The all-source Green's function (ASGF) and its applications to storm surge modeling, part II: from the ASGF convolution to forcing data compression and a regression model

Zhigang Xu¹

Received: 30 October 2014 / Accepted: 29 September 2015 / Published online: 16 November 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract This study first validates the ASGS algorithm developed in part I with an analytical solution in a simplified dynamical system and with a real storm surge event. It then assesses the computational efficiency by the ASGF method compared to the traditional method. By analyzing a realistic case, the ASGF method is shown to be three orders of magnitude more computationally efficient than the traditional method. Using the singular value decomposition (SVD) and the fast Fourier transform and its inverse (FFT/IFFT), this study further demonstrates how to compress atmospheric forcing data and how to cast the ASGF convolution as a simple and efficient regression model for data assimilation. When tested with the real storm surge event, the output from the regression model can account for 98 % of the observed variance.

Keywords ASGF convolution · SVD · FFT/IFFT · Linear regression · Storm surges · Data assimilation

1 Introduction

In part I of this study, Xu (2015) proposed a new method of modeling storm surges called the ASGF method, which uses

the all-source Green's function (ASGF) as its core technique. The ASGF is a matrix pre-calculated from a numerical storm surge model. Each column of the ASGF matrix is a Green's function that corresponds to an impulse at a grid point; there are as many such columns as the total number of model grid points. As shown in Eq. (45) of Appendix 3 in part I (copied here with the subscripts and tilde sign omitted), a complex storm surge model can be described as a simple convolution:

$$\boldsymbol{\eta} = \mathbf{G} * \mathbf{f} \quad (1)$$

where $\mathbf{G}$ is an ASGF matrix prepared for a point of interest (POI), $\mathbf{f}$ is a time-variant globally distributed atmospheric forcing vector, and $\boldsymbol{\eta}$ is the solution to the time series of the sea surface elevation at the POI. Part I also interpreted the meaning of the ASGF from different perspectives. It also introduced the memory time scale of the ocean to remember past storm surges and the sampling rate for the ASGF (cf. Sections 3.2 and 3.3 in part I).

This paper is the second part of this study and will first test the ASGF convolution with a simple case and then with a realistic case. It will then assess the computational efficiency of the ASGF method compared to the traditional method. Then, a regression model based on the ASGF convolution will be developed for data assimilation. The simple case test will be performed in a non-rotational, rectangular basin with a flat bottom. The analytical solution to the wave motions in such a basin is obtainable and can be used to verify the ASGF convolution. Section 2.1 will focus on this simple case test. The realistic case test will be presented in Section 2.2, where real storm surges observed at Sept-Îles in Quebec, Canada, in the Gulf of St. Lawrence in December 2010 will be used to test the performance of the ASGF convolution. The analysis of the computational efficiency of the ASGF method will be presented in Section 3.

Responsible Editor: Kevin Lamb

This article is part of the Topical Collection on the *6th International Workshop on Modeling the Ocean (IWMO) in Halifax, Nova Scotia, Canada 23-27 June 2014*

✉ Zhigang Xu
Zhigang.Xu@dfo-mpo.gc.ca

¹ Fisheries and Oceans Canada, Maurice Lamontagne Institute, Mont-Joli, Quebec, Canada

To obtain a regression model, this study uses the singular value decomposition (SVD) to decompose the ASGF convolution matrix. The application of SVD provides two benefits: the atmospheric forcing field can be significantly compressed, which in turn can facilitate data storage and speed up data retrieval and computations, and the singular values from the SVD can be used as the regression parameters to best fit the simulations to the observations. These points are described in detail in Sections 4 and 5. This study will also demonstrate how the fast Fourier transform and its inverse (FFT/IFFT) can be used to speed up the ASGF convolution. Three appendixes provide complementary details and a MATLAB function that is used to compute the ASGF convolution.

2 Testing the ASGF algorithm

This section tests the ASGF algorithm in two ways: validating the algorithm with a simple problem for which an analytical solution is available as a standard for comparison, and testing the ASGF algorithm with a real storm surge event to see how well the predictions by the algorithm agree with the observations. As we will see soon, the algorithm passes both tests very well.

2.1 Test in a simplified dynamical system

This subsection validates the ADI and ASGF numerical approaches with an analytical solution in a rectangular, non-rotational, and flat-bottom basin. Appendix 1 details how to obtain the analytical solution driven by an arbitrary atmospheric forcing field and by a constant west wind. Equation (A34) in the appendix shows how the sea surface evolves under the constant and uniform west wind; it is used in this study for comparison with the numerical solution.

To obtain the numerical solution to the same test problem, the Arakawa C-grid (Arakawa and Lamb 1977) is placed over the rectangular basin: let $a=b=100\text{ km}$ be the sizes of the rectangular basin, and let $\Delta x=\Delta y=1\text{ km}$ be the grid spacing. The abovementioned constant and uniformly wind stresses are applied to the velocity points of the grid. Additionally, the real water depth at Sept-Îles, $h=41\text{ m}$, is set to be the depth of the basin. The friction parameter is set to $\kappa=0.0028\text{ m/s}$. (cf. Eqs. (A1) to (A5)).

The above specifications completely determine the numerical solution. Snapshots of the solution at four times are presented in the left panels of Fig. 1, together with their analytical counterparts on the right panels. The numerical solution was obtained using Eq. (10) in part I with the ADI scheme, where a 10-sec time step was used. The maximum domain-wise absolute differences of the two solutions are noted in the left panels. As shown, the numerical solution is very close to the analytical solution, differing by approximately 1 mm, which is

approximately 1 % of the relative error. The top three panels on the left and right both show how the constant west wind results in the sea level dropping down at the west coast and piling up at the east coast within the first 30 min. Both the left and right bottom panels show that the sea level appears to be a plane with a constant slope over 1 day (i.e., 1440 min). The constant slope was developed through a series of decaying waves, which are clearly shown in the next figure.

The ASGFs used in this study are derived from Eq. (10), the correctness of which hence needs to be verified first. The comparison shown in Fig. 1 gives a satisfactory test result. However, it would not be adequate if this concluded the comparison because there is a significant amount of algebra involved from Eq. (10) to the ASGF convolution. In the comparisons shown in Figs. 2 and 3 below, the numerical solutions are all obtained using the ASGF convolution, Eq. (1). Figure 2 consists of four panels, where panels A and B show the time series of the sea levels near the east and west coasts (i.e., half grid size away from the coasts) for the first 12 hours, and panels C and D show the same time series but over 72 hours, which all demonstrate the closeness between the numerical solution in green and the analytical solution in red (the red is almost completely masked by the green). The maximum of the absolute and relative errors of the numerical solutions for the entire length of the time series is shown on the panels. The maximum absolute error is 4.23 mm, and the relative error is 1 %. Figure 3 shows the sea levels after 3 days from the west to the east coast: The straight line in red is the analytical solution given by Eq. (A36), and the two green “+” symbols are the outputs of the ASGF convolution. As shown, the two green “+” symbols lie perfectly on the theoretical line. The value of the sea-surface slope calculated based on the two green

“+” symbols and the value of the theoretical asymptotic slope are denoted as “slope” and π/gh , respectively, in the panel. The two values are shown to be very close, with a difference of 7.7562×10^{-10} .

2.2 Test with a real storm surge event

The ASGF algorithm is now tested with a real storm surge case. Between the 6th and 7th of December 2010, a large storm moved over the Estuary and Gulf of St. Lawrence. Figure 4a shows a snapshot of the air pressures at the mean sea level based on the Modern-Era Retrospective Analysis for Research and Applications (MERRA) data. The storm caused a large surge in the sea level that damaged coastal highways and many residential properties (Fig. 4b, c). For this test, the tidal gauge station at Sept-Îles, which is operated by the Canadian Hydrographic Service, is used as the POI (Fig. 4d). The ASGF matrix $\mathbf{G}$ for this POI has been calculated and is shown in part I (see Section 4.1 and Fig. 3 in part I). The MERRA dataset is used to supply the atmospheric forcing field. This dataset was produced by NASA and is available

Fig. 1 Comparison of the sea-surface elevations, driven by the west wind, given by the numerical solution (the *left panels*) with those given by the analytical solution at four times indicated in the titles of the panels. The maximum of the absolute differences between the numerical and analytical solutions are noted in each of the *left panels* in units of mm

at <http://gmao.gsfc.nasa.gov/merra/>. It is a re-analyzed dataset that uses NASA's global data assimilation system and a variety of global observing systems and is meant to provide the science and applications communities with a state-of-the-art global dataset. Its temporal resolution is hourly, its spatial resolution is 0.50 degrees of latitude and 0.67 degrees of longitude, and it covers a time period from 1979 to the present. This dataset is adopted in this study because of its highly realistic solutions, its fine temporal resolution, and its global coverage. For any hour, the MERRA dataset provides a forcing vector of 408,622 elements consisting of the points of air pressures and wind stresses in the global ocean.

Equation (1) should be computed through the fast Fourier transform (FFT) and its inverse (IFFT) because the computational efficiency will be significantly enhanced thereby. Computing a convolution through FFT/IFFT is known as the convolution theorem or the convolution rule (Strang 1986). Equation (B16) in Appendix 2.B4 shows how the theorem can be applied to matrices; Appendix 2.B5 shows how the theorem is implemented in a MATLAB function, `conv_FFT`. Figure 5 compares the observation and the simulation; the red curve is the simulation computed using the function `conv_FFT`. The figure shows that the observed surge that peaked at 0 hours on 7 December is well captured by the simulation. The overall agreement between the observation and simulation for the entire

Fig. 2 Comparison of the solutions obtained by the analytical method and the numerical ASGF method: Panels **a** to **d** show the time series of the sea levels at two points near the west and east coasts, where panels **a** and **b** are zoomed-in views of **c** and **d**. The analytical curves are shown in

red, and the ASGF curves are shown in green; the red curves are almost completely masked by the green curves. The maximum absolute and relative differences between the versions of the solutions are denoted by: $\max|er|/\max|\eta|$

simulation period (441 hours) is also good. γ^2 , which is defined below, quantitatively measures the overall misfit:

$$\gamma^2 = \frac{\text{sum of squares of misfits}}{\text{sum of squares of observations}}. \quad (2)$$

A smaller value of this ratio indicates a better agreement. The value of γ^2 shown in the title of the figure is 0.18, which means that 82 % of the observed variance is accounted for by the simulation. The agreement is very satisfactory, considering that no data assimilation technique has been applied yet. Section 5

Fig. 3 Asymptotic sea surface by the analytical solution (in red) and by the ASGF method, which is denoted by the two green “+” symbols. The slope of the sea surface based on the two green “+” symbols is denoted as the slope, whose value is very near to the analytical value given by τ/gh

Fig. 4 **a** Air pressure at mean sea level. **b, c** Damage to the coastal highway and residential properties caused by the storm surge. **d** Location of Sept-Îles, which was used in the simulation of the storm

will introduce a data assimilation technique, with which the value of γ^2 will be further reduced to 0.02. The effects of global forcing and the global ocean geometry have been accounted for in the simulation by the ASGF.

3 Analysis of the computational efficiency gain for the forced wave problem

This section analyzes the computational efficiency gained by the ASGF method compared to the traditional method. The number of multiplications involved in each of the methods to complete a given simulation task will be used to indicate their computational efficiency; the method that involves fewer multiplications is a more efficient method. To use the number of multiplications as a measure of computational efficiency is a standard approach; for example, the efficiency of the FFT algorithm is assessed by considering how many fewer multiplications it requires than the discrete Fourier transform (DFT) algorithm. Addition and subtraction operations are not

counted because they are far less computationally expensive than multiplications. A division can be treated as a multiplication of the inverse of the divisor.

Using $\#\text{trad}$ and $\#\text{asgf}$ to denote the number of multiplications required by the traditional and ASGF methods respectively, we can define the following measure to quantify the computational gain by the ASGF method:

$$\text{gain} = \frac{\#\text{trad}}{\#\text{asgf}} - 1. \quad (3)$$

When $\text{gain} > 0$ or $\text{gain} < 0$, the ASGF method is more or less efficient, respectively, than the traditional method. When $\text{gain} = 0$, the efficiencies of the two methods are equal.

As noted at the end of Section 2 in part I, it is appropriate to view Eq. (10) of part I as a representative of the traditional method. The number of multiplications per time step required by Eq. (10) of part I can be generally expressed as $N \times m$, where N indicates the number the elements of the state vector and m indicates the number of multiplications needed to update each of the elements; the latter value depends on the

Fig. 5 Comparisons of the storm surge simulations (in red) with observations (in black) at Sept-Îles for part of December 2010. The value of γ^2 shown in the title indicates the overall misfit between the observations and simulations. There are 441 hourly data points in each time series

actual difference scheme involved in Eq. (10) of part I. Appendix 5 in part I analyzes the ADI scheme (Leendertse 1967) and finds that $m=12$. Although not shown in Appendix 5 in part I, it could be easily found that $m=3$ for Sielecki's (1968) explicit-implicit (EI) scheme. The ADI method requires a higher value of m than the EI scheme; however, the former is unconditionally stable, whereas the latter is only conditionally stable. In the real case shown later, the time step with the ADI scheme can be 60 sec, whereas the time step with the EI scheme can only be 5 sec because it is constrained by the CFL condition. The total number of multiplications involved in a simulation is proportional to m and inversely proportional to the time step; thus, the ADI scheme is preferred to the EI scheme in terms of computational efficiency in addition to its nearly perfect conservation of the total energy, as discussed in Appendix 1 of part I.

Using T as the total simulation time and Δt_{trad} as the time step used by the traditional method, the number of time steps is $T/\Delta t_{\text{trad}}$. The total number of the multiplications involved can then be calculated by:

$$^{**}\text{trad} = (T/\Delta t_{\text{trad}})Nm. \quad (4)$$

To perform the same simulation, the ASGF method uses Eq. (B16) in Appendix 2.B3, which uses the FFT/IFFT to quickly perform the ASGF convolution. The number of multiplications required by Eq. (B16) is expressed in Eq. (B20) in the appendix and is copied here as $^{**}\text{asgf}$:

$$^{**}\text{asgf} = [(0.5 + 0.5/p + 0.5/n)\log_2 L_{\text{opt}} + 1]nT/\Delta t_{\text{smp}} \quad (5)$$

where n indicates the number of columns of the matrix $\mathbf{G}$ (which is the same as that of $\mathbf{F}$), Δt_{smp} is the sampling rate used to prepare the matrix $\mathbf{G}$ (cf. Section 3.3 of part I), and L_{opt} is an optimal piecewise convolution length, which is introduced and discussed in detail in Appendix 2.B4. Substituting Eqs. (4) and (5) into Eq. (3), we obtain

$$\text{gain} = \left(\frac{\Delta t_{\text{smp}}}{\Delta t_{\text{trad}}} \right) \left(\frac{N}{n} \right) \left(\frac{m}{(0.5 + 0.5/p + 0.5/n)\log_2 L_{\text{opt}} + 1} \right) - 1 \quad (6)$$

$$= \left(\frac{3600}{60} \right) \left(\frac{32,224,425}{408,622} \right) \left(\frac{12}{7.75} \right) - 1 \approx 7.3255 \times 10^3 \quad (7)$$

where the specific numbers pertaining to the realistic case shown in Section 2.2 and consequently $p=2$ have been substituted in the second step. Equation (7) shows that the ASGF method is 7.3255×10^3 times more efficient than the traditional method for the realistic case. The gain is three orders of magnitude and is contributed by the following three factors.

The first contributing factor is $\Delta t_{\text{smp}}/\Delta t_{\text{trad}}$, which is the ratio of the ASGF sampling rate to the time step used by the traditional model. The ASGF sampling rate, Δt_{smp} , can be arbitrarily large and is chosen to be the same as the time resolution of a given atmospheric forcing field, which is 3600 sec for the MERRA forcing field. The choice of the sampling rate does not affect the accuracy of the solution; it only matters how frequently we sample the model solution (cf. Section 3.3 of part I). Conversely, the choice of Δt_{trad} cannot be as arbitrary. Although the ADI scheme may be

unconditionally stable, the choice of the time step must also consider the accuracy of the model solution; a larger time step produces a less accurate model solution. For the real storm surge case presented in the previous section, $\Delta t_{\text{trad}}=60\text{sec}$ was chosen and is already 12 times as large as the CFL condition time step (i.e., 5 sec) that must be used if the difference scheme was the EI scheme instead of the ADI scheme. Thus, the ratio $\Delta t_{\text{sm}}/\Delta t_{\text{trad}}$ contributes a factor of 60 to the gain.

The second contributing factor is a ratio of the number of grid points in the storm surge model to that of the atmospheric model, N/n . In the above example, $N=32,224,425$ and $n=408,622$; their ratio contributes a factor of 78.86 to the gain. An atmospheric model usually has a coarser grid than does an ocean model. The traditional model must spread the coarse-gridded forcing data onto the more-finely gridded ocean model. The ASGF method does the opposite: it folds the more-finely gridded ocean model to fit to the coarse-gridded atmospheric field (see Appendix 3 in part I for details).

The third contributing factor is $m/[(0.5+0.5/p+0.5/n)\log_2 L_{\text{opt}}+1]$, of which the numerator is the number of multiplications per time step required by the traditional model, and the denominator is the number of multiplications per time step required by the ASGF convolution via the FFT/IFFT. The ADI scheme makes the numerator $m=12$, and the value of the denominator is 7.75 when $n=408,622$, $p=2$, and $\log_2 L_{\text{opt}}=9$ (Appendix 2.B3 states that the optimal piecewise convolution length L_{opt} is 2^9 when $L_G=72$); thus, the third ratio contributes a factor of 1.5 to the gain.

The gain expressed by Eqs. (6) or (7) is theoretical. We can also obtain an empirical gain by recording how much computer time is required by the ASGF and the traditional methods to perform simulations with the same length. To perform the simulation shown in Fig. 5, the ASGF method required 10.6036 sec^1 using conv_FFT in Appendix 2.B5. To perform the same simulation, the traditional method using the ADI scheme required $9.2210\times 10^4\text{sec}$ when $\Delta t=60\text{sec}$ and the same grid was used. The ratio of the latter to the former is 8.6961×10^3 , which is close to the theoretical gain obtained by Eq. (7). Both the empirical and theoretical gains indicate that the ASGF method is three orders of magnitude faster than the traditional method.

4 SVD compression of the forcing field

The convolution matrix $\mathbf{G}$ is generally short and wide. The sizes of $\mathbf{G}$ for the real case example used above are $72\times 408,622$; this large aspect ratio implies that there is a large null space that can be squeezed out for the sake of computational efficiency and data assimilation. The singular value

decomposition (SVD, e.g., Strang 2007) is the method to find such a null space. Applying the SVD to the example $\mathbf{G}$ matrix results in:

$$\mathbf{G}_{72\times 408622} = \mathbf{U}_{72\times 72} \begin{bmatrix} \mathbf{S}_{72\times 72} & \mathbf{0}_{72\times 408550} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T_{72\times 408622} \\ \mathbf{V}_2^T_{408550\times 408622} \end{bmatrix} \quad (8)$$

where $\mathbf{U}$ and $[\mathbf{V}_1 \ \mathbf{V}_2]$ are unitary matrices (i.e., $\mathbf{U}^T\mathbf{U}=\mathbf{U}\mathbf{U}^T=\mathbf{I}$ and $[\mathbf{V}_1 \ \mathbf{V}_2]^T[\mathbf{V}_1 \ \mathbf{V}_2]=[\mathbf{V}_1 \ \mathbf{V}_2][\mathbf{V}_1 \ \mathbf{V}_2]^T=\mathbf{I}$, in which each $\mathbf{I}$ is an identity matrix with its appropriate dimensions understood), and $\mathbf{S}$ is a diagonal matrix with non-negative real numbers on the diagonal, which are known as the singular values of $\mathbf{G}$. Next to $\mathbf{S}$ in the middle is a zero matrix, which makes $\mathbf{V}_2$ contribute null to $\mathbf{G}$. Discarding $\mathbf{V}_2$, the decomposition reduces to

$$\mathbf{G}_{72\times 408550} = \mathbf{U}_{72\times 72} \mathbf{S}_{72\times 72} \mathbf{V}_1^T_{72\times 408550}. \quad (9)$$

The singular values are arranged in descending order. They regulate the importance of the input modes. Figure 6 shows the singular values of $\mathbf{G}$ for Sept-Îles; values larger than one (there are 39 of them) amplify the effects of the corresponding forcing components, whereas those less than one (there are 33 of them) reduce the effects. The figure also indicates a ratio of the last singular value versus the first singular value, which is 0.0019 and means that the significance of mode 72 is less than 0.2 % of that of mode 1. In retrospect, this may justify the choice of 72 hours as the memory time scale for calculating the matrix $\mathbf{G}$.

Substituting Eq. (9) into Eq. (1) results in:

$$\boldsymbol{\eta} = (\mathbf{U}\mathbf{S}\mathbf{V}_1^T)^* \mathbf{f} \quad (10)$$

$$= (\mathbf{U}\mathbf{S})^* (\mathbf{V}_1^T \mathbf{f}) \quad (11)$$

$$= (\mathbf{U}\mathbf{S})^* \boldsymbol{\psi} \quad (12)$$

where the associative property of the matrices in convolution is used in the second step (see Eq. (B5) in Appendix 2.B2), and a new forcing vector $\boldsymbol{\psi}$ in the third step is defined as:

$$\boldsymbol{\psi}_{m\times 1} = \mathbf{V}_1^T_{m\times n} \mathbf{f}_{n\times 1}. \quad (13)$$

The new forcing vector is a compressed forcing vector. The uncompressed forcing vector, $\mathbf{f}$, has 408,622 elements, whereas the compressed one, $\boldsymbol{\psi}$, has only 72 elements. The compression rate is 5675 times; thereby, the storage and retrieval of the forcing data are greatly facilitated. For example, the global MERRA atmospheric data from 1979 to 2013 require 400 GB of disk space. The SVD compression reduces the size of this dataset to 72 MB per POI. If we have 1000 POIs, which

¹ All the computations mentioned in the paper are carried out in Dell M 6600, with Intel Core i7-2960XM CPU @ 2.70GHz and 32 GB RAM.

Fig. 6 Singular values of the ASGF matrix, $\mathbf{G}$, at Sept-Îles

is perhaps a quite large number in practice, the total disk space required to store the compressed forcing data is 70 GB, which can be easily stored on a portable computer.

The SVD compression also increases the computational efficiency. The convolution of η using Eq. (12) is much faster than using Eq. (10) because there is much less to convolute after the compression. The convolution theorem discussed in Appendix 2.B4 can be well applied to Eq. (12) to further increase computational efficiency. Using $\mathbf{U}\mathbf{S}$ and $\boldsymbol{\psi}$ as the first and second input to the function `conv_FFT` in Appendix 2.B5, the same simulation shown in Fig. 5 would be completed in 0.0018 sec; this would increase the empirical gain, which was discussed at the end of the last section, from three orders of magnitude to seven orders of magnitude.

The gain in computational efficiency due to the SVD compression of the forcing data was not accounted for in favor of the ASGF method in the last section because one could argue that the data compression itself requires time. However, the gain due to the SVD compression should not be overlooked. Actually, the retrieval of the forcing data from a computer's hard disk takes much more time than does the compression performed within the RAM. Therefore, if 1000 POIs are considered, we can simultaneously compress the same forcing vector $\mathbf{f}$ for these POIs once the forcing vector is loaded into the RAM. The time required for the forcing data compression per POI is thus reduced by a factor of 1000. The abovementioned 34 years of hourly MERRA forcing field data were compressed this way for all of the permanent tidal gauges in eastern Canada. The compressions need to be

performed only once. Then, simulations of storm surges for any period within the 34 years period for any of the POIs can be performed quickly. For example, it takes only half of a second to compute a 10-year-long hourly simulation. Such a high simulation speed can make simulated storm surges instantaneously available for further related studies. We may find that many of our studies are explorative, requiring repetitive simulations in order to determine a best approach or to draw a conclusion. For example, in data assimilation with weighted least-squares regression, we may have to try many simulations with different weighting schemes. Without an instantaneous simulation capability, such explorative studies would be greatly hindered.

5 From the ASGF convolution to a regression model

From Eq. (12), we have:

$$\eta = (\mathbf{U}\mathbf{S}) * \boldsymbol{\psi} \quad (14)$$

which we can transform into:

$$\eta = \mathbf{U} * (\mathbf{S}\boldsymbol{\psi}) \quad (15)$$

using the associative property shown in Eq. (B5) in Appendix 2.B2. The following relationship has been proved in Appendix 3:

$$\mathbf{U} * (\mathbf{S}\boldsymbol{\psi}) = (\mathbf{U} * \boldsymbol{\psi})\mathbf{s} \quad (16)$$

where $\mathbf{S}$ and $\mathbf{s}$ are of the same content but are arranged differently, so are Ψ and ψ . Substituting Eq. (16) into Eq. (15), we have:

$$\eta = \mathbf{C}\mathbf{s} \quad (17)$$

where $\mathbf{C}$ is a matrix given by the column-wise convolution between the matrix $\mathbf{U}$ and Ψ :

$$\mathbf{C} = \mathbf{U} * \Psi \quad (18)$$

which can be computed very quickly with the FFT/IFFT. The MATLAB function `conv_FFT` provided in Appendix B5 can be used to evaluate the $\mathbf{C}$ matrix. It takes only one second to evaluate a $\mathbf{C}$ matrix with 10 years of hourly forcing field.

By admitting an error term, ε , which may consist of errors from the observations, the forcing data, and imperfections of the model, we can rewrite Eq. (17) as a regression model:

$$\eta = \mathbf{C}\mathbf{s} + \varepsilon \quad (19)$$

where $\mathbf{s}$ should now be viewed as a vector of regression parameters. These parameters originate from the surge

model based on the SVD of $\mathbf{G}$; they can now be relaxed from the given values to best fit the observations. The least-squares fitting parameter vector, $\hat{\mathbf{s}}$, is given by:

$$\hat{\mathbf{s}} = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \eta_o \quad (20)$$

where η_o is the observed data (i.e., the detided signals). The least-squares fitted solution, $\hat{\eta}$, is given by:

$$\hat{\eta} = \mathbf{C}\hat{\mathbf{s}} \quad (21)$$

(e.g., Strang 1986, 2007; Seber and Lee 2012).

Section 2.2 shows a simulation of storm surges at Sept-Îles for 441 hours, starting from December 1, 2010. The simulation is performed without data assimilation. The γ^2 value, which measures the misfit between the simulations and the observations (see Eq. (2)), is equal to 0.18. Let us now see how much the regression model described in Eq. (19) can further reduce the misfit. The simulation without data assimilation from Fig. 5 is copied in the top panel of Fig. 7. The simulation with data assimilation using Eqs. (19) to (21) is shown on the bottom panel of the figure. As shown, the data assimilation significantly improves the fitting between the simulation (in red) and the observations (in black). The γ^2 value is reduced to 0.02, which indicates that 98 % of the observed variance is accounted for by the simulation.

Fig. 7 Comparisons of the simulations (in red) of storm surges against the observations (in black) at Sept-Îles for part of December 2010. The simulation shown in the *top panel* is obtained without data assimilation, whereas the simulation on the *bottom panel* is obtained with data assimilation. The value of γ^2 shown in each panel measures the overall misfits between the observation and the simulations. There are 441 hourly data points in the time series

6 Summary and discussions

This paper is the second part of this study. Starting with the ASGF convolution of Eq. (1) developed in part I, this paper first validated the ASGF algorithm with an analytical solution obtained from a simplified dynamical system: waves in a non-rotational, rectangular basin with a flat bottom. The validation produced excellent results (Figs. 1, 2 and 3). The ASGF algorithm was then further tested with a real storm surge event that occurred in Sept-Îles during December 2010. The test yielded a very satisfactory result: 82 % of the observed variance was explained by the simulation (Fig. 5), which also indicates that the linear and depth-averaged shallow water dynamics is indeed a good basis for developing the ASGF method.

The gain in the computational efficiency due to the ASGF method compared to the traditional method was theoretically analyzed and empirically estimated. Both the theoretical and empirical gains indicate that the ASGF method is three orders of magnitude faster than the traditional method in producing the same simulation shown in Fig. 5. There are three factors that make the ASGF method perform better than the traditional method. First, to produce an hourly time series, the ASGF method can use an hourly time step without affecting the accuracy of the simulation (cf. the sampling rate discussed in Section 3.2 of part I), whereas the traditional method has to use a much smaller time step for the sake of computational stability or accuracy. Second, the spatial resolution in a given atmospheric forcing field is usually much coarser than the grid spacing of a surge model in the ocean; consequently, the atmospheric forcing vector is much shorter than the surge state vector. The ASGF method takes advantage of the difference in the two spatial resolutions by compressing the columns of its convolution matrix to fit to the shorter forcing vector; conversely, the traditional method must stretch the forcing vector to fit to the longer state vector. Third, the ASGF method can use the FFT/IFFT to speed up the computation, whereas the traditional method cannot.

Another advantage of having cast a traditional storm surge model into an ASGF convolution is that the SVD can be applied to the convolution matrix. Using the SVD, the forcing data can be greatly compressed. The real case example shows that the compression rate is 5675 times; thus, a long forcing vector with 408,622 elements can be replaced by a shorter vector composed of only 72 elements. The compression significantly facilitates the storage and retrieval of the forcing data and makes the convolution performed more rapidly, allowing for long-term storm surge simulations to be finished within seconds.

The most important benefit of the application of SVD is data assimilation. With the SVD, the ASGF convolution, Eq. (1), is cast into a regression model, Eq. (19). The singular values can be moved from the middle position to the right position so that it can be relaxed from being given by the original surge model to the one best adjusted by the observed

data. The power of the regression model to fit the simulations to the observations is well demonstrated. Without data assimilation, the overall misfit between the observations and the simulations is 0.18. With the data assimilation, the overall misfit is decreased to 0.02 (cf. Fig. 7).

The main point of this study is that a complicated and time-consuming storm surge model is cast into a simple and efficient regression model, with which we can easily conduct various regression analyses. The data assimilation demonstrated in the last section is not the only way; for example, a weighting matrix could be multiplied to both sides of Eq. (19) to allow for the large data to have more weight in determining the least-squares solutions. Large data are more likely caused by storms than are small data.

Due to its fast simulation speed and data assimilation capability, the ASGF regression model shown in Eq. (19) provides an effective tool for long-term hindcasts and forecasts of storm surges. With past observations and realistic atmospheric forcing data, a hindcast can yield the best estimated model parameter vector $\hat{s}$. With the best estimated $\hat{s}$ and a set of climate model solutions for the future that can be used as the atmospheric forcing, we can climatologically forecast storm surges. Xu et al. (2015) has taken this approach and produced storm surges at Sept-Îles over the next hundred years, which can serve as a database for further statistical analyses such as extreme value analyses.

Acknowledgments This study was partially supported by le Ministère des Transports du Québec and by the ACCASP program of Department of Fisheries and Oceans. This study also benefited from collaboration with the OURANOS Consortium². Free access to the MERRA data provided by NASA is also significantly appreciated. The author also gratefully acknowledges the support from his own institute.

Appendix 1: An analytical solution in a simplified dynamical system

This section verifies the numerical approach presented in this study with an analytical solution in a simplified dynamical system. A rectangular, non-rotational, and flat-bottom basin is considered. This simple setting permits an analytical solution. For this case, the governing equations of Eq. (1) in part I simplifies to:

$$\frac{\partial \eta}{\partial t} = -\frac{\partial U}{\partial x} - \frac{\partial V}{\partial y} \quad (\text{A1})$$

$$\frac{\partial U}{\partial t} = -gh \frac{\partial \eta}{\partial x} - \frac{\kappa}{h} U + gh \frac{\partial \eta_a}{\partial x} + \tau_x \quad (\text{A2})$$

$$\frac{\partial V}{\partial t} = -gh \frac{\partial \eta}{\partial y} - \frac{\kappa}{h} V + gh \frac{\partial \eta_a}{\partial y} + \tau_y \quad (\text{A3})$$

² <http://www.ouranos.ca/>

$$U = 0 \quad \text{at } x=0, x=a \quad (\text{A4})$$

$$V = 0 \quad \text{at } y=0, y=b \quad (\text{A5})$$

where x and y are horizontal Cartesian coordinates along the east and north directions, respectively, and the other notations are the same as those used in Eq. (1) of part I. The lateral boundary conditions are zero fluxes at $x=(0,a)$ and $y=(0,b)$, and the initial values for η , U and V are all assumed to be zero.

We can eliminate U and V in favor of η :

$$\frac{\partial^2 \eta}{\partial t^2} = gh \left(\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} \right) - \frac{\kappa}{h} \frac{\partial \eta}{\partial t} + F(t, x, y) \quad (\text{A6})$$

$$\frac{\partial \eta}{\partial x} = 0, \quad \text{at } x=0 \text{ and } a, \quad (\text{A7})$$

$$\frac{\partial \eta}{\partial y} = 0, \quad \text{at } y=0 \text{ and } b, \quad (\text{A8})$$

$$\eta = 0, \quad t=0 \quad (\text{A9})$$

$$\frac{\partial \eta}{\partial t} = 0, \quad t=0 \quad (\text{A10})$$

$$\int_0^a \int_0^b \frac{\partial \eta}{\partial t} dx dy = 0 \quad (\text{for all } t) \quad (\text{A11})$$

where:

$$F = -gh \left(\frac{\partial^2 \eta_a}{\partial x^2} + \frac{\partial^2 \eta_a}{\partial y^2} \right) - \left(\frac{\partial \tau_x}{\partial x} + \frac{\partial \tau_y}{\partial y} \right) \quad (\text{A12})$$

which represents the atmospheric forcing field. The lateral boundary conditions, namely Eqs. (A7) and (A8), are a consequence of the zero normal fluxes and zero external forcing at the solid vertical coast³.

Condition (A11) is a result of the domain-wise integration of the continuity equation⁴ and ensures that mass is conserved globally. From a mathematical perspective, a governing equation with the second-order time derivative needs only two initial conditions, as described by Eqs. (A9) and (A10). However, the solution so obtained may be mathematically valid but not physical. Using Eq. (A11) as an additional constraint helps eliminate any unphysical solutions. Consistently, the domain-wise integral of the forcing field must also vanish:

$$\int_0^a \int_0^b F(t, x, y) dx dy = 0 \quad (\text{A13})$$

³ For a solid vertical coast, applying a zero or non-zero horizontal force makes no difference if we are concerned with only water movement. However, the former choice makes the boundary conditions homogeneous, which will in turn make the solution much easier to obtain.

⁴ Using the divergence theorem of Gauss and the zero fluxes at the coasts, the domain-wise integral of the RHS of (A1) is zero.

which can be verified by the domain-wise integration of Eq. (A6) using Eqs. (A7), (A8), and (A11).

Equation (A6) is linear and inhomogeneous. Its solution can be obtained with Duhamel's principle (e.g., Courant and Hilbert 1962) and the linear superposition principle. With Duhamel's principle, the inhomogeneous equation can be modified to become homogeneous. With the linear superposition principle, the homogeneous solutions can be superimposed to recover the inhomogeneous solution. The solution to Eq. (A6) is:

$$\eta(t, x, y) = \int_0^t \xi(t-t', x, y) dt' \quad (\text{A14})$$

where ξ is the solution of the following homogenous equation:

$$\frac{\partial^2 \xi}{\partial t^2} = gh \left(\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} \right) - \frac{\kappa}{h} \frac{\partial \xi}{\partial t}, \quad (\text{A15})$$

$$\frac{\partial \xi}{\partial x} = 0, \quad \text{at } x=0 \text{ and } a, \quad (\text{A16})$$

$$\frac{\partial \xi}{\partial y} = 0, \quad \text{at } y=0 \text{ and } b, \quad (\text{A17})$$

$$\xi = 0, \quad t=t', \quad (\text{A18})$$

$$\frac{\partial \xi}{\partial t} = F(t', x_j, y_i), \quad t=t', \quad (x_j, y_i) \in [a, b]. \quad (\text{A19})$$

Using the technique of separation of variables (e.g., Myint-U and Debnath 2007), ξ can be expressed as:

$$\xi = \begin{cases} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} e^{-s(t-t')} B_{mn} \sin \omega_{mn} (t-t') \cos \frac{m\pi y}{b} \cos \frac{n\pi x}{a}, & t \geq t' \\ 0, & t < t' \end{cases} \quad (\text{A20})$$

where s is a frictional parameter, ω_{mn} are the wave frequencies, and B_{mn} are the Fourier series coefficients; these values are given below:

$$s = \frac{\kappa}{2h} \quad (\text{A21})$$

$$\omega_{00} = 0 \quad (\text{A22})$$

$$\omega_{mn} = \sqrt{gh\pi^2 \left(\frac{n^2}{a^2} + \frac{m^2}{b^2} \right) - s^2} \quad (m=0, 1, 2, \dots, \infty, n=0, 1, 2, \dots, \infty, \text{ exclude } m=n=0) \quad (\text{A23})$$

$$B_{00} = 0 \quad (\text{A24})$$

$$B_{m0} = \frac{2}{ab\omega_{m0}} \int_0^b \int_0^a F(t, x, y) \cos \frac{m\pi y}{b} dx dy \quad (\text{A25})$$

$$B_{0n} = \frac{2}{ab\omega_{0n}} \int_0^b \int_0^a F(t, x, y) \cos \frac{n\pi x}{a} dx dy \quad (\text{A26})$$

$$B_{mn} = \frac{4}{ab\omega_{mn}} \int_0^b \int_0^a F(t, x, y) \cos \frac{n\pi x}{a} \cos \frac{m\pi y}{b} dx dy \quad (\text{A27})$$

where $0 \leq \tau \leq t$, $m=1, 2, \dots$ and $n=1, 2, \dots$. The value of the frictional parameter $s=\kappa/(2h)$ should be chosen such that all ω_{nm} are real valued.

The time convolution expressed by Eq. (A14), together with Eqs. (A20) through (A27), provides a general solution to wave motion in the rectangular domain driven by any type of forcing field, provided that the condition stated by Eq. (A13) holds. As a simple example, consider a forcing field associated with the following wind and air pressure fields:

$$\left. \begin{aligned} U_{10}(x, y) &= \begin{cases} 20 \text{ m/s}, & \text{for } x \in (0, a), \text{ any } y, \\ 0 \text{ m/s}, & x = 0, x = a, \text{ any } y, \end{cases} \\ V_{10}(x, y) &= 0, \\ p_a(x, y) &= \text{constant}. \end{aligned} \right\} \quad (\text{A28})$$

In this case, the forcing field expressed by Eq. (A12) is reduced to:

$$F = -\frac{\partial \tau_x(t, x, y)}{\partial x} \quad (\text{A29})$$

where τ_x can be calculated from U_{10} and V_{10} with Eqs. (3) and (4) in part I.

The U_{10} specified by Eq. (A28) represents a constant west wind uniformly blowing east within the domain at a speed of 20 m/s. However, the wind speeds at the two boundary ends (i.e., $x=0$ and $x=a$) are set equal to zero; this is because a non-zero wind cannot produce any effect on a vertical concrete wall anyway. The given wind speed is therefore a step function across the solid boundary, so is the associated wind stress. Specifically, from $x=0$ to $x=\varepsilon$, where ε is any positive infinitesimal, the wind stress increases from 0 to a positive value, and from $x=a-\varepsilon$ to $x=a$, it decreases from the same positive value to 0. Therefore, based on Eq. (A29), the forcing field F consists of two Dirac δ - functions:

$$F = \tau \delta(x-a) - \tau \delta(x) \quad (\text{A30})$$

where τ is a constant stress ($\tau = C_d |U_{10}| U_{10} = 2.8 \times 10^{-3} \times 20^2 = 1.12 \text{ m}^2/\text{s}^2$). It is evident that the above F respects the condition of Eq. (A13).

Substituting the above F into Eqs. (A25) to (A27), we find that:

$$B_{0n} = \begin{cases} \frac{-4\tau}{a\omega_{0n}} & \text{when } n \text{ is odd} \\ 0 & \text{when } n \text{ is even} \end{cases} \quad (\text{A31})$$

$$B_{m0} = 0, \quad B_{mn} = 0. \quad (\text{A32})$$

The homogeneous solution, Eq. (A20), then becomes:

$$\xi = \begin{cases} \frac{-4\tau}{a} \sum_{n=1}^{\infty} \frac{e^{-s(t-t')}}{\omega_{0n}} \sin \omega_{0n}(t-t') \cos \frac{n\pi x}{a}, & t \geq t' \\ 0, & t < t' \end{cases} \quad (\text{A33})$$

and the inhomogeneous solution, Eq. (A14), can be evaluated as:

$$\eta = -\frac{4\tau a}{gh\pi^2} \sum_{n=1,3,5,\dots}^{\infty} \frac{1-e^{-st}(\cos \omega_{0n}t + s/\omega_{0n} \sin \omega_{0n}t)}{n^2} \cos \frac{n\pi x}{a}. \quad (\text{A34})$$

The solution shows that the wave motion is independent of y , which is expected because the external forcing is applied only in the x -direction. The absence of y in the solution indicates that the wave motion in the y -direction is uniform.

The slope of the sea surface can be determined by taking the x -derivative of each term of the above series:

$$\frac{\partial \eta}{\partial x} = \frac{4\tau}{gh\pi} \sum_{n=1,3,5,\dots}^{\infty} \frac{1-e^{-st}(\cos \omega_{0n}t + s/\omega_{0n} \sin \omega_{0n}t)}{n} \sin \frac{n\pi x}{a}. \quad (\text{A35})$$

When $t \rightarrow \infty$, the asymptotic sea surface and its slope become:

$$\eta = -\frac{4\tau a}{gh\pi^2} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n^2} \cos \frac{n\pi x}{a} = \frac{\tau_x}{gh}(x-a/2), \quad \text{for } x \in [0, a] \quad (\text{A36})$$

$$\frac{\partial \eta}{\partial x} = \frac{4\tau}{gh\pi} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n} \sin \frac{n\pi x}{a} = \begin{cases} \frac{\tau}{gh}, & x \in (0, a), \\ 0, & x = 0, x = a. \end{cases} \quad (\text{A37})$$

In the second steps of the above equations, the following two well-known Fourier series have been used:

$$x = a/2 - \frac{4a}{\pi^2} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n^2} \cos \frac{n\pi x}{a} \quad \text{for } x \in [0, a] \quad (\text{A38})$$

$$1 = \frac{4}{\pi} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n} \sin \frac{n\pi x}{a} \quad \text{for } x \in (0, a) \quad (\text{A39})$$

which one can verify or easily find references for⁵.

⁵ For example, for Eq. (A38), http://ocw.mit.edu/courses/mathematics/18-03sc-differential-equations-fall-2011/unit-iii-fourier-series-and-laplace-transform/fourier-series-basics/MIT18_03SCF11_s21_6text.pdf, and for Eq. (A39), <http://mathworld.wolfram.com/FourierSeriesSquareWave.html>.

The asymptotic solutions given by Eqs. (A36) and (A37) can also be obtained directly from the original momentum equations. For the constant wind stress considered here, we can immediately see the asymptotic solution from Eq. (A2): the asymptotic u and its time derivative all vanish due to friction. This leaves only the slope in the sea surface balancing the wind stress as $t \rightarrow \infty$:

$$\frac{\partial \eta}{\partial x} = \frac{\tau}{gh}. \quad (\text{A40})$$

Integrating the above equation, we can determine the asymptotic sea surface as follows:

$$\begin{aligned} \eta &= \frac{\tau}{gh}x + \text{const.} \\ &= \frac{\tau}{gh}(x - a/2) \end{aligned} \quad (\text{A41})$$

where the integral constant is determined as $\text{const} = a/2$ in the second step by the global mass conservation $\int_0^a \eta dx = 0$.

Appendix 2: Convolution of matrices, its definition, and properties

B1: Definition of convolution of matrices

Equation (19) in part I is copied here for convenience after dropping the subscript “c”:

$$(\mathbf{G} * \mathbf{f})^{(k+1)} \equiv \sum_{i=0}^k \mathbf{G}(i+1, :) \mathbf{f}^{(k-i)}, \quad (k = 0, 1, 2, \dots, k_{\max}) \quad (\text{B1})$$

where $\mathbf{f}$ is a forcing vector, which changes in time, as implied by its superscript k . The forcing field consists of the set of the forcing vectors $\mathbf{f}^{(k)}$ ($k=0, 1, 2, \dots, k_{\max}$). Introduce a matrix $\mathbf{F}$ to accommodate the forcing vectors:

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}^{(0)} & \mathbf{f}^{(1)} & \dots & \mathbf{f}^{(k)} & \dots & \mathbf{f}^{(k_{\max})} \end{bmatrix}^T. \quad (\text{B2})$$

A column of $\mathbf{F}$ is a time series of the forcing field at a particular spatial point, and a row of $\mathbf{F}$ is a forcing vector at a particular time. The arrangement of the temporal-spatial information in $\mathbf{F}$ is such that the temporal information is contained in the columns, and the spatial information is contained in the rows. The matrix $\mathbf{G}$ shares the same temporal-spatial information arrangement (cf. Eq. (17) and Section 4.1 of part I). The actual time span corresponding to the length of $\mathbf{F}$ is equal to $L_F \times \Delta t_F$, where Δt_F is the time interval that comes with the atmospheric model solutions, which is usually hourly. The actual time span corresponding to the length of $\mathbf{G}$ is equal to $L_G \times \Delta t_{\text{smp}}$, where Δt_{smp} is the sampling rate used to pre-

calculate the matrix $\mathbf{G}$ (cf. Section 3.3 of part I). For storm surge problems, Δt_{smp} should be chosen to be the same as Δt_F .

The convolution defined by Eq. (B1) can now be re-expressed as:

$$(\mathbf{G} * \mathbf{F})^{(k+1)} \equiv \sum_{i=0}^k \mathbf{G}(i+1, :) \mathbf{F}^T(:, k-i+1), \quad (k = 0, 1, 2, \dots, k_{\max}).$$

Using Eqs. (B2) and (B3), we can re-express Eq. (1) as: (B3)

$$\eta = \mathbf{G} * \mathbf{F}. \quad (\text{B4})$$

Equation (B3) defines a convolution between two matrices. It is an extension from a convolution between two time series (or to say two vectors) because the latter is a special case when there is only one column in $\mathbf{G}$ and in $\mathbf{F}$. Because of this connection, the number of rows in $\mathbf{G}$ and $\mathbf{F}$ will be frequently referred to as the lengths of $\mathbf{G}$ and $\mathbf{F}$ in the text below.

B2: Associative and commutative properties of matrices in convolution

Associative property

If $\mathbf{G} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, the following associative property holds:

$$\mathbf{G} * \mathbf{f} = (\mathbf{U}\mathbf{S}\mathbf{V}^T) * \mathbf{f} = (\mathbf{U}\mathbf{S}) * (\mathbf{V}^T \mathbf{f}) = \mathbf{U} * (\mathbf{S}\mathbf{V}^T \mathbf{f}). \quad (\text{B5})$$

This property can be proved as below:

$$[(\mathbf{U}\mathbf{S}\mathbf{V}^T) * \mathbf{f}]^{(k+1)} = \sum_{i=0}^k (\mathbf{U}(i+1, :) \mathbf{S}\mathbf{V}^T \mathbf{f}^{(k-i)}) \quad (\text{B6})$$

$$= \sum_{i=0}^k (\mathbf{U}(i+1, :) \mathbf{S}) (\mathbf{V}^T \mathbf{f}^{(k-i)}) \equiv [(\mathbf{U}\mathbf{S}) * (\mathbf{V}^T \mathbf{f})]^{(k+1)} \quad (\text{B7})$$

$$\text{or} \sum_{i=0}^k \mathbf{U}(i+1, :) (\mathbf{S}\mathbf{V}^T \mathbf{f}^{(k-i)}) \equiv [\mathbf{U} * (\mathbf{S}\mathbf{V}^T \mathbf{f})]^{(k+1)} \quad (\text{B8})$$

where $k=0, 1, 2, \dots$.

Commutative property

$$\mathbf{G} * \mathbf{F} = \mathbf{F} * \mathbf{G}. \quad (\text{B9})$$

In many textbooks, this property is proved for two vector operands. It also holds when the two operands are matrices. This can be proved as follows:

$$(\mathbf{G} * \mathbf{F})^{(k+1)} \equiv \sum_{i=0}^k \mathbf{G}(i+1, :) \mathbf{F}^T(:, k-i+1) \quad (\text{B10})$$

$$= \sum_{m=0}^k \mathbf{F}(m+1, :) \mathbf{G}^T(:, k-m+1) \quad (\text{B11})$$

$$= (\mathbf{F} * \mathbf{G})^{(k+1)} \quad (\text{B12})$$

for any $k=0, 1, 2, \dots, k_{\max}$, where $m=k-i$ in the second step, and the positions of $\mathbf{G}$ and $\mathbf{F}^T$ are exchanged with their transposes. The property is thus proved.

B3: The convolution length and the number of multiplications

Equation (B4) may appear abstract; it may be worthwhile to explain it with a concrete example. Without loss of generality, let us assume that the matrix $\mathbf{G}$ has only two rows and the matrix $\mathbf{F}$ has three rows, i.e., $\mathbf{G}=[\mathbf{r}_0; \mathbf{r}_1]$ and $\mathbf{F}=[\mathbf{f}^{(0)} \ \mathbf{f}^{(1)} \ \mathbf{f}^{(2)}]^T$. In this case, based on the definition given in Eq. (B3), we can write Eq. (B4) as:

$$\begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{r}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{F}^T(:, 1) + \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{r}_1 \\ \mathbf{r}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{F}^T(:, 2) + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{r}_0 \\ \mathbf{r}_1 \end{bmatrix} \mathbf{F}^T(:, 3) \quad (\text{B13})$$

where η_1, η_2, η_3 and η_4 are the four convolution values at four times ($t=\Delta t, 2\Delta t, 3\Delta t, 4\Delta t$, where Δt may equal one hour, for example). The first term of the right-hand side (RHS) represents a train of waves that is set into motion by the first instantaneous forcing vector, $\mathbf{F}^T(:, 1)$. The second term is another train of waves that is excited by the second instantaneous forcing vector, $\mathbf{F}^T(:, 2)$, and similarly for the third term. The second instantaneous forcing vector does not exist until the second time step; this is why we see that $\mathbf{r}_0$ and $\mathbf{r}_1$ are shifted down by one position in the column vector in the second term. Similarly, the reason we see two zeros before $\mathbf{r}_0$ and $\mathbf{r}_1$ in the column vector of the third term is because the instantaneous forcing vector $\mathbf{F}^T(:, 3)$ does not exist until the third time step. Once set into motion, these wave trains become free waves because their associated instantaneous forces no longer exist. The total response is a sum of these free wave trains.

This simple example assumes that there are only three successive forces and that the wave set up by each force only lasts two time steps; therefore, the length of the total response is four time steps (i.e., $2+3-1=4$). Generally speaking, the length of the convolution response vector is given by the formula:

$$L_R = L_G + L_F - 1 \quad (\text{B14})$$

where L_R is the length of the response vector, L_G is the length of the convolution kernel, and L_F is the length of the forcing vector. As noted at the end of Appendix 2.B1, the length of a matrix is a synonym for the number of rows of the matrix.

The number of multiplications entailed in Eq. (B13) can be analyzed in this way: $\mathbf{G}$ and $\mathbf{F}$ have the same number of columns, and let this number be denoted as n . This means that each row times a column on the RHS of the equation (e.g., $\mathbf{r}_0 \times \mathbf{F}^T(:, 1)$) requires n multiplications. There are two such rows and three such columns; the total number of multiplications is thus $2 \times 3 \times n$. In general, the number of multiplications, $"\#*"_{\text{eqB3}}$, required to perform a convolution using Eq. (B3) is equal to:

$$"\#*"_{\text{eqB3}} = nL_GL_F. \quad (\text{B15})$$

B4: Convolution theorem

The ASGF convolution can also be performed via FFT/IFFT:

$$\mathbf{G} * \mathbf{F} = \text{ifft} \left(\text{sum} \left(\hat{\mathbf{G}}_z * \hat{\mathbf{F}}_z, 2 \right) \right) \quad (\text{B16})$$

where $\hat{\mathbf{G}}_z = \text{fft}(\mathbf{G}_z)$, $\hat{\mathbf{F}}_z = \text{fft}(\mathbf{F}_z)$, $\mathbf{G}_z$ and $\mathbf{F}_z$ are zero-padded versions of $\mathbf{G}$ and $\mathbf{F}$. The symbol “ $*$ ” denotes a dot-multiplication operator that multiplies its two operands element by element; and $\text{sum}(\hat{\mathbf{G}} * \hat{\mathbf{F}}, 2)$ means to sum the product of $\hat{\mathbf{G}} * \hat{\mathbf{F}}$ row-wisely. Note that the result of $\text{sum}(\hat{\mathbf{G}} * \hat{\mathbf{F}}, 2)$ is a one-column vector. Appropriate zero padding is necessary in FFT so that the tail of the resultant vector will not be folded back to its head. Equation (B16) extends the well-known convolution theorem (e.g., Strang 2007) from vectors to matrices.

Computing a convolution via FFT/IFFT is generally much faster than directly computing it according to the convolution definition. Let us examine how many multiplications are required by Eq. (B16). When an FFT or IFFT is applied to a L_R -point vector, it requires $L_R/2 \log_2(L_R)$ multiplications if L_R is a power of 2. In Eq. (B16), there are two FFT's applied to $\mathbf{G}_z$ and $\mathbf{F}_z$, both of which have n columns; this requires $nL_R \log_2(L_R)$ multiplications. There are also element-by-element multiplications between $\hat{\mathbf{G}}$ and $\hat{\mathbf{F}}$, which requires nL_R multiplications; one IFFT is applied to a one-column vector, which is $\text{sum}(\hat{\mathbf{G}} * \hat{\mathbf{F}}, 2)$. This requires $L_R/2 \log_2(L_R)$ multiplications. The total number of multiplication is then given by:

$$"\#*"_{\text{eqB16}} = nL_R \log_2(L_R) + nL_R + L_R/2 \log_2(L_R) = [(1 + 0.5/n) \log_2(L_R) + 1] nL_R. \quad (\text{B17})$$

The ratio of $"\#*"_{\text{eqB3}}$ to $"\#*"_{\text{eqB16}}$ is:

$$\frac{"\#*"_{\text{eqB3}}}{"\#*"_{\text{eqB16}}} = \frac{nL_FL_G}{[(1 + 0.5/n) \log_2(L_R) + 1] nL_R} = \frac{(1 - L_G/L_R + 1/L_R)L_G}{[(1 + 0.5/n) \log_2(L_R) + 1]} \quad (\text{B18})$$

where Eq. (B14) has been used. Figure 8 shows how this ratio varies with $\log_2 L_R$ when $L_G=72$ and $n=408,622$, both of

Fig. 8 Ratio of “#”_{eq80} / “#”_{eq90} varies with $\log_2(L_R)$ when L_G is fixed at 72

which pertain to the realistic example in Section 2.2. The ratio peaks at $\log_2 L_R = 9$, after which it monotonically decreases toward zero. The peak ratio is 6.2, which indicates that the calculation of the same convolution using Eq. (B16) requires 5.2 times fewer multiplications than using Eq. (B3). In other words, Eq. (B16) is 5.2 times faster or more efficient than Eq. (B3). After the peak, the ratio decreases but remains above 3 up to $\log_2 L_R = 20$, which is equivalent to 119.7 years of hourly points. Although not shown in the figure, the ratio changes from above one to below one when $\log_2 L_R$ increases from 41 to 42. This indicates that for all $\log_2 L_R < 42$, Eq. (B16) is more efficient than Eq. (B3). The reason why the convolution theorem does not win for all L_R is because the convolution theorem is applied here to two sets of different length time series, L_G and L_F , where L_G is fixed, and L_F increases as L_R increases. If L_G and L_F were of the same length and if both increased with L_R , the convolution theorem would always win.

The peak ratio occurs at $L_R = 2^9$, which therefore should be considered to be the optimal convolution length. Denoting this length as L_{opt} , we can use the length $L_f = L_{opt} - L_G + 1$ to divide the entire length of the forcing matrix into m pieces using $p = \text{ceil}(L_F / L_f)$, where “ceil” is a function that rounds its argument to the nearest integer toward infinity. For each piece, we can apply the convolution theorem to obtain a response vector; we can then assemble the piecewise response vectors into a final long response vector. Note that $\widehat{\mathbf{G}}_z = \text{fft}(\mathbf{G}_z)$ needs to be computed only once; it can then be used for all of the pieces of the forcing vector (more precisely, the forcing matrix). Denote $\mathbf{F}_z^{(p)}$ as a zero-padded version of the p th piece and still use $\mathbf{G}_z$ to denote a zero-padded version of $\mathbf{G}$ but now only to the length of L_{opt} . Then, the piecewise approach requires the

following numbers of multiplications: to compute $\widehat{\mathbf{G}}_z = \text{fft}(\mathbf{G}_z)$ once requires $nL_{opt}/2\log_2 L_{opt}$ multiplications; to compute $\widehat{\mathbf{F}}_z^{(p)} = \text{fft}(\mathbf{F}_z^{(p)})$ p times requires $pnL_{opt}/2\log_2 L_{opt}$ multiplications; to compute $\widehat{\mathbf{G}}_z * \widehat{\mathbf{F}}_z^{(p)}$ p times requires pnL_{opt} multiplications; and to compute $\text{ifft}(\text{sum}(\widehat{\mathbf{G}}_z * \widehat{\mathbf{F}}_z^{(p)}, 2))$ p times requires $pL_{opt}/2\log_2 L_{opt}$ multiplications. The total number of multiplications entailed in Eq. (B16) with the optimal piece-wise approach is:

$$“\#”_{eqB16} = \left[\frac{1 + 1/p + 1/n}{2} \log_2 L_{opt} + 1 \right] pnL_{opt} \quad (\text{B19})$$

$$= \left[\frac{1 + 1/p + 1/n}{2} \log_2 L_{opt} + 1 \right] nT / \Delta t_{\text{smp}} \quad (\text{B20})$$

where the following relation is used in the second step:

$$T = pL_{opt} \Delta t_{\text{smp}} \quad (\text{B21})$$

where T is the time span of the convolution response, and Δt_{smp} is the sampling rate used to pre-calculate the matrix $\mathbf{G}$ (cf. Section 3.3 of part I), which is chosen to be the same as the time interval in a given atmospheric forcing field.

The ratio of “#”_{eqB3} to the above optimized “#”_{eqB16} is equal to:

$$\frac{“\#”_{eqB3}}{“\#”_{eqB16}} = \left(\frac{L_G}{\frac{1 + 1/p + 1/n}{2} \log_2 L_{opt} + 1} \right) \left(\frac{L_F}{pL_{opt}} \right) \quad (\text{B22})$$

$$\approx \left(\frac{72}{\frac{1+1/p}{2}9+1} \right) \left(\frac{L_F}{2^9 p} \right) \quad (\text{B23})$$

where the specific values of $L_G=72$, $L_{\text{opt}}=2^9$ and $n=408,622$, which are pertinent to the real storm surge case simulated in Section 2.2, have been substituted in the second step. Equation (B22) is general, whereas Eq. (B23) is specific because different values of L_G have different values of L_{opt} .

Table 1 lists the ratios for different values of L_F . As shown, the above divide-and-conquer algorithm significantly increases the ratios. For example, when $L_F=4380 \times 24$, the ratio is 11.2, whereas Fig. 8 shows that the ratio is below 4.2 for the same L_F ($\log_2 L_F=16.6817$).

In the above calculations, $L_G=72$ has been extensively used, which results in the optimal convolution length of $L_{\text{opt}}=2^9$. In general, L_{opt} is a function of L_G and can be solved from:

$$2^y \log_2 e + (1-L_G)y + (1-L_G) \left(\log_2 e + \frac{2n}{2n+1} \right) = 0, \quad (\text{B24})$$

where $y=\log_2(L_{\text{opt}})$. Table 2 lists the solutions for some values of L_G .

B5: MATLAB function conv_FFT

The MATLAB function called conv_FFT, which is given in Table 3, implements Eq. (B16) via the divide-and-conquer approach.

Appendix 3: Position swap between the singular values and the compressed forcing vector

The following relation can be proven:

$$\mathbf{U}^*(\mathbf{S}\boldsymbol{\psi}) = (\mathbf{U}^*\boldsymbol{\Psi})\mathbf{s} \quad (\text{C1})$$

where $\mathbf{S}$ and $\mathbf{s}$ contain the same elements but are arranged differently, so are $\boldsymbol{\Psi}$ and $\boldsymbol{\psi}$. This relation states that the singular values contained in the diagonal matrix $\mathbf{S}$ and the elements in the compressed forcing vector $\boldsymbol{\psi}$ can swap positions

Table 2 Solutions to $y=\log_2 L_{\text{opt}}$ from Eq. (B24) for a set of L_G . The solutions are rounded to the nearest integer

L_G	24	36	48	60	72	84	96	108	120	132
$y=\log_2 L_{\text{opt}}$	7	8	8	8	9	9	10	10	10	10
L_G	144	156	168	180	192	204	216	228	240	264
$y=\log_2 L_{\text{opt}}$	10	10	11	11	11	11	11	11	11	11

after an appropriate re-arrangement of their contents. The appropriate re-arrangement is as follows: $\mathbf{s}$ is a column vector consisting of the diagonal elements of $\mathbf{S}$, i.e., $\mathbf{s}=[s_1 \ s_2 \ \cdots \ s_m]^T$; $\boldsymbol{\psi}_j = [\psi_j^{(0)} \ \psi_j^{(1)} \ \cdots \ \psi_j^{(k_{\text{max}})}]^T$ is a column vector consisting of the time series of ψ_j , with $j=1,2,\cdots,m$; and $\boldsymbol{\Psi}=[\boldsymbol{\psi}_1 \ \boldsymbol{\psi}_2 \ \cdots \ \boldsymbol{\psi}_m]$.

To prove this relation, let us assume for simplicity but without loss of generality that $\mathbf{U}$ is of dimensions 2×2 , so is $\mathbf{S}$

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \quad (\text{C2})$$

and that the forcing field consists of only two instantaneous vectors, $\boldsymbol{\psi}^{(0)}$ and $\boldsymbol{\psi}^{(1)}$,

$$\boldsymbol{\psi}^{(0)} = \begin{bmatrix} \psi_1^{(0)} \\ \psi_2^{(0)} \end{bmatrix}, \quad \boldsymbol{\psi}^{(1)} = \begin{bmatrix} \psi_1^{(1)} \\ \psi_2^{(1)} \end{bmatrix}. \quad (\text{C3})$$

In this simple case, from the convolution definition given by Eq. (B1):

$$\mathbf{U}^*(\mathbf{S}\boldsymbol{\psi}) = \begin{bmatrix} u_{11} & u_{12} & 0 & 0 \\ u_{21} & u_{22} & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} s_1 \psi_1^{(0)} \\ s_2 \psi_2^{(0)} \\ s_1 \psi_1^{(1)} \\ s_2 \psi_2^{(1)} \end{bmatrix} \quad (\text{C4})$$

$$= \begin{bmatrix} u_{11} & 0 \\ u_{21} & u_{11} \\ 0 & u_{21} \end{bmatrix} \begin{bmatrix} \psi_1^{(0)} \\ \psi_1^{(1)} \end{bmatrix} s_1 + \begin{bmatrix} u_{12} & 0 \\ u_{22} & u_{12} \\ 0 & u_{22} \end{bmatrix} \begin{bmatrix} \psi_2^{(0)} \\ \psi_2^{(1)} \end{bmatrix} s_2 \quad (\text{C5})$$

$$= \mathbf{U}_1^* \boldsymbol{\psi}_1 \ s_1 + \mathbf{U}_2^* \boldsymbol{\psi}_2 \ s_2 \quad (\text{C6})$$

$$= \begin{bmatrix} \mathbf{U}_1^* \boldsymbol{\psi}_1 & \mathbf{U}_2^* \boldsymbol{\psi}_2 \end{bmatrix} \mathbf{s} \quad (\text{C7})$$

Table 1 Ratios of the numbers of multiplications for different forcing vector lengths, L_F , based on Eq. (B22), where $L_G=72$, $L_{\text{opt}}=2^9$, and $n=408,622$. It is assumed here that $m=\text{ceil}(L_F/L_G)$ and $L_F=L_{\text{opt}}-L_G+1=441$

L_F (h)	60×24	120×24	180×24	240×24	300×24	360×24
p	4	7	10	14	17	20
"#*"eqB3/"#*"eqB16	7.6	9.4	10.2	9.9	10.3	10.6
L_F (h)	730×24	1460×24	2190×24	2920×24	3650×24	4380×24
p	40	80	120	159	199	239
"#*"eqB3/"#*"eqB16	11.0	11.1	11.2	11.2	11.2	11.2

Table 3 MATLAB function “conv_FFT,” which is used to compute the convolution of two matrices

```

1  function R=conv_FFT(G,F,sumcol,Lopt)
2  % To compute convolution through fft/iff.
3  % Inputs:
4  %      G:      a convolution kernel matrix
5  %      F:      an external forcing matrix
6  %      sumcol, an optional logical input, when it is true
7  %              (the default value), a summation over the
8  %              columns will be performed and the
9  %              output R will be column vector.
10 %      Lopt,   an optional input to specify the optimal
11 %              piece wise convolution length. If not input,
12 %              it will be computed by the function.
13 % Output: R:   the convolution response vector or matrix.
14
15
16 LF=size(F,1); [LG,n]=size(G);
17
18 if n~=size(F,2)
19     error('G and F should have the same number of columns.')
20 end
21
22 if nargin<3, sumcol=true; end
23
24 if nargin < 4
25     tab(1:19,1)=(24:12:240).'; tab(1,2)=2^7; tab(2:3,2)=2^8;
26     tab(4:6,2)=2^9; tab(7:12,2)=2^10; tab(13:19,2)=2^11;
27
28     if LG>=tab(1,1) && LG<=tab(1,end)
29         Lopt=interp1(tab(:,1), tab(:,2), LG);
30         Lopt=round(Lopt);
31     else
32         e=exp(1); a=LG; y0=7;
33         fun=@(y) 2.^(y)*log2(e)+(1-a)*y+(1-a)*(log2(e)+2*n/(2*n+1));
34         y0=fzero(fun, y0);
35         y0=round(y0);
36         Lopt=2^y0;
37     end
38 end
39
40 Lz=Lopt-LG; Z=zeros(Lz,n); Ghat=fft([G;Z]);
41 Lf=Lopt-LG+1; Lz=Lopt-Lf; Z=zeros(Lz,n);
42 m=ceil(LF/Lf); i0=(1:Lopt).'; k0=(1:Lf).';
43 imax=(m-1)*Lf+i0(end);
44
45 % pre-allocate memory
46 if sumcol, R(imax,1)=0; else, R(imax,n)=0; end
47
48 % calculate the convolution
49 for p=1:m
50     disp(m-p)
51     k=(p-1)*Lf+k0; i=(p-1)*Lf+i0;
52     if p==m
53         k=(k(1):LF).';
54         Z=zeros(Lopt-numel(k),n);
55     end
56     Fhat=fft([F(k,:);Z]);
57     if sumcol
58         R(i)=R(i)+ifft(sum(Ghat.*Fhat,2));
59     else
60         R(i,:)=R(i,:)+ifft(Ghat.*Fhat);
61     end
62 end
63 L=LF+LG-1;
64 R=R(1:L,:);

```

where $\mathbf{U}_1$ and $\mathbf{U}_2$ are the first and second columns of $\mathbf{U}$, and ψ_1 and ψ_2 are defined by:

$$\psi_1 = \begin{bmatrix} \psi_1^{(0)} \\ \psi_1^{(1)} \end{bmatrix}, \quad \psi_2 = \begin{bmatrix} \psi_2^{(0)} \\ \psi_2^{(1)} \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}. \quad (\text{C8})$$

These are vectors in time, whereas those defined in Eq. (C3) are vectors in space. The same procedure demonstrated by this simple example can be used to prove the general relationship of Eq. (C1) by mathematical induction.

Let

$$\mathbf{C} = [\mathbf{U}_1 * \psi_1 \quad \mathbf{U}_2 * \psi_2 \quad \cdots \quad \mathbf{U}_m * \psi_m] = \mathbf{U} * \Psi. \quad (\text{C9})$$

Applying the convolution theorem to the column-wise convolutions in Eq. (C9), we obtain:

$$\mathbf{C} = \text{ifft}(\hat{\mathbf{U}} .* \hat{\Psi}) \quad (\text{C10})$$

where $\hat{\mathbf{U}} = \text{fft}(\mathbf{U})$ and $\hat{\Psi} = \text{fft}(\Psi)$ with the necessary zero paddings. The same MATLAB function shown in Appendix 2.B5 can be used to compute $\mathbf{C}$ when $\mathbf{U}$ and Ψ are input as the first and second arguments to the function, and when the third logical argument, *sumcol*, is set to be false.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give

appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Arakawa A and Lamb VR (1977). Computational design of the basic dynamical process of the UCLA general circulation model. *Methods Comput Phys.* 17, Academic Press, 173–265
- Courant R, & Hilbert D. (1962) *Methods of mathematical physics* (vol. ii): partial differential equations. Interscience. New York
- Leendertse JJ (1967) Aspects of a computational model for long-period water-wave propagation (p. 165). Rand Corporation, Santa Monica
- Myint-U T, Debnath L (2007) *Linear partial differential equations for scientists and engineers*. Birkhäuser, Boston
- Seber GA, and Lee AJ (2012) *Linear regression analysis* (Vol. 936). John Wiley & Sons
- Sielecki A (1968) An energy-conserving difference scheme for the storm surge equations. *Mon Wea Rev* 96:150–156
- Strang G (1986) *Introduction to applied mathematics*. Wellesley-Cambridge Press, Wellesley
- Strang G (2007) *Computational science and engineering*. Wellesley-Cambridge Press, Wellesley
- Xu Z (2015) The all-source Green's function (ASGF) and its applications to storm surge modeling, part I: from the governing equations to the ASGF convolution. *Ocean Dynamics*. doi:10.1007/s10236-015-0893-z
- Xu Z, J-P Savard and D Lefavre (2015) Data assimilative hindcast and climatological forecast of storm surges at Sept-Îles in the Estuary of Gulf of St. Lawrence. *Atmosphere and Ocean*. doi:10.1080/07055900.2015.1079774