 i h1 = h0/2 fins a x, i després les combino amb l'algorisme (1) de l'enunciat per obtenir un pas d'extrapolació de Richardson.

- Tots els els he executat posteriorment amb el comandament valgrind ./programa.exe.
  Tots m'han donat el missatge:
  All heap blocks were freed -- no leaks are possible
  ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
  Interpreto que no hi ha memòria que no s'alliberi correctament.

Resposta apartat (f):

El que he fet ha sigut fer un programa que es diu ***gnuplot.c***. S'ha de compilar i executar el programa com s'ha indicat anteriorment. Un cop executat, es demana donar el nom d'un fitxer on imprimir les dades, el que jo he fet es dir-li a aquest fitxer ***gnuplot.out*** que és el fitxer d'on després obtindré les dades per a la gràfica en el gnuplot.

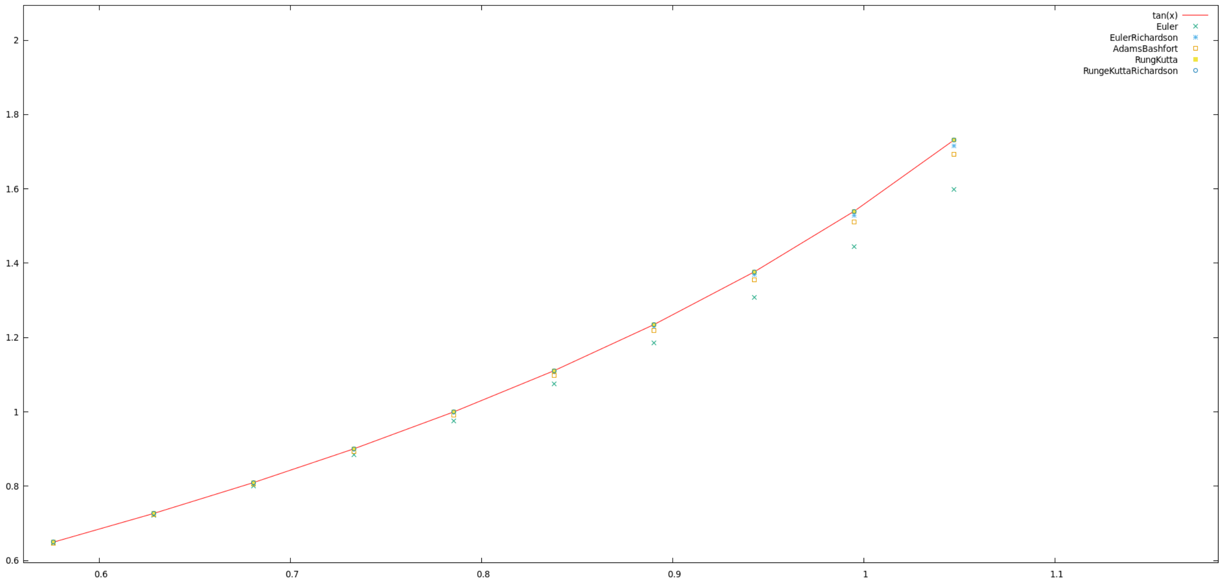
En el programa ***gnuplot.c*** un cop llegit el fitxer de sortida, s'obren els 5 fitxers on hem imprès les dades dels 5 mètodes numèrics dels apartats **(a), (b), (c), (d), (e).** Posteriorment llegint d'aquests 5 fitxers es va imprimint iterativament les y de cada mètode numèric (columnes 3 a 7). La columna 1 correspon a les x que corresponen a cada aproximació de y i la columna 2 correspon al valor real de la funció solució en aquesta x, que sabem que és tan(x).

<u>Com obtenir el gràfic?</u>

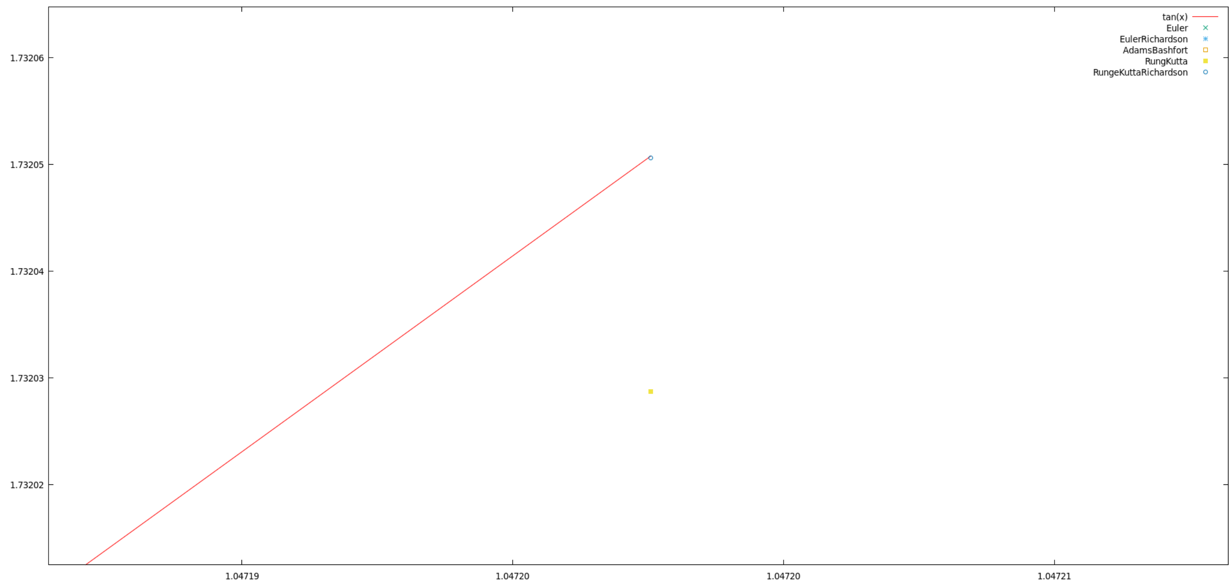
Obrirem el gnuplot:
→ gnuplot

Escriurem el següent:
→ plot "gnuplot.out" u 1:2 title 'tan(x)' w l lt rgb "red", "gnuplot.out" u 1:3 title 'Euler' w points, "gnuplot.out" u 1:4 title 'EulerRichardson' w points, "gnuplot.out" u 1:5 title 'AdamsBashfort' w points, "gnuplot.out" u 1:6 title 'RungeKutta' w points, "gnuplot.out" u 1:7 title 'RungeKuttaRichardson' w points

<u>Explicació d'aquestes últimes línies de codi</u>:
Si el fitxer té més de dues columnes (el nostre cas) indicarem quines variables volem usar, i ho indicarem amb using(u) xcol:ycol. Després s'afageix with(w) lines(l) per a la gràfica de tan(x) i with(w) points per a les altres gràfiques. Cada mètode en el gràfic s'indica correctament a la llegenda amb title 'mètode' i el color de la gràfica de tan(x) s'ha posat de color vermell amb linetype(lt) "red".



S'observa que els mètodes numèrics de runge_kutta i runge_kutta_with_richardson és solapen (són molt properes les aproximacions de y). He fet el gràfic amb escala logarítmica però no m'ha convençut el resultat, així que adjunto un zoom del gràfic en el punt $x = \frac{\pi}{3}$ perquè es vegi que realment runge_kutta_ with_richardson és més precís que aquest mateix mètode sense un pas d'extrapolació de Richardson.



Quin és millor mètode?

El millor mètode és el mètode de Runge Kutta amb un pas d'extrapolació de Richardson. Això és degut a que el mètode de Runge Kutta té un error de truncament d'ordre h^4, major que euler amb i sense extrapolació i adams_bashfort. A l'aplicar un pas d'extrapolació al mètode de Runge Kutta l'error de truncament encara disminueix més, a l'arxiu gnuplot.out podem veure que tan(x) i Runge Kutta amb un pas d'extrapolació coincideixen fins en 6 decimals, en canvi Runge Kutta només fins a 4 decimals (h^4).

Quin és el pitjor mètode?

El pitjor mètode és clarament el mètode d'Euler sense extrapolació. L'error és d'ordre h i a l'aplicar un pas d'extrapolació seguim estant lluny de Runge Kutta ja que l'error de truncament passa a ser d'ordre h^2i no pash^4.