

# Tarea 6-Métodos numéricos

1<sup>st</sup> Daniel Vallejo Aldana  
Maestría en Ciencias de la Computación  
Centro de Investigación en Matemáticas  
daniel.vallejo@cimat.mx

**Resumen**—En el presente trabajo se estudia y se implementa el método de Jacobi para encontrar los valores y vectores propios de una matriz simétrica  $A$ , donde  $A$  es la matriz asociada al problema parabólico de la ecuación de calor.

**Index Terms**—Método de Jacobi, Valores propios, Vectores propios

## I. INTRODUCCIÓN

Como fue mencionado en la Tarea 2, encontrar valores y vectores propios es un problema que puede ser resuelto con diferentes aproximaciones, tales como el método de la potencia o el método de la potencia inversa descritos en la Tarea 2. En el presente trabajo discutiremos el método de Jacobi, basado en realizar operaciones de rotación a una matriz simétrica  $A$ , en la sección de resultados discutiremos aspectos de la implementación tales como tiempo de ejecución y errores obtenidos por el método. Así mismo mostraremos el uso del método de Jacobi para la obtención de los valores y vectores propios de la matriz asociada al problema parabólico de la ecuación de calor.

## II. MÉTODO/ALGORITMO

El método de Jacobi es un método iterativo usado para calcular los valores y vectores propios de una matriz  $A$ , donde  $A$  es simétrica con  $a_{i,j} \in \mathbb{R}$ . La actualización del método de Jacobi sobre la matriz  $A$  se realiza de la siguiente forma [1]

$$A' = J_{p,q}^T A J_{p,q}$$

Donde definimos  $J_{p,q}$  de la siguiente forma

$$J_{p,q} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cos(\theta) & \dots & \sin(\theta) & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & -\sin(\theta) & \dots & \cos(\theta) & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \dots & 1 \end{bmatrix}$$

Donde  $J_{p,q}$  es la matriz identidad salvo en las entradas  $(p,p), (q,q), (q,p), (p,q)$  cuyos valores son como se ilustra en la matriz anterior.

Al realizar la actualización de la matriz  $A$  mediante rotaciones de Jacobi podemos ver que las entradas de la nueva matriz tienen los siguientes valores

$$\begin{aligned} a'_{i,j} &= a_{i,j} \forall i, j \neq p, q \\ a'_{i,p} &= a'_{p,i} = \cos(\theta)a_{p,i} - \sin(\theta)a_{q,i} \forall i \neq p \\ a'_{i,q} &= a'_{q,i} = \sin(\theta)a_{p,i} + \cos(\theta)a_{q,i} \forall i \neq p \\ a'_{p,p} &= \cos^2(\theta)a_{p,p} - 2\sin(\theta)\cos(\theta)a_{p,q} + \sin^2(\theta)a_{q,q} \\ a'_{q,q} &= \sin^2(\theta)a_{p,p} + 2\sin(\theta)\cos(\theta)a_{p,q} + \cos^2(\theta)a_{q,q} \\ a'_{q,p} &= a'_{p,q} = \sin(\theta)\cos(\theta)(a_{p,p} - a_{q,q}) + (\cos^2(\theta) - \sin^2(\theta))a_{p,q} \end{aligned}$$

Donde

$$\theta = \frac{1}{2} \operatorname{atan} \left( \frac{2a_{p,q}}{a_{q,q} - a_{p,p}} \right)$$

De igual forma, consideremos  $V = I$  la matriz identidad de  $n \times n$ , entonces la actualización de la matriz  $V$  en cada una de las rotaciones de Jacobi es la siguiente [1]

$$\begin{aligned} v_{i,j} &= v_{i,j} \neq q, p \\ v_{i,p} &= \cos(\theta)v_{i,p} - \sin(\theta)v_{i,q} \\ v_{i,q} &= \sin(\theta)v_{i,p} + \cos(\theta)v_{i,q} \end{aligned}$$

Lo anterior equivale a hacer rotaciones por la derecha de la matriz identidad, lo anterior lo podemos expresar de la siguiente manera

$$V' = V J_{p,q}$$

Para la elección de los valores  $p, q$  seguimos un enfoque clásico en donde  $p, q$  son escogidos de la siguiente forma

$$p, q = \operatorname{argmax}_{i \in \text{Rows}, j \in \text{Cols} \mid j > i} |a_{i,j}|$$

Es decir, representa la entrada fuera de la diagonal que sea más grande en valor absoluto.

El criterio de paro del algoritmo escogido para esta implementación es la suma de los elementos fuera de la diagonal en valor absoluto, tales que su suma sea menor que una tolerancia dada, en este caso escogemos la tolerancia como  $1 \times 10^{-8}$ . Podemos escribir el error

considerado para el criterio de paro del algoritmo de la siguiente manera

$$\nu = \sum_{i=0}^{n-1} \sum_{j=1}^n |a_{i,j}|$$

### III. RESULTADOS

Para los experimentos realizados en este trabajo consideramos matrices de la forma

$$\begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & 0 \\ \vdots & 0 & \ddots & \dots & 0 \\ \dots & \dots & \dots & 1 & -2 \end{bmatrix}$$

Para el primer experimento consideramos una matriz de dimensión  $8 \times 8$

Los primeros valores propios asociados a dicha matriz son  $-3,87939, -3,53209, -3$ , mientras que los tres más pequeños son  $-0,120615, -0,467911, -1$ , notemos que dichas salidas concuerdan con las obtenidas por WolframAlpha como se muestra a continuación.

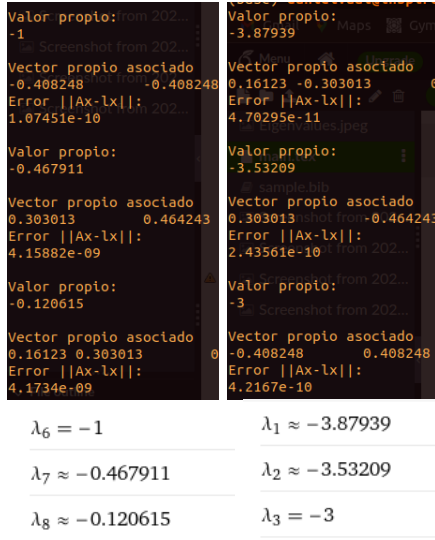


Figura 1. Valores propios más pequeños (izquierda) y más grandes (derecha) calculados con el método de Jacobi (primer renglón) y con WolframAlpha (Segundo renglón)

Así mismo en la figura 2 podemos ver los valores propios acomodados de mayor a menor en cuanto a magnitud

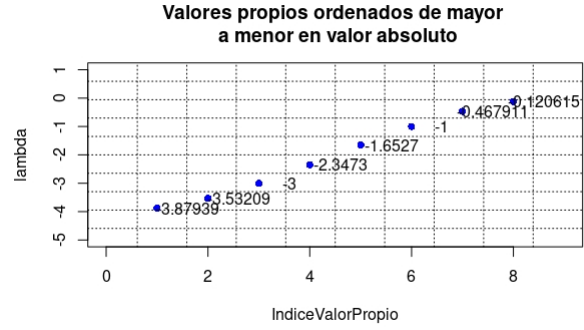


Figura 2. Valores propios obtenidos con el método de Jacobi

Como podemos ver en la figura 2, los errores de la forma  $\|A\mathbf{v}_i - \lambda_i \mathbf{v}_i\|$  son del orden de  $10^{-10}$  lo cual nos dice que dichos vectores propios encontrados por el algoritmo son igual correctos.

#### III-A. Resultado sobre una matriz de 1000 nodos

Para este experimento consideremos una matriz como la previamente descrita, considerando esta vez 1000 nodos. En este caso reportaremos el valor propio más grande, así como el valor propio más pequeño, para el error de los vectores propios reportaremos un resumen de dichos errores. La captura de pantalla donde se muestra la funcionalidad del programa se encuentra en el Apéndice A.

MinMax de valores propios	
$\lambda_{max}$	$\lambda_{min}$
-3.999	$-9,8499e^{-06}$

Cuadro I

VALORES PROPIOS MÁXIMOS Y MÍNIMOS ENCONTRADOS POR EL MÉTODO DE JACOBI PARA UNA MATRIZ DE 1000 NODOS

Para reportar los errores en cuanto a los vectores propios tenemos la siguiente tabla

Errores $\ A\mathbf{x} - \lambda\mathbf{x}\ $			
Min	Mediana	Media	Max
$1,113e^{-07}$	$1,493e^{-07}$	$1,496e^{-07}$	$2,259e^{-07}$

Cuadro II

RESUMEN DE LOS ERRORES OBTENIDOS AL CALCULAR LOS VECTORES PROPIOS DE LA MATRIZ DE 1000 NODOS

A continuación se adjunta un box-plot de los errores obtenidos al calcular los vectores propios de la matriz con 1000 nodos.

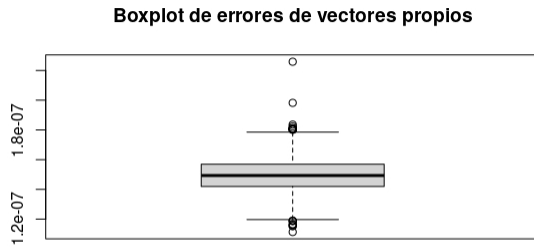


Figura 3. Boxplot de errores obtenidos en el cálculo de los vectores propios

En cuanto al tiempo de ejecución, el cálculo de los 1000 valores y vectores propios con una tolerancia de  $1e^{-3}$  fue de  $60 >$  minutos, lo cual lo convierte en un método ineficiente. No obstante, la ventaja es que se pueden encontrar todos los vectores propios y valores propios sin necesidad de quitar contribuciones.

#### IV. CONCLUSIONES

Con base en los resultados obtenidos en los experimentos realizados en el presente trabajo podemos concluir que el método de Jacobi es eficaz para encontrar vectores y valores propios de una matriz simétrica y real, no obstante carece de eficiencia al hacerlo cuando se trata de una matriz cuyas dimensiones son grandes, ya que tiene que llevar a cabo muchas operaciones de rotación lo cual aumenta el costo computacional de forma drástica.

#### REFERENCIAS

- [1] Jacobi method. <https://phys.au.dk/~fedorov/Numeric/now/Book/eigen.pdf>. Accessed: 2022-09-17.

#### V. APÉNDICE A

Adjuntamos algunas capturas de pantalla para ilustrar el funcionamiento de los métodos implementados.

```
Error [Jacobi]:
1.78076e-07

Valor propio:
-9.8499e-06

Vector propio asociado:
-0.00044023 -0.00020051 -0.000420077 -0.00056095 -0.000700106 -0.000840114 -0.000900115 -0.00112012 -0.00126011 -0.001
0009 -0.00154006 -0.00168 -0.00181993 -0.00195989 -0.00209977 -0.00223965 -0.00237951 -0.00251936 -0.00265915 -
0.00279894 -0.00293869 -0.00307843 -0.00321813 -0.0033578 -0.00349743 -0.00363702 -0.00377657 -0.00391608 -0.004
0556 -0.004195 -0.0043344 -0.00447377 -0.0046131 -0.00475239 -0.00489155 -0.00503085 -0.00517004 -0.00530912 -
0.00544818 -0.0055871 -0.0057261 -0.00586498 -0.00600381 -0.00614257 -0.00628127 -0.00641992 -0.00655849 -0.006
69699 -0.00683543 -0.00697302 -0.00711213 -0.00725037 -0.00738855 -0.00752664 -0.00766465 -0.00780259 -0.00794045 -
0.00807824 -0.00821594 -0.00835354 -0.00849109 -0.00862854 -0.00876591 -0.00890319 -0.00904037 -0.00917746 -0.009
31447 -0.00945137 -0.00958819 -0.00972492 -0.00986154 -0.00999807 -0.0101345 -0.0102709 -0.0104071 -0.0105432 -
0.0106793 -0.0108152 -0.0109511 -0.0110868 -0.0112224 -0.0113578 -0.0114933 -0.0116286 -0.0117637 -0.011898
0.0120337 -0.0121685 -0.0123032 -0.0124378 -0.0125722 -0.0127065 -0.0128407 -0.0129748 -0.0131087 -0.0132425
0.0133762 -0.0135098 -0.0136432 -0.0137765 -0.0139097 -0.0140427 -0.0141756 -0.0143084 -0.014441 -0.0145735
-0.0147059 -0.0148381 -0.0149701 -0.0151021 -0.0152338 -0.0153655 -0.0154967 -0.0156283 -0.0157594 -0.0158905
-0.0160213 -0.016152 -0.0162826 -0.016413 -0.0165432 -0.0166733 -0.0168032 -0.016932 -0.0170614 -0.0171906
-0.0173192 -0.0174481 -0.0175769 -0.0177056 -0.0178343 -0.017963 -0.0180917 -0.0182203 -0.0183489 -0.0184775
-0.0186061 -0.0187346 -0.0188631 -0.0189916 -0.0191201 -0.0192486 -0.0193771 -0.0195056 -0.0196341 -0.0197626
-0.0198911 -0.0200196 -0.0201481 -0.0202766 -0.0204051 -0.0205336 -0.0206621 -0.0207906 -0.0209191 -0.0210476
-0.0211761 -0.0213046 -0.0214331 -0.0215616 -0.0216901 -0.0218186 -0.0219471 -0.0220756 -0.0222041 -0.0223326
-0.0224611 -0.0225896 -0.0227181 -0.0228466 -0.0229751 -0.0231036 -0.0232321 -0.0233606 -0.0234891 -0.0236176
-0.0237461 -0.0238746 -0.0240031 -0.0241316 -0.0242601 -0.0243886 -0.0245171 -0.0246456 -0.0247741 -0.0249026
-0.0250311 -0.0251596 -0.0252881 -0.0254166 -0.0255451 -0.0256736 -0.0258021 -0.0259306 -0.0260591 -0.0261876
-0.0263161 -0.0264446 -0.0265731 -0.0267016 -0.0268301 -0.0269586 -0.0270871 -0.0272156 -0.0273441 -0.0274726
-0.0276011 -0.0277296 -0.0278581 -0.0279866 -0.0281151 -0.0282436 -0.0283721 -0.0285006 -0.0286291 -0.0287576
-0.0288861 -0.0290146 -0.0291431 -0.0292716 -0.0294001 -0.0295286 -0.0296571 -0.0297856 -0.0299141 -0.0300426
-0.0301711 -0.0303096 -0.0304381 -0.0305666 -0.0306951 -0.0308236 -0.0309521 -0.0310806 -0.0312091 -0.0313376
-0.0314661 -0.0315946 -0.0317231 -0.0318516 -0.0319801 -0.0321086 -0.0322371 -0.0323656 -0.0324941 -0.0326226
-0.0327511 -0.0328796 -0.0330081 -0.0331366 -0.0332651 -0.0333936 -0.0335221 -0.0336506 -0.0337791 -0.0339076
-0.0340361 -0.0341646 -0.0342931 -0.0344216 -0.0345501 -0.0346786 -0.0348071 -0.0349356 -0.0350641 -0.0351926
-0.0353211 -0.0354496 -0.0355781 -0.0357066 -0.0358351 -0.0359636 -0.0360921 -0.0362206 -0.0363491 -0.0364776
-0.0366061 -0.0367346 -0.0368631 -0.0369916 -0.0371201 -0.0372486 -0.0373771 -0.0375056 -0.0376341 -0.0377626
-0.0378911 -0.0380196 -0.0381481 -0.0382766 -0.0384051 -0.0385336 -0.0386621 -0.0387906 -0.0389191 -0.0390476
-0.0391761 -0.0393046 -0.0394331 -0.0395616 -0.0396901 -0.0398186 -0.0399471 -0.0400756 -0.0402041 -0.0403326
-0.0404611 -0.0405896 -0.0407181 -0.0408466 -0.0409751 -0.0411036 -0.0412321 -0.0413606 -0.0414891 -0.0416176
-0.0417461 -0.0418746 -0.0420031 -0.0421316 -0.0422601 -0.0423886 -0.0425171 -0.0426456 -0.0427741 -0.0429026
-0.0430311 -0.0431596 -0.0432881 -0.0434166 -0.0435451 -0.0436736 -0.0438021 -0.0439306 -0.0440591 -0.0441876
-0.0443161 -0.0444446 -0.0445731 -0.0447016 -0.0448301 -0.0449586 -0.0450871 -0.0452156 -0.0453441 -0.0454726
-0.0456011 -0.0457296 -0.0458581 -0.0459866 -0.0461151 -0.0462436 -0.0463721 -0.0465006 -0.0466291 -0.0467576
-0.0468861 -0.0470146 -0.0471431 -0.0472716 -0.0474001 -0.0475286 -0.0476571 -0.0477856 -0.0479141 -0.0480426
-0.0481711 -0.0483096 -0.0484381 -0.0485666 -0.0486951 -0.0488236 -0.0489521 -0.0490806 -0.0492091 -0.0493376
-0.0494661 -0.0495946 -0.0497231 -0.0498516 -0.0499801 -0.0501086 -0.0502371 -0.0503656 -0.0504941 -0.0506226
-0.0507511 -0.0508796 -0.0510081 -0.0511366 -0.0512651 -0.0513936 -0.0515221 -0.0516506 -0.0517791 -0.0519076
-0.0520361 -0.0521646 -0.0522931 -0.0524216 -0.0525501 -0.0526786 -0.0528071 -0.0529356 -0.0530641 -0.0531926
-0.0533211 -0.0534496 -0.0535781 -0.0537066 -0.0538351 -0.0539636 -0.0540921 -0.0542206 -0.0543491 -0.0544776
-0.0546061 -0.0547346 -0.0548631 -0.0549916 -0.0551201 -0.0552486 -0.0553771 -0.0555056 -0.0556341 -0.0557626
-0.0558911 -0.0560196 -0.0561481 -0.0562766 -0.0564051 -0.0565336 -0.0566621 -0.0567906 -0.0569191 -0.0570476
-0.0571761 -0.0573046 -0.0574331 -0.0575616 -0.0576901 -0.0578186 -0.0579471 -0.0580756 -0.0582041 -0.0583326
-0.0584611 -0.0585896 -0.0587181 -0.0588466 -0.0589751 -0.0591036 -0.0592321 -0.0593606 -0.0594891 -0.0596176
-0.0597461 -0.0598746 -0.0600031 -0.0601316 -0.0602601 -0.0603886 -0.0605171 -0.0606456 -0.0607741 -0.0609026
-0.0610311 -0.0611596 -0.0612881 -0.0614166 -0.0615451 -0.0616736 -0.0618021 -0.0619306 -0.0620591 -0.0621876
-0.0623161 -0.0624446 -0.0625731 -0.0627016 -0.0628301 -0.0629586 -0.0630871 -0.0632156 -0.0633441 -0.0634726
-0.0636011 -0.0637296 -0.0638581 -0.0639866 -0.0641151 -0.0642436 -0.0643721 -0.0645006 -0.0646291 -0.0647576
-0.0648861 -0.0650146 -0.0651431 -0.0652716 -0.0654001 -0.0655286 -0.0656571 -0.0657856 -0.0659141 -0.0660426
-0.0661711 -0.0663096 -0.0664381 -0.0665666 -0.0666951 -0.0668236 -0.0669521 -0.0670806 -0.0672091 -0.0673376
-0.0674661 -0.0675946 -0.0677231 -0.0678516 -0.0679801 -0.0681086 -0.0682371 -0.0683656 -0.0684941 -0.0686226
-0.0687511 -0.0688796 -0.0690081 -0.0691366 -0.0692651 -0.0693936 -0.0695221 -0.0696506 -0.0697791 -0.0699076
-0.0700361 -0.0701646 -0.0702931 -0.0704216 -0.0705501 -0.0706786 -0.0708071 -0.0709356 -0.0710641 -0.0711926
-0.0713211 -0.0714496 -0.0715781 -0.0717066 -0.0718351 -0.0719636 -0.0720921 -0.0722206 -0.0723491 -0.0724776
-0.0726061 -0.0727346 -0.0728631 -0.0729916 -0.0731201 -0.0732486 -0.0733771 -0.0735056 -0.0736341 -0.0737626
-0.0738911 -0.0740196 -0.0741481 -0.0742766 -0.0744051 -0.0745336 -0.0746621 -0.0747906 -0.0749191 -0.0750476
-0.0751761 -0.0753046 -0.0754331 -0.0755616 -0.0756901 -0.0758186 -0.0759471 -0.0760756 -0.0762041 -0.0763326
-0.0764611 -0.0765896 -0.0767181 -0.0768466 -0.0769751 -0.0771036 -0.0772321 -0.0773606 -0.0774891 -0.0776176
-0.0777461 -0.0778746 -0.0780031 -0.0781316 -0.0782601 -0.0783886 -0.0785171 -0.0786456 -0.0787741 -0.0789026
-0.0790311 -0.0791596 -0.0792881 -0.0794166 -0.0795451 -0.0796736 -0.0798021 -0.0799306 -0.0800591 -0.0801876
-0.0803161 -0.0804446 -0.0805731 -0.0807016 -0.0808301 -0.0809586 -0.0810871 -0.0812156 -0.0813441 -0.0814726
-0.0816011 -0.0817296 -0.0818581 -0.0819866 -0.0821151 -0.0822436 -0.0823721 -0.0825006 -0.0826291 -0.0827576
-0.0828861 -0.0830146 -0.0831431 -0.0832716 -0.0834001 -0.0835286 -0.0836571 -0.0837856 -0.0839141 -0.0840426
-0.0841711 -0.0843096 -0.0844381 -0.0845666 -0.0846951 -0.0848236 -0.0849521 -0.0850806 -0.0852091 -0.0853376
-0.0854661 -0.0855946 -0.0857231 -0.0858516 -0.0859801 -0.0861086 -0.0862371 -0.0863656 -0.0864941 -0.0866226
-0.0867511 -0.0868796 -0.0870081 -0.0871366 -0.0872651 -0.0873936 -0.0875221 -0.0876506 -0.0877791 -0.0879076
-0.0880361 -0.0881646 -0.0882931 -0.0884216 -0.0885501 -0.0886786 -0.0888071 -0.0889356 -0.0890641 -0.0891926
-0.0893211 -0.0894496 -0.0895781 -0.0897066 -0.0898351 -0.0899636 -0.0900921 -0.0902206 -0.0903491 -0.0904776
-0.0906061 -0.0907346 -0.0908631 -0.0909916 -0.0911201 -0.0912486 -0.0913771 -0.0915056 -0.0916341 -0.0917626
-0.0918911 -0.0920196 -0.0921481 -0.0922766 -0.0924051 -0.0925336 -0.0926621 -0.0927906 -0.0929191 -0.0930476
-0.0931761 -0.0933046 -0.0934331 -0.0935616 -0.0936901 -0.0938186 -0.0939471 -0.0940756 -0.0942041 -0.0943326
-0.0944611 -0.0945896 -0.0947181 -0.0948466 -0.0949751 -0.0951036 -0.0952321 -0.0953606 -0.0954891 -0.0956176
-0.0957461 -0.0958746 -0.0960031 -0.0961316 -0.0962601 -0.0963886 -0.0965171 -0.0966456 -0.0967741 -0.0969026
-0.0970311 -0.0971596 -0.0972881 -0.0974166 -0.0975451 -0.0976736 -0.0978021 -0.0979306 -0.0980591 -0.0981876
-0.0983161 -0.0984446 -0.0985731 -0.0987016 -0.0988301 -0.0989586 -0.0990871 -0.0992156 -0.0993441 -0.0994726
-0.0996011 -0.0997296 -0.0998581 -0.0999866 -0.1001151 -0.1002436 -0.1003721 -0.1005006 -0.1006291 -0.1007576
-0.1008861 -0.1010146 -0.1011431 -0.1012716 -0.1014001 -0.1015286 -0.1016571 -0.1017856 -0.1019141 -0.1020426
-0.1021711 -0.1023096 -0.1024381 -0.1025666 -0.1026951 -0.1028236 -0.1029521 -0.1030806 -0.1032091 -0.1033376
-0.1034661 -0.1035946 -0.1037231 -0.1038516 -0.1039801 -0.1041086 -0.1042371 -0.1043656 -0.1044941 -0.1046226
-0.1047511 -0.1048796 -0.1050081 -0.1051366 -0.1052651 -0.1053936 -0.1055221 -0.1056506 -0.1057791 -0.1059076
-0.1060361 -0.1061646 -0.1062931 -0.1064216 -0.1065501 -0.1066786 -0.1068071 -0.1069356 -0.1070641 -0.1071926
-0.1073211 -0.1074496 -0.1075781 -0.1077066 -0.1078351 -0.1079636 -0.1080921 -0.1082206 -0.1083491 -0.1084776
-0.1086061 -0.1087346 -0.1088631 -0.1089916 -0.1091201 -0.1092486 -0.1093771 -0.1095056 -0.1096341 -0.1097626
-0.1098911 -0.1100196 -0.1101481 -0.1102766 -0.1104051 -0.1105336 -0.1106621 -0.1107906 -0.1109191 -0.1110476
-0.1111761 -0.1113046 -0.1114331 -0.1115616 -0.1116901 -0.1118186 -0.1119471 -0.1120756 -0.1122041 -0.1123326
-0.1124611 -0.1125896 -0.1127181 -0.1128466 -0.1129751 -0.1131036 -0.1132321 -0.1133606 -0.1134891 -0.1136176
-0.1137461 -0.1138746 -0.1140031 -0.1141316 -0.1142601 -0.1143886 -0.1145171 -0.1146456 -0.1147741 -0.1149026
-0.1150311 -0.1151596 -0.1152881 -0.1154166 -0.1155451 -0.1156736 -0.1158021 -0.1159306 -0.1160591 -0.1161876
-0.1163161 -0.1164446 -0.1165731 -0.1167016 -0.1168301 -0.1169586 -0.1170871 -0.1172156 -0.1173441 -0.1174726
-0.1176011 -0.1177296 -0.1178581 -0.1179866 -0.1181151 -0.1182436 -0.1183721 -0.1185006 -0.1186291 -0.1187576
-0.1188861 -0.1190146 -0.1191431 -0.1192716 -0.1194001 -0.1195286 -0.1196571 -0.1197856 -0.1199141 -0.1200426
-0.1201711 -0.1203096 -0.1204381 -0.1205666 -0.1206951 -0.1208236 -0.1209521 -0.1210806 -0.1212091 -0.1213376
-0.1214661 -0.1215946 -0.1217231 -0.1218516 -0.1219801 -0.1221086 -0.1222371 -0.1223656 -0.1224941 -0.1226226
-0.1227511 -0.1228796 -0.1230081 -0.1231366 -0.1232651 -0.1233936 -0.1235221 -0.1236506 -0.1237791 -0.1239076
-0.1240361 -0.1241646 -0.1242931 -0.1244216 -0.1245501 -0.1246786 -0.1248071 -0.1249356 -0.1250641 -0.1251926
-0.1253211 -0.1254496 -0.1255781 -0.1257066 -0.1258351 -0.1259636 -0.1260921 -0.1262206 -0.1263491 -0.1264776
-0.1266061 -0.1267346 -0.1268631 -0.1269916 -0.1271201 -0.1272486 -0.1273771 -0.1275056 -0.1276341 -0.1277626
-0.1278911 -0.1280196 -0.1281481 -0.1282766 -0.1284051 -0.1285336 -0.1286621 -0.1287906 -0.1289191 -0.1290476
-0.1291761 -0.1293046 -0.1294331 -0.1295616 -0.1296901 -0.1298186 -0.1299471 -0.1300756 -0.1302041 -0.1303326
-0.1304611 -0.1305896 -0.1307181 -0.1308466 -0.1309751 -0.1311036 -0.1312321 -0.1313606 -0.1314891 -0.1316176
-0.1317461 -0.1318746 -0.1320031 -0.1321316 -0.1322601 -0.1323886 -0.1325171 -0.1326456 -0.1327741 -0.1329026
-0.1330311 -0.1331596 -0.1332881 -0.1334166 -0.1335451 -0.1336736 -0.1338021 -0.1339306 -0.1340591 -0.1341876
-0.1343161 -0.1344446 -0.1345731 -0.1347016 -0.1348301 -0.1349586 -0.1350871 -0.1352156 -0.1353441 -0.1354726
-0.1356011 -0.1357296 -0.1358581 -0.1359866 -0.1361151 -0.1362436 -0.1363721 -0.1365006 -0.1366291 -0.1367576
-0.1368861 -0.1370146 -0.1371431 -0.1372716 -0.1374001 -0.1375286 -0.1376571 -0.1377856 -0.1379141 -0.1380426
-0.1381711 -0.1383096 -0.1384381 -0.1385666 -0.1386951 -0.1388236 -0.1389521 -0.1390806 -0.1392091 -0.1393376
-0.1394661 -0.1395946 -0.1397231 -0.1398516 -0.1399801 -0.1401086 -0.1402371 -0.1403656 -0.1404941 -0.1406226
-0.1407511 -0.1408796 -0.1410081 -0.1411366 -0.1412651 -0.1413936 -0.1415221 -0.1416506 -0.1417791 -0.1419076
-0.1420361 -0.1421646 -0.1422931 -0.1424216 -0.1425501 -0.1426786 -0.1428071 -0.1429356 -0.1430641 -0.1431926
-0.1433211 -0.1434496 -0.1435781 -0.1437066 -0.1438351 -0.1439636 -0.1440921 -0.1442206 -0.1443491 -0.1444776
-0.1446061 -0.1447346 -0.1448631 -0.1449916 -0.1451201 -0.1452486 -0.1453771 -0.1455056 -0.1456341 -0.1457626
-0.1458911 -0.1460196 -0.1461481 -0.1462766 -0.1464051 -0.1465336 -0.1466621 -0.1467906 -0.1469191 -0.1470476
-0.1471761 -0.1473046 -0.1474331 -0.1475616 -0.1476901 -0.147818
```