1. Порівняйте результат розрахунку інтегралу trapezoidal.m (не векторизована функція з л10) та trapezoidal\_vec.m (векторизована) з лекції 11 для n = [10 1e2 1e3 1e4 1e5 1e6 1e7] або враховуючи потужніть Вашого комп’ютера.

Функція для інтегрування: 

Функція sinc визначається як **sin(x)/x**, в точці 0 дорівнює 1-ці.

Анонімна функція: v = @(t) sinc(t);

Використайте пару tic toc для розрахунку витраченого часу для кожного n. Побудуйте графіки залежності витраченого часу від кількості n. Можливо знадобиться логарифмічний масштаб. Функція для побудови в цьому випадку: loglog().

Розрахуйте у скільки разів векторизований код швидше порахував результат для кожного n та побудуйте графік залежності від кількості елементів в масиві n. Наприклад plot(percetage), а масив percentage містить виграш у разах для кожного n.

1. Використайте вбудовані функції для чисельного інтегрування та перевірте результат з п.1

trapz() – Метод трапецій

interal() – адаптивний метод квадратур

Quadv() – метод Сімпсона

Quad() – метод квадратури Гауса

1. Проінтегруйте наступний подвійний інтеграл за допомогою векторизованої функції midpoint\_double1.m:



*function result = midpoint\_double1(f, a, b, c, d, nx, ny)*

*hx = (b - a)/nx;*

*hy = (d - c)/ny;*

*I = 0;*

*for i = 0:(nx-1)*

*for j = 0:(ny-1)*

*xi = a + hx/2 + i\*hx;*

*yj = c + hy/2 + j\*hy;*

*I = I + hx\*hy\*f(xi, yj);*

*end*

*end*

*result = I;*

*end*

Побудуйте графік функцій:





В матлабі це можна зробити за допомогою символьних розрахунків та функції

*fmesh(@(x,y) вираз для побудови)*

В Octave:

*x = linspace(-2\*pi, 2\*pi,100)*

*y = linspace(-2\*pi, 2\*pi,100)'*

*mesh(x,y, вираз для побудови )*