Level Set Clustering with Kernel Density Estimator & Spectral Clustering on data600

Bryan Ng
University of Washington
Department of Statistics
Seattle, WA 98105 USA
ngcw0303@uw.edu

1 Introduction

In this project, we applied two nonparametric clustering methods to the `data600` dataset, which contains $n = 10{,}000$ observations in $d = 600$ dimensions. The first method is level-set clustering using a kernel density estimator, and using nearest neighbor to assign low density points to clusters. The second method is spectral clustering. To avoid the curse of dimensionality and improve clustering performance, we reduced the data's dimensionality by principal component analysis (PCA), random projection (RP), and Uniform Manifold Approximation and Projection (UMAP).

2 Theory about Dimension Reduction and Clustering Methods

2.1 Random Projecting with Gaussian

Suppose the original dataset is

$$X = \{\, x_1,\, x_2,\, \dots,\, x_N \} \subset \mathbb{R}^d,$$

and we wish to project these points into $\mathbb{R}^k$ (with $k \ll d$). Random Project is to construct a random matrix R of size $k \times d$, and then map each sample $x \in \mathbb{R}^d$ to

$$y = \frac{1}{k} R\, x \ \in\ \mathbb{R}^k$$

$$R_{ij} \sim \mathcal{N}(0, 1)$$

Here, the factor $\frac{1}{k}$ is a normalization constant to help preserve the expected length of the projected vectors.

The Johnson–Lindenstrauss lemma [1] [2] states that, in order for a random projection to preserve all pairwise Euclidean distances among N points up to a relative error of at most ε, the target dimension k must satisfy

$$k \geq C\, \frac{\ln N}{\varepsilon^2},$$

where $C > 0$ is an absolute constant. Equivalently, one often sees the more precise form:

$$k \geq \frac{4\, \ln N}{\varepsilon^2/2\ -\ \varepsilon^3/3}.$$

Then, with probability at least $1 - \frac{1}{N^2}$, for every pair $i < j$ one has

$$(1 - \varepsilon)\, \|x_i - x_j\|_2^2 \ \leq\ \left\| \tfrac{1}{\sqrt{k}}\, R\, x_i - \tfrac{1}{\sqrt{k}}\, R\, x_j \right\|_2^2 \ \leq\ (1 + \varepsilon)\, \|x_i - x_j\|_2^2$$

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2.2 Kernel Density Estimator in High Dimension

Let $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ be a sample from a distribution F with density f, and x be a d-dimensional target point at which we want to estimate $f(x)$. The general multivariate kernel density estimator [3] is:

$$\hat{f}_{\mathbf{H}}(x) = \frac{1}{n\det(\mathbf{H})}\sum_{i=1}^{n}\mathcal{K}\left(\mathbf{H}^{-1}(x - X_i)\right)$$

where $\mathcal{K}$ is a symmetric kernel with $\int \mathcal{K}(u)du = 1$ and $\int u\mathcal{K}(u)du = 0_d$, $\mathbf{H} \in \mathbb{R}^{d\times d}$ is a symmetric positive definite bandwidth matrix. $\mathcal{K}$ is a product kernel and $\mathbf{H} = \text{diag}(h_1, \ldots, h_d)$ is diagonal, so that we can express the estimator as:

$$\begin{aligned}\hat{f}_{\mathbf{H}}(x) &= \frac{1}{n\ \det(\mathbf{H})}\sum_{i=1}^{n}\mathcal{K}\Big(\mathbf{H}^{-1}(x - X_i)\Big)\\ &= \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{d}\frac{1}{h_j}\mathcal{K}\Big(\tfrac{x_j - X_{ij}}{h_j}\Big)\\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h^d}\mathcal{K}\Big(\tfrac{x_j - X_{ij}}{h_j}\Big)\end{aligned}$$

For bandwidth selection, let $\hat{\sigma} = \left(\prod_{j=1}^{d}\hat{\sigma}_j\right)^{1/d}$, where $\hat{\sigma}_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{i,j} - \bar{x}_j\right)^2}$. Then silverman's rule of thumb [4] for an gaussian kernel is:

$$h_{\text{opt}} \ = \ \left(\frac{4}{d+2}\right)^{1/(d+4)} \quad n^{-\,1/(d+4)}\ \hat{\sigma}$$

2.3 Level Set Clustering

Level set clustering [5] is a density-based clustering method. Consider the λ-upper level set:

$$L_\lambda \ = \ \{\, x : p(x) \geq \lambda \,\}$$

Namely, L_λ is the collection of regions where the PDF is greater than or equal to λ. When p has multiple local modes, some level sets would lead to several connected components. The level set clustering groups observations into the same cluster if they all lie in the same connected component.

Again, in practice we estimate the PDF by a density estimator and use the estimated level set to perform clustering. Namely, suppose that $\hat{p}$ is a density estimator. Then we use

$$\widehat{L}_\lambda \ = \ \{\, x : \hat{p}(x) \geq \lambda \,\}$$

as an estimator of L_λ, and the corresponding connected components of $\widehat{L}_\lambda$ to cluster observations.

2.4 Spectral Clustering

The spectral clustering [6] is another very popular clustering method. It first converts observations into a graph such that the nodes are the observations and edges represent the similarity between pairs of observations and then forms a graph Laplacian to perform clustering.

For the similarity matrix $S \in \mathbb{R}^{n\times n}$, we define $S_{ij} = 1$ if X_i is among the k-NN of X_j or X_j is among the k-NN of X_i.

With a similarity matrix S, we construct a degree matrix D that is a diagonal matrix with

$$D_{ii} = \sum_{j=1}^{n} S_{ij}$$

The Symmetric Normalized Laplacian is

$$L \;=\; D^{-1/2}\,L\,D^{-1/2} \;=\; I \;-\; D^{-1/2}\,W\,D^{-1/2}.$$

We computed k eigenvectors corresponding to the k smallest eigenvalues of Laplacian matrix L, then we form the matrix

$$U \;=\; \bigl[\,u_n, \ldots, u_{n-k+1}\,\bigr] \;\in\; \mathbb{R}^{n\times k}$$

Each data point is then represented by the corresponding row of U: for $i = 1, \ldots, n$, set

$$Y_i \;=\; U_{i,\cdot},$$

so that $\{Y_i\} \subset \mathbb{R}^k$. Finally, we apply k-means clustering to $\{Y_1, \ldots, Y_n\}$, and the resulting clusters in this low-dimensional space become the final clusters for the original data.

2.5 Silhouette Score

The Silhouette Score [7] is a metric for evaluating clustering performance. Suppose you have a data set of n observations $\{x_1, \ldots, x_n\}$ in some metric space $(\mathcal{X}, d)$, and these points have been partitioned into K disjoint clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$. For each point x_i, define: Let $\mathcal{C}(i)$ denote the cluster to which x_i belongs. Then

$$a(i) \;=\; \frac{1}{|\mathcal{C}(i)| \;-\; 1} \sum_{\substack{x_j \in \mathcal{C}(i)\\ j\neq i}} d\bigl(x_i,\, x_j\bigr)$$

In words, $a(i)$ is the average distance from x_i to all the other points in its own cluster. (If $|\mathcal{C}(i)| = 1$, one typically sets $a(i) = 0$ by convention, though in practice one usually avoids singleton clusters when computing silhouettes.)

For each cluster index $\ell \neq \mathcal{C}(i)$, define

$$d_\ell(i) \;=\; \frac{1}{|\mathcal{C}_\ell|} \sum_{x_j \in \mathcal{C}_\ell} d\bigl(x_i,\, x_j\bigr)$$

That is, $d_\ell(i)$ is the average distance from x_i to all points in cluster $\mathcal{C}_\ell$. Then

$$b(i) \;=\; \min_{\ell \neq \mathcal{C}(i)} \; d_\ell(i)$$

In other words, $b(i)$ is the smallest average distance from x_i to any other cluster (i.e. the “best alternative” cluster). Call the cluster achieving the minimum in $b(i)$ the “nearest neighboring cluster” of x_i.

Finally, define the point-wise silhouette coefficient

$$s(i) \;=\; \frac{b(i) \;-\; a(i)}{\max\{\,a(i),\, b(i)\}}.$$

The global silhouette score is the average over all n points:

$$\bar{s} \;=\; \frac{1}{n}\sum_{i=1}^{n} s(i), \quad -1 < \bar{s} < 1$$

The Silhouette score ranges from –1 to 1. A value close to +1 indicates that a sample is well matched to its own cluster and poorly matched to neighboring clusters, implying good clustering. A value around 0 means the sample lies near the boundary between clusters, so the clustering is ambiguous. A value near –1 suggests the sample would be better placed in a different cluster, indicating a poor clustering result.

3 Method 1 Level Set Clustering

3.1 Dimensionality Reduction

In the first stage of dimensionality reduction, we applied Gaussian random projection. Initially, we aimed to preserve pairwise distances as accurately as possible by setting $\varepsilon = 0.1$. However, the Johnson– Lindenstrauss bound

$$k \;\geq\; \frac{4\ln(10{,}000)}{\frac{0.1^2}{2} \;-\; \frac{0.1^3}{3}} \;=\; \frac{4 \times \ln(10^4)}{0.005 \;-\; 0.000333} \;=\; \frac{4 \times 9.21034}{0.004667} \;=\; \frac{36.8414}{0.004667} \;\approx\; 7{,}900$$

far exceeded our original dimension $d = 600$. Because our sample size $n = 10{,}000$ is relatively small, we chose instead to preserve the cluster structure rather than the pairwise distances. By increasing ε to 0.7, the boundary is $k \approx 282$, so we reduced d from 600 down to 300 by random projection.

In the second stage, we applied principal component analysis (PCA) to remove noisy directions while retaining most of the signal, thereby reducing the dimension from 300 to 50. Finally, we used UMAP to capture the local neighborhood structure in a very low-dimensional space, embedding into three dimensions in order to avoid the curse of dimensionality when we later apply the kernel density estimator.

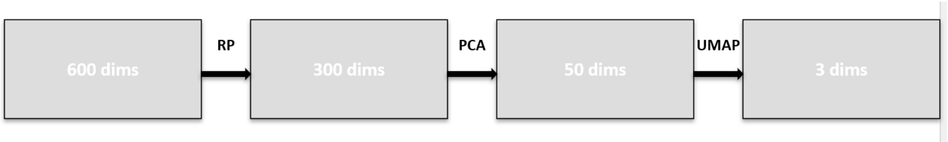

Figure 1: Dimensionality Reduction Pipeline for Method 1

3.2 Kernel Density Estimator

In this analysis, we employed Silverman's rule of thumb to select the bandwidth:

$$h \;=\; \left(\frac{4}{d+2}\right)^{1/(d+4)} \; n^{-\,1/(d+4)}\;\hat{\sigma} \approx 0.34$$

To further smooth the KDE, we introduced a tuning hyperparameter $alpha$. After careful fine-tuning, we set $\alpha = 1.3$, yielding an optimal bandwidth of

$$h_{\text{opt}} = \alpha \times h \approx 0.44$$

After fitting the KDE, we plotted its 2D contour.

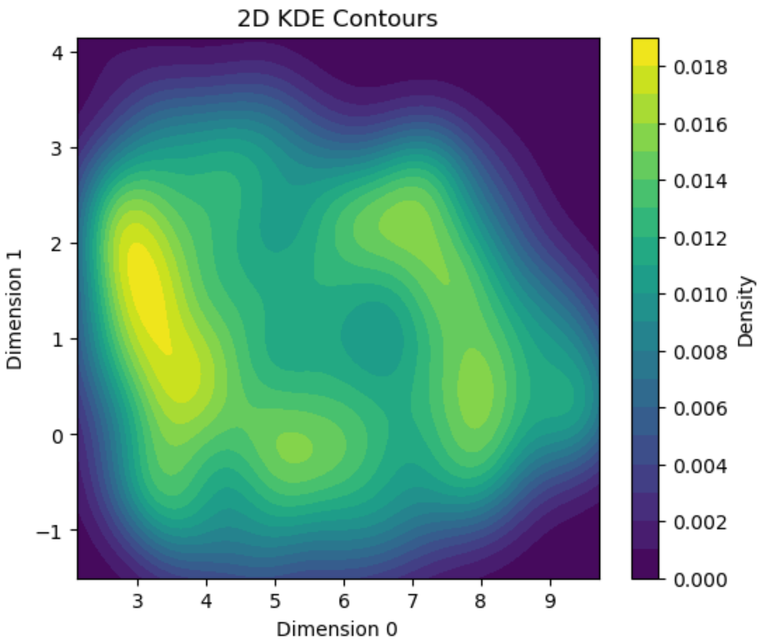

Figure 2: The 2D contour of the KDE

Clearly, two distinct density peaks are visible on the 2D contour, so we proceed to apply level set clustering for further analysis.

3.3 Level Set Clustering

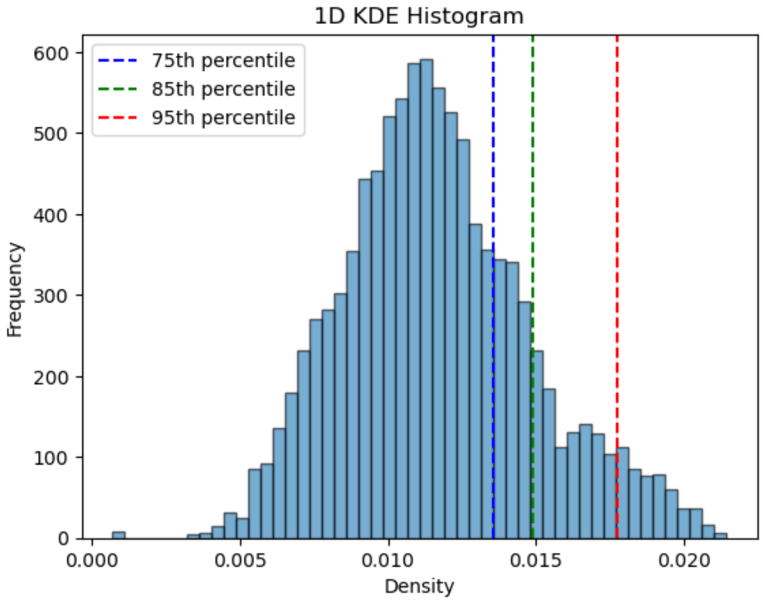

Figure 3: The histogram of 1D density

Given a level λ, points with density exceeding λ are selected as cluster cores. A radius-neighborhood graph is then constructed among these core points, and each connected component defines a cluster. The remaining low-density points are assigned to existing clusters via a 1-nearest-neighbor query.

To determine the optimal λ_{opt}, we varied λ between the 85^{th} and 95^{th} percentiles and evaluated clustering quality using the silhouette score.

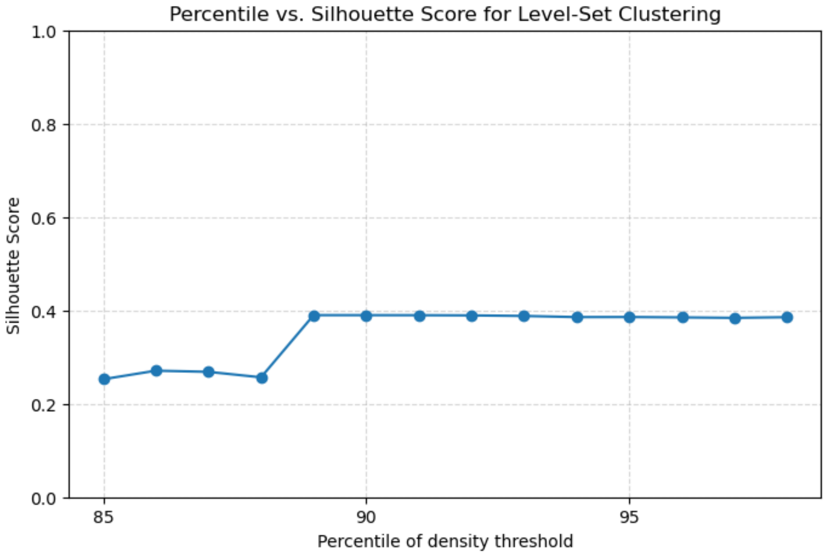

Figure 4: Percentile vs. Silhouette Score

We found $\lambda_{\text{opt}} = 0.89$, which yielded a silhouette score of 0.39, which is acceptable for high-dimensional clustering. And since the data is three-dimension, we can visualize it without projection into low dimension space.

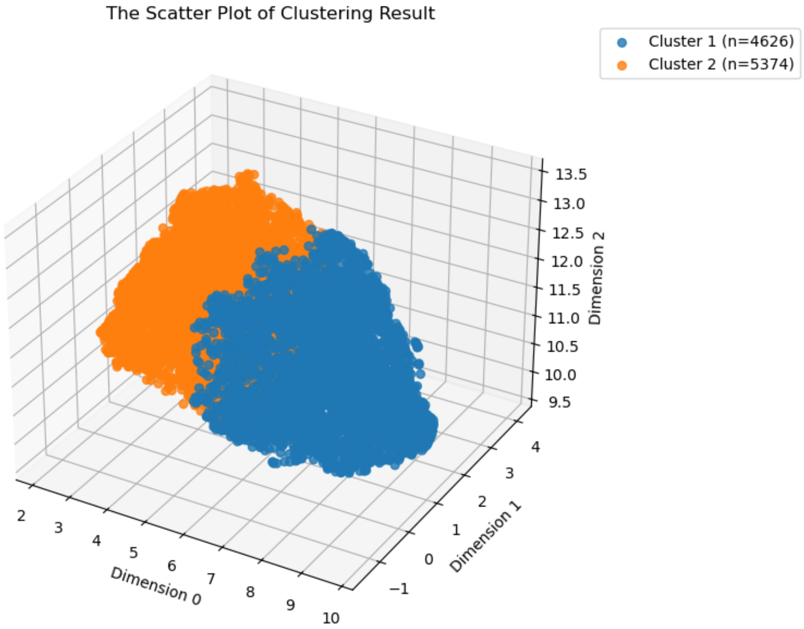

Figure 5: The Result of Clustering with level = 0.89

At this level, the data split into two clusters: cluster 1 contains 4,626 points and cluster 2 contains 5,374 points. The boundary between these two clusters is diffuse, indicating that the data are not highly separable.

We save the clustering results to alg1_out.txt, which has the following format:

```
1
2
2
2
1
1
1
2
1
1
1
...
```

(10,000 lines)

4 Method 2 Spectral Clustering

4.1 Dimensionality Reduction

Similar to Method 1, we follow the same procedure for spectral clustering on dimensionality reduction. However, instead of reducing the data to three dimensions, we project it into 20 dimensions to better preserve its underlying structure, since spectral clustering does not require extremely low dimensional setting for optimal performance.

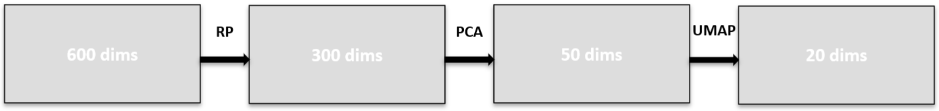

Figure 6: Dimensionality Reduction Pipeline for Method 2

4.2 The Selection of the Number of Clusters

For the selection of the number of clusters k, the idea is to observe the eigenvalues of Laplacian matrix L, we first sort the eigenvalues of L in ascending order:

$$0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$$

If there truly exist k clusters, then L will have exactly k eigenvalues that are nearly zero, and you will see a significant gap between λ_k and λ_{k+1}. Therefore, we can look for the largest gap between eigenvalues, picking k to be the index before the index i which maximizes $\lambda i + 1 - \lambda i$.

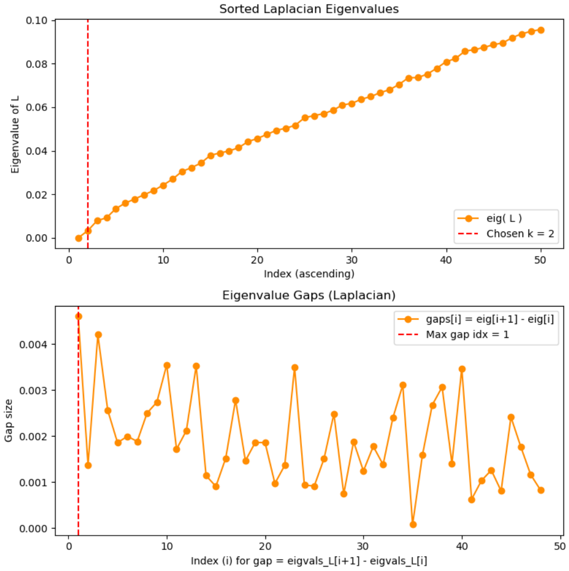

Figure 7: The Plot of Eigenvalues Gaps

We determined that the optimal number of clusters is $k = 2$ and performed spectral clustering on the data. To visualize the result, we applied PCA to reduce the 20-dimensional data to three dimensions and plotted the clusters.

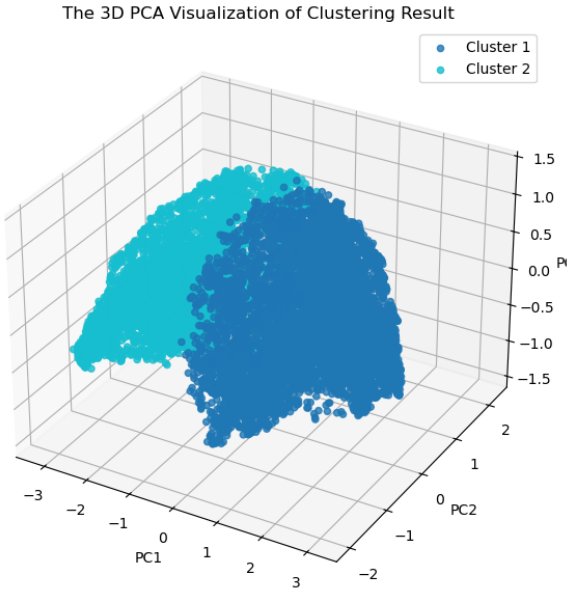

Figure 8: The 3D Visualization of the Clustering Result

Based on the visualization, the data split into two clusters: cluster 1 contains 6,099 points and cluster 2 contains 3,901 points. The two clusters separate clearly along PC1. A silhouette score of approximately 0.37, indicates that while each cluster is fairly compact, the separation between them is substantial but not complete.

We save the clustering results to alg2_out.txt, which has the following format:

```
1
2
```

```
2
2
2
1
1
1
1
2
1
...
```

(10,000 lines)

5 Conclusion

In this project, we successfully applied two nonparametric clustering methods, level-set clustering with kernel density estimation and spectral clustering on a high-dimensional dataset with $n = 10000$ and $d = 600$. To address the curse of dimensionality, we utilized a carefully designed dimensionality reduction methods involving random projection, PCA, and UMAP for both level-set clustering and spectral clustering. Both methods identified two clusters within the data, with level-set clustering yielding a silhouette score of approximately 0.39 while the spectral clustering achieving about 0.37, indicating relatively good cluster performance given the high-dimensional setting. Visualizations further confirmed clear but not entirely distinct separations, highlighting the complexity of the underlying data structure.

6 Acknowledgements

This report received support and guidance of the STAT 548 faculty, particularly Professor Marina Meila, Ph.D. at the University of Washington. As well as helpful feedback from teaching assistants and fellow classmates in the course.

References

[1] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson–lindenstrauss lemma. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[2] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. In *Conference in Modern Analysis and Probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. 1984.

[3] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 2nd edition, 2015.

[4] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, UK, 1986.

[5] J. A. Hartigan. *Clustering Algorithms*. Wiley, New York, NY, 1975.

[6] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[7] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.