IEOR160 Project

Dhrushil Badani, Dhruvil Badani, Ravneet Madaan, Shreyas Bhave

April 2017

Contents

1 Introduction

In this project, we aim to create a linear regression model to calculate the disease progression for diabetes from patients' baseline measurements. We have 10 features to choose from, but restrict ourselves to 4 for a more interpretable model with better predictive value. In Part I, we optimize the selection of the best subset of 4 features, using statistics like the p-value and R^2 value. We compare different methods like heuristic-based selection, Lasso regularization and mixed-integer programming (for a subset of size 4) for this subset selection. In Part 2, we consider second-order interactions, and aim to choose 10 independent variables out of the 65 possible second-order variables. We use the same statistics (from Part 1) like the p-value and R^2 value, and also compare different methods like Lasso regularization and mixed-integer programming. The criterion we use to decide the better method in both parts is the out-of-sample accuracy.

2 Part 1

2a Four Most Significant Variables

The four most significant variables are BMI, S5, Sex, BP in order of increasing p-value. Each of these variables has a p-value below 0.05 which is a widely used significance level. This p-value is testing the null hypothesis that the coefficient is equal to 0 (there is no relation between the given variable and response variable). Thus, the smaller the p-value the more confidence we have in rejecting the null hypothesis, which indicates that there is a significant relationship between the given variable and response variable. The code for this part shows the coefficients and statistics with just these four variables used.

Error	R^2
772097	0.455053

2b Best Combination of Four Independent Variables

With the given heuristic of using the variables with the five lowest p-values and using a subset of size four, we find that the subset that uses Sex, BMI, BP and S5 has the highest R^2 value as shown in the table below. The coefficients, SSE and R^2 value are all shown in the second table. These are the four features with the least p-value so it makes sense that they have the minimum SSE and R^2 value.

Subset of Variables	Error	R^2
$\{BMI, BP, S5, S6\}$	778372	0.450624
$\{Sex, BP, S5, S6\}$	874717	0.382623
$\{Sex, BMI, S5, S6\}$	774756	0.453176
$\{Sex, BMI, BP, S6\}$	862701	0.391104
$\{Sex, BMI, BP, S5\}$	772097	0.455053

Best Subset	Error	R^2	B_{sex}	B_{BMI}	B_{BP}	B_{S5}
$\{Sex, BMI, BP, S5\}$	772097	0.455053	-155.447	597.687	213.927	600.31

2c Lasso Regularization

The Lasso Regularization employs an L1 penalty to penalize for large coefficient values and to select features (drives coefficient of insignificant features to zero). The first table shows the error and corresponding R^2 values for each of the different lambda values. As lambda increases, we penalize more and, thus, the error increases and R^2 decreases. This is because more weight is being placed on the penalty term in the loss function and less weight is being placed on minimizing the original least squares term. The tradeoff is that as the lambda term increases, the model becomes simpler as fewer variables are selected.

In this case, the minimum lambda value that yielded four features was 340 as shown in the first table. We would like the minimum lambda value because as lambda increases R^2 decreases, which results in a worse fit. In this case, BMI, BP, S5 and S6 were selected via lasso regularization. The coefficients are given in the second table.

The best value of lambda in terms of R^2 is obviously 200, the minimum lambda value but it results in five variables being selected when we want to restrict the model to encompassing four variables.

λ	Error	R^2	Number of Variables Selected
200	802793	0.433387	5
220	810566	0.427902	5
240	819078	0.421893	5
260	828331	0.415363	5
280	838324	0.40831	5
300	849057	0.400734	5
320	860531	0.392636	5
340	871720	0.384739	4
360	881977	0.377499	3
380	892820	0.369846	3
400	904250	0.361779	3

λ	340
Error	871720
R^2	0.384739
B_{AGE}	0
B_{sex}	0
B_{BMI}	405.494
B_{BP}	0.000540714
B_{S1}	0
B_{S2}	0
B_{S3}	0
B_{S4}	0
B_{S5}	414.866
B_{S6}	23.1162

2d Mixed-integer Optimization

Using Mixed-integer Optimization, the regression coefficients, error and R^2 are shown in the table below. This modeled as a mixed-integer optimization problem that uses at most 4 variables. In this case, BMI, BP, S1, and S5 are chosen. This subset did not appear under the heuristic method used in part b.

Error	755094
R^2	0.467054
B_{AGE}	0
B_{sex}	0
B_{BMI}	603.681
B_{BP}	200.878
B_{S1}	-267.699
B_{S2}	0
B_{S3}	0
B_{S4}	0
B_{S5}	711.644
B_{S6}	0

2e Comparison of Results

Looking at the accuracy of each method, using all 10 variables yielded the lowest out-of-sample error. This is likely because it gives the model better expressivity and captures the complexities of the linear relationship without overfitting the training data too much. However, the other methods result in a simpler model and do nearly as well. For example, the lowest four p-values and the heuristic selection method are a very close second place to using all 10 variables. These methods both result in the same error since they ended up choosing the same variables and the same coefficient values. The mixed integer optimization method is the next best, while lasso does the worst in terms of out-of-sample accuracy. Thus, simplifying the model did result in almost the same out-of-sample accuracy as using all 10 variables in the case of heuristic selection and lowest four p-value selection.

Method	Error
All 10 Variables	574367.96703
Lowest Four p-values	579396.855961
Heuristic Selection	579396.855961
Lasso Regularization	693913.938847
Mixed Integer Optimization	583706.734175

3 Part 2

3a Lasso with Second Order Interactions

We used a regularization parameter of 9.57 while implementing lasso regression with second order interactions. Using 9.57 as the regularization parameter we obtained the following values of the coefficients :

Coefficients for linear terms(β_i) =

1. -46.3591,
2. -226.708
3. 534.415
4. 230.201,
5. 316.992
6. 0
7. 12.5382
8. 204.877
9. 582.585
10. 160.929

Coefficients for quadratic terms (β_{jk}) =

0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0
.	.	0	0	0	0	0	0	0	0
.	.	.	0	0	0	0	0	0	0
.	.	.	.	0	0	0	0	0	0
.	.	.	.	.	0	0	0	0	0
.	.	.	.	.	.	0	0	0	0
.	.	.	.	.	.	.	0	0	0
.	.	.	.	.	.	.	.	0	0
.	.	.	.	.	.	.	.	.	102.204

We used the value of regularization parameter 9.57 as that enables us to select 10 independent variables out of the 65 possible variables available to us.

The value of R^2 for this problem = 0.498677

3b Mixed Integer Optimization with Second Order Interactions

Linear regression coefficients (β_i) =

1. 0
2. -233.553
3. 595.908
4. 263.431
5. -327.887
6. 0
7. 0
8. 242.18
9. 561.239
10. 121.949

Coefficients for quadratic terms (β_{jk}) =

$$\begin{array}{cccccccccc}
0 & 0 & 0 & 3540.81 & 0 & 0 & 0 & 0 & 0 & 0 \\
. & 0 & 4250.64 & 0 & 0 & 0 & -3850.05 & 0 & 0 & 0 \\
. & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
. & . & . & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
. & . & . & . & 0 & 0 & 0 & 0 & 0 & 0 \\
. & . & . & . & . & 0 & 0 & 0 & 0 & 0 \\
. & . & . & . & . & . & 0 & 0 & 0 & 0 \\
. & . & . & . & . & . & . & 0 & 0 & 0 \\
. & . & . & . & . & . & . & . & 0 & 0 \\
. & . & . & . & . & . & . & . & . & 0
\end{array}$$

The value of R^2 for this problem = 0.523882

3c Comparison of Results with Second Order Interactions

Out of sample error in part (f) = 573404.8630648409

Out of sample error in part(g) = 573285.4410441783

We observe that the mixed integer programming method of regression gives us a higher accuracy than the Lasso regression method when 10 independent variables are considered.

4 Conclusion

We find that in Part 1, when we were dealing with solely first-order interactions, using the given heuristic, we find that the four features with the lowest p-values (BMI, S5, Sex, BP) are also the ones that have the lowest SSE and R^2 values. Lasso regularization, with the ideal lambda (340), chose a subset of 4 features - BMI, BP, S5 and S6. When we tried mixed-integer programming, it chose a different subset of 4 features - BMI, BP, S1, and S5. Heuristic selection performed the best, with mixed-integer programming at a very close second place. In part 2, when we are considering second-order interactions, mixed integer programming outperforms Lasso regularization in terms of the out-of-sample error. One insight we gained from this is that Lasso regularization has a trade-off between the number of features it selects and the R^2 value - as we increase lambda, the model becomes simpler with the number of features reducing, but the R^2 value decreases too. Overall, the result in both parts is similar - mixed-integer programming outperforms Lasso regularization in both parts.

5 Appendix: The Code

Listing 1: 1a Code

```
option solver minos; # No integer programming here

##########################################################################

reset;

data reg_data.dat;

print "Looking at the p-values, we should use BMI, S5, Sex, BP"; # Features from
    p-values

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2; # We need to minimize sum of squared errors
subject to not_use_S1: coeffs[S1] = 0; # Use only the 4 features mentioned above
subject to not_use_S2: coeffs[S2] = 0; # Use only the 4 features mentioned above
subject to not_use_S3: coeffs[S3] = 0; # Use only the 4 features mentioned above
subject to not_use_S4: coeffs[S4] = 0; # Use only the 4 features mentioned above
subject to not_use_AGE: coeffs[AGE] = 0; # Use only the 4 features mentioned above
subject to not_use_S6: coeffs[S6] = 0; # Use only the 4 features mentioned above

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs; # Displaying results
```

Listing 2: 1b Code

```
option solver minos;

##########################################################################

reset;

data reg_data.dat;

print "Leaving out SEX";

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2; # We need to minimize the sum of squared errors
subject to not_use_S1: coeffs[S1] = 0; # Use only the features mentioned above
subject to not_use_S2: coeffs[S2] = 0; # Use only the features mentioned above
subject to not_use_S3: coeffs[S3] = 0; # Use only the features mentioned above
subject to not_use_S4: coeffs[S4] = 0; # Use only the features mentioned above
subject to not_use_AGE: coeffs[AGE] = 0; # Use only the features mentioned above
subject to not_use_SEX: coeffs[SEX] = 0; # Use only the features mentioned above

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs; # Displaying results

##########################################################################
```

```
reset;

data reg_data.dat;

print "Leaving out BMI";

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2; # We need to minimize the sum of squared errors
subject to not_use_S1: coeffs[S1] = 0; # Use only the features mentioned above
subject to not_use_S2: coeffs[S2] = 0; # Use only the features mentioned above
subject to not_use_S3: coeffs[S3] = 0; # Use only the features mentioned above
subject to not_use_S4: coeffs[S4] = 0; # Use only the features mentioned above
subject to not_use_AGE: coeffs[AGE] = 0; # Use only the features mentioned above
subject to not_use_BMI: coeffs[BMI] = 0; # Use only the features mentioned above

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs; # Displaying results

##########################################################################

reset;

data reg_data.dat;

print "Leaving out BP";
```

```
var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2; # We need to minimize the sum of squared errors
subject to not_use_S1: coeffs[S1] = 0; # Use only the features mentioned above
subject to not_use_S2: coeffs[S2] = 0; # Use only the features mentioned above
subject to not_use_S3: coeffs[S3] = 0; # Use only the features mentioned above
subject to not_use_S4: coeffs[S4] = 0; # Use only the features mentioned above
subject to not_use_AGE: coeffs[AGE] = 0; # Use only the features mentioned above
subject to not_use_BP: coeffs[BP] = 0; # Use only the features mentioned above

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs; # Displaying results

##########################################################################

reset;

data reg_data.dat;

print "Leaving out S5";

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
```

```
    mat[i, Y])^2; # We need to minimize the sum of squared errors
subject to not_use_S1: coeffs[S1] = 0; # Use only the features mentioned above
subject to not_use_S2: coeffs[S2] = 0; # Use only the features mentioned above
subject to not_use_S3: coeffs[S3] = 0; # Use only the features mentioned above
subject to not_use_S4: coeffs[S4] = 0; # Use only the features mentioned above
subject to not_use_AGE: coeffs[AGE] = 0; # Use only the features mentioned above
subject to not_use_S5: coeffs[S5] = 0; # Use only the features mentioned above

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs; # Displaying results

##########################################################################

reset;

data reg_data.dat;

print "Leaving out S6";

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2; # We need to minimize the sum of squared errors
subject to not_use_S1: coeffs[S1] = 0; # Use only the features mentioned above
subject to not_use_S2: coeffs[S2] = 0; # Use only the features mentioned above
subject to not_use_S3: coeffs[S3] = 0; # Use only the features mentioned above
subject to not_use_S4: coeffs[S4] = 0; # Use only the features mentioned above
```

```
subject to not_use_AGE: coeffs[AGE] = 0; # Use only the features mentioned above
subject to not_use_S6: coeffs[S6] = 0; # Use only the features mentioned above

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs; # Displaying results
```

Listing 3: 1c Code

```
reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[1]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[1]*z[k]; # We need to minimize the
    sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty
```

```
solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

##############################################################################

reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[2]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[2]*z[k]; # We need to minimize the
    sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty

solve;
```

```
var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

##############################################################################

reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[3]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[3]*z[k]; # We need to minimize the
    sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
```

```
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

#############################################################################

reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[4]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[4]*z[k]; # We need to minimize the
    sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
```

```
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

##############################################################################

reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[5]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[5]*z[k]; # We need to minimize the
    sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
```

```
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

############################################################################

reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[6]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[6]*z[k]; # We need to minimize the
    sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared
```

```
display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

#############################################################################

reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[7]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i, Y])^2 + sum{k in 1..n_features}lambdas[7]*z[k]; # We need to minimize the sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i, Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;
```

```
################################################################################

reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[8]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i, Y])^2 + sum{k in 1..n_features}lambdas[8]*z[k]; # We need to minimize the sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i, Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

################################################################################
```

```
reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[9]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[9]*z[k]; # We need to minimize the
    sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

##############################################################################

reset;
reset options;
```

```
option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[n_features]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[n_features]*z[k]; # We need to
    minimize the sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

##############################################################################

reset;
reset options;

option solver cplex;
```

```
data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var z{1..n_features}; # z[i] is the absolute value of coeffs[i]

print "Lambda =", lambdas[11]; # Setting lambda

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] -
    mat[i, Y])^2 + sum{k in 1..n_features}lambdas[11]*z[k]; # We need to minimize the
    sum of squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z[j] >= coeffs[j]; # To implement the absolute
    value l1-penalty
subject to abs_2 {j in 1..n_features}: z[j] >= -coeffs[j]; # To implement the
    absolute value l1-penalty

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display sse, error, y_hat, ss_tot, r_sq, coeffs, z;

#############################################################################
```

Listing 4: 1d Code

```
reset;
reset options;

data reg_data.dat;
```

```
var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var used{1..n_features} binary; # used[i] is 1 if the ith feature is used, 0 otherwise

minimize sse: sum{i in 1..n_train}(mat[i,Y] - sum{j in
    1..n_features}(coeffs[j]*mat[i,j]))^2; # We need to minimize the sum of squared
    errors
subject to sum_of_coeffs_constraint: sum{j in 1..n_features}used[j] <= 4; # Can use
    at most 4 features
subject to binary1{j in 1..n_features}: coeffs[j] <= first_order_coeff_bound*used[j];
    # To make sure that coeffs[j] is not nonzero when used[j] is zero
subject to binary2{j in 1..n_features}: coeffs[j] >=
    -first_order_coeff_bound*used[j]; # To make sure that coeffs[j] is not nonzero
    when used[j] is zero

option solver cplex;

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}coeffs[j]*mat[i,j] - mat[i,
    Y])^2; # The SSE with coeffs
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
    calculation
var r_sq = 1 - error/ss_tot; # Formula for r-squared

display r_sq, coeffs, used;
```

Listing 5: 1e Code

```
n_features = 10 # The number of features
n_train = 250 # The number of training samples

def get_predictions(beta, X):
   Y_pred = [] # List to store the predicitions
   for x in X: # Going through all the data
      y_pred = 0
```

```
        for i in range(n_features):
            y_pred += beta[i] * x[i] # simulating the dot product
        Y_pred.append(y_pred) # Adding the prediction to the list
    return Y_pred # Returning the list

def sse(beta, X, Y_actual):
    Y_pred = get_predictions(beta, X) # Getting the predictions
    squared_differences = [(Y_actual[i] - Y_pred[i])**2 for i in range(len(Y_pred))] #
        Calculating the sum of squared errors
    return sum(squared_differences) # Returning the SSE

def read_data(filename):
    X = [] # List to store all the data points
    Y = [] # List to store all the target variables
    with open(filename) as f: # Opening the file containing the data (NOT necessarily
        .dat)
        lines = f.readlines() # Reading all the lines in the file
        lines = lines[n_train + 1:] # Skipping the first n_train + 1 lines since we
            need out-of-sample error and the first line is just feature names
        for line in lines: # Going through all lines
            if "AGE" in line: # Checking if the word "AGE" is in the line
                continue # Skipping if the word "AGE" is in the line because if it is, the
                    line is just feature names, not an actual sample
            numbers = line.split() # Splitting the string
            numbers = [float(number) for number in numbers] # Converting strings to
                numbers
            y = numbers[-1] # Extracting the target variable
            x = numbers[0:-1] # Extracting the features
            X.append(x) # Adding the sample
            Y.append(y) # Adding the target variable
    return X, Y # returning the data points and target variables

X, Y_actual = read_data('data.txt') # Getting the data points as X and target
    variables as Y
# X[i] is the ith sample. X[i][j] is the jth feature of the ith sample
# Y[i] is the ith target variable
```

```
coeffs_given = [-59.6, -241.6, 535.1, 241.7, -844.9, 407.4, -224.3, 285.2, 762.4,
    169.6] # The coefficients given to us
print("SSE All 10: {0}".format(sse(coeffs_given, X, Y_actual))) # The SSE using the
    coefficients given

coeffs_part_a = [0, -155.447, 597.687, 213.927, 0, 0, 0, 0, 600.31, 0] # The
    coefficients from part (a)
print("SSE Part 1A: {0}".format(sse(coeffs_part_a, X, Y_actual))) # The SSE using the
    coefficients from part (a)

coeffs_part_b = [0, -155.447, 597.687, 213.927, 0, 0, 0, 0, 600.31, 0] # The
    coefficients from part (b)
print("SSE Part 1B: {0}".format(sse(coeffs_part_b, X, Y_actual))) # The SSE using the
    coefficients from part (b)

coeffs_part_c = [0, 0, 405.494, 0.000540696, 0, 0, 0, 0, 414.866, 23.1162] # The
    coefficients from part (c)
print("SSE Part 1C: {0}".format(sse(coeffs_part_c, X, Y_actual))) # The SSE using the
    coefficients from part (c)

coeffs_part_d = [0, 0, 603.681, 200.878, -267.699, 0, 0, 0, 711.644, 0] # The
    coefficients from part (d)
print("SSE Part 1D: {0}".format(sse(coeffs_part_d, X, Y_actual))) # The SSE using the
    coefficients from part (d)
```

Listing 6: 2f Code

```
reset;
reset options;

option solver cplex;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var coeffs_second_order{j in 1..n_features, k in j..n_features}; # coeffs[i,j] is the
    regression coefficient for the second order interaction between the ith and jth
```

```
    feature
var z_coeffs{1..n_features}; # z[i] is the absolute value of coeffs[i]
var z_coeffs_second_order{j in 1..n_features, k in j..n_features}; # z[i,j] is the
    absolute value of coeffs_second_order[i,j]

print "Lambda =", lambda_part_2;

minimize sse: sum{i in 1..n_train} (sum{j in 1..n_features}(coeffs[j]*mat[i,j]) +
    sum{j in 1..n_features, k in j..n_features}(mat[i, j] * mat[i, k] *
    coeffs_second_order[j,k]) - mat[i, Y])**2 + sum{k in
    1..n_features}lambda_part_2*z_coeffs[k] + sum{j in 1..n_features, k in
    j..n_features} lambda_part_2*z_coeffs_second_order[j,k]; # We need to minimize
    the sum of the squared errors + the regularization penalty
subject to abs_1 {j in 1..n_features}: z_coeffs[j] >= coeffs[j]; # To implement
    absolute value function
subject to abs_2 {j in 1..n_features}: z_coeffs[j] >= -coeffs[j]; # To implement
    absolute value function
subject to abs_3{j in 1..n_features, k in j..n_features}: z_coeffs_second_order[j,k]
    >= coeffs_second_order[j,k]; # To implement absolute value function
subject to abs_4{j in 1..n_features, k in j..n_features}: z_coeffs_second_order[j,k]
    >= -coeffs_second_order[j,k]; # To implement absolute value function
solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}(coeffs[j]*mat[i,j]) + sum{j
    in 1..n_features, k in j..n_features}(mat[i, j] * mat[i, k] *
    coeffs_second_order[j,k]) - mat[i, Y])^2; # The SSE with coeffs and
    coeffs_second_order
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
var r_sq = 1 - error/ss_tot; # r-squared
var adj_r_sq = r_sq - (1-r_sq)*(2/247);

display sse, error, y_hat, ss_tot, r_sq, coeffs, coeffs_second_order;
```

Listing 7: 2g Code

```
reset;
```

```
reset options;

data reg_data.dat;

var coeffs{1..n_features}; # coeffs[i] is the regression coefficient for the ith
    feature
var coeffs_second_order{j in 1..n_features, k in j..n_features}; # coeffs[i,j] is the
    regression coefficient for the second order interaction between the ith and jth
    feature
var used{1..n_features} binary; # used[i] is 1 if the ith feature is used, 0 otherwise
var used_second_order{j in 1..n_features, k in j..n_features} binary; #
    used_second_order[i,j] is 1 if the second order interaction between the ith and
    jth feature is used, 0 otherwise

minimize sse: sum{i in 1..n_train}(mat[i,Y] - sum{j in
    1..n_features}(coeffs[j]*mat[i,j]) - sum{j in 1..n_features, k in
    j..n_features}(mat[i,j]*mat[i,k]*coeffs_second_order[j,k]))^2; # We need to
    minimize the sum of squared errors
subject to sum_of_coeffs_constraint: sum{j in 1..n_features}used[j] + sum{j in
    1..n_features, k in j..n_features} used_second_order[j,k] = n_features; # Use at
    most n_features features
subject to binary1{j in 1..n_features}: coeffs[j] <= first_order_coeff_bound*used[j];
    # To make sure that coeffs[i] is not nonzero when used[i] is 0
subject to binary2{j in 1..n_features}: coeffs[j] >=
    -first_order_coeff_bound*used[j]; # To make sure that coeffs[i] is not nonzero
    when used[i] is 0
subject to binary3{j in 1..n_features, k in j..n_features}: coeffs_second_order[j,k]
    <= second_order_coeff_bound*used_second_order[j,k]; # To make sure that
    coeffs[i,j] is not nonzero when used[i,j] is 0
subject to binary4{j in 1..n_features, k in j..n_features}: coeffs_second_order[j,k]
    >= -second_order_coeff_bound*used_second_order[j,k];
option solver cplex; # To make sure that coeffs[i,j] is not nonzero when used[i,j] is
    0

solve;

var error = sum{i in 1..n_train} (sum{j in 1..n_features}(coeffs[j]*mat[i,j]) + sum{j
```

```
    in 1..n_features, k in j..n_features}(mat[i, j] * mat[i, k] *
    coeffs_second_order[j,k]) - mat[i, Y])^2; # The SSE with coeffs and
    coeffs_second_order
param y_tot = sum{i in 1..n_train} mat[i,Y]; # Sum of all y
param y_hat = y_tot/n_train; # Mean of all y
param ss_tot = sum{i in 1..n_train}(mat[i,Y] - y_hat)^2; # Denominator in r-squared
var r_sq = 1 - error/ss_tot; # r-squared
var adj_r_sq = r_sq - (1-r_sq)*(2/247);

display sse,r_sq, error, coeffs,coeffs_second_order, used, used_second_order;
```

Listing 8: 2h Code

```
n_features = 10 # The number of features (first order)
n_train = 250 # The number of training samples

def get_predictions(beta, beta_matrix, X):
  Y_pred = [] # List to store the predicitions
  for x in X: # Going through all the data
    y_pred = 0
    for i in range(n_features):
      y_pred += beta[i] * x[i] # simulating the dot product
    for i in range(n_features):
      for j in range(i, n_features):
        y_pred += beta_matrix[i][j]*x[i]*x[j] # including second order interactions
    Y_pred.append(y_pred) # Adding the prediction to the list
  return Y_pred # Returning the list

def sse(beta, beta_matrix, X, Y_actual):
  Y_pred = get_predictions(beta, beta_matrix, X) # Getting the predictions
  squared_differences = [(Y_actual[i] - Y_pred[i])**2 for i in range(len(Y_pred))] #
      Calculating the sum of squared errors
  return sum(squared_differences) # Returning the SSE

def read_data(filename):
  X = [] # List to store all the data points
  Y = [] # List to store all the target variables
  with open(filename) as f: # Opening the file containing the data (NOT necessarily
```

```
    .dat)
    lines = f.readlines() # Reading all the lines in the file
    lines = lines[n_train + 1:] # Skipping the first 251 lines since we need
      out-of-sample error and the first line is just feature names
    for line in lines: # Going through all lines
      if "AGE" in line: # Checking if the word "AGE" is in the line
        continue # Skipping if the word "AGE" is in the line because if it is, the
          line is just feature names, not an actual sample
      numbers = line.split() # Splitting the string
      numbers = [float(number) for number in numbers] # Converting strings to numbers
      y = numbers[-1] # Extracting the target variable
      x = numbers[0:-1] # Extracting the features
      X.append(x) # Adding the sample
      Y.append(y) # Adding the target variable
  return X, Y # returning the data points and target variables

X, Y_actual = read_data('data.txt') # Getting the data points as X and target
    variables as Y
# X[i] is the ith sample. X[i][j] is the jth feature of the ith sample
# Y[i] is the ith target variable

coeffs_part_a = [-46.3591,-226.708, 534.415, 230.201, -316.992, -4.3493e-08, 12.5382,
    204.877, 582.585, 160.929] # coefficients for first order features in Part 2f
coeffs_part_a_matrix= [[1.09417e-06, 0.000102456, 1.09665e-07, 4.80696e-07,
    1.91375e-08, -2.53955e-08, 1.98485e-08, 8.45594e-08, 3.59663e-07, 1.69066e-07],
[0,-1.80499e-08, 2.86425e-07, 4.22607e-07, 9.68603e-08, -2.63781e-10, -1.32483e-07,
    -3.07998e-08, 1.38711e-07, 7.5646e-08],
[0, 0, 1.46462e-07, 1.92339e-07, 6.51947e-08, 9.46737e-08, 7.90349e-08, 1.8639e-07,
    8.70322e-08, 4.95379e-07],
[0,0, 0, 7.47294e-08,-1.63143e-08, -4.19059e-09, 2.81844e-08, 5.16799e-08,
    5.4629e-08, 1.00506e-07],
[0,0,0,0,3.59483e-08,5.64592e-09, -6.18133e-08, -5.2328e-08, 5.35758e-08,7.07302e-08],
[0,0,0,0,0,-5.44664e-08, -8.03162e-09, -1.5056e-08, 1.33778e-07, 7.95601e-08],
[0,0,0,0,0,0,4.66955e-08, 3.47648e-08, -4.29466e-08, 8.84204e-08],
[0,0,0,0,0,0,0,4.1884e-08, -1.03307e-09, 2.61773e-07],
[0,0,0,0,0,0,0,0,-1.75994e-08, 2.13799e-07],
[0,0,0,0,0,0,0,0,0,102.204]] # coefficients for second order features in Part 2f
```

```
print("SSE Part 2F: {0}".format(sse(coeffs_part_a,coeffs_part_a_matrix, X, Y_actual)))

coeffs_part_b = [0,-233.553, 595.908, 263.431, -327.887 ,0, 0, 242.18 ,561.239,
    121.949] # coefficients for first order features in Part 2g
coeffs_part_b_matrix= [[0,0,0, 3540.81, 0, 0, 0, 0, 0, 0],
[0, 0, 4250.64, 0, 0, 0, -3850.05,0,0,0],
[0,0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0,0],
[0,0,0,0,0,0,0,0,0,0]] # coefficients for second order features in Part 2g
print("SSE Part 2G: {0}".format(sse(coeffs_part_b,coeffs_part_b_matrix, X, Y_actual)))
```

Listing 9: data file

```
reset;
param n_samples >= 0; # Total number of samples. In this case, 442
param n_features >= 0; # Total number of features. In this case, 10
param n_train >= 0; # Total number of training samples. In this case, 250
param first_order_coeff_bound; # Upper limit on any first order coefficient
param second_order_coeff_bound; # Upper limit on any second order coefficient
param mat{1..n_samples, 1..n_features+1}; # mat[i,j] is the jth feature of the ith
    sample. If j=11, it is the target variable of the ith sample
param AGE; # number to refer to AGE feature
param SEX; # number to refer to SEX feature
param BMI; # number to refer to BMI feature
param BP; # number to refer to BP feature
param S1; # number to refer to S1 feature
param S2; # number to refer to S2 feature
param S3; # number to refer to S3 feature
param S4; # number to refer to S4 feature
param S5; # number to refer to S5 feature
param S6; # number to refer to S6 feature
param Y; # number to refer to target variable
param lambdas{1..11}; # All possible lambda values for Part I as a list
```

```
param lambda_part_2; # Lambda value for Part II
data;
param n_samples := 442;
param n_train := 250;
param n_features := 10;
param first_order_coeff_bound := 1000;
param second_order_coeff_bound := 5000;
param AGE := 1;
param SEX := 2;
param BMI := 3;
param BP := 4;
param S1 := 5;
param S2 := 6;
param S3 := 7;
param S4 := 8;
param S5 := 9;
param S6 := 10;
param Y := 11;
param lambdas := 1 200 2 220 3 240 4 260 5 280 6 300 7 320 8 340 9 360 10 380 11 400;
param lambda_part_2 := 9.57;
param mat: 1 2 3 4 5 6 7 8 9 10 11 :=
1 0.038075906 0.05068012 0.0616962065 2.187235e-02 -0.044223498 -3.482076e-02
    0.043400846 -0.0025922620 0.0199084209 -0.017646125 -1.1334842
2  -0.001882017 -0.04464164 -0.0514740612 -2.632783e-02 -0.008448724 -1.916334e-02
    -0.074411564 -0.0394933829 -0.0683297436 -0.092204050 -77.1334842
3   0.085298906 0.05068012 0.0444512133 -5.670611e-03 -0.045599451 -3.419447e-02
    0.032355932 -0.0025922620 0.0028637705 -0.025930339 -11.1334842
4  -0.089062939 -0.04464164 -0.0115950145 -3.665645e-02 0.012190569 2.499059e-02
    0.036037570 0.0343088589 0.0226920226 -0.009361911 53.8665158
5   0.005383060 -0.04464164 -0.0363846922 2.187235e-02 0.003934852 1.559614e-02
    -0.008142084 -0.0025922620 -0.0319914449 -0.046640874 -17.1334842
6  -0.092695478 -0.04464164 -0.0406959405 -1.944209e-02 -0.068990650 -7.928784e-02
    -0.041276824 -0.0763945038 -0.0411803852 -0.096346157 -55.1334842
7  -0.045472478 0.05068012 -0.0471628129 -1.599922e-02 -0.040095640 -2.480001e-02
    -0.000778808 -0.0394933829 -0.0629129499 -0.038356660 -14.1334842
8   0.063503676 0.05068012 -0.0018947058 6.662967e-02 0.090619882 1.089144e-01
    -0.022868635 0.0177033545 -0.0358167281 0.003064409 -89.1334842
```

```
9   0.041708445 0.05068012 0.0616962065 -4.009932e-02 -0.013952536 6.201686e-03
    0.028674294 -0.0025922620 -0.0149564750 0.011348623 -42.1334842
10  -0.070900247 -0.04464164 0.0390621530 -3.321358e-02 -0.012576583 -3.450761e-02
    0.024992657 -0.0025922620 0.0677363261 -0.013504018 157.8665158
11  -0.096328016 -0.04464164 -0.0838084235 8.100872e-03 -0.103389471 -9.056119e-02
    0.013947743 -0.0763945038 -0.0629129499 -0.034214553 -51.1334842
12   0.027178291 0.05068012 0.0175059115 -3.321358e-02 -0.007072771 4.597154e-02
    0.065490672 0.0712099798 -0.0964332229 -0.059067194 -83.1334842
13   0.016280676 -0.04464164 -0.0288400077 -9.113481e-03 -0.004320866 -9.768886e-03
    -0.044958462 -0.0394933829 -0.0307512099 -0.042498767 26.8665158
14   0.005383060 0.05068012 -0.0018947058 8.100872e-03 -0.004320866 -1.571871e-02
    0.002902830 -0.0025922620 0.0383932482 -0.013504018 32.8665158
15   0.045340983 -0.04464164 -0.0256065715 -1.255635e-02 0.017694380 -6.128358e-05
    -0.081774840 -0.0394933829 -0.0319914449 -0.075635622 -34.1334842
16  -0.052737555 0.05068012 -0.0180618869 8.040116e-02 0.089243929 1.076618e-01
    0.039719208 0.1081111006 0.0360557901 -0.042498767 18.8665158
17  -0.005514555 -0.04464164 0.0422955892 4.941532e-02 0.024574144 -2.386057e-02
    -0.074411564 -0.0394933829 0.0522799998 0.027917051 13.8665158
18   0.070768752 0.05068012 0.0121168511 5.630106e-02 0.034205814 4.941617e-02
    0.039719208 0.0343088589 0.0273677075 -0.001077698 -8.1334842
19  -0.038207401 -0.04464164 -0.0105172024 -3.665645e-02 -0.037343734 -1.947649e-02
    0.028674294 -0.0025922620 -0.0181182673 -0.017646125 -55.1334842
20  -0.027309786 -0.04464164 -0.0180618869 -4.009932e-02 -0.002944913 -1.133463e-02
    -0.037595186 -0.0394933829 -0.0089440190 -0.054925087 15.8665158
21  -0.049105016 -0.04464164 -0.0568631216 -4.354219e-02 -0.045599451 -4.327577e-02
    -0.000778808 -0.0394933829 -0.0119006848 0.015490730 -84.1334842
22  -0.085430401 0.05068012 -0.0223731352 1.215131e-03 -0.037343734 -2.636575e-02
    -0.015505359 -0.0394933829 -0.0721284546 -0.017646125 -103.1334842
23  -0.085430401 -0.04464164 -0.0040503300 -9.113481e-03 -0.002944913 7.767428e-03
    -0.022868635 -0.0394933829 -0.0611765951 -0.013504018 -84.1334842
24   0.045340983 0.05068012 0.0606183944 3.105334e-02 0.028702003 -4.734670e-02
    0.054445759 0.0712099798 0.1335989800 0.135611831 92.8665158
25  -0.063635170 -0.04464164 0.0358287167 -2.288496e-02 -0.030463970 -1.885019e-02
    0.006584468 -0.0025922620 -0.0259524244 -0.054925087 31.8665158
26  -0.067267709 0.05068012 -0.0126728266 -4.009932e-02 -0.015328488 4.635943e-03
    0.058127397 0.0343088589 0.0191990331 -0.034214553 49.8665158
27  -0.107225632 -0.04464164 -0.0773415510 -2.632783e-02 -0.089629943 -9.619786e-02
```

```
   -0.026550273 -0.0763945038 -0.0425721049 -0.005219804 -15.1334842
28 -0.023677247 -0.04464164 0.0595405824 -4.009932e-02 -0.042847546 -4.358892e-02
   -0.011823721 -0.0394933829 -0.0159982678 0.040343372 -67.1334842
29  0.052606060 -0.04464164 -0.0212953232 -7.452802e-02 -0.040095640 -3.763910e-02
    0.006584468 -0.0394933829 -0.0006092542 -0.054925087 -21.1334842
30  0.067136214 0.05068012 -0.0062059541 6.318680e-02 -0.042847546 -9.588471e-02
   -0.052321737 -0.0763945038 0.0594238004 0.052769692 130.8665158
31 -0.060002632 -0.04464164 0.0444512133 -1.944209e-02 -0.009824677 -7.576847e-03
   -0.022868635 -0.0394933829 -0.0271286456 -0.009361911 -23.1334842
32 -0.023677247 -0.04464164 -0.0654856182 -8.141377e-02 -0.038719687 -5.360967e-02
   -0.059685013 -0.0763945038 -0.0371283460 -0.042498767 -93.1334842
33  0.034443368 0.05068012 0.1252871189 2.875810e-02 -0.053855168 -1.290037e-02
    0.102307051 0.1081111006 0.0002714857 0.027917051 188.8665158
34  0.030810830 -0.04464164 -0.0503962492 -2.227740e-03 -0.044223498 -8.993489e-02
   -0.118591218 -0.0763945038 -0.0181182673 0.003064409 -65.1334842
35  0.016280676 -0.04464164 -0.0633299941 -5.731367e-02 -0.057983027 -4.891244e-02
   -0.008142084 -0.0394933829 -0.0594726974 -0.067351408 -87.1334842
36  0.048973522 0.05068012 -0.0309956318 -4.928031e-02 0.049341296 -4.132214e-03
   -0.133317769 -0.0535158088 0.0213108466 0.019632837 -50.1334842
37  0.012648137 -0.04464164 0.0228949719 5.285819e-02 0.008062710 -2.855779e-02
   -0.037595186 -0.0394933829 0.0547240033 -0.025930339 112.8665158
38 -0.009147093 -0.04464164 0.0110390390 -5.731367e-02 -0.024960158 -4.296262e-02
   -0.030231910 -0.0394933829 0.0170371324 -0.005219804 123.8665158
39 -0.001882017 0.05068012 0.0713965152 9.761551e-02 0.087867976 7.540750e-02
    0.021311019 0.0712099798 0.0714240328 0.023774944 99.8665158
40 -0.001882017 0.05068012 0.0142724753 -7.452802e-02 0.002558899 6.201686e-03
    0.013947743 -0.0025922620 0.0191990331 0.003064409 -62.1334842
41  0.005383060 0.05068012 -0.0083615783 2.187235e-02 0.054845107 7.321546e-02
    0.024992657 0.0343088589 0.0125531528 0.094190762 -52.1334842
42 -0.099960555 -0.04464164 -0.0676412423 -1.089567e-01 -0.074494461 -7.271173e-02
   -0.015505359 -0.0394933829 -0.0498684677 -0.009361911 -97.1334842
43 -0.060002632 0.05068012 -0.0105172024 -1.485160e-02 -0.049727310 -2.354742e-02
    0.058127397 0.0158582984 -0.0099189574 -0.034214553 -91.1334842
44  0.019913214 -0.04464164 -0.0234509473 -7.108515e-02 0.020446286 -1.008203e-02
   -0.118591218 -0.0763945038 -0.0425721049 0.073480227 -60.1334842
45  0.045340983 0.05068012 0.0681630790 8.100872e-03 -0.016704441 4.635943e-03
    0.076535586 0.0712099798 0.0324332258 -0.017646125 106.8665158
```

```
46  0.027178291 0.05068012 -0.0353068801 3.220097e-02 -0.011200630 1.504459e-03
    0.010266105 -0.0025922620 -0.0149564750 -0.050782980 -99.1334842
47 -0.056370093 -0.04464164 -0.0115950145 -3.321358e-02 -0.046975404 -4.765985e-02
   -0.004460446 -0.0394933829 -0.0079793976 -0.088061943 37.8665158
48 -0.078165324 -0.04464164 -0.0730303027 -5.731367e-02 -0.084126131 -7.427747e-02
    0.024992657 -0.0394933829 -0.0181182673 -0.083919836 -10.1334842
49  0.067136214 0.05068012 -0.0417737526 1.154374e-02 0.002558899 5.888537e-03
   -0.041276824 -0.0394933829 -0.0594726974 -0.021788232 -77.1334842
50 -0.041839939 0.05068012 0.0142724753 -5.670611e-03 -0.012576583 6.201686e-03
    0.072853948 0.0712099798 0.0354619387 -0.013504018 -10.1334842
51  0.034443368 -0.04464164 -0.0072837662 1.498661e-02 -0.044223498 -3.732595e-02
    0.002902830 -0.0394933829 -0.0213936809 0.007206516 2.8665158
52  0.059871137 0.05068012 0.0164280994 2.875810e-02 -0.041471593 -2.918409e-02
    0.028674294 -0.0025922620 -0.0023966815 -0.021788232 72.8665158
53 -0.052737555 -0.04464164 -0.0094393904 -5.670611e-03 0.039709626 4.471895e-02
   -0.026550273 -0.0025922620 -0.0181182673 -0.013504018 -93.1334842
54 -0.009147093 -0.04464164 -0.0159062628 7.007254e-02 0.012190569 2.217226e-02
   -0.015505359 -0.0025922620 -0.0332487872 0.048627585 -48.1334842
55 -0.049105016 -0.04464164 0.0250505960 8.100872e-03 0.020446286 1.778818e-02
   -0.052321737 -0.0394933829 -0.0411803852 0.007206516 29.8665158
56 -0.041839939 -0.04464164 -0.0493184371 -3.665645e-02 -0.007072771 -2.260797e-02
   -0.085456477 -0.0394933829 -0.0664881482 0.007206516 -24.1334842
57 -0.041839939 -0.04464164 0.0412177771 -2.632783e-02 -0.031839923 -3.043668e-02
    0.036037570 0.0029429061 0.0336568129 -0.017646125 -100.1334842
58 -0.027309786 -0.04464164 -0.0633299941 -5.042793e-02 -0.089629943 -1.043397e-01
   -0.052321737 -0.0763945038 -0.0561575731 -0.067351408 -115.1334842
59  0.041708445 -0.04464164 -0.0644078061 3.564384e-02 0.012190569 -5.799375e-02
   -0.181179060 -0.0763945038 -0.0006092542 -0.050782980 17.8665158
60  0.063503676 0.05068012 -0.0256065715 1.154374e-02 0.064476777 4.847673e-02
   -0.030231910 -0.0025922620 0.0383932482 0.019632837 17.8665158
61 -0.070900247 -0.04464164 -0.0040503300 -4.009932e-02 -0.066238744 -7.866155e-02
   -0.052321737 -0.0763945038 -0.0514005353 -0.034214553 -91.1334842
62 -0.041839939 0.05068012 0.0045721666 -5.387080e-02 -0.044223498 -2.730520e-02
    0.080217224 0.0712099798 0.0366457978 0.019632837 -8.1334842
63 -0.027309786 0.05068012 -0.0072837662 -4.009932e-02 -0.011200630 -1.383982e-02
   -0.059685013 -0.0394933829 -0.0823814833 -0.025930339 -100.1334842
64 -0.034574863 -0.04464164 -0.0374625043 -6.075654e-02 0.020446286 4.346635e-02
```

```
    0.013947743 -0.0025922620 -0.0307512099 -0.071493515 -24.1334842
65   0.067136214 0.05068012 -0.0256065715 -4.009932e-02 -0.063486838 -5.987264e-02
    0.002902830 -0.0394933829 -0.0191970476 0.011348623 -81.1334842
66  -0.045472478 0.05068012 -0.0245287594 5.974393e-02 0.005310804 1.496984e-02
    0.054445759 0.0712099798 0.0423448954 0.015490730 10.8665158
67  -0.009147093 0.05068012 -0.0180618869 -3.321358e-02 -0.020832300 1.215151e-02
    0.072853948 0.0712099798 0.0002714857 0.019632837 -2.1334842
68   0.041708445 0.05068012 -0.0148284507 -1.714685e-02 -0.005696818 8.393725e-03
    0.013947743 -0.0018542396 -0.0119006848 0.003064409 -55.1334842
69   0.038075906 0.05068012 -0.0299178198 -4.009932e-02 -0.033215876 -2.417372e-02
    0.010266105 -0.0025922620 -0.0129079423 0.003064409 7.8665158
70   0.016280676 -0.04464164 -0.0460850009 -5.670611e-03 -0.075870414 -6.143838e-02
    0.013947743 -0.0394933829 -0.0514005353 0.019632837 25.8665158
71  -0.001882017 -0.04464164 -0.0697968665 -1.255635e-02 -0.000193007 -9.142589e-03
    -0.070729926 -0.0394933829 -0.0629129499 0.040343372 -104.1334842
72  -0.001882017 -0.04464164 0.0336730926 1.251585e-01 0.024574144 2.624319e-02
    0.010266105 -0.0025922620 0.0267142576 0.061053906 117.8665158
73   0.063503676 0.05068012 -0.0040503300 -1.255635e-02 0.103003457 4.878988e-02
    -0.056003375 -0.0025922620 0.0844952822 -0.017646125 49.8665158
74   0.012648137 0.05068012 -0.0202175111 -2.227740e-03 0.038333673 5.317395e-02
    0.006584468 0.0343088589 -0.0051453080 -0.009361911 -41.1334842
75   0.012648137 0.05068012 0.0024165425 5.630106e-02 0.027326050 1.716188e-02
    -0.041276824 -0.0394933829 0.0037117382 0.073480227 -67.1334842
76  -0.009147093 0.05068012 -0.0309956318 -2.632783e-02 -0.011200630 -1.000729e-03
    0.021311019 -0.0025922620 0.0062093156 0.027917051 -110.1334842
77  -0.030942324 0.05068012 0.0282840322 7.007254e-02 -0.126780670 -1.068449e-01
    0.054445759 -0.0479806407 -0.0307512099 0.015490730 17.8665158
78  -0.096328016 -0.04464164 -0.0363846922 -7.452802e-02 -0.038719687 -2.761835e-02
    -0.015505359 -0.0394933829 -0.0740888715 -0.001077698 47.8665158
79   0.005383060 -0.04464164 -0.0579409337 -2.288496e-02 -0.067614697 -6.832765e-02
    0.054445759 -0.0025922620 0.0428956879 -0.083919836 99.8665158
80  -0.103593093 -0.04464164 -0.0374625043 -2.632783e-02 0.002558899 1.998022e-02
    -0.011823721 -0.0025922620 -0.0683297436 -0.025930339 -39.1334842
81   0.070768752 -0.04464164 0.0121168511 4.252958e-02 0.071356542 5.348710e-02
    -0.052321737 -0.0025922620 0.0253931349 -0.005219804 -9.1334842
82   0.012648137 0.05068012 -0.0223731352 -2.977071e-02 0.010814616 2.843523e-02
    0.021311019 0.0343088589 -0.0060802482 -0.001077698 -101.1334842
```

```
83  -0.016412170 -0.04464164 -0.0353068801 -2.632783e-02 0.032829862 1.716188e-02
    -0.100183029 -0.0394933829 -0.0702093127 -0.079777729 -100.1334842
84  -0.038207401 -0.04464164 0.0099612270 -4.698506e-02 -0.059358980 -5.298337e-02
    0.010266105 -0.0394933829 -0.0159982678 -0.042498767 57.8665158
85   0.001750522 -0.04464164 -0.0396181284 -1.009234e-01 -0.029088017 -3.012354e-02
    -0.044958462 -0.0501947079 -0.0683297436 -0.129483012 -87.1334842
86   0.045340983 -0.04464164 0.0713965152 1.215131e-03 -0.009824677 -1.000729e-03
    -0.015505359 -0.0394933829 -0.0411803852 -0.071493515 -11.1334842
87  -0.070900247 0.05068012 -0.0751859269 -4.009932e-02 -0.051103263 -1.509241e-02
    0.039719208 -0.0025922620 -0.0964332229 -0.034214553 -97.1334842
88   0.045340983 -0.04464164 -0.0062059541 1.154374e-02 0.063100825 1.622244e-02
    -0.096501391 -0.0394933829 0.0428956879 -0.038356660 -18.1334842
89  -0.052737555 0.05068012 -0.0406959405 -6.764228e-02 -0.031839923 -3.701280e-02
    -0.037595186 -0.0394933829 -0.0345237153 0.069338120 -110.1334842
90  -0.045472478 -0.04464164 -0.0482406250 -1.944209e-02 -0.000193007 -1.603186e-02
    -0.067048288 -0.0394933829 -0.0247911874 0.019632837 -41.1334842
91   0.012648137 -0.04464164 -0.0256065715 -4.009932e-02 -0.030463970 -4.515466e-02
    -0.078093202 -0.0763945038 -0.0721284546 0.011348623 -54.1334842
92   0.045340983 -0.04464164 0.0519958979 -5.387080e-02 0.063100825 6.476045e-02
    0.010266105 0.0343088589 0.0372320112 0.019632837 11.8665158
93  -0.020044709 -0.04464164 0.0045721666 9.761551e-02 0.005310804 -2.072908e-02
    -0.063366651 -0.0394933829 0.0125531528 0.011348623 -104.1334842
94  -0.049105016 -0.04464164 -0.0644078061 -1.020710e-01 -0.002944913 -1.540556e-02
    -0.063366651 -0.0472426183 -0.0332487872 -0.054925087 -56.1334842
95  -0.078165324 -0.04464164 -0.0169840749 -1.255635e-02 -0.000193007 -1.352667e-02
    -0.070729926 -0.0394933829 -0.0411803852 -0.092204050 -62.1334842
96  -0.070900247 -0.04464164 -0.0579409337 -8.141377e-02 -0.045599451 -2.887094e-02
    0.043400846 -0.0025922620 0.0011437974 -0.005219804 9.8665158
97   0.056238599 0.05068012 0.0099612270 4.941532e-02 -0.004320866 -1.227407e-02
    0.043400846 0.0343088589 0.0607877542 0.032059158 -2.1334842
98  -0.027309786 -0.04464164 0.0886415084 -2.518021e-02 0.021822239 4.252691e-02
    0.032355932 0.0343088589 0.0028637705 0.077622334 126.8665158
99   0.001750522 0.05068012 -0.0051281421 -1.255635e-02 -0.015328488 -1.383982e-02
    -0.008142084 -0.0394933829 -0.0060802482 -0.067351408 -60.1334842
100  -0.001882017 -0.04464164 -0.0644078061 1.154374e-02 0.027326050 3.751653e-02
    0.013947743 0.0343088589 0.0117839004 -0.054925087 -69.1334842
101   0.016280676 -0.04464164 0.0175059115 -2.288496e-02 0.060348919 4.440580e-02
```

```
    -0.030231910 -0.0025922620 0.0372320112 -0.001077698 -24.1334842
102  0.016280676 0.05068012 -0.0450071888 6.318680e-02 0.010814616 -3.744320e-04
    -0.063366651 -0.0394933829 -0.0307512099 0.036201265 -50.1334842
103 -0.092695478 -0.04464164 0.0282840322 -1.599922e-02 0.036957720 2.499059e-02
    -0.056003375 -0.0394933829 -0.0051453080 -0.001077698 149.8665158
104  0.059871137 0.05068012 0.0412177771 1.154374e-02 0.041085579 7.071027e-02
    0.036037570 0.0343088589 -0.0109044358 -0.030072446 45.8665158
105 -0.027309786 -0.04464164 0.0649296427 -2.227740e-03 -0.024960158 -1.728445e-02
    -0.022868635 -0.0394933829 -0.0611765951 -0.063209301 -57.1334842
106  0.023545753 0.05068012 -0.0320734439 -4.009932e-02 -0.031839923 -2.166853e-02
    0.013947743 -0.0025922620 -0.0109044358 0.019632837 -99.1334842
107 -0.096328016 -0.04464164 -0.0762637389 -4.354219e-02 -0.045599451 -3.482076e-02
    -0.008142084 -0.0394933829 -0.0594726974 -0.083919836 -18.1334842
108  0.027178291 -0.04464164 0.0498402737 -5.501842e-02 -0.002944913 4.064802e-02
    0.058127397 0.0527594193 -0.0529587932 -0.005219804 -8.1334842
109  0.019913214 0.05068012 0.0455290254 2.990572e-02 -0.062110886 -5.580171e-02
    0.072853948 0.0269286347 0.0456008084 0.040343372 79.8665158
110  0.038075906 0.05068012 -0.0094393904 2.362754e-03 0.001182946 3.751653e-02
    0.054445759 0.0501763409 -0.0259524244 0.106617082 -71.1334842
111  0.041708445 0.05068012 -0.0320734439 -2.288496e-02 -0.049727310 -4.014429e-02
    -0.030231910 -0.0394933829 -0.1260973856 0.015490730 -48.1334842
112  0.019913214 -0.04464164 0.0045721666 -2.632783e-02 0.023198192 1.027262e-02
    -0.067048288 -0.0394933829 -0.0236445576 -0.046640874 -93.1334842
113 -0.085430401 -0.04464164 0.0207393477 -2.632783e-02 0.005310804 1.966707e-02
    0.002902830 -0.0025922620 -0.0236445576 0.003064409 93.8665158
114  0.019913214 0.05068012 0.0142724753 6.318680e-02 0.014942474 2.029337e-02
    0.047082483 0.0343088589 0.0466607724 0.090048655 144.8665158
115  0.023545753 -0.04464164 0.1101977498 6.318680e-02 0.013566522 -3.294187e-02
    0.024992657 0.0206554442 0.0992402257 0.023774944 105.8665158
116 -0.030942324 0.05068012 0.0013387304 -5.670611e-03 0.064476777 4.941617e-02
    0.047082483 0.1081111006 0.0837967664 0.003064409 76.8665158
117  0.048973522 0.05068012 0.0584627703 7.007254e-02 0.013566522 2.060651e-02
    0.021311019 0.0343088589 0.0220040505 0.027917051 122.8665158
118  0.059871137 -0.04464164 -0.0212953232 8.728690e-02 0.045213437 3.156671e-02
    0.047082483 0.0712099798 0.0791210814 0.135611831 128.8665158
119 -0.056370093 0.05068012 -0.0105172024 2.531523e-02 0.023198192 4.002172e-02
    0.039719208 0.0343088589 0.0206123307 0.056911799 26.8665158
```

```
120  0.016280676 -0.04464164 -0.0471628129 -2.227740e-03 -0.019456347 -4.296262e-02
    -0.033913548 -0.0394933829 0.0273677075 0.027917051 47.8665158
121 -0.049105016 -0.04464164 0.0045721666 1.154374e-02 -0.037343734 -1.853704e-02
    0.017629381 -0.0025922620 -0.0398095944 -0.021788232 47.8665158
122  0.063503676 -0.04464164 0.0175059115 2.187235e-02 0.008062710 2.154596e-02
    0.036037570 0.0343088589 0.0199084209 0.011348623 20.8665158
123  0.048973522 0.05068012 0.0810968238 2.187235e-02 0.043837485 6.413415e-02
    0.054445759 0.0712099798 0.0324332258 0.048627585 27.8665158
124  0.005383060 0.05068012 0.0347509047 -1.080116e-03 0.152537760 1.987880e-01
    0.061809035 0.1852344433 0.0155668445 0.073480227 -68.1334842
125 -0.005514555 -0.04464164 0.0239727839 8.100872e-03 -0.034591828 -3.889169e-02
    -0.022868635 -0.0394933829 -0.0159982678 -0.013504018 -31.1334842
126 -0.005514555 0.05068012 -0.0083615783 -2.227740e-03 -0.033215876 -6.363042e-02
    0.036037570 -0.0025922620 0.0805854642 0.007206516 8.8665158
127 -0.089062939 -0.04464164 -0.0611743699 -2.632783e-02 -0.055231121 -5.454912e-02
    -0.041276824 -0.0763945038 -0.0939356455 -0.054925087 -53.1334842
128  0.034443368 0.05068012 -0.0018947058 -1.255635e-02 0.038333673 1.371725e-02
    -0.078093202 -0.0394933829 0.0045518905 -0.096346157 -43.1334842
129 -0.052737555 -0.04464164 -0.0622521820 -2.632783e-02 -0.005696818 -5.071659e-03
    -0.030231910 -0.0394933829 -0.0307512099 -0.071493515 -37.1334842
130  0.009015599 -0.04464164 0.0164280994 4.658002e-03 0.009438663 1.058576e-02
    0.028674294 0.0343088589 0.0389683660 0.119043403 115.8665158
131 -0.063635170 0.05068012 0.0961861929 1.045013e-01 -0.002944913 -4.758511e-03
    0.006584468 -0.0025922620 0.0226920226 0.073480227 121.8665158
132 -0.096328016 -0.04464164 -0.0697968665 -6.764228e-02 -0.019456347 -1.070833e-02
    -0.015505359 -0.0394933829 -0.0468794828 -0.079777729 5.8665158
133  0.016280676 0.05068012 -0.0212953232 -9.113481e-03 0.034205814 4.785043e-02
    -0.000778808 -0.0025922620 -0.0129079423 0.023774944 -45.1334842
134 -0.041839939 0.05068012 -0.0536296854 -4.009932e-02 -0.084126131 -7.177228e-02
    0.002902830 -0.0394933829 -0.0721284546 -0.030072446 -69.1334842
135 -0.074532786 -0.04464164 0.0433734013 -3.321358e-02 0.012190569 2.518649e-04
    -0.063366651 -0.0394933829 -0.0271286456 -0.046640874 -49.1334842
136 -0.005514555 -0.04464164 0.0563071461 -3.665645e-02 -0.048351357 -4.296262e-02
    0.072853948 0.0379989710 0.0507815134 0.056911799 119.8665158
137 -0.092695478 -0.04464164 -0.0816527993 -5.731367e-02 -0.060734933 -6.801450e-02
    -0.048640099 -0.0763945038 -0.0664881482 -0.021788232 -67.1334842
138  0.005383060 -0.04464164 0.0498402737 9.761551e-02 -0.015328488 -1.634500e-02
```

```
    0.006584468 -0.0025922620 0.0170371324 -0.013504018 127.8665158
139   0.034443368 0.05068012 0.1112755619 7.695829e-02 -0.031839923 -3.388132e-02
    0.021311019 -0.0025922620 0.0280165065 0.073480227 183.8665158
140   0.023545753 -0.04464164 0.0616962065 5.285819e-02 -0.034591828 -4.891244e-02
    0.028674294 -0.0025922620 0.0547240033 -0.005219804 128.8665158
141   0.041708445 0.05068012 0.0142724753 4.252958e-02 -0.030463970 -1.313877e-03
    0.043400846 -0.0025922620 -0.0332487872 0.015490730 -34.1334842
142  -0.027309786 -0.04464164 0.0476846496 -4.698506e-02 0.034205814 5.724488e-02
    0.080217224 0.1302517732 0.0450661683 0.131469724 164.8665158
143   0.041708445 0.05068012 0.0121168511 3.908671e-02 0.054845107 4.440580e-02
    -0.004460446 -0.0025922620 0.0456008084 -0.001077698 82.8665158
144  -0.030942324 -0.04464164 0.0056499787 -9.113481e-03 0.019070333 6.827983e-03
    -0.074411564 -0.0394933829 -0.0411803852 -0.042498767 -92.1334842
145   0.030810830 0.05068012 0.0466068375 -1.599922e-02 0.020446286 5.066877e-02
    0.058127397 0.0712099798 0.0062093156 0.007206516 21.8665158
146  -0.041839939 -0.04464164 0.1285205551 6.318680e-02 -0.033215876 -3.262872e-02
    -0.011823721 -0.0394933829 -0.0159982678 -0.050782980 106.8665158
147  -0.030942324 0.05068012 0.0595405824 1.215131e-03 0.012190569 3.156671e-02
    0.043400846 0.0343088589 0.0148227108 0.007206516 25.8665158
148  -0.056370093 -0.04464164 0.0929527567 -1.944209e-02 0.014942474 2.342485e-02
    0.028674294 0.0254525899 0.0260560896 0.040343372 -24.1334842
149  -0.060002632 0.05068012 0.0153502873 -1.944209e-02 0.036957720 4.816358e-02
    -0.019186997 -0.0025922620 -0.0307512099 -0.001077698 -56.1334842
150  -0.049105016 0.05068012 -0.0051281421 -4.698506e-02 -0.020832300 -2.041593e-02
    0.069172310 0.0712099798 0.0612379075 -0.038356660 -26.1334842
151   0.023545753 -0.04464164 0.0703187031 2.531523e-02 -0.034591828 -1.446611e-02
    0.032355932 -0.0025922620 -0.0191970476 -0.009361911 135.8665158
152   0.001750522 -0.04464164 -0.0040503300 -5.670611e-03 -0.008448724 -2.386057e-02
    -0.052321737 -0.0394933829 -0.0089440190 -0.013504018 -64.1334842
153  -0.034574863 0.05068012 -0.0008168938 7.007254e-02 0.039709626 6.695249e-02
    0.065490672 0.1081111006 0.0267142576 0.073480227 139.8665158
154   0.041708445 0.05068012 -0.0439293767 6.318680e-02 -0.004320866 1.622244e-02
    0.013947743 -0.0025922620 -0.0345237153 0.011348623 -81.1334842
155   0.067136214 0.05068012 0.0207393477 -5.670611e-03 0.020446286 2.624319e-02
    0.002902830 -0.0025922620 0.0086402829 0.003064409 44.8665158
156  -0.027309786 0.05068012 0.0606183944 4.941532e-02 0.085116070 8.636769e-02
    0.002902830 0.0343088589 0.0378144788 0.048627585 33.8665158
```

```
157  -0.016412170 -0.04464164 -0.0105172024 1.215131e-03 -0.037343734 -3.576021e-02
    -0.011823721 -0.0394933829 -0.0213936809 -0.034214553 -127.1334842
158  -0.001882017 0.05068012 -0.0331512560 -1.829447e-02 0.031453909 4.284006e-02
    0.013947743 0.0199174217 0.0102256424 0.027917051 -68.1334842
159  -0.012779632 -0.04464164 -0.0654856182 -6.993753e-02 0.001182946 1.684873e-02
    0.002902830 -0.0070203965 -0.0307512099 -0.050782980 -56.1334842
160  -0.005514555 -0.04464164 0.0433734013 8.728690e-02 0.013566522 7.141131e-03
    0.013947743 -0.0025922620 0.0423448954 -0.017646125 42.8665158
161  -0.009147093 -0.04464164 -0.0622521820 -7.452802e-02 -0.023584206 -1.321352e-02
    -0.004460446 -0.0394933829 -0.0358167281 -0.046640874 -99.1334842
162  -0.045472478 0.05068012 0.0638518307 7.007254e-02 0.133274420 1.314611e-01
    0.039719208 0.1081111006 0.0757375885 0.085906548 64.8665158
163  -0.052737555 -0.04464164 0.0304396564 -7.452802e-02 -0.023584206 -1.133463e-02
    0.002902830 -0.0025922620 -0.0307512099 -0.001077698 19.8665158
164   0.016280676 0.05068012 0.0724743273 7.695829e-02 -0.008448724 5.575389e-03
    0.006584468 -0.0025922620 -0.0236445576 0.061053906 -21.1334842
165   0.045340983 -0.04464164 -0.0191396990 2.187235e-02 0.027326050 -1.352667e-02
    -0.100183029 -0.0394933829 0.0177634779 -0.013504018 61.8665158
166  -0.041839939 -0.04464164 -0.0665634303 -4.698506e-02 -0.037343734 -4.327577e-02
    -0.048640099 -0.0394933829 -0.0561575731 -0.013504018 -93.1334842
167  -0.056370093 0.05068012 -0.0600965578 -3.665645e-02 -0.088253990 -7.083284e-02
    0.013947743 -0.0394933829 -0.0781409107 -0.104630370 -82.1334842
168   0.070768752 -0.04464164 0.0692408910 3.793909e-02 0.021822239 1.504459e-03
    0.036037570 0.0391060046 0.0776327892 0.106617082 67.8665158
169   0.001750522 0.05068012 0.0595405824 -2.227740e-03 0.061724872 6.319471e-02
    0.058127397 0.1081111006 0.0689822116 0.127327617 115.8665158
170  -0.001882017 -0.04464164 -0.0266843835 4.941532e-02 0.058972966 -1.603186e-02
    0.047082483 0.0712099798 0.1335989800 0.019632837 -0.1334842
171   0.023545753 0.05068012 -0.0202175111 -3.665645e-02 -0.013952536 -1.509241e-02
    -0.059685013 -0.0394933829 -0.0964332229 -0.017646125 -105.1334842
172  -0.020044709 -0.04464164 -0.0460850009 -9.862812e-02 -0.075870414 -5.987264e-02
    0.017629381 -0.0394933829 -0.0514005353 -0.046640874 -78.1334842
173   0.041708445 0.05068012 0.0713965152 8.100872e-03 0.038333673 1.590929e-02
    0.017629381 0.0343088589 0.0734100780 0.085906548 142.8665158
174  -0.063635170 0.05068012 -0.0794971752 -5.670611e-03 -0.071742556 -6.644876e-02
    0.010266105 -0.0394933829 -0.0181182673 -0.054925087 -51.1334842
175   0.016280676 0.05068012 0.0099612270 -4.354219e-02 -0.096509707 -9.463212e-02
```

```
    0.039719208 -0.0394933829 0.0170371324 0.007206516 -1.1334842
176   0.067136214 -0.04464164 -0.0385403164 -2.632783e-02 -0.031839923 -2.636575e-02
    -0.008142084 -0.0394933829 -0.0271286456 0.003064409 -25.1334842
177   0.045340983 0.05068012 0.0196615356 3.908671e-02 0.020446286 2.593004e-02
    -0.008142084 -0.0025922620 -0.0033037126 0.019632837 84.8665158
178   0.048973522 -0.04464164 0.0272062202 -2.518021e-02 0.023198192 1.841448e-02
    0.061809035 0.0800662488 0.0722236508 0.032059158 72.8665158
179   0.041708445 -0.04464164 -0.0083615783 -2.632783e-02 0.024574144 1.622244e-02
    -0.070729926 -0.0394933829 -0.0483617248 -0.030072446 -71.1334842
180  -0.023677247 -0.04464164 -0.0159062628 -1.255635e-02 0.020446286 4.127431e-02
    0.043400846 0.0343088589 0.0140724525 -0.009361911 -1.1334842
181  -0.038207401 0.05068012 0.0045721666 3.564384e-02 -0.011200630 5.888537e-03
    0.047082483 0.0343088589 0.0163049528 -0.001077698 -45.1334842
182   0.048973522 -0.04464164 -0.0428515646 -5.387080e-02 0.045213437 5.004247e-02
    -0.033913548 -0.0025922620 -0.0259524244 -0.063209301 -88.1334842
183   0.045340983 0.05068012 0.0056499787 5.630106e-02 0.064476777 8.918603e-02
    0.039719208 0.0712099798 0.0155668445 -0.009361911 -14.1334842
184   0.045340983 0.05068012 -0.0353068801 6.318680e-02 -0.004320866 -1.627026e-03
    0.010266105 -0.0025922620 0.0155668445 0.056911799 32.8665158
185   0.016280676 -0.04464164 0.0239727839 -2.288496e-02 -0.024960158 -2.605261e-02
    0.032355932 -0.0025922620 0.0372320112 0.032059158 112.8665158
186  -0.074532786 0.05068012 -0.0180618869 8.100872e-03 -0.019456347 -2.480001e-02
    0.065490672 0.0343088589 0.0673172179 -0.017646125 -51.1334842
187  -0.081797862 0.05068012 0.0422955892 -1.944209e-02 0.039709626 5.755803e-02
    0.069172310 0.1081111006 0.0471861679 -0.038356660 -15.1334842
188  -0.067267709 -0.04464164 -0.0547074975 -2.632783e-02 -0.075870414 -8.210618e-02
    -0.048640099 -0.0763945038 -0.0868289932 -0.104630370 -9.1334842
189   0.005383060 -0.04464164 -0.0029725179 4.941532e-02 0.074108447 7.071027e-02
    -0.044958462 -0.0025922620 -0.0014985868 -0.009361911 -11.1334842
190  -0.001882017 -0.04464164 -0.0665634303 1.215131e-03 -0.002944913 3.070201e-03
    -0.011823721 -0.0025922620 -0.0202887478 -0.025930339 -73.1334842
191   0.009015599 -0.04464164 -0.0126728266 2.875810e-02 -0.018080394 -5.071659e-03
    0.047082483 0.0343088589 0.0233748413 -0.005219804 139.8665158
192  -0.005514555 0.05068012 -0.0417737526 -4.354219e-02 -0.079998273 -7.615636e-02
    0.032355932 -0.0394933829 0.0102256424 -0.009361911 25.8665158
193   0.056238599 0.05068012 -0.0309956318 8.100872e-03 0.019070333 2.123281e-02
    -0.033913548 -0.0394933829 -0.0295276227 -0.059067194 -61.1334842
```

```
194  0.009015599 0.05068012 -0.0051281421 -6.419941e-02 0.069980589 8.386250e-02
   0.039719208 0.0712099798 0.0395398781 0.019632837 -36.1334842
195 -0.067267709 -0.04464164 -0.0590187458 3.220097e-02 -0.051103263 -4.953874e-02
   0.010266105 -0.0394933829 0.0020078405 0.023774944 -66.1334842
196  0.027178291 0.05068012 0.0250505960 1.498661e-02 0.025950097 4.847673e-02
   0.039719208 0.0343088589 0.0078371423 0.023774944 -30.1334842
197 -0.023677247 -0.04464164 -0.0460850009 -3.321358e-02 0.032829862 3.626394e-02
   -0.037595186 -0.0025922620 -0.0332487872 0.011348623 -80.1334842
198  0.048973522 0.05068012 0.0034943545 7.007254e-02 -0.008448724 1.340410e-02
   0.054445759 0.0343088589 0.0133159679 0.036201265 -23.1334842
199 -0.052737555 -0.04464164 0.0541515220 -2.632783e-02 -0.055231121 -3.388132e-02
   0.013947743 -0.0394933829 -0.0740888715 -0.059067194 -10.1334842
200  0.041708445 -0.04464164 -0.0450071888 3.449621e-02 0.043837485 -1.571871e-02
   -0.037595186 -0.0144006207 0.0898986933 0.007206516 -62.1334842
201  0.056238599 -0.04464164 -0.0579409337 -7.965858e-03 0.052093202 4.910302e-02
   -0.056003375 -0.0214118336 -0.0283202425 0.044485479 5.8665158
202 -0.034574863 0.05068012 -0.0557853095 -1.599922e-02 -0.009824677 -7.889995e-03
   -0.037595186 -0.0394933829 -0.0529587932 0.027917051 -113.1334842
203  0.081666368 0.05068012 0.0013387304 3.564384e-02 0.126394656 9.106492e-02
   -0.019186997 0.0343088589 0.0844952822 -0.030072446 43.8665158
204 -0.001882017 0.05068012 0.0304396564 5.285819e-02 0.039709626 5.661859e-02
   0.039719208 0.0712099798 0.0253931349 0.027917051 69.8665158
205  0.110726675 0.05068012 0.0067277908 2.875810e-02 -0.027712064 -7.263698e-03
   0.047082483 0.0343088589 0.0020078405 0.077622334 124.8665158
206 -0.030942324 -0.04464164 0.0466068375 1.498661e-02 -0.016704441 -4.703355e-02
   -0.000778808 -0.0025922620 0.0634559214 -0.025930339 -53.1334842
207  0.001750522 0.05068012 0.0261284081 -9.113481e-03 0.024574144 3.845598e-02
   0.021311019 0.0343088589 0.0094364091 0.003064409 43.8665158
208  0.009015599 -0.04464164 0.0455290254 2.875810e-02 0.012190569 -1.383982e-02
   -0.026550273 -0.0394933829 0.0461323310 0.036201265 49.8665158
209  0.030810830 -0.04464164 0.0401399650 7.695829e-02 0.017694380 3.782968e-02
   0.028674294 0.0343088589 -0.0014985868 0.119043403 2.8665158
210  0.038075906 0.05068012 -0.0180618869 6.662967e-02 -0.051103263 -1.665815e-02
   0.076535586 0.0343088589 -0.0119006848 -0.013504018 -75.1334842
211  0.009015599 -0.04464164 0.0142724753 1.498661e-02 0.054845107 4.722413e-02
   -0.070729926 -0.0394933829 -0.0332487872 -0.059067194 38.8665158
212  0.092563983 -0.04464164 0.0369065288 2.187235e-02 -0.024960158 -1.665815e-02
```

```
    -0.000778808 -0.0394933829 -0.0225121719 -0.021788232 -82.1334842
213  0.067136214 -0.04464164 0.0034943545 3.564384e-02 0.049341296 3.125356e-02
    -0.070729926 -0.0394933829 -0.0006092542 0.019632837 -79.1334842
214  0.001750522 -0.04464164 -0.0708746786 -2.288496e-02 -0.001568960 -1.000729e-03
    -0.026550273 -0.0394933829 -0.0225121719 0.007206516 -103.1334842
215  0.030810830 -0.04464164 -0.0331512560 -2.288496e-02 -0.046975404 -8.116674e-02
    -0.103864667 -0.0763945038 -0.0398095944 -0.054925087 -87.1334842
216  0.027178291 0.05068012 0.0940305687 9.761551e-02 -0.034591828 -3.200243e-02
    0.043400846 -0.0025922620 0.0366457978 0.106617082 110.8665158
217  0.012648137 0.05068012 0.0358287167 4.941532e-02 0.053469155 7.415490e-02
    0.069172310 0.1450122215 0.0456008084 0.048627585 95.8665158
218  0.074401291 -0.04464164 0.0315174685 1.010584e-01 0.046589390 3.689023e-02
    -0.015505359 -0.0025922620 0.0336568129 0.044485479 143.8665158
219 -0.041839939 -0.04464164 -0.0654856182 -4.009932e-02 -0.005696818 1.434355e-02
    0.043400846 0.0343088589 0.0070268625 -0.013504018 61.8665158
220 -0.089062939 -0.04464164 -0.0417737526 -1.944209e-02 -0.066238744 -7.427747e-02
    -0.008142084 -0.0394933829 0.0011437974 -0.030072446 32.8665158
221  0.023545753 0.05068012 -0.0396181284 -5.670611e-03 -0.048351357 -3.325502e-02
    -0.011823721 -0.0394933829 -0.1016435479 -0.067351408 -74.1334842
222 -0.045472478 -0.04464164 -0.0385403164 -2.632783e-02 -0.015328488 8.781618e-04
    0.032355932 -0.0025922620 0.0011437974 -0.038356660 -59.1334842
223 -0.023677247 0.05068012 -0.0256065715 4.252958e-02 -0.053855168 -4.765985e-02
    0.021311019 -0.0394933829 0.0011437974 0.019632837 99.8665158
224 -0.099960555 -0.04464164 -0.0234509473 -6.419941e-02 -0.057983027 -6.018579e-02
    -0.011823721 -0.0394933829 -0.0181182673 -0.050782980 -2.1334842
225 -0.027309786 -0.04464164 -0.0665634303 -1.123996e-01 -0.049727310 -4.139688e-02
    -0.000778808 -0.0394933829 -0.0358167281 -0.009361911 -75.1334842
226  0.030810830 0.05068012 0.0325952805 4.941532e-02 -0.040095640 -4.358892e-02
    0.069172310 0.0343088589 0.0630166151 0.003064409 55.8665158
227 -0.103593093 0.05068012 -0.0460850009 -2.632783e-02 -0.024960158 -2.480001e-02
    -0.030231910 -0.0394933829 -0.0398095944 -0.054925087 -75.1334842
228  0.067136214 0.05068012 -0.0299178198 5.744869e-02 -0.000193007 -1.571871e-02
    -0.074411564 -0.0505637191 -0.0384591123 0.007206516 -44.1334842
229 -0.052737555 -0.04464164 -0.0126728266 -6.075654e-02 -0.000193007 8.080576e-03
    -0.011823721 -0.0025922620 -0.0271286456 -0.050782980 7.8665158
230 -0.027309786 0.05068012 -0.0159062628 -2.977071e-02 0.003934852 -6.875805e-04
    -0.041276824 -0.0394933829 -0.0236445576 0.011348623 -99.1334842
```

```
231 -0.038207401 0.05068012 0.0713965152 -5.731367e-02 0.153913713 1.558867e-01
    -0.000778808 0.0719480022 0.0502764934 0.069338120 67.8665158
232  0.009015599 -0.04464164 -0.0309956318 2.187235e-02 0.008062710 8.706873e-03
    -0.004460446 -0.0025922620 0.0094364091 0.011348623 1.8665158
233  0.012648137 0.05068012 0.0002609183 -1.140873e-02 0.039709626 5.724488e-02
    0.039719208 0.0560805202 0.0240525832 0.032059158 106.8665158
234  0.067136214 -0.04464164 0.0369065288 -5.042793e-02 -0.023584206 -3.450761e-02
    -0.048640099 -0.0394933829 -0.0259524244 -0.038356660 -62.1334842
235  0.045340983 -0.04464164 0.0390621530 4.597245e-02 0.006686757 -2.417372e-02
    -0.008142084 -0.0125555646 0.0643282330 0.056911799 93.8665158
236  0.067136214 0.05068012 -0.0148284507 5.859631e-02 -0.059358980 -3.450761e-02
    0.061809035 0.0129062088 -0.0051453080 0.048627585 -28.1334842
237  0.027178291 -0.04464164 0.0067277908 3.564384e-02 0.079612259 7.071027e-02
    -0.015505359 0.0343088589 0.0406722637 0.011348623 -85.1334842
238  0.056238599 -0.04464164 -0.0687190544 -6.878991e-02 -0.000193007 -1.000729e-03
    -0.044958462 -0.0376483268 -0.0483617248 -0.001077698 -80.1334842
239  0.034443368 0.05068012 -0.0094393904 5.974393e-02 -0.035967781 -7.576847e-03
    0.076535586 0.0712099798 0.0110081010 -0.021788232 104.8665158
240  0.023545753 -0.04464164 0.0196615356 -1.255635e-02 0.083740117 3.876913e-02
    -0.063366651 -0.0025922620 0.0660482062 0.048627585 109.8665158
241  0.048973522 0.05068012 0.0746299514 6.662967e-02 -0.009824677 -2.253323e-03
    0.043400846 0.0343088589 0.0336568129 0.019632837 122.8665158
242  0.030810830 0.05068012 -0.0083615783 4.658002e-03 0.014942474 2.749578e-02
    -0.008142084 -0.0081274301 -0.0295276227 0.056911799 24.8665158
243 -0.103593093 0.05068012 -0.0234509473 -2.288496e-02 -0.086878037 -6.770135e-02
    0.017629381 -0.0394933829 -0.0781409107 -0.071493515 -81.1334842
244  0.016280676 0.05068012 -0.0460850009 1.154374e-02 -0.033215876 -1.603186e-02
    0.010266105 -0.0025922620 -0.0439854026 -0.042498767 -105.1334842
245 -0.060002632 0.05068012 0.0541515220 -1.944209e-02 -0.049727310 -4.891244e-02
    -0.022868635 -0.0394933829 -0.0439854026 -0.005219804 34.8665158
246 -0.027309786 -0.04464164 -0.0353068801 -2.977071e-02 -0.056607074 -5.862005e-02
    -0.030231910 -0.0394933829 -0.0498684677 -0.129483012 -27.1334842
247  0.041708445 -0.04464164 -0.0320734439 -6.190417e-02 0.079612259 5.098192e-02
    -0.056003375 -0.0099724862 0.0450661683 -0.059067194 -74.1334842
248 -0.081797862 -0.04464164 -0.0816527993 -4.009932e-02 0.002558899 -1.853704e-02
    -0.070729926 -0.0394933829 -0.0109044358 -0.092204050 -101.1334842
249 -0.041839939 -0.04464164 0.0476846496 5.974393e-02 0.127770609 1.280164e-01
```

```
    0.024992657 0.1081111006 0.0638931206 0.040343372 105.8665158
250 -0.012779632 -0.04464164 0.0606183944 5.285819e-02 0.047965343 2.937467e-02
    0.017629381 0.0343088589 0.0702112982 0.007206516 62.8665158
251  0.067136214 -0.04464164 0.0563071461 7.351542e-02 -0.013952536 -3.920484e-02
    0.032355932 -0.0025922620 0.0757375885 0.036201265 150.8665158
252 -0.052737555 0.05068012 0.0983418170 8.728690e-02 0.060348919 4.878988e-02
    0.058127397 0.1081111006 0.0844952822 0.040343372 90.8665158
253  0.005383060 -0.04464164 0.0595405824 -5.616605e-02 0.024574144 5.286081e-02
    0.043400846 0.0509143633 -0.0042198597 -0.030072446 -61.1334842
254  0.081666368 -0.04464164 0.0336730926 8.100872e-03 0.052093202 5.661859e-02
    0.017629381 0.0343088589 0.0348641931 0.069338120 -2.1334842
255  0.030810830 0.05068012 0.0563071461 7.695829e-02 0.049341296 -1.227407e-02
    0.036037570 0.0712099798 0.1200533820 0.090048655 157.8665158
256  0.001750522 -0.04464164 -0.0654856182 -5.670611e-03 -0.007072771 -1.947649e-02
    -0.041276824 -0.0394933829 -0.0033037126 0.007206516 0.8665158
257 -0.049105016 -0.04464164 0.1608549173 -4.698506e-02 -0.029088017 -1.978964e-02
    0.047082483 0.0343088589 0.0280165065 0.011348623 193.8665158
258 -0.027309786 0.05068012 -0.0557853095 2.531523e-02 -0.007072771 -2.354742e-02
    -0.052321737 -0.0394933829 -0.0051453080 -0.050782980 -89.1334842
259  0.078033829 0.05068012 -0.0245287594 -4.239456e-02 0.006686757 5.286081e-02
    0.069172310 0.0808042712 -0.0371283460 0.056911799 -63.1334842
260  0.012648137 -0.04464164 -0.0363846922 4.252958e-02 -0.013952536 1.293438e-02
    0.026833476 0.0051569734 -0.0439854026 0.007206516 -102.1334842
261  0.041708445 -0.04464164 -0.0083615783 -5.731367e-02 0.008062710 -3.137613e-02
    -0.151725958 -0.0763945038 -0.0802365402 -0.017646125 -113.1334842
262  0.048973522 -0.04464164 -0.0417737526 1.045013e-01 0.035581767 -2.573946e-02
    -0.177497423 -0.0763945038 -0.0129079423 0.015490730 -49.1334842
263 -0.016412170 0.05068012 0.1274427430 9.761551e-02 0.016318427 1.747503e-02
    0.021311019 0.0343088589 0.0348641931 0.003064409 155.8665158
264 -0.074532786 0.05068012 -0.0773415510 -4.698506e-02 -0.046975404 -3.262872e-02
    -0.004460446 -0.0394933829 -0.0721284546 -0.017646125 -36.1334842
265  0.034443368 0.05068012 0.0282840322 -3.321358e-02 -0.045599451 -9.768886e-03
    0.050764121 -0.0025922620 -0.0594726974 -0.021788232 -7.1334842
266 -0.034574863 0.05068012 -0.0256065715 -1.714685e-02 0.001182946 -2.879620e-03
    -0.008142084 -0.0155076543 0.0148227108 0.040343372 -78.1334842
267 -0.052737555 0.05068012 -0.0622521820 1.154374e-02 -0.008448724 -3.669965e-02
    -0.122272856 -0.0763945038 -0.0868289932 0.003064409 -107.1334842
```

```
268   0.059871137 -0.04464164 -0.0008168938 -8.485664e-02 0.075484400 7.947843e-02
    -0.004460446 0.0343088589 0.0233748413 0.027917051 -37.1334842
269   0.063503676 0.05068012 0.0886415084 7.007254e-02 0.020446286 3.751653e-02
    0.050764121 0.0712099798 0.0293004133 0.073480227 111.8665158
270   0.009015599 -0.04464164 -0.0320734439 -2.632783e-02 0.042461532 -1.039518e-02
    -0.159089234 -0.0763945038 -0.0119006848 -0.038356660 -65.1334842
271   0.005383060 0.05068012 0.0304396564 8.384403e-02 -0.037343734 -4.734670e-02
    -0.015505359 -0.0394933829 0.0086402829 0.015490730 49.8665158
272   0.038075906 0.05068012 0.0088834149 4.252958e-02 -0.042847546 -2.104223e-02
    0.039719208 -0.0025922620 -0.0181182673 0.007206516 -25.1334842
273   0.012648137 -0.04464164 0.0067277908 -5.616605e-02 -0.075870414 -6.644876e-02
    0.021311019 -0.0376483268 -0.0181182673 -0.092204050 29.8665158
274   0.074401291 0.05068012 -0.0202175111 4.597245e-02 0.074108447 3.281930e-02
    0.036037570 0.0712099798 0.1063542767 0.036201265 88.8665158
275   0.016280676 -0.04464164 -0.0245287594 3.564384e-02 -0.007072771 -3.192768e-03
    0.013947743 -0.0025922620 0.0155668445 0.015490730 -86.1334842
276  -0.005514555 0.05068012 -0.0115950145 1.154374e-02 -0.022208253 -1.540556e-02
    0.021311019 -0.0025922620 0.0110081010 0.069338120 -58.1334842
277   0.012648137 -0.04464164 0.0261284081 6.318680e-02 0.125018703 9.169122e-02
    -0.063366651 -0.0025922620 0.0575728562 -0.021788232 130.8665158
278  -0.034574863 -0.04464164 -0.0590187458 1.215131e-03 -0.053855168 -7.803525e-02
    -0.067048288 -0.0763945038 -0.0213936809 0.015490730 -88.1334842
279   0.067136214 0.05068012 -0.0363846922 -8.485664e-02 -0.007072771 1.966707e-02
    0.054445759 0.0343088589 0.0011437974 0.032059158 -50.1334842
280   0.038075906 0.05068012 -0.0245287594 4.658002e-03 -0.026336111 -2.636575e-02
    -0.015505359 -0.0394933829 -0.0159982678 -0.025930339 47.8665158
281   0.009015599 0.05068012 0.0185837236 3.908671e-02 0.017694380 1.058576e-02
    -0.019186997 -0.0025922620 0.0163049528 -0.017646125 112.8665158
282  -0.092695478 0.05068012 -0.0902752959 -5.731367e-02 -0.024960158 -3.043668e-02
    0.006584468 -0.0025922620 0.0240525832 0.003064409 -58.1334842
283   0.070768752 -0.04464164 -0.0051281421 -5.670611e-03 0.087867976 1.029646e-01
    -0.011823721 0.0343088589 -0.0089440190 0.027917051 77.8665158
284  -0.016412170 -0.04464164 -0.0525518733 -3.321358e-02 -0.044223498 -3.638651e-02
    -0.019186997 -0.0394933829 -0.0683297436 -0.030072446 28.8665158
285   0.041708445 0.05068012 -0.0223731352 2.875810e-02 -0.066238744 -4.515466e-02
    0.061809035 -0.0025922620 0.0028637705 -0.054925087 3.8665158
286   0.012648137 -0.04464164 -0.0202175111 -1.599922e-02 0.012190569 2.123281e-02
```

```
    0.076535586 0.1081111006 0.0598807231 -0.021788232 80.8665158
287 -0.038207401 -0.04464164 -0.0547074975 -7.797090e-02 -0.033215876 -8.649026e-02
    -0.140681045 -0.0763945038 -0.0191970476 -0.005219804 -92.1334842
288  0.045340983 -0.04464164 -0.0062059541 -1.599922e-02 0.125018703 1.251981e-01
    -0.019186997 0.0343088589 0.0324332258 -0.005219804 66.8665158
289  0.070768752 0.05068012 -0.0169840749 2.187235e-02 0.043837485 5.630544e-02
    -0.037595186 -0.0025922620 -0.0702093127 -0.017646125 -72.1334842
290 -0.074532786 0.05068012 0.0552293341 -4.009932e-02 0.053469155 5.317395e-02
    0.043400846 0.0712099798 0.0612379075 -0.034214553 -84.1334842
291  0.059871137 0.05068012 0.0767855756 2.531523e-02 0.001182946 1.684873e-02
    0.054445759 0.0343088589 0.0299356484 0.044485479 179.8665158
292  0.074401291 -0.04464164 0.0185837236 6.318680e-02 0.061724872 4.284006e-02
    -0.008142084 -0.0025922620 0.0580391277 -0.059067194 95.8665158
293  0.009015599 -0.04464164 -0.0223731352 -3.206595e-02 -0.049727310 -6.864080e-02
    -0.078093202 -0.0708593356 -0.0629129499 -0.038356660 -68.1334842
294 -0.070900247 -0.04464164 0.0929527567 1.269137e-02 0.020446286 4.252691e-02
    -0.000778808 0.0003598277 -0.0545441527 -0.001077698 47.8665158
295  0.023545753 0.05068012 -0.0309956318 -5.670611e-03 -0.016704441 1.778818e-02
    0.032355932 -0.0025922620 -0.0740888715 -0.034214553 -97.1334842
296 -0.052737555 0.05068012 0.0390621530 -4.009932e-02 -0.005696818 -1.290037e-02
    -0.011823721 -0.0394933829 0.0163049528 0.003064409 -67.1334842
297  0.067136214 -0.04464164 -0.0611743699 -4.009932e-02 -0.026336111 -2.448686e-02
    -0.033913548 -0.0394933829 -0.0561575731 -0.059067194 -63.1334842
298  0.001750522 -0.04464164 -0.0083615783 -6.419941e-02 -0.038719687 -2.448686e-02
    -0.004460446 -0.0394933829 -0.0646830225 -0.054925087 -121.1334842
299  0.023545753 0.05068012 -0.0374625043 -4.698506e-02 -0.091005896 -7.553006e-02
    0.032355932 -0.0394933829 -0.0307512099 -0.013504018 -23.1334842
300  0.038075906 0.05068012 -0.0137506387 -1.599922e-02 -0.035967781 -2.198168e-02
    0.013947743 -0.0025922620 -0.0259524244 -0.001077698 -69.1334842
301  0.016280676 -0.04464164 0.0735521393 -4.124694e-02 -0.004320866 -1.352667e-02
    0.013947743 -0.0011162172 0.0428956879 0.044485479 122.8665158
302 -0.001882017 0.05068012 -0.0245287594 5.285819e-02 0.027326050 3.000097e-02
    -0.030231910 -0.0025922620 -0.0213936809 0.036201265 -87.1334842
303  0.012648137 -0.04464164 0.0336730926 3.334859e-02 0.030077956 2.718263e-02
    0.002902830 0.0088470855 0.0311929907 0.027917051 45.8665158
304  0.074401291 -0.04464164 0.0347509047 9.417264e-02 0.057597013 2.029337e-02
    -0.022868635 -0.0025922620 0.0738021469 -0.021788232 83.8665158
```

```
305  0.041708445 0.05068012 -0.0385403164 5.285819e-02 0.076860353 1.164299e-01
    0.039719208 0.0712099798 -0.0225121719 -0.013504018 100.8665158
306 -0.009147093 0.05068012 -0.0396181284 -4.009932e-02 -0.008448724 1.622244e-02
    0.065490672 0.0712099798 0.0177634779 -0.067351408 -28.1334842
307  0.009015599 0.05068012 -0.0018947058 2.187235e-02 -0.038719687 -2.480001e-02
    0.006584468 -0.0394933829 -0.0398095944 -0.013504018 -108.1334842
308  0.067136214 0.05068012 -0.0309956318 4.658002e-03 0.024574144 3.563764e-02
    0.028674294 0.0343088589 0.0233748413 0.081764441 19.8665158
309  0.001750522 -0.04464164 -0.0460850009 -3.321358e-02 -0.073118508 -8.147988e-02
    -0.044958462 -0.0693832908 -0.0611765951 -0.079777729 -38.1334842
310 -0.009147093 0.05068012 0.0013387304 -2.227740e-03 0.079612259 7.008397e-02
    -0.033913548 -0.0025922620 0.0267142576 0.081764441 -10.1334842
311 -0.005514555 -0.04464164 0.0649296427 3.564384e-02 -0.001568960 1.496984e-02
    0.013947743 0.0007288389 -0.0181182673 0.032059158 -43.1334842
312  0.096196522 -0.04464164 0.0401399650 -5.731367e-02 0.045213437 6.068952e-02
    0.021311019 0.0361539149 0.0125531528 0.023774944 27.8665158
313 -0.074532786 -0.04464164 -0.0234509473 -5.670611e-03 -0.020832300 -1.415296e-02
    -0.015505359 -0.0394933829 -0.0384591123 -0.030072446 -8.1334842
314  0.059871137 0.05068012 0.0530737099 5.285819e-02 0.032829862 1.966707e-02
    0.010266105 0.0343088589 0.0552050381 -0.001077698 10.8665158
315 -0.023677247 -0.04464164 0.0401399650 -1.255635e-02 -0.009824677 -1.000729e-03
    0.002902830 -0.0025922620 -0.0119006848 -0.038356660 -5.1334842
316  0.009015599 -0.04464164 -0.0202175111 -5.387080e-02 0.031453909 2.060651e-02
    -0.056003375 -0.0394933829 -0.0109044358 -0.001077698 -55.1334842
317  0.016280676 0.05068012 0.0142724753 1.215131e-03 0.001182946 -2.135538e-02
    0.032355932 0.0343088589 0.0749683360 0.040343372 67.8665158
318  0.019913214 -0.04464164 -0.0342290681 5.515344e-02 0.067228683 7.415490e-02
    0.006584468 0.0328328140 0.0247253233 0.069338120 37.8665158
319  0.088931445 -0.04464164 0.0067277908 2.531523e-02 0.030077956 8.706873e-03
    -0.063366651 -0.0394933829 0.0094364091 0.032059158 -43.1334842
320  0.019913214 -0.04464164 0.0045721666 4.597245e-02 -0.018080394 -5.454912e-02
    -0.063366651 -0.0394933829 0.0286607203 0.061053906 38.8665158
321 -0.023677247 -0.04464164 0.0304396564 -5.670611e-03 0.082364165 9.200436e-02
    0.017629381 0.0712099798 0.0330470724 0.003064409 -30.1334842
322  0.096196522 -0.04464164 0.0519958979 7.925353e-02 0.054845107 3.657709e-02
    0.076535586 0.1413221094 0.0986463743 0.061053906 77.8665158
323  0.023545753 0.05068012 0.0616962065 6.203918e-02 0.024574144 -3.607336e-02
```

```
    0.091262137 0.1553445354 0.1333957338 0.081764441 89.8665158
324   0.070768752 0.05068012 -0.0072837662 4.941532e-02 0.060348919 -4.445362e-03
    0.054445759 0.1081111006 0.1290194116 0.056911799 95.8665158
325   0.030810830 -0.04464164 0.0056499787 1.154374e-02 0.078236306 7.791268e-02
    0.043400846 0.1081111006 0.0660482062 0.019632837 96.8665158
326  -0.001882017 -0.04464164 0.0541515220 -6.649466e-02 0.072732495 5.661859e-02
    0.043400846 0.0848633945 0.0844952822 0.048627585 39.8665158
327   0.045340983 0.05068012 -0.0083615783 -3.321358e-02 -0.007072771 1.191310e-03
    0.039719208 0.0343088589 0.0299356484 0.027917051 -21.1334842
328   0.074401291 -0.04464164 0.1145089981 2.875810e-02 0.024574144 2.499059e-02
    -0.019186997 -0.0025922620 -0.0006092542 -0.005219804 84.8665158
329  -0.038207401 -0.04464164 0.0670852669 -6.075654e-02 -0.029088017 -2.323427e-02
    0.010266105 -0.0025922620 -0.0014985868 0.019632837 -74.1334842
330  -0.012779632 0.05068012 -0.0557853095 -2.227740e-03 -0.027712064 -2.918409e-02
    -0.019186997 -0.0394933829 -0.0170521046 0.044485479 -17.1334842
331   0.009015599 0.05068012 0.0304396564 4.252958e-02 -0.002944913 3.689023e-02
    0.065490672 0.0712099798 -0.0236445576 0.015490730 91.8665158
332   0.081666368 0.05068012 -0.0256065715 -3.665645e-02 -0.070366603 -4.640726e-02
    0.039719208 -0.0025922620 -0.0411803852 -0.005219804 46.8665158
333   0.030810830 -0.04464164 0.1048086895 7.695829e-02 -0.011200630 -1.133463e-02
    0.058127397 0.0343088589 0.0571041874 0.036201265 117.8665158
334   0.027178291 0.05068012 -0.0062059541 2.875810e-02 -0.016704441 -1.627026e-03
    0.058127397 0.0343088589 0.0293004133 0.032059158 11.8665158
335  -0.060002632 0.05068012 -0.0471628129 -2.288496e-02 -0.071742556 -5.768060e-02
    0.006584468 -0.0394933829 -0.0629129499 -0.054925087 -80.1334842
336   0.005383060 -0.04464164 -0.0482406250 -1.255635e-02 0.001182946 -6.637401e-03
    -0.063366651 -0.0394933829 -0.0514005353 -0.059067194 -56.1334842
337  -0.020044709 -0.04464164 0.0854080721 -3.665645e-02 0.091995835 8.949918e-02
    0.061809035 0.1450122215 0.0809479135 0.052769692 153.8665158
338   0.019913214 0.05068012 -0.0126728266 7.007254e-02 -0.011200630 7.141131e-03
    0.039719208 0.0343088589 0.0053843700 0.003064409 -61.1334842
339  -0.063635170 -0.04464164 -0.0331512560 -3.321358e-02 0.001182946 2.405115e-02
    0.024992657 -0.0025922620 -0.0225121719 -0.059067194 61.8665158
340   0.027178291 -0.04464164 -0.0072837662 -5.042793e-02 0.075484400 5.661859e-02
    -0.033913548 -0.0025922620 0.0434431723 0.015490730 -57.1334842
341  -0.016412170 -0.04464164 -0.0137506387 1.320442e-01 -0.009824677 -3.819065e-03
    -0.019186997 -0.0394933829 -0.0358167281 -0.030072446 63.8665158
```

```
342  0.030810830 0.05068012 0.0595405824 5.630106e-02 -0.022208253 1.191310e-03
    0.032355932 -0.0025922620 -0.0247911874 -0.017646125 110.8665158
343  0.056238599 0.05068012 0.0218171598 5.630106e-02 -0.007072771 1.810133e-02
    0.032355932 -0.0025922620 -0.0236445576 0.023774944 25.8665158
344 -0.020044709 -0.04464164 0.0185837236 9.072977e-02 0.003934852 8.706873e-03
    -0.037595186 -0.0394933829 -0.0578000657 0.007206516 -39.1334842
345 -0.107225632 -0.04464164 -0.0115950145 -4.009932e-02 0.049341296 6.444730e-02
    0.013947743 0.0343088589 0.0070268625 -0.030072446 47.8665158
346  0.081666368 0.05068012 -0.0029725179 -3.321358e-02 0.042461532 5.787118e-02
    0.010266105 0.0343088589 -0.0006092542 -0.001077698 -13.1334842
347  0.005383060 0.05068012 0.0175059115 3.220097e-02 0.127770609 1.273901e-01
    0.021311019 0.0712099798 0.0625751815 0.015490730 -13.1334842
348  0.038075906 0.05068012 -0.0299178198 -7.452802e-02 -0.012576583 -1.258722e-02
    -0.004460446 -0.0025922620 0.0037117382 -0.030072446 -64.1334842
349  0.030810830 -0.04464164 -0.0202175111 -5.670611e-03 -0.004320866 -2.949724e-02
    -0.078093202 -0.0394933829 -0.0109044358 -0.001077698 -4.1334842
350  0.001750522 0.05068012 -0.0579409337 -4.354219e-02 -0.096509707 -4.703355e-02
    0.098625413 0.0343088589 -0.0611765951 -0.071493515 -64.1334842
351 -0.027309786 0.05068012 0.0606183944 1.079441e-01 0.012190569 -1.759760e-02
    0.002902830 -0.0025922620 0.0702112982 0.135611831 90.8665158
352 -0.085430401 0.05068012 -0.0406959405 -3.321358e-02 -0.081374226 -6.958024e-02
    0.006584468 -0.0394933829 -0.0578000657 -0.042498767 -81.1334842
353  0.012648137 0.05068012 -0.0719524906 -4.698506e-02 -0.051103263 -9.713731e-02
    -0.118591218 -0.0763945038 -0.0202887478 -0.038356660 -75.1334842
354 -0.052737555 -0.04464164 -0.0557853095 -3.665645e-02 0.089243929 -3.192768e-03
    -0.008142084 0.0343088589 0.1323726493 0.003064409 -43.1334842
355 -0.023677247 0.05068012 0.0455290254 2.187235e-02 0.109883222 8.887288e-02
    -0.000778808 0.0343088589 0.0741925367 0.061053906 119.8665158
356 -0.074532786 0.05068012 -0.0094393904 1.498661e-02 -0.037343734 -2.166853e-02
    0.013947743 -0.0025922620 -0.0332487872 0.011348623 -92.1334842
357 -0.005514555 0.05068012 -0.0331512560 -1.599922e-02 0.008062710 1.622244e-02
    -0.015505359 -0.0025922620 -0.0283202425 -0.075635622 -98.1334842
358 -0.060002632 0.05068012 0.0498402737 1.842948e-02 -0.016704441 -3.012354e-02
    0.017629381 -0.0025922620 0.0497686599 -0.059067194 68.8665158
359 -0.020044709 -0.04464164 -0.0848862355 -2.632783e-02 -0.035967781 -3.419447e-02
    -0.041276824 -0.0516707528 -0.0823814833 -0.046640874 -62.1334842
360  0.038075906 0.05068012 0.0056499787 3.220097e-02 0.006686757 1.747503e-02
```

```
    0.024992657 0.0343088589 0.0148227108 0.061053906 158.8665158
361   0.016280676 -0.04464164 0.0207393477 2.187235e-02 -0.013952536 -1.321352e-02
    0.006584468 -0.0025922620 0.0133159679 0.040343372 128.8665158
362   0.041708445 -0.04464164 -0.0072837662 2.875810e-02 -0.042847546 -4.828615e-02
    -0.052321737 -0.0763945038 -0.0721284546 0.023774944 29.8665158
363   0.019913214 0.05068012 0.1048086895 7.007254e-02 -0.035967781 -2.667890e-02
    0.024992657 -0.0025922620 0.0037117382 0.040343372 168.8665158
364  -0.049105016 0.05068012 -0.0245287594 6.750728e-05 -0.046975404 -2.824465e-02
    0.065490672 0.0284046795 0.0191990331 0.011348623 -94.1334842
365   0.001750522 0.05068012 -0.0062059541 -1.944209e-02 -0.009824677 4.949092e-03
    0.039719208 0.0343088589 0.0148227108 0.098332868 109.8665158
366   0.034443368 -0.04464164 -0.0385403164 -1.255635e-02 0.009438663 5.262240e-03
    0.006584468 -0.0025922620 0.0311929907 0.098332868 53.8665158
367  -0.045472478 0.05068012 0.1371430517 -1.599922e-02 0.041085579 3.187986e-02
    0.043400846 0.0712099798 0.0710215779 0.048627585 80.8665158
368  -0.009147093 0.05068012 0.1705552260 1.498661e-02 0.030077956 3.375875e-02
    0.021311019 0.0343088589 0.0336568129 0.032059158 89.8665158
369  -0.016412170 0.05068012 0.0024165425 1.498661e-02 0.021822239 -1.008203e-02
    0.024992657 0.0343088589 0.0855331212 0.081764441 -29.1334842
370  -0.009147093 -0.04464164 0.0379843409 -4.009932e-02 -0.024960158 -3.819065e-03
    0.043400846 0.0158582984 -0.0051453080 0.027917051 14.8665158
371   0.019913214 -0.04464164 -0.0579409337 -5.731367e-02 -0.001568960 -1.258722e-02
    -0.074411564 -0.0394933829 -0.0611765951 -0.075635622 -89.1334842
372   0.052606060 0.05068012 -0.0094393904 4.941532e-02 0.050717249 -1.916334e-02
    0.013947743 0.0343088589 0.1193439942 -0.017646125 44.8665158
373  -0.027309786 0.05068012 -0.0234509473 -1.599922e-02 0.013566522 1.277780e-02
    -0.026550273 -0.0025922620 -0.0109044358 -0.021788232 -81.1334842
374  -0.074532786 -0.04464164 -0.0105172024 -5.670611e-03 -0.066238744 -5.705430e-02
    0.002902830 -0.0394933829 -0.0425721049 -0.001077698 15.8665158
375  -0.107225632 -0.04464164 -0.0342290681 -6.764228e-02 -0.063486838 -7.051969e-02
    -0.008142084 -0.0394933829 -0.0006092542 -0.079777729 -12.1334842
376   0.045340983 0.05068012 -0.0029725179 1.079441e-01 0.035581767 2.248541e-02
    -0.026550273 -0.0025922620 0.0280165065 0.019632837 64.8665158
377  -0.001882017 -0.04464164 0.0681630790 -5.670611e-03 0.119514892 1.302085e-01
    0.024992657 0.0867084505 0.0461323310 -0.001077698 -31.1334842
378   0.019913214 0.05068012 0.0099612270 1.842948e-02 0.014942474 4.471895e-02
    0.061809035 0.0712099798 0.0094364091 -0.063209301 82.8665158
```

```
379  0.016280676 0.05068012 0.0024165425 -5.670611e-03 -0.005696818 1.089891e-02
    0.050764121 0.0343088589 0.0226920226 -0.038356660 92.8665158
380 -0.001882017 -0.04464164 -0.0385403164 2.187235e-02 -0.108893283 -1.156131e-01
    -0.022868635 -0.0763945038 -0.0468794828 0.023774944 -112.1334842
381  0.016280676 -0.04464164 0.0261284081 5.859631e-02 -0.060734933 -4.421522e-02
    0.013947743 -0.0339582147 -0.0514005353 -0.025930339 -100.1334842
382 -0.070900247 0.05068012 -0.0891974838 -7.452802e-02 -0.042847546 -2.573946e-02
    0.032355932 -0.0025922620 -0.0129079423 -0.054925087 -48.1334842
383  0.048973522 -0.04464164 0.0606183944 -2.288496e-02 -0.023584206 -7.271173e-02
    0.043400846 -0.0025922620 0.1041376114 0.036201265 -20.1334842
384  0.005383060 0.05068012 -0.0288400077 -9.113481e-03 -0.031839923 -2.887094e-02
    -0.008142084 -0.0394933829 -0.0181182673 0.007206516 -64.1334842
385  0.034443368 0.05068012 -0.0299178198 4.658002e-03 0.093371787 8.699399e-02
    -0.033913548 -0.0025922620 0.0240525832 -0.038356660 -83.1334842
386  0.023545753 0.05068012 -0.0191396990 4.941532e-02 -0.063486838 -6.112523e-02
    -0.004460446 -0.0394933829 -0.0259524244 -0.013504018 66.8665158
387  0.019913214 -0.04464164 -0.0406959405 -1.599922e-02 -0.008448724 -1.759760e-02
    -0.052321737 -0.0394933829 -0.0307512099 0.003064409 -80.1334842
388 -0.045472478 -0.04464164 0.0153502873 -7.452802e-02 -0.049727310 -1.728445e-02
    0.028674294 -0.0025922620 -0.1043648208 -0.075635622 48.8665158
389  0.052606060 0.05068012 -0.0245287594 5.630106e-02 -0.007072771 -5.071659e-03
    0.021311019 -0.0025922620 0.0267142576 -0.038356660 -42.1334842
390 -0.005514555 0.05068012 0.0013387304 -8.485664e-02 -0.011200630 -1.665815e-02
    -0.048640099 -0.0394933829 -0.0411803852 -0.088061943 -101.1334842
391  0.009015599 0.05068012 0.0692408910 5.974393e-02 0.017694380 -2.323427e-02
    0.047082483 0.0343088589 0.1032922649 0.073480227 124.8665158
392 -0.023677247 -0.04464164 -0.0697968665 -6.419941e-02 -0.059358980 -5.047819e-02
    -0.019186997 -0.0394933829 -0.0891368601 -0.050782980 -89.1334842
393 -0.041839939 0.05068012 -0.0299178198 -2.227740e-03 0.021822239 3.657709e-02
    -0.011823721 -0.0025922620 -0.0411803852 0.065196013 -34.1334842
394 -0.074532786 -0.04464164 -0.0460850009 -4.354219e-02 -0.029088017 -2.323427e-02
    -0.015505359 -0.0394933829 -0.0398095944 -0.021788232 -83.1334842
395  0.034443368 -0.04464164 0.0185837236 5.630106e-02 0.012190569 -5.454912e-02
    0.069172310 0.0712099798 0.1300806095 0.007206516 120.8665158
396 -0.060002632 -0.04464164 0.0013387304 -2.977071e-02 -0.007072771 -2.166853e-02
    -0.011823721 -0.0025922620 0.0318152175 -0.054925087 105.8665158
397 -0.085430401 0.05068012 -0.0309956318 -2.288496e-02 -0.063486838 -5.423597e-02
```

```
   -0.019186997 -0.0394933829 -0.0964332229 -0.034214553 -109.1334842
398  0.052606060 -0.04464164 -0.0040503300 -3.091833e-02 -0.046975404 -5.830690e-02
   0.013947743 -0.0258399682 0.0360557901 0.023774944 45.8665158
399  0.012648137 -0.04464164 0.0153502873 -3.321358e-02 0.041085579 3.219301e-02
   0.002902830 -0.0025922620 0.0450661683 -0.067351408 89.8665158
400  0.059871137 0.05068012 0.0228949719 4.941532e-02 0.016318427 1.183836e-02
   0.013947743 -0.0025922620 0.0395398781 0.019632837 79.8665158
401 -0.023677247 -0.04464164 0.0455290254 9.072977e-02 -0.018080394 -3.544706e-02
   -0.070729926 -0.0394933829 -0.0345237153 -0.009361911 22.8665158
402  0.016280676 -0.04464164 -0.0450071888 -5.731367e-02 -0.034591828 -5.392282e-02
   -0.074411564 -0.0763945038 -0.0425721049 0.040343372 -59.1334842
403  0.110726675 0.05068012 -0.0331512560 -2.288496e-02 -0.004320866 2.029337e-02
   0.061809035 0.0712099798 0.0155668445 0.044485479 15.8665158
404 -0.020044709 -0.04464164 0.0972640050 -5.670611e-03 -0.005696818 -2.386057e-02
   0.021311019 -0.0025922620 0.0616858488 0.040343372 122.8665158
405 -0.016412170 -0.04464164 0.0541515220 7.007254e-02 -0.033215876 -2.793150e-02
   -0.008142084 -0.0394933829 -0.0271286456 -0.009361911 140.8665158
406  0.048973522 0.05068012 0.1231314947 8.384403e-02 -0.104765424 -1.008951e-01
   0.069172310 -0.0025922620 0.0366457978 -0.030072446 128.8665158
407 -0.056370093 -0.04464164 -0.0805749872 -8.485664e-02 -0.037343734 -3.701280e-02
   -0.033913548 -0.0394933829 -0.0561575731 -0.137767226 -80.1334842
408  0.027178291 -0.04464164 0.0929527567 -5.272318e-02 0.008062710 3.970857e-02
   0.028674294 0.0210244554 -0.0483617248 0.019632837 -12.1334842
409  0.063503676 -0.04464164 -0.0503962492 1.079441e-01 0.031453909 1.935392e-02
   0.017629381 0.0236075338 0.0580391277 0.040343372 36.8665158
410 -0.052737555 0.05068012 -0.0115950145 5.630106e-02 0.056221060 7.290231e-02
   0.039719208 0.0712099798 0.0305664874 -0.005219804 28.8665158
411 -0.009147093 0.05068012 -0.0277621956 8.100872e-03 0.047965343 3.720338e-02
   0.028674294 0.0343088589 0.0660482062 -0.042498767 56.8665158
412  0.005383060 -0.04464164 0.0584627703 -4.354219e-02 -0.073118508 -7.239858e-02
   -0.019186997 -0.0763945038 -0.0514005353 -0.025930339 -16.1334842
413  0.074401291 -0.04464164 0.0854080721 6.318680e-02 0.014942474 1.309095e-02
   -0.015505359 -0.0025922620 0.0062093156 0.085906548 108.8665158
414 -0.052737555 -0.04464164 -0.0008168938 -2.632783e-02 0.010814616 7.141131e-03
   -0.048640099 -0.0394933829 -0.0358167281 0.019632837 -39.1334842
415  0.081666368 0.05068012 0.0067277908 -4.522987e-03 0.109883222 1.170562e-01
   0.032355932 0.0918746074 0.0547240033 0.007206516 -21.1334842
```

```
416 -0.005514555 -0.04464164 0.0088834149 -5.042793e-02 0.025950097 4.722413e-02
    0.043400846 0.0712099798 0.0148227108 0.003064409 21.8665158
417 -0.027309786 -0.04464164 0.0800190118 9.876313e-02 -0.002944913 1.810133e-02
    0.017629381 0.0033119173 -0.0295276227 0.036201265 104.8665158
418 -0.052737555 -0.04464164 0.0713965152 -7.452802e-02 -0.015328488 -1.313877e-03
    -0.004460446 -0.0214118336 -0.0468794828 0.003064409 -97.1334842
419  0.009015599 -0.04464164 -0.0245287594 -2.632783e-02 0.098875599 9.419640e-02
    -0.070729926 -0.0025922620 -0.0213936809 0.007206516 -68.1334842
420 -0.020044709 -0.04464164 -0.0547074975 -5.387080e-02 -0.066238744 -5.736745e-02
    -0.011823721 -0.0394933829 -0.0740888715 -0.005219804 -110.1334842
421  0.023545753 -0.04464164 -0.0363846922 6.750728e-05 0.001182946 3.469820e-02
    0.043400846 0.0343088589 -0.0332487872 0.061053906 -6.1334842
422  0.038075906 0.05068012 0.0164280994 2.187235e-02 0.039709626 4.503209e-02
    0.043400846 0.0712099798 0.0497686599 0.015490730 59.8665158
423 -0.078165324 0.05068012 0.0778633876 5.285819e-02 0.078236306 6.444730e-02
    -0.026550273 -0.0025922620 0.0406722637 -0.009361911 80.8665158
424  0.009015599 0.05068012 -0.0396181284 2.875810e-02 0.038333673 7.352860e-02
    0.072853948 0.1081111006 0.0155668445 -0.046640874 -61.1334842
425  0.001750522 0.05068012 0.0110390390 -1.944209e-02 -0.016704441 -3.819065e-03
    0.047082483 0.0343088589 0.0240525832 0.023774944 -41.1334842
426 -0.078165324 -0.04464164 -0.0406959405 -8.141377e-02 -0.100637566 -1.127947e-01
    -0.022868635 -0.0763945038 -0.0202887478 -0.050782980 -0.1334842
427  0.030810830 0.05068012 -0.0342290681 4.367720e-02 0.057597013 6.883138e-02
    0.032355932 0.0575565650 0.0354619387 0.085906548 -32.1334842
428 -0.034574863 0.05068012 0.0056499787 -5.670611e-03 -0.073118508 -6.269098e-02
    0.006584468 -0.0394933829 -0.0454209578 0.032059158 -85.1334842
429  0.048973522 0.05068012 0.0886415084 8.728690e-02 0.035581767 2.154596e-02
    0.024992657 0.0343088589 0.0660482062 0.131469724 157.8665158
430 -0.041839939 -0.04464164 -0.0331512560 -2.288496e-02 0.046589390 4.158746e-02
    -0.056003375 -0.0247329345 -0.0259524244 -0.038356660 -58.1334842
431 -0.009147093 -0.04464164 -0.0568631216 -5.042793e-02 0.021822239 4.534524e-02
    0.028674294 0.0343088589 -0.0099189574 -0.017646125 30.8665158
432  0.070768752 0.05068012 -0.0309956318 2.187235e-02 -0.037343734 -4.703355e-02
    -0.033913548 -0.0394933829 -0.0149564750 -0.001077698 -86.1334842
433  0.009015599 -0.04464164 0.0552293341 -5.670611e-03 0.057597013 4.471895e-02
    0.002902830 0.0232385226 0.0556835477 0.106617082 20.8665158
434 -0.027309786 -0.04464164 -0.0600965578 -2.977071e-02 0.046589390 1.998022e-02
```

```
    -0.122272856 -0.0394933829 -0.0514005353 -0.009361911 -80.1334842
435  0.016280676 -0.04464164 0.0013387304 8.100872e-03 0.005310804 1.089891e-02
    -0.030231910 -0.0394933829 -0.0454209578 0.032059158 -103.1334842
436 -0.012779632 -0.04464164 -0.0234509473 -4.009932e-02 -0.016704441 4.635943e-03
    0.017629381 -0.0025922620 -0.0384591123 -0.038356660 -88.1334842
437 -0.056370093 -0.04464164 -0.0741081148 -5.042793e-02 -0.024960158 -4.703355e-02
    -0.092819753 -0.0763945038 -0.0611765951 -0.046640874 -104.1334842
438  0.041708445 0.05068012 0.0196615356 5.974393e-02 -0.005696818 -2.566471e-03
    0.028674294 -0.0025922620 0.0311929907 0.007206516 25.8665158
439 -0.005514555 0.05068012 -0.0159062628 -6.764228e-02 0.049341296 7.916528e-02
    0.028674294 0.0343088589 -0.0181182673 0.044485479 -48.1334842
440  0.041708445 0.05068012 -0.0159062628 1.728186e-02 -0.037343734 -1.383982e-02
    0.024992657 -0.0110795198 -0.0468794828 0.015490730 -20.1334842
441 -0.045472478 -0.04464164 0.0390621530 1.215131e-03 0.016318427 1.528299e-02
    0.028674294 0.0265596235 0.0445283740 -0.025930339 67.8665158
442 -0.045472478 -0.04464164 -0.0730303027 -8.141377e-02 0.083740117 2.780893e-02
    -0.173815785 -0.0394933829 -0.0042198597 0.003064409 -95.1334842;
```