Literature Study

The Mild-Slope Equation and its Numerical Implementation

Gemma van de Sande

TU Delft

December 8, 2011

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Svašek Hydraulics

- Founded in 1969 by Mr. J.N. Svašek;

Svašek Hydraulics

- Founded in 1969 by Mr. J.N. Svašek;
- Consultant in coastal, harbour and river engineering;

Svašek Hydraulics

- Founded in 1969 by Mr. J.N. Svašek;
- Consultant in coastal, harbour and river engineering;
- Specialised in numerical fluid dynamics;

Svašek Hydraulics

- Founded in 1969 by Mr. J.N. Svašek;
- Consultant in coastal, harbour and river engineering;
- Specialised in numerical fluid dynamics;
- 17 employees.

MSc project

- HARES (HArbour RESonance).
  - Determines wave penetration into harbours;
  - Numerical implementation of the Mild-Slope equation with the finite element method.

MSc project

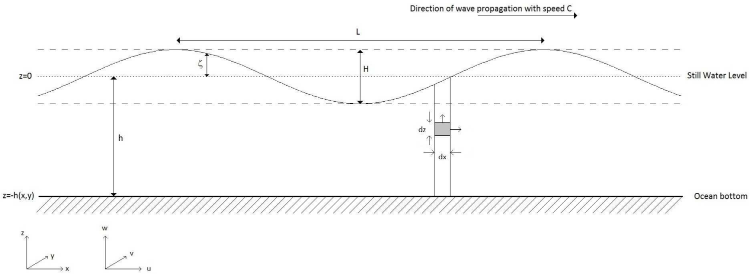
- HARES (HArbour RESonance).
  - Determines wave penetration into harbours;
  - Numerical implementation of the Mild-Slope equation with the finite element method.
- Non-linearity of the problem is treated with
  - Outer iteration: *Picard's method*;
  - Inner iteration: *ILU-BiCGSTAB*.

MSc project

- HARES (HArbour RESonance).
  - Determines wave penetration into harbours;
  - Numerical implementation of the Mild-Slope equation with the finite element method.
- Non-linearity of the problem is treated with
  - Outer iteration: *Picard's method*;
  - Inner iteration: *ILU-BiCGSTAB*.
- Improve the undesired long computational time.

Outline

Geometry



h	Water height	k_0	Wave number
ζ	Elevation of the free surface	ω	Wave frequency
a	Absolute elevation, $a = \|\zeta\|$	L	Wave length, $L = 2\pi/k_0$
H	Wave height, $H = 2A$	T	Wave period, $T = 2\pi/\omega$

To model the wave motion the following assumptions are made [3].

To model the wave motion the following assumptions are made [3].

- Water is an ideal fluid, i.e. homogeneous, inviscid, irrotational and incompressible;

To model the wave motion the following assumptions are made [3].

- Water is an ideal fluid, i.e. homogeneous, inviscid, irrotational and incompressible;
- The pressure is constant and uniform at the free surface;

Wave motion

Assumptions on the wave motion

To model the wave motion the following assumptions are made [3].

- Water is an ideal fluid, i.e. homogeneous, inviscid, irrotational and incompressible;
- The pressure is constant and uniform at the free surface;
- The wave slope $\epsilon = \frac{2\pi A}{L}$ is small;

To model the wave motion the following assumptions are made [3].

- Water is an ideal fluid, i.e. homogeneous, inviscid, irrotational and incompressible;
- The pressure is constant and uniform at the free surface;
- The wave slope $\epsilon = \frac{2\pi A}{L}$ is small;
- The wave motion is time harmonic;

To model the wave motion the following assumptions are made [3].

- Water is an ideal fluid, i.e. homogeneous, inviscid, irrotational and incompressible;
- The pressure is constant and uniform at the free surface;
- The wave slope $\epsilon = \frac{2\pi A}{L}$ is small;
- The wave motion is time harmonic;
- The changes in bottom topography are small;

To model the wave motion the following assumptions are made [3].

- Water is an ideal fluid, i.e. homogeneous, inviscid, irrotational and incompressible;
- The pressure is constant and uniform at the free surface;
- The wave slope $\epsilon = \frac{2\pi A}{L}$ is small;
- The wave motion is time harmonic;
- The changes in bottom topography are small;
- The surface tension can be neglected;
- The Coriolis effect can be neglected.

The following effects are included in HARES.

The following effects are included in HARES.

- Diffraction;

Wave transforming effects

Included in HARES

The following effects are included in HARES.

- Diffraction;
- Reflection;

The following effects are included in HARES.

- Diffraction;
- Reflection;
- Refraction;

The following effects are included in HARES.

- Diffraction;
- Reflection;
- Refraction;
- Shoaling;

The following effects are included in HARES.

- Diffraction;
- Reflection;
- Refraction;
- Shoaling;
- Wave breaking;

The following effects are included in HARES.

- Diffraction;
- Reflection;
- Refraction;
- Shoaling;
- Wave breaking;
- Bottom friction.

Definition

Wave breaking - The process that causes large amounts of wave energy to be transformed into turbulent kinetic energy.

Definition

Wave breaking - The process that causes large amounts of wave energy to be transformed into turbulent kinetic energy.

The energy dissipation due to wave breaking is described by the coefficient W_b [1];

$$W_b = \frac{2\alpha}{T} Q_b \frac{H_m^2}{4a^2}.$$

With

- α Adjustable constant;
- Q_b Fraction of breaking waves;
- H_m Maximal possible wave height;
- a Modulus of the free surface elevation, $a = |\tilde{\zeta}|$.

Definition

Bottom friction - The momentum transfer of wave energy to the solid earth by friction at the ocean bottom.

Definition

Bottom friction - The momentum transfer of wave energy to the solid earth by friction at the ocean bottom.

The energy dissipation due to bottom friction is described by the coefficient W_f [4, 22];

$$W_f = \frac{8}{3\pi} c_f \frac{a\omega^3}{\sinh^3(k_0 h)}.$$

With

c_f Bottom friction coefficient.

Outline

The Mild-Slope equation

The Mild-Slope equation, with energy dissipation included, is given by [4]

$$\nabla \cdot \left(\frac{n_0}{k_0^2} \nabla \tilde{\zeta} \right) + \left(n_0 - \frac{iW}{\omega} \right) \tilde{\zeta} = 0, \tag{1}$$

with

- W The energy dissipation term $W = W_f + W_b$;
- n_0 A constant $n_0 = \frac{1}{2} \left(1 + \frac{2k_0 h}{\sinh(2k_0 h)} \right)$;
- ∇ $= \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)^T$.

The Mild-Slope equation

The Mild-Slope equation, with energy dissipation included, is given by [4]

$$\nabla \cdot \left(\frac{n_0}{k_0^2} \nabla \tilde{\zeta} \right) + \left(n_0 - \frac{iW}{\omega} \right) \tilde{\zeta} = 0, \tag{1}$$

Non-linearity:

$$W\tilde{\zeta} = \left(\frac{8}{3\pi} c_f \frac{|\tilde{\zeta}|\omega^3}{\sinh^3(k_0 h)} + \frac{2\alpha}{T} Q_b \frac{H_m 2}{4|\tilde{\zeta}|^2} \right) \tilde{\zeta}$$

There are two types of boundaries, i.e.

- The *open boundary* with an incoming and an outgoing wave.
- The *closed boundary* where partial reflection due to interaction with the boundary occurs.

The condition for the open boundary (Γ_1);

$$\frac{\partial \tilde{\zeta}}{\partial n} = -p\tilde{\zeta}_{in}\left(\boldsymbol{e}_{in} \cdot \boldsymbol{n}\right) - p(\tilde{\zeta} - \tilde{\zeta}_{in}) + \frac{1}{2p}\left(\frac{\partial^2 \tilde{\zeta}}{\partial s^2} + \frac{\partial^2 \tilde{\zeta}_{in}}{\partial s^2}\right).$$

The condition for the open boundary (Γ_1);

$$\frac{\partial \tilde{\zeta}}{\partial n} = -p\tilde{\zeta}_{in}\left(\boldsymbol{e}_{in} \cdot \boldsymbol{n}\right) - p(\tilde{\zeta} - \tilde{\zeta}_{in}) + \frac{1}{2p}\left(\frac{\partial^2 \tilde{\zeta}}{\partial s^2} + \frac{\partial^2 \tilde{\zeta}_{in}}{\partial s^2}\right).$$

The condition for the closed boundary (Γ_2);

$$\frac{\partial \tilde{\zeta}}{\partial n} = -\left(\frac{1-R}{1+R}\right)\left\{p\tilde{\zeta} - \frac{1}{2p}\frac{\partial^2 \tilde{\zeta}}{\partial s^2}\right\}.$$

The condition for the open boundary (Γ_1);

$$\frac{\partial \tilde{\zeta}}{\partial n} = -p\tilde{\zeta}_{in}\left(\boldsymbol{e}_{in}\cdot\boldsymbol{n}\right) - p(\tilde{\zeta}-\tilde{\zeta}_{in}) + \frac{1}{2p}\left(\frac{\partial^2\tilde{\zeta}}{\partial s^2} + \frac{\partial^2\tilde{\zeta}_{in}}{\partial s^2}\right).$$

The condition for the closed boundary (Γ_2);

$$\frac{\partial \tilde{\zeta}}{\partial n} = -\left(\frac{1-R}{1+R}\right)\left\{p\tilde{\zeta} - \frac{1}{2p}\frac{\partial^2\tilde{\zeta}}{\partial s^2}\right\}.$$

With

- R the reflection coefficient, $0 \leq R \leq 1$;
- p The modified wave number $p = ik_0\sqrt{1-\frac{iW}{\omega n_0}}$.

Outline

Currently used Numerical Methods

- Ritz-Galerkin Finite Element Method
- Bi-CGSTAB
- Incomplete LU factorization
- Picard iteration

The weak formulation of the Mild-Slope equation is given by

$$\int_\Omega \left\{ \left(n_0 - \frac{iW}{\omega} \right) \tilde{\zeta}\eta - \frac{n_0}{k_0^2} \nabla\tilde{\zeta} \cdot \nabla\eta \right\} \, d\Omega$$

$$- \int_{\Gamma_2} \frac{n_0}{k_0^2} \left(\frac{1-R}{1+R} \right) \left\{ p\tilde{\zeta}\eta + \frac{1}{2p} \frac{\partial\tilde{\zeta}}{\partial s} \frac{\partial\eta}{\partial s} \right\} \, d\Gamma$$

$$- \int_{\Gamma_1} \frac{n_0}{k_0^2} \left\{ p\tilde{\zeta}\eta + \frac{1}{2p} \frac{\partial\tilde{\zeta}}{\partial s} \frac{\partial\eta}{\partial s} \right\} \, d\Gamma$$

$$= \int_{\Gamma_1} \frac{n_0}{k_0^2} \left\{ p\tilde{\zeta}_{in} \left(\boldsymbol{e}_{in} \cdot \boldsymbol{n} \right) \eta - p\tilde{\zeta}_{in}\eta - \frac{1}{2p} \left(\frac{\partial\tilde{\zeta}_{in}}{\partial s} \frac{\partial\eta}{\partial s} \right) \right\} \, d\Gamma.$$

The domain is divided into triangles, with piecewise linear basis functions. This results in the following system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y},$$

$\boldsymbol{A} \in \mathbb{C}^{N \times N}$, $\boldsymbol{x} \in \mathbb{C}^N$ and $\boldsymbol{y} \in \mathbb{C}^N$. With

$$\boldsymbol{A} = \left(-\boldsymbol{L} - \boldsymbol{C} + z_1 \boldsymbol{M}\right).$$

The domain is divided into triangles, with piecewise linear basis functions. This results in the following system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y},$$

$\boldsymbol{A} \in \mathbb{C}^{N \times N}$, $\boldsymbol{x} \in \mathbb{C}^{N}$ and $\boldsymbol{y} \in \mathbb{C}^{N}$. With

$$\boldsymbol{A} = \left(-\boldsymbol{L} - \boldsymbol{C} + z_1 \boldsymbol{M}\right).$$

With

$$\int_{\Omega} \frac{n_0}{k_0^2} \nabla \tilde{\zeta} \cdot \nabla \eta \; d\Omega \quad \Rightarrow \quad \boldsymbol{L}$$

The domain is divided into triangles, with piecewise linear basis functions. This results in the following system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y},$$

$\boldsymbol{A} \in \mathbb{C}^{N\times N}$, $\boldsymbol{x} \in \mathbb{C}^N$ and $\boldsymbol{y} \in \mathbb{C}^N$. With

$$\boldsymbol{A} = \left(-\boldsymbol{L} - \boldsymbol{C} + z_1 \boldsymbol{M}\right).$$

With

$$\int_{\Gamma_2} \frac{n_0}{k_0^2}\left(\frac{1-R}{1+R}\right)\left\{p\tilde{\zeta}\eta + \frac{1}{2p}\frac{\partial\tilde{\zeta}}{\partial s}\frac{\partial\eta}{\partial s}\right\}\, d\Gamma$$
$$+ \int_{\Gamma_1} \frac{n_0}{k_0^2}\left\{p\tilde{\zeta}\eta + \frac{1}{2p}\frac{\partial\tilde{\zeta}}{\partial s}\frac{\partial\eta}{\partial s}\right\}\, d\Gamma \quad \Rightarrow \quad \boldsymbol{C}$$

The domain is divided into triangles, with piecewise linear basis functions. This results in the following system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y},$$

$\boldsymbol{A} \in \mathbb{C}^{N\times N}$, $\boldsymbol{x} \in \mathbb{C}^{N}$ and $\boldsymbol{y} \in \mathbb{C}^{N}$. With

$$\boldsymbol{A} = \left(-\boldsymbol{L} - \boldsymbol{C} + z_1\boldsymbol{M}\right).$$

With

$$\int_\Omega \tilde{\zeta}\eta \; d\Omega \quad \Rightarrow \quad \boldsymbol{M} \quad \text{and} \quad z_1 = \left(n_0 - \frac{iW}{\omega}\right)$$

The domain is divided into triangles, with piecewise linear basis functions. This results in the following system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y},$$

$\boldsymbol{A} \in \mathbb{C}^{N \times N}$, $\boldsymbol{x} \in \mathbb{C}^{N}$ and $\boldsymbol{y} \in \mathbb{C}^{N}$. With

$$\boldsymbol{A} = \left(-\boldsymbol{L} - \boldsymbol{C} + z_1 \boldsymbol{M}\right).$$

With

$$\int_{\Gamma_1} \frac{n_0}{k_0^2} \left\{ p\tilde{\zeta}_{in} \left(\boldsymbol{e}_{in} \cdot \boldsymbol{n}\right) \eta - p\tilde{\zeta}_{in}\eta - \frac{1}{2p} \left(\frac{\partial \tilde{\zeta}_{in}}{\partial s} \frac{\partial \eta}{\partial s} \right) \right\} d\Gamma \quad \Rightarrow \quad \boldsymbol{y}$$

- Bi-CGSTAB solves $\boldsymbol{Ax} = \boldsymbol{b}$ with the residual $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{Ax}_0$.

Numerical Matrix Solver
Bi-CGSTAB

- Bi-CGSTAB solves $\boldsymbol{Ax} = \boldsymbol{b}$ with the residual $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{Ax}_0$.
- Proposed by H.A. van der Vorst in 1992.

- Bi-CGSTAB solves $\boldsymbol{Ax} = \boldsymbol{b}$ with the residual $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{Ax}_0$.
- Proposed by H.A. van der Vorst in 1992.
- Bi-CGSTAB is a Krylov subspace method, the Krylov subspace of dimension m is given by

$$\mathcal{K}_m(\boldsymbol{A}; \boldsymbol{r}_0) = span\{\boldsymbol{r}_0, \boldsymbol{Ar}_0, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{r}_0\}.$$

- Bi-CGSTAB solves $\boldsymbol{Ax} = \boldsymbol{b}$ with the residual $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{Ax}_0$.
- Proposed by H.A. van der Vorst in 1992.
- Bi-CGSTAB is a Krylov subspace method, the Krylov subspace of dimension m is given by

$$\mathcal{K}_m(\boldsymbol{A}; \boldsymbol{r}_0) = span\{\boldsymbol{r}_0, \boldsymbol{Ar}_0, \ldots, \boldsymbol{A}^{m-1}\boldsymbol{r}_0\}.$$

- The residual of Bi-CGSTAB can be written as

$$\boldsymbol{r}_i^{Bi-CGSTAB} = Q_i(\boldsymbol{A})P_i(\boldsymbol{A})\boldsymbol{r}_0,$$

with

$$Q_i(\boldsymbol{A}) = (I - \omega_1\boldsymbol{A})(I - \omega_2\boldsymbol{A})\ldots(I - \omega_i\boldsymbol{A}).$$

- Test performed in the literature study based on the incomplete LU factorization without fill-in (ILU(0)).

- Test performed in the literature study based on the incomplete LU factorization without fill-in (ILU(0)).
- ILU(0) is based on
  - Matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ have the same zero-pattern as $\boldsymbol{A}$, i.e. $u_{i,j} = l_{i,j} = 0$ if $a_{i,j} = 0$ and if $a_{i,j} \neq 0$ then $u_{i,j} \neq 0$ and $l_{i,j} \neq 0$.
  - $l_{i,i} = 1$ and $u_{i,i}$ is determined by the algorithm.

- Test performed in the literature study based on the incomplete LU factorization without fill-in (ILU(0)).
- ILU(0) is based on
  - Matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ have the same zero-pattern as $\boldsymbol{A}$, i.e. $u_{i,j} = l_{i,j} = 0$ if $a_{i,j} = 0$ and if $a_{i,j} \neq 0$ then $u_{i,j} \neq 0$ and $l_{i,j} \neq 0$.
  - $l_{i,i} = 1$ and $u_{i,i}$ is determined by the algorithm.
- Preconditioning is done by $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{U}^{-1}\boldsymbol{y} = \boldsymbol{L}^{-1}\boldsymbol{b}$ with $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{x}$.

- Test performed in the literature study based on the incomplete LU factorization without fill-in (ILU(0)).
- ILU(0) is based on
  - Matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ have the same zero-pattern as $\boldsymbol{A}$, i.e. $u_{i,j} = l_{i,j} = 0$ if $a_{i,j} = 0$ and if $a_{i,j} \neq 0$ then $u_{i,j} \neq 0$ and $l_{i,j} \neq 0$.
  - $l_{i,i} = 1$ and $u_{i,i}$ is determined by the algorithm.
- Preconditioning is done by $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{U}^{-1}\boldsymbol{y} = \boldsymbol{L}^{-1}\boldsymbol{b}$ with $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{x}$.
- The LU decomposition proposed by A. van der Ploeg [16] is used in HARES.

Picard iteration

The following algorithm is used for the Picard iteration:

	$W^0 = 0$; initial value for the dissipation term
for	$i = 1, 2, \dots$
	Solve $\boldsymbol{x}^i$ from $\boldsymbol{A}(W^{i-1})\boldsymbol{x}^i = \boldsymbol{b}$
	$W^i = W(\boldsymbol{x}^i)$
end	

Outline

Proposed Numerical Matrix Solver
IDR(s)

- IDR is proposed by P. Sonneveld in 1980;

- IDR is proposed by P. Sonneveld in 1980;
- Krylov subspace method;

- IDR is proposed by P. Sonneveld in 1980;
- Krylov subspace method;
- Generate residuals $\boldsymbol{r}_n$ that are in the subspace $\mathcal{G}_j$ with decreasing dimension;

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A}) \left(\mathcal{G}_{j-1} \cap \boldsymbol{P}^{\perp} \right),$$

with $\mathcal{G}_0 = \mathcal{K}^N(\boldsymbol{A}; \boldsymbol{v}_0)$ and $\boldsymbol{P} \in \mathbb{C}^{N \times s}$.

- IDR is proposed by P. Sonneveld in 1980;
- Krylov subspace method;
- Generate residuals $\boldsymbol{r}_n$ that are in the subspace $\mathcal{G}_j$ with decreasing dimension;

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A}) \left(\mathcal{G}_{j-1} \cap \boldsymbol{P}^\perp \right),$$

  with $\mathcal{G}_0 = \mathcal{K}^N(\boldsymbol{A}; \boldsymbol{v}_0)$ and $\boldsymbol{P} \in \mathbb{C}^{N \times s}$.
- Based on the IDR theorem [13], which states that
  - (i) $\mathcal{G}_j \subset \mathcal{G}_{j-1}$ for all $\mathcal{G}_{j-1} \neq \{\mathbf{0}\}, j > 0$,
  - (ii) $\mathcal{G}_j = \{\mathbf{0}\}$ for some $j \leq N$

- IDR is proposed by P. Sonneveld in 1980;
- Krylov subspace method;
- Generate residuals $\boldsymbol{r}_n$ that are in the subspace $\mathcal{G}_j$ with decreasing dimension;

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A}) \left(\mathcal{G}_{j-1} \cap \boldsymbol{P}^{\perp} \right),$$

  with $\mathcal{G}_0 = \mathcal{K}^N(\boldsymbol{A}; \boldsymbol{v}_0)$ and $\boldsymbol{P} \in \mathbb{C}^{N \times s}$.
- Based on the IDR theorem [13], which states that
  - (i) $\mathcal{G}_j \subset \mathcal{G}_{j-1}$ for all $\mathcal{G}_{j-1} \neq \{\boldsymbol{0}\}, j > 0$,
  - (ii) $\mathcal{G}_j = \{\boldsymbol{0}\}$ for some $j \leq N$
- Residuals are obtained by

$$\boldsymbol{r}_{n+1} = (\boldsymbol{I} - \omega_{j+1} \boldsymbol{A}) \boldsymbol{v}_n \quad \text{with} \quad \boldsymbol{v}_n \in \mathcal{G}_j \cap \boldsymbol{P}^{\perp}.$$

- IDR is proposed by P. Sonneveld in 1980;
- Krylov subspace method;
- Generate residuals $\boldsymbol{r}_n$ that are in the subspace $\mathcal{G}_j$ with decreasing dimension;

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A}) \left(\mathcal{G}_{j-1} \cap \boldsymbol{P}^{\perp} \right),$$

  with $\mathcal{G}_0 = \mathcal{K}^N(\boldsymbol{A}; \boldsymbol{v}_0)$ and $\boldsymbol{P} \in \mathbb{C}^{N \times s}$.
- Based on the IDR theorem [13], which states that
  - (i) $\mathcal{G}_j \subset \mathcal{G}_{j-1}$ for all $\mathcal{G}_{j-1} \neq \{\mathbf{0}\}, j > 0$,
  - (ii) $\mathcal{G}_j = \{\mathbf{0}\}$ for some $j \leq N$
- Residuals are obtained by

$$\boldsymbol{r}_{n+1} = (\boldsymbol{I} - \omega_{j+1} \boldsymbol{A}) \boldsymbol{v}_n \quad \text{with} \quad \boldsymbol{v}_n \in \mathcal{G}_j \cap \boldsymbol{P}^{\perp}.$$

- Requires at most $N + \frac{N}{s}$ matrix-vector multiplications.

Freedom in the IDR(s) algorithm

Freedom in the IDR(s) algorithm

- Choosing $\boldsymbol{P}$.

Freedom in the IDR(s) algorithm

- Choosing $\boldsymbol{P}$.

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A}) \left(\mathcal{G}_{j-1} \cap \boldsymbol{P}^{\perp} \right)$$

Currently orthogonalized random vectors are used.

Freedom in the IDR(s) algorithm

- Choosing $\boldsymbol{P}$.

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A}) \left(\mathcal{G}_{j-1} \cap \boldsymbol{P}^{\perp} \right)$$

- Choosing ω_j.

Freedom in the IDR(s) algorithm

- Choosing $\boldsymbol{P}$.

$$\mathcal{G}_j = (\boldsymbol{I} - \omega_j \boldsymbol{A}) \left(\mathcal{G}_{j-1} \cap \boldsymbol{P}^{\perp} \right)$$

- Choosing ω_j. For each residual in $\mathcal{G}_j$ the same ω_j is needed, currently based on a strategy proposed by Sleijpen and Van der Vorst. [11]

Freedom in the IDR(s) algorithm

- Choosing $\boldsymbol{P}$.
- Choosing ω_j.
- Building $\mathcal{G}_0$.

Freedom in the IDR(s) algorithm

- Choosing $\boldsymbol{P}$.
- Choosing ω_j.
- Building $\mathcal{G}_0$.
    (i) Can be done using a simple Krylov method;

Freedom in the IDR(s) algorithm

- Choosing $\boldsymbol{P}$.
- Choosing ω_j.
- Building $\mathcal{G}_0$.
  (i) Can be done using a simple Krylov method;
  (ii) Can be chosen freely as long as $\mathcal{G}_0$ is the complete Krylov subspace $\mathcal{K}^N$.

- The shifted Laplace preconditioner [6], used in our experiments, is given by

$$\boldsymbol{K} = (-\boldsymbol{L} - \boldsymbol{C} - i|z_1|\boldsymbol{M}),$$

with $z_1 = \left(n_0 - \frac{iW}{\omega}\right)$.

- The shifted Laplace preconditioner [6], used in our experiments, is given by

$$\boldsymbol{K} = (-\boldsymbol{L} - \boldsymbol{C} - i|z_1|\boldsymbol{M}),$$

  with $z_1 = \left(n_0 - \frac{iW}{\omega}\right)$.
- The matrix $\boldsymbol{K}$ is approximated with the incomplete LU factorization.

- The shifted Laplace preconditioner [6], used in our experiments, is given by

$$\boldsymbol{K} = (-\boldsymbol{L} - \boldsymbol{C} - i|z_1|\boldsymbol{M}),$$

  with $z_1 = \left(n_0 - \frac{iW}{\omega}\right)$.
- The matrix $\boldsymbol{K}$ is approximated with the incomplete LU factorization.
- The preconditioned system is given by

$$(-\boldsymbol{L} - \boldsymbol{C} - i|z_1|\boldsymbol{M})^{-1}(-\boldsymbol{L} - \boldsymbol{C} + z_1\boldsymbol{M})\boldsymbol{a} = (-\boldsymbol{L} - \boldsymbol{C} - i|z_1|\boldsymbol{M})^{-1}\boldsymbol{f}.$$

- The shifted Laplace preconditioner [6], used in our experiments, is given by
  $$\boldsymbol{K} = (-\boldsymbol{L} - \boldsymbol{C} - i|z_1|\boldsymbol{M}),$$
  with $z_1 = \left(n_0 - \frac{iW}{\omega}\right)$.
- The matrix $\boldsymbol{K}$ is approximated with the incomplete LU factorization.
- The preconditioned system is given by
  $$(-\boldsymbol{L} - \boldsymbol{C} - i|z_1|\boldsymbol{M})^{-1}(-\boldsymbol{L} - \boldsymbol{C} + z_1\boldsymbol{M})\boldsymbol{a} = (-\boldsymbol{L} - \boldsymbol{C} - i|z_1|\boldsymbol{M})^{-1}\boldsymbol{f}.$$

If $\boldsymbol{L}$ and $\boldsymbol{C}$ are symmetric positive semidefinite real matrices and $\boldsymbol{M}$ symmetric positive definite real matrix, that the eigenvalues of the preconditioned system lie inside or on a circle. [19]

Outline

The following test problem is considered:

The following test problem is considered:

- Harbour of Scheveningen;

Numerical Experiments

Problem dimension & used matrix-solvers

The following test problem is considered:

- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;

The following test problem is considered:

- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;
- 25 outer iterations.

The following test problem is considered:

- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;
- 25 outer iterations.

With the numerical methods and preconditioners:

The following test problem is considered:

- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;
- 25 outer iterations.

With the numerical methods and preconditioners:

- Bi-CGSTAB,

Numerical Experiments

Problem dimension & used matrix-solvers

The following test problem is considered:

- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;
- 25 outer iterations.

With the numerical methods and preconditioners:

- Bi-CGSTAB, CG-S,

The following test problem is considered:

- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;
- 25 outer iterations.

With the numerical methods and preconditioners:

- Bi-CGSTAB, CG-S, IDR(2),

The following test problem is considered:

- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;
- 25 outer iterations.

With the numerical methods and preconditioners:

- Bi-CGSTAB, CG-S, IDR(2), IDR(4);

The following test problem is considered:

- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;
- 25 outer iterations.

With the numerical methods and preconditioners:

- Bi-CGSTAB, CG-S, IDR(2), IDR(4);
- ILU(0),

The following test problem is considered:

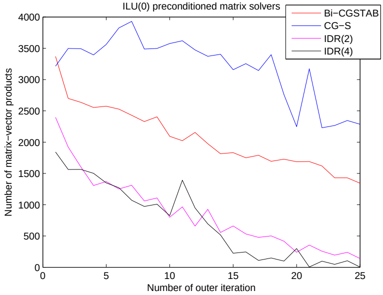
- Harbour of Scheveningen;
- 126.504 internal triangular elements;
- 2.213 boundary line segments;
- 63.253 unknowns;
- 25 outer iterations.

With the numerical methods and preconditioners:

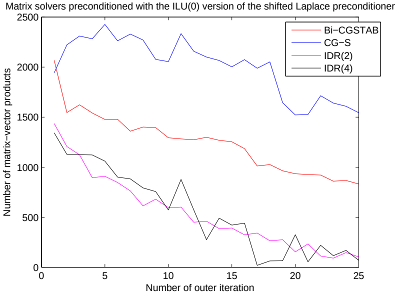
- Bi-CGSTAB, CG-S, IDR(2), IDR(4);
- ILU(0), ILU(0)-Shifted Laplace.

Numerical Experiments

Number of matrix-vector products



(c) Number of matvecs for the ILU(0) preconditioned system



(d) Number of matvecs for the ILU(0)-shifted laplace preconditioned system

Numerical Experiments

Computational time

	Numerical method			
	Bi-CGSTAB	CG-S	IDR(2)	IDR(4)
ILU(0)	$3.0192 \cdot 10^3$	$4.1053 \cdot 10^3$	$1.2303 \cdot 10^3$	$1.2863 \cdot 10^3$
ILU(0)-SL	$1.8525 \cdot 10^3$	$2.6634 \cdot 10^3$	$0.7361 \cdot 10^3$	$0.9741 \cdot 10^3$

Table: CPU time until the whole process is completed.

	Numerical method			
	Bi-CGSTAB	CG-S	IDR(2)	IDR(4)
ILU(0)	$3.0192 \cdot 10^3$	$4.1053 \cdot 10^3$	$1.2303 \cdot 10^3$	$1.2863 \cdot 10^3$
ILU(0)-SL	$1.8525 \cdot 10^3$	$2.6634 \cdot 10^3$	$0.7361 \cdot 10^3$	$0.9741 \cdot 10^3$

Table: CPU time until the whole process is completed.

Using IDR(s) preconditioned with the incomplete LU factorization of the shifted Laplace matrix speeds the computational time up with a factor three.

Outline

Research objectives

Improvement of HARES :

Research objectives

Improvement of HARES :

- The non-linear part;

Research objectives

Improvement of HARES :

- The non-linear part;

(i) Currently just 25 outer iterations;

Research objectives

Improvement of HARES :

- The non-linear part;

(i) Currently just 25 outer iterations;

(ii) Implementing a stopping criterion;

Research objectives

Improvement of HARES :

- The non-linear part;

(i) Currently just 25 outer iterations;

(ii) Implementing a stopping criterion;

(iii) Total error.

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Theoretical research:

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Theoretical research:

- Spectral analysis for preconditioner of the Mild-Slope equation;

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Theoretical research:

- Spectral analysis for preconditioner of the Mild-Slope equation;

(i) Analysis in [19] not identical for Mild-Slope equation;

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Theoretical research:

- Spectral analysis for preconditioner of the Mild-Slope equation;

(i) Analysis in [19] not identical for Mild-Slope equation;

(ii) Determine an optimal shift when possible;

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Theoretical research:

- Spectral analysis for preconditioner of the Mild-Slope equation;

(i) Analysis in [19] not identical for Mild-Slope equation;

(ii) Determine an optimal shift when possible;

(iii) How should we approximate the shifted Laplace preconditioner.

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Theoretical research:

- Spectral analysis for preconditioner of the Mild-Slope equation;
- Choosing the coefficients ω_j based on Ritz-values;

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Theoretical research:

- Spectral analysis for preconditioner of the Mild-Slope equation;
- Choosing the coefficients ω_j based on Ritz-values;

When the spectrum is a circle we might be able to determine the Ritz-values such that the polynomial

$$Q_j(\boldsymbol{A}) = (\boldsymbol{I} - \omega_1 \boldsymbol{A}) \ldots (\boldsymbol{I} - \omega_j \boldsymbol{A}),$$

has a minimal maximum on the spectrum.

Research objectives

Improvement of HARES :

- The non-linear part;
- Testing the numerical methods on several test problems;
- Implementation into FORTRAN.

Theoretical research:

- Spectral analysis for preconditioner of the Mild-Slope equation;
- Choosing the coefficients ω_j based on Ritz-values;
- Can smartly building $\mathcal{G}_0$ lead to convergence speed up.

Questions

QUESTIONS?

Figure diffraction & reflection

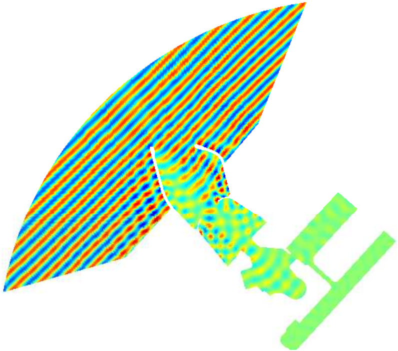
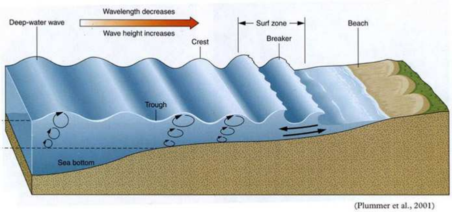


Figure: The harbour of Scheveningen. The effects of diffraction - reflection visible

Figure refraction & shoaling

(a) Refraction



(b) Shoaling

Bibliography I

J.A. Battjes and J.P.F.M. Janssen. Energy loss and set-up due to breaking of random waves.
Proc. 16th Int. Conf. on Coastal Engineering, 1978.

J.C.W. Berkhoff. *Mathematical models for simple harmonic linear water wave models; wave refraction and diffraction*.
PhD thesis, Technical University of Delft, 1976.

U.S. Army Coastal Engineering Research Center. *Shore Protection Manual*.
U.S. Government Printing Office, Washington D.C., 1984.

M.W. Dingemans. *Water wave propagation over uneven bottoms, Part 1 - Linear Wave Propagation*.
World Scientific, first edition, 1997.

B.J.O. Eikema and B.C. van Prooijen. HARES - numerical model for the determination of wave penetration in harbour basins.
Svasek-rapport t.b.v. validatie HARES, 2005.

Y.A. Erlangga, C. Vuik, and C.W. Oosterlee. On a class of preconditioners for solving the helmholtz equation.
Applied Numerical Mathematics, 50:409–425, 2004.

R. Fletcher. Conjugate gradient methods for indefinite systems.
Lecture Notes in Math, 506:73–89, 1976.

C.C. Mei. *The Applied Dynamics of Ocean Surface Waves*.
World Scientific, second edition, 1989.

J.A. Meijerink and H.A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric m-matrix.
Math. of Comput., 31:148–162, 1977.

Bibliography II

Y. Saad. *Iterative Methods for Sparse Linear Systems*.
SIAM, second edition, 2003.

G.L.G. Sleijpen and H.A. van der Vorst. Maintaining convergence properties of BiCGstab methods in finite precision arithmetic.
Numerical Algorithms, 10:203–223, 1995.

P. Sonneveld. CGS, a fast lanczos-type solver for non symmetric systems.
SIAM J. Sci. Statist. Comput., 20:36–52, 1989.

P. Sonneveld and M.B. van Gijzen. IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations.
SIAM J. Sci. Comput., 31(2):1035–1062, 2008.

R.M. Sorensen. *Basic Coastal Engineering*.
Chapman and Hall, second edition, 1997.

H.N. Southgate. Review of wave breaking in shallow water.
Society for Underwater Technology, Wave Kinematics and Environmental Forces, Volume 29, 1993.

A. van der Ploeg. *Preconditioning for sparse matrices with applications*.
PhD thesis, Rijksuniversiteit Groningen, 1994.

H.A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems.
SIAM J. Sci. Statist. Comput., Volume 13:631 – 644, 1992.

H.A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems*.
Cambridge University Press, first edition, 2003.

Bibliography III

M.B. van Gijzen, Y.A. Erlangga, and C. Vuik. Spectral analysis of the discrete helmholtz operator preconditioned with a shifted laplacian.
SIAM J. Sci. Comput., 29(5), 2007.

M.B. van Gijzen and P. Sonneveld. An elegant $\mathrm{IDR}(s)$ variant that efficiently exploits bi-orthogonality properties.
Technical Report 10-16, Department of Applied Mathematical Analysis, 2010.

J. van Kan, A. Segal, and F. Vermolen. *Numerical methods in Scientific Computing*.
VSSD, improved edition, 2008.

P.J. Visser. A mathematical model of uniform longshore currents and the comparison with laboratory data.
Communications on hydraulics 84-2, 1984.

O.C. Zienkewicz. *Finite element method in Engineering Science*.
MC Graw-Hill, 1971.