Classical and Quasi-Newton methods on the numerical solution of a Boundary Value Problem which rises in the prediction of meteorological parameters using finite differences

This work is co-funded by the European Union (European Social Fund) and Greek national resources under the framework of the "Archimedes III: Funding of Research Groups in TEI of Athens" project of the "Education & Lifelong Learning" Operational Programme.

Framework of the talk:

1. The Physical Problem and Information Geometry
2. The Numerical Solution using Finite Differences
3. Using Newton's method with a LU modification.
4. Quasi Newton's Methods
5. Numerical Tests and Observations

Environmental Parameter Forecasting

Need for high quality environmental predictions-simulations due to important **applications**:

Climate change, Renewable energy production,Transportation,Marine pollution,Ship safety

Two are the main approaches today:

1. Use of in site or remote sensing **observations** (e.g. satellite).
2. Use of numerical **predictions models** governing the atmospheric and wave evolution solved numerically.

Weather and wave forecasting models are successful in simulating general environmental conditions on global or intermediate scale but not on local conditions due to

1. the strong dependence on the initial and lateral conditions,
2. the inability to capture sub-scale phenomena,
3. the parametrization of certain atmospheric or wave procedures.

To overcome this drawback someone can

1. **increase the model resolution**,
2. **improve the initial conditions** based on assimilation systems,
3. **filter-optimize the outputs** of the model using statistical models (MOS methods, Neural networks, Kalman filters).

In all previous options a **"cost function"** measuring the **bias ("the distance")** of the model should be minimized.

When the distance/cost-function is measured by means of classical Euclidean Geometry tools is it correctly estimated?

The role of Information Geometry (IG)

- **IG** is a relatively new branch of Mathematics which applies methods and techniques of non-Euclidean geometry to stochastic processes.
- Given two probability distributions or two data sets we can define a notion of **distance** between them.
- In Euclidean/flat geometry functions are based on least square methods.
- **IG** shows that this assumption is false, in general, and provides a theoretical recipe to avoid such simplifications.
- **IG** naturally introduces geometrical entities (Riemannian metrics, distances, curvature and affine connections) for samilies of probability distributions (manifolds).

Home Page

Title Page

Page 7 of 45

Go Back

Full Screen

Close

Quit

The **minimum distance** between two elements f_1 *and* f_2 of a statistical manifold S is defined by the corresponding ***geodesic*** ω which is the minimum length curve that connects them. Such a curve

$$\omega = (\omega_i) \quad : \quad \Re \to S \tag{1}$$

satisfies the following system of **2^{nd} order differential equations**:

$$\omega_i^{''}(t) + \sum_{j,k=1}^{n} \Gamma_{jk}^{i}(t)\, \omega_j^{'}(t)\, \omega_k^{'}(t) = 0, \quad i = 1,\ 2, \ldots,\ n. \tag{2}$$

under the conditions $\omega(0) = f_1, \ \omega(1) = f_2$.

Home Page

Title Page

Page 8 of 45

Go Back

Full Screen

Close

Quit

The two parameter Weibull distributions have been proved a good choice for fitting wind and wave data.
These distributions form a 2-dimensional statistical manifold with $\xi{=}[\alpha{,}\beta]$, $\Xi = \{[\alpha{,}\beta];\ \alpha$ and $\beta{>}0\}$ (where α is the shape and β the scale parameter) and

$$p\left(x\right) = \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} \mathrm{e}^{-\left(\frac{x}{\beta}\right)^{\alpha}}, \quad \alpha,\ \beta > 0. \tag{3}$$

Home Page

Title Page

Page 9 of 45

Go Back

Full Screen

Close

Quit

Let us have $\xi_0=[\alpha_0,\beta_0]$, $\xi_1=[\alpha_1,\beta_1]$ two members of the Weibull statistical manifold, then the previous system becomes:

$$\begin{aligned}
\omega_1^{''}(t) + \frac{6\left(\gamma\alpha_0 - \alpha_0 - \frac{\pi^2}{6}\right)}{\pi^2\beta_0}\left(\omega_1^{'}(t)\right)^2 + \frac{12\left(\gamma^2 - 2\gamma + \frac{\pi^2}{6} + 1\right)}{\pi^2\alpha_0}\omega_1^{'}(t)\,\omega_2^{'}(t) & - \\
\frac{6(1-\gamma)\beta_0\left(\gamma^2 - 2\gamma + \frac{\pi^2}{6} + 1\right)}{\pi^2 a^3}\left(\omega_2^{'}(t)\right)^2 & = 0 \\
\omega_2^{''}(t) - \frac{\alpha_0^3}{\pi^2\beta_0^2}\left(\omega_1^{'}(t)\right)^2 + \frac{12\alpha_0(1-\gamma)}{\pi^2\beta_0}\omega_1^{'}(t)\,\omega_2^{'}(t) & - \\
\frac{6\left(\gamma^2 - 2\gamma + \frac{\pi^2}{6} + 1\right)}{\pi^2\alpha_0}\left(\omega_2^{'}(t)\right)^2 & = 0
\end{aligned}$$

under the conditions $\omega(0) = \begin{bmatrix}\alpha_0\\ \beta_0\end{bmatrix}, \quad \omega(1) = \begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix}$

where $\omega(t) = \begin{bmatrix}\omega_1(t)\\ \omega_2(t)\end{bmatrix}$ and is $\gamma =$ the Euler gamma.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας

So, we need to study the numerical solution of the following system of differential equations

$$\begin{aligned} \omega_1'' + a_{11}(\omega_1')^2 + a_{12}\omega_1'\omega_2' + a_{22}(\omega_2')^2 &= 0 \\ \omega_2'' + b_{11}(\omega_1')^2 + b_{12}\omega_1'\omega_2' + b_{22}(\omega_2')^2 &= 0 \end{aligned} \tag{4}$$

under the conditions

$$\omega_1(0) = \omega_1^0, \quad \omega_2(0) = \omega_2^0, \quad \omega_1(1) = \omega_1^{N+1}, \quad \omega_2(1) = \omega_2^{N+1}.$$

This is a **second order Boundary Value Problem** of a form

$$\widetilde{\omega}'' = F(\widetilde{\omega}, \widetilde{\omega}') \text{ where } \widetilde{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \text{ defined on the interval } [0, 1].$$

Home Page

Title Page

Page 11 of 45

Go Back

Full Screen

Close

Quit

Finite Differences approach

We divide $[0,1]$ into $N+1$ equal subintervals with endpoints $t_i = 0 + ih$, for $i = 0, 1, \dots, N, N+1$.
If the exact solution has a bounded fourth derivative we can discretize and replace $\omega_1''(t_i)$,$\omega_2''(t_i)$,$\omega_1'(t_i)$,$\omega_2'(t_i)$ by the finite differences :

$$\begin{aligned}
\omega_1''(t_i) &= \frac{\omega_1(t_{i+1}) - 2\omega_1(t_i) + \omega_1(t_{i-1})}{h^2} - \frac{h^2}{12}\omega_2^{(4)}(\xi_i) \\
\omega_2''(t_i) &= \frac{\omega_2(t_{i+1}) - 2\omega_2(t_i) + \omega_2(t_{i-1})}{h^2} - \frac{h^2}{12}\omega_2^{(4)}(\xi_i) \\
\omega_1'(t_i) &= \frac{\omega_1(t_{i+1}) - \omega_1(t_{i-1})}{2h} - \frac{h^2}{6}\omega_1^{(3)}(\eta_i) \\
\omega_2'(t_i) &= \frac{\omega_2(t_{i+1}) - \omega_2(t_{i-1})}{2h} - \frac{h^2}{6}\omega_2^{(3)}(\eta_i)
\end{aligned}$$

for some ξ_i, η_i in the interval (t_{i-1}, t_{i+1}).

The numerical finite differences method results when we substitute the above to the differential equation the error terms are deleted and the boundary conditions are employed:

$$\omega_1\left(0\right)=\omega_1^0,\quad \omega_2\left(0\right)=\omega_2^0,\quad \omega_1\left(1\right)=\omega_1^{N+1},\quad \omega_2\left(1\right)=\omega_2^{N+1}.$$

and we approximate $\omega_1^i \approx \omega_1(t_i), \omega_2^i \approx \omega_2(t_i)$ for $i = 1, \ldots, N$.

The outcome is a nonlinear system of $2N$ equations with $2N$ unknowns of the form

$$\widehat{F}(\widehat{\omega}) = \mathbf{0}$$

where $\mathbf{0} = \left[0, \ldots, 0\right]^T$ and $\widehat{\omega} = \left[\omega_1^1, \ldots, \omega_1^N, \omega_2^1, \ldots, \omega_2^N\right]^T$.

which we solve using Newton's method.

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - J^{-1}(\widehat{\omega}^{(k-1)})\widehat{F}(\widehat{\omega}^{(k-1)}) \quad k = 1, 2, \ldots$$

where $\widehat{\omega}^{(0)}$ is given.

Direct computation of the inverse of the $2N \times 2N$ Jacobian matrix J and multiplication with $\widehat{F}(\widehat{\omega}^{(k-1)})$ is not suggested

We transform our problem

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - J^{-1}(\widehat{\omega}^{(k-1)})\widehat{F}(\widehat{\omega}^{(k-1)}) \Leftrightarrow$$

$$\widehat{\omega}^{(k-1)} - \widehat{\omega}^{(k)} = J^{-1}(\widehat{\omega}^{(k-1)})\widehat{F}(\widehat{\omega}^{(k-1)}) \Leftrightarrow$$

$$J(\widehat{\omega}^{(k-1)})(\widehat{\omega}^{(k-1)} - \widehat{\omega}^{(k)}) = \widehat{F}(\widehat{\omega}^{(k-1)}) \Leftrightarrow$$

$$J(\widehat{\omega}^{(k-1)})X = \widehat{F}(\widehat{\omega}^{(k-1)})$$

where

$$X = (\widehat{\omega}^{(k-1)} - \widehat{\omega}^{(k)})$$

In each step we solve

$$J(\widehat{\omega}^{(k-1)}) \cdot X = \widehat{F}(\widehat{\omega}^{(k-1)})$$

factorize

$$J(\widehat{\omega}^{(k-1)}) = L \cdot U$$

so solve

$$L \cdot U \cdot X = \widehat{F}(\widehat{\omega}^{(k-1)})$$

Let $Y = U \cdot X$

solve the lower triangular system $L \cdot Y = \widehat{F}(\widehat{\omega}^{(k-1)})$,

solve the upper triangular system $U \cdot X = Y$,

update the solution

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - X$$

The classical approach is to apply the common LU factorization approach to the Jacobian matrix J in each step or take advantage of its specific form.

Home Page

Title Page

Page 15 of 45

Go Back

Full Screen

Close

Quit

The Jacobian of the specific problem

is a block matrix of the form:

$$J = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

where A, B, C, D are tridiagonal matrices.

We apply row interchanges and bring J to the following form:

$$\begin{pmatrix}
-8 & 4-a_1 & 0 & 0 & \dots & 0 & 0 & 0 & -b_1 & 0 & 0 & \dots & 0 & 0 \\
4+d_2 & -8 & 4-a_2 & 0 & \dots & 0 & 0 & b_2 & 0 & -b_2 & 0 & \dots & 0 & 0 \\
c_2 & 0 & -c_2 & 0 & \dots & 0 & 0 & 4+d_2 & -8 & 4-d_2 & 0 & \dots & 0 & 0 \\
0 & 4+a_3 & -8 & 4-a_3 & \dots & 0 & 0 & 0 & b_3 & 0 & -b_3 & \dots & 0 & 0 \\
0 & -c_1 & 0 & 0 & \dots & 0 & 0 & -8 & 4-d_1 & 0 & 0 & \dots & 0 & 0 \\
0 & c_3 & 0 & -c_3 & \dots & 0 & 0 & 0 & 4+d_3 & -8 & 4-d_3 & \dots & 0 & 0 \\
0 & 0 & 4+a_4 & -8 & \dots & 0 & 0 & 0 & 0 & b_4 & 0 & \dots & 0 & 0 \\
0 & 0 & c_4 & 0 & \dots & 0 & 0 & 0 & 0 & 4+d_4 & -8 & \dots & 0 & 0 \\
0 & 0 & 0 & 4+a_5 & \dots & 0 & 0 & 0 & 0 & 0 & b_5 & \dots & 0 & 0 \\
0 & 0 & 0 & c_5 & \dots & 0 & 0 & 0 & 0 & 0 & 4+d_5 & \dots & 0 & 0 \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . \\
. & . & . & . & . & . & . & . & . & . & . & . & . & . \\
0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & \dots & . & . \\
0 & 0 & 0 & 0 & \dots & 4+a_n & -8 & 0 & 0 & 0 & \dots & 0 & b_N & 0 \\
0 & 0 & 0 & 0 & \dots & c_n & 0 & 0 & 0 & 0 & \dots & 0 & 4+d_N & -8
\end{pmatrix}$$

Now we can apply a Modified LU factorization

1st Step: Zero only two elements under the main diagonal.
2nd-3rd row: Update only 5 elements in every row (the 2nd, 3rd, (N+1)-th, (N+2)-th, (N+3)-th)
Right here we have a significant reduction of floating point operations as the classical LU updates the entries of an $(2N-1) \times (2N-1)$ submatrix.

2nd Step: Zero only 4 elements under the main diagonal.
3nd-6th row: Update only 6 elements in every row (the 3rd,4th, $N+1, \ldots, N+4$).

3rd Step: Zero only 5 elements under the main diagonal.
4rd-8th row: Update only 6 elements in every row (the 4th,5th, $N+1, \ldots, N+6$).

... and so on

In every step

- The number of elements which must be zeroed is increased per 1 until the $N-2$-th step.
- The number of elements in every row which must be updated is increased per 1 until the (N-4)-th step.
- Then these numbers are decreased per 1 in every step.

Computational Complexity

Floating point operations for triangularizing the $2N \times 2N$ Jacobian matrix through

- modified gaussian elimination are $O(\frac{2N^3}{3})$.
- classical LU factorization requires $O(\frac{8N^3}{3})$.

Modified LU is 4 times cheaper than the classical LU.

Remark

- Reduction is achieved in the first half of the factorization (update specific entries and not whole submatrices).
- Second half of the procedure requires the same cost as classical LU.

Numerical Justification

We compare the two LU approaches for matrices which have the form of the Jacobian J and random elements. We average the computational time needed for sets of 50 matrices.

matrix dim.	classical LU	Modified LU	% of gain
200	0.0176	0.0096	45.6654
500	0.3188	0.0753	76.3882
1000	3.8151	0.7283	80.9093
2000	35.2260	8.7771	75.0834

Newton's Method

$$\widehat{F}:\mathbb{R}^{2N}\to\mathbb{R}^{2N}:\widehat{F}(\widehat{\omega})=\mathbf{0}$$

the system of the $2N$ non linear equations.

$$\widehat{\omega}^{(k)}=\widehat{\omega}^{(k-1)}-J(\widehat{\omega}^{(k-1)})^{-1}\cdot\widehat{F}(\widehat{\omega}^{(k-1)}),k=1,2,\ldots.$$

Complexity

$O(k_0\cdot\frac{8N^3}{3})$ flops for the classical LU factorization

$O(k_0\cdot\frac{2N^3}{3})$ flops for modified LU approacch

for k_0 iterations.

Brezinski's Work

- C. Brezinski, Projection Methods for Systems of Equations (North-Holland, Amsterdam, 1997)
- C. Brezinski, A classification of quasi-Newton methods (Numer Algor, 33, 1997, 123-135)

classified and proposed theoretically, Quasi Newton methods.

We implement numerically four of them.

Since the Jacobian matrix J of our system is of a special form, we adapt these methods to J in order to reduce the required floating point operations.

Quasi Newton Methods

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)}), k = 1, 2, \ldots .$$

where $\Lambda_k \in \mathbb{R}^{2N \times 2N}$.

Brezinski studied the cases where

- Λ_k is a scalar matrix
- Λ_k is a diagonal matrix
- Λ_k is a full matrix

Scalar matrix case (SMC)

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$$

$$\Lambda_k = \lambda_k \cdot I$$

$$\lambda_k = \frac{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), \widehat{F}(\widehat{\omega}^{(k)}))}{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}))}$$

Complexity

For our problem demands $O(k_1 \cdot 18N)$ flops for k_1 iterations plus the computation of $\widehat{F}$ and J at the point $\widehat{\omega}^{(k)}$ at every iteration.

In general, SMC requires $O(k_1 \cdot (4N^2))$ flops for solving an $2N \times 2N$ system of non linear equations.

The reduction in complexity due to the special structure of the jacobian matrix J.

Home Page

Title Page

Page 24 of 45

Go Back

Full Screen

Close

Quit

Diagonal Matrix Case 1 (DMC1)

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$$

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \widetilde{F}(\widehat{\omega}^{(k-1)}) \cdot \widetilde{\Lambda}_{k-1}$$

$$\widetilde{F}(\widehat{\omega}^{(k)}) = diag(\widehat{F}_1(\widehat{\omega}^{(k)}), \widehat{F}_2(\widehat{\omega}^{(k)}), \ldots, \widehat{F}_{2N}(\widehat{\omega}^{(k)}))$$

$$\widetilde{\Lambda}_k = (\lambda_k^1, \lambda_k^2, \ldots, \lambda_k^{2N})^T = J(\widehat{\omega}^{(k-1)})^{-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$$

(Newton's method with a diagonal preconditioner)

Complexity

We use the modified LU factorization in order to compute $J(\widehat{\omega}^{(k-1)})^{-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$ reducing significant the required flops.

$O(k_2 \cdot \frac{2N^3}{3})$ flops for k_2 iterations plus the computation of $\widehat{F}$ and J at the point $\widehat{\omega}^{(k)}$ at every iteration.

In general, $O(k_2 \cdot (\frac{8N^3}{3}))$ flops for solving an $2N \times 2N$ system.

Home Page

Title Page

Page 25 of 45

Go Back

Full Screen

Close

Quit

Diagonal Matrix Case 2 (DMC2)

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$$

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \widetilde{F}(\widehat{\omega}^{(k-1)}) \cdot \widetilde{\Lambda}_{k-1}$$

$$\widetilde{F}(\widehat{\omega}^{(k)}) = diag(\widehat{F}_1(\widehat{\omega}^{(k)}), \widehat{F}_2(\widehat{\omega}^{(k)}), \ldots, \widehat{F}_{2N}(\widehat{\omega}^{(k)}))$$

$\widetilde{\Lambda}_k$ is computed using forward differences, thus,
$\widetilde{\Lambda}_k = \Delta\widetilde{F}(\widehat{\omega}^{(k-1)})^{-1} \cdot \Delta\widehat{\omega}^{(k-1)}$

Complexity

DMC2 demands $O(k_2 \cdot 8N)$ flops for k_2 iterations plus the computation of $\widehat{F}$ and J at the point $\widehat{\omega}^{(k)}$ at every iteration.

Full Matrix Case (FMC)

$$\widehat{\omega}^{(k)} = \widehat{\omega}^{(k-1)} - \Lambda_{k-1} \cdot \widehat{F}(\widehat{\omega}^{(k-1)})$$

$$\Lambda_k = \frac{\widehat{F}(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)})^T \cdot J(\widehat{\omega}^{(k)})^T}{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)})}$$

Complexity

The FMC algorithm demands $O(k_3 \cdot 9N^2)$ flops at every iteration for k_3 iterations plus the computation of $\widehat{F}$ and J at the point $\widehat{\omega}^{(k)}$ at every iteration.

In general, FMC requires $O(k_3 \cdot (13N^2))$ flops for solving an $2N \times 2N$ system of non linear equations.

24 Test Problems based on Weibull distribution

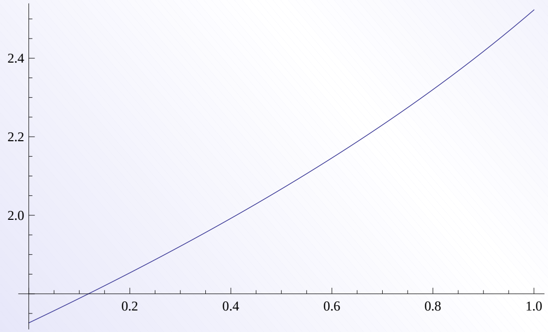
	model data no current		model data with current		satelite data	
Weibull Parameters	shape α_0	scale β_0	shape α_0	scale β_0	shape α_1	scale β_1
Jan	1.600	1.010	1.726	1.095	2.523	1.441
Feb	1.500	1.400	1.571	1.464	2.450	1.762
Mar	1.462	1.132	1.578	1.225	2.560	1.509
Apr	1.564	0.695	1.719	0.754	2.140	1.012
May	1.533	0.608	1.608	0.661	1.576	0.780
Jun	2.333	0.633	2.542	0.680	3.759	0.759
Jul	2.557	0.837	2.688	0.876	3.515	0.960
Aug	3.099	0.716	3.341	0.759	4.938	0.889
Sep	2.418	0.754	2.580	0.800	3.491	0.968
Oct	1.629	0.551	1.850	0.609	2.204	0.665
Nov	1.446	0.892	1.499	0.919	1.911	1.224
Dec	1.435	1.216	1.512	1.283	2.208	1.442

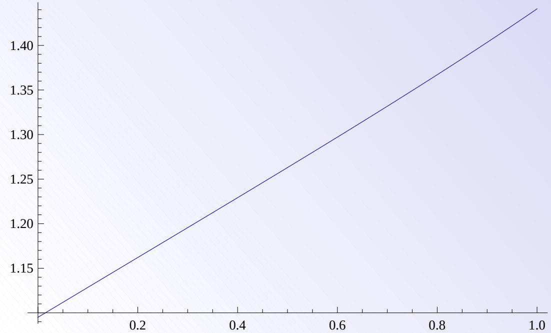
Refer to the area of Levantive area in South Egypt Mediterranean Sea.

Reference Solutions using Mathematica

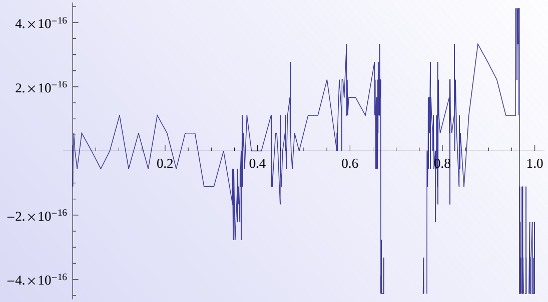
- NDSolve of Mathematica has been used to solve the 24 test problems.
- The Mathematica uses Shooting method and we have set accuracy options (Working Precision, Accuracy Goal, Accuracy Goal) so to get an considerably accurate solution.
- The Mathematica produces a "continuous" interpolating form of the solution.
- The resulted solution has been substituted in the test differential equations and for an abscissae on $[0, 1]$ of width 10^{-5} while the maximum residual has been recorded.
- Such solutions can be used as high accurate reference solutions for the comparison to the other numerical methods which attain a significantly lower precision.

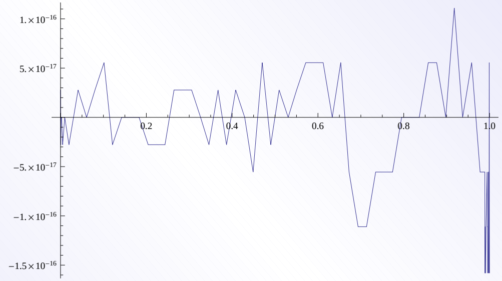
the reference solution of problem Jun with current



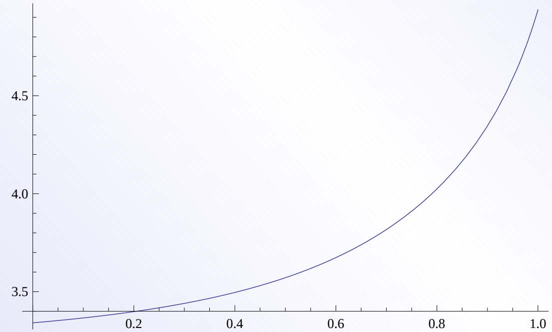


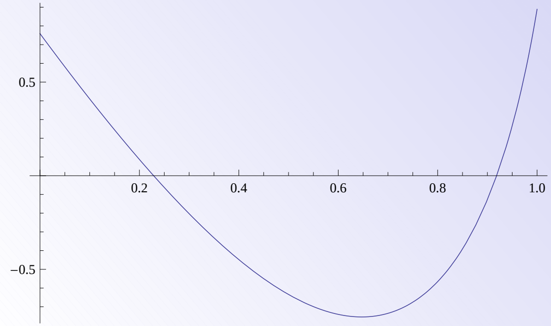
the error 1.11×10^{-15}



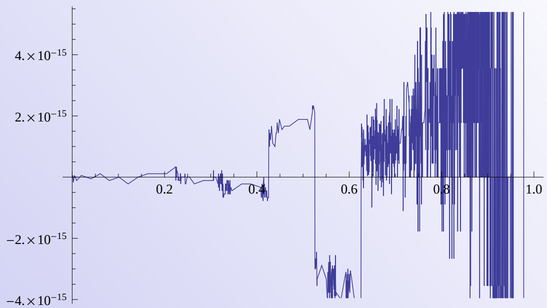


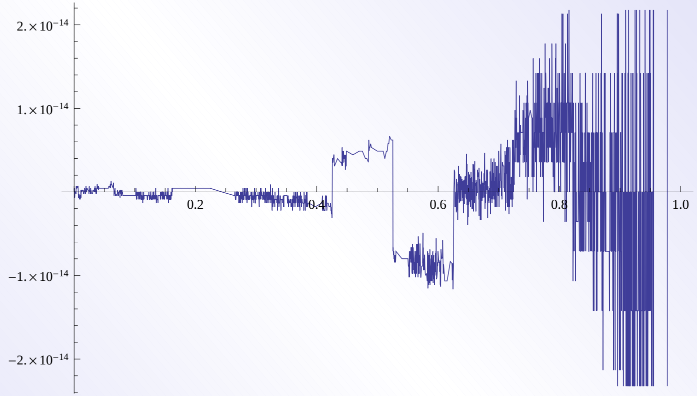
the reference solution of Aug with current (stiffness)





the error 6.25×10^{-13}





Numerical tests

- We calculate and program the analytical form of the of F and Jacobian J. We choose $N = 100$ so we have a system of 200 equations.
- For the 24 problems we produce a reference solution.
- For an initial guess we use a perturbation with random numbers of the initial conditions on $t = 0$.

Numerical tests

- We solve numerically the 24 test problems for tolerances $10^{-3}, 10^{-4}, \ldots, 10^{-13}$ to compare efficiency and computational cost. We use two error measures at the grid points.
  - The first one is $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$ the maximum absolute value that the numerical solution fails to satisfy the nonlinear problem resulted by the finite difference method.
  - The second one is the $\|\widehat{\omega}_{so} - \widehat{\omega}_{ref}\|_\infty$ maximum absolute value of the difference of the numerical solution and the reference solution.
- We investigate the sensitivity in the choice of initial guess for tolerances $10^{-3}, 10^{-4}, \ldots, 10^{-13}$ with respect to its distance from the reference solution, in order to evaluate the range of convergence for each method.

Accuracy of the Newton's Method

Both NR with classical LU and NR with modified LU have the same iterations and similar error measures at the grid points for all 24 problems.

	Average							
	no of iter.		time in secs		$\lVert\widehat{F}(\widehat{\omega}_{sol})\rVert_\infty$		$\lVert\widehat{\omega}_{so} - \widehat{\omega}_{ref}\rVert_\infty$	
TOL	clas.	mod.	clas.	mod.	clas.	mod.	clas.	mod.
10^{-8}	8.33	8.04	0.1529	0.099	0.355e-10	0.176e-10	0.605e-5	0.605e-5
10^{-10}	8.7	8.67	0.1731	0.1196	0.080e-12	0.173e-12	0.123e-4	0.123e-4

In some problems the iteration diverges for both approaches (e.g. Problem Aug with Current).

Time comparisons.

We compare the average of the average times time to solve each of the 24 problem for 50 different choices of initial conditions using NR with classical LU and NR with modified LU.

	Average		
TOL	clas.	mod.	% of gain
10^{-8}	0.199	0.135	32.70
10^{-9}	0.209	0.143	31.53
10^{-10}	0.217	0.147	32.05
10^{-11}	0.224	0.151	32.57
10^{-12}	0.362	0.169	40.44
10^{-13}	0.952	0.088	56.41

Home Page

Title Page

Page 35 of 45

Go Back

Full Screen

Close

Quit

General remarks for the comparison of mod-NR, SMC,DMC1,DMC2,FMC.

We solve for $TOL = 10^{-3}, \ldots, 10^{-12}$ all the 24 problems for a common initial condition each time and compare the average values of the results.

- DMC1 does not work at all. The preconditioning matrix is singular.
- DMC2 works only for $TOL = 10^{-3}, 10^{-4}$. For smaller tolerances it fails as in $\widetilde{\Lambda}_k = \Delta\widetilde{F}(\widehat{\omega}^{(k-1)})^{-1} \cdot \Delta\widehat{\omega}^{(k-1)}$ the denominator becomes less than eps.
- SMC and works only for $TOL = 10^{-3}, \ldots, 10^{-8}$. For smaller tolerances even if the methods do not seem to diverge the iteration stops as the denominator of

$$\frac{\#}{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)})}$$

becomes less than eps.

Mean # of Iterations comparisons.

for problems with convergence

	average		
TOL	mod-NR	SMC	FMC
10^{-3}	7.4	485	95
10^{-4}	7.6	3460	835
10^{-5}	8.4	6853	2959
10^{-6}	8.6	10057	7363
10^{-7}	9.2	13429	10596
10^{-8}	9.6	16296	13632

Mean time in secs comparisons.

for problems which converge.

	average		
TOL	mod-NR	SMC	FMC
10^{-3}	0.098	0.166	0.443
10^{-4}	0.098	4.062	1.070
10^{-5}	0.112	14.684	8.063
10^{-6}	0.117	35.450	24.140
10^{-7}	0.126	56.277	44.580
10^{-8}	0.123	85.410	71.087

No mater the theoretical complexity cost,
the time needed for the solution using modified Newton's method is considerably smaller.
SMC takes the longer time over the three methods.

Mean $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$ **comparisons.**
for problems which converge.

	Average		
TOL	mod-NR	SMC	FMC
10^{-3}	1.55e-6	1.83e-3	9.92e-3
10^{-4}	6.77e-8	1.45e-4	1.09e-3
10^{-5}	7.29e-9	1.53e-5	1.13e-4
10^{-6}	1.93e-9	1.48e-6	1.11e-5
10^{-7}	8.24e-11	1.62e-7	1.19e-6
10^{-8}	2.82e-11	1.71e-8	1.20e-7

Newton's method attains a better convergence.

Mean $\|\widehat{\omega}_{so} - \widehat{\omega}_{ref}\|_\infty$ comparisons.

for problems which converge.

	Average		
TOL	mod-NR	SMC	FMC
10^{-3}	2.87e-6	2.53e-1	4.17e-1
10^{-4}	1.59e-5	3.77e-2	2.05e-1
10^{-5}	3.49e-6	3.96e-3	2.96e-2
10^{-6}	1.56e-5	3.94e-4	2.88e-3
10^{-7}	2.53e-5	6.61e-5	3.29e-4
10^{-8}	3.57e-6	7.14e-6	3.45e-5

Newton's method goes closer to the reference solution for the bigger tolerances

mod-NR Sensitivity in initial condition choice.

Number of convergent solution of problems (out of 24).

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	24	24	21	3
10^{-4}	24	24	22	1
10^{-5}	24	24	23	2
10^{-6}	24	24	21	2
10^{-7}	24	24	23	2
10^{-8}	24	24	21	1
10^{-9}	24	24	21	1
10^{-10}	24	24	19	2
10^{-11}	24	24	23	3
10^{-12}	24	24	23	3
10^{-13}	24	22	19	3

Very sensitive in the choice of initial guess. Shorter interval of convergence.

Home Page

Title Page

Page 41 of 45

Go Back

Full Screen

Close

Quit

SMC Sensitivity in initial condition choice.

Number of convergent solution of problems (out of 24).

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	24	24	24	24
10^{-4}	24	24	24	24
10^{-5}	24	24	24	24
10^{-6}	24	24	24	24
10^{-7}	24	24	24	24
10^{-8}	24	24	24	24
10^{-9}	12	13	13	14

Longer interval of convergence. Method does not diverge for $TOL = 10^{-9}$.

SMC average $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$ comparisons

for all 24 problems.

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	1.78e-3	1.61e-3	1.32e-3	9.40e-3
10^{-4}	1.00e-4	1.00e-4	1.08e-4	1.11e-4
10^{-5}	1.10e-5	1.10e-5	1.10e-5	1.11e-5
10^{-6}	1.20e-6	1.19e-6	1.22e-6	1.21e-6
10^{-7}	1.27e-7	1.07e-7	1.27e-6	1.26e-7
10^{-8}	1.40e-8	1.40e-8	1.41e-8	1.34e-8
10^{-9}	1.32e-8	1.20e-8	1.30e-8	1.25e-8

The iteration stops as the denominator of

$$\frac{\#}{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}))}$$

becomes less than eps.

FMC Sensitivity in initial condition choice.

Number of solution of convergent problems (out of 24).

	$\lVert\widehat{\omega}_0 - \widehat{\omega}_{ref}\rVert_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	24	24	24	24
10^{-4}	24	24	24	24
10^{-5}	24	24	24	24
10^{-6}	24	24	24	24
10^{-7}	24	24	24	24
10^{-8}	24	24	24	24
10^{-8}	15	14	10	14

Long interval of convergence again. Method does not diverge for $TOL = 10^{-9}$.

FMC average $\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$ comparisons

for all 24 problems.

	$\|\widehat{\omega}_0 - \widehat{\omega}_{ref}\|_\infty \leq$			
TOL	0.05	0.10	0.2	0.5
10^{-3}	7.14e-3	7.80e-3	8.16e-3	8.27e-3
10^{-4}	8.46e-4	8.31e-4	8.46e-4	8.78e-4
10^{-5}	8.99e-5	9.49e-5	9.62e-5	9.51e-5
10^{-6}	1.02e-5	1.03e-5	1.04e-5	1.04e-5
10^{-7}	1.08e-6	1.07e-6	1.09e-6	1.08e-6
10^{-8}	1.09e-7	1.11e-7	1.56e-7	1.15e-7
10^{-9}	1.30e-8	1.34e-8	1.42e-8	1.31e-8

The iteration stops as the denominator of

$$\frac{\#}{(J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}), J(\widehat{\omega}^{(k)}) \cdot \widehat{F}(\widehat{\omega}^{(k)}))}$$

becomes less than eps.

SMC and FMC can be used as starting procedures.

For problem Aug with current mod NR diverges. We can use either SMC or FMC to get an initial guess for NR and then solve with NR.

$TOL = 10^{-11}$			
	no of iter.	$\|\widehat{F}(\widehat{\omega}_{sol})\|_\infty$	$\|\widehat{\omega}_{so} - \widehat{\omega}_{ref}\|_\infty$
mod-NR	182	4.38e+177	9.03e+88
SMC	9219	1.75e-6	7.51e-4
mod-NR	7	4.78e-14	3.74e-4
FMC	5602	1.39e-5	3.40e-3
mod-NR	7	5.59e-14	3.74e-4