About Newton-Krylov methods

Jocelyne Erhel

INRIA, UR Rennes - Campus de Beaulieu, 35042 Rennes Cedex, France
http://www.irisa.fr/aladin/perso/erhel.html

1. Introduction

The resolution of Partial Differential Equations involves first a space discretisation by Finite Element Methods for example which leads to an ordinary differential equation such that

$$\begin{cases} \frac{du}{dt} + g(u) = 0 \quad u \in \mathbb{R}^N \\ u(t=0) = u_0 \end{cases}$$

where g is a nonlinear function from $\mathbb{R}^N$ to $\mathbb{R}^N$ and N is the number of space discretisation points. If an implicit scheme is used to solve this differential equation, each time set then requires the resolution of a nonlinear problem of dimension N of the type $u_{n+1} + \Delta t g_1(u_{n+1}) - \Delta t g_2(u_n) = 0$ where Δt is the time step. Let F be a sufficiently smooth nonlinear function from $\mathbb{R}^N$ to $\mathbb{R}^N$. In the example above, $F(u) = u + \Delta t g_1(u) - \Delta t g_2(u_n)$. This paper focuses on the resolution of nonlinear problems

$$F(u) = 0. \tag{0.1}$$

In the sequel, we assume that (0.1) has a unique solution u_*. The Newton method is an efficient iterative method to solve (0.1) provided that it starts with a good approximation of u_*. Indeed, under correct assumptions on F, the local convergence is quadratic hence extremely fast. Newton method is often combined with techniques such as backtracking to ensure global convergence (with any initial guess), but convergence is then only linear or superlinear [DS83, OR70]. Although Newton method requires only a few iterations, it is quite costly because each iteration means to compute a Jacobian matrix $J(u_k)$ of order N, to store it and to solve a linear system of order N. If a direct method is used, it implies to store also the LU decomposition. In practice, the Jacobian is freezed during several iterations and refreshed when convergence slows down, in order to reduce the cost of LU factorisations.

Another way to reduce both storage and CPU time is to apply an iterative linear solver. Such inexact Newton methods loose the local quadratic convergence but each iteration is less costly [EW94, Bro87]. Indeed, it requires to store only one matrix and to use the matrix only through matrix vector products. The CPU cost comes mainly from these products which serve as a basis to measure the complexity of the method. Newton-Krylov methods for example use linear solvers based on projections onto Krylov subspaces. If the Jacobian is symmetric and positive definite, Conjugate Gradient (CG) is the best method ; If it is only symmetric, then Lanczos-based methods are commonly used; the general case with a nonsymmetric Jacobian is the most difficult, where the different methods such as GMRES, BICGSTAB, TFQMR, and so on, have all merits and drawbacks. Recent surveys on iterative linear solvers describe these Krylov methods, see for example [BBC+93, Bru95, FGN91, Saa95]. Because such inexact Newton methods require only matrix vector products, it is possible to get free of any matrix storage by approximating the product $J(u)v$ with a finite difference scheme. These methods are called matrix-free Newton variants [BH86]. The cost to approximate $J(u)v$ is then one or a few evaluations of $F(u + \sigma v)$ with σ correctly chosen. Although these Newton-Krylov methods are attractive, they converge sometimes too slowly compared to exact Newton methods. If the Jacobian $J(u)$ is ill-conditioned then the linear solver will converge slowly or give a poor solution which will affect the convergence in Newton iterations. In the context of matrix-free variants, a bad approximation of $J(u)v$ will also slown down the convergence.

To get a useful and robust nonlinear solver, it is necessary to speed-up the convergence. In the context of matrix-free techniques, the use of an accurate scheme to approximate $J(u)v$ improves noticably the convergence [CE96]. A general idea is to reuse the information collected during the resolution of a linear solver when solving the next systems. In other words, the idea is to reuse the Krylov subspace generated at one step. However this idea is limited by storage considerations ; it is not possible to store all Krylov basis of all linear systems encountered. Hence different strategies must be developed to get a practical realisation of this idea. This paper will focus on several researchs conducted recently in this field. The first case will be the successive resolution of linear systems with the same matrix $Ax_k = b_k$. This happens when the Jacobian is freezed in Newton iterations. A particular case, which happens in other nonlinear resolutions, is of the type $(A + \sigma_k I)x_k = b$ with the same right-hand side [WB95, SW96]. The general case is the successive resolution of systems with slightly varying matrix and right-hand side $A_k x_k = b_k$. The acceleration can be linear by augmenting the Krylov subspace [Saa87, Mor95, CS95, Guy96] or by preconditioning the linear system [EBP96, Cho95, BCGR96] or non linear by solving easier non linear problems between Newton iterations [Cho95, AC94, FSvdV95, KA95].

Section 2 describes briefly Newton-Krylov methods, Krylov linear solvers and gives some results on matrix-free implementations. Section 3 deals with acceleration schemes for a serie of symmetric linear systems, based on Lanczos algorithm. Section 4 is similar but for nonsymmetric systems and is based on GMRES algorithm. Section 5 gives some new results about nonlinear acceleration schemes based also on Krylov subspaces.

2. Newton-Krylov methods

Below is the general form of a Newton-Krylov algorithm to solve (0.1) :

ALGORITHM 1: Newton-Krylov

choose u_0 ; choose α ;
for $k = 0, 1, \dots$ **until** $\|F(u_k)\| \leq \alpha \|F(u_0)\|$
  approximately solve $J(u_k)x_k = -F(u_k)$
  to get $J(u_k)x_k = -F(u_k) - r_k$ with $\|r_k\| < \|F(u_k)\|$;
  $u_{k+1} = u_k + \mu x_k$ with $\|F(u_{k+1}\| < \|F(u)\|$;
endfor

The condition $\|r_k\| < \|F(u_k)\|$ ensures that the approximate solution x_k is a descent direction, or in other words that

$$F(u_k)^T J(u_k) x_k < 0.$$

It implies, under correct assumptions, the existence of μ which can be computed by a backtracking algorithm. This Newton version guarantees a global convergence but only linear. Thus an important point is that the iterative linear solver must converge with a sufficiently small residual.

2.1. Krylov linear solvers

We focus now on Krylov linear solvers based on Lanczos and Arnoldi algorithms. Let $K_m(v, A) = span\{v, Av, \dots, A^{m-1}v\}$ the Krylov subspace generated by the vector v. Let us assume that K_m is of dimension m. Krylov linear solvers start with an initial guess x_0 and compute approximate solutions $x_m = x_0 + V_m\ y$ with various requirements for the residual $r_m = b - Ax_m = r_0 - AV_m y$, where V_m is a basis of the Krylov subspace $K_m(r_0, A)$.

Both Lanczos and Arnoldi methods construct an orthonormal basis V_m of $K_m(r_0, A)$. If A is symmetric, the Lanczos basis and the residual satisfy

$$AV_m = V_m T_m + s_m e_m^T, \quad \|r_m\| = \|V_m(\beta e_1 - T_m y) - s_m e_m^T y\| \text{ with } \beta = \|r_0\| \quad (0.2)$$

where T_m is a symmetric tridiagonal matrix. If A is unsymmetric, the Arnoldi basis and the residual satisfy

$$AV_m = V_{m+1}\overline{H}_m, \quad \|r_m\| = \|\beta e_1 - \overline{H}_m y\| \text{ with } \beta = \|r_0\| \quad (0.3)$$

where $\overline{H}_m$ is a upper Hessenberg matrix.

If A is symmetric definite positive, the Conjugate Gradient algorithm (CG), equivalent to Lanczos in that case, ensures the Galerkin condition

$$V_m^T r_m = 0 \quad (0.4)$$

and also the minimisation property

$$\|r_m\| = \min_{y \in \mathbb{R}^m} \|r_0 - AV_m y\| \quad (0.5)$$

If A is symmetric undefinite, two variants based on Lanczos, require either (0.4) or (0.5). They are called respectively SYMMLQ and MINRES. Clearly, they are equivalent to CG if A is definite positive.

If A is unsymmetric, the method which require (0.5), called GMRES (Generalized Minimum Residual), uses the Arnoldi method. It is equivalent to MINRES if A is symmetric.

There exist many other methods for unsymmetric matrices, for example biCGstab, QMR, hybrid methods, and so on. The rest of the paper focuses only on CG, SYMMLQ, MINRES and GMRES.

2.2. Matrix-free techniques

Lanczos-based and Arnoldi-based Krylov methods check the convergence with the quantity $\|\beta e_1 - T_m y\|$or $\|\beta e_1 - \overline{H}_m y\|$, using the relation (0.2) or (0.3). But a matrix-free technique involves errors in the computation of AV_m so that these relations are no longer true and the test of convergence is not valid if the error in AV_m is too large.

Usually the Jacobian is approximated by a first-order scheme

$$J(u)v \approx \frac{F(u+\sigma v) - F(u)}{\sigma} \text{ with } \sigma = \frac{\sqrt{\epsilon} u^t.v}{\|v\|_2}$$

so that the error, including rounding errors, is in $O(\sqrt{\epsilon})$. In [CE96], numerical experiments show that this first-order scheme is not accurate enough. The test of convergence in GMRES fails so that the Newton method converges slowly. On the other hand, if a second-order scheme defined by

$$J(u)v \approx \frac{F(u+\sigma v) - F(u-\sigma v)}{2\sigma} \text{ with } \sigma = \sqrt[3]{\epsilon}\frac{(u^t v)}{\|v\|_2},$$

the error is in $O(\epsilon^{2/3})$ and the accuracy gets better. Numerical experiments in [CE96] show a clear improvement over the first-order scheme.

3. Symmetric Lanczos acceleration

In this section, the Jacobian is assumed to be symmetric. In many Newton applications, the Jacobian matrix is stored and freezed during several iterations so that the method leads to a sequence of linear systems of the form $Ax_k = b_k$.

3.1. A particular case : identical right-hand sides

In some instances, for example in GCV applications [WB95], the method leads to a sequence of the form $(A + \sigma_k I)x_k = b$ with the same right-hand side and with a symmetric positive definite matrix A. In this case, all linear systems will be projected onto the same Krylov subspace so that it is sufficient to compute one Krylov subspace with an orthonormal basis V. Indeed, since $V^TAV = T$ and $V^TV = I$ then

$$V^T(A + \sigma_k I)V = T + \sigma_k I$$

so that each new system is simply a new tridiagonal system with the diagonal updated. In practice, to guarantee convergence, the size of V can vary from one system to another. The main drawback of this approach is the memory cost to store V. However, in the context of GCV applications, it appears sufficient to compute only dot products which can be expressed by three-term recurrences [SW96]. This new fast-GCV method using Lanczos is thus a very powerful method with a low memory cost.

3.2. Initial guess

In the general case, the right-hand side varies but the hope is that it will vary slowly. Therefore, it is natural to extend the previous method. The first key idea is to start with an initial guess computed with the previous Krylov basis.

Let $Ax^{(1)} = b^{(1)}$ and $Ax = b$ two successive linear systems to solve. Let V_m be the orthonormal basis of the Krylov subspace $K_m(r_0^{(1)}, A)$. If x_0 is a given guess for the second system (usually $x_0 = 0$, the idea is to choose as an initial guess (to build the Krylov subspace) the vector $x_1 = x_0 + V_m y$ such that $V_m^T r_1 = 0$. So y must satisfy

$$T_m y = V_m^T r_0 \text{ so that } \|r_1\| = \|(I - V_m V_m^T) r_0\| + \alpha$$

with α usually small.

3.3. Augmented subspace approach

However, the residual $\|r_1\|$ is usually not accurate enough, so it is necessary to restart a Krylov process. If the second system is solved using SYMMLQ with r_1 as initial guess, then the second resolution is equivalent to a block Krylov method so that convergence is improved [Saa87]. It seems also natural to still use the Krylov subspace K_m in the second resolution. The approximate solution is then no longer searched in a Krylov subspace but in an augmented subspace $K_{m,p} = span\{V_m, r_1, Ar_1, \ldots, A^{p-1} r_1\}$ with a basis $W_{m,p}$ composed of the previous Krylov basis V_m and the Krylov basis W_p of $K_p(r_1, A)$. Starting with w_1 orthogonal to V_m, the modified Lanczos-SYMMLQ algorithm constructs w_j orthogonal to v_m, so that $W_{m,p}$ is an orthonormal basis. The modified Lanczos-SYMMLQ algorithms leads then to

$$W_{m,p}^T A W_{m,p} = T_{m,p} \text{ with } T_{m,p} = \begin{pmatrix} T_1 & B^T \\ B & T_2 \end{pmatrix}$$

where T_1 and T_2 are tridiagonal and $B = (0, e)$ with $e \in \mathbb{R}^p$. The modified Krylov method computes an approximation $x_p = x_1 + W_{m,p} y$ such that

$$T_{m,p} y = W_{m,p}^T b.$$

The main drawback is the storage of the vectors V_m. However, it must be noted that only the last one is required during the algorithm whereas the others serve only to compute the solution x_m at the very end. Therefore it is reasonable to store V_m in a secondary storage memory.

Here the test of convergence is slightly more complicated but can still be expressed by means of recurrences [Guy96]. Preliminary numerical experiments on small matrices (order 64) show an improvement in CPU time of about 20% over the usual Lanczos method.

3.4. *Definite case*

Similar ideas are developed in [Rey94, Rou94] for example which apply to CG algorithm for definite symmetric matrices. The algorithm selects an initial guess with the same technique and the augmented approach is defined by an orthogonality condition on the new conjugate direction : it is orthogonal not only to the previous one but also to the last one from the first system. Moreover, conjugate directions are fully orthogonalized against previous one to reduce loss of orthogonality. This algorithm solves an interface problem issued from a domain decomposition so that memory cost and CPU cost are quite low and the global method is very efficient.

4. Linear acceleration in GMRES

Now A is assumed to be nonsymmetric and GMRES is the linear solver. Let $Ax^{(1)} = b^{(1)}$ and $Ax = b$ two successive linear systems to solve. Let V_m be the orthonormal basis of the Krylov subspace $K_m(r_0^{(1)}, A)$. Recall that

$$AV_m = V_{m+1}\overline{H}_m \text{ and } \overline{H}_m = \overline{Q}_m \begin{pmatrix} R_m \\ 0 \end{pmatrix} \text{ where } \overline{Q}_m = \begin{pmatrix} Q_m & q_{m+1} \end{pmatrix}$$

and Q_m is an orthogonal matrix.

4.1. *Krylov preconditioner*

The idea here is to build a preconditioner using the Krylov subspace K_m and the factorisation above. The objective is to get an exact inverse of A onto the subspace $span(V_{m+1}Q_m)$. The Krylov preconditioner [Cho95] is defined by

$$M^{-1} = V_{m+1}\left\{ \begin{pmatrix} \alpha R_m^{-1} & 0 \\ 0 & 1 \end{pmatrix} \overline{Q}_m^T - I_{m+1} \right\} V_{m+1}^T + I_N$$

where α is a scaling factor. The subspace $span(V_{m+1}Q_m)$ is an invariant subspace of the preconditioned matrix AM^{-1} (with the multiple eigenvalue α).

In practice, the matrix A varies from one Newton iteration to another. However the matrix M^{-1} is still a good candidate to precondition the new Jacobian.

Numerical experiments were conducted to solve Navier-Stokes equations for compressible flows, using an implicit time discretisation. Two strategies have been tested : use the preconditioner inside one time step through Newton iterations or use it across time steps. In these experiments [Cho95], the second strategy was much better. The implicit scheme was faster than the explicit one and the preconditioner achieved an improvement of about 15% over the unpreconditioned method.

4.2. *Deflation preconditioner*

The previous method requires the storage of the basis V_{m+1} and the use of V_{m+1} at each inner iteration of GMRES so that its use is limited by memory considerations. This drawback can be circumvented by applying the same technique with a subspace of K_m.

The choice of the subspace is guided by two remarks : first, the convergence of GMRES is mainly related to the eigenvalues of A and second, the Arnoldi process embedded into GMRES gives for almost free approximations of eigenvalues and eigenvectors of A (the Ritz values and Ritz vectors). The deflation preconditioner [EBP96, BCGR96] relies on these properties. Let U_k an approximate invariant subspace corresponding to the k smallest eigenvalues of A. The deflation preconditioner [EBP96] is defined by

$$M^{-1} = U_k(R_k^{-1} - I_k)U_k^T + I_N, \quad R_k = U_k^T A U_k.$$

If U_k were an invariant subspace, then the projection of AM^{-1} onto $span(U_k)$ would be the identity.

The same idea can be used in an augmented subspace framework as for Lanczos. Here the second Krylov subspace $span(W_m)$ is augmented with U_k, a set of approximated eigenvectors [Mor95, CS95]. A modified Arnoldi process is used to get

$$A(W_m, U_k) = V\overline{H} \text{ so that } x = x_0 + (W_m, U_k)y \text{ and } r = r_0 - V\overline{H}y = V(\|r_0\|e_1 - \overline{H}y)$$

with V an orthonormal set of $m + 1 + k$ vectors.

There are various ways to compute approximate eigenvectors. An harmonic projection involving a small generalized eigenvalue problem appears to be more accurate than a classical oblique projection of the form $H_m u = \lambda u, U = V_m u$, [Mor95, CS95, EBar] ; an Implicitely Restarted Arnoldi (IRA) scheme is also promising [BCGR96]. It appears necessary to update U_k at each restart using the information of the last cycle in order to get more and more accurate approximate eigenvectors [Mor95, EBar]. With this strategy, the deflation preconditioner is much more robust than the unpreconditioned restarted GMRES [EBar].

5. Nonlinear acceleration in Newton-Krylov methods

Up to now, acceleration have been applied to linear systems arising at each Newton iteration. New schemes have been recently developed [AC94, Cho95, FSvdV95, KA95] where each Newton iteration is modified to speed up convergence. The basic idea is still to project a problem onto a Krylov subspace, but now the nonlinear problem is projected instead of the linear system. Let $span(V_k)$ a search subspace expanded at each Newton iteration. The Newton iterate is defined by $u_{k+1} = u_k + V_k y$ [FSvdV95] where y_k is one solution of the reduced nonlinear problem

$$\min_{y\in\mathbb{R}^k} \|F(u_k + V_k y)\| \text{ or } W_k^T F(u_k + V_k y) = 0 \text{ or } \min_{y\in\mathbb{R}^k} W_k^T \|F(u_k + V_k y)\| \qquad (0.6)$$

which are respectively a least squares problem, a Galerkin problem and a Restricted Minimum Residual (RMR) problem, where W_k is a shadow space chosen for example as $W_k = V_k$.

In [BS90, Cho95], the search basis V_k is taken to the Krylov basis $\tilde{V}_m$ generated by GMRES when solving $J(u_k)x_k = -F(u_k)$. In [BS90], the problem (0.6) is approximately solved and the solution is injected into a Dogleg strategy to guarantee global convergence. In [Cho95], the problem (0.6) is solved by a Gauss-Newton method. To avoid storage and computational time, Gauss-Newton is modified by keeping the Jacobian constant equal to $J(u_k)$. This amounts then to several iterations of the approximation described in [BS90]. Numerical experiments in [Cho95] show the efficacy of this nonlinear acceleration when it is combined with the Krylov preconditioner.

In [AC94, KA95], the search basis V_k is a combination of Krylov vectors from $\tilde{V_m}$ and previous search directions x_l from previous Newton iterations. Here too, an approximation of Gauss-Newton is used to solve (0.6).

In [FSvdV95], the search basis V_k is composed of previous search directions and the new search direction x_k ; the reduced Restricted Minimum Residual (RMR) problem is solved by Gauss-Newton.

Numerical experiments in [FSvdV95] compare the approaches from [BS90, AC94, KA95] with RMR which seems quite powerful.

6. Concluding remarks

Nonlinear problems arise in many numerical applications and will be one of the most challenging problems for scientific computing in the next years. A very promising approach is to combine Newton method with a Krylov subspace method. The underlying key idea behind several emerging algorithms is to retain information from previous Newton iterations and to reuse it in order to speed-up convergence of the current iteration. This idea was first applied to a sequence of linear systems. It was implemented either as an augmented subspace or as a preconditioning technique. It also appeared efficient to select a few vectors in the Krylov subspace using eigenvalue and eigenvector approximations. Future work will concern the design of a truly adaptive method which would automate the deflation process. Nonlinear Krylov solvers go one step further. They use the Krylov subspace in the nonlinear iteration itself. Preliminary results show their efficacy. Future work will concern the design of such nonlinear solvers and numerous numerical experiments.

REFERENCES

[AC94] Axelsson O. and Chronopoulos A. (1994) On nonlinear generalized conjugate gradient methods. *Numerische Mathematik* 69: 1–15.

[BBC+93] Barret R., Berry M., Chan T., Demmel J., Donato J., Dongarra J., Eijkhout V., Pozo R., Romine C., and van der Vorst H. (1993) *Templates for the solution of linear systems: building blocks for iterative methods.* SIAM / netlib.

[BCGR96] Baglami J., Calvetti D., Golub G., and Reichel L. (1996) Adaptively preconditioned gmres algorithms. Technical report, Kent State University.

[BH86] Brown P. and Hindmarsh A. (June 1986) Matrix-free methods for stiff systems of ode's. *SIAM J. Numer. Anal.* 23(3): 610–638.

[Bro87] Brown P. (1987) A local convergence theory for combined inexact-newton/finite difference projection methods. *SIAM J.Numer.Anal* 24: 407–434.

[Bru95] Bruaset A. (1995) *A survey of preconditioned iterative methods.* Pitman Research Notes in Mathematics Series. Longman Scientific and Technical.

[BS90] Brown P. N. and Saad Y. (Mai 1990) Hybrid krylov methods for nonlinear systems of equations. *SIAM J.Sci.Stat.Comput* 11(3): 450–481.

[CE96] Choquet R. and Erhel J. (1996) Newton-GMRES algorithm applied to compressible flows. *International Journal for numerical methods in fluids* 23: 177–190.

[Cho95] Choquet R. (December 1995) *Etude de la methode de Newton-GMRES - Application aux equations de Navier-Stokes compressibles*. Thèse de doctorat, Université de Rennes 1.

[CS95] Chapman A. and Saad Y. (1995) Deflated and augmented Krylov subspace techniques. Supercomputer Institute Research report UMSI 95/181, University of Minnesota.

[DS83] Dennis J. E. and Schnabel R. B. (1983) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall series in Computational Mathematics.

[EBP96] Erhel J., Burrage K., and Pohl B. (1996) Restarted gmres preconditioned by deflation. *Journal of Computational and Applied Mathematics* 69(303-318).

[EBar] Erhel J. and Burrage K. (to appear) On the performance of various adaptive preconditioned GMRES strategies. Research report, INRIA.

[EW94] Eisenstat S. and Walker H. (May 1994) Globally convergent inexact newton methods. *SIAM J.Optimization* 4(2): 393–422.

[FGN91] Freund R., Golub G., and Nachtigal N. (1991) Iterative solution of linear systems. *Acta Numerica* pages 57–100.

[FSvdV95] Fokkema D., Sleijpen G., and van der Vorst H. (1995) Accelerated inexact Newton schemes for large systems of nonlinear equations. Preprint 918, University of Utrecht.

[Guy96] Guyomarc'h F. (1996) Acce'le'ration des me'thodes de newton-krylov. Rapport de stage de DEA, IRISA.

[KA95] Kaporin I. and Axelsson O. (January 1995) On a class of nonlinear equation solvers based on the residual norm reduction over a sequence of affine subspaces. *Siam.S.Sci.Comput.* 16(1): 228–249.

[Mor95] Morgan R. (1995) A restarted gmres method augmented with eigenvectors. *SIAM J. matrix Anal. App.* 16: 1154–1171.

[OR70] Ortega J. and Rheinboldt W. (1970) *Iterative Solution of Nonlinear Equations in Several Variables*. Computer science and applied mathematics. Academic Press.

[Rey94] Rey C. (1994) *Développement d'algorithmes parallèles de résolution en calcul non-linéaire de structures hétérogènes : Application au cas de la butée acier-élastomère*. Thèse de doctorat, Ecole normale supérieure de Cachan.

[Rou94] Roux F.-X. (1994) Parallel implementation of a domain decomposition method for non-linear elasticity problems. In Keyes D. E., Saad Y., and Truhlar D. (eds) *Domain-based parallelism and problem decomposition methods in computational science and engineering*. SIAM.

[Saa87] Saad Y. (April 1987) On the lanczos method for solving symmetric linear systems with several right-hand sides. *Mathematics of computation* 178: 651–662.

[Saa95] Saad Y. (1995) *Iterative methods for sparse linear systems*. PWS publishing.

[SW96] Sidje R. and Williams A. (1996) Fast generalised cross validation. In *International Linear Algebra Year:Linear Algebra in Optimization*. CERFACS, Albi-Toulouse, France.

[WB95] Williams A. and Burrage K. (1995) Surface fitting using GCV smoothing splines on supercomputers. In *Proceedings of Supercomputing*.