1. Root finding

Formulate Newton's method for solving the nonlinear 2×2 system of equations

$$\begin{cases} f(x,y) &= 0 \\ g(x,y) &= 0 \end{cases}.$$

In the same style as how one proves quadratic convergence in the scalar case for $f(x)=0$, show quadratic convergence (assuming sufficient smoothness of *f*, *g*, root being simple, etc.) in the 2×2 case. Assuming the root $x=\alpha, y=\beta$ to be of multiplicity one, define $\varepsilon_n = x_n - \alpha,\ \eta_n = y_n - \beta$, and show that both ε_{n+1} and η_{n+1} are of size $O(\varepsilon_n^2, \eta_n^2)$..

Solution:

We first recall the proof for the scalar case $f(x)=0$. Newton then becomes $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$. With $x_n = \alpha + \varepsilon_n$, this can be written as $f'(x_n)(\varepsilon_{n+1} - \varepsilon_n) = -f(x_n)$. Re-expanding around the root α gives

$[f'(\alpha)+O(\varepsilon_n)](\varepsilon_{n+1}-\varepsilon_n) = -[\underbrace{f(\alpha)}_{=0} + \varepsilon_n f'(\alpha) + O(\varepsilon_n^2)]$. The expression $\varepsilon_n f'(\alpha)$ cancels between the two sides, and we are left with $\varepsilon_{n+1} = O(\varepsilon_n^2)$.

In the 2x2 case, the Newton system takes the form

$$\begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix}_{\text{eval. at } x_n, y_n} \begin{bmatrix} x_{n+1}-x_n \\ y_{n+1}-y_n \end{bmatrix} = -\begin{bmatrix} f \\ g \end{bmatrix}_{\text{eval. at } x_n, y_n}. \qquad (1)$$

For simplicity, denote quantities of sizes $O(\varepsilon_n, \eta_n), O(\varepsilon_n^2, \eta_n^2)$ by $O(\Delta), O(\Delta^2)$, respectively. Expanding the matrix and the right hand side (RHS) of (1) around the root (α,β) (rather than evaluating at (x_n, y_n)) changes (1) to

$$\left\{\begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix}_{\text{eval. at } \alpha,\beta} + \begin{bmatrix} O(\Delta) \\ \vdots \end{bmatrix}\right\}\begin{bmatrix} \varepsilon_{n+1}-\varepsilon_n \\ \eta_{n+1}-\eta_n \end{bmatrix} = -\left\{\underbrace{\begin{bmatrix} f \\ g \end{bmatrix}_{\text{eval. at } \alpha,\beta,}}_{\text{becomes zero}} + \begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix}_{\text{eval. at } \alpha,\beta}\begin{bmatrix} \varepsilon_n \\ \eta_n \end{bmatrix} + \begin{bmatrix} O(\Delta^2) \\ \vdots \end{bmatrix}\right\}$$

Multiplying by the inverse of $\begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix}_{\text{eval. at } \alpha,\beta}$ gives $\left\{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} O(\Delta) & \cdots \\ \vdots & \ddots \end{bmatrix}\right\}\begin{bmatrix} \varepsilon_{n+1}-\varepsilon_n \\ \eta_{n+1}-\eta_n \end{bmatrix} = -\begin{bmatrix} \varepsilon_n \\ \eta_n \end{bmatrix} + \begin{bmatrix} O(\Delta^2) \\ \vdots \end{bmatrix}$, which simplifies to $\begin{bmatrix} \varepsilon_{n+1} \\ \eta_{n+1} \end{bmatrix} = \begin{bmatrix} O(\Delta^2) \\ \vdots \end{bmatrix}$.

2. Quadrature

(1) Consider quadrature

$$I_{quad} = \sum_{i=0}^{n} \alpha_i f(x_i), \quad x_i \in [-1, 1] \tag{0.1}$$

for the integral

$$I = \int_{-1}^{1} f(x) w(x) dx,$$

where w is a positive weight in $(-1, 1)$. Let

$$\Omega_{n+1}(x) = \prod_{i=1}^{n} (x - x_i)$$

denote the polynomial of degree $n+1$ associated with the (distinct) quadrature nodes x_0, x_1,...,x_n. Prove that

$$\int_{-1}^{1} \Omega_{n+1}(x) p(x) w(x) dx = 0 \tag{0.2}$$

for any polynomial of degree less or equal to $m-1$ if and only if the quadrature formula is exact for all polynomials of degree less or equal $n+m$.

Proof:

(1) If the quadrature formula (0.1) is exact for all polynomials of degree less or equal $n+m$, then

$$\int_{-1}^{1} \Omega_{n+1}(x) p(x) w(x) dx = \sum_{i=0}^{n} \alpha_i \Omega_{n+1}(x_i) p(x_i) w(x_i) = 0.$$

Let us consider polynomial $f(x)$ of degree less or equal to $n+m$. We can write

$$f(x) = \Omega_{n+1}(x) \pi_{m-1}(x) + q_n(x),$$

where q_n is the remainder of the division of f by the polynomial Ω_{n+1}. We have using (0.2)

$$\begin{aligned} I &= \int_{-1}^{1} f(x) w(x) dx \\ &= \int_{-1}^{1} \Omega_{n+1}(x) \pi_{m-1}(x) w(x) dx + \int_{-1}^{1} q_n(x) w(x) dx \\ &= \int_{-1}^{1} q_n(x) w(x) dx \end{aligned}$$

and, by the direct evaluation,

$$\begin{aligned} I_{quad} &= \sum_{i=0}^{n} \alpha_i f(x_i) \\ &= \sum_{i=0}^{n} \alpha_i \Omega_{n+1}(x_i) p(x_i) w(x_i) + \sum_{i=0}^{n} \alpha_i q_n(x_i) \\ &= \sum_{i=0}^{n} \alpha_i q_n(x_i). \end{aligned}$$

We obtain the result by observing that the quadrature weights α_i can always be chosen to satisfy

$$I = I_{quad}$$

for an arbitrary polynomial of degree less or equal to n.

3. Interpolation / Approximation

Assuming that $\varphi_n,\ n=0,1,2,\ldots$ form a set of orthogonal polynomials of degrees *n* over some interval [*a*,*b*] with weight function *w*(*x*) > 0, show that they obey a three-term recursion relation of the form

$$\varphi_{n+1}(x)-(a_n x+b_n)\varphi_n(x)+c_n\varphi_{n-1}(x)=0,\quad n=1,2,3,\ldots$$

where the coefficients a_n, b_n, c_n do not depend on *x*.

Solution:

Given the polynomials $\varphi_n(x)$, we can determine a_n, b_n, c_n such that we eliminate the terms of degrees $n+1, n, n-1$ appearing in $\varphi_{n+1}(x)$. Instead of a zero in the right hand side (RHS), one might expect it to become a polynomial of degree *n*-2, which we can write as $d_{n-2}\varphi_{n-2}+d_{n-3}\varphi_{n-3}+\ldots+d_1\varphi_1+d_0\varphi_0$. It remains to show that all these *d*-coefficients vanish. Thus, form the scalar product of the LHS with $\varphi_k(x)$, with $0\le k\le n-2$. We want to show that each of the four terms in the LHS then vanish.

$(\varphi_{n+1},\varphi_k)=0$ due to orthogonality

$(x\varphi_n,\varphi_k)$: Move across the "x"; then rewrite $x\varphi_k(x)$ as an expansion in $\varphi_{k+1},\varphi_k,\ldots,\varphi_0$. The polynomial $\varphi_n(x)$ is orthogonal to all these, since $k\le n-2$.

(φ_n,φ_k) and $(\varphi_{n-1},\varphi_k)$ both vanish, again since $k\le n-2$.

Hence, all the coefficients d_k vanish, and therefore so does any possible non-zero RHS of the recursion relation.

4. Linear Algebra

Let $A \in \mathbb{C}^{n\times n}$ be a symmetric complex valued matrix, $A = A^t$. It is possible to show that one can find vectors u and nonnegative numbers μ solving the so-called con-eigenvalue problem,

$$A\overline{u} = \mu u.$$

Show that vectors u form an orthonormal basis and there exists a unitary matrix $U \in \mathbb{C}^{n\times n}$ and a real nonnegative matrix

$$\Sigma = \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_n)$$

such that

$$A = U\Sigma U^t.$$

Proof:

If $A = A^t$, then $\overline{A} = A^*$and $A\overline{A} = AA^*$ and $\overline{A}A = A^*A$. Since we have

$$A\overline{u} = \mu u,$$

we also have

$$\overline{A}u = \mu\overline{u}$$

as well as

$$A\overline{A}u = \mu A\overline{u} = \mu^2 u.$$
$$\overline{A}A\overline{u} = \mu\overline{A}u = \mu^2\overline{u}.$$

We recognize that u and $\overline{u}$ are the singular vectors and, thus, they are orthonormal. Defining $\Sigma = \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_n)$ and the unitary matrix $U = (u_1, u_2, \ldots u_n)$, we have

$$A\overline{U} = U\Sigma$$

or

$$A = U\Sigma\overline{U}^* = U\Sigma U^t.$$

5. ODE

Consider the 4th order Adams-Bashforth scheme (AB4) for solving the ODE $y' = f(x, y)$:

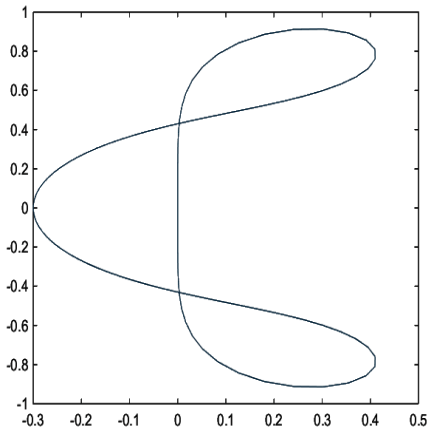
$$y_{n+1} = y_n + \frac{h}{24}(55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}) \ .$$

a. Apply the *root condition* to this scheme. Explain the outcome of the test, and explain what information this provides regarding the scheme.

b. The Matlab code

```
r  = exp(complex(0,linspace(0,2*pi)));
xi = 24*(r.^4-r.^3)./(55*r.^3-59*r.^2+37*r-9);
plot(xi);
```

generates the figure shown to the right.



i. Derive the relation used in the code.

ii. Explain (no need to do the algebra) how you would go from the figure to obtain the *stability domain* for the AB4 scheme.

iii. Explain what information a stability domain conveys.

Solution:

a. To apply the root condition, we apply the scheme in question to the ODE $y' = 0$, which in this case gives the linear recursion relation $y_{n+1} - y_n = 0$, with characteristic equation $r - 1 = 0$. The root condition tells that a scheme is *stable* when all roots are on or inside or on the periphery of the unit circle, and the ones on the unit circle have multiplicity one. With only one root $r = 1$ in the present case, the root condition is obviously satisfied.

The key information this provides is that, if furthermore consistency is satisfied (which is the case here, since the scheme has accuracy 1st order or better), numerical solutions to $y' = f(x, y)$ will converge to the true ones when the step size $h \to 0$.

b.

i. Stability domains are obtained when applying a scheme to the ODE $y' = \lambda y$. We obtain here $y_{n+1} = y_n + \frac{h\lambda}{24}(55 y_n - 59 y_{n-1} + 37 y_{n-2} - 9 y_{n-3})$. Write $h\lambda = \xi$, and note that this is a linear recursion relation with characteristic equation $r^4 = r^3 + \frac{\xi}{24}(55r - 59r^2 + 37r - 9)$. Solving this relation for ξ gives the relation used in the Matlab code.

ii. The generated curve marks all possible ξ-values for when a root *r* is on the periphery of the unit circle. If ξ crosses a curve segment, a root moves between the inside and the outside of the unit circle. The stability domain (in the complex ξ-plane) is characterized by no root falling outside the unit circle. In order to check when this is the case, one needs to select a test point ξ inside each of the enclosed regions, and for this ξ-value check the four roots *r*. Doing so, one will find that the enclosed region just left of the origin satisfies all *r*-roots being inside the unit circle, but none of the other regions do. For example, crossing the boundary in the upper or lower right loops corresponds to a root *r* moving across the unit circle while another root remains outside it.

iii. For ODEs of the form $y' = \lambda y$, the stability domain tells the values of *h* for which there are no growing numerical solutions. (Not part of the present question: If an ODE system is obtained from Method-of-Lines discretization of a time dependent PDE, it provides information on the step sizes that can be used in time vs. space).

6. PDE

Consider the Poisson's equation

$$(\partial_{xx} + \partial_{yy})\, u = f\,(x,y) \quad (x,y) \in B = [0,1] \times [0,1]$$

with the Dirichlet boundary condition

$$u\,\big|_{(x,y)\in\partial B} = 0.$$

Set f to be

$$\begin{aligned} f\,(x,y) &= -4\pi^2 \left[\cos(2\pi x) - 4\cos(4\pi x)\right]\left[\cos(2\pi y) - \cos(4\pi y)\right] \\ &+ -4\pi^2 \left[\cos(2\pi y) - 4\cos(4\pi y)\right]\left[\cos(2\pi x) - \cos(4\pi x)\right] \end{aligned}$$

yielding the solution

$$u\,(x,y) = \left(\cos(2\pi x) - \cos(4\pi x)\right)\left(\cos(2\pi y) - \cos(4\pi y)\right).$$

At the first glance it may appear that seeking solution as a sine series,

$$u\,(x,y) = \sum_{m,n=1}^{\infty} u_{mn} \sin(\pi m x) \sin(\pi n y)$$

should be an efficient approach. However, it turns out that the sine series converges rather slowly.

(1) Can you figure out why the convergence of the sine series is fairly slow?
(2) What are other bases one can use to achieve high accuracy? Suggest a basis that would be more efficient in this case.
(3) Sketch a numerical scheme to compute the solution with high accuracy.

Proof:

(1) The even derivatives of the sine series,

$$\partial_x^{2k}\partial_y^{2l}\left[\sum_{m,n=1}^{\infty} u_{mn}\sin(\pi mx)\sin(\pi ny)\right]$$

is also a sine series,

$$\left[\sum_{m,n=1}^{\infty}(-\pi m)^k(-\pi n)^l u_{mn}\sin(\pi mx)\sin(\pi ny)\right]$$

and, thus, is zero on the boundary. However, it is easy to see that this is not the case for the solution $u(x,y)$. This mismatch of properties requires a large number of terms to approximate the solution near the boundary.

(2) One can use orthogonal polynomials, e.g. Legendre polynomials, to approximate the solution,

$$u(x,y)=\sum_{m,n=0}^{\infty} u_{mn}P_m(2x-1)P(2y-1).$$

This series converges rapidly since the boundary conditions for the even derivatives are not forced on the solution.

(3) We can truncate the Legendre series so that

$$u_N(x,y)=\sum_{m,n=0}^{N} u_{mn}P_m(2x-1)P(2y-1),$$

is an accurate approximation of the solution. We use the Legendre series for the right hand side and discretize the equation using the Gauss-Legendre nodes (appropriate for the polynomials of degree N) and set up the numerical scheme using such nodes in both x and y directions.