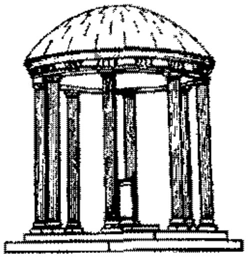
A Very Fast Algorithm for Finding Eigenvalues and Eigenvectors

TR96-043
1996

John H. Halton

Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175



UNC is an Equal Opportunity/Affirmative Action Institution.

A VERY FAST ALGORITHM FOR FINDING EIGENVALUES AND EIGENVECTORS

John H. Halton
Professor of Computer Science
The University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175
halton@cs.unc.edu
919/962-1752

ABSTRACT

This paper describes, illustrates, and analyzes a new iterative technique for computing the eigenvalues and eigenvectors of a matrix, converging superlinearly with exponent $2 + \sqrt{3}$ [in the sense that "quadratic" convergence has exponent 2].

1. INTRODUCTION

Let $\mathbf{H}$ be an arbitrary $(m \times m)$, possibly complex, matrix. Then the eigenvalue equation is

$$\mathbf{Hx} = \lambda\mathbf{x}, \tag{1.1}$$

with

$$\mathbf{x} \neq \mathbf{0}. \tag{1.2}$$

If $\mathbf{x}$ is a solution (called an *eigenvector*), so is any multiple $\kappa\mathbf{x}$, so long as κ is not zero. Thus, $\mathbf{x}$ really identifies an *eigendirection*. The eigenvector $\mathbf{x}$ and the *eigenvalue* λ are then said to belong to each other. Each eigenvector belongs to only one eigenvalue, and the eigenvectors belonging to any given eigenvalue λ form a linear space E_λ without the *null vector* $\mathbf{0}$[1] (E_λ is called the *eigensubspace* belonging to λ), and all such eigensubspaces are *disjoint*.

The eigenvalues of a given matrix $\mathbf{H}$ are solutions of the polynomial equation of degree m,

$$\det(\mathbf{H} - \lambda\mathbf{I}) = 0, \tag{1.3}$$

which is called the *characteristic equation*. It has just m solutions $\lambda_1, \lambda_2, \ldots, \lambda_m$ (if we take multiplicity into account), which can always be ordered so that

[1] This is because of (1.2).

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_m|. \tag{1.4}$$

In the general case, the set of all eigenvectors can (if necessary) be augmented by further vectors, to form a base of the m-dimensional vector space. In this paper, we shall assume[2] that

$$|\lambda_1| > |\lambda_2| > |\lambda_3| > \ldots > |\lambda_m| \geq 0, \tag{1.5}$$

and then corresponding eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ are *linearly independent*, so that any vector $\mathbf{v}$ can be written uniquely as

$$\mathbf{v} = \xi_1\mathbf{x}_1 + \xi_2\mathbf{x}_2 + \xi_3\mathbf{x}_3 + \ldots + \xi_m\mathbf{x}_m. \tag{1.6}$$

It then follows from (1.1) that

$$\mathbf{Hv} = \xi_1\lambda_1\mathbf{x}_1 + \xi_2\lambda_2\mathbf{x}_2 + \ldots + \xi_m\lambda_m\mathbf{x}_m. \tag{1.7}$$

Statistically, it is *almost surely* the case that no $\xi_h = 0$, and we shall also assume this here.

Since an eigenvector really identifies an *eigendirection*,[3] we can always assume, without loss of generality, that the base vectors are all *normalized*. In this paper, we shall use the *Euclidean* (or L^2) *norm*,[4]

$$\|\mathbf{v}\| = (\mathbf{v}^*\mathbf{v})^{1/2} = \left(\sum_{h=1}^{m} |v_h|^2\right)^{1/2}, \tag{1.8}$$

and set

$$\|\mathbf{x}_1\| = \|\mathbf{x}_2\| = \|\mathbf{x}_3\| = \ldots = \|\mathbf{x}_m\| = 1. \tag{1.9}$$

We are still free to multiply each such base vector $\mathbf{x}_h$ by an arbitrary *phase-factor* $\mathrm{e}^{\mathrm{i}\psi_h}$, preserving (1.9). We choose a k_h, such that $x_{hk_h} \neq 0$,

[2] The complications arising if eigenvalues are not distinct are well-understood and their treatment, while not easy, is described in the literature; appropriate techniques are available. In a statistical sense, it is highly unlikely that any two eigenvalues should be equal, or should even have the same magnitude, unless the particular class of problems considered constrains such equality.

[3] See the explanation just after (1.2).

[4] For a possibly complex number $z = x + \mathrm{i}y$, where x and y are real numbers, z^* denotes the *complex conjugate* number, $z^* = x - \mathrm{i}y$. For a possibly complex matrix, $\mathbf{H} = \mathbf{L} + \mathrm{i}\mathbf{M}$, again with matrices $\mathbf{L}$ and $\mathbf{M}$ real, $\mathbf{H}^*$ denotes the *Hermitian transpose*, $\mathbf{H}^* = \mathbf{L}^{\mathrm{T}} - \mathrm{i}\mathbf{M}^{\mathrm{T}}$, where $(\mathbf{L}^{\mathrm{T}})_{hk} = L_{kh}$ and $(\mathbf{H}^*)_{hk} = H_{kh}{}^*$.

and then choose $e^{i\psi_h}$, so that

$$x_{hk_h} \text{ is real,} \quad x_{hk_h} > 0. \tag{1.10}$$

Of course, we do not yet *know* these eigenvectors (the whole purpose of this paper is to describe a method of finding them), but what (1.9) and (1.10) mean is that, when we determine any $\mathbf{x}_h$, it will take this canonical form.

2. THE POWER METHOD

Write $\mathbf{O}(f(n))$ for an asymptotic characterization of a *vector* whose norm increases with $n \to \infty$ no faster than a given (usually, an easy) function $f(n)$.[5] Then we see, by (1.6) and (1.7), that we can write

$$\mathbf{u}^{(0)} = \mathbf{v} \tag{2.1}$$

and, as $r \to \infty$,

$$\mathbf{u}^{(r)} = \mathbf{H}^r\mathbf{v} = \xi_1\lambda_1{}^r\mathbf{x}_1 + \xi_2\lambda_2{}^r\mathbf{x}_2 + \mathbf{O}(\lambda_3{}^r). \tag{2.2}$$

If we put $\kappa = \xi_2/\xi_1, \quad \alpha = \lambda_2/\lambda_1, \quad \text{and} \quad \beta = \lambda_3/\lambda_1,$ (2.3)

so that, by (1.5), $|\beta| < |\alpha| < 1,$ (2.4)

we see, by (2.2), that, as $r \to \infty$,

$$\mathbf{u}^{(r)} = \xi_1\lambda_1{}^r\{\mathbf{x}_1 + \kappa\alpha^r\mathbf{x}_2 + \mathbf{O}(\beta^r)\}, \tag{2.5}$$

or $\mathbf{u}^{(r)} = \xi_1\lambda_1{}^r\{\mathbf{x}_1 + \mathbf{O}(\alpha^r)\}.$ (2.6)

In particular, $\mathbf{u}^{(r)} \sim \xi_1\lambda_1{}^r\mathbf{x}_1;$ (2.7)

i.e., for all h, $u_h^{(r)} \sim \xi_1\lambda_1{}^r x_{1h}.$ (2.8)

[5] This is an extension of the well-known "big-Oh" notation from asymptotics: the boldface **O** indicates a vector.

Now let us normalize the vectors $\mathbf{u}^{(r)}$ defined above, by the relations

$$\mathbf{z}^{(0)} = \frac{\mathrm{e}^{\mathrm{i}\varphi^{(0)}}}{\|\mathbf{u}^{(0)}\|}\mathbf{u}^{(0)} = \frac{\mathrm{e}^{\mathrm{i}\phi^{(0)}}}{\|\mathbf{v}\|}\mathbf{v} \tag{2.9}$$

and

$$\mathbf{z}^{(r)} = \frac{\mathrm{e}^{\mathrm{i}\varphi^{(r)}}}{\|\mathbf{u}^{(r)}\|}\mathbf{u}^{(r)}; \tag{2.10}$$

so that

$$\|\mathbf{z}^{(r)}\| = 1. \tag{2.11}$$

The phase-factors $\mathrm{e}^{\mathrm{i}\phi^{(r)}}$ may be determined by requiring that, in every case,[6]

$$z_{k_h}^{(r)} \text{ is real,} \quad z_{k_h}^{(r)} > 0. \tag{2.12}$$

Then, by (1.5), (1.8), (2.6), (2.10), as $r \to \infty$,[7]

$$\begin{aligned}
\mathbf{z}^{(r)} &= \frac{\mathrm{e}i\varphi^{(r)}}{\|\mathbf{u}(r)\|}\mathbf{u}(r) \\
&= \frac{\mathrm{e}^{\mathrm{i}\varphi^{(r)}}}{\left(|\xi_1|^2|\lambda_1|^{2r}\{1+O(\alpha^r)\}\right)^{1/2}}\,\xi_1\lambda_1{}^r\{\mathbf{x}_1 + \mathbf{O}(\alpha^r)\} \\
&= \frac{\mathrm{e}^{\mathrm{i}\varphi^{(r)}}}{\{1+O(\alpha^r)\}^{1/2}}\{\mathbf{x}_1 + \mathbf{O}(\alpha^r)\} \\
&= [1+O(\alpha^r)]\,\{\mathbf{x}_1 + \mathbf{O}(\alpha^r)\} \qquad \text{[by (1.10) and (2.12)]} \\
&= [1+O(\alpha^r)]\,\mathbf{x}_1 + \mathbf{O}(\alpha^r),
\end{aligned} \tag{2.13}$$

[6] Compare (1.9)–(1.10). Note that $|\mathrm{e}^{\mathrm{i}\phi^{(r)}}| = 1.$

[7] Here, we use well-known properties of asymptotic expressions, such as

(i) $O(\alpha^{r-1}) = O(\alpha^r),$

(ii) $\{1+O(\alpha^r)\}/\{1+O(\alpha^r)\} = 1+O(\alpha^r),$

and (iii) $\{1+O(\alpha^r)\}^{1/2} = 1+O(\alpha^r).$

For the result (iii), observe that, by the Binomial Theorem, since $|\alpha| < 1$, as $r \to \infty$,

$$[1+O(\alpha^r)]^{1/2} = 1 + \tfrac{1}{2}O(\alpha^r) - \tfrac{1}{8}O(\alpha^{2r}) + \ldots = 1 + O(\alpha^r).$$

whence
$$\mathbf{z}^{(r)} \to \mathbf{x}_1 \quad \text{as} \quad r \to \infty. \tag{2.14}$$

Similarly, by (1.7) and (2.13),
$$\mathbf{H}\mathbf{z}^{(r-1)} = \lambda_1 \, [1 + O(\alpha^r)] \, \mathbf{x}_1 + \mathbf{O}(\alpha^r); \tag{2.15}$$

so that, if we write, because of (2.10),
$$\mathbf{y}^{(r)} = \mathbf{H}\mathbf{z}^{(r-1)} = \tau_r \, \mathbf{z}^{(r)}, \tag{2.16}$$

then
$$\tau_r \, [1 + O(\alpha^r)] \, \mathbf{x}_1 + \mathbf{O}(\alpha^r) = \lambda_1 \, [1 + O(\alpha^r)] \, \mathbf{x}_1 + \mathbf{O}(\alpha^r),$$

whence
$$\tau_r = \lambda_1 \, [1 + O(\alpha^r)] \to \lambda_1 \quad \text{as} \quad r \to \infty. \tag{2.17}$$

The process embodied in (2.1)–(2.17) and leading to the results (2.13), (2.14), and (2.17) constitutes the "power method." The convergence is *geometric*, with constant factor $|\alpha|$, since the error is given by (2.17) as $O(\alpha^r)$.

3. RAYLEIGH QUOTIENTS

If the matrix **H** has additional properties, even faster methods are available. For example, if **H** is Hermitian, i.e.,[4]
$$\mathbf{H} = \mathbf{H}^* \quad \text{or} \quad (\forall h, k) \;\; H_{hk} = H_{kh}{}^*, \tag{3.1}$$

then, for all h,
$$\lambda_h{}^* = \lambda_h; \tag{3.2}$$

that is, all the eigenvalues are *real*; and if $h \neq k$ and $\lambda_h \neq \lambda_k$,
$$\mathbf{x}_h{}^*\mathbf{x}_k = 0, \tag{3.3}$$

i.e., the eigenvectors belonging to distinct eigenvalues are *orthogonal*. Hence, if we define the *Rayleigh quotient* as
$$\mathscr{R}_r = \frac{\mathbf{z}^{(r-1)*}(\mathbf{H}\mathbf{z}^{(r-1)})}{\mathbf{z}^{(r-1)*}\mathbf{z}^{(r-1)}} = \frac{\mathbf{u}^{(r-1)*}(\mathbf{H}\mathbf{u}^{(r-1)})}{\mathbf{u}^{(r-1)*}\mathbf{u}^{(r-1)}}, \tag{3.4}$$

we get, by (1.9), (2.5), and (3.3), that

$$\begin{aligned}
\mathfrak{R}_r &= \frac{\xi_1{}^*(\lambda_1{}^*)^{r-1}\,\{\mathbf{x}_1{}^* + \kappa^*(\alpha^*)^{r-1}\mathbf{x}_2{}^* + \mathbf{O}(\beta^{r-1})\}\,\xi_1\lambda_1{}^r\,\{\mathbf{x}_1 + \kappa\alpha^r\mathbf{x}_2 + \mathbf{O}(\beta^r)\}}{\xi_1{}^*(\lambda_1{}^*)^{r-1}\,\{\mathbf{x}_1{}^* + \kappa^*(\alpha^*)^{r-1}\mathbf{x}_2{}^* + \mathbf{O}(\beta^{r-1})\}\,\xi_1\lambda_1{}^{r-1}\,\{\mathbf{x}_1 + \kappa\alpha^{r-1}\mathbf{x}_2 + \mathbf{O}(\beta^{r-1})\}} \\
&= \lambda_1 \frac{\mathbf{x}_1{}^*\mathbf{x}_1 + \alpha|\kappa|^2|\alpha|^{2r-2}\mathbf{x}_2{}^*\mathbf{x}_2 + O(\beta^{2r-1})}{\mathbf{x}_1{}^*\mathbf{x}_1 + |\kappa|^2|\alpha|^{2r-2}\mathbf{x}_2{}^*\mathbf{x}_2 + O(\beta^{2r-2})} \\
&= \lambda_1 \frac{1 + O(\alpha^{2r-1})}{1 + O(\alpha^{2r-2})} = \lambda_1\,[1 + O(\alpha^{2r})]. \qquad (3.5)
\end{aligned}$$

Thus,
$$\mathfrak{R}_r \to \lambda_1 \quad \text{as} \quad r \to \infty; \qquad (3.6)$$

and, furthermore, the convergence is again *geometric*, with constant factor $|\alpha|^2$, *twice as fast* as the regular power method exhibits in (2.17). The cost is the additional condition, that **H** be Hermitian, and the more laborious computation of $\mathfrak{R}_r$.

4. THE INVERSE POWER METHOD

We return to the eigenvalue problem—equations (1.1) and (1.2), with the assumptions embodied in (1.5), (1.9), (1.10), (2.11), and (2.12).

Now let μ be any real number and consider the matrix

$$\mathbf{M} = \mathbf{H} - \mu\mathbf{I}, \qquad (4.1)$$

Clearly, for any eigenvector $\mathbf{x}_h$ of **H**,

$$\mathbf{M}\mathbf{x}_h = (\mathbf{H} - \mu\mathbf{I})\mathbf{x}_h = (\lambda_h - \mu)\mathbf{x}_h. \qquad (4.2)$$

If we assume that **M** is invertible,[8] and write

$$\mathbf{N} = \mathbf{M}^{-1} = (\mathbf{H} - \mu\mathbf{I})^{-1}, \qquad (4.3)$$

so that
$$\mathbf{MN} = \mathbf{NM} = \mathbf{I}, \qquad (4.4)$$

then[9]
$$\mathbf{N}\mathbf{x}_h = (\mathbf{H} - \mu\mathbf{I})^{-1}\mathbf{x}_h = (\lambda_h - \mu)^{-1}\mathbf{x}_h. \qquad (4.5)$$

[8] This assumption is equivalent to assuming that μ is itself *not* an eigenvalue of **H**.

[9] To obtain (4.5), premultiply (4.2) by **N** and divide by the (presumed non-zero) scalar $\lambda_h - \mu$.

That is to say, the vectors $\mathbf{x}_h$ are also eigenvectors of the matrices **M** and **N**. Therefore, by (1.6) and (1.7),

$$\mathbf{Mv} = \xi_1(\lambda_1 - \mu)\mathbf{x}_1 + \xi_2(\lambda_2 - \mu)\mathbf{x}_2 + \ldots + \xi_m(\lambda_m - \mu)\mathbf{x}_m \tag{4.6}$$

and $$\mathbf{Nv} = \xi_1(\lambda_1 - \mu)^{-1}\mathbf{x}_1 + \xi_2(\lambda_2 - \mu)^{-1}\mathbf{x}_2 + \ldots + \xi_m(\lambda_m - \mu)^{-1}\mathbf{x}_m. \tag{4.7}$$

Now consider carrying out the power method with the matrix **N** instead of **H**. Suppose that μ is nearest to the eigenvalue λ_s, and nearer to it than to any other eigenvalue, and write

$$\lambda_h = \mu + \nu_h. \tag{4.8}$$

Then $$\min_{1 \le h \le m} |\lambda_h - \mu| = |\lambda_s - \mu|, \tag{4.9}$$

or $$\min_{1 \le h \le m} |\nu_h| = |\nu_s|. \tag{4.10}$$

If we define the parameter κ by

$$0 < |\nu_s| < \kappa = \min_{h \ne s} |\lambda_h - \lambda_s|, \tag{4.11}$$

then $$\min_{h \ne s} |\nu_h| = \min_{h \ne s} |\lambda_h - \mu| = \min_{h \ne s} |(\lambda_h - \lambda_s) + (\lambda_s - \mu)|,$$

so that $$\min_{h \ne s} |\nu_h| \ge \kappa + |\nu_s|, \tag{4.12}$$

whence $$\max_{h \ne s} (|\nu_h|^{-1}) \le (\kappa + |\nu_s|)^{-1} < \kappa^{-1}. \tag{4.13}$$

Let us also define γ, by (4.11) and (4.13), as satisfying

$$\max_{h \ne s} \left(\frac{|\nu_s|}{|\nu_h|} \right) \le \frac{|\nu_s|}{\kappa + |\nu_s|} = \gamma < \frac{|\nu_s|}{\kappa} < 1. \tag{4.14}$$

Arguing just as in §2, we see that, if[10]

$$\mathbf{u}^{(0)} = \mathbf{v} \quad \text{and} \quad \mathbf{u}^{(r)} = \mathbf{N}^r\mathbf{v}, \tag{4.15}$$

then, as $r \to \infty$,[11]

$$\mathbf{u}^{(r)} = \xi_s \nu_s^{-r}\, \mathbf{x}_s + \mathbf{O}\left(\left(\frac{|\nu_s|}{\kappa + |\nu_s|}\right)^r\right) = \xi_s \nu_s^{-r}\, [\mathbf{x}_s + \mathbf{O}(\gamma^r)]. \tag{4.16}$$

[10] Compare (2.1) and (2.2).

[11] See (2.6), (4.7), and (4.14).

As before, we can normalize the $\mathbf{u}^{(r)}$ as $\mathbf{z}^{(r)}$, and get[12]

$$\|\mathbf{u}^{(r)}\| = (\mathbf{u}^{(r)*}\mathbf{u}^{(r)})^{1/2} = |\xi_s \nu_s^{-r}|\, [1 + O(\gamma^r)], \tag{4.17}$$

and so
$$\mathbf{z}^{(r)} = \mathrm{e}^{\mathrm{i}\phi^{(r)}} (\mathbf{u}^{(r)*}\mathbf{u}^{(r)})^{-1/2}\mathbf{u}^{(r)} = [1 + O(\gamma^r)]\, [\mathbf{x}_s + \mathbf{O}(\gamma^r)]$$
$$= [1 + O(\gamma^r)]\, \mathbf{x}_s + \mathbf{O}(\gamma^r), \tag{4.18}$$

whence
$$\mathbf{z}^{(r)} \to \mathbf{x}_s \quad \text{as} \quad r \to \infty. \tag{4.19}$$

Similarly,
$$\mathbf{N}\mathbf{z}^{(r-1)} = \nu_s^{-1}\, [1 + O(\gamma^r)]\, \mathbf{x}_s + \mathbf{O}(\gamma^r); \tag{4.20}$$

so that, if[13]
$$\mathbf{y}^{(r)} = \mathbf{N}\mathbf{z}^{(r-1)} = \rho_r^{-1}\, \mathbf{z}^{(r)}, \tag{4.21}$$

then
$$\rho_r = \nu_s\, [1 + O(\gamma^r)] \to \nu_s \quad \text{as} \quad r \to \infty. \tag{4.22}$$

This shows that this *inverse power method* converges to ν_s; and so, by (4.8), as $r \to \infty$,

$$\mu + \rho_r \to \mu + \nu_s = \lambda_s, \tag{4.23}$$

the nearest eigenvalue to μ, and the convergence is again *geometric*, with constant factor γ.

If we further assume, as in §3, that the matrix $\mathbf{H}$ is Hermitian,[14] with its eigenvalues λ_h real and its eigenvectors $\mathbf{x}_h$ forming a base of m-space and *orthonormal,*[15]

$$(\forall h, k) \;\; \mathbf{x}_h{}^*\mathbf{x}_k = \delta_{hk}, \tag{4.24}$$

then, by (4.16)—and because of (4.24)—(4.17), (4.18), and (4.20) respectively become

$$\|\mathbf{u}^{(r)}\| = (\mathbf{u}^{(r)*}\mathbf{u}^{(r)})^{1/2} = |\xi_s \nu_s^{-r}|\, [1 + O(\gamma^{2r})], \tag{4.25}$$

$$\mathbf{z}^{(r)} = \mathrm{e}^{\mathrm{i}\phi^{(r)}} (\mathbf{u}^{(r)*}\mathbf{u}^{(r)})^{-1/2}\, \mathbf{u}^{(r)} = [1 + O(\gamma^{2r})]\, \mathbf{x}_s + \mathbf{O}(\gamma^r), \tag{4.26}$$

[12] See (2.9)–(2.17).

[13] Use (4.15). We can compute $\rho_r = z^{(r)}_{k_h} \big/ y^{(r)}_{k_h}$.

[14] See (1.9) and (3.1)–(3.3).

[15] Here, δ_{ij} is "Kronecker's delta"—$\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ if $i \neq j$.

and
$$\mathbf{y}^{(r)} = \mathbf{N}\mathbf{z}^{(r-1)} = \nu_s^{-1}\,[1 + O(\gamma^{2r})]\,\mathbf{x}_s + \mathbf{O}(\gamma^r). \tag{4.27}$$

Furthermore, if we define[16]

$$\mathcal{T}_r = \frac{1}{\mathbf{z}^{(r-1)*}\mathbf{y}^{(r)}} = \frac{\mathbf{u}^{(r-1)*}\mathbf{u}^{(r-1)}}{\mathbf{u}^{(r-1)*}\mathbf{u}^{(r)}}\,, \tag{4.28}$$

then[17]
$$\mathcal{T}_r = \frac{1}{\{\,[1 + O(\gamma^{2r})]\,\mathbf{x}_s^* + \mathbf{O}(\gamma^r)\}\{\nu_s^{-1}\,[1 + O(\gamma^{2r})]\,\mathbf{x}_s + \mathbf{O}(\gamma^r)\}}$$

$$= \nu_s\,\frac{1}{[1 + O(\gamma^{2r})] + O(\gamma^{2r})} = \nu_s\,[1 + O(\gamma^{2r})]. \tag{4.29}$$

Thus,
$$\mathcal{T}_r \to \nu_s \quad \text{as} \quad r \to \infty, \tag{4.30}$$

or, by (4.8),
$$\mathcal{T}_r + \mu \to \lambda_s \quad \text{as} \quad r \to \infty. \tag{4.31}$$

This shows that this *inverse power method* converges to λ_s, the nearest eigenvalue to μ, as $r \to \infty$, and the convergence is again *geometric*, with constant factor γ^2.

The formal computational procedure is thus as follows.

```
0.          Begin with an arbitrary vector $\mathbf{v} = \mathbf{u}^{(0)}$ and an arbitrary real number
            $\mu = \mu^{(0)}$. Define $\mathbf{z}^{(0)} = (\mathbf{v}^*\mathbf{v})^{-1/2}\mathbf{v}$, so that $\|\mathbf{z}^{(0)}\| = 1$. Take $r = 1$.
1.   i.     Define the matrix                  $\mathbf{M} = \mathbf{H} - \mu\mathbf{I}$.                        (4.32)
     ii.    Solve the equation                 $\mathbf{M}\mathbf{y}^{(r)} = \mathbf{z}^{(r-1)}$                          (4.33)
            and put                            $\mathbf{z}^{(r)} = (\mathbf{y}^{(r)*}\mathbf{y}^{(r)})^{-1/2}\mathbf{y}^{(r)}$.             (4.34)
2.          Compute either
               (a) [in general][18]            $\rho_r = z^{(r)}_{k_h}/y^{(r)}_{k_h}$                          (4.35)
            or (b) [if H is Hermitian]         $\mathcal{T}_r = (\mathbf{z}^{(r-1)*}\mathbf{y}^{(r)})^{-1}$.                (4.36)
3.          Increment $r$. Repeat [1] and [2] until  $\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\| \le \varepsilon$.        (4.37)
```

[16] Compare (3.4).

[17] See (4.26) and (4.27).

[18] See (2.12) and (4.21).

5. ACCELERATING THE INVERSE POWER METHOD

We know that the inverse power method (described above) converges geometrically to λ_s. It is also clear that the rate of convergence of the method is governed by the constant γ defined in (4.8)–(4.14). As ν_s decreases, so does γ, and the convergence becomes faster. Thus, if we change μ so as to decrease ν_s in the course of the computation, we can only accelerate the convergence of the method to λ_s.

In all cases, by (4.21)–(4.23) and (4.35), we can use $\rho_r = z^{(r)}_{k_h} / y^{(r)}_{k_h}$, and, if $\mathbf{H}$ is Hermitian, by (4.28)–(4.31) and (4.37), we can use $\mathfrak{T}_r = (\mathbf{z}^{(r-1)*}(\mathbf{N}\mathbf{z}^{(r-1)}))^{-1}$ to approximate ν_s. Suppose, therefore, that we replace the inverse power method outlined in §4 above by a two-part sequential method as follows.[19]

```
0.        Begin with an arbitrary vector $\mathbf{v} = \mathbf{u}^{(0)}$ and an arbitrary real number
          $\mu = \mu^{(0)}$. Define $\mathbf{z}^{(0)} = (\mathbf{v}^*\mathbf{v})^{-1/2}\mathbf{v}$, so that $\|\mathbf{z}^{(0)}\| = 1$. Take $r = 1$.
1.   i.   Define the matrix                    $\mathbf{M} = \mathbf{H} - \mu\mathbf{I}$.                          (5.1)
     ii.  Solve the equation                   $\mathbf{M}\mathbf{y}^{(r)} = \mathbf{z}^{(r-1)}$                     (5.2)
          and put                              $\mathbf{z}^{(r)} = (\mathbf{y}^{(r)*}\mathbf{y}^{(r)})^{-1/2}\mathbf{y}^{(r)}$.    (5.3)
2.   i.   Compute either
              (a)  [in general][20]            $\rho_r = z^{(r)}_{k_h} / y^{(r)}_{k_h}$                     (5.4)
          or  (b)  [if H is Hermitian]         $\mathfrak{T}_r = (\mathbf{z}^{(r-1)*}\mathbf{y}^{(r)})^{-1}$.          (5.5)
     ii.  Put either
              (a)  [if (5.4) is used]          $\mu^{(r)} = \mu^{(r-1)} + \rho_r$.                  (5.6)
          or  (b)  [if (5.5) is used]          $\mu^{(r)} = \mu^{(r-1)} + \mathfrak{T}_r$.                  (5.7)
3.        Increment $r$. Repeat [1] and [2] until  $\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\| \leq \varepsilon$.     (5.8)
```

[19] Compare the closely similar procedure in (4.32)–(4.37). The initialization is [0]; the two major steps are [1] and [2]; the conditional looping command is [3]. The crucial change is the iterative improvement of $\mu^{(r)}$ through (5.6) or (5.7)), in [2][ii].

[20] See (2.12) and (4.21).

The change from the inverse Rayleigh quotient method is solely in the lines [2][ii][a] or [2][ii][b]; in the original method, all $\mu^{(r)} = \mu^{(0)}$. Note, too, that the application of the *Monte Carlo* method is to perform [1][ii] by PMC.[21]

For the original method, we get, by (4.18), that

$$\mathbf{z}^{(r)} = [1 + O(\gamma^r)]\,\mathbf{x}_s + \mathbf{O}(\gamma^r), \tag{5.9}$$

whence

$$\mathbf{z}^{(r)} - \mathbf{x}_s = \mathbf{O}(\gamma^r); \tag{5.10}$$

i.e.,

$$\left\|\mathbf{z}^{(r)} - \mathbf{x}_s\right\| = O(\gamma^r). \tag{5.11}$$

Similarly, by (4.22),

$$\left|\rho_r - \nu_s\right| = O(\gamma^r) \quad \text{as} \quad r \to \infty, \tag{5.12}$$

and, by (4.29)–(4.31),

$$\left|\mathfrak{T}_r - \nu_s\right| = \left|(\mathfrak{T}_r + \mu) - \lambda_s\right| = O(\gamma^{2r}) \quad \text{as} \quad r \to \infty. \tag{5.13}$$

In the new algorithm, it is clear that ν_s must be replaced by a changing parameter, given, in the general case, by (5.6), i.e.,

$$\nu_s^{(r)} = \lambda_s - \mu^{(r)} = \lambda_s - \mu^{(r-1)} - \rho_r = \nu_s^{(r-1)} - \rho_r. \tag{5.14}$$

and, in the case of Hermitian $\mathbf{H}$, by (5.7), i.e.,

$$\nu_s^{(r)} = \lambda_s - \mu^{(r)} = \lambda_s - \mu^{(r-1)} - \mathfrak{T}_r = \nu_s^{(r-1)} - \mathfrak{T}_r. \tag{5.15}$$

Going over the previous line of argument, modified for the present situation, we observe that, if we now write

$$P_r = \prod_{t=0}^{r-1} \gamma_r = \prod_{t=0}^{r-1} \frac{\left|\nu_s^{(t)}\right|}{\kappa} \geq \prod_{t=0}^{r-1} \frac{\left|\nu_s^{(t)}\right|}{\kappa + \left|\nu_s^{(t)}\right|}, \tag{5.16}$$

then[22]

$$\mathbf{z}^{(r)} = [1 + O(P_r)]\,\mathbf{x}_s + \mathbf{O}(P_r), \tag{5.17}$$

[21] It is also possible to replace the scalar products in (5.3) and (5.5) by MC estimates.

[22] Compare (5.9).

whence[23]
$$\mathbf{z}^{(r)} - \mathbf{x}_s = \mathbf{O}(P_r), \tag{5.18}$$

i.e.,[24]
$$\left\| \mathbf{z}^{(r)} - \mathbf{x}_s \right\| = O(P_r). \tag{5.19}$$

The counterpart of (4.22) and (5.12) is now

$$\rho_r = v_s^{(r-1)} \left[1 + O(P_r)\right] \to v_s \quad \text{as} \quad r \to \infty, \tag{5.20}$$

whence

$$\left| \rho_r - v_s^{(r-1)} \right| = \left| (\rho_r + \mu^{(r-1)}) - \lambda_s \right| = O(P_r) \quad \text{as} \quad r \to \infty; \tag{5.21}$$

and that of (4.29) is

$$\mathcal{T}_r = v_s^{(r-1)} \left[1 + O({P_r}^2)\right] \to v_s^{(r-1)} \quad \text{as} \quad r \to \infty, \tag{5.22}$$

whence[25]

$$\left| \mathcal{T}_r - v_s^{(r-1)} \right| = \left| (\mathcal{T}_r + \mu^{(r-1)}) - \lambda_s \right| = O({P_r}^2) \quad \text{as} \quad r \to \infty. \tag{5.23}$$

Note that, because each $\mu^{(r)}$ is an improvement on its predecessor, and therefore each $v_s^{(r)}$ is less than its predecessor, P_r will necessarily be less than the corresponding $\mathscr{Y}$ in (5.9)–(5.13).

By (5.14) and (5.20), we see that, for the general case,

$$v_s^{(r)} = v_s^{(r-1)} - v_s^{(r-1)} \left[1 + O(P_r)\right] = v_s^{(r-1)} O(P_r); \tag{5.24}$$

similarly, by (5.15) and (5.22), for Hermitian **H**,

$$v_s^{(r)} = v_s^{(r-1)} - v_s^{(r-1)} \left[1 + O({P_r}^2)\right] = v_s^{(r-1)} O({P_r}^2). \tag{5.25}$$

It is important to observe that (5.25) differs from (5.24) *only* in that the P_r factor is squared in (5.25).

[23] Compare (5.10).

[24] Compare (5.11).

[25] Compare (5.13).

In the general case, we can attempt to solve the relations (5.16) and (5.24) by putting, for some

$$Q \geq 1, \quad C > 1, \quad \text{and} \quad 0 < q < 1, \tag{5.26}$$

that

$$\gamma_r = \frac{\left| v_s^{(t)} \right|}{\kappa} \sim Qq^{C^r}. \tag{5.27}$$

Then, by (5.16),

$$P_r = \prod_{t=0}^{r-1} \gamma_r \sim \prod_{t=0}^{r-1} Qq^{C^t}$$

$$= Q^r q^{1+C+C^2+\ldots+C^{r-1}} = Q^r q^{(C^r-1)/(C-1)}, \tag{5.28}$$

so that we need, by (5.24), that

$$q^{C^r-C^{r-1}} \sim \frac{\gamma_r}{\gamma_{r-1}} = \frac{v_s^{(r)}}{v_s^{(r-1)}} = O(Q^r q^{(C^r-1)/(C-1)}). \tag{5.29}$$

Since, by our assumption, $C > 1$ and $q < 1$, and (5.29) should hold for *all* sufficiently large r, we need

$$C^r - C^{r-1} \geq \frac{C^r - 1}{C-1} \geq \frac{C^r}{C-1}. \tag{5.30}$$

This holds for all r if $(C-1)^2 \geq C,$

i.e., if $C^2 - 3C + 1 \geq 0,$

i.e., if $C \geq (3 + \sqrt{5})/2$.[26] The tightest bound for (5.29) is clearly obtained when

$$C = \frac{3+\sqrt{5}}{2} \approx 2.618, \tag{5.31}$$

i.e., by (5.27),

$$v_s^{(r)} \sim \kappa Q\, q^{[(3+\sqrt{5})/2]^r} \approx \kappa Q\, q^{2.618^r}. \tag{5.32}$$

[26] The full solution is $C \in (-\infty, (3 - \sqrt{5})/2] \cup [(3 + \sqrt{5})/2, \infty)$; since $\sqrt{5} \approx 2.236$, the first (left) range is *inadmissible*, by our assumption that $C > 1$.

Hence, by (5.28),[27]

$$\begin{aligned} P_r &\sim (q^{(\sqrt{5}-1)/2})^{-1} Q^r (q^{(\sqrt{5}-1)/2})^{[(3+\sqrt{5})/2]^r} \\ &\approx (q^{0.618})^{-1} Q^r (q^{0.618})^{2.618^r}. \end{aligned} \tag{5.33}$$

Of course, by (5.26),

$$0 < q < q^{\dagger} = q^{(\sqrt{5}-1)/2} \approx q^{0.618} < 1, \tag{5.34}$$

and, as $r \to \infty$, however large Q may be, $Q^r (q^{\dagger})^{[(3+\sqrt{5})/2]^r}$ tends to zero faster than $(q^{\dagger} + \varepsilon)^{[(3+\sqrt{5})/2]^r}$ for any $\varepsilon > 0$, however small. For relative simplicity, let us write, for some small ε,

$$p = q^{\dagger} + \varepsilon; \tag{5.35}$$

then, by (5.33),
$$P_r = o(p^{[(3+\sqrt{5})/2]^r}) \approx o(p^{2.618^r}). \tag{5.36}$$

It now follows from (5.19) that

$$\boxed{\|\mathbf{z}^{(r)} - \mathbf{x}_s\| = o(p^{[(3+\sqrt{5})/2]^r}) \approx o(p^{2.618^r})} \tag{5.37}$$

and from (5.21) that

$$\boxed{\left|\rho_r - v_s^{(r-1)}\right| = o(p^{[(3+\sqrt{5})/2]^r}) \approx o(p^{2.618^r})}\,. \tag{5.38}$$

This shows that the accelerative, sequential method increases the convergence of the process from *linear* (sometimes also called *geometric* or *exponential*) to *degree about* 2.618. By comparison, *Newton's method*, which has stood the test of time for over 300 years, converges only *quadratically* (degree 2), under the most general conditions.[28]

Similarly, in the case of Hermitian **H**, we must solve the relations (5.16) and (5.25), and we attempt this by putting, for some

$$Q' \geq 1, \quad C' > 1, \quad \text{and} \quad 0 < q' < 1, \tag{5.39}$$

[27] $(C-1)^{-1} = 2/(\sqrt{5}+1) = 2(\sqrt{5}-1)/(5-1) = (\sqrt{5}-1)/2 \approx 0.618.$

[28] More precisely, if we must solve an equation $f(\xi) = 0$, where f is at least twice-differentiable in some neighborhood of ξ, and if both $f'(\xi) \neq 0$ and $f''(\xi) \neq 0$, then Newton's method is quadratically convergent.

that
$$\gamma'_r = \frac{\left| v_s^{(t)} \right|}{\kappa} \sim Q'q'^{C'^r}. \tag{5.40}$$

Then, by (5.16),
$$P_r = \prod_{t=0}^{r-1} \gamma'_r \sim \prod_{t=0}^{r-1} Q'q'^{C'^t}$$
$$= Q'^r q'^{1+C'+C'^2+\ldots+C'^{r-1}} = Q'^r q'^{(C'^r-1)/(C'-1)}, \tag{5.41}$$

so that we need, by (5.25), that
$$q'^{C'^r - C'^{r-1}} \sim \frac{\gamma'_r}{\gamma'_{r-1}} = \frac{v_s^{(r)}}{v_s^{(r-1)}} = O(Q'^{2r} q'^{2(C'^r-1)/(C'-1)}). \tag{5.42}$$

Arguing just as before, we see that, now,
$$C'^r - C'^{r-1} \geq \frac{2(C'^r - 1)}{C' - 1} \geq \frac{2C'^r}{C' - 1}. \tag{5.43}$$

This holds for all r if $(C' - 1)^2 \geq 2C'$,

i.e., if $C'^2 - 4C' + 1 \geq 0$,

i.e., if $C' \geq 2 + \sqrt{3}$.[29] The tightest bound for (5.42) is clearly obtained, again, when
$$C = 2 + \sqrt{3} \approx 3.732, \tag{5.44}$$

i.e., by (5.40),
$$v_s^{(r)} \sim \kappa Q'q'^{(2+\sqrt{3})^r} \approx \kappa Q'q'^{3.732^r}. \tag{5.45}$$

Hence, by (5.41),[30]
$$P_r \sim (q'^{(\sqrt{3}-1)/2})^{-1} Q'^r (q'^{(\sqrt{3}-1)/2})^{(2+\sqrt{3})^r}$$
$$\approx (q'^{0.366})^{-1} Q'^r (q'^{0.366})^{3.732^r}. \tag{5.46}$$

Of course, by (5.39),

[29] The full solution is $C' \in (-\infty, 2 - \sqrt{3}] \cup [2 + \sqrt{3}, \infty)$; since $\sqrt{3} \approx 1.732$, the first (left) range is *inadmissible*, by our assumption that $C' > 1$.

[30] $(C' - 1)^{-1} = 1/(\sqrt{3} + 1) = (\sqrt{3} - 1)/(3 - 1) = (\sqrt{3} - 1)/2 \approx 0.366.$

$$0 < q' < q'' = q^{(\sqrt{3}-1)/2} \approx q^{0.366} < 1, \tag{5.47}$$

and, as $r \to \infty$, however large Q' may be, $Q'^{r} q''^{(2+\sqrt{3})^r}$ tends to zero faster than $(q'' + \varepsilon)^{(2+\sqrt{3})^r}$ for any $\varepsilon > 0$, however small. For relative simplicity, let us again write, for some small ε,

$$p' = q'' + \varepsilon; \tag{5.48}$$

then, by (5.46), $$P_r = o(p'^{(2+\sqrt{3})^r}) \approx o(p'^{3.732^r}). \tag{5.49}$$

It now follows from (5.19) that

$$\boxed{\left\|\mathbf{z}^{(r)} - \mathbf{x}_s\right\| = o(p'^{(2+\sqrt{3})^r}) \approx o(p'^{3.732^r})} \tag{5.50}$$

and from (5.23), if we put $$p'' = p'^{2}, \tag{5.51}$$

that $$\boxed{\left|\mathcal{T}_r - v_s^{(r-1)}\right| = o(p''^{(2+\sqrt{3})^r}) \approx o(p''^{3.732^r})} \; . \tag{5.52}$$

This shows that the accelerative method increases the convergence of the process even more when we use $\mathcal{T}_r$ than when we use ρ_r— to *degree about* 3.732.

To illustrate this concept in a different way, suppose that we gain d decimal places of accuracy by a single iteration of a *linearly-converging* method. Then this is because the error is multiplied by about 10^{-d}. Thus, subsequent errors after this will be multiplied by about 10^{-d} also; so that we shall gain d decimal places of accuracy at each iteration. However, if we were to gain d decimal places of accuracy by a single iteration of a *quadratically-converging* method, like Newton's method, then we would have, for some $T > 0$ and $0 < t < 1$,

$$10^{-d} = \frac{Tt^{2^{r+1}}}{Tt^{2^r}} = t^{2^r}, \tag{5.53}$$

and subsequent errors would be $Tt^{2^{r+2}}, Tt^{2^{r+3}}$, and so on, successively multiplying the current error, first, by $t^{2^{r+1}} = (t^{2^r})^2 = 10^{-2d}$, then by $t^{2^{r+2}} = (t^{2^r})^{2^2} = 10^{-4d}$, and so on, *doubling* the number of accurate decimal digits gained at each step. This is what leads to the spectacular convergence for which Newton's method is rightly admired. Finally, if we use the present method, in general, it is easily seen that the number of accurate decimal digits gained at each step is successively *multiplied* *by about* 2.618, and by about 3.732 if $\mathbf{H}$ is Hermitian and we use the Rayleigh quotient.

6. COMPUTATIONAL RESULTS

Examples I and II use the (4 × 4) matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 6 & 7 & 8 \\ 3 & 7 & 0 & 0 \\ 4 & 8 & 0 & 1 \end{bmatrix}. \tag{6.1}$$

The initial vectors are

$$\mathbf{v} = \mathbf{u}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{z}^{(0)} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}. \tag{6.2}$$

All values were computed throughout these calculations by *Mathematica*®, to 16 significant digits. Only 12 decimal places are shown here, to save space; but all answers obtained by the methods presented here agreed with the values determined by *Mathematica*® to all 16 significant digits. The error bound[31] ε is 10^{-16}. The eigenvalues are determined by *Mathematica*® to be

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 15.756757465243 \\ -8.552000310307 \\ 0.766185719967 \\ 0.029057125097 \end{bmatrix}. \tag{6.4}$$

[31] See (4.37) and (5.8).

The corresponding eigenvectors are determined by *Mathematica*® to be

$$\mathbf{x}_1 = \begin{bmatrix} 0.306133128240 \\ 0.729060231265 \\ 0.382173871550 \\ 0.478222562084 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0.298644543541 \\ 0.535551558890 \\ -0.543123757521 \\ -0.573596154449 \end{bmatrix},$$

$$\mathbf{x}_3 = \begin{bmatrix} -0.596539846524 \\ 0.313392323685 \\ 0.527452699381 \\ -0.517415802697 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 0.679142220684 \\ -0.288861486553 \\ 0.529861647825 \\ -0.417817567647 \end{bmatrix}. \tag{6.5}$$

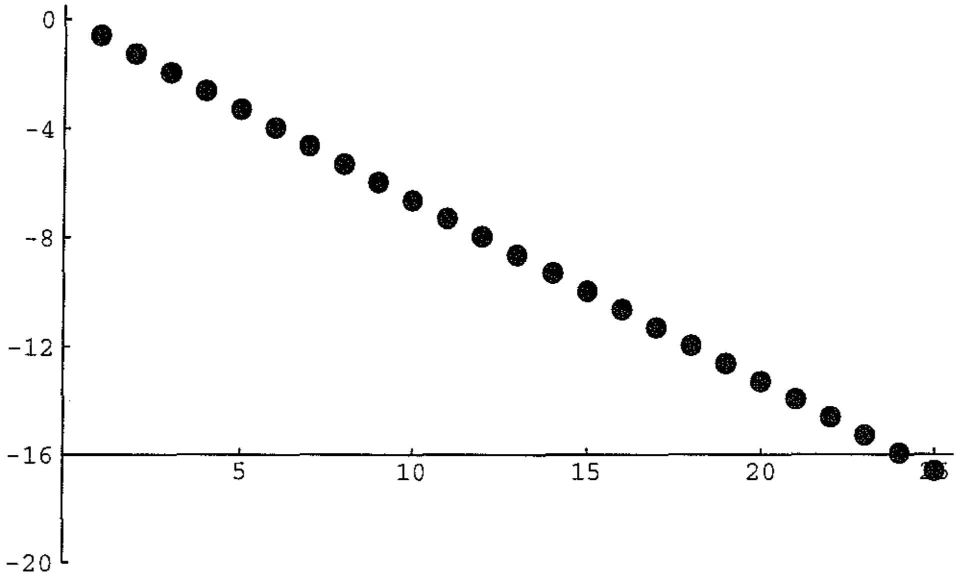
EXAMPLE I.

With the initial estimate $\mu = 20$, the regular inverse power method [IP], given in (4.32)–(4.37), took 25 iterations to converge to within $\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\| \le \varepsilon$. Part of the actual output is shown below.[32]

```
 r=0, lambda(r)=mu+RQ=20.
         Vnorm Error=1.
         Value Error=1.
      z={0.5, 0.5, 0.5, 0.5}
 r=1, lambda(r)=mu+RQ=15.38174510630908
         Vnorm Error=0.2563217959904484
         Value Error=4.618254893690922
      z={0.3558876224787782, 0.6893643064870076,
         0.4063242313279146, 0.4827229766065305}
 r=2, lambda(r)=mu+RQ=15.74106543759154
         Vnorm Error=0.05406817445296523
         Value Error=0.3593203312824598
      z={0.3174254721607415, 0.7215707899472572,
         0.3865625401294966, 0.4786920380443494}
                        . . .
                        . . .
                        . . .
r=24, lambda(r)=mu+RQ=15.75675746524333
         Vnorm Error=1.141759030693568 10^-16
         Value Error=8.673617379884035 10^-19
      z={0.3061331282401873, 0.7290602312648116,
         0.3821738715504973, 0.4782225620838905}
r=25, lambda(r)=mu+RQ=15.75675746524333
         Vnorm Error=2.485478796778427 10^-17
         Value Error=8.673617379884035 10^-19
      z={0.3061331282401873, 0.7290602312648116,
         0.3821738715504973, 0.4782225620838905}
```

[32] "`Vnorm Error`" in the output denotes the current value of $\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\|$. "`Value Error`" denotes the change in the Rayleigh quotient [RQ] estimate, $|\mathcal{T}_r - \mathcal{T}_{r-1}|$. "`lambda(r)`" denotes the current eigenvalue estimate, $\mu + \mathcal{T}_r$.

The graph of $\log_{10}\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\|$ against r is shown below. The linear convergence derived in §4 is vividly demonstrated.



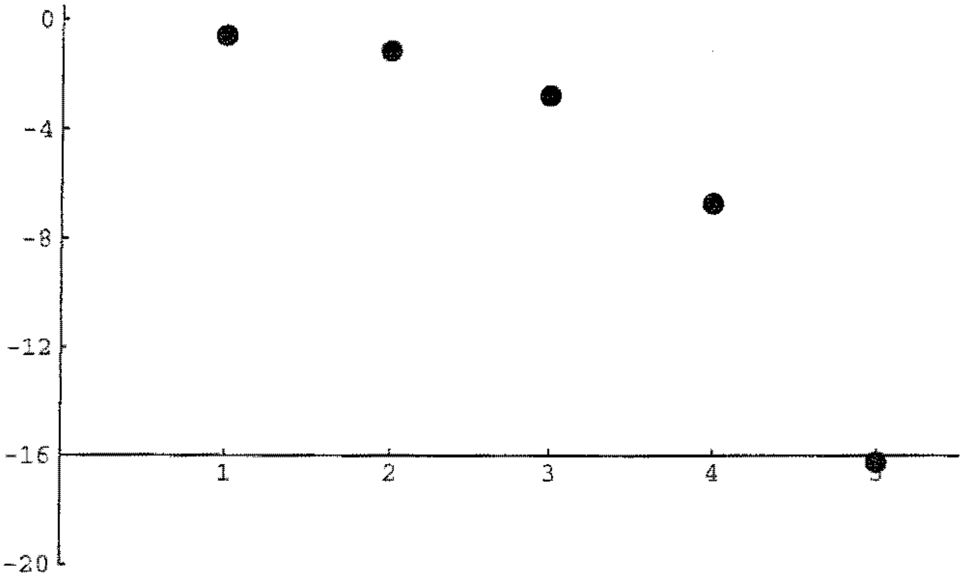
With the same initial initial estimate $\mu = 20$, the accelerated method [AIP], given in (5.1)–(5.8), took only 5 iterations to converge to within the same bound $\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\| \le \varepsilon$. The actual output is shown below.[33]

```
r=0, mu(r)=20.
        Vnorm Error=1.
        Value Error=1.
     z={0.5, 0.5, 0.5, 0.5}
r=1, mu(r)=15.38174510630908
        Vnorm Error=0.2563217959904484
        Value Error=0.3750123589342516
     z={0.3558876224787782, 0.6893643064870076,
        0.4063242313279146, 0.4827229766065305}
r=2, mu(r)=15.75855101712347
        Vnorm Error=0.0698499464770337
        Value Error=4.995060804505312
     z={0.3048158220002099, 0.72989798876998,
        0.3816863090922068, 0.4781754929933601}
r=3, mu(r)=15.75675746044241
        Vnorm Error=0.001635988320891794
        Value Error=0.3785994674954485
     z={0.3061329711820046, 0.7290603208533179,
        0.3821738260128561, 0.4782225624363935}
r=4, mu(r)=15.75675746524333
        Vnorm Error=1.864595780042779 10^-7
        Value Error=0.001793561481979872
     z={0.3061331282401873, 0.7290602312648116,
        0.3821738715504973, 0.4782225620838905}
```

[33] Here, "`mu(r)`" denotes the current eigenvalue estimate, $\mu^{(r)}$.

```
r=5, mu(r)=15.75675746524333
          Vnorm Error=5.741119883861845 10^-17
          Value Error=4.800920948279867 10^-9
      z={0.3061331282401873, 0.7290602312648116,
          0.3821738715504973, 0.4782225620838905}
```

The graph of $\log_{10}\|\mathbf{z}^{(r)}-\mathbf{z}^{(r-1)}\|$ against r is shown below. The super-linear convergence is easily observed.



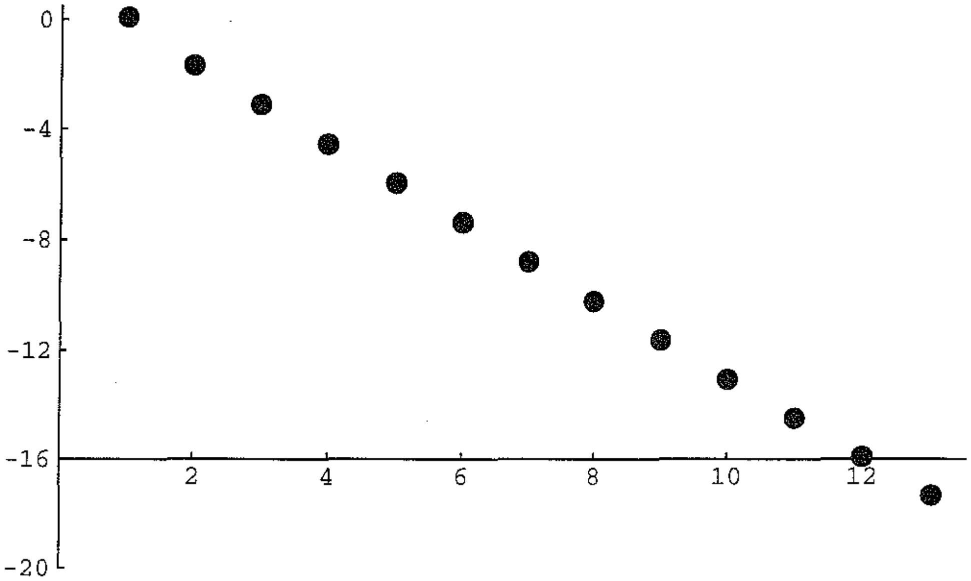
EXAMPLE II.

With the initial estimate $\mu = 0$, IP took 13 iterations to converge to within ε. Part of the actual output is shown below.

```
r=0, lambda(r)=mu+RQ=0.
          Vnorm Error=1.
          Value Error=1.
      z={0.5, 0.5, 0.5, 0.5}
r=1, lambda(r)=mu+RQ=0.4444444444444444
          Vnorm Error=1.216353797721035
          Value Error=0.4444444444444444
      z={0.6939779183594883, -0.2891574659831201,
          0.5204834387696162, -0.4048204523763682}
r=2, lambda(r)=mu+RQ=0.02907052113041224
          Vnorm Error=0.02110164683557881
          Value Error=0.4153739233140322
      z={0.6796104797834141, -0.2891005833208917,
          0.5294574636400052, -0.4174029739807449}
                         . . .
                         . . .
                         . . .
r=12, lambda(r)=mu+RQ=0.02905712509674624
          Vnorm Error=1.220338449750021 10^-16
          Value Error=6.776263578034403 10^-21
      z={0.6791422206843521, -0.2888614865526053,
          0.5298616478250049, -0.4178175676473122}
```

```
r=13, lambda(r)=mu+RQ=0.02905712509674624
          Vnorm Error=4.64288297743686 10^-18
          Value Error=0.
      z={0.6791422206843521, -0.2888614865526053,
          0.5298616478250049, -0.4178175676473122}
```

The graph of $\log_{10}\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\|$ against r is shown below. The linear convergence is again clear.

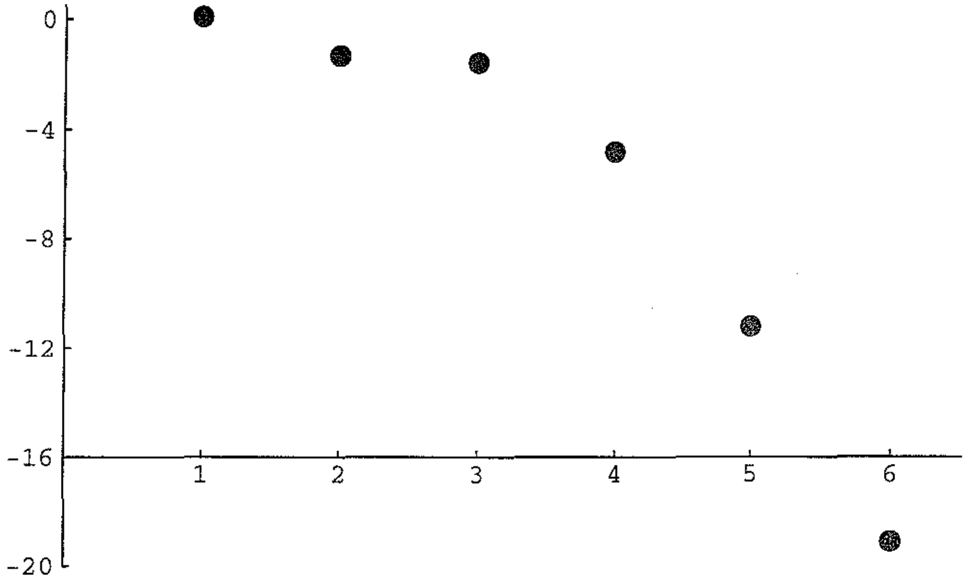


With the same initial estimate $\mu = 0$, AIP took only 6 iterations to converge to within ε. The actual output is shown below.

```
r=0, mu(r)=0.
          Vnorm Error=1.
          Value Error=1.
      z={0.5, 0.5, 0.5, 0.5}
r=1, mu(r)=0.4444444444444444
          Vnorm Error=1.216353797721035
          Value Error=0.4444444444444444
      z={0.6939779183594883, -0.2891574659831201,
          0.5204834387696162, -0.4048204523763682}
r=2, mu(r)=0.02863017320949641
          Vnorm Error=0.04779805000254311
          Value Error=0.8602587156793925
      z={0.6629955957955055, -0.280509769408892,
          0.5435901357026047, -0.4315795101586227}
r=3, mu(r)=0.02905742750381033
          Vnorm Error=0.02659900735452802
          Value Error=0.416241525529262
      z={0.6791330250325291, -0.288856663223585,
          0.5298697762449758, -0.4178255409182185}
r=4, mu(r)=0.02905712509674617
          Vnorm Error=0.0001541002823809772
          Value Error=0.0004275567013780883
      z={0.6791422206881234, -0.2888614865545865,
          0.5298616478216705, -0.417817567644041}
```

```
r=5, mu(r)=0.02905712509674624
          Vnorm Error=6.321964687733428 10^-12
          Value Error=3.024070642389589 10^-7
       z={0.6791422206843521, -0.2888614865526053,
          0.5298616478250049, -0.4178175676473122}
r=6, mu(r)=0.02905712509674624
          Vnorm Error=7.66646708341687 10^-20
          Value Error=7.192869928293844 10^-17
       z={0.6791422206843521, -0.2888614865526053,
          0.5298616478250049, -0.4178175676473122}
```

The graph of $\log_{10}\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\|$ against r is shown below. The super-linear convergence clearly emerges after a couple of false starts.



Examples III–VI use the (4×4) matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 2 & 4 & 16 \\ 2 & 7 & 25 & 125 \\ 4 & 25 & -3 & 81 \\ 16 & 125 & 81 & -111 \end{bmatrix}. \tag{6.6}$$

The initial vectors are again given by (6.2). The eigenvalues are now

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} -206.877064266574 \\ 123.379669314113 \\ -23.086712601609 \\ 0.584107554070 \end{bmatrix}. \tag{6.7}$$

Corresponding eigenvectors are

$$\mathbf{x}_1 = \begin{bmatrix} -0.055066203536 \\ -0.459273707944 \\ -0.277183977131 \\ 0.842142753482 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0.097335742424 \\ 0.683582471329 \\ 0.483269908212 \\ 0.538229462177 \end{bmatrix},$$

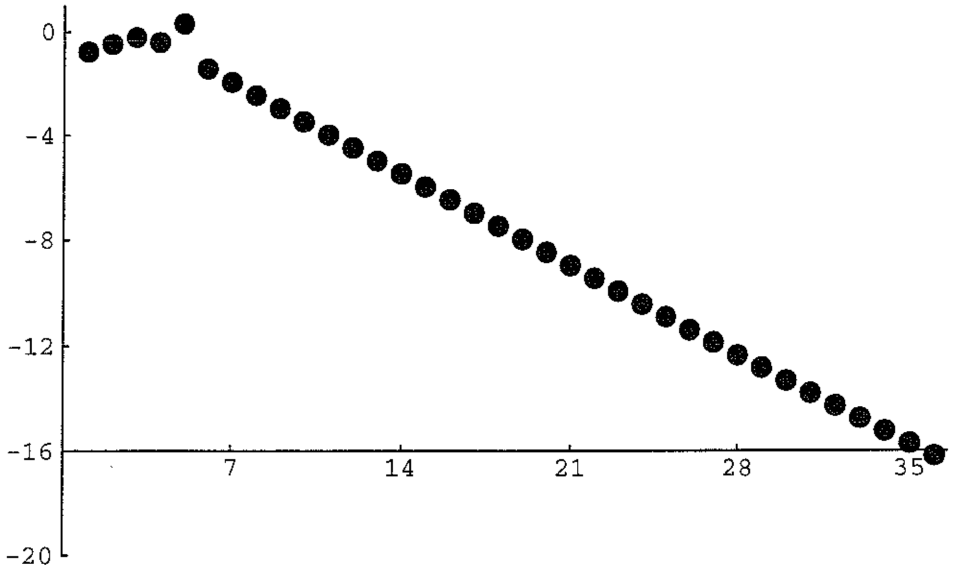
$$\mathbf{x}_3 = \begin{bmatrix} -0.070730731270 \\ -0.551684872783 \\ 0.830425368954 \\ -0.032166308378 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 0.991206653594 \\ -0.132009345219 \\ -0.003597949009 \\ -0.008364054569 \end{bmatrix}. \qquad (6.8)$$

EXAMPLE III.

With the initial estimate $\mu = -300$, IP took 36 iterations to converge to within ε. Part of the actual output is shown below.

```
r=0, lambda(r)=mu+RQ=-300.
          Vnorm Error=1.
          Value Error=1.
     z={0.5, 0.5, 0.5, 0.5}
r=1, lambda(r)=mu+RQ=92.13777152378339
          Vnorm Error=0.168514433738702
          Value Error=392.1377715237834
     z={0.6056346096963456, 0.3746049629111113,
          0.4685041648549635, 0.5228591481593041}
r=2, lambda(r)=mu+RQ=62.36857379171802
          Vnorm Error=0.3323263249874912
          Value Error=29.76919773206538
     z={0.6478200964098767, 0.1118190722872811,
          0.3382783073630464, 0.6733449372557146}
                          . . .
                          . . .
                          . . .
r=35, lambda(r)=mu+RQ=-206.8770642665739
          Vnorm Error=1.952874170585284 10^-16
          Value Error=0.
     z={0.05506620353562544, 0.4592737079437599,
          0.2771839771313178, -0.8421427534815152}
r=36, lambda(r)=mu+RQ=-206.8770642665739
          Vnorm Error=6.526755256843316 10^-17
          Value Error=0.
     z={0.05506620353562546, 0.4592737079437598,
          0.2771839771313178, -0.8421427534815152}
```

The graph of $\log_{10}\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\|$ against r is shown below. The linear convergence again emerges, after some false starts.

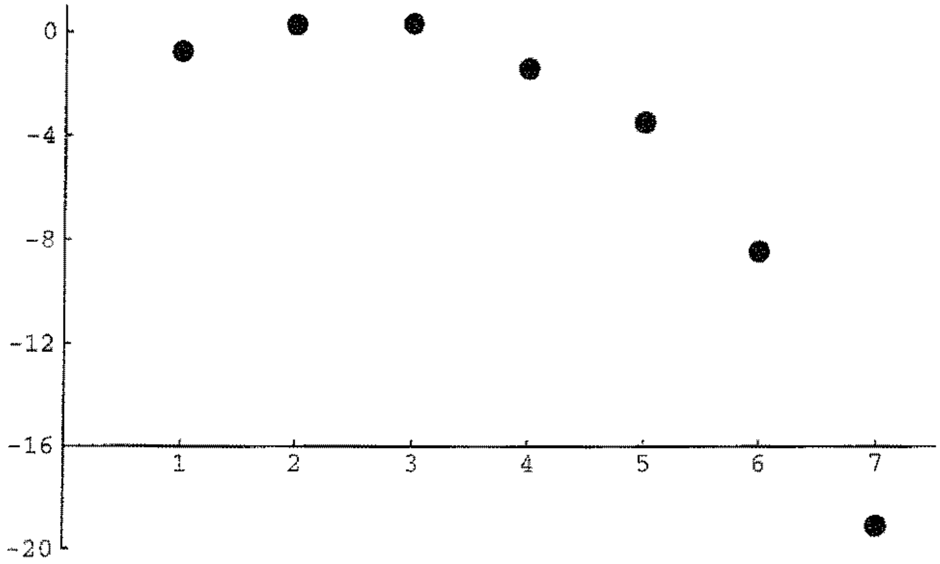


With the same initial initial estimate $\mu = -300$, AIP took 7 iterations to converge to within ε. The actual output is shown below.

```
r=0, mu(r)=-300.
        Vnorm Error=1.
        Value Error=1.
     z={0.5, 0.5, 0.5, 0.5}
r=1, mu(r)=92.13777152378339
        Vnorm Error=0.168514433738702
        Value Error=299.0148357903573
     z={0.6056346096963456, 0.3746049629111113,
        0.4685041648549635, 0.5228591481593041}
r=2, mu(r)=146.8932946710548
        Vnorm Error=1.830722951316671
        Value Error=337.382248376512
     z={0.1198849108324352, -0.7229847016041613,
        -0.4426471056574259, -0.5167052053677653}
r=3, mu(r)=122.3440173628091
        Vnorm Error=1.991049238668443
        Value Error=79.30480045551704
     z={0.06174821539945175, 0.6914026898535324,
        0.4786772901508737, 0.5376035065474362}
r=4, mu(r)=123.3810798026253
        Vnorm Error=0.03704148672163792
        Value Error=25.58633974806183
     z={0.09763888914970034, 0.6835184079039021,
        0.4833010853201706, 0.5382279204114857}
r=5, mu(r)=123.3796693139761
        Vnorm Error=0.000311406851123907
        Value Error=1.038472928465325
     z={0.09733574591277309, 0.6835824706439215,
        0.4832699085141786, 0.538229462144775}
r=6, mu(r)=123.3796693141129
        Vnorm Error=3.568136075032542 10
        Value Error=0.001410488785939755
     z={0.09733574242433119, 0.6835824713293507,
        0.4832699082116358, 0.5382294621767548}
```

```
r=7, mu(r)=123.3796693141129
        Vnorm Error=8.159701876572675 10^-20
        Value Error=1.367826112843432 10^-10
     z={0.09733574242433119, 0.6835824713293507,
        0.4832699082116358, 0.5382294621767548}
```

The graph of $\log_{10}\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\|$ against r is shown below. The super-linear convergence clearly emerges, again after some erring steps.



Note, however, that, this time, the iterations settle down to give, *not*, as we would expect, the nearest eigenvalue to $\mu = -300$, namely, $\lambda_1 = -206.877064266574$ and its eigenvector $\mathbf{x}_1$, but another eigenvalue, $\lambda_2 = 123.379669314113$ and its eigenvector $\mathbf{x}_2$.[34] The explanation is that, despite possible initial wandering, the IP process is firmly tied to proximity to μ, which remains constant. By contrast, in AIP, the parameter $\mu^{(r)}$ varies with r, which is the reason for the fast convergence, but at the same time, the process loses its fixed focus. More specifically, since $\mathbf{H}$ is Hermitian,[35]

$$\begin{aligned}\mathbf{z}^{(0)} &= (0.5, 0.5, 0.5, 0.5)\\ &= (\mathbf{z}^{(0)*}\mathbf{x}_1)\mathbf{x}_1 + (\mathbf{z}^{(0)*}\mathbf{x}_2)\mathbf{x}_2 + (\mathbf{z}^{(0)*}\mathbf{x}_3)\mathbf{x}_3 + (\mathbf{z}^{(0)*}\mathbf{x}_4)\mathbf{x}_4\\ &= 0.025309432435\mathbf{x}_1 + 0.901208792071\mathbf{x}_2\\ &\qquad + 0.087921728261\mathbf{x}_3 + 0.423617652398\mathbf{x}_4. \end{aligned} \tag{6.9}$$

[34] See (6.4) and (6.5).

[35] See (1.6), (3.3), and (6.8).

Thus, the dominant eigenvector is $\mathbf{x}_2$, and we can verify that, by (4.7),

$$\begin{aligned}\mathbf{N}\mathbf{z}^{(0)} &= \frac{\mathbf{z}^{(0)*}\mathbf{x}_1}{\lambda_1-\mu}\mathbf{x}_1 + \frac{\mathbf{z}^{(0)*}\mathbf{x}_2}{\lambda_2-\mu}\mathbf{x}_2 + \frac{\mathbf{z}^{(0)*}\mathbf{x}_3}{\lambda_3-\mu}\mathbf{x}_3 + \frac{\mathbf{z}^{(0)*}\mathbf{x}_4}{\lambda_4-\mu}\mathbf{x}_4 \\ &= \frac{0.025309432435}{93.122935733426}\mathbf{x}_1 + \frac{0.901208792071}{423.379669314113}\mathbf{x}_2 \\ &\qquad + \frac{0.087921728261}{276.913287398391}\mathbf{x}_3 + \frac{0.423617652398}{300.584107554070}\mathbf{x}_4 \\ &= 0.000271785165\mathbf{x}_1 + 0.002128606680\mathbf{x}_2 \\ &\qquad + 0.000317506354\mathbf{x}_3 + 0.001409314870\mathbf{x}_4. \qquad (6.10)\end{aligned}$$

Evidently, the $\mathbf{x}_2$ component still dominates, and the large Rayleigh quotient correction, +392.137771523783, will reflect this (as we see in the computer output). After three iterations, the computed point in $(\mu^{(r)}, \mathbf{z}^{(r)})$-space is firmly located near enough to $(\lambda_2, \mathbf{x}_2)$ to remain there.

EXAMPLE IV.

In an attempt to overcome the difficulty arising in Example III, when one seeks the *largest* eigenvalue and a corresponding eigenvector, one can create a *hybrid method*, which first performs power method [PM] iterations until $\|\mathbf{z}^{(r)} - \mathbf{z}^{(r-1)}\| < \delta$, say, so that the kind of thrashing-about seen above is made much less likely, and then continues with either IP or AIP iterations. We start at $\mu = -300$, as before.[36]

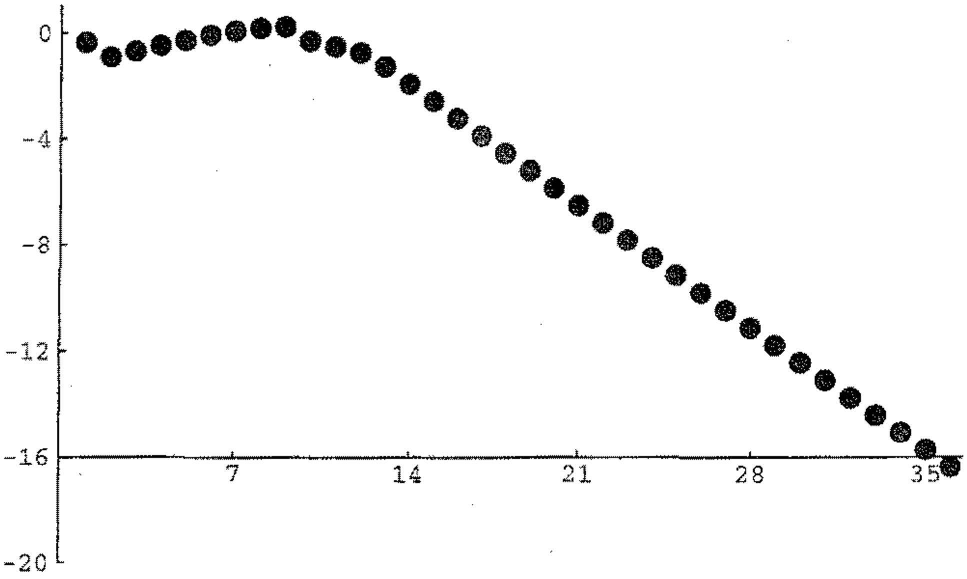
Computing the PM iterations with $\delta = 0.3$,[37] the method took 12 PM iterations[38] to converge to within the first bound δ, and 24 more IP iterations (for a total of 36, as in Example III) to converge to within the second bound ε.

[36] During the PM phase, it is not necessary to estimate the eigenvalue at all.

[37] This was arrived at by trial and error. Note that $\varepsilon = 10^{-16}$, as before.

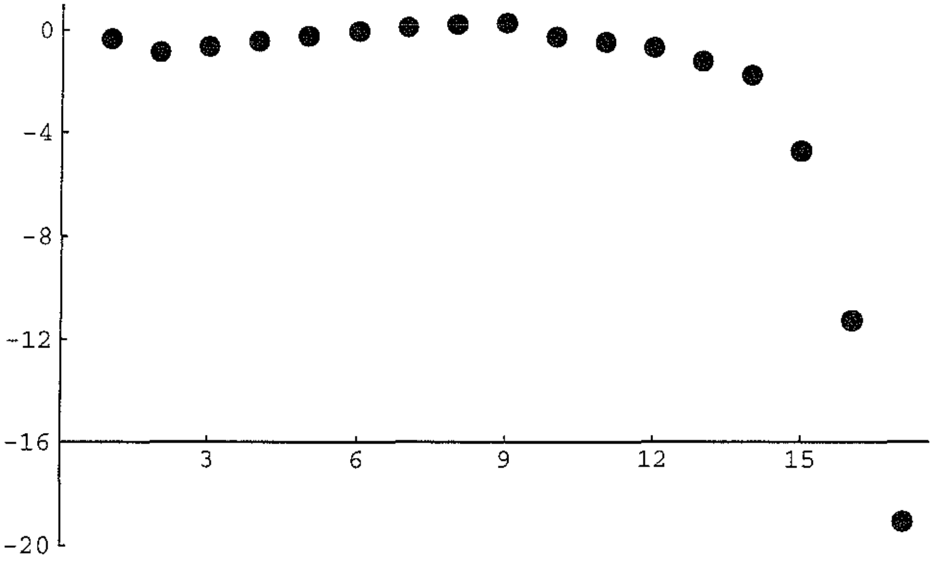
[38] The acute observer will note that "`Vnorm Error`" fell below $\delta = 0.3$ after 2 iterations. However, in all the programs used here, there was an additional condition preventing termination (exiting the loop) until $r \geq 4$. We note the "`Vnorm Error`" increases as far as 1.823880063366 (for $r = 9$) before dropping back down to 0.191981790985 (for $r = 12$).

```
  r=0,     Vnorm Error=1.
  r=1,     Vnorm Error=0.4512036820809108
  r=2,     Vnorm Error=0.1276222884234312
  r=3,     Vnorm Error=0.2100714088735469
  r=4,     Vnorm Error=0.3482898195298783
  r=5,     Vnorm Error=0.5668994029558392
  r=6,     Vnorm Error=0.8827847381837408
  r=7,     Vnorm Error=1.262309744760841
  r=8,     Vnorm Error=1.603791858874887
  r=9,     Vnorm Error=1.823880063366282
 r=10,     Vnorm Error=0.521522427100639
 r=11,     Vnorm Error=0.3189290380473227
 r=12, lambda(r)=mu+RQ=-300.
           Vnorm Error=0.1919817909845165
           Value Error=1.
       z={0.04792418037206863, 0.40892793358177,
           0.2417141929360335, -0.8786668691862069}
 r=13, lambda(r)=mu+RQ=-206.4999106345518
           Vnorm Error=0.05610871826911112
           Value Error=93.50008936544824
       z={0.0535159378608344,  0.4483771747042251,
           0.2694864653746498, -0.8505709838567324}
                          . . .
                          . . .
                          . . .
 r=35, lambda(r)=mu+RQ=-206.8770642665739
           Vnorm Error=1.910111649564712 10^-16
           Value Error=1.387778780781446 10^-17
       z={0.05506620353562546, 0.4592737079437598,
           0.2771839771313178, -0.8421427534815153}
 r=36, lambda(r)=mu+RQ=-206.8770642665739
           Vnorm Error=4.207428179014308 10^-17
           Value Error=0.
       z={0.05506620353562546, 0.4592737079437598,
           0.2771839771313178, -0.8421427534815152}
```



The same problem, program, and parameters, with AIP, took the *same* 12 PM iterations to converge to within δ, followed by only 5 more AIP iterations (for a total of 17—as compared with only 7 iterations in Example III, but to the "wrong" eigenvalue and eigenvector!) to converge to within ε. Note that, now, the iterations do indeed converge to the nearest eigenvalue to –300, namely, $\lambda_1 = -206.877064266574$, and to its eigenvector $\mathbf{x}_1$.

```
  r=0,     Vnorm Error=1.
  r=1,     Vnorm Error=0.4512036820809108
  r=2,     Vnorm Error=0.1276222884234312
  r=3,     Vnorm Error=0.2100714088735469
  r=4,     Vnorm Error=0.3482898195298783
  r=5,     Vnorm Error=0.5668994029558392
  r=6,     Vnorm Error=0.8827847381837408
  r=7,     Vnorm Error=1.262309744760841
  r=8,     Vnorm Error=1.603791858874887
  r=9,     Vnorm Error=1.823880063366282
 r=10,     Vnorm Error=0.521522427100639
 r=11,     Vnorm Error=0.3189290380473227
 r=12, mu(r)=-300.
           Vnorm Error=0.1919817909845165
           Value Error=1.
       z={0.04792418037206863, 0.40892793358177,
           0.2417141929360335, -0.8786668691862069}
 r=13, mu(r)=-206.4999106345518
           Vnorm Error=0.05610871826911112
           Value Error=0.377153632022128
       z={0.0535159378608344,  0.4483771747042251,
           0.2694864653746498, -0.8505709838567324}
 r=14, mu(r)=-206.8771592185895
           Vnorm Error=0.01587450126555757
           Value Error=93.87733794948602
       z={0.05506796826347818, 0.4592861014986463,
           0.2771927389672755, -0.8421329950223585}
 r=15, mu(r)=-206.8770642665739
           Vnorm Error=0.00001813040464427551
           Value Error=0.377343536053472
       z={0.05506620353613284, 0.4592737079473231,
           0.277183977133837,  -0.8421427534787096}
 r=16, mu(r)=-206.8770642665739
           Vnorm Error=5.212667346914816 10^-12
           Value Error=0.00009495201571879723
       z={0.05506620353562546, 0.4592737079437598,
           0.2771839771313178, -0.8421427534815152}
 r=17, mu(r)=-206.8770642665739
           Vnorm Error=8.194797530478168 10^-20
           Value Error=3.121204001056531 10^-14
       z={0.05506620353562546, 0.4592737079437598,
           0.2771839771313178, -0.8421427534815152}
```



EXAMPLE V.

Another, more direct, attempt to overcome the difficulty arising in Example III, when seeking the nearest eigenvalue to $\mu = -300$, and a corresponding eigenvector, one can try a more appropriate initial vector,[39]

$$\mathbf{v} = \mathbf{u}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{z}^{(0)} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \\ 0.5 \end{bmatrix}. \tag{6.11}$$

We then observe that[40]

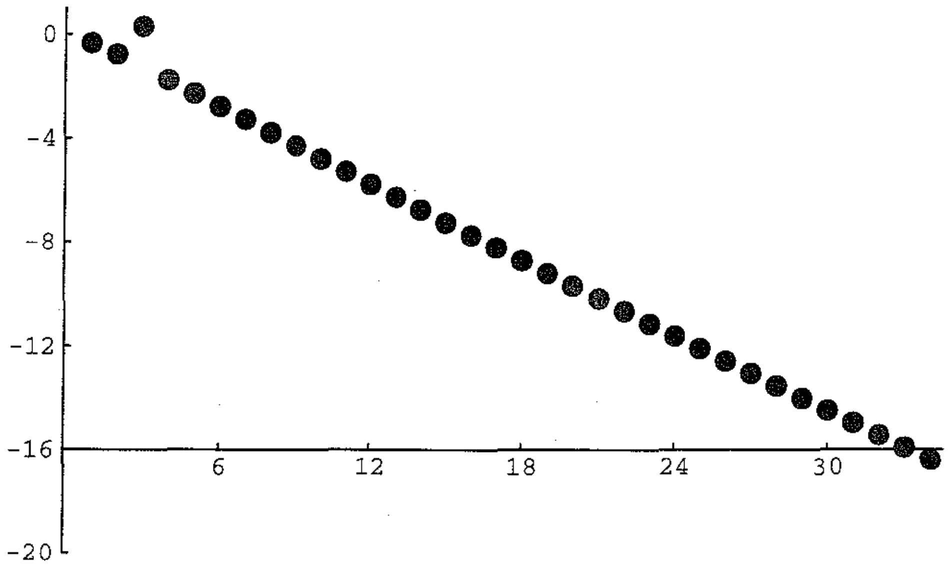
$$\begin{aligned} \mathbf{z}^{(0)} &= (0.5, -0.5, -0.5, 0.5) \\ &= (\mathbf{z}^{(0)*}\mathbf{x}_1)\mathbf{x}_1 + (\mathbf{z}^{(0)*}\mathbf{x}_2)\mathbf{x}_2 + (\mathbf{z}^{(0)*}\mathbf{x}_3)\mathbf{x}_3 + (\mathbf{z}^{(0)*}\mathbf{x}_4)\mathbf{x}_4 \\ &= 0.761767117510\mathbf{x}_1 - 0.265643587470\mathbf{x}_2 \\ &\qquad - 0.190818767909\mathbf{x}_3 + 0.559224946627\mathbf{x}_4. \end{aligned} \tag{6.12}$$

Now we see that the $\mathbf{x}_1$ component of $\mathbf{z}^{(0)}$ is dominant, and we can expect good behavior. For IP, with $\mu = -300$, the improvement is only minor: it now takes 34 (rather than 36) iterations to converge to within ε.

[39] Compare (6.2).

[40] Compare (6.9), with the same (6.7) and (6.8).

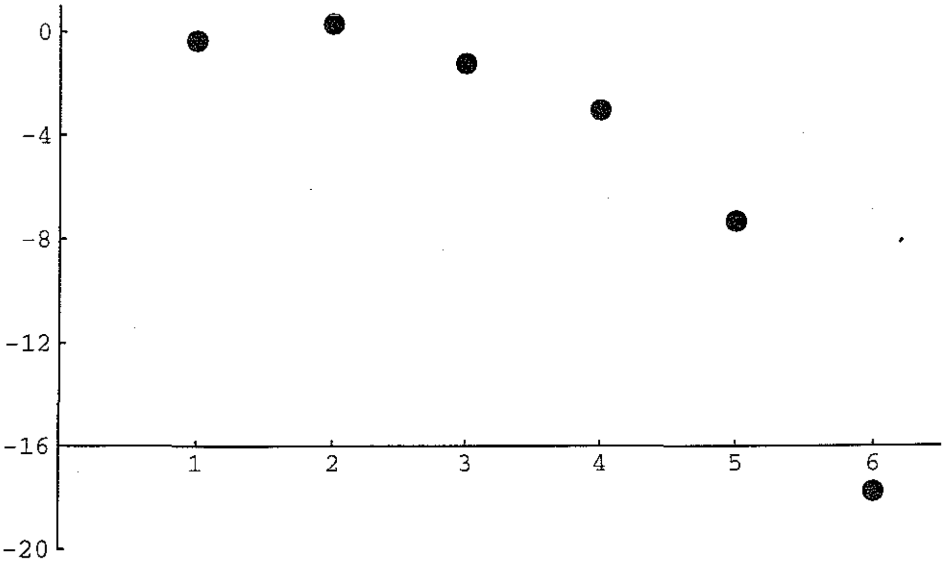
```
r=0,  lambda(r)=mu+RQ=-300.
          Vnorm Error=1.
          Value Error=1.
      z={0.5, -0.5, -0.5, 0.5}
r=1,  lambda(r)=mu+RQ=-167.8998311514028
          Vnorm Error=0.4542184052739204
          Value Error=132.1001688485972
      z={0.163648642843276, -0.4799706900344641,
        -0.3731408524051213, 0.7769254550275553}
r=2,  lambda(r)=mu+RQ=-202.7675177044516
          Vnorm Error=0.1717107461002237
          Value Error=34.86768655304878
      z={0.0150906880049384, -0.4630800312400062,
        -0.3081841733338339,  0.8308740404591023}

                            . . .
                            . . .
                            . . .
r=33,  lambda(r)=mu+RQ=-206.8770642665739
          Vnorm Error=1.218745707169549 10^-16
          Value Error=2.081668171172169 10^-17
      z={0.05506620353562545, 0.4592737079437598,
          0.2771839771313179, -0.8421427534815152}
r=34:   Final Eigenvalue Estimate=-206.8770642665739
          Vnorm Error=4.082209718626033 10^-17
          Value Error=2.081668171172169 10^-17
      z={0.05506620353562546, 0.4592737079437598,
          0.2771839771313179, -0.8421427534815152}
```



For AIP, with $\mu = -300$, the improvement is more marked: it now takes 6 (rather than 7) iterations to reach the bound ε, but we also converge to the "right" eigenvalue and eigenvector.

```
r=0,  mu(r)=-300.
          Vnorm Error=1.
          Value Error=1.
      z={0.5, -0.5, -0.5, 0.5}
```

```
r=1, mu(r)=-167.8998311514028
        Vnorm Error=0.4542184052739204
        Value Error=38.97723311517105
     z={0.163648642843276, -0.4799706900344641,
       -0.3731408524051213, 0.7769254550275553}
r=2, mu(r)=-210.0166857168165
        Vnorm Error=1.97651854190868
        Value Error=174.2170234140109
     z={0.1076407849746226,  0.4570469671255226,
        0.2527782259173111, -0.8459460383224839}
r=3, mu(r)=-206.8665883920687
        Vnorm Error=0.0572564294969489
        Value Error=45.2669518901615
     z={0.05586118889832137, 0.4593140043780091,
        0.2768181005343005, -0.8421887627923364}
r=4, mu(r)=-206.8770642746366
        Vnorm Error=0.000877318263624 9167
        Value Error=3.1605732073157
     z={0.05506616303741576, 0.4592737032928834,
        0.2771839967818216, -0.8421427521982436}
r=5, mu(r)=-206.8770642665739
        Vnorm Error=4.527167685961772 10^-8
        Value Error=0.01047589063063442
     z={0.05506620353562546, 0.4592737079437598,
        0.2771839771313178, -0.8421427534815152}
r=6, mu(r)=-206.8770642665739
        Vnorm Error=1.801597463405437 10^-18
        Value Error=8.062737393889801 10^-9
     z={0.05506620353562546, 0.4592737079437598,
        0.2771839771313178, -0.8421427534815152}
```

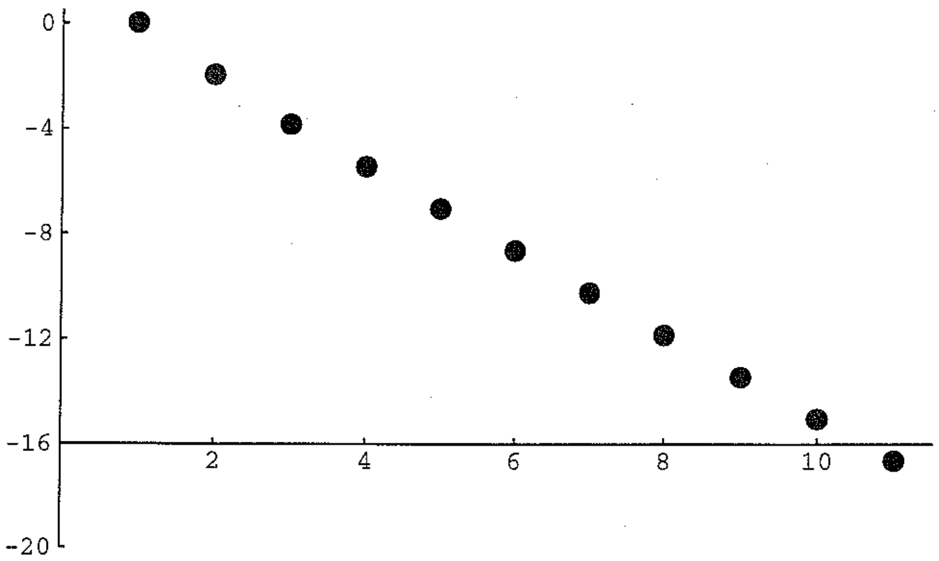


Of course, it may be argued that this kind of “ad-hockery” is unfair, in realistic situations.

EXAMPLE VI.

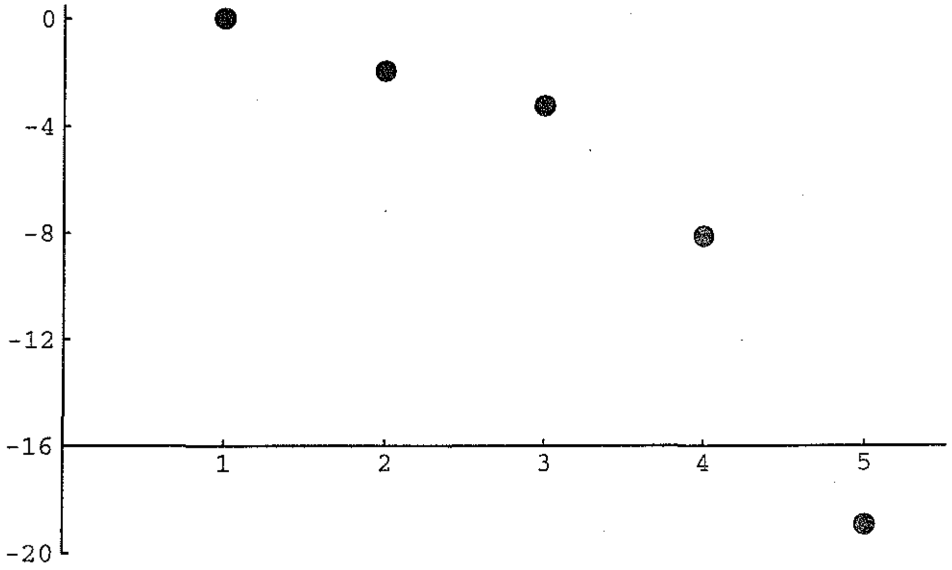
With the original (non-hybrid) program and initial estimate $\mu = 0$, IP took 11 iterations to converge to within ε.

```
 r=0, lambda(r)=mu+RQ=0.
          Vnorm Error=1.
          Value Error=1.
      z={0.5, 0.5, 0.5, 0.5}
 r=1, lambda(r)=mu+RQ=3.190108993845926
          Vnorm Error=1.06564490488079
          Value Error=3.190108993845926
      z={0.9925036557816415,   -0.1221422173292132,
        -0.003044342292083236, -0.002916159458732363}
 r=2, lambda(r)=mu+RQ=0.5841830549253467
          Vnorm Error=0.01137907549194079
          Value Error=2.605925938920579
      z={0.9912018615500829,   -0.1320502634781779,
        -0.003464710350629079, -0.008342263312816713}
                     . . .
                     . . .
                     . . .
r=10, lambda(r)=mu+RQ=0.5841075540696886
          Vnorm Error=9.039646669689694 10^-16
          Value Error=0.
      z={0.9912066535939302,   -0.1320093452192289,
        -0.003597949009324531, -0.008364054569363917}
r=11, lambda(r)=0.5841075540696886
          Vnorm Error=2.28753905383707 10^-17
          Value Error=0.
      z={0.9912066535939302,  -0.1320093452192288,
        -0.00359794900932455, -0.008364054569363917}
```



With the same program and initial estimate $\mu = 0$, but the AIP method, it took only 5 iterations to converge to within ε.

```
r=0, mu(r)=0.
        Vnorm Error=1.
        Value Error=1.
     z={0.5, 0.5, 0.5, 0.5}
r=1, mu(r)=3.190108993845926
        Vnorm Error=1.06564490488079
        Value Error=2.606001439776237
     z={0.9925036557816415,   -0.1221422173292132,
       -0.003044342292083236, -0.002916159458732363}
r=2, mu(r)=0.583772669145121
        Vnorm Error=0.01132590803146794
        Value Error=5.79644531854673
     z={0.9912221899970447,   -0.1318703355908494,
       -0.004135373048095307, -0.008466601347197999}
r=3, mu(r)=0.5841075541764865
        Vnorm Error=0.0005647237464109799
        Value Error=2.60667120973217
     z={0.9912066530146393,   -0.1320093496926433,
       -0.003597943179590633, -0.008364055124060995}
r=4, mu(r)=0.5841075540696886
        Vnorm Error=7.391921351924644 10^-9
        Value Error=0.0003348851381633282
     z={0.9912066535939302,  -0.1320093452192288,
       -0.00359794900932455, -0.008364054569363917}
r=5, mu(r)=0.5841075540696886
       Vnorm Error=1.110622998317774 10^-19
       Value Error=1.06797909846106 10^-10
     z={0.9912066535939302,  -0.1320093452192288,
       -0.00359794900932455, -0.008364054569363917}
```



Finally, in Examples VII and VIII, we consider the Hilbert matrix, renowned for its recalcitrance. For any given n, and $h, k = 1, 2, \ldots, n$, we have

$$H_{hk} = \frac{1}{h+k}. \tag{6.13}$$

Mathematica® then tells us that, when $n = 20$, the 20 eigenvalues are

1.495352204385832, 0.2662138191329257, 0.03350135685965074, 0.003473843326499648,
0.0003043209080894834, 0.00002274233432443311, 1.457389096680593 10^{-6}, 8.027465070767417 10^{-8},
3.800859824118565 10^{-9}, 1.543912655621322 10^{-10}, 5.358255526957966 10^{-12}, 1.578654924397504 10^{-13},
3.911865116114538 10^{-15}, 8.047716736391524 10^{-17}, 1.349641003305385 10^{-18}, 1.508302996225652 10^{-20},
2.762323149294507 10^{-21}, 2.06645914621975 10^{-21}, -5.886670361439534 10^{-23}, 6.110811071171222 10^{-24}

The 20 eigenvectors are given by *Mathematica*® as follows.

```
{0.5041806365514644, 0.3935825403429613, 0.3282539151595543, 0.2838867511484085,
 0.2512909913174308, 0.2260954759866652, 0.2059125745832828, 0.1893104475692812,
 0.1753704071242256, 0.1634719901755816, 0.1531789076720339, 0.1441742216070118,
 0.1362214256525829, 0.129140006700855, 0.1227895147144929, 0.1170588444755607,
 0.111858825213388, 0.1071169748048021, 0.1027737085982034, 0.09877954890054407},

{0.6480239115184489, 0.2387143716657485, 0.05079113886076188, -0.04986417629134016,
 -0.1083609326660147, -0.1439127643390442, -0.1659610496201564, -0.1796157153666856,
 -0.1878422455740795, -0.1924547174398204, -0.1946093388895351, -0.195067494630972,
 -0.1943439865528096, -0.1927944703331703, -0.1906690222286761, -0.1881460250678076,
 -0.185354200292569, -0.1823872763093872, -0.1793139580611349, -0.17618482690396},

{0.4834833960931377, -0.2502307771048199, -0.3699666852917489, -0.3360562764599146,
 -0.263269890568063, -0.1860340773706191, -0.1150682115386825, -0.05317691847992966,
 -0.0004008869645613846, 0.04413785715855579, 0.08154381728667568, 0.1128901151285482,
 0.1391315233445164, 0.161085562463415, 0.1794397284134993, 0.1947674221478341,
 0.2075454053018027, 0.2181699960730938, 0.2269711068810826, 0.2342240172573906},

{0.2713669281541966, -0.5383712285507337, -0.2715322451777863, 0.001310281583693457,
 0.1678862495053529, 0.2466780402475296, 0.2670722345691564, 0.2515301150453019,
 0.2151087091026247, 0.1674737277261281, 0.114706208074917, 0.06057151597682305,
 0.007352783327976641, -0.04361295011437891, -0.09159130923302719, -0.1362301639169136,
-0.1774176038229354, -0.2151900855695717, -0.2496728962235361, -0.2810415744307477},

{0.1252243740882118, -0.5179681500946902, 0.1596968313156553, 0.3639743075602481,
 0.2949954564446806, 0.1407757224588609, -0.01033302450713386, -0.125895465748964,
 -0.1989676655147132, -0.2329195805769525, -0.2345131778768188, -0.2109345187557612,
 -0.1686417775001603, -0.1130211676869036, -0.04838414045533918, 0.02190698365838858,
 0.09527943310463125, 0.1697959973105349, 0.2440209990238481, 0.3169094126963807},

{0.04959790803883326, -0.3512242069090768, 0.4683020287693117, 0.2394032162900977,
 -0.1019634167152302, -0.2769168020749264, -0.289228942055834, -0.2043182520994467,
 -0.08117548159877127, 0.04074051128486709, 0.1390902349414806, 0.2034684675088582,
 0.2309292830633975, 0.2228113229112095, 0.1826736446658321, 0.1150257040248895,
 0.02458327443797373, -0.08414212905798609, -0.207048939827325, -0.3405291194798499},

{-0.01727376405685467, 0.188748165635823, -0.4991001678821485, 0.1667206012114114,
 0.3670345819360689, 0.1927164630237408, -0.0559344106692483, -0.2224982172750302,
 -0.2705813984971959, -0.221764401327087, -0.115689975907641, 0.009070682803628391,
 0.1221012986297096, 0.2024093513042962, 0.2373156407509627, 0.2206360817944708,
 0.150883382367082, 0.02975354175903626, -0.1390476674755348, -0.3506680720016088},

{-0.005367256792747952, 0.08470482677354368, -0.3631869311962081, 0.449104622222499,
 0.1603185533648228, -0.2287235271498013, -0.3143188967281199, -0.1661306244025298,
 0.0466119193799537, 0.2043220295466777, 0.258805825894044, 0.2138005248968111,
 0.1009095408015005, -0.0383256919252122, -0.1632319887545191, -0.2394223201670633,
 -0.2407770851465791, -0.1494810175944184, 0.04490196654837962, 0.346726636152667},

{0.00150099878839796, -0.03265771361278547, 0.2058947514266868, -0.4724781896685332,
 0.2342051181008476, 0.3375322631910225, 0.01068436119984088, -0.2538016502779332,
 -0.277534678686319, -0.1212696775142714, 0.08178690391456504, 0.2235128230190346,
 0.2526362375843638, 0.1713237778175301, 0.01906142324232151, -0.1445699703398631,
 -0.2536753387118064, -0.2454210872225956, -0.0655378526097891, 0.329083429528218},
```

{-0.0003797253406780959, 0.01099954664660682, -0.09608062238091676, 0.3409160974497724,
-0.4579637973891156, -0.02547496254582848, 0.3333400899054442, 0.2046874120297418,
-0.1036059385958212, -0.2768580175129968, -0.2259611694520045, -0.03282550568782794,
0.1651935910138099, 0.2592052747427298, 0.2065875523991033, 0.03521194521684132,
-0.1701640386882847, -0.2877708065854425, -0.178275926161359, 0.2992757498981749},

{0.0000871014521163186, -0.003268693483995639, 0.03797770853182227, -0.1908126604283491,
0.4366860370783452, -0.3346164802194856, -0.22564846859623, 0.2147071834978153,
0.2980315003429365, 0.0467245454331403, -0.216017763776778, -0.2700385961308855,
-0.1105891207087062, 0.1223863734246819, 0.2629161479087287, 0.213075909600473,
-0.008192388360209092, -0.256106256387057, -0.2772774794186706, 0.2599861765541569},

{0.00001811583651580042, -0.0008615428268279387, 0.01291982732079156, -0.08716761079194454,
0.2935624731118381, -0.4634417522731047, 0.1616258781929904, 0.3273463281598678,
-0.06653189056783016, -0.3082443583298225, -0.1536113722523564, 0.1481515994132132,
0.2817016174917599, 0.1468640057926992, -0.1194659111154249, -0.2803239803184798,
-0.1738195149016436, 0.148487465913019, 0.3476018891522521, -0.2148131506350096},

{-3.409023920931223 10^{-6}, 0.0002017438667946826, -0.003814763840115894, 0.03333444516584534,
-0.1534370059886022, 0.3780049525445956, -0.4265476292541117, -0.002413047399803785,
0.347469756914429, 0.05592312106850736, -0.2801867952992514, -0.2126469542654212,
0.1079877473449268, 0.288255414889392, 0.1390205503672768, -0.1737817463012076,
-0.29447846937787, -0.01392471666610966, 0.3788768522083848, -0.1678403486475497},

{-5.777722404495544 10^{-7}, 0.00004189910985663479, -0.0009804087665769293, 0.01080514448922657,
-0.06495371553659007, 0.2241862341271318, -0.4296346574420421, 0.3545237247163165,
0.1246438948147871, -0.3265819901687861, -0.1328547941975982, 0.2548894310850409,
0.2341681036150333, -0.1159627050923964, -0.2993029403197905, -0.06371616693719829,
0.2855531131957914, 0.190230159058885, -0.3681302977493858, 0.1230765937664016},

{8.766454886290479 10^{-8}, -7.690497208365699 10^{-6}, 0.0002193341749795074, -0.002988145918648076,
0.02274948796649071, -0.1038332702635005, 0.287789624358488, -0.4538693637786623,
0.2857956123567423, 0.1953865921968175, -0.3059642564939714, -0.1602011845475979,
0.2613941909140993, 0.2147635358925148, -0.1872585681393275, -0.2795405670969035,
0.1328883031455838, 0.3275301833526263, -0.3181329181957755, 0.08327901868962491},

{-1.18004200461032 10^{-8}, 1.240587548102007 10^{-6}, -0.00004267160724397834, 0.0007091301306221839,
-0.006708416763953261, 0.03921627889132849, -0.1468799023724086, 0.3508139848746969,
-0.4964304810403104, 0.2921788752684586, 0.1917672234229964, -0.3286916493651176,
-0.09154801533553359, 0.2924409855806304, 0.1009183884548201, -0.2993746407405913,
-0.07163663030877223, 0.3523832506500895, -0.2275587688576928, 0.04844183099601651},

{-2.855351088850798 10^{-9}, 3.070167201045139 10^{-7}, -0.00001084729650846593, 0.0001859345199800574,
-0.00182097915886021, 0.01104380490633145, -0.0427809570759132, 0.1033846110629457,
-0.1312940750878218, -0.01812657374817054, 0.3798546031813172, -0.6545437480868981,
0.4881888290426953, -0.05845581698388477, -0.09409576074925591, -0.1187755299988843,
0.2862607311712608, -0.2121056440391469, 0.07302472194196313, -0.009933607679575157},

{3.241146263864129 10^{-9}, -3.509225793592372 10^{-7}, 0.00001252304755767436, -0.0002176485845281631,
0.002172953242058473, -0.01354981822846365, 0.05483038047088615, -0.1436002181836039,
0.2252878495930163, -0.1341197311733123, -0.2153013241922545, 0.5467932849600198,
-0.4597641228227631, 0.08236101383895025, 0.0263409995591055, 0.2400971002395031,
-0.4248445180689863, 0.3055491095977352, -0.1072920951555556, 0.0152446094718594},

{-8.478751721189957 10^{-10}, 8.733814552053651 10^{-8}, -2.957065916292503 10^{-6}, 0.00004871607764825144,
-0.0004623343018794425, 0.002768607885355166, -0.01103053295613856, 0.03005556055052029,
-0.05565863446441461, 0.06173386334362336, 0.001420673004167286, -0.1820686134850123,
0.4529318729709084, -0.635522732959227, 0.5169856922460922, -0.1482579697910053,
-0.1471354709301253, 0.1802073551354304, -0.07944844972485333, 0.01343526803255518},

{-9.096073580970803 10^{-10}, 9.380376764715067 10^{-8}, -3.17910244280599 10^{-6}, 0.00005240241169167076,
-0.0004971124000081888, 0.002970039982875677, -0.01176470588338262, 0.03167398528543261,
-0.05734097019742619, 0.06072531161295895, 0.006872130124165105, -0.1880602124300221,
0.4533228382607304, -0.6326466701222113, 0.5210086966221203, -0.1622447650756229,
-0.1319010942028362, 0.1716270706987682, -0.07692164709623166, 0.01312778867723054}

EXAMPLE VII

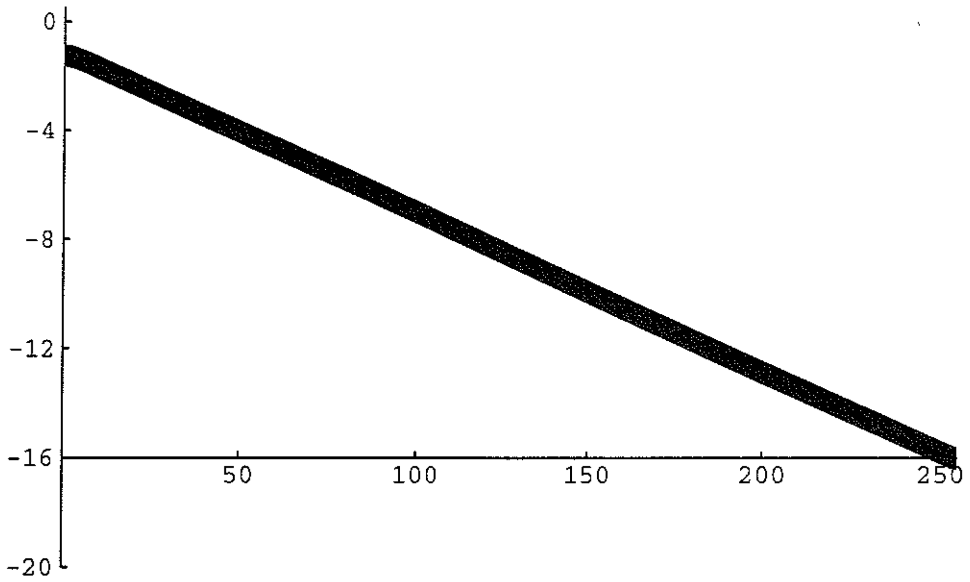
Starting with $n = 20$, $\mu = 10$, and an initial vector[41] given, for $h = 1, 2, \ldots, n$, by

$$z_h^{(0)} = \frac{1}{\sqrt{n}}. \tag{6.14}$$

we find that it takes 254 IP iterations to converge to within ε.

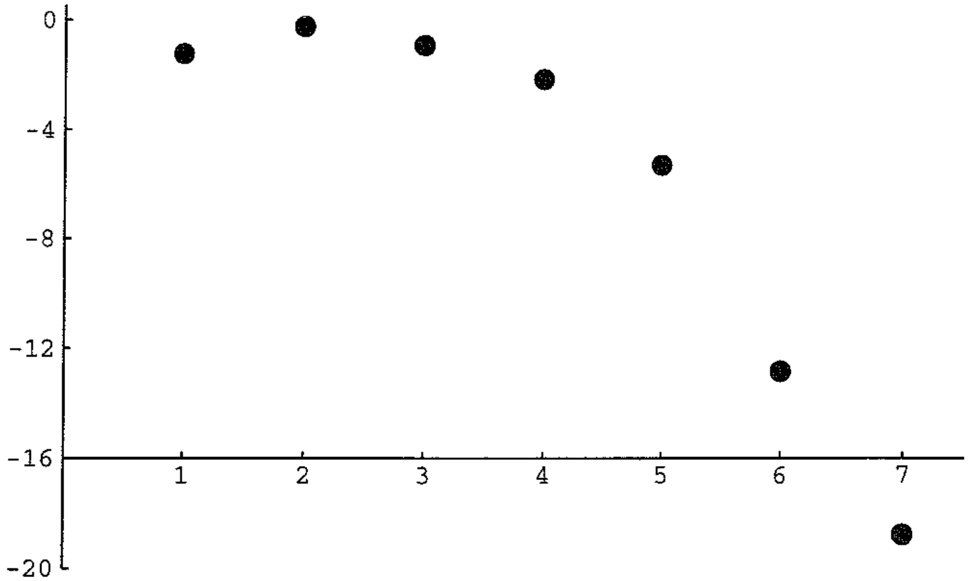
```
                     r=0, lambda(r)=mu+RQ=10.
                            Vnorm Error=1.
                            Value Error=20.
                     r=1, lambda(r)=mu+RQ=1.24474526409473
                            Vnorm Error=0.05519056193654381
                            Value Error=8.75525473590527
                     r=2, lambda(r)=mu+RQ=1.295752607295662
                            Vnorm Error=0.05046166517794363
                            Value Error=0.05100734320093126
                                            . . .
                                            . . .
                                            . . .
                   r=253, lambda(r)=mu+RQ=1.495352204385832
                            Vnorm Error=1.061788040474785 10^-16
                            Value Error=1.734723475976807 10^-18
                   r=254, lambda(r)=1.495352204385832
                            Vnorm Error=9.278524252499485 10^-17
                            Value Error=8.673617379884035 10^-19
z={0.504180636551464, 0.3935825403429611, 0.3282539151595543, 0.2838867511484085,
   0.2512909913174309, 0.2260954759866653, 0.2059125745832829, 0.1893104475692813,
   0.1753704071242257, 0.1634719901755817, 0.153178907672034, 0.1441742216070119,
   0.136221425652583, 0.1291400067008551, 0.122789514714493, 0.1170588444755609,
   0.1118588252133881, 0.1071169748048023, 0.1027737085982035, 0.09877954890054418}
```

[41] Compare (6.2).



By contrast, it takes only 7 AIP iterations to converge to within ε.

```
                    r=0, mu(r)=10.
                            Vnorm Error=10.
                            Value Error=20.
                    r=1, mu(r)=1.24474526409473
                            Vnorm Error=0.05519056193654381
                            Value Error=8.75525473590527
                    r=2, mu(r)=1.56710149200233
                            Vnorm Error=0.5445567524177484
                            Value Error=9.07761096381287
                    r=3, mu(r)=1.494430050597607
                            Vnorm Error=0.1097001883694708
                            Value Error=0.3950276693123232
                    r=4, mu(r)=1.495352241680401
                            Vnorm Error=0.006361740039393082
                            Value Error=0.07359363248751653
                    r=5, mu(r)=1.495352204385832
                            Vnorm Error=4.730034941734246 10^-6
                            Value Error=0.0009222283773617332
                    r=6, mu(r)=1.495352204385832
                            Vnorm Error=1.425955877959313 10^-13
                            Value Error=3.729456834207726 10^-8
                    r=7, mu(r)=1.495352204385832
                            Vnorm Error=1.704873442047949 10^-19
                            Value Error=8.88245211254223 10^-19
z={0.5041806365514644, 0.3935825403429613, 0.3282539151595543, 0.2838867511484085,
   0.2512909913174308, 0.2260954759866652, 0.2059125745832828, 0.1893104475692812,
   0.1753704071242256, 0.1634719901755816, 0.1531789076720339, 0.1441742216070118,
   0.1362214256525829, 0.129140006700855, 0.1227895147144929, 0.1170588444755607,
   0.111858825213388, 0.1071169748048021, 0.1027737085982034, 0.09877954890054407}
```



EXAMPLE VIII

For $n = 100$ and the same μ and initial vector (6.14), *Mathematica*® gives us the 100 eigenvalues

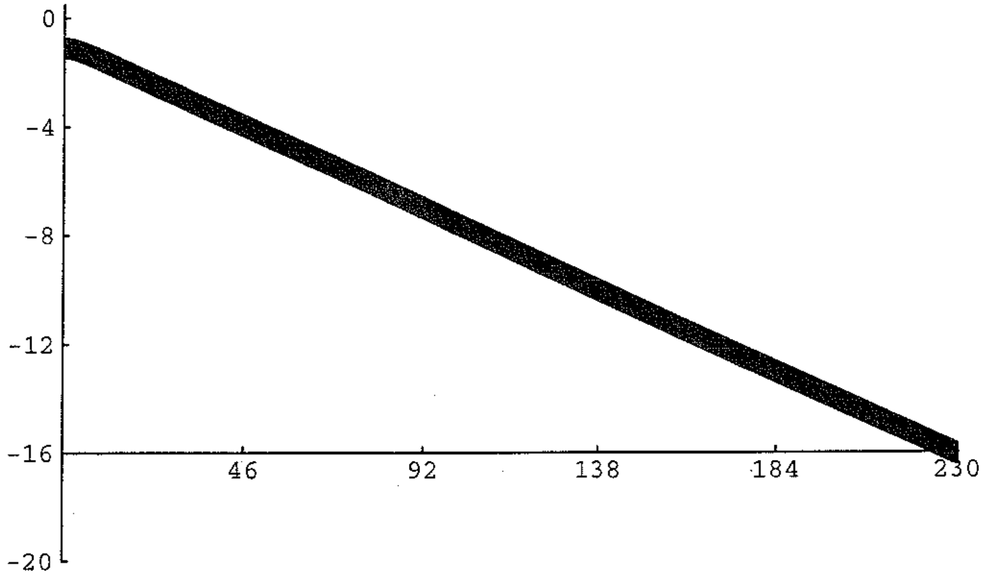
1.880008825927228, 0.5550331165448031, 0.1266319531030056, 0.02600124907395677, 0.004952987576952973, 0.0008840359236167338, 0.0001487228892239044, 0.00002368601211885019, 3.583520096842338 10^{-6}, 5.164739048780091 10^{-7}, 7.107498106624535 10^{-8}, 9.357609145541284 10^{-9}, 1.180640193202602 10^{-9}, 1.429548890235539 10^{-10}, 1.663227041160557 10^{-11}, 1.861442112572553 10^{-12}, 2.005905380218932 10^{-13}, 2.08306765120074 10^{-14}, 2.086200668003608 10^{-15}, 2.016314073697826 10^{-16}, 1.881857797960655 10^{-17}, 1.694421353560435 10^{-18}, 1.330809513600019 10^{-19}, 2.61720988477372 10^{-20}, –1.52282772510051 10^{-21}, –1.336103448444317 10^{-21}, 1.311346328440333 10^{-21}, 1.163644958442451 10^{-21}, 1.077798909423929 10^{-21}, –1.063789782751367 10^{-21}, 7.852097542996503 10^{-22}, 7.611274237987543 10^{-22}, –7.525784411885117 10^{-22}, –7.211720939154661 10^{-22}, 7.079696157207022 10^{-22}, 6.923933821964385 10^{-22}, 6.777429899511379 10^{-22}, –6.690751005493032 10^{-22}, –6.382042074407996 10^{-22}, 6.097497399198586 10^{-22}, 6.058352934119355 10^{-22}, 5.91768604119817 10^{-22}, 5.768396941562 10^{-22}, –5.740623748580146 10^{-22}, 5.243095546538781 10^{-22}, –4.979652877295539 10^{-22}, –4.66963503375929 10^{-22}, 4.598945975645116 10^{-22}, –4.462969256975034 10^{-22}, 4.423977023229678 10^{-22}, –4.326117001617834 10^{-22}, 4.234261040245412 10^{-22}, –3.9487559193658 10^{-22}, 3.86942888078468 10^{-22}, –3.848148423819167 10^{-22}, –3.845946594592571 10^{-22}, –3.799124438568015 10^{-22}, 3.770012957947321 10^{-22}, –3.752011084299232 10^{-22}, –3.711291210097171 10^{-22}, –3.671052285833306 10^{-22}, 3.661192482760501 10^{-22}, 3.605431111522212 10^{-22}, –3.563378525719871 10^{-22}, –3.482626389003396 10^{-22}, 3.390196090977445 10^{-22}, –3.214506921974549 10^{-22}, –2.970677192957928 10^{-22}, 2.965412918802807 10^{-22}, 2.915908207410664 10^{-22}, –2.642300656258056 10^{-22}, 2.586927435528196 10^{-22}, –2.499552892954179 10^{-22}, 2.427610956702677 10^{-22}, 2.214799684152595 10^{-22}, 1.968279597182291 10^{-22}, –1.892488305052214 10^{-22}, 1.880227259660107 10^{-22}, –1.862143275094695 10^{-22}, –1.579288319286261 10^{-22}, 1.572480798316952 10^{-22}, –1.495728146536534 10^{-22}, –1.390965438076585 10^{-22}, 1.265450098137378 10^{-22}, –1.034147965248146 10^{-22}, 9.39195257528831 10^{-23}, –8.984891568204238 10^{-23}, 8.872198114407388 10^{-23}, 7.092000562339177 10^{-23}, –6.851842141050946 10^{-23}, 4.736305263649143 10^{-23}, –4.091966322076936 10^{-23}, –3.51778599308963 10^{-23}, –2.926611202455219 10^{-23}, –2.726157836091398 10^{-23}, –2.517353127540145 10^{-23}, 2.289791613116807 10^{-23}, –1.219975411830902 10^{-23}, 1.184545480113436 10^{-23}, 1.139047565247706 10^{-23}

The 100 eigenvectors are also computed by *Mathematica*®, but their listing would be too voluminous and not very useful, since we only seek the largest eigenvalue λ_1 and its eigenvector $\mathbf{x}_1$. This last, is given by *Mathematica*® as follows.

```
{0.3879474482439856, 0.3156131226042447, 0.2718998512938576, 0.2416083114723318,
 0.2189419903478028, 0.2011218174436219, 0.1866184494639301, 0.1745082269257985,
 0.1641944548499937, 0.1552714535740959, 0.147452105188247 7, 0.1405265019296156,
 0.1343370191102279, 0.1287626015907389, 0.1237084818225791, 0.1190992359144487,
 0.1148739650077249, 0.1109828717232243, 0.1073847769907385, 0.1040452857419471,
 0.1009354096806065, 0.09803051803407117, 0.09530952759009177, 0.09275426994592504,
 0.09034899179789101, 0.08807995635764446, 0.08593512251607043, 0.08390388440889482,
 0.08197685836292265, 0.08014570734214793, 0.07840299532050804, 0.07674206572244142,
 0.07515693935924757, 0.07364222826445048, 0.07219306257703752, 0.07080502819640552,
 0.06947411337967638, 0.06819666280187692, 0.06696933787528452, 0.06578908234312143,
 0.06465309233756929, 0.06355879023247926, 0.06250380173456868, 0.06148593574899702,
 0.06050316663038739, 0.05955361849201051, 0.05863555129664224, 0.05774734849463766,
 0.05688750600969215, 0.05605462240189986, 0.05524739006212454, 0.0544645873122114,
 0.05370507130287324, 0.05296777161572998, 0.05225168448841742, 0.05155586759227474,
 0.05087943530117054, 0.05022155439778234, 0.04958144017030742, 0.04895835285832328,
 0.04835159441147471, 0.04776050552895774, 0.04718446295149808, 0.04662287698076431,
 0.04607518920398373, 0.04554087040400013, 0.04501941863717729, 0.04451035746345109,
 0.04401323431450349, 0.04352761898750343, 0.04305310225315813, 0.04258929456796733,
 0.04213582488159034, 0.04169233953113871, 0.0412585012150099, 0.04083398803959195,
 0.04041849263280621, 0.04001172131902418, 0.03961339335040364, 0.03922324019014483,
 0.03884100484357653, 0.03846644123334919, 0.03809931361534232, 0.03773939603219129,
 0.03738647180160642, 0.03704033303690004, 0.03670078019735568, 0.03636762166627188,
 0.0360406733546929, 0.03571975832900108, 0.03540470646069388, 0.03509535409680289,
 0.03479154374953466, 0.03449312380382455, 0.03419994824159657, 0.03391187638161503,
 0.03362877263389856, 0.03335050626774461, 0.03307695119248374, 0.03280798575014788}
```

It took 230 IP iterations to converge to within ε.

```
          r=0, lambda(r)=mu+RQ=10.
                 Vnorm Error=1.
                 Value Error=100.
          r=1, lambda(r)=mu+RQ=1.39081346702972
                 Vnorm Error=0.07736812684567432
                 Value Error=8.60918653297028
          r=2, lambda(r)=mu+RQ=1.490347703819621
                 Vnorm Error=0.07233446029730175
                 Value Error=0.09953423678990068
                            ...
                            ...
                            ...
        r=229, lambda(r)=mu+RQ=1.880008825927228
                 Vnorm Error=1.04329820294765 10^-16
                 Value Error=1.734723475976807 10^-18
        r=230, lambda(r)=mu+RQ=1.880008825927228
                 Value Error=1.734723475976807 10^-18
                 Vnorm Error=8.972121623123063 10^-17
z={0.3879474482439853, 0.3156131226042445, 0.2718998512938575, 0.2416083114723317,
   0.2189419903478028, 0.2011218174436219, 0.1866184494639301, 0.1745082269257985,
   0.1641944548499937, 0.1552714535740959, 0.147452105188247 7, 0.1405265019296156,
   0.134337019110228, 0.1287626015907389, 0.1237084818225792, 0.1190992359144487,
   0.114873965007725, 0.1109828717232244, 0.1073847769907386, 0.1040452857419472,
   0.1009354096806065, 0.09803051803407122, 0.09530952759009182, 0.09275426994592508,
0.09034899179789106, 0.08807995635764451, 0.08593512251607048, 0.08390388440889487,
0.0819768583629227, 0.08014570734214798, 0.07840299532050809, 0.07674206572244147,
0.07515693935924761, 0.07364222826445053, 0.07219306257703757, 0.07080502819640556,
0.06947411337967643, 0.06819666280187697, 0.06696933787528457, 0.06578908234312148,
0.06465309233756933, 0.06355879023247931, 0.06250380173456873, 0.06148593574899707,
0.06050316663038743, 0.05955361849201055, 0.05863555129664229, 0.05774734849463771,
0.05688750600969219, 0.0560546224018999, 0.05524739006212458, 0.05446458731221144,
0.05370507130287329, 0.05296777161573003, 0.05225168448841747, 0.05155586759227478,
0.05087943530117058, 0.05022155439778238, 0.04958144017030746, 0.04895835285832332,
0.04835159441147476, 0.04776050552895778, 0.04718446295149812, 0.04662287698076435,
0.04607518920398377, 0.04554087040400016, 0.04501941863717733, 0.04451035746345113,
0.04401323431450353, 0.04352761898750347, 0.04305310225315817, 0.04258929456796736,
0.04213582488159038, 0.04169233953113875, 0.04125850121500993, 0.04083398803959199,
0.04041849263280624, 0.04001172131902422, 0.03961339335040368, 0.03922324019014487,
0.03884100484357657, 0.03846644123334923, 0.03809931361534236, 0.03773939603219132,
0.03738647180160645, 0.03704033303690008, 0.03670078019735571, 0.03636762166627191,
0.03604067335469293, 0.03571975832900111, 0.03540470646069391, 0.03509535409680292,
0.03479154374953469, 0.03449312380382458, 0.0341999482415966, 0.03391187638161507,
0.0336287726338986, 0.03335050626774464, 0.03307695119248377, 0.03280798575014791}
```



It took only 8 AIP iterations to converge to within ε.

```
r=0, mu(r)=10.
        Vnorm Error=10.
        Value Error=100.
r=1, mu(r)=1.39081346702972
        Vnorm Error=0.07736812684567432
        Value Error=8.60918653297028
r=2, mu(r)=2.323499697877416
        Vnorm Error=0.9031249820550541
        Value Error=9.541872763817976
r=3, mu(r)=1.836555047158089
        Vnorm Error=0.2606938699687896
        Value Error=1.419630881567023
r=4, mu(r)=1.880373811621394
        Vnorm Error=0.09287371059880845
        Value Error=0.5307634151826321
```

```
                  r=5, mu(r)=1.880008822621152
                          Vnorm Error=0.003009246571974264
                          Value Error=0.04418375346354659
                  r=6, mu(r)=1.880008825927228
                          Vnorm Error=8.219505757837224 10^-7
                          Value Error=0.0003649923063175174
                  r=7, mu(r)=1.880008825927228
                          Vnorm Error=2.041063705835868 10^-15
                          Value Error=3.306075817546101 10^-9
                  r=8, mu(r)=1.880008825927228
                          Vnorm Error=2.739781718899118 10^-19
                          Value Error=2.755064884739736 10^-40
z={0.3879474482439856, 0.3156131226042447, 0.2718998512938576, 0.2416083114723318,
   0.2189419903478028, 0.2011218174436219, 0.1866184494639301, 0.1745082269257985,
   0.1641944548499937, 0.1552714535740959, 0.1474521051882477, 0.140526501929615 6,
   0.1343370191102279, 0.1287626015907389, 0.1237084818225791, 0.1190992359144487,
   0.1148739650077249, 0.1109828717232243, 0.1073847769907385, 0.1040452857419471,
   0.1009354096806065, 0.09803051803407117, 0.09530952759009177, 0.09275426994592504,
   0.09034899179789101, 0.08807995635764446, 0.08593512251607043, 0.08390388440889482,
   0.08197685836292265, 0.08014570734214793, 0.07840299532050804, 0.07674206572244142,
   0.07515693935924757, 0.07364222826445048, 0.07219306257703752, 0.07080502819640552,
   0.06947411337967638, 0.06819666280187692, 0.06696933787528452, 0.06578908234312143,
   0.06465309233756929, 0.06355879023247926, 0.06250380173456868, 0.06148593574899702,
   0.06050316663038739, 0.05955361849201051, 0.05863555129664224, 0.05774734849463766,
   0.05688750600969215, 0.05605462240189986, 0.05524739006212454, 0.0544645873122114,
   0.05370507130287324, 0.05296777161572998, 0.05225168448841742, 0.05155586759227474,
   0.05087943530117054, 0.05022155439778234, 0.04958144017030742, 0.04895835285832328,
   0.04835159441147471, 0.04776050552895774, 0.04718446295149808, 0.04662287698076431,
   0.04607518920398373, 0.04554087040400013, 0.04501941863717729, 0.04451035746345109,
   0.04401323431450349, 0.04352761898750343, 0.04305310225315813, 0.04258929456796733,
   0.04213582488159034, 0.04169233953113871, 0.0412585012150099, 0.04083398803959195,
   0.04041849263280621, 0.04001172131902418, 0.03961339335040364, 0.03922324019014483,
   0.03884100484357653, 0.03846644123334919, 0.03809931361534232, 0.03773939603219129,
   0.03738647180160642, 0.03704033303690004, 0.03670078019735568, 0.03636762166627188,
   0.0360406733546929, 0.03571975832900108, 0.03540470646069388, 0.03509535409680289,
   0.03479154374953466, 0.03449312380382455, 0.03419994824159657, 0.03391187638161503,
   0.03362877263389856, 0.03335050626774461, 0.03307695119248374, 0.03280798575014788}
```

CONCLUSIONS

These admittedly preliminary findings are very promising. The method appears to live up to the predictions of its theoretical analysis in a very satisfactory manner, and presents an extremely rapid way to find, to great precision, the eigenvalue nearest to any given value μ, and a corresponding eigenvector. The rate of convergence, most generally of degree 2.618, and reaching 3.732, for Hermitian matrices, when a Rayleigh quotient is employed, is extremely impressive. The only possible problems are (i) the possible presence of multiple eigenvalues (and, if the matrix is not normal,[42] of an asymmetric Jordan form): this problem is well understood and intrinsically troublesome to all algorithms; (ii) the possibility that the algorithm should stray away

[42] A matrix $\mathbf{H}$ is *normal* if and only if $\mathbf{H}^*\mathbf{H} = \mathbf{H}\mathbf{H}^*$.

from the eigenvalue being sought (this is discussed a little in Examples III–V). These are common problems of many algorithms and do not much lessen the potential utility of this new tool.

7. REFERENCES

M. A. CELIA, W. G. GRAY, 1992. *Numerical Methods for Differential Equations—Fundamental Concepts for Scientific and Engineering Applications.* Prentice-Hall, Englewood Cliffs, NJ (1992) 436 pp.

G. DAHLQUIST, Å. BJÖRCK, 1974. *Numerical Methods,* Prentice-Hall, Englewood Cliffs, NJ (1974) 573 pp.

B. N. DATTA, 1995. *Numerical Linear Algebra and Applications.* Brooks/Cole Publishing, Pacific Grove, CA (1995) 680 pp.

G. E. FORSYTHE, C. B. MOLER, 1967. *Computer Solution of Linear Algebraic Systems.* Prentice-Hall, Inc., Englewood Cliffs, NJ (1967) 148 pp.

G. GOLUB, J. M. ORTEGA, 1992. *Scientific Computing and Differential Equations—An Introduction to Numerical Methods.* Academic Press, San Diego, CA (1992) 337 pp.

A. S. HOUSEHOLDER, 1964. *The Theory of Matrices in Numerical Analysis,* Blaisdell Publishing Co., New York, NY (1964) 257 pp.

E. ISAACSON, H. B. KELLER, 1966. *Analysis of Numerical Methods*, John Wiley & Sons, New York (1966) 541 pp.

M. MARCUS, H. MINC, 1964. *A Survey of Matrix Theory and Matrix Inequalities.* Allyn & Bacon, Boston, MA (1964) 180 pp.

B. NOBLE, 1969. *Applied Linear Algebra.* Prentice-Hall, Englewood Cliffs, NJ (1969) 523 pp.

J. M. ORTEGA, 1988. *Introduction to Parallel and Vector Solution of Linear Systems,* Plenum Press, New York, NY (1988) 305 pp.

J. M. ORTEGA, W. C. RHEINBOLDT, 1970. *Iterative Solution of Nonlinear Equations in Several Variables,* Academic Press, Inc., New York, NY (1970) 572 pp.

A. M. OSTROWSKI, 1966. *Solution of Equations and Systems of Equations,* Academic Press, Inc., New York, NY (1966) 338 pp.

S. PISSANETZKY, 1984. *Sparse Matrix Technology*, Academic Press, Inc., London, England (1984) 321 pp.

W. C. RHEINBOLDT, 1974. *Methods for Solving Systems of Nonlinear Equations,* SIAM, Philadelphia, PA (1974) 104 pp.

Y. SAAD, 1996. *Iterative Methods for Sparse Linear Systems*, PWS Publishing Co., Boston, MA (1996) 447 pp.

G. W. STEWART, 1973. *Introduction to Matrix Computations,* Academic Press, New York (1973) 441 pp.

J. F. TRAUB, 1964. *Iterative Methods for the Solution of Equations,* Prentice-Hall, Inc., Englewood Cliffs, NJ (1964) 310 pp.

R. S. VARGA, 1962. *Matrix Iterative Analysis,* Prentice-Hall, Englewood Cliffs, NJ (1962) 322 pp.

J. H. WILKINSON, 1965. *The Algebraic Eigenvalue Problem,* Clarendon Press (Oxford University Press), Oxford, England (1965) 662 pp.

D. M. YOUNG, 1971. *Iterative Solution of Large Linear Systems,* Academic Press, Inc., New York, NY (1971) 570 pp.

D. ZWILLINGER, 1992. *Handbook of Differential Equations*, Academic Press, Inc., San Diego, CA, Second Edition (1992) 787 pp.