Додаток 1. (Код програми)

Підпрограма для створення початкового альтернанcу (дискретний випадок).  
Вхідні параметри:  
1) масив значень x  
2) degree – степінь апроксимуючого многочлена  
Результат виконання функції: початковий альтернанс

**def** make\_initial\_alternance(x, degree):

alternance = [x[0]] *# add first*

busy = []

for i in range(degree):

rand\_index = np.random.randint(1, len(x)-1)

while rand\_index in busy:

rand\_index = np.random.randint(1, len(x)-1)

busy.append(rand\_index)

alternance.append(x[rand\_index])

alternance.append(x[len(x)-1]) *# add last*

alternance.sort()

return alternance

В цій функції використана бібліотека *numpy(*[*http://www.numpy.org/*](http://www.numpy.org/)*).*

Підпрограма для визначення максимального значення функції в дискретних точках.  
Вхідні параметри:  
1) func – функція  
2) x\_vals – масив дискретних точок

**def** max\_error(func, x\_vals):

y\_vals = func(x\_vals)

neg\_err = min(y\_vals)

pos\_err = max(y\_vals)

if abs(neg\_err) > pos\_err:

e\_max = neg\_err

else:

e\_max = pos\_err

return e\_max

Результат виконання функції: максимальне значення заданої функції

Підпрограма для заміни точок альтернансу.  
Вхідні параметри:  
1) err\_func – функція похибки  
2) alternance – попередній альтернанс  
3) x\_vals – значення x для таблично заданої функції

Результат виконання функції: новий альтернанс

**def** change\_alternance(err\_func, alternance, x\_vals):

x\_err = x\_of\_max\_error(err\_func, x\_vals)

temp = alternance[:]

temp.append(x\_err)

temp.sort()

index\_of\_x\_err = temp.index(x\_err)

if index\_of\_x\_err != 0 and index\_of\_x\_err != (len(temp)-1):

if sign(err\_func(temp[index\_of\_x\_err])) == sign(err\_func(temp[index\_of\_x\_err-1])):

del temp[index\_of\_x\_err-1]

else: del temp[index\_of\_x\_err+1]

elif index\_of\_x\_err == 0:

if sign(err\_func(temp[index\_of\_x\_err])) == sign(err\_func(temp[1])):

del temp[1]

else:

del temp[len(temp)-1]

elif index\_of\_x\_err == (len(temp)-1):

if sign(err\_func(temp[index\_of\_x\_err])) == sign(err\_func(temp[index\_of\_x\_err-1])):

del temp[index\_of\_x\_err-1]

else:

del temp[0]

return temp

Підпрограма для побудови аналітичного вигляду многочлена.

Вхідні параметри:  
1) t – альтернанс  
2) degree – степінь апроксимуючого многочлена  
3) f\_discrete – дискретна функція

**def** pol(t, degree, f\_discrete):

x = Symbol('x')

e = Symbol('e')

vars\_str = ' '.join(['a' + str(i) for i in range(degree+1)])

variables = symbols(vars\_str)

eqs = []

for i in range(degree+2):

eqs.append(make\_eq(variables, t[i], f\_discrete(t[i])) + e)

e \*= -1

if degree % 2 == 1:

e \*= -1

solution = solve(eqs, variables + (e,))

error\_on\_iteration = solution[e]

polynom = x - x

for i, v in enumerate(variables):

polynom += solution[v] \* x\*\*i

return [polynom, error\_on\_iteration]

Результат виконання функції: аналітичний вигляд многочлена.