

Método de Newton Inexacto para Sistemas No Lineales de Gran Escala. NITSOL: Código en FORTRAN para estos problemas

Isidro A. Abelló Ugalde

Seminario Semanal del Laboratorio de Cómputo Científico
Posgrado en Matemáticas, Facultad de Ciencias, CU
Octubre 2010

Esquema de la Presentación

- 1 Introducción
 - Definición y supuestos sobre el sistema no lineal
 - Método de Newton
- 2 Método Inexacto de Newton
 - Motivación de los método inexactos
 - Métodos de Backtracking (reducción hacia atrás?)
 - Selección del término de forzamiento
- 3 El Sistema Lineal
 - Solvers basados en Subespacios de Krylov
 - Solvers de NITSOL

Presentación

- 1 **Introducción**
 - **Definición y supuestos sobre el sistema no lineal**
 - Método de Newton
- 2 Método Inexacto de Newton
 - Motivación de los método inexactos
 - Métodos de Backtracking (reducción hacia atrás?)
 - Selección del término de forzamiento
- 3 El Sistema Lineal
 - Solvers basados en Subespacios de Krylov
 - Solvers de NITSOL

Definición y supuestos sobre el sistema no lineal

Problema: Solución de Sistemas de Ecuaciones No Lineales

Modelo Lineal

Resolver $F(x) = 0$, no lineal y continuamente diferenciable.

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, F(x) = \begin{bmatrix} F_1(x) \\ \vdots \\ F_n(x) \end{bmatrix}$$

Modelo lineal: $F(x) \approx F(x_0) + F'(x_0)(x - x_0)$

donde $F'(x) \equiv J(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} F_1(x) & \cdots & \frac{\partial}{\partial x_n} F_1(x) \\ \frac{\partial}{\partial x_1} F_2(x) & \cdots & \frac{\partial}{\partial x_n} F_2(x) \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \frac{\partial}{\partial x_1} F_n(x) & \cdots & \frac{\partial}{\partial x_n} F_n(x) \end{bmatrix}, \in \mathbb{R}^{n \times n}$

Definición y supuestos sobre el sistema no lineal

Problema: Solución de Sistemas de Ecuaciones No Lineales

Modelo Lineal

Resolver $F(x) = 0$, no lineal y continuamente diferenciable.

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, F(x) = \begin{bmatrix} F_1(x) \\ \vdots \\ F_n(x) \end{bmatrix}$$

Modelo lineal: $F(x) \approx F(x_0) + F'(x_0)(x - x_0)$

donde $F'(x) \equiv J(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} F_1(x) & \cdots & \frac{\partial}{\partial x_n} F_1(x) \\ \frac{\partial}{\partial x_1} F_2(x) & \cdots & \frac{\partial}{\partial x_n} F_2(x) \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \frac{\partial}{\partial x_1} F_n(x) & \cdots & \frac{\partial}{\partial x_n} F_n(x) \end{bmatrix} \in \mathbb{R}^{n \times n}$

Definición y supuestos sobre el sistema no lineal

Problema: Solución de Sistemas de Ecuaciones No Lineales

Modelo Lineal

Resolver $F(x) = 0$, no lineal y continuamente diferenciable.

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, F(x) = \begin{bmatrix} F_1(x) \\ \vdots \\ F_n(x) \end{bmatrix}$$

Modelo lineal: $F(x) \approx F(x_0) + F'(x_0)(x - x_0)$

donde $F'(x) \equiv J(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} F_1(x) & \cdots & \frac{\partial}{\partial x_n} F_1(x) \\ \frac{\partial}{\partial x_1} F_2(x) & \cdots & \frac{\partial}{\partial x_n} F_2(x) \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \frac{\partial}{\partial x_1} F_n(x) & \cdots & \frac{\partial}{\partial x_n} F_n(x) \end{bmatrix}, \in \mathbb{R}^{n \times n}$

Propiedades del Jacobiano

Cuando $F(x)$ proviene de PDE

Si el sistema de ecuaciones proviene de la discretización de una ecuación diferencial parcial no lineales, como es el caso de Reservoir Simulation in Oil Recovery, *el Jacobiano*:

- n es grande (cientos de miles)
- No es simétrico
- No es definido positivo (en general)
- Es *sparse*
- Es una matriz de banda.
- Puede ser mal condicionada.
- Frecuentemente **métodos de Krylov** se usan para resolver SL que involucren estas matrices.

Presentación

- 1 **Introducción**
 - Definición y supuestos sobre el sistema no lineal
 - **Método de Newton**
- 2 Método Inexacto de Newton
 - Motivación de los método inexactos
 - Métodos de Backtracking (reducción hacia atrás?)
 - Selección del término de forzamiento
- 3 El Sistema Lineal
 - Solvers basados en Subespacios de Krylov
 - Solvers de NITSOL

Esquema de Iterativo Newton

Ventajas

Dado x inicial

```
while no convergencia
```

```
    Resolver  $J(x)s = -F(x)$ 
```

```
     $x \leftarrow x + s$ 
```

```
endwhile
```

Propiedades

- Localmente cuadráticamente convergente.
- Independencia de la malla, para problemas de discretizaciones de PDE.

(Esto último no se que significa pero es algo bueno para el problema que nos ocupa)

Esquema de Iterativo Newton

Ventajas

Dado x inicial

```
while no convergencia
```

```
    Resolver  $J(x)s = -F(x)$ 
```

```
     $x \leftarrow x + s$ 
```

```
endwhile
```

Propiedades

- Localmente cuadráticamente convergente.
- Independencia de la malla, para problemas de discretizaciones de PDE.

(Esto último no se que significa pero es algo bueno para el problema que nos ocupa)

Esquema de Iterativo Newton

Ventajas

Dado x inicial

```
while no convergencia
```

```
  Resolver  $J(x)s = -F(x)$ 
```

```
   $x \leftarrow x + s$ 
```

```
endwhile
```

Propiedades

- Localmente cuadráticamente convergente.
- Independencia de la malla, para problemas de discretizaciones de PDE.

(Esto último no se que significa pero es algo bueno para el problema que nos ocupa)

Método de Newton Truncado

Dado x inicial

while no convergencia

 Resolver $J(x)s = -F(x)$

$x \leftarrow x + s$

endwhile

Cuando se resuelve *aproximadamente*

$$J(x)s = -F(x)$$

por un método iterativo (como los de subespacios de Krylov)
en los que se realiza algún tipo de minimización del residual:

$$\min_s \|F(x) + J(x)s\|$$

el esquema se conoce como:

Método de Newton Truncado

Presentación

- 1 Introducción
 - Definición y supuestos sobre el sistema no lineal
 - Método de Newton
- 2 Método Inexacto de Newton
 - **Motivación de los método inexactos**
 - Métodos de Backtracking (reducción hacia atrás?)
 - Selección del término de forzamiento
- 3 El Sistema Lineal
 - Solvers basados en Subespacios de Krylov
 - Solvers de NITSOL

Principales preguntas

- ¿Cuándo parar las iteraciones del sistema lineal?
- ¿Cómo hacer que el sistema sea global?
Básicamente deseamos que $\|F(x + s)\| < \|F(x)\|$
- ¿Qué método utilizar para resolver el sistema lineal?

Principales preguntas

- ¿Cuándo parar las iteraciones del sistema lineal?
- ¿Cómo hacer que el sistema sea global?
Básicamente deseamos que $\|F(x + s)\| < \|F(x)\|$
- ¿Qué método utilizar para resolver el sistema lineal?

Principales preguntas

- ¿Cuándo parar las iteraciones del sistema lineal?
- ¿Cómo hacer que el sistema sea global?
Básicamente deseamos que $\|F(x + s)\| < \|F(x)\|$
- ¿Qué método utilizar para resolver el sistema lineal?

Principales preguntas

- ¿Cuándo parar las iteraciones del sistema lineal?
- ¿Cómo hacer que el sistema sea global?
Básicamente deseamos que $\|F(x + s)\| < \|F(x)\|$
- ¿Qué método utilizar para resolver el sistema lineal?

Esquema de Iterativo Inexacto de Newton

Definiciones, características y observaciones

Dado x inicial

while no convergencia

 Encontrar $\eta \in [0; 1)$ y s tal que

$$\|F(x) + J(x)s\| \leq \eta \|F(x)\|$$

$$x \leftarrow x + s$$

endwhile

- El parámetro η es conocido como **término de forzamiento**
- El esquema de Newton está dentro del marco de los métodos inexactos
- Aplicaremos el método iterativo lineal hasta que

$$\|F(x) + J(x)s\| \leq \eta \|F(x)\|$$
- La cuestión de detener las iteraciones del esquema lineal se convierte en el tema de la elección del término de forzamiento
- Un método inexacto es cualquiera que reduzca la norma del modelo lineal con respecto al valor de la norma F .

Esquema de Iterativo Inexacto de Newton

Definiciones, características y observaciones

Dado x inicial

while no convergencia

 Encontrar $\eta \in [0; 1)$ y s tal que

$$\|F(x) + J(x)s\| \leq \eta \|F(x)\|$$

$$x \leftarrow x + s$$

endwhile

- El parámetro η es conocido como **término de forzamiento**
- El esquema de Newton está dentro del marco de los métodos inexactos
- Aplicaremos el método iterativo lineal hasta que

$$\|F(x) + J(x)s\| \leq \eta \|F(x)\|$$
- La cuestión de detener las iteraciones del esquema lineal se convierte en el tema de la elección del término de forzamiento
- Un método inexacto es cualquiera que reduzca la norma del modelo lineal con respecto al valor de la norma F .

Resultados de Convergencia

Observaciones

Teorema (*Dembo, Eisenstat, Steihaug, 1982*)

Suponga que $F(x^*) = 0$ y $J(x^*)$ inversible (invertible?).

Si $\{x_k\}$ es generada por un algoritmo inexacto de Newton con x_0 suficientemente cerca de x^* entonces

- $\eta_k \leq \eta_{max} < 1 \implies x_k \longrightarrow x^*$ *q-linealmente*
- $\eta_k \longrightarrow 0 \implies x_k \longrightarrow x^*$ *q-superlinealmente*

Si además J es Lipschitz continua en x^* entonces

- $\eta_k = O(\|F(x_k)\|) \implies x_k \longrightarrow x^*$ *q-cuadráticamente*

Este teorema es **local**.

Se requiere una estrategia global.

Resultados de Convergencia

Observaciones

Teorema (*Dembo, Eisenstat, Steihaug, 1982*)

Suponga que $F(x^*) = 0$ y $J(x^*)$ inversible (invertible?).

Si $\{x_k\}$ es generada por un algoritmo inexacto de Newton con x_0 suficientemente cerca de x^* entonces

- $\eta_k \leq \eta_{max} < 1 \implies x_k \longrightarrow x^*$ q -linealmente
- $\eta_k \longrightarrow 0 \implies x_k \longrightarrow x^*$ q -superlinealmente

Si además J es Lipschitz continua en x^* entonces

- $\eta_k = O(\|F(x_k)\|) \implies x_k \longrightarrow x^*$ q -cuadráticamente

Este teorema es **local**.

Se requiere una estrategia global.

Estrategias Globales de Convergencia

Vínculo con optimización

- Pedir solamente $\|F(x + s)\| < \|F(x)\|$ no es suficiente.
- Criterios basados en norma de la reducción actual y la reducción predicha por algún modelo (lineal usualmente)
- Conexión con optimización y las estrategias de búsqueda en la línea o región de confianza.

$$F(x) = 0 \iff \min_x f(x)$$

donde

$$f(x) = \frac{1}{2} F(x)^t F(x) = \frac{1}{2} \|F(x)\|_2^2$$

o sea, convertir en un problema de mínimos cuadrados

Estrategias Globales de Convergencia

Vínculo con optimización

- Pedir solamente $\|F(x + s)\| < \|F(x)\|$ no es suficiente.
- Criterios basados en norma de la reducción actual y la reducción predicha por algún modelo (lineal usualmente)
- Conexión con optimización y las estrategias de búsqueda en la línea o región de confianza.

$$F(x) = 0 \iff \min_x f(x)$$

donde

$$f(x) = \frac{1}{2} F(x)^t F(x) = \frac{1}{2} \|F(x)\|_2^2$$

o sea, convertir en un problema de mínimos cuadrados

Estrategias Globales de Convergencia

Vínculo con optimización

- Pedir solamente $\|F(x + s)\| < \|F(x)\|$ no es suficiente.
- Criterios basados en norma de la reducción actual y la reducción predicha por algún modelo (lineal usualmente)
- Conexión con optimización y las estrategias de búsqueda en la línea o región de confianza.

$$F(x) = 0 \iff \min_x f(x)$$

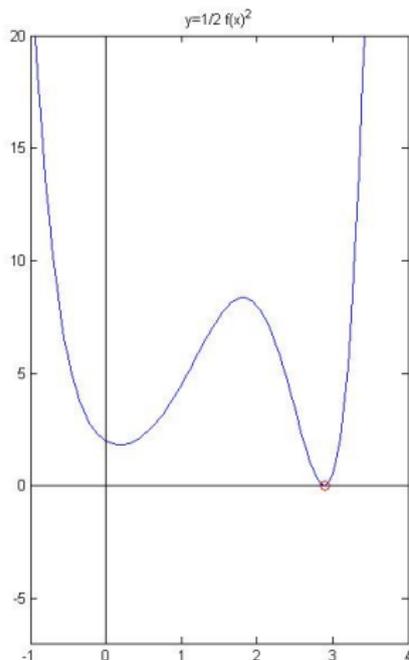
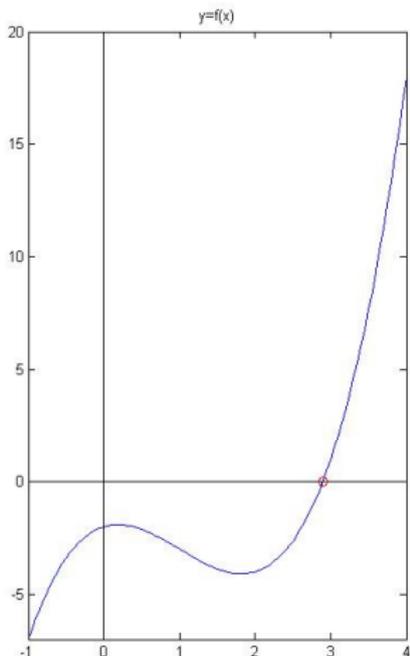
donde

$$f(x) = \frac{1}{2} F(x)^t F(x) = \frac{1}{2} \|F(x)\|_2^2$$

o sea, convertir en un problema de mínimos cuadrados

Motivación de los métodos inexactos

$$F(x) = 0 \text{ contra } \min_x f(x)$$

Ejemplo en \mathbb{R} de la raíz de la eq. y el problema de minimización

Presentación

1

Introducción

- Definición y supuestos sobre el sistema no lineal
- Método de Newton

2

Método Inexacto de Newton

- Motivación de los método inexactos
- **Métodos de Backtracking (reducción hacia atrás?)**
- Selección del término de forzamiento

3

El Sistema Lineal

- Solvers basados en Subespacios de Krylov
- Solvers de NITSOL

Métodos de Backtracking (reducción hacia atrás?)

Backtracking Methods Idea

Esquema iterativo básico: Inexact Newton Backtracking (INB) Method

Escoja s lo más cercano a $(-J(x)^{-1}F(x))$ tal que $\|F(x) + J(x)s\| < \|F(x)\|$
y verifique si

$$\|F(x + s)\| < \|F(x)\|$$

si no, reduzca el paso.

Dados x inicial, $t \in (0; 1)$, $\eta_{max} \in [0; 1)$ y $0 < \theta_{min} < \theta_{max} < 1$

while no convergencia

 Seleccionar $\eta \in [0; \eta_{max})$

 Encontrar s tal que $\|F(x) + J(x)s\| \leq \eta \|F(x)\|$

 Evaluar $F(x + s)$

while $\|F(x + s)\| > (1 - t(1 - \eta)) \|F(x)\|$

 Seleccionar $\theta \in [\theta_{min}; \theta_{max}]$

$s \leftarrow \theta s$

$\eta \leftarrow 1 - \theta(1 - \eta)$

 Reevaluar $F(x + s)$

 endwhile

$x \leftarrow x + s$

$F(x) \leftarrow F(x + s)$

endwhile

Métodos de Backtracking (reducción hacia atrás?)

Backtracking Methods Idea

Esquema iterativo básico: Inexact Newton Backtracking (INB) Method

Escoja s lo más cercano a $(-J(x)^{-1}F(x))$ tal que $\|F(x) + J(x)s\| < \|F(x)\|$
y verifique si

$$\|F(x + s)\| < \|F(x)\|$$

si no, reduzca el paso.

Dados x inicial, $t \in (0; 1)$, $\eta_{max} \in [0; 1)$ y $0 < \theta_{min} < \theta_{max} < 1$

while no convergencia

 Seleccionar $\eta \in [0; \eta_{max})$

 Encontrar s tal que $\|F(x) + J(x)s\| \leq \eta \|F(x)\|$

 Evaluar $F(x + s)$

while $\|F(x + s)\| > (1 - t(1 - \eta)) \|F(x)\|$

 Seleccionar $\theta \in [\theta_{min}; \theta_{max}]$

$s \leftarrow \theta s$

$\eta \leftarrow 1 - \theta(1 - \eta)$

 Reevaluar $F(x + s)$

 endwhile

$x \leftarrow x + s$

$F(x) \leftarrow F(x + s)$

endwhile

Resultado de Convergencia

Posibilidades

Teorema (*Eisenstat y Walker, 1994*)

Suponga que $\{x_k\}$ es producida por el algoritmo INB. Si $\{x_k\}$ tiene un punto de acumulación x^ y $J(x^*)$ es no singular entonces*

$$F(x^*) = 0 \quad y \quad \{x_k\} \longrightarrow x^*$$

Además s_k y η_k son aceptables para todo $k \geq k_M$ suficientemente grande.

- $\|x_k\| \longrightarrow \infty$
- $\{x_k\}$ tiene varios puntos de acumulación y J es singular en cada uno de ellos.
- $\{x_k\}$ converge a x^* , $F(x^*) = 0$, $J(x^*)$ es no singular y la tasa de convergencia la determina la sucesión $\{\eta_k\}$

Resultado de Convergencia

Posibilidades

Teorema (*Eisenstat y Walker, 1994*)

Suponga que $\{x_k\}$ es producida por el algoritmo INB. Si $\{x_k\}$ tiene un punto de acumulación x^ y $J(x^*)$ es no singular entonces*

$$F(x^*) = 0 \quad y \quad \{x_k\} \longrightarrow x^*$$

Además s_k y η_k son aceptables para todo $k \geq k_M$ suficientemente grande.

- $\|x_k\| \longrightarrow \infty$
- $\{x_k\}$ tiene varios puntos de acumulación y J es singular en cada uno de ellos.
- $\{x_k\}$ converge a x^* , $F(x^*) = 0$, $J(x^*)$ es no singular y la tasa de convergencia la determina la sucesión $\{\eta_k\}$

Métodos de Backtracking (reducción hacia atrás?)

Practicalities

Detalles de la implementación

- Escoger η_{max} cerca de 1, por ejemplo $\eta_{max} = 0.9$
- Escoger t pequeño, por ejemplo $t = 10^{-4}$
- Los valores usuales de θ_{min} y θ_{max} son 0.1 y 0.5 respectivamente.
- Utilizar una norma inducida por un producto interno, por ejemplo, $\|\bullet\|_2$
- Escoger $\theta \in [\theta_{min}; \theta_{max}]$ tal que minimice una cuadrática o una cúbica que interpole $\|F(x_k + \theta s_k)\|$

Presentación

- 1 Introducción
 - Definición y supuestos sobre el sistema no lineal
 - Método de Newton
- 2 Método Inexacto de Newton
 - Motivación de los método inexactos
 - Métodos de Backtracking (reducción hacia atrás?)
 - **Selección del término de forzamiento**
- 3 El Sistema Lineal
 - Solvers basados en Subespacios de Krylov
 - Solvers de NITSOL

Selección del término de forzamiento

Selección del término de forzamiento

Peligro de exceso de cálculos

La selección adecuada de los forzamientos es muy importante, ya que corremos el peligro de que haya un divorcio entre la reducción efectiva $\|F(x + s)\|$ y la de su modelo lineal $F(x) + J(x)s$

Propuesta de Eisenstat y Walker, 1996

Seleccionar $\eta_k = \min \{ \eta_{max} ; \tilde{\eta}_k \}$ donde

$$\tilde{\eta}_k = \frac{\| \|F(x_k)\| - \|F(x_{k-1}) + J(x_{k-1})s_{k-1}\| \|}{\|F(x_{k-1})\|}$$

Protección de los forzamiento

$$\eta_k = \max \left\{ \eta_k ; \eta_{k-1}^{(1+\sqrt{5})/2} \right\} \text{ si } \eta_{k-1}^{\frac{1+\sqrt{5}}{2}} > 0.1$$

Selección del término de forzamiento

Peligro de exceso de cálculos

La selección adecuada de los forzamientos es muy importante, ya que corremos el peligro de que haya un divorcio entre la reducción efectiva $\|F(x + s)\|$ y la de su modelo lineal $F(x) + J(x)s$

Propuesta de Eisenstat y Walker, 1996

Seleccionar $\eta_k = \min \{ \eta_{max} ; \tilde{\eta}_k \}$ donde

$$\tilde{\eta}_k = \frac{\| \|F(x_k)\| - \|F(x_{k-1}) + J(x_{k-1})s_{k-1}\| \|}{\|F(x_{k-1})\|}$$

Protección de los forzamiento

$$\eta_k = \max \left\{ \eta_k ; \eta_{k-1}^{(1+\sqrt{5})/2} \right\} \text{ si } \eta_{k-1}^{\frac{1+\sqrt{5}}{2}} > 0.1$$

Elección de los terminos de forzamiento

Ideas y Justificaciones

- La elección anterior de los forzamientos mide la concordancia (o la discordancia :-)) entre F y el su modelo lineal local en la iteración. anterior.

$$\left(\tilde{\eta}_k = \frac{\| \|F(x_k)\| - \|F(x_{k-1}) + J(x_{k-1})s_{k-1}\| \|}{\|F(x_{k-1})\|} \right)$$

- Esta elección corre el riesgo de ser muy pequeña lejos de la solución, lo que produce un sobrecálculo.
- La justificación de la protección es que una disminución drástica de η_k a la iteración siguiente debe estar justificada por varias iteraciones.
- Esta elección de los forzamiento es invariante a la multiplicación de un escalar por F .
- Hay un teorema de convergencia bonito ... pero me lo salto.

Elección de los terminos de forzamiento

Ideas y Justificaciones

- La elección anterior de los forzamientos mide la concordancia (o la discordancia :-)) entre F y el su modelo lineal local en la iteración. anterior.

$$\left(\tilde{\eta}_k = \frac{\| \|F(x_k)\| - \|F(x_{k-1}) + J(x_{k-1})s_{k-1}\| \|}{\|F(x_{k-1})\|} \right)$$

- Esta elección corre el riesgo de ser muy pequeña lejos de la solución, lo que produce un sobrecálculo.
- La justificación de la protección es que una disminución drástica de η_k a la iteración siguiente debe estar justificada por varias iteraciones.
- Esta elección de los forzamiento es invariante a la multiplicación de un escalar por F .
- Hay un teorema de convergencia bonito ... pero me lo salto.

Presentación

- 1 Introducción
 - Definición y supuestos sobre el sistema no lineal
 - Método de Newton
- 2 Método Inexacto de Newton
 - Motivación de los método inexactos
 - Métodos de Backtracking (reducción hacia atrás?)
 - Selección del término de forzamiento
- 3 El Sistema Lineal
 - **Solvers basados en Subespacios de Krylov**
 - Solvers de NITSOL

Métodos de Subespacios de Krylov

Idea General y Propiedades Generales

Resolver $Ax = b$ utilizando el subespacio $\mathcal{K}_k(v)$ con un v adecuado

$$\mathcal{K}_k(v) \equiv \{v, Av, \dots, A^{k-1}v\}$$

e intrínsecamente los procesos de **Lanczos** o de **Arnoldi**

Algunas Propiedades Generales

- Son procesos iterativos.
- Controlan la reducción en cada iteración de $\|r_k\| = \|b - Ax_k\|$
- Fuertes propiedades de convergencia.
- La operación más costosa es el producto `Matriz-vector`.
- Incorporación natural al esquema de preconditionadores.
- No requiere de disponer de la matriz A sino de una función que calcule el producto `Matriz-vector`.
- Ideales para sistemas de gran escala.
- Su naturaleza es paralelizable.

Métodos de Subespacios de Krylov

Idea General y Propiedades Generales

Resolver $Ax = b$ utilizando el subespacio $\mathcal{K}_k(v)$ con un v adecuado

$$\mathcal{K}_k(v) \equiv \{v, Av, \dots, A^{k-1}v\}$$

e intrínsecamente los procesos de **Lanczos** o de **Arnoldi**

Algunas Propiedades Generales

- Son procesos iterativos.
- Controlan la reducción en cada iteración de $\|r_k\| = \|b - Ax_k\|$
- Fuertes propiedades de convergencia.
- La operación más costosa es el producto `Matriz-vector`.
- Incorporación natural al esquema de preconditionadores.
- No requiere de disponer de la matriz A sino de una función que calcule el producto `Matriz-vector`.
- Ideales para sistemas de gran escala.
- Su naturaleza es paralelizable.

Métodos de Subespacios de Krylov

Métodos más importantes

- **PCG** Gradiente Conjugado (Para matrices simétricas y definidas positivas) 1952-1958
(uno de los más fantásticos algoritmos de los últimos 50 años)
- *MINRES* (Para matrices simétricas) 1975
Minimal Residual
- *GMRES* (Para matrices generales) 1986
Generalized Minimal Residual

Métodos de Subespacios de Krylov

Métodos más importantes

- **PCG** Gradiente Conjugado (Para matrices simétricas y definidas positivas) 1952-1958
(uno de los más fantásticos algoritmos de los últimos 50 años)
- **MINRES** (Para matrices simétricas) 1975
Minimal Residual
- **GMRES** (Para matrices generales) 1986
Generalized Minimal Residual

Métodos de Subespacios de Krylov

Métodos más importantes

- **PCG** Gradiente Conjugado (Para matrices simétricas y definidas positivas) 1952-1958
(uno de los más fantásticos algoritmos de los últimos 50 años)
- **MINRES** (Para matrices simétricas) 1975
Minimal Residual
- **GMRES** (Para matrices generales) 1986
Generalized Minimal Residual

Métodos de Subespacios de Krylov

Métodos más importantes

- **PCG** Gradiente Conjugado (Para matrices simétricas y definidas positivas) 1952-1958
(uno de los más fantásticos algoritmos de los últimos 50 años)
- **MINRES** (Para matrices simétricas) 1975
Minimal Residual
- **GMRES** (Para matrices generales) 1986
Generalized Minimal Residual

Presentación

- 1 Introducción
 - Definición y supuestos sobre el sistema no lineal
 - Método de Newton
- 2 Método Inexacto de Newton
 - Motivación de los método inexactos
 - Métodos de Backtracking (reducción hacia atrás?)
 - Selección del término de forzamiento
- 3 El Sistema Lineal
 - Solvers basados en Subespacios de Krylov
 - Solvers de NITSOL

Solvers de NITSOL

Propiedades

La naturaleza de los sistemas lineales involucrados en $F(x) = 0$ es no simétrica por lo que los *solvers* utilizados son de propósito general.

- *Simpler GMRES(m)* 1994
Simpler Generalized Minimal Residual: restart m
- *BiCGSTAB* 1992
Gradiente Biconjugado Estabilizado
- *TFQMR* 1993
Transpose Free Quasi Minimal Residual

Solvers de NITSOL

Propiedades

La naturaleza de los sistemas lineales involucrados en $F(x) = 0$ es no simétrica por lo que los *solvers* utilizados son de propósito general.

- *Simpler GMRES(m)* 1994
Simpler Generalized Minimal Residual: restart m
- *BiCGSTAB* 1992
Gradiente Biconjugado Estabilizado
- *TFQMR* 1993
Transpose Free Quasi Minimal Residual

GMRES

Idea General del Algoritmo: Proceso de ortogonalización de Arnoldi

Partir de x_0 inicial y $x_k = x_0 + z_k$, donde z_k resuelve

$$\min_{z \in \mathcal{K}_k(r_0)} \|b - A(x_0 + z)\|_2 = \min_{z \in \mathcal{K}_k(r_0)} \|r_0 - Az\|_2$$

Dado x_0 , iniciar $r \leftarrow b - Ax_0$; $w_1 \leftarrow \|r\|_2$, $V_1 \leftarrow r / \|r\|_2$, $j \leftarrow 1$, $k \leftarrow 1$;

while ($j \leq n$) **do**

for $i=1:j$ **do**

$$H_{(i,j)} \leftarrow V_j^t * A * V_i;$$

 Resolver $y = H^{-1}w$; (Nota: H es de Hessenberg!!)

$$x \leftarrow x_0 + V * y;$$

$$V_{j+1} \leftarrow A * V_j - V * H_j;$$

$$H_{(j+1,j)} \leftarrow \|V_{j+1}\|_2;$$

$$\|r\|_2 \leftarrow |y_{(1)}| * H_{(j+1,j)};$$

if $\|r\|_2 < \text{tol}$ **then** PARAR;

$$V_{j+1} \leftarrow V_{j+1} / H_{(j+1,j)};$$

$$w_{j+1} \leftarrow 0;$$

$$j \leftarrow j + 1;$$

endwhile

GMRES

Idea General del Algoritmo: Proceso de ortogonalización de Arnoldi

Partir de x_0 inicial y $x_k = x_0 + z_k$, donde z_k resuelve

$$\min_{z \in \mathcal{K}_k(r_0)} \|b - A(x_0 + z)\|_2 = \min_{z \in \mathcal{K}_k(r_0)} \|r_0 - Az\|_2$$

Dado x_0 , iniciar $r \leftarrow b - Ax_0$; $w_1 \leftarrow \|r\|_2$, $V_1 \leftarrow r / \|r\|_2$, $j \leftarrow 1$, $k \leftarrow 1$;

while ($j \leq n$) **do**

for $i=1:j$ **do**

$$H_{(i,j)} \leftarrow V_j^t * A * V_i;$$

 Resolver $y = H^{-1}w$; (Nota: H es de Hessenberg!!)

$$x \leftarrow x_0 + V * y;$$

$$V_{j+1} \leftarrow A * V_j - V * H_j;$$

$$H_{(j+1,j)} \leftarrow \|V_{j+1}\|_2;$$

$$\|r\|_2 \leftarrow |y_{(1)}| * H_{(j+1,j)};$$

if $\|r\|_2 < \text{tol}$ **then PARAR**;

$$V_{j+1} \leftarrow V_{j+1} / H_{(j+1,j)};$$

$$w_{j+1} \leftarrow 0;$$

$$j \leftarrow j + 1;$$

endwhile

GMRES(m): Modificaciones fundamentales

```

Dados  $x_0$ , MaxIter y m, iniciar  $iter \leftarrow 0$ ;
repeat
   $r \leftarrow b - Ax_0$ ;  $w_1 \leftarrow \|r\|_2$ ,  $V_1 \leftarrow r / \|r\|_2$ ,  $j \leftarrow 1$ ;
  while ( $j \leq m$ ) do
    for  $i=1:j$  do
       $H_{(i,j)} \leftarrow V_j^t * A * V_i$ ;
     $V_{j+1} \leftarrow A * V_j - V * H_j$ ;
     $H_{(j+1,j)} \leftarrow \|V_{j+1}\|_2$ ;
     $V_{j+1} \leftarrow V_{j+1} / H_{(j+1,j)}$ ;
     $w_{(j+1)} \leftarrow 0$ ;
     $y \leftarrow \min_z \|w - H * z\|_2$  ( $H \in \mathbb{R}^{(j+1 \times j)}$ ) de Hessenberg;
     $x \leftarrow x_0 + V * y$ ;
     $iter \leftarrow iter + 1$ ;
     $j \leftarrow j + 1$ ;
    if  $\|r\|_2 < tol$  then PARAR;
  endwhile
   $x_0 \leftarrow x$ ;
until  $iter < \text{MaxIter}$ 

```

Forma de Hessenberg

$$H = \begin{bmatrix} \times & \times & \dots & \times \\ \times & \times & \dots & \times \\ 0 & \times & \ddots & \times \\ \vdots & 0 & \ddots & \times \\ 0 & 0 & \dots & \times \end{bmatrix} \in \mathbb{R}^{j+1 \times j}$$

En el GMRES(m) de NITSOL se utiliza un truco para no realizar rotaciones de Givens y es q se escoge

$$v_1 = \frac{Ar_0}{\|Ar_0\|_2}$$

BiCGSTAB y TFQMR

Idea General de Ambos: Proceso de ortogonalización no simétrico de Lanczos

1 BiCGSTAB

Es una combinación de BiCG y GMRES(1).
Utiliza un truco para mejorar la inestabilidad de
Gradiente BiConjugado

2 TFQMR

$$Ax = b \iff A^t Ax = A^t b$$

El sistema de la derecha simétrico y definido positivo
Utiliza un método ingenioso a evitar el producto por la
transpuesta.

BiCGSTAB y TFQMR

Idea General de Ambos: Proceso de ortogonalización no simétrico de Lanczos

1 BiCGSTAB

Es una combinación de BiCG y GMRES(1).

Utiliza un truco para mejorar la inestabilidad de *Gradiente BiConjugado*

2 TFQMR

$$Ax = b \iff A^t Ax = A^t b$$

El sistema de la derecha simétrico y definido positivo

Utiliza un método ingenioso a evitar el producto por la transpuesta.

BiCGSTAB y TFQMR

Idea General de Ambos: Proceso de ortogonalización no simétrico de Lanczos

Algunas características

- Utilizan fórmulas de recurrencia pequeñas.
- Tienen muy modestos costos de almacenamiento y de operaciones por iteración.
- El decrecimiento de la norma del gradiente es suave, casi monótono.
- No disfrutan de las propiedades de minimización del residual de GMRES(m).
- Pueden fallar prematuramente.
- Son más sensibles a errores de redondeo que GMRES.
- Si no tenemos muchos recursos (memoria) y convergen, son los tipos.

El Jacobiano en Diferencias

Diferentes aproximaciones

$$J(x_k) * d \approx$$

$$\textcircled{1} \frac{1}{h} [F(x_k + h d) - F(x_k)] + O(h)$$

$$\textcircled{2} \frac{1}{2h} [F(x_k + h d) - F(x_k - h d)] + O(h^2)$$

$$\textcircled{3} \frac{1}{6h} \left[8F\left(x_k + \frac{h}{2}d\right) - 8F\left(x_k - \frac{h}{2}d\right) - F(x_k + h d) + F(x_k - h d) \right] + O(h^4)$$

Observaciones Importantes

- No necesitamos la matriz explícitamente $J(x)$
- BiCGSTAB y TFQMR utilizan la jacobiana en cada iteración.
- BiCGSTAB y TFQMR aumenta las evaluaciones de F al usar fórmulas de orden alto.
- GMRES(m) solo utiliza cada m iteraciones la jacobiana.
- GMRES(m) permite usar fórmulas de orden alto con un costo no prohibitivo.

El Jacobiano en Diferencias

Diferentes aproximaciones

$$J(x_k) * d \approx$$

$$① \frac{1}{h} [F(x_k + h d) - F(x_k)] + O(h)$$

$$② \frac{1}{2h} [F(x_k + h d) - F(x_k - h d)] + O(h^2)$$

$$③ \frac{1}{6h} \left[8F\left(x_k + \frac{h}{2}d\right) - 8F\left(x_k - \frac{h}{2}d\right) - F(x_k + h d) + F(x_k - h d) \right] + O(h^4)$$

Observaciones Importantes

- No necesitamos la matriz explícitamente $J(x)$
- BiCGSTAB y TFQMR utilizan la jacobiana en cada iteración.
- BiCGSTAB y TFQMR aumenta las evaluaciones de F al usar fórmulas de orden alto.
- GMRES(m) solo utiliza cada m iteraciones la jacobiana.
- GMRES(m) permite usar fórmulas de orden alto con un costo no prohibitivo.

Presentación

- Subrutina NITSOL
- Especificación de F y JAC

Subrutina NITSOL

- Permite utilizar parámetros por defecto.
- Se pueden modificar gran cantidad de parámetros.
- Los requerimientos mínimos a especificar son:
 - 1 dimensión del problema
 - 2 vector de la aproximación inicial
 - 3 tolerancia de parada.
 - 4 espacio de trabajo para el método de Krylov.
 - 5 nombre de la rutina que evalúa a F .
 - 6 nombre de la rutina que calcula el producto del Jac por un vector.
- Indicar las subrutinas que calculan las normas y los productor internos, como `DNRM2` y `DDOT` de la biblioteca **BLAS**. Estas deben ser declaradas `EXTERNAL`
- El bloque `common NITINFO` contiene información sobre el progreso del algoritmo, tal como:
 - Cantidad de Iteraciones (no lineales)
 - $\|F(x_k)\|$
 - tasa de convergencia con la iteración previa.
 - status (fallo o no fallo)

Subrutina NITSOL

- Permite utilizar parámetros por defecto.
- Se pueden modificar gran cantidad de parámetros.
- Los requerimientos mínimos a especificar son:
 - 1 dimensión del problema
 - 2 vector de la aproximación inicial
 - 3 tolerancia de parada.
 - 4 espacio de trabajo para el método de Krylov.
 - 5 nombre de la rutina que evalúa a F .
 - 6 nombre de la rutina que calcula el producto del Jac por un vector.
- Indicar las subrutinas que calculan las normas y los productor internos, como `DNRM2` y `DDOT` de la biblioteca **BLAS**. Estas deben ser declaradas `EXTERNAL`
- El bloque `common NITINFO` contiene información sobre el progreso del algoritmo, tal como:
 - Cantidad de Iteraciones (no lineales)
 - $\|F(x_k)\|$
 - tasa de convergencia con la iteración previa.
 - status (fallo o no fallo)

Subrutina NITSOL

- Permite utilizar parámetros por defecto.
- Se pueden modificar gran cantidad de parámetros.
- Los requerimientos mínimos a especificar son:
 - 1 dimensión del problema
 - 2 vector de la aproximación inicial
 - 3 tolerancia de parada.
 - 4 espacio de trabajo para el método de Krylov.
 - 5 nombre de la rutina que evalúa a F .
 - 6 nombre de la rutina que calcula el producto del Jac por un vector.
- Indicar las subrutinas que calculan las normas y los productor internos, como `DNRM2` y `DDOT` de la biblioteca **BLAS**. Estas deben ser declaradas `EXTERNAL`
- El bloque `common NITINFO` contiene información sobre el progreso del algoritmo, tal como:
 - Cantidad de Iteraciones (no lineales)
 - $\|F(x_k)\|$
 - tasa de convergencia con la iteración previa.
 - status (fallo o no fallo)

Subrutina NITSOL

- Permite utilizar parámetros por defecto.
- Se pueden modificar gran cantidad de parámetros.
- Los requerimientos mínimos a especificar son:
 - 1 dimensión del problema
 - 2 vector de la aproximación inicial
 - 3 tolerancia de parada.
 - 4 espacio de trabajo para el método de Krylov.
 - 5 nombre de la rutina que evalúa a F .
 - 6 nombre de la rutina que calcula el producto del Jac por un vector.
- Indicar las subrutinas que calculan las normas y los productor internos, como `DNRM2` y `DDOT` de la biblioteca **BLAS**. Estas deben ser declaradas `EXTERNAL`
- El bloque `common NITINFO` contiene información sobre el progreso del algoritmo, tal como:
 - Cantidad de Iteraciones (no lineales)
 - $\|F(x_k)\|$
 - tasa de convergencia con la iteración previa.
 - status (fallo o no fallo)

Presentación

- Subrutina NITSOL
- Especificación de F y JAC

Función Objetivo que debe dar el usuario

SUBROUTINE F(N, XCUR, FCUR, RPAR, IPAR, ITRMF)

- **N**: dimensión (En el caso del petroleo la cantidad de nodos de la malla)
- **XCUR**: vector de dim N donde se evaluará F .
- **FCUR**: vector de dim N de la evaluación de $F(XCUR)$.
- **RPAR** y **IPAR** vectores de parámetros reales y enteros respectivamente. Por ejemplo en **IPAR** puede estar información relativa a la malla: cantidad de puntos en cada una de las direcciones, etc.
- **ITRMF** Es una bandera entera que indica si la evaluación de F fue exitosa y no dio algo como NaN o ∞ . Estas excepciones deben ser controladas por el usuario.

Implementación de Jacobiano

```
SUBROUTINE JACV(N, XCUR, FCUR, IJOB, V, Z, RPAR, IPAR,  
ITRMJV)
```

- **N**: dimensión del problema.
- **XCUR**: vector de dim **N** donde se evaluó **F**.
- **FCUR**: vector de evaluación de **F** (**XCUR**).
- **IJOB**: Bandera que indica si se usara preconditionador.
- **V**: vector de dim **N** por el que se multiplicará el jacobiano.
- **Z**: vector de dim **N** resultado de multiplicar por el jacobiano.
- **RPAR** y **IPAR** vectores de parámetros reales y enteros respectivamente.
- **ITRMJV** Bandera que indica si el proceso fue exitoso.

Implementación de Jacobiano

Detalles: SUBROUTINE JACV(N, XCUR, FCUR, IJOB, V, Z, RPAR, IPAR, ITRMJV)

```

IF (IJOB .EQ. 0) THEN !Realiza el prod Jac por vector
  Código ...
  Código ...
  Código ...
ELSE (IJOB .EQ. 1) THEN !aplica el preconditiondor
  Código ...
  Código ...
  Código ...
END IF

```

Aplicar un preconditionador, es aplicar un procedimiento que reciba un vector y devuelva otro!!

Necesidad de XCUR y FCUR

$$J(x_k) d \approx \frac{1}{h} \underbrace{\left[F(x_k + h \overset{V}{d}) - F(x_k) \right]}_Z$$

Implementación de Jacobiano

Detalles: SUBROUTINE JACV(N, XCUR, FCUR, IJOB, V, Z, RPAR, IPAR, ITRMJV)

```

IF (IJOB .EQ. 0) THEN !Realiza el prod Jac por vector
  Código ...
  Código ...
  Código ...
ELSE (IJOB .EQ. 1) THEN !aplica el preconditionador
  Código ...
  Código ...
  Código ...
END IF

```

Aplicar un preconditionador, es aplicar un procedimiento que reciba un vector y devuelva otro!!

Necesidad de XCUR y FCUR

$$J(x_k) d \approx \frac{1}{h} \underbrace{\left[F(x_k + h \overset{V}{d}) - F(x_k) \right]}_Z$$

AL
FIN
ACABÓ