FACULTAD DE INGENIERIA
Universidad de Buenos Aires

“ANÁLISIS NUMÉRICO I”

“MÉTODOS MATEMÁTICOS Y NUMÉRICOS”

<75.12>

DATOS DEL TRABAJO PRÁCTICO

1	2023	Resolución de sistema de valores de contorno
	AÑO	utilizando el método SOR
	2	
TP NRO	CUAT	TEMA

INTEGRANTES DEL GRUPO

Integrantes: 1
Apellido, Nombre (1): Palavecino Arnold, Nestor Fabian
Padrón: 108244

INTRODUCCIÓN

Este trabajo práctico consiste en la obtención de una solución aproximada al problema físico de conocer la temperatura en cada punto interno de una placa, dadas las temperaturas en sus 4 contornos externos.

La solución ideal de este problema se obtiene con la ecuación de Laplace:

$$\nabla^2 T = 0$$

Pero para eso necesitaríamos calcular infinitos puntos, lo cual no es posible en una computadora. Con lo cual, para resolver este problema, planteamos la resolución de un sistema de ecuaciones lineales utilizando métodos numéricos, en concreto, con el método de sobrerrelajaciones sucesivas, o ***SOR*** *(Successive over-relaxation)*.

A continuación veremos el planteo inicial del problema, cómo se arma el sistema de ecuaciones lineales, cómo se plantea el algoritmo de resolución del sistema, el código de dicho algoritmo, el análisis de los resultados y sus conclusiones.

OBJETIVOS

Los objetivos principales de este trabajo son:

1. Obtener una solución numérica al problema diferencial
2. Que dicha solución pueda ser resuelta en la menor cantidad de pasos posibles
3. Observar qué pasa con la velocidad de convergencia del problema
4. Observar qué sucede si aumentamos la discretización del problema
5. Resolver un caso práctico al problema de la placa

DESARROLLO

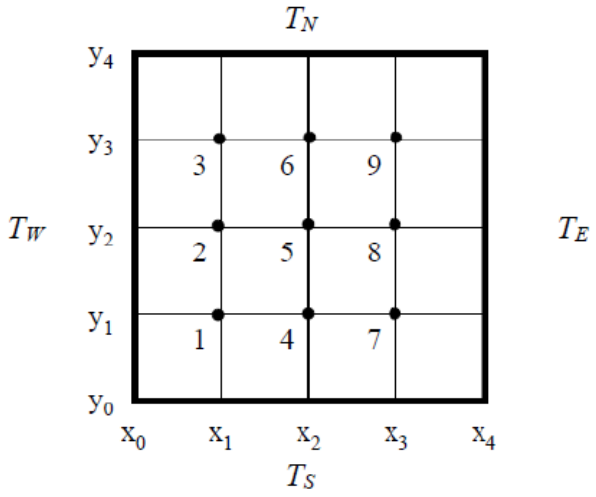
PARTE 1

A)

La matriz de temperaturas y su sistema de ecuaciones

Para resolver el problema de la placa, la dividimos en N regiones, formando una grilla:

(Figura A)



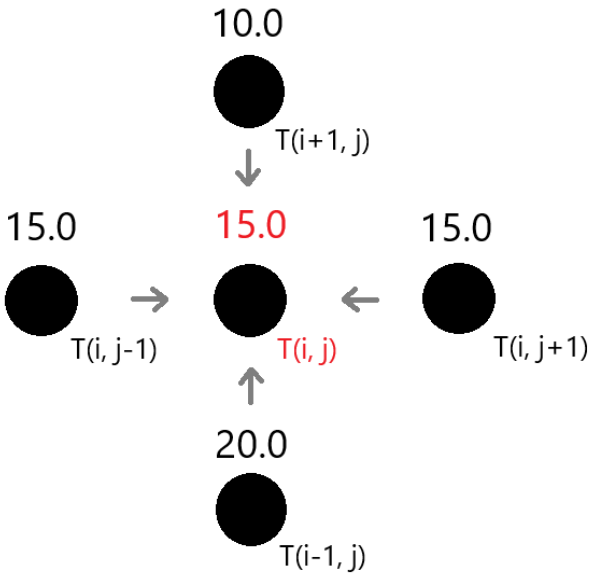
En este caso: $\mathrm{N} = 4$. Observar que donde se cruzan las líneas divisorias se forman unos puntos. A partir de ahora los llamaremos **nodos**. En cada uno de esos nodos es donde mediremos la temperatura interna de la placa. La temperatura en cada nodo puede ser diferente.

La solución aproximada T de la ecuación de Laplace $\nabla^2 T = 0$ en un dominio bidimensional cuadrado discretizado usando una grilla uniforme $(x_i = x_0 + ih;\ y_j = y_0 + jh)$ puede obtenerse resolviendo el sistema de ecuaciones lineales que surge de aplicar el siguiente operador a cada uno de los nodos de la grilla:

$$4\,T_{i\,j} \;-\; T_{i-1\,j} \;-\; T_{i+1\,j} \;-\; T_{i\,j-1} \;-\; T_{i\,j+1} \;=\; 0 \qquad (1)$$

Esta ecuación se traduce como: "el nodo $T_{i\,j}$ tiene que ser igual al promedio de los nodos adyacentes de arriba, abajo, de su izquierda y de su derecha". Dicha operación la podemos visualizar en la **figura B**:

(Figura B)



Aplicando el operador **(1)** a cada uno de los $(N-1)^2$ nodos, en el orden numérico indicado en la **figura A**, y pasando a la derecha del "**=**" los valores correspondientes a los nodos T_{ij} que pertenecen a los puntos contorno de la placa, nos queda un sistema de $(N-1)^2$ ecuaciones con $(N-1)^2$ incógnitas.

Por ejemplo, para $N = 4$, nos queda un sistema de 9 ecuaciones con 9 incógnitas.

Este sistema lo podemos representar como $Ax = b$, donde A es la matriz de coeficientes de los nodos, x es el vector temperatura y b son los valores de temperatura de contorno para cada nodo interno.

Para $N = 4$, las matrices A y b nos quedan de la siguiente manera:

(Figura C)

$$A = \begin{array}{c} \begin{matrix} T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 & T9 \end{matrix} \\ \left(\begin{array}{ccc|ccc|ccc} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{array}\right) \end{array} \qquad b = \begin{pmatrix} TS + TW \\ TW \\ TN + TW \\ TS \\ 0 \\ TN \\ TS + TE \\ TE \\ TN + TE \end{pmatrix}$$

Para $N = 3$, las matrices A y b nos quedan:

(Figura D)

$$A = \begin{array}{c} \begin{matrix} T1 & T2 & T3 & T4 \end{matrix} \\ \left(\begin{array}{cc|cc} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ \hline -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{array}\right) \end{array} \qquad b = \begin{pmatrix} TS + TW \\ TN + TW \\ TS + TE \\ TN + TE \end{pmatrix}$$

Para $N = 2$, las matrices A y b nos quedan:

(Figura E)

$$A = \begin{array}{c} \begin{matrix} T1 \end{matrix} \\ \begin{pmatrix} 4 \end{pmatrix} \end{array} \qquad b = \begin{pmatrix} TN + TW + TS + TE \end{pmatrix}$$

Y, en general, para un N genérico, la matriz A tiene la forma:

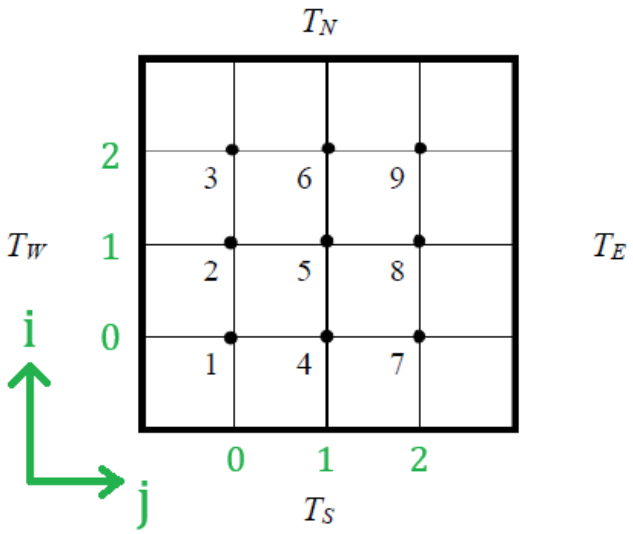
(Figura F)

$$A = \begin{pmatrix} T & -I & 0 & 0 & 0 & 0 & 0 \\ -I & T & -I & 0 & 0 & 0 & 0 \\ 0 & -I & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T & -I & 0 \\ 0 & 0 & 0 & 0 & -I & T & -I \\ 0 & 0 & 0 & 0 & 0 & -I & T \end{pmatrix}$$

$$T = \begin{pmatrix} 4 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 4 \end{pmatrix} \qquad -I = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Y, para determinar el b, utilizamos la siguiente ecuación condicional, recorriendo los nodos de la siguiente forma:

(Figura G)



 # Si hay un solo nodo
si N = 2:
  b = T_N + T_S + T_W + T_E

 # Si el nodo está en una de las 4 esquinas
si (i = 0) y (j = 0):
  b = T_S + T_W

si $(i = 0)$ y $(j = N - 2)$:
  $b = T_S + T_E$
si $(i = N - 2)$ y $(j = 0)$:
  $b = T_N + T_W$
si $(i = N - 2)$ y $(j = N - 2)$:
  $b = T_N + T_E$

 # Si el nodo está en uno de los 4 bordes (pero no es una esquina)
si $(i = 0)$ y $(1 \leq j \leq N - 3)$:
  $b = T_S$
si $(i = N - 2)$ y $(1 \leq j \leq N - 3)$:
  $b = T_N$
si $(j = 0)$ y $(1 \leq i \leq N - 3)$:
  $b = T_W$
si $(j = N - 2)$ y $(1 \leq i \leq N - 3)$:
  $b = T_E$

 # Si el nodo está en el centro
sino: $b = 0$

Un tema importante a tener en cuenta, es que la estructura que adquiere la matriz A depende de la forma de numerar los nodos, ya que si a los nodos los numeramos, por ejemplo, al revés, las filas de la matriz A van a quedar de abajo hacia arriba, en vez de arriba hacia abajo.

El algoritmo

Para resolver este problema, **no es necesario calcular la matriz A**, pero sí la matriz b. Al no calcular la matriz A, estamos ahorrando mucha memoria, y a la vez, acelerando los tiempos de cálculo de nuestros resultados.

Para obtener los índices (i, j) de los nodos dado su número z (y viceversa), en ambos casos sabiendo el N, tenemos las funciones **`matrix_index_from(z, N)`** y **`z_index_from(i, j, N)`** respectivamente.

Entonces, para calcular $T \pm \Delta T$, lo que hacemos es: mientras el residuo sea menor a R_{tol}, es decir, en la iteración $k+1$, recorremos cada uno de los $z = (N-1)^2$ nodos, y para cada uno calculamos su nuevo valor utilizando el método SOR a través de la ecuación:

(Figura H)

$$T_{z,\,k+1} = \left(\frac{contorno(z,\ N) \ +\ adyacentes(z,\ N,\ T_{k+1},\ T_k)}{4} - T_{z,\,k} \right) \cdot w + T_{z,\,k}$$

Donde:

- N es el tamaño de la grilla de temperaturas,
- z es el número de nodo actual,
- $T_{z,\,k+1}$ es el nuevo valor de temperatura del nodo z,
- $T_{z,\,k}$ es el valor de temperatura del nodo z de la iteración anterior,
- T_{k+1} es el vector de temperaturas con los nuevos valores acumulados hasta el momento,
- T_k es el vector de temperaturas completo, calculado en la iteración anterior
- $\text{contorno}(z, N)$ es la suma de los valores de temperatura de contorno adyacentes al nodo z,
- $\text{adyacentes}(z, N, T_{k+1,} T_k)$ es la suma de los valores de temperatura de los nodos internos adyacentes al nodo z. Suma las temperaturas de T_{k+1} (es decir, toma los cálculos de la iteración actual) cuyos números de nodo sean menores a z, y las temperaturas de T_k (es decir, toma los cálculos de la iteración anterior) cuyos números de nodo sean iguales o mayores a z.

B)

Código en en Anexo I

C)

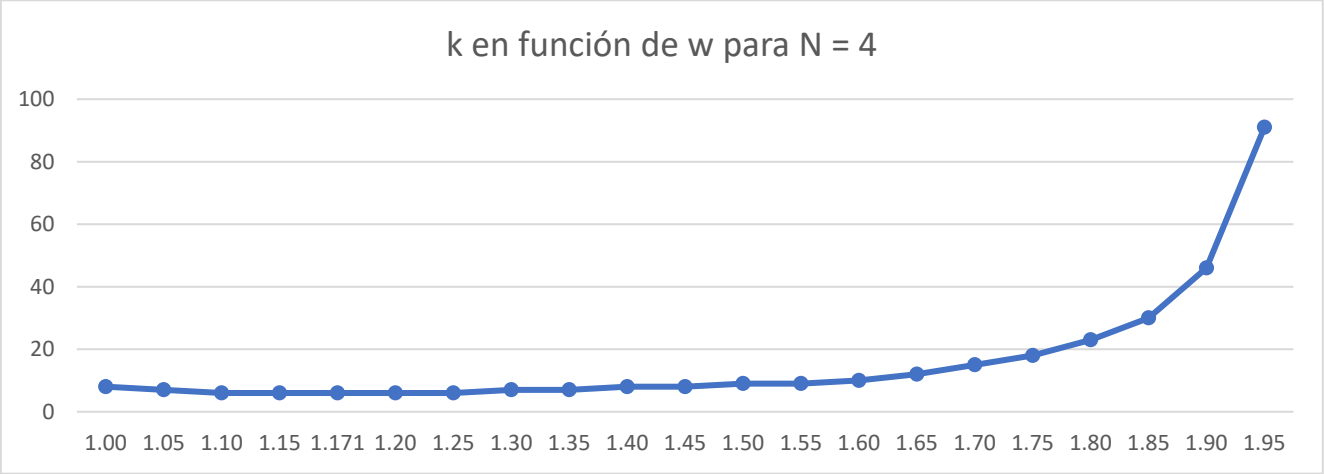
Corrida del código en el anexo II

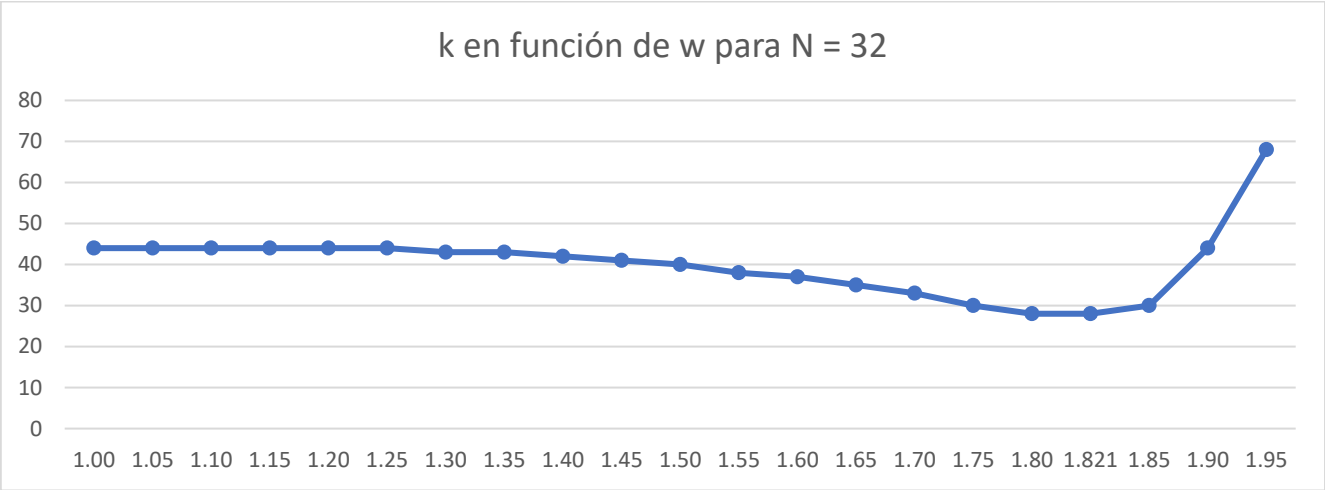
D)

Corrida del código en el anexo II

E)

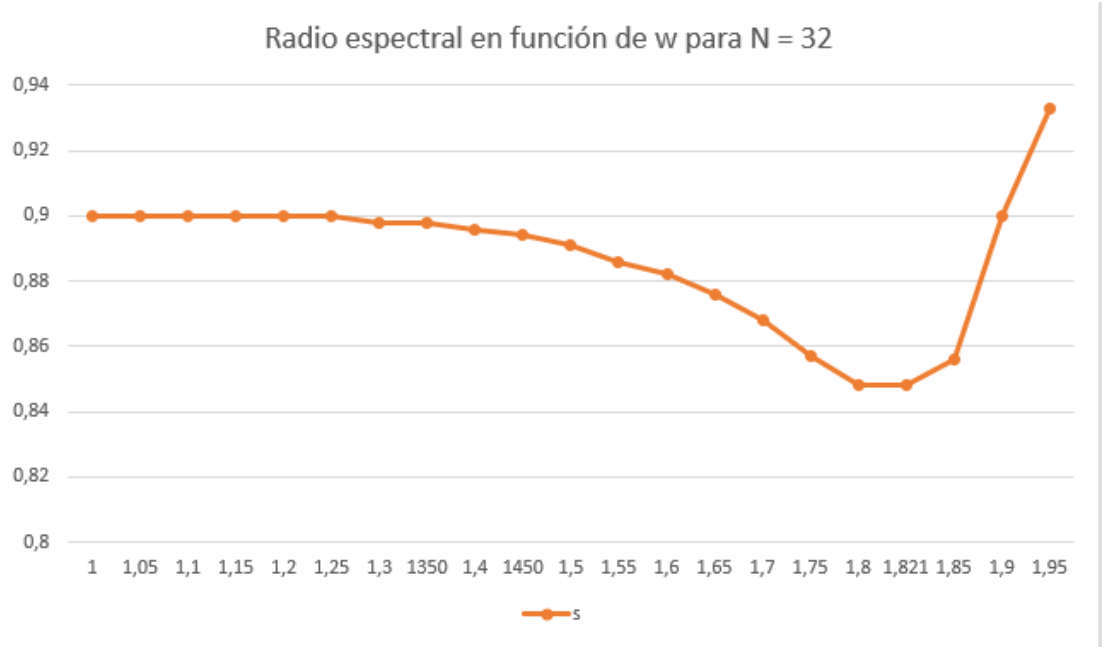
Gráfico de k en función de w:





F)

Vamos a comprobar si los datos obtenidos experimentalmente se corresponden con los valores teóricos de $\rho(T_{GS})$, $w_{óptimo}$ y $\rho(T_{SOR})$.



Como podemos observar en los gráficos, el radio espectral de $\mathrm{T}_{\mathrm{SOR}}$ más bajo siempre corresponde al valor de w más bajo, es decir, el óptimo. Esto significa que el valor de w óptimo calculado es correcto y corresponde, efectivamente, al coeficiente que proporciona la convergencia del algoritmo SOR más rápida.

También podemos observar que en todos los casos, el radio espectral se mantiene menor a 1, lo cual significa que para todos los valores de w, tanto para $\mathrm{N}=4$ como para $\mathrm{N}=32$, el método converge.

PARTE 2

Ahora resolveremos un caso práctico del problema de la placa para un *motherboard* de computadora. Tomaremos $N=8$, $R_{TOL}=0.001$ y el $w=w_{óptimo}$ para efectuar estos cálculos. Los valores de contorno son: $\mathrm{T_N} = 80$, $\mathrm{T_S} = 20$, $\mathrm{T_W} = 40$ y $\mathrm{T_E} = 40$.

El $w_{óptimo}$ se calcula utilizando la función **`best_w_value(N)`** de nuestro programa.

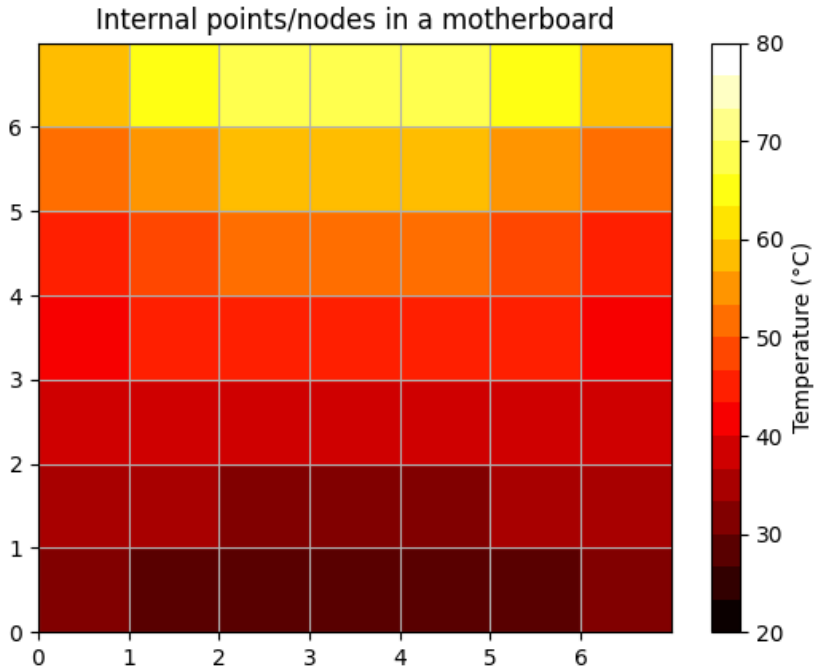
Los resultados son los siguientes:

```
TEST_1_X_VALUE = [
    30.945026743222666,
    36.044337387419354,
    39.24441237284024,
    42.0167272365443,
    45.230185361473396,
    50.005517372274646,
    58.949288422386395,
    27.99029893147639,
    34.05827561765416,
    38.968029343237696,
    43.628778365069536,
    48.913904625073414,
```

```
    55.846502298637056,
    65.79509260589951,
    26.99359153480291,
    33.267995531762885,
    38.96381549934404,
    44.62925057460862,
    50.95477370989943,
    58.6735712611409,
    68.3853512715077,
    26.76710971087393,
    33.08143278955388,
    39.00211866300342,
    44.97135760463382,
    51.60089909447619,
    59.507060059047106,
    69.07077823930118,
    27.026166395392426,
    33.30450610957336,
    38.99468077569874,
    44.650703956119656,
    50.96790450347006,
    58.68107236933291,
    68.38850075399537,
    28.04591735524012,
    34.12118125482002,
    39.02026725770265,
    43.666069422178936,
    48.93677607654262,
    55.85909299761288,
    65.80070720519885,
    31.03973564334656,
    36.11472402104687,
    39.29781818486474,
    42.054542436334,
    45.25235436515735,
    50.0166770637502,
    58.95447072072414
]

TEST_1_X_ERROR_BOUND = [
    0.17043109387879696,
    0.007561820203498826,
    0.016076335241685058,
    0.011290638121515428,
    0.009122624519868339,
    0.003974181510741914,
    0.00048176434645341715,
    0.03654765001853022,
    0.06362492258416097,
    0.06000221390196003,
    0.041284324629287994,
    0.02269670100089627,
    0.009295013923455997,
    0.0028614864030629406,
    0.017196186919139933,
    0.05610994702909977,
    0.0559608788008 2479,
    0.04107261333954426,
    0.02334831997931275,
    0.010691115619863467,
    0.004304454862449347,
    0.016159070043133994,
    0.03844763532298856,
```

```
    0.04015065714325061,
    0.029084169914170843,
    0.01666620958090448,
    0.008297445262563485,
    0.0033514429132139867,
    0.014254764240906326,
    0.023316924397377647,
    0.023299179014536264,
    0.016783939979539753,
    0.009778708471017694,
    0.004973586008816255,
    0.0019369298170914817,
    0.007150073987617134,
    0.010925445630697084,
    0.010854882060492343,
    0.00813978147103711,
    0.004933014080776843,
    0.0024724610643218625,
    0.000946195584333509,
    0.0013161452325611833,
    0.0033237594156929617,
    0.003942622352155922,
    0.0031286678203841234,
    0.001934983825329084,
    0.0009671111443836367,
    0.0004038559579129241 7
]

TEST_1_RESIDUAL = 0.0007298476528922345

TEST_1_ITERATIONS = 14
```

Con estos datos, procedemos a graficar las regiones de temperatura en forma bidimensional:



CONCLUSIONES

Como pudimos observar en este trabajo práctico, es posible resolver el problema diferencial aplicando métodos numéricos. Sin embargo, la solución no deja de ser aproximada, puesto que al limitar la cantidad de iteraciones del algoritmo, estamos inevitablemente introduciendo un error de truncamiento.

Dicho error de truncamiento también lo estimamos con el algoritmo y lo devolvemos al usuario. Este error va a ser menor cuanto mayor sea la cantidad de iteraciones. Esta cantidad de iteraciones la podemos aumentar disminuyendo el valor del residuo de tolerancia $\mathrm{R_{tol}}$.

Ahora bien, si queremos reducir la cantidad de iteraciones, tenemos varias formas. La primera es aumentar la tolerancia, para un $\mathbf{W}$ constante, en el caso de que ya tengamos el $\mathbf{W}$ óptimo. Si no tenemos el $\mathbf{W}$ óptimo, entonces acercarnos a este valor también reduce la cantidad de iteraciones para resolver el problema, ya que aumentamos la velocidad de convergencia del algoritmo.

Otra forma de reducir la cantidad de iteraciones es disminuir el valor de N, pero el problema que conlleva eso es que estamos empeorando la discretización del problema, ya que al tomar menos puntos, estamos resolviendo el problema con menos definición, lo cual puede ser útil o no dependiendo del problema que se quiera resolver, además del tiempo y los recursos computacionales que se tengan para llevar a cabo el procesamiento de datos.

Por último, utilizamos todos estos parámetros más los datos de entrada para resolver el problema de la placa para un *motherboard*, y graficamos las regiones de la temperatura calculada en cada uno de los nodos internos de la misma.

ANEXO I

Código de la solución

(Utilizando Python 3.11 junto con la librería `numpy`)

```python
from numpy import cos, power, sin, subtract, array, pi
from numpy.linalg import norm


PI = pi

# Boundaries list
UPPER = 0  # TN
LOWER = 1  # TS
LEFT = 2  # TW
RIGHT = 3  # TE
```

```python
# Results tuple
X_VALUE = 0
X_ERROR_BOUND = 1
RESIDUAL = 2
ITERATIONS = 3


# Punto de entrada de la aplicación
def solve_discrete_laplace_sor(
        n: int,
        r_tol: float,
        boundaries: list[int],  # Contorno: [UPPER, LOWER, LEFT, RIGHT]
        w: float = -1,
        seed: list[float] = None  # X(0)
) -> tuple[list[float], list[float], float, int]:
    # X(k + 1)
    x1 = list()
    # X(k)
    x0 = seed if (seed is not None) else initial_seed(n)
    if w == -1:
        w = best_w_value(n)
    delta_x = []
    k = 1  # iterations
    r = r_tol
    max_z = power(n - 1, 2)  # amount of internal nodes
    while r >= r_tol:
        for z in range(1, max_z + 1):  # internal nodes
            new_z_node_value = node_sor(z, n, x1, x0, w, boundaries)
            x1.append(new_z_node_value)
        r = residual(x1, x0)
        if r >= r_tol:
            k += 1
            x0 = x1.copy()
            x1 = []
        else:
            delta_x = error_bound(x1, x0)
    return x1, delta_x, r, k


def initial_seed(n: int) -> list[int]:
    return [0 for i in range(power(n - 1, 2))]


def node_sor(
        z: int,
        n: int,
        x1: list[float],
        x0: list[float],
        w: float,
        boundaries: list[float]
) -> float:
    b = boundary_values_sum(z, n, boundaries)  # Temperaturas de los puntos del
contorno
    i = internal_values_sum(z, n, x1, x0)  # Temperaturas de los nodos internos
    adjacent_sum = b + i   # Suma de todas las temperaturas adyacentes
    z0_value = x0[z - 1]  # T(z, k)
    z1_value = (adjacent_sum / 4 - z0_value) * w + z0_value  # SOR
```

```
    return z1_value  # T(z, k+1)


def residual(x1: list[float], x0: list[float]) -> float:
    arr1 = array(x1)
    arr0 = array(x0)
    return norm(subtract(arr1, arr0), 2) / norm(arr1, 2)


def error_bound(x1: list[float], x0: list[float]) -> list[float]:
    arr1 = array(x1)
    arr0 = array(x0)
    return list(abs(subtract(arr1, arr0)))


def boundary_values_sum(
        z: int,
        n: int,
        boundaries: list[float]
) -> float:
    i, j = matrix_index_from(z, n)  # No generamos la matriz A, calculamos sus
posiciones
    return b_value_from_matrix_index(i, j, n, boundaries)


def b_value_from_matrix_index(
        i: int,
        j: int,
        n: int,
        boundaries: list[float]
) -> float:
    TN = boundaries[UPPER]
    TS = boundaries[LOWER]
    TW = boundaries[LEFT]
    TE = boundaries[RIGHT]

    # only one node
    if n == 2:
        return TN + TS + TW + TE

    # node is corner
    elif (i == 0) and (j == 0):
        return TS + TW
    elif (i == 0) and (j == n - 2):
        return TS + TE
    elif (i == n - 2) and (j == 0):
        return TN + TW
    elif (i == n - 2) and (j == n - 2):
        return TN + TE

    # node is border
    elif (i == 0) and (1 <= j <= n - 3):
        return TS
    elif (i == n - 2) and (1 <= j <= n - 3):
        return TN
    elif (j == 0) and (1 <= i <= n - 3):
        return TW
```

```
    elif (j == n - 2) and (1 <= i <= n - 3):
        return TE

    # node is center
    return 0.0


def internal_values_sum(
        z: int,
        n: int,
        x1: list[float],
        x0: list[float]
) -> float:
    total = 0.0
    for adj in internal_adjacents(z, n):
        if adj < z:
            total += x1[adj - 1]  # current iteration
        else:
            total += x0[adj - 1]  # last iteration
    return total


def internal_adjacents(z: int, n: int) -> list[int]:
    adj = []
    i, j = matrix_index_from(z, n)

    if j - 1 >= 0:
        z_adj = z_index_from(i, j - 1, n)
        adj.append(z_adj)

    if i - 1 >= 0:
        z_adj = z_index_from(i - 1, j, n)
        adj.append(z_adj)

    if i + 1 <= n - 2:
        z_adj = z_index_from(i + 1, j, n)
        adj.append(z_adj)

    if j + 1 <= n - 2:
        z_adj = z_index_from(i, j + 1, n)
        adj.append(z_adj)

    return adj


def matrix_index_from(z: int, n: int) -> tuple[int, int]:
    i = (z - 1) // (n - 1)
    j = z - (n - 1) * i - 1
    return j, i


def z_index_from(i: int, j: int, n: int) -> int:
    return (n - 1) * j + i + 1


def best_w_value(n: int) -> float:
    return 2 / (1 + sin(PI / n))
```

```
def spectral_radius_gs(n: int) -> float:
    return power(cos(PI / n), 2)


def spectral_radius_sor(n: int) -> float:
    return best_w_value(n) - 1


def spectral_radius_gs_from_residual(r: float, k: int) -> float:
    return r ** (1 / k)
```

Un ejemplo de como utilizar el programa:

```
from main import \
    solve_discrete_laplace_sor, \
    UPPER, LOWER, LEFT, RIGHT, \
    X_VALUE, X_ERROR_BOUND, RESIDUAL, ITERATIONS

N = 32
r_tol = 0.01  # tolerancia

# Contorno de la placa
boundaries = [-1, -1, -1, -1]
boundaries[UPPER] = 1.0  # TN
boundaries[LOWER] = 1.0  # TS
boundaries[LEFT] = 1.0  # TW
boundaries[RIGHT] = 1.0  # TE

# Si no se especifica "w", por defecto se calcula el mejor valor
# Si no se especifica "seed", por defecto se utiliza x0 = [0,0,...,0]
data = solve_discrete_laplace_sor(N, r_tol, boundaries)

T = data[X_VALUE]  # temperaturas en los nodos
DELTA_T = data[X_ERROR_BOUND]  # cotas de error
residual = data[RESIDUAL]  # residuo
iterations = data[ITERATIONS]  # cantidad de iteraciones

print(f"Los valores de temperatura de los nodos son: T =", T, "\n")
print(f"Las cotas de error para T son: △T =", DELTA_T, "\n")
print(f"El residuo final es: R =", residual, "\n")
print(f"La cantidad de iteraciones fue: K =", iterations, "\n")
```

ANEXO II

C)

Corridas del programa mostrando los resultados numéricos calculados con Python, para R_{tol} = 0.01, N = 4 y w variable (1.00, 1.05, 1.10, ..., 1.95)

(La precisión es de 24 cifras para la mantisa, es decir la precisión por defecto de `float` en Python)

Formato del resultado para cada valor de w:

```
valor_de_w: {
      "vector_x": list[float],
      "cota_de_x": list[float],
      "residuo": float,
      "valor de n (o, en este TP, k)": int
}
```

(Se incluye adicionalmente el valor $w = 1.17157287525381$, que es el óptimo para $N = 4$)


```
RESULTS_1C_FOR_W_VALUE = {
    1.00: {
        "x_value": [0.9926770925521851, 0.9926761090755463, 0.996337890625, 0.9926761090755463,
0.99267578125, 0.9963378496468067, 0.996337890625, 0.9963378496468067, 0.9981689248234034],
        "x_error_bound": [0.00731503963470459, 0.0073219239711761475, 0.003662109375,
0.0073219239711761475, 0.00732421875, 0.0036623962223529816, 0.003662109375,
0.0036623962223529816, 0.0018311981111764908],
        "residual": 0.005519231850580564,
        "iterations": 8
    },
    1.05: {
        "x_value": [0.9919037670536189, 0.992356448533281, 0.996391942926121, 0.992356448533281,
0.9927838850709919, 0.9965937162382192, 0.996391942926121, 0.9965937162382192,
0.9983921045194817],
        "x_error_bound": [0.010075438202589715, 0.009508479969146943, 0.004488318788533396,
0.009508479969146943, 0.008976621170816768, 0.004237310931498928, 0.004488318788533396,
0.004237310931498928, 0.0020001744078126382],
        "residual": 0.007052994937660294,
        "iterations": 7
    },
    1.10: {
        "x_value": [0.9904388536409124, 0.991669342802714, 0.9963741222051301,
0.991669342802714, 0.99274924441026, 0.9968466793150472, 0.9963741222051301, 0.9968466793150472,
0.9986276694999902],
        "x_error_bound": [0.015402580612653094, 0.013592104723189724, 0.005938810860383947,
0.013592104723189613, 0.011888621720767767, 0.005179100882835352, 0.005938810860383947,
0.005179100882835352, 0.0022476282671315406],
        "residual": 0.009926467042285742,
        "iterations": 6
    },
    1.15: {
        "x_value": [0.9963966603280348, 0.9973381093985701, 0.9990099272868874,
0.9973381093985701, 0.998031245198775, 0.9992642792127158, 0.9990099272868874,
0.9992642792127158, 0.99971793738871],
        "x_error_bound": [0.006825083598134607, 0.006325245333318463, 0.0024327316981361813,
0.006325245333318463, 0.004952791521272482, 0.0018319307845028332, 0.0024327316981361813,
```

```
0.0018319307845028332, 0.0006577829980327632],
        "residual": 0.004354621262334481,
        "iterations": 6
    },
    1.17157287525381: {
        "x_value": [0.9980821446295344, 0.9988659802012234, 0.9996498157729125,
0.9988659802012234, 0.9993251404484483, 0.9997843006956734, 0.9996498157729125,
0.9997843006956734, 0.9999187856184342],
        "x_error_bound": [0.002655956203621468, 0.0034796025036609723, 0.0011524327727938655,
0.0034796025036609723, 0.0024790512281850363, 0.0007260972938265553, 0.0011524327727938655,
0.0007260972938265553, 0.0001818770042230078],
        "residual": 0.002139947493037298,
        "iterations": 6
    },
    1.20: {
        "x_value": [0.9994314700799999, 1.0000911744, 1.0000919420928, 1.0000911744,
1.0002478841855997, 1.0000806363647998, 1.0000919420928, 1.0000806363647998,
0.9999992071065598],
        "x_error_bound": [0.003311569920000257, 0.0001243776000000005849, 0.00029816094719992137,
0.0001243776000000005849, 0.00021232189440034688, 0.00039473157120006874, 0.00029816094719992137,
0.00039473157120006874, 0.00024666645504001217],
        "residual": 0.001134913328472275,
        "iterations": 6
    },
    1.25: {
        "x_value": [0.9992225153928302, 1.0002787908734945, 0.9999292856809916,
1.0002787908734945, 1.0001027119869832, 0.9997966802051073, 0.9999292856809916,
0.9997966802051073, 0.9998097012863466],
        "x_error_bound": [0.01524083781077934, 0.004262575036477756, 0.0021638045291183516,
0.004262575036477756, 0.0031069059332367033, 0.0013727152686593413, 0.0021638045291183516,
0.0013727152686593413, 0.00044319408103510316],
        "residual": 0.005692682743705237,
        "iterations": 6
    },
    1.30: {
        "x_value": [1.000338636296191, 0.9997072668190278, 0.99990783437733, 0.9997072668190278,
0.9995969687546599, 0.9999894437046725, 0.99990783437733, 0.9999894437046725,
1.0001456790446923],
        "x_error_bound": [0.004943079052190047, 0.0013111415327656662, 0.0010575364591998149,
0.0013111415327656662, 0.0011167372918399301, 0.0010415605151800067, 0.0010575364591998149,
0.0010415605151800067, 0.0006541478335431172],
        "residual": 0.0019145826156498914,
        "iterations": 7
    },
    1.35: {
        "x_value": [1.0010737502759632, 0.9998466422509549, 1.0002944666049933,
0.9998466422509549, 0.9999455402412366, 1.000536834575022, 1.0002944666049933,
1.000536834575022, 1.0005747633804005],
        "x_error_bound": [0.0135547809322617, 0.004727546626867807, 0.003103829219501586,
0.004727546626867807, 0.003725999845253236, 0.0024243962632647387, 0.003103829219501586,
0.0024243962632647387, 0.0011816206440025123],
        "residual": 0.0055232200498079335,
        "iterations": 7
    },
    1.40: {
        "x_value": [0.9993496642997913, 1.0003763106357344, 1.0001534387498776,
1.0003763106357344, 1.0009622374997549, 1.0001122428804632, 1.0001534387498776,
1.0001122428804632, 0.9995229869064611],
        "x_error_bound": [0.00455674707047804, 0.0009268734326930428, 0.0014643153055302616,
0.0009268734326930428, 0.0006348706110610447, 0.0017987046896328351, 0.0014643153055302616,
0.0017987046896328351, 0.0018659708681961762],
        "residual": 0.002030788561813908,
        "iterations": 8
```

```
    },
    1.45: {
        "x_value": [0.9988380671223897, 1.0008065781871915, 0.99987349998729,
1.0008065781871915, 1.0014285125136426, 0.9994471785780412, 0.99987349998729,
0.9994471785780412, 0.9984438019651166],
        "x_error_bound": [0.0117528144341007, 0.004164406395995135, 0.00455801604520433 8,
0.004164406395995135, 0.0036978250200956353, 0.004898469998663835, 0.00455801604520433 8,
0.004898469998663835, 0.004123759154801676],
        "residual": 0.005706728118859898,
        "iterations": 8
    },
    1.50: {
        "x_value": [1.0019213105359768, 1.000206530236781, 0.9998050841968507,
1.000206530236781, 0.9976570433937013, 0.9988968172968682, 0.9998050841968507,
0.9988968172968682, 1.001161915605599],
        "x_error_bound": [0.004775318534370854, 0.0004525451455918983, 0.0012710995506495237,
0.0004525451455918983, 0.0033171758987009525, 0.0015778101471126682, 0.0012710995506495237,
0.0015778101471126682, 0.005140520871495191],
        "residual": 0.0027663496430651653,
        "iterations": 9
    },
    1.55: {
        "x_value": [1.0030801969042042, 0.999207268636609, 0.9995279189733173,
0.999207268636609, 0.9944504713626505, 0.9984769060255225, 0.9995279189733173,
0.9984769060255225, 1.0036014266009747],
        "x_error_bound": [0.01033066198850996, 0.00037831236843333116, 0.004581744185820691,
0.00037831236843336642, 0.003815272001404857, 0.006075619548736699, 0.004581744185820691,
0.006075619548736699, 0.012295652839510729],
        "residual": 0.006573860381760565,
        "iterations": 9
    },
    1.60: {
        "x_value": [0.9948328833063063, 0.9951302306263728, 0.9979052218721858,
0.995130230626373, 1.0018570613443716, 1.0042515354332167, 0.9979052218721858,
1.0042515354332167, 0.9976979642823394],
        "x_error_bound": [0.010089326548772037, 0.002102530897818644, 0.001734621976451911,
0.002102530897818533, 0.012655069647095951, 0.005158945606168586, 0.001734621976451911,
0.005158945606168586, 0.011807475824717528],
        "residual": 0.00723109009707769,
        "iterations": 10
    },
    1.65: {
        "x_value": [1.0056706904170978, 1.0048489075510507, 0.995599721733677,
1.0048489075510507, 0.9968874525304597, 0.9992005573792494, 0.9955997217336768,
0.9992005573792494, 0.9975223792199752],
        "x_error_bound": [0.005253156210882848, 0.0023536229351763804, 0.013128432310221205,
0.0023536229351766025, 0.011818072383328215, 0.0009633513249402981, 0.013128432310221538,
0.0009633513249402981, 0.005274667884495043],
        "residual": 0.00783722779846329,
        "iterations": 12
    },
    1.70: {
        "x_value": [1.0107146222998058, 1.0039652398337633, 1.0044593951630505,
1.0039652398337635, 1.0041712288161582, 1.0017679660865157, 1.0044593951630505,
1.0017679660865157, 0.9992289971013196],
        "x_error_bound": [0.022472333610870976, 0.0010426229136921705, 0.005577113484001 6945,
0.0010426229136926146, 0.0003755652704293677, 0.002918540758273247, 0.005577113484001 6945,
0.002918540758273247, 0.004019251573278537],
        "residual": 0.008152163440319507,
        "iterations": 15
    },
    1.75: {
        "x_value": [0.9937789560340302, 0.9969948533205718, 0.9993693001109707,
```

```
0.9969948533205718, 1.0043763103356023, 1.0024946376446766, 0.9993693001109707,
1.0024946376446766, 0.9960233238835765],
        "x_error_bound": [0.012325247251312677, 0.0011275563914708853, 0.0018474858491697166,
0.0011275563914708853, 0.009459685233535464, 0.0011548280772115 67, 0.0018474858491697166,
0.0011548280772115 67, 0.012189321523777608],
        "residual": 0.006689410542080807,
        "iterations": 18
    },
    1.80: {
        "x_value": [1.0094466554721524, 1.002192748088452, 1.0045997081405469,
1.0021927480884512, 1.0032964581775068, 0.9990669142048747, 1.004599708140547,
0.9990669142048746, 0.9968964010088309],
        "x_error_bound": [0.014576631751148206, 0.0037435568548322973, 0.005066090994366945,
0.0037435568548331855, 0.003149473744336717, 0.00813278627887537, 0.0050660909943675,
0.008132786278875703, 0.005933376210614516],
        "residual": 0.007203553586902034,
        "iterations": 23
    },
    1.85: {
        "x_value": [0.9988890180158344, 1.0062064438877056, 0.999746304420902,
1.0062064438877054, 1.0071233684365932, 1.0084016705067311, 0.9997463044209018,
1.0084016705067311, 1.0008271913368776],
        "x_error_bound": [0.006616617817249 87, 0.014508543147564001, 0.003003619644316724,
0.014508543147563557, 0.010600884535319821, 0.010102730878030486, 0.003003619644317057,
0.010102730878030486, 0.007342636759414933],
        "residual": 0.009698392634518445,
        "iterations": 30
    },
    1.90: {
        "x_value": [1.0007409721732339, 1.0030354926648777, 0.9998910405462256,
1.0030354926648772, 1.0076372483037295, 1.0072901718165914, 0.9998910405462251,
1.007290171816591, 0.9949383721943491],
        "x_error_bound": [0.004133744425095842, 0.00546972724744843, 0.003364048894560323,
0.005469727247447542, 0.009855032990245816, 0.004387514917400193, 0.003364048894560767,
0.004387514917400193, 0.018380840062776183],
        "residual": 0.007958316408537273,
        "iterations": 46
    },
    1.95: {
        "x_value": [1.004138369853569, 1.0008828143687496, 1.005280244261307, 1.000882814368749,
1.001166542048438, 0.9912286643409778, 1.0052802442613062, 0.9912286643409784,
0.9963585523862055],
        "x_error_bound": [0.0021094130853724913, 0.0053385997680253094, 0.005310928227725498,
0.005338599768026642, 0.008660454728382616, 0.01865999717377842, 0.005310928227725054,
0.018659997173777088, 0.001527609916470496],
        "residual": 0.00995697032894699,
        "iterations": 91
    }
}
```

D)

Corridas del programa mostrando los resultados numéricos calculados con Python, para R_{tol} = 0.01, N = 32 y w variable (1.00, 1.05, 1.10, ..., 1.95)

(Adicionalmente, incluimos los resultados para $w = 1.8214651907890225$, que es el valor ideal para $N = 32$)

(Incluimos los valores de T y ΔT solamente para $w = 1.00$ porque los resultados son demasiado extensos. Para el resto de valores de w, mostramos únicamente el residuo y el valor de iteraciones o k)

w = 1.00

```
TEST_N_32_W_1_00_X_VALUE = [0.9844438155853504, 0.9695961686188925, 0.9557397464704432,
0.9430881365050494, 0.9317795297399498, 0.9218778798476873, 0.9133801933883477,
0.9062281406471326, 0.9003220688252344, 0.8955357102870329, 0.8917303033386854,
0.8887673571649911, 0.8865197726892158, 0.8848813717193552, 0.8837750155269996,
0.8831593872319141, 0.8830342074986216, 0.8834432520297959, 0.8844741945043484,
0.8862541774345397, 0.8889402471736884, 0.8927044206004099, 0.8977141006457932,
0.9041096282218752, 0.9119816784434354, 0.9213517113025722, 0.9321585884729993,
0.9442537318145379, 0.9574059513799948, 0.9713155677558096, 0.9856360121146523,
0.9695961686188925, 0.9405561646126602, 0.9134301434036136, 0.8886355827401646,
0.8664448394050065, 0.8469869736345725, 0.8302613768425862, 0.8161597981729006,
0.8044931393206176, 0.795019765319884, 0.7874728658443966, 0.7815853615602592,
0.777111748144097, 0.7738469056929613, 0.7716421373298005, 0.7704184964247083,
0.7701768931696238, 0.7710037355579037, 0.7730702457997782, 0.7766234122590461,
0.7819670267312322, 0.7894324851881928, 0.7993408372456252, 0.8119595787057556,
0.827459385547406, 0.8458768913280654, 0.8670893810871123, 0.8908058541614483,
0.9165765380175963, 0.943820097273609, 0.971865067812271, 0.9557397464704431,
0.9134301434036136, 0.8738705906808647, 0.8376704214358328, 0.8052305286839543,
0.7767453880723925, 0.7522220513115128, 0.7315113273536911, 0.7143460168354646,
0.7003815705995662, 0.689235631825631, 0.6805242607511359, 0.6738938970120505,
0.6690490014606313, 0.665775645352465, 0.6639610254981039, 0.6636080929479172,
0.6648434736138268, 0.6679160398550339, 0.6731832956576564, 0.6810834875670759,
0.6920931301489429, 0.70667220152656, 0.7252020858656735, 0.7479237055538089,
0.7748845098440109, 0.8059026150587763, 0.8405543516628889, 0.8781881116292616,
0.9179633825514601, 0.9589100293192024, 0.9430881365050493, 0.8886355827401645,
0.8376704214358326, 0.7909798661009428, 0.7490862387656058, 0.7122487694005666,
0.6804870004603343, 0.653619843405596, 0.6313138538282435, 0.6131348881604459,
0.5985986393840356, 0.5872172054237077, 0.5785404016369835, 0.5721916282999768,
0.5678985059842042, 0.5655181318445025, 0.5650558361024998, 0.5666750898538688,
0.5706952428391385, 0.577573587750345, 0.5878692469192102, 0.6021886444692296,
0.6211155546129934, 0.6451322316826427, 0.6745410479061025, 0.7093975454316066,
0.7494652966704678, 0.7942003818108356, 0.8427690704756263, 0.8940972786701883,
0.9469455875986716, 0.9317795297399497, 0.8664448394050064, 0.8052305286839541,
0.7490862387656058, 0.6986473211781286, 0.6542360539929891, 0.6158885540890838,
0.5834004580154588, 0.5563838355087926, 0.5343284561213187, 0.5166620587677251,
0.5028061895928805, 0.4922598138854934, 0.48447352483124606, 0.4792249483270977,
0.4763109072348841, 0.4757390629987268, 0.4777057326157733, 0.48259280480330247,
0.49094586952234864, 0.5034307369023183, 0.5207682175215917, 0.5436508285251631,
0.5726491896071015, 0.6081192543763898, 0.6501232094665358, 0.6983762323791278,
```

0.7522282451625246, 0.8106848467937866, 0.8724657336235019, 0.9360933154816627,
0.9218778798476872, 0.8469869736345723, 0.7767453880723924, 0.7122487694005668,
0.6542360539929892, 0.603089815090941, 0.5588658591706535, 0.5213443494773675,
0.4900940013296257, 0.464541584584021, 0.4440406483184004, 0.4279355063539412,
0.4156185339507041, 0.40658024950021504, 0.40045217649814274, 0.39704202235261554,
0.39635947564676244, 0.39862939089953014, 0.4042879663446303, 0.4139574119392296,
0.4283960388886105, 0.44842376426024627, 0.47482729168336685, 0.5082538152407967,
0.5491058555890426, 0.5974516914089736, 0.6529650965429903, 0.7149046461414277,
0.7821372884980136, 0.8532042875717043, 0.926421351895541, 0.9133801933883476,
0.8302613768425862, 0.7522220513115129, 0.6804870004603343, 0.6158885540890838,
0.5588658591706535, 0.5094964375305918, 0.4675516724375994, 0.4325670157552877,
0.4039184122652072, 0.3808982297880561, 0.3627862705668807, 0.34891361446939795,
0.33871858727684306, 0.3317947155018871, 0.3279300394410099, 0.32713583351015435,
0.3296611501430279, 0.3359883846809947, 0.3468049943470748, 0.36294811507528585,
0.38532218898590664, 0.41479437552241094, 0.4520775052843524, 0.49761440906313126,
0.5514794452274361, 0.6133122044932202, 0.682294601874291, 0.757176495979138,
0.8363477918675883, 0.9179481234809694, 0.9062281406471325, 0.8161597981729006,
0.7315113273536911, 0.653619843405596, 0.5834004580154588, 0.5213443494773675,
0.4675516724375993, 0.4217904527724103, 0.3835716540053371, 0.3522313058230584,
0.32701245832321674, 0.3071421378291 56, 0.291900783793453, 0.2806832847133306, 0.27305133859824,
0.2687763523969933, 0.26787070654740525, 0.2706035035650993, 0.2774956630511627,
0.28928920341037556, 0.30688730956642873, 0.3312654068649711, 0.3633584414968664,
0.40393487851214277, 0.4534722551575559, 0.5120512301898322, 0.5792841542662854,
0.6542901590131239, 0.7357222867796092, 0.821844518008708, 0.9106492251910216,
0.9003220688252344, 0.8044931393206176, 0.7143460168354646, 0.6313138538282435,
0.5563838355087926, 0.49009400132962566, 0.43256701575528766, 0.3835716540053371,
0.3426017119315098, 0.3089627423625547, 0.2818589494643393, 0.26047507887518656,
0.2440505443588676, 0.23194474569137669, 0.22369317296664937, 0.21905336739351386,
0.21803837237612167, 0.22093354692128683, 0.2282913317184623, 0.24089857415697835,
0.25971289917218304, 0.2857684369477336, 0.3200564554844254, 0.36339201742104993,
0.41628231130092813, 0.4788145029683408, 0.5505799837700353, 0.6306476618711281,
0.7175921450815204, 0.8095746133796353, 0.904466473312292, 0.8955357102870329,
0.795019765319884, 0.7003815705995663, 0.6131348881604459, 0.5343284561213187,
0.464541584584021, 0.4039184122652072, 0.3522313058230584, 0.3089627423625547,
0.2733956918626318, 0.24470448251704474, 0.2220407068895471, 0.20461120479497308,
0.1917469312879762, 0.1829621853401717, 0.1780031409605202, 0.1768831516003871,
0.17990049753325477, 0.187632947702171, 0.20090355611798677, 0.22071409135030068,
0.24814648334991218, 0.2842381124622664, 0.32984254380965206, 0.38549200413292384,
0.4512801695910857, 0.5267828252366709, 0.6110295696734371, 0.7025326882296983,
0.7993709638389352, 0.8993181877251577, 0.8917303033386854, 0.7874728658443967,
0.689235631825631, 0.5985986393840357, 0.5166620587677251, 0.4440406483184004,
0.3808982297880561, 0.32701245832321674, 0.2818589494643393, 0.24470448251704469,
0.21470098888904823, 0.1909746575702677, 0.1727070230881726, 0.15920671773105904,
0.1499712578905846, 0.14473769507094902, 0.14351946632163257, 0.14662494940067075,
0.15465191921319404, 0.1684521780643428, 0.18906268852447516, 0.21760365059703618,
0.25514956540059786, 0.302585268458725, 0.36046373820331346, 0.4288848175802944,
0.5074129505034083, 0.5950475272332223, 0.6902521873302179, 0.7910408382528588,
0.8951099049749078, 0.8887673571649911, 0.7815853615602593, 0.6805242607511359,
0.5872172054237077, 0.5028061895928805, 0.4279355063539412, 0.36278627056688073,
0.30714213782915606, 0.26047507887518656, 0.22204070688954708, 0.1909746575702677,
0.1663841783750613, 0.14743165535433608, 0.13340865389534207, 0.123799752731651,
0.11833490711632517, 0.11702756303850688, 0.12019389462139819, 0.12844722181068857,
0.1426617646772588, 0.16390200906126676, 0.1933181712392001, 0.23201397459291762,
0.2808990181523425, 0.3405429382862605, 0.41105094307446255, 0.4919792397818515,
0.5823042747465826, 0.6804523094708351, 0.7843870853138275, 0.8917449068455414,
0.8865197726892158, 0.7771117481440972, 0.6738938970120505, 0.5785404016369835,
0.49222598138854934, 0.41561853395070414, 0.34891361446939795, 0.291900783793453,
0.2440505443588676, 0.20461120479497308, 0.17270702308817257, 0.1474316553543361,
0.12793353429991178, 0.11349166954858328, 0.10358107375827348, 0.0979264703779976,
0.09654141077946482, 0.0997480643485206, 0.10817162762179515, 0.12270341883232462,
0.14442889387308985, 0.174521112331814, 0.21410600445384287, 0.2641119520836614,
0.3251211899392191, 0.3972429415001384, 0.4800271215897129, 0.5724327616102276,
0.6728578030045187, 0.7792279986494088, 0.8891341089274929, 0.8848813717193553,

0.7738469056929613, 0.6690490014606313, 0.5721916282999768, 0.48447352483124606,
0.4065802495002151, 0.3387185872768431, 0.2806832847133306, 0.23194474569137669,
0.1917469312879762, 0.15920671773105904, 0.13340865389534207, 0.11349166954858328,
0.09872616294626001, 0.08858060981989473, 0.08277628099829161, 0.08132711601390663,
0.08455992637300969, 0.09310878671310674, 0.1078776141616971, 0.12996715104032158,
0.160566923823968, 0.20081864781043648, 0.2516637770667949, 0.3136929377542017,
0.387017401268299, 0.4711816425580523, 0.5651312890849203, 0.6672431685914267,
0.7754151638283622, 0.8872049440484525, 0.8837750155269996, 0.7716421373298004,
0.665775645352465, 0.5678985059842043, 0.4792249483270977, 0.4004521764981428,
0.3317947155018871, 0.27305133859824005, 0.22369317296664937, 0.18296218534017167,
0.1499712578905846, 0.12379975273165097, 0.10358107375827347, 0.08858060981989473,
0.07826315020951297, 0.07234829843401495, 0.07085085999412029, 0.07410130190008629,
0.0827400747534777, 0.09767975648871488, 0.12003123533526884, 0.1509945621159291,
0.19172105303824524, 0.2431594951262197, 0.3059043613023269, 0.380066347340024,
0.4651843918946132, 0.5601935432806643, 0.6634553765197171, 0.7728486081851496,
0.8859088777653472, 0.8831593872319141, 0.7704184964247082, 0.6639610254981037,
0.5655181318445025, 0.4763109072348841, 0.39704202235261554, 0.32793003944100996,
0.2687763523969934, 0.21905336739351383, 0.17800314096052017, 0.14473769507094902,
0.11833490711632517, 0.09792647037799758, 0.08277628099829162, 0.07234829843401495,
0.0663623468104945, 0.06483476568069628, 0.06809894332028929, 0.07679947706067244,
0.09185390790429092, 0.11437828078617357, 0.14557724082749987, 0.18660536119542065,
0.23841268011971084, 0.30159246228952463, 0.3762515633370675, 0.4619225699556133,
0.5575320369437701, 0.661431443584249, 0.7714884199749634, 0.8852271227298876,
0.8830342074986215, 0.7701768931696237, 0.6636080929479171, 0.5650558361024998,
0.4757390629987267 6, 0.3963594756467623, 0.3271358335101543, 0.2678707065474052,
0.21803837237612161, 0.17688315160038703, 0.14351946632163254, 0.11702756303850687,
0.09654141077946479, 0.08132711601390663, 0.07085085999412029, 0.06483476568069628,
0.06329753710210347, 0.0665748514034935, 0.07531322724070821, 0.09043134236614303,
0.11304512690772556, 0.14435745495339594, 0.18551925238874076, 0.2374750959131759,
0.300811358706789, 0.3756272463809646, 0.46144778820705223, 0.5571929488870921,
0.6612093169327189, 0.7713617843731615, 0.8851740231811175, 0.8834432520297957,
0.7710037355579036, 0.6648434736138267, 0.5666750898538687, 0.47770573261577326,
0.39862939089953003, 0.3296611501430278, 0.27060350356509927, 0.2209335469212868,
0.17990049753325474, 0.14662494940067 07, 0.12019389462139816, 0.09974806434852057,
0.08455992637300967, 0.0741013019000863, 0.06809894332028929, 0.0665748514034935,
0.06986627501557227, 0.07861912995013715, 0.09374888364947224, 0.11636535908818917,
0.14766243078180566, 0.18877955988639922, 0.24064830306232043, 0.30384180606269534,
0.3784474684168621, 0.4639815978763363, 0.5593599355636969, 0.6629302500797715,
0.7725642436430675, 0.885797528489585, 0.8844741945043482, 0.773070245799778,
0.6679160398550337, 0.5706952428391383, 0.48259280480330224, 0.4042879663446302,
0.3359883846809946, 0.27749566305116 27, 0.22829133171846228, 0.18763294770217098,
0.154651919213194, 0.12844722181068857, 0.10817162762179515, 0.09310878671310674,
0.0827400747534777, 0.07679947706067243, 0.0753132272407082, 0.07861912995013715,
0.08735936264634356, 0.10244093753318192, 0.12496046401076993, 0.15609437410859017,
0.19696168473484887, 0.24847243502071528, 0.3111796552947718, 0.38515474790935955,
0.4699046910052299, 0.5643445201156992, 0.6668309349235553, 0.7752540769321675,
0.887176235319676, 0.8862541774345396, 0.7766234122590459, 0.6731832956576562,
0.5775735877503447, 0.4909458695223485, 0.4139574119392295, 0.3468049943470747,
0.2892892034103755, 0.2408985741569783, 0.20090355611798674, 0.16845217806434276,
0.1426617646772588, 0.12270341883232462, 0.10787761416169708, 0.09767975648871487,
0.0918539079042909, 0.090431342366143, 0.09374888364947223, 0.10244093753318194,
0.1173996057619433, 0.13969977412270537, 0.17049056028576776, 0.21086030306848413,
0.26168815257799893, 0.32349982250574666, 0.3963469005050658, 0.4797275374598972,
0.5725613935455058, 0.673224248870022, 0.7796391184394422, 0.8894126995482082,
0.8889402471736884, 0.7819670267312322, 0.6810834875670758, 0.58786924691921,
0.5034307369023181, 0.42839603888861044, 0.36294811507528574, 0.3068873095664286,
0.259712899172183, 0.22071409135030062, 0.1890626885244751, 0.16390200906126678,
0.14442889387308985, 0.12996715104032158, 0.12003123533526883, 0.11437828078617357,
0.11304512690772554, 0.11636535908818915, 0.12496046401076993, 0.13969977412270537,
0.16162640939903067, 0.1918508459790893, 0.23141935804292885, 0.2811701996522932,
0.341594618887312, 0.4127214102884671, 0.4940420368947421, 0.5844884693218961,
0.6824686331107467, 0.7859561121170383, 0.8926231078953031, 0.8927044206004099,
0.7894324851881928, 0.692093130148943, 0.6021886444692295, 0.5207682175215916,

0.44842376426024616, 0.3853221889859066, 0.331265406864971, 0.2857684369477336,
0.24814648334991213, 0.21760365059703618, 0.19331817123920014, 0.17452111233181405,
0.16056692382396803, 0.15099456211592913, 0.1455772408274999, 0.14435745495339597,
0.14766243078180566, 0.15609437410859017, 0.17049056028576776, 0.19185084597908925,
0.22123447702716803, 0.2596334313497408, 0.30783482412125546, 0.3662887948175152,
0.43499966541039237, 0.5134564040817584, 0.600613663960524, 0.6949277127499358,
0.7944437494531431, 0.8969239103339867, 0.897714100645793, 0.7993408372456252, 0.70667220152656,
0.6211155546129934, 0.5436508285251631, 0.47482729168336685, 0.41479437552241094,
0.36335844149686636, 0.3200564554844254, 0.2842381124622664, 0.25514956540059786,
0.23201397459291762, 0.21410600445384292, 0.2008186478104365, 0.19172105303824527,
0.18660536119542068, 0.18551925238874076, 0.18877955988639922, 0.19696168473484887,
0.21086030306848413, 0.23141935804292882, 0.25963343134974076, 0.2964276231729734,
0.3425279518127538, 0.3983377935049415, 0.46383699734918554, 0.5385185081299377,
0.621372756098539, 0.7109235188871132, 0.805311667523694, 0.9024165566059927,
0.9041096282218751, 0.8119595787057555, 0.7252020858656734, 0.6451322316826427,
0.5726491896071015, 0.5082538152407968, 0.45207750528435253, 0.40393487851214277,
0.36339201742104993, 0.32984254380965206, 0.302585268458725, 0.28089901815234253,
0.26411195208366145, 0.2516637770667949, 0.2431594951262197, 0.23841268011971084,
0.23747509591317587, 0.24064830306232038, 0.24847243502071525, 0.261688152577999,
0.28117019965229323, 0.30783482412125546, 0.3425279518127538, 0.38590541044583704,
0.43831958774171165, 0.499727769110632, 0.5696355962549936, 0.6470847861109803,
0.7306881956310294, 0.8187086514914838, 0.9091719360411217, 0.9119816784434354,
0.8274593855474058, 0.7479237055538088, 0.6745410479061027, 0.60811925437639,
0.5491058555890427, 0.4976144090631314, 0.45347225515755596, 0.4162823113009282,
0.38549200413292384, 0.3604637382033134, 0.34054293828626053, 0.32512118993921924,
0.31369293775420176, 0.3059043613023269, 0.30159246228952463, 0.300811358706789,
0.30384180606269534, 0.3111796552947718, 0.3234998225057467, 0.341594618887312,
0.3662887948175152, 0.3983377935049415, 0.43831958774171165, 0.48653311414526823,
0.5429169427889144, 0.6070000688172745, 0.6778927663841399, 0.7543199881901476,
0.8346938460390471, 0.9172163749808027, 0.9213517113025722, 0.8458768913280654,
0.7748845098440109, 0.7093975454316066, 0.6501232094665359, 0.5974516914089737,
0.5514794452274361, 0.5120512301898322, 0.4788145029683408, 0.45128016959108574,
0.4288848175802945, 0.4110509430744626, 0.3972429415001385, 0.3870174012682991,
0.3800663473400241, 0.37625156333706755, 0.3756272463809647, 0.37844746841686216,
0.3851547479093596, 0.39634690050506594, 0.41272141028846704, 0.4349996654103924,
0.46383699734918554, 0.499727769110632, 0.5429169427889144, 0.5933299702970306,
0.6505312124153044, 0.7137175816707337, 0.7817493311945116, 0.8532147596893929,
0.9265210201362851, 0.9321585884729992, 0.8670893810871124, 0.8059026150587763,
0.7494652966704677, 0.6983762323791277, 0.6529650965429903, 0.6133122044932202,
0.5792841542662854, 0.5505799837700353, 0.5267828252366709, 0.5074129505034084,
0.4919792397818516, 0.480027121589713, 0.47118164255805234, 0.46518439189461325,
0.4619225699556134, 0.46144778820705223, 0.4639815978763363, 0.46990469100522997,
0.47972753745989727, 0.4940420368947421, 0.5134564040817584, 0.5385185081299377,
0.5696355962549935, 0.6070000688172745, 0.6505312124153044, 0.6998413318991477,
0.7542317210336293, 0.8127198924913497, 0.8740951894584347, 0.9369961059820233,
0.9442537318145379, 0.8908058541614483, 0.8405543516628889, 0.7942003818108356,
0.7522282451625245, 0.7149046461414277, 0.682294601874291, 0.654290159013124,
0.6306476618711281, 0.6110295696734371, 0.5950475272332223, 0.5823042747465828,
0.5724327616102277, 0.5651312890849203, 0.5601935432806643, 0.5575320369437701,
0.5571929488870921, 0.5593599355636969, 0.5643445201156992, 0.572561393545506,
0.5844884693218962, 0.6006136639605241, 0.6213727560985391, 0.6470847861109803,
0.6778927663841399, 0.7137175816707336, 0.7542317210336293, 0.7988570475449581,
0.846787596635523, 0.897034983891795, 0.9484910120718925, 0.9574059513799948,
0.9165765380175963, 0.8781881116292616, 0.8427690704756261, 0.8106848467937866,
0.7821372884980134, 0.7571764959791378, 0.7357222867796092, 0.7175921450815204,
0.7025326882296982, 0.6902521873302179, 0.6804523094708351, 0.6728578030045187,
0.6672431685914268, 0.6634553765197171, 0.661431443584249, 0.6612093169327189,
0.6629302500797714, 0.6668309349235553, 0.673224248870022, 0.6824686331107467,
0.6949277127499358, 0.7109235188871132, 0.7306881956310293, 0.7543199881901476,
0.7817493311945115, 0.8127198924913497, 0.846787596635523, 0.8833382658876859,
0.9216220044635122, 0.9608002610191924, 0.9713155677558096, 0.9438200972736089,
0.9179633825514599, 0.8940972786701882, 0.8724657336235018, 0.8532042875717041,
0.8363477918675883, 0.821844518008708, 0.8095746133796353, 0.799370963838935,

```
0.7910408382528586, 0.7843870853138275, 0.7792279986494087, 0.7754151638283624,
0.7728486081851498, 0.7714884199749635, 0.7713617843731616, 0.7725642436430675,
0.7752540769321676, 0.7796391184394422, 0.7859561121170383, 0.7944437494531431,
0.805311667523694, 0.8187086514914838, 0.834693846039047, 0.8532147596893928,
0.8740951894584347, 0.8970349838917951, 0.9216220044635122, 0.9473550160024385,
0.9736748226980783, 0.9856360121146523, 0.9718650678122709, 0.9589100293192023,
0.9469455875986716, 0.9360933154816627, 0.9264213518955409, 0.9179481234809694,
0.9106492251910216, 0.904466473312292, 0.8993181877251577, 0.8951099049749077,
0.8917449068455414, 0.8891341089274929, 0.8872049440484526, 0.8859088777653473,
0.8852271227298877, 0.8851740231811176, 0.885797528489585, 0.8871762353196762,
0.8894126995482083, 0.8926231078953031, 0.8969239103339867, 0.9024165566059927,
0.9091719360411217, 0.9172163749808027, 0.9265210201362851, 0.9369961059820233,
0.9484910120718926, 0.9608002610191924, 0.9736748226980783, 0.9868374113490391]

TEST_N_32_W_1_00_X_ERROR_BOUND = [0.00037096949102066734, 0.0007085374481916595,
0.0010010526382279172, 0.0012414798485969047, 0.0014274803314391349, 0.0015609104868173906,
0.0016469224586322944, 0.0016928790981006747, 0.0017070281057458765, 0.0016985767765654,
0.0016759098885130719, 0.0016459251958577115, 0.0016154221803145186, 0.0015899333590496,
0.0015740089085549691, 0.0015711273312435380, 0.0015834367258661386, 0.0016113196627476345,
0.0016528613396367131, 0.0017033812345105260, 0.0017552335294167554, 0.0017980675431311655,
0.0018196584151429773, 0.0018072847395571578, 0.0017494786030273213, 0.0016378495633956325,
0.0014686275465310983, 0.0012436013281955650, 0.0009702434808318561, 0.0006609790731264553,
0.0003317294691886907 3, 0.0007085374481916595, 0.0013539995546557604, 0.0019139827971856870,
0.0023748325989210173, 0.0027318263111267838, 0.0029882549134885306, 0.0031537446859144502,
0.0032422167404936310, 0.0032698572761892253, 0.0032533864471591123, 0.0032087947458211420,
0.0031505867478814897, 0.0030914529156753634, 0.0030421981232006123, 0.0030117068418282766,
0.0030067337968067154, 0.0030313827740998667, 0.003082662681020534 8, 0.0031675160122099920,
0.0032659106695072104, 0.0033366568528297087, 0.0034494921908918563, 0.0034911987451808900,
0.0034673663044559877, 0.0033561617347162010, 0.0031416854876800127, 0.0028168689840299965,
0.0023852232754590920, 0.0018610529948381860, 0.0012686026027582950, 0.0006365871094711206,
0.0010010526382278062, 0.0019139827971857981, 0.0027066782907573250, 0.0033595383477717026,
0.0038650022651266490, 0.0042278449434678400, 0.0044612923964952870, 0.0045849341962357350,
0.0046217054238540944, 0.0045959430026255842, 0.0045290119189916880, 0.0044433327479085350,
0.0043567462548976810, 0.0042849290402106055, 0.0042407798811259220, 0.0042340747565092760,
0.0042706916486021650, 0.0043514063692000615, 0.0044704883077119060, 0.0046145280587254200,
0.0047620330721824810, 0.0048842752282579616, 0.0049476507793447810, 0.0049174603228412380,
0.0047626263537619010, 0.0044605534848098930, 0.0040010139739738050, 0.0033895329415037923,
0.0026458020980996144, 0.0018035799332965980, 0.0009058674583997980, 0.0012414798485967937,
0.0023748325989210173, 0.0033593834777168040, 0.0041698801974687780, 0.0047969279995528910,
0.0052454783445094140, 0.0055315332867168510, 0.0056806948868555200, 0.0057203937340040145,
0.0056810078241693596, 0.0055914765287020660, 0.0054783065622070470, 0.0053648949044351470,
0.0052712906338027166, 0.0052140141063706790, 0.0052055729285757080, 0.0052534502231851300,
0.0053585754517048170, 0.0055135704315394385, 0.0057013117286922600, 0.0058944739604801510,
0.0060566473759414710, 0.0061453412441540180, 0.0061161747652311627, 0.0059316534631473905,
0.0055615165988329720, 0.0049935546087871170, 0.0042335375391066627, 0.0033307316257295634,
0.0022561192280409204, 0.0011333920685593259 3, 0.0014274803314390239, 0.0027318263111267838,
0.0038650022651265380, 0.0047969279995528910, 0.0055160519566282980, 0.0060276504023378810,
0.0063505214785665180, 0.0065128723013394560, 0.0065481607731556760, 0.0064914907626781870,
0.0063769200791602420, 0.0062357747657694200, 0.0060959548149425730, 0.0059809066067870180,
0.0059107569586493440, 0.0059002972184966640, 0.0059584577783311610, 0.0060863553242236360,
0.0062752391773037710, 0.0065048299764937645, 0.0067428216361868350, 0.0069462298816659330,
0.0070649400993100240, 0.0070474729779815280, 0.0068470288212046930, 0.0066430336072819509,
0.0057818340878091900, 0.0049081204708689710, 0.0038382177735257760, 0.0026207388829135870,
0.0013182730763666717, 0.0015609104868172796, 0.0029888254913488530 6, 0.0042278449434679510,
0.0052454783445411435, 0.0060276504023379920, 0.0065797223081970335, 0.0069222829817608970,
0.0070865896458490730, 0.0071099364967981970, 0.0070316224565492025, 0.0068899235799652070,
0.0067201786198537450, 0.0065538178508299790, 0.0064179512322051770, 0.0063350240215896410,
0.0063220788000943880, 0.0063893442807888000, 0.0065381789150310565, 0.0067587621321874570,
0.0070282416696140770, 0.0073101953608218360, 0.0075561639712802830, 0.0077096415262481455,
0.0077123425753813370, 0.0075119489904034165, 0.0070700516421453585, 0.0063687978551383310,
0.0054149087015533670, 0.0042402117016060450, 0.0028985209469679640, 0.0014594111910634089,
0.0016469224586322944, 0.0031537446859146723, 0.0044612923964953980, 0.0055315332867168510,
0.0063505214785665180, 0.0069222829817608970, 0.0072691298595212170, 0.0074246565470690860,
```

0.007429157550416954, 0.007325256118502765, 0.0071546076002879655, 0.006955799162270904,
0.006763264677081937, 0.006606800534485979, 0.0065111508891500525, 0.0064951651295482304,
0.006570228258946209, 0.00673799901988581, 0.0069878852749213904, 0.007295025984924042,
0.007619708651357748, 0.007909038512242506, 0.008101274239768252, 0.008132632210272883,
0.007945696090975785, 0.007498043334971949, 0.0067694795492347914, 0.005766435810502624,
0.004522601547034433, 0.003095604406764485, 0.001560318952172679, 0.0016928790981005637,
0.003242216740493631, 0.004584934196235735, 0.00568069488685552, 0.006512872301339567,
0.007086589645849184, 0.007424656547069086, 0.007562408234842832, 0.007542420898115798,
0.007409879289306587, 0.007209071159209712, 0.006981141495946586, 0.006762917175803185,
0.006586364188308269, 0.006478114677139157, 0.006458537738664338, 0.006540038717702312,
0.006724626808745637, 0.0070012091769928175, 0.007343429066228979, 0.007709032974634877,
0.00804163153951909, 0.00827529315057568, 0.008341761110096046, 0.008179381670889885,
0.007742274696178497, 0.007008044127619462, 0.005982496582710994, 0.004700380696185835,
0.0032219386179934695, 0.0016258750302731162, 0.001707281057458765, 0.0032698572761892253,
0.004621705423854094, 0.0057203937340040145, 0.006548160773155787, 0.007109936496798253,
0.007429157550417009, 0.007542420898115854, 0.007493990971611186, 0.007330972504872357,
0.007099648115796331, 0.0068431230663097975, 0.006600082008631797, 0.006404201829451256,
0.006283633447236009, 0.006260003674493336, 0.0063466088500020446, 0.006545843225492537,
0.006846342104052078, 0.007220694505507497, 0.00762475471758145, 0.007999456386475257,
0.008275589286388552, 0.008381323640363869, 0.008251533739632277, 0.007837392732669868,
0.007114463818358896, 0.0060876880426320445, 0.004792232346060299, 0.003289970152749455,
0.0016622140378828032, 0.001698857677656429, 0.0032533864471591123, 0.004595430026255953,
0.005681007824169359 6, 0.006491490762678187, 0.0070316224565492025, 0.007325256118502765,
0.007409879289306587, 0.007330972504872357, 0.0071370415155241895, 0.006875841154335843,
0.006591941032532023, 0.006325433890067705, 0.006111317114081055, 0.005978941482211342,
0.005950960276227157, 0.006041439732613 84, 0.006253175403738598, 0.006574710452242144,
0.0069779390735127955, 0.007417358946233854, 0.007831907794168674, 0.008149862749850745,
0.008296585683316482, 0.008204141743487992, 0.007821219160159265, 0.007121521219652149,
0.006108977707544039, 0.0048186997812398324, 0.003313432685031592, 0.0016761330466465996,
0.0016759098885130719, 0.003208794745821142, 0.004529011918991466, 0.005591476528702177,
0.006376920079160242, 0.006889923579965207, 0.0071546076002879655, 0.007209071159209712,
0.007099648115796331, 0.006875841154335788, 0.006586466004855546, 0.006277162309215933,
0.005989069330590152, 0.0057581869234644145, 0.005614801468816377, 0.005582395830327047,
0.005675695386242241, 0.005897895039254447, 0.006237574700701021, 0.006666207855611783,
0.0071373539691858, 0.007588495207837442, 0.007946012512205392, 0.008133085492868841,
0.008079527213124826, 0.007731950334235016, 0.007062394883738721, 0.006073724095731037,
0.004800680320286244, 0.0033063391849627033, 0.001674591225504063, 0.0016459251958577115,
0.0031505867478814897, 0.004443332747908313, 0.005478306562206936, 0.006235774765769475,
0.006720178619853745, 0.006955799162271015, 0.006981141495946586, 0.0068431230663097975,
0.006591941032532023, 0.006277162309215933, 0.005945197630168403, 0.005637956174481151,
0.005392193769022835, 0.0052389234811653085, 0.005202296619024174, 0.00529759878472906,
0.0055284053860956, 0.005883412269485244, 0.006333862306357824, 0.006832679767208416,
0.007316293613835151, 0.007709659015684989, 0.007934264517880929, 0.007918125310715007,
0.007606136167353394 5, 0.006968883722674624, 0.006008193000470841, 0.004758274680780272,
0.0032821972209495787, 0.0016643129239268806, 0.0016154221803145186, 0.0030914529156755854,
0.004356746254897681, 0.005364894904435147, 0.006095814942547573, 0.006553817850830035,
0.006763264677081882, 0.006762917175803129, 0.006600082008631769, 0.006325433890067678,
0.005989069330590124, 0.005637956174481207, 0.005314574472120553 4, 0.005056256303095863,
0.004894585114897196, 0.004854253969074712, 0.004951021298697622, 0.005188807843308787 5,
0.005556456584534006, 0.006025089292379146, 0.006547188455121644, 0.007058402610549186,
0.007482596356431947 5, 0.007739935501479556, 0.007757000056942132, 0.00747728112801993,
0.006870137786289532, 0.005936464429025734, 0.004709915290856848, 0.0032533990394072454,
0.0016514718340626677, 0.001589933355905071, 0.0030421981232005013, 0.0042849290402106055,
0.005271290633802716 6, 0.005980906606787018, 0.006417951232205232, 0.006606800534486035,
0.006586364188308269, 0.006404201829451256, 0.006111317114081027, 0.005758186923464387,
0.005392193769022835, 0.005056256303095863, 0.004788161590557061, 0.004619953475957991,
0.004576769841667105, 0.004674762817763661, 0.004918145417051462, 0.005295891925080773,
0.005779036549420774, 0.0063197131628183745, 0.006852947706412787 5, 0.007301733501978408,
0.007585181220611636 5, 0.007628728739365542, 0.007374752084850333, 0.006791634662392465,
0.005879526693966852, 0.004671627264416389, 0.0032306946425035843, 0.0016414138334311934,
0.001574008908554858, 0.0030117068418281656, 0.004240779881125922, 0.00521401410637079,
0.005910756958649399, 0.006335024021589697, 0.006511150889149997, 0.006478114677139157,
0.006283633447236037, 0.005978941482211342, 0.005614801468816405, 0.005238923481165281,

0.004894585114897182, 0.004619953475957991, 0.004447458791092027, 0.00440260783488376 2,
0.004501868672208448, 0.0047496715585007565, 0.005135059508131881, 0.005628943264861061,
0.006183115326324623, 0.006732044500650708, 0.007197985920493338, 0.007499195767974309,
0.007560226474982667 5, 0.007322629416827819, 0.006754106911369351, 0.005854332436273535,
0.004656263429307361, 0.003222650327557419, 0.0016383743273922224, 0.001571127331243538,
0.003006733796806604 4, 0.004234074756509054, 0.005205572928575708, 0.005900297218496719 5,
0.006322078800094388, 0.006495165129548286, 0.006458537738664338, 0.006260003674493308 5,
0.005950960276227157, 0.005582395830327075, 0.005202296619024202, 0.004854253969074712,
0.004576769841667119, 0.004402607834883762, 0.004357575604151093, 0.004458370493722261,
0.004709537332419855, 0.005100079883808534, 0.005600690873550376, 0.006162764589413047,
0.006720218640772185, 0.007194658327050601 5, 0.00750366454721646, 0.007571167883834606,
0.007338221530841293 5, 0.006772203119627618, 0.005872657891056066, 0.004672607235064019,
0.003235031020399237, 0.001645163902100677, 0.0015834367258660276, 0.003031382774099866 7,
0.004270691648602165, 0.005253450223185241, 0.005958457783311555, 0.006389344280788745,
0.006570228258946209, 0.006540038717702257, 0.006346608850002044 6, 0.006041439732613812 5,
0.005675695386242241, 0.00529759878472906, 0.004951021298697608, 0.004674762817763661,
0.004501868672208448, 0.004458370493722261, 0.004561087171802577 5, 0.004814542879982855 5,
0.005207554102525991, 0.00571046049512719, 0.0062741691893187235, 0.006832037413869174,
0.007305117045297682, 0.007610526161662995, 0.00767189277695457, 0.007430169498229056,
0.006852842668993497, 0.005939750697474655, 0.004724345013773945, 0.003270115070493240 2,
0.0016628300018778353, 0.0016113196627474125, 0.00308626810205348, 0.004351406369200061 5,
0.005358575451704817, 0.006086355324223636, 0.006538178915031001, 0.006737999019885865,
0.006724626808745582, 0.006545843225492565, 0.006253175403738626, 0.005897895039254447,
0.005528405386095586, 0.005188807843308774, 0.004918145417051448, 0.00474967155850077,
0.004709537332419855, 0.004814542879982855 5, 0.005069022962173236, 0.005461431322486096,
0.005961607048306131, 0.006519892149130332 4, 0.007069114046890396, 0.007529936231765733,
0.007819317847902962, 0.007861006320975417, 0.007596350474918745, 0.006993459574323069,
0.006052937085095245, 0.004809044122905504, 0.003326036237343177 3, 0.001690342868164163,
0.0016528613396367131, 0.003167516012209992, 0.004470488307711795, 0.005515704315393275,
0.006275239177303 66, 0.006758762132187457, 0.006978788527492139 04, 0.007001209176992873,
0.006846342104052105 5, 0.006574710452242144, 0.006237574700700993, 0.005883412269485244,
0.005556456584534006, 0.005295891925080787, 0.005135059508131867, 0.00510007988380852,
0.005207554102525977, 0.005461431322486082, 0.005849621703241589, 0.006341342190631968,
0.006886354950126364, 0.007417088677586142, 0.007854113574215893, 0.00811467831272869,
0.008123210755043297, 0.007822064987584254, 0.007180554602400857, 0.006200530581206398,
0.004917394506609662 5, 0.00339632243425092, 0.0017243881855211196, 0.001703381234510526,
0.003265910669506988 4, 0.004614528058725309, 0.005701311728692149, 0.006504829976493653 5,
0.007028241669614132 5, 0.007295025984923931, 0.007343429066228924, 0.007220694505507497,
0.006977939073512823, 0.006666207855611783, 0.006333862306357851, 0.006025089292379132,
0.005779036549420774, 0.005628943264861061, 0.005600690873550376, 0.005710460495127176,
0.005961607048306117, 0.006341342190631996, 0.006818210985674949 5, 0.007341500132000689,
0.007843529804615 64, 0.008245254832064675, 0.00846484483203569, 0.008428125331845282,
0.008079168982090545, 0.007389108759166096 4, 0.006361480068581282, 0.005033033869187964,
0.003469836411549631 6, 0.0017593636912566213, 0.0017552335294167554, 0.003366568528297087,
0.004762033072182592, 0.005894473960480151, 0.006742821636186724, 0.007310195360821892,
0.007619708651357693, 0.007709032974634766, 0.007624754717581506, 0.007417358946233854,
0.007137353969185772, 0.006832679767208444, 0.006547188455121616, 0.006319713162818388,
0.006183115326324637, 0.006162764589413061, 0.006274169189318 71, 0.006519892149130318 6,
0.006886354950126364, 0.007341500132000689, 0.007834413162074505, 0.00829780130373331,
0.008653698535572185, 0.008822024603398093, 0.0087308682600713, 0.008326807384040902,
0.007583392042373627, 0.006506167716303079 6, 0.005133248127612844, 0.003535130001841017 73,
0.001787663461588207, 0.0017980675431311655, 0.003449492190891856 3, 0.004488427522825818 4,
0.006056647375941471, 0.006946229881665933, 0.007556163971280283, 0.007909038512242561,
0.008041631539519145, 0.007999456386475312, 0.007831907794168647, 0.00758849520783747,
0.007316293613835179, 0.007058402610549214, 0.006852947706412815, 0.006732044500650763,
0.006720218640772213, 0.006832037413869174, 0.007069114046890368, 0.007417088677586142,
0.007843529804615 64, 0.008297801303733254, 0.00871372230296602, 0.009015323274494003,
0.009125286760251 15, 0.008974944330733314, 0.008514188501270448, 0.007719505856747633,
0.006598602096320905, 0.005190710597495318, 0.003562496968893325, 0.001800290418021 1,
0.001819658415142866 3, 0.003491198745181001, 0.004947650779344781, 0.006145341444154018,
0.007064940099310024, 0.007709641526248201, 0.008101274239768252, 0.00827529315057568,
0.008275589286388607, 0.008149862749850745, 0.007946012512205364, 0.007709659015684933,
0.007482596356431947 5, 0.007301733501978436, 0.007197985920493338, 0.007194658327050629,

0.007305117045297682, 0.007529936231765733, 0.007854113574215893, 0.008245254832064702,
0.008653698535572185, 0.009015323274493947, 0.009257270205111412, 0.00930613681654785,
0.009097534152570197, 0.008585439038608922, 0.007749659722416425, 0.006599975812538017,
0.005176217442890207 5, 0.003544114494061334 5, 0.001787772618492389 2, 0.001807284739557046 8,
0.003467366304455876 7, 0.004917460322841127, 0.006116747652311627, 0.007047297798152829 5,
0.007712342575381448, 0.008132632210272994, 0.008341761110096046, 0.008381323640363869,
0.008296585683316426, 0.008133085492868841, 0.007934264517880985, 0.007739935501479556,
0.007585181220611636 5, 0.007499195767974337, 0.007503664547216432, 0.007610526161662912,
0.007819317847902907, 0.008114678312728663, 0.008464844832035745, 0.008822024603398149,
0.009125286760251 15, 0.009306136816547 85, 0.009296309841421935, 0.009036718662791432,
0.008486090786919 92, 0.007627745452187584, 0.006473235251508891, 0.005062141960683664,
0.003458033724464515 4, 0.001741287474831598 7, 0.001749478603027321 3, 0.003356161734716 09,
0.004762626353761 79, 0.005931653463147 5015, 0.006847028821204804, 0.007511948990403527 5,
0.007945696090975951, 0.008179381670889885, 0.008251533739632277, 0.008204141743487992,
0.008079527213124 77, 0.007918125310715063, 0.007757000056942243, 0.007628728739365598,
0.007560226474982612, 0.007571167883834606, 0.007671892776954 57, 0.007861006320975417,
0.008123210755043242, 0.008428125331845226, 0.008730868260071 3, 0.008974944330733314,
0.009097534152570141, 0.009036718662791432, 0.008739645452912337, 0.008170302240089633,
0.007315511897341254, 0.006188027964096676, 0.004826128431143051, 0.003289751794125495,
0.001653838913792560 2, 0.001637849563395743 5, 0.003141685487680123 7, 0.004460553484809893,
0.005561516598832 75, 0.006430336072819509, 0.007070051642145469 6, 0.007498043334971838,
0.007742746961784 97, 0.007837392732669868, 0.007821219160159 32, 0.007731950334235 0715,
0.007606136167353394 5, 0.007477281128019 93, 0.007374752084850444, 0.007322629416827930 4,
0.007338221530841293 5, 0.007430169498229056, 0.007596350474918745, 0.007822064987584 31,
0.008079168982090656, 0.008326807384040846, 0.008514188501270559, 0.008585439038608922,
0.008486090786919 92, 0.008170302240089633, 0.007607639317206316, 0.006788214610157661,
0.005725230821433391, 0.004454434694012077, 0.003030551281846039, 0.001521302293242987 6,
0.001468627546531098 3, 0.002816868984030107 5, 0.004001201397380694, 0.004993546087871059,
0.005781834087809 19, 0.006368797855138442, 0.006769479549234791 4, 0.007008044127619351,
0.007114463818358896, 0.007121521219652149, 0.007062394883738721, 0.006968883722674568,
0.006870137786289476, 0.006791634662392465, 0.006754106911369351, 0.006772203119627673,
0.006852842668993442, 0.006993459574322958, 0.007180554602400857, 0.007389108759166152,
0.007583392042373627, 0.007719505856747633, 0.007749659722416425, 0.007627745452187473,
0.007315511897341365, 0.006788214610157772, 0.006038872201535317, 0.005080269650638458,
0.003944345981731656, 0.002679045777463806 6, 0.001343160107078356, 0.001243601328195676,
0.002385223275459092, 0.003389532941503792 3, 0.004233753910666271, 0.004908120470869082,
0.005414908701553589, 0.005766435810502735, 0.005982496582711105, 0.006076876880426320445,
0.006108977707544039, 0.006073724095730926, 0.006008193000470952, 0.005936464429025734,
0.005879526693966852, 0.005854332436273424, 0.005872657891055955, 0.005939750697474544,
0.006052937085095245, 0.006200530581206398, 0.006361480068581393, 0.006506167716303079 6,
0.006598602096321016, 0.006599975812538239, 0.006473235251508891, 0.006188027964096676,
0.005725230821433391, 0.005080269650638569, 0.004266423489940167, 0.003305222579236155 6,
0.002241824113743962, 0.001122797738518977, 0.000970243480831856 1, 0.001861052994838186,
0.002645802098099725 4, 0.003307316257295634, 0.003838217773526109, 0.004240211701605934,
0.004522601547034322, 0.004700380696185835, 0.004792232346060188, 0.004818699781239721,
0.004800680320286244, 0.004758274680780272, 0.004709915290856848, 0.0046716272644165,
0.004656263429307361, 0.004672607235063908, 0.004724345013773945, 0.004809044122905393,
0.004917394506609662 5, 0.005033033869187964, 0.005133248127612733, 0.005190710597495429,
0.005176217442890207 5, 0.005062141960683664, 0.004826128431143162, 0.004454434694012077,
0.003944345981731656, 0.003305222579236044 6, 0.002557997915995259 47, 0.001733069323663238 8,
0.000867302605440878 3, 0.000660979073126455 3, 0.001268060275828819 5, 0.001803579933229659 8,
0.002256119228041031 4, 0.002620738882913698, 0.002898520946967853, 0.003095604406764485,
0.003221938617993469 5, 0.003289970152749455, 0.003313432685031 37, 0.003306339184962592 3,
0.003282197220949578 7, 0.003253399039407134 4, 0.003230694642503806 3, 0.003222265032755764 12,
0.003235031020399348, 0.003270115070493240 2, 0.003326036237343288 3, 0.003396322434251253 2,
0.003469836411549631 6, 0.003531300018410177 3, 0.003562496968893436, 0.003544114494061334 5,
0.003458033724464515 4, 0.003289751794125384, 0.003030551281846039, 0.002679045777463806 6,
0.002241824113743962, 0.001733069323663238 8, 0.001173206341622434 8, 0.000586795156713448 8,
0.000331729469188690 73, 0.000636587109471120 6, 0.000905867458379949 8, 0.001133920685593370 3,
0.001318273076366671 7, 0.001459411191063297 9, 0.001560318952172679, 0.001625875030273116 2,
0.001662214037882803 2, 0.001676761330466465996, 0.001674591225503952, 0.001664312923926880 6,
0.001651471834062667 7, 0.001641413833431304 4, 0.001638374327392333 5, 0.001645163902100788,
0.001662830001877835 3, 0.001690342868164163, 0.001724388185521230 6, 0.001759363691256732 3,

```
0.0017876634615882O7, 0.0018002904180212ll, 0.0017877726184923892, 0.0017412874748315987,
0.0016538389137925602, 0.0015213022932429876, 0.0013431601070782l5, 0.0011227977385188659,
0.0008673026054408783, 0.0005867951567134488, 0.0002933975783566689]

TEST_N_32_W_1_00_RESIDUAL = 0.009803151513324613

TEST_N_32_W_1_00_ITERATIONS = 44
```

w = 1.05

```
TEST_N_32_W_1_05_RESIDUAL = 0.00981454626829747

TEST_N_32_W_1_05_ITERATIONS = 44
```

w = 1.10

```
TEST_N_32_W_1_10_RESIDUAL = 0.009830136048462598

TEST_N_32_W_1_10_ITERATIONS = 44
```

w = 1.15

```
TEST_N_32_W_1_15_RESIDUAL = 0.00984148161611429

TEST_N_32_W_1_15_ITERATIONS = 44
```

w = 1.20

```
TEST_N_32_W_1_20_RESIDUAL = 0.009836943330840557

TEST_N_32_W_1_20_ITERATIONS = 44
```

w = 1.25

```
TEST_N_32_W_1_25_RESIDUAL = 0.009801941170213371

TEST_N_32_W_1_25_ITERATIONS = 44
```

w = 1.30

```
TEST_N_32_W_1_30_RESIDUAL = 0.009969969125522355

TEST_N_32_W_1_30_ITERATIONS = 43
```

w = 1.35

```
TEST_N_32_W_1_35_RESIDUAL = 0.00983276533436991

TEST_N_32_W_1_35_ITERATIONS = 43
```

w = 1.40

```
TEST_N_32_W_1_40_RESIDUAL = 0.009895482472228398

TEST_N_32_W_1_40_ITERATIONS = 42
```

w = 1.45

```
TEST_N_32_W_1_45_RESIDUAL = 0.009890417350052092

TEST_N_32_W_1_45_ITERATIONS = 41
```

w = 1.50

```
TEST_N_32_W_1_50_RESIDUAL = 0.009796978850527127

TEST_N_32_W_1_50_ITERATIONS = 40
```

w = 1.55

```
TEST_N_32_W_1_55_RESIDUAL = 0.00997039942101104

TEST_N_32_W_1_55_ITERATIONS = 38
```

w = 1.60

```
TEST_N_32_W_1_60_RESIDUAL = 0.009651585496575982

TEST_N_32_W_1_60_ITERATIONS = 37
```

w = 1.65

```
TEST_N_32_W_1_65_RESIDUAL = 0.009617600848207787

TEST_N_32_W_1_65_ITERATIONS = 35
```

w = 1.70

```
TEST_N_32_W_1_70_RESIDUAL = 0.009458970731247194

TEST_N_32_W_1_70_ITERATIONS = 33
```

w = 1.75

```
TEST_N_32_W_1_75_RESIDUAL = 0.00981032804717245

TEST_N_32_W_1_75_ITERATIONS = 30
```

w = 1.80

```
TEST_N_32_W_1_80_RESIDUAL = 0.009934669995474304

TEST_N_32_W_1_80_ITERATIONS = 28
```

w = 1.8214651907890225

```
TEST_N_32_W_1_82_RESIDUAL = 0.009862712316656284

TEST_N_32_W_1_82_ITERATIONS = 28
```

w = 1.85

```
TEST_N_32_W_1_85_RESIDUAL = 0.00939904098616889

TEST_N_32_W_1_85_ITERATIONS = 30
```

w = 1.90

```
TEST_N_32_W_1_90_RESIDUAL = 0.009899670352999799

TEST_N_32_W_1_90_ITERATIONS = 44
```

w = 1.95

```
TEST_N_32_W_1_95_RESIDUAL = 0.008860612644302752

TEST_N_32_W_1_95_ITERATIONS = 68
```